Universidad ORT Uruguay

Facultad de Ingeniería

Estudio de factibilidad del uso de *Machine Learning* con múltiples fuentes de datos en el pronóstico del tiempo.

Entregado como requisito para la obtención del título de Ingeniero en Sistemas.

Natalie Gnoza – 150375

Marcelo Barberena – 92571

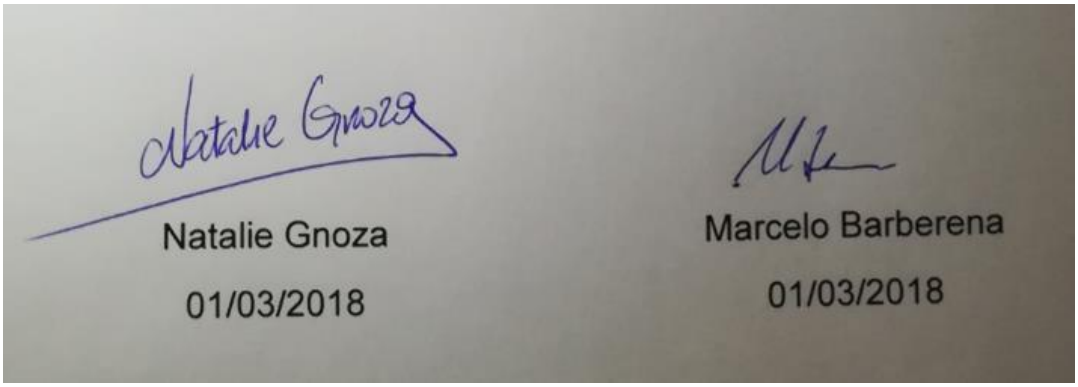
Tutor: Sergio Yovine

2018

Declaración de autoría

Nosotros, Natalie Gnoza y Marcelo Barberena, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el Proyecto de Grado;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Agradecimientos

Al meteorólogo Pablo Leites del Instituto Uruguayo de Meteorología por los datos brindados y su buena voluntad de colaboración hacia el proyecto.

A Libertad Tansini, Ignacio Chiazzo, Guillermo Leopold y Felipe García de la Facultad de Ingeniería de la Universidad de la República quienes nos brindaron amablemente información sobre su proyecto de grado que mucho tiene que ver con el nuestro ya que abordamos una temática similar.

A Max Patissier socio director de *Sinergia Tech* por brindarnos asesoramiento en la construcción de la PWS e informarnos de los proyectos relacionados en los que está trabajando.

Al consultor de IBM en analítica de datos y plataforma DSX Diego Aguirre, quien nos brindó asesoramiento sobre la plataforma Bluemix.

A Franz Mayr por ayudarnos a mejorar el modelo predictivo y a nuestro tutor Sergio Yovine por guiarnos en este largo proceso.

A Ing. Gonzalo Garat por ayudarnos en la integración y comunicación de los componentes de la mini estación meteorológica.

A nuestras familias y amigos, en particular a Nicolás Peri y Lorena Paola López por soportar la tempestad y oficiar de cocineros, cadetes, choferes y psicólogos.

Abstract

Este trabajo de investigación trata del estudio de factibilidad de la utilización de *Machine Learning* en el pronóstico del tiempo.

La idea subyacente es recopilar datos de diferentes fuentes, como ser la información provista por APIs, la generada por un prototipo de mini estación meteorológica desarrollado mediante *Arduino* y datos históricos proporcionados por el Instituto Uruguayo de Meteorología. Para posteriormente alimentar un modelo predictivo diseñado aplicando técnicas y algoritmos de *Machine Learning* que a partir de las mediciones de humedad, presión y temperatura realicen predicciones de la variable precipitación.

Se presenta una exposición que resume los objetivos actuales del proyecto, indicadores de logros verificables, estudio del estado del arte, contexto tecnológico, conclusiones y futuros pasos.

También se describe detalladamente la aplicación desarrollada en Azure, modelos predictivos y el desarrollo de la mini estación con Arduino.

En conclusión, se demuestra que el abordaje del tema del pronóstico del tiempo a través de estas técnicas es perfectamente viable y que se requiere seguir avanzando en el estudio, para poder concluir si es posible mejorar las predicciones del clima.

Glosario

- FI – Facultad de Ingeniería
- PWS - *Personal Weather Station*
- ML – *Machine Learning*
- IoT – *Internet Of Things*
- GPS – *Global Positioning System*
- INUMET – Instituto Uruguayo de Meteorología
- IBM - *International Business Machines*
- JSON - *JavaScript Object Notation*
- XML - *eXtensible Markup Language*
- HTML - *HyperText Markup Language*
- REST – *Representational State Transfer*
- USA – *United State of America*
- WiFi – *Wireless Fidelity*
- Led – *Light emitting diode*
- CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas
- IEEM – Escuela de negocios de Montevideo
- UM – Universidad de Montevideo
- UTC – *Universal Time Coordinated*
- MQTT - *Message Queue Telemetry Transport*
- HTTP - *Hypertext Transfer Protocol*
- BI – *Business Inteligence*
- HAARP- *Hig Frequency Active Auroral Research Program*
- MVOTMA - Ministerio de Vivienda Ordenamiento Territorial y Medio Ambiente
- SNRCC - Sistema Nacional de Respuesta al Cambio Climático
- OACI - Convención de Aviación Civil Internacional
- SINAE - Sistema Nacional de Emergencias
- DACC - Desarrollo y Adaptación al Cambio Climático
- ECMWF - *European Centre for Medium-Range Weather Forecasts*
- UKMO - *United Kingdon Model*
- NASA – *National Aeronautics and Space Administration*
- WRF - *Weather Research and Forecast Model*
- NCAR - *National Center of Atmospheric Research*
- NOAA - *National Oceanic and Atmospheric Administration*
- ISO – *International Organization for Standarization*
- GFS – *Forecast of Vertical Velocity and Precipitation*
- CRM – *Customer Relationship Management*
- ERP – *Enterprise Resource Planning*
- USB – *Universal Serial Bus*
- SMS – *Short Message Service*
- ENIAC – *Electronic Numerical Integrator And Computer*
- NOGAPS – *Navy Opertational Global Atmospheric Prediction System*

Palabras clave

Personal Weather Station, *Machine Learning*, *Internet Of Things*, Estación Meteorológica, Instituto Uruguayo de Meteorología, *Big Data*, Aprendizaje Automático, Predicción, Precipitaciones, Modelo Predictivo.

Índice

1. Introducción 9
2. Objetivos 10
2.1 Del proyecto 10
2.2 Específicos 10
2.3 Indicadores de logro verificables 11
2.4 Metodología elegida para el proyecto 12
3. Acerca del tiempo 13
3.1 Primeros pronósticos 13
3.2 Problemática actual 15
3.3 ¿Qué pasa en Uruguay? 16
3.5 Principales variables meteorológicas 19
3.6 Modelos matemáticos utilizados hoy en día para realizar predicciones . 34
4. Estado del arte 38
4.1 Contexto tecnológico 38
4.2 Trabajos de aplicación de *Machine Learning* en el pronóstico del tiempo 61
4.3 Aporte de nuestro proyecto en el citado contexto 77
5. Análisis, diseño e implementación de la solución 80
5.1 Plataforma 80
5.1.1 Solución de alto nivel y componentes 80
5.1.2 Descripción de la arquitectura 82
5.1.2.1 Vistas de Módulos 82
5.1.2.2 Vistas de *Layers* 84
5.1.2.3 Vistas de Componentes y conectores 84

5.1.2.4 Vistas de Asignación 89
5.1.3 Primer prototipo en IBM Cloud 91
5.1.4 Aplicaciones de capturas de datos en Azure 95
5.1.5 Mini estación meteorológica 103
5.2 Analítica 107
5.2.1 Análisis de fuentes de datos 107
5.2.2 Obtención de los datos 109
5.2.3 Análisis de datos exploratorios 110
5.2.4 Separación del *dataset* en *train*, *test* y *validation* 128
5.2.5 Construcción de los modelos predictivos 128
5.2.5.1 Construcción modelo predictivo discreto 129
5.2.5.2 Construcción modelo predictivo continuo 138
6. Conclusiones 144
7. Referencias bibliográficas 148
ANEXO 1 – Planificación del Proyecto 155
ANEXO 2 – Mini estación meteorológica 160
ANEXO 3 - Modelos de *Machine Learning* utilizados 162
ANEXO 4 – Cartas compromiso y presentación INUMET 189
INFORME DE LOS CORRECTORES 191

1. Introducción

Este proyecto de grado tiene como fin realizar el estudio de la factibilidad del uso de *Machine Learning* con múltiples fuentes de datos en la predicción del clima.

En una primera instancia focalizamos esfuerzos en el análisis y obtención de datos públicos históricos y actuales. Para ello recurrimos a fuentes como IBM, INUMET y diferentes APIs disponibles.

En una segunda etapa, nos enfocamos en el procesamiento de los datos obtenidos, mejorar su calidad y almacenarlos de modo tal que resulten de utilidad para la creación de los modelos predictivos. Logramos una base de datos con registros de lecturas de diferentes variables meteorológicas para el periodo que va del 01/01/2012 hasta la actualidad ya que las APIs continúan capturando datos de forma continua cada una hora.

En la última etapa trabajamos en la construcción del modelo predictivo, para eso se estudiaron diversas técnicas y algoritmos de aprendizaje automático, seleccionando aquellos que tengan mayor precisión y menor error sobre los datos de prueba.

El software construido recibe el nombre de *Kairos Weather App* en honor al dios griego del clima y las estaciones, quien es el hijo menor de Zeus y Tique y por tanto nieto de Cronos. *Kairós* es representado con un mechón de pelo en la parte delantera de su cabeza, alas y con una balanza (desequilibrada) en su mano izquierda.

Logo aplicación *Kairós*

2. Objetivos

2.1 Del proyecto

Disponer de la posibilidad de realizar un trabajo de I+D en disciplinas nuevas (*Machine Learning, IoT* y *Big Data*) que están teniendo actualmente un gran impulso en la comunidad técnica debido al gran potencial de aplicación que tienen en diferentes áreas del conocimiento.

Incursionar en técnicas y metodologías utilizadas en los equipos de investigación de la Facultad de Ingeniería enriqueciendo la experiencia profesional del equipo de proyecto.

Desarrollar un prototipo (herramienta predictiva y PWS) para estudiar la factibilidad de uso para mejorar las predicciones meteorológicas en Uruguay.

2.2 Específicos

El trabajo que se realizó consiste en la realización de un estudio del estado del arte de las técnicas y algoritmos disponibles en la actualidad identificando los escenarios de aplicabilidad de esta tecnología.

Asimismo, se propone el desarrollo de un prototipo de *software*/*hardware* con el fin de estudiar la factibilidad del uso de *Machine Learning* utilizando diversas fuentes de datos tales como información histórica de INUMET*,* IBM*, Weather* APIs*,* PWS y eventualmente contenido publicado en redes sociales, para el pronóstico del tiempo (ver Figura 1).

A partir de variables como Temperatura, Humedad y Presión Atmosférica se pronosticará con cierto grado de precisión el valor de la variable Precipitaciones.

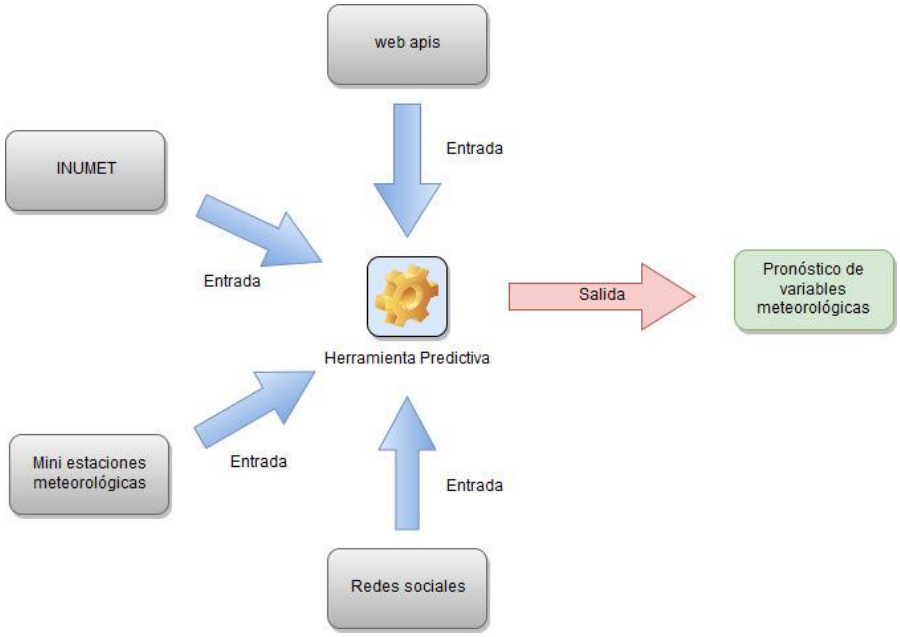


Figura 1 - Esquema del sistema a desarrollar

Esto involucra el diseño y la implementación de un *software* que permita la recopilación de datos históricos de distintas variables meteorológicas (Temperatura, Humedad, Presión, Velocidad, Dirección del viento, etc) de diferentes fuentes, la creación del modelo predictivo, y la presentación de los datos almacenados y resultados obtenidos de una manera fácil de comprender.

Mediante la utilización de tecnología Arduino se plantea la construcción de un prototipo de mini estación meteorológica de bajo costo (o PWS) cuyo objetivo es alimentar el modelo predictivo con datos meteorológicos captados a escala "local".

2.3 Indicadores de logro verificables

- Informe de estudio del estado del arte.
- Software de integración de diferentes fuentes con modelo predictivo aplicado al pronóstico del tiempo (*Kairos Weather App*).
- Prototipo de mini estación meteorológica desarrollado con Arduino.

2.4 Metodología elegida para el proyecto

Dadas las características de proyecto de investigación y tomando en cuenta que la incertidumbre es un factor clave que se debe gestionar de forma adecuada, se optó por un enfoque incremental e iterativo con el objetivo de tener retroalimentación temprana y de minimizar los riesgos.

Para ello se propuso realizar primeramente pruebas de concepto en las áreas de *expertise* o temas tecnológicos en los que identificamos un mayor riesgo, como lo son la generación de la aplicación de captura de datos primeramente realizada en Bluemix de IBM, luego en Azure, creación del modelo predictivo y el ensamblado de la mini estación meteorológica.

En el Anexo 1 podrá ser consultada la planificación del proyecto para cada una de sus tareas.

3. Acerca del tiempo

Los fenómenos climáticos son una constante en el mundo y han estado ocurriendo durante milenos. En la antigüedad diversas civilizaciones observaron la naturaleza y tomaron registro de las estaciones, notaron que el sol es de vital importancia para el crecimiento de cultivos y hasta lo adoraron como un dios.

3.1 Primeros pronósticos

En torno al año 340 *a.C.* aparece por primera vez en el libro publicado por Aristóteles el término “meteorología” (del griego *meteoros* que significa “alto en el cielo”), allí estaban contenidas observaciones y algunas especulaciones en torno al origen de ciertos fenómenos que se presentaban en la atmósfera.

Posteriormente un alumno de Aritstóteles llamado Teofrasto publica la obra “*On Winds and on Weather Signs*” [62] enfocada principalmente en pronósticos, aunque basada totalmente en la de su maestro. Cerca del 580 *a.C*. Thales de Mileto (624 - 546 *a.C*.) hizo el primer pronóstico de cosecha estacional de olivas; para eso reservó en baja temporada todos los molinos de olivas y los sub alquiló por mayor precio cuando la cosecha estuvo pronta.

Por muchos años la meteorología estuvo dormida sin instrumentos adecuados para medir, mejorar predicciones y explicar correctamente las estaciones. Hacia fines de la Edad Media en 1543, aparece la revolucionaria y hereje teoría heliocéntrica propuesta por Nicolás Copérnico, dejando en evidencia que la Tierra no era el centro del universo.

Se lograron múltiples progresos en el área meteorológica, desarrollándose nuevos y mejores instrumentos. En 1480 Leonardo Da Vinci crea una versión del anemómetro (si bien ya había sido inventado por Leon Battista Alberti en 1450), también de un higrómetro para conocer “la calidad del aire cuando va a llover”. Consistía en una balanza con una sustancia higroscópica (algodón) en uno de los platos y una sustancia que no lo es en el otro (bola de cera) tal como lo muestra la Figura 2.

Figura 2 – Higrómetro de Leonardo Da Vinci
Fuente: [63]

Galileo Galilei en 1593 inventa el termoscopio, un dispositivo antecesor al termómetro que muestra cambios en la temperatura. Constaba de un tubo de vidrio con una ampolla en la parte superior, en el que líquido sube o baja en función de la variación de temperatura. A comienzos del siglo XVII el médico italiano Santorio Santorio le añade una escala, un siglo después se normaliza su diseño cuando Gabriel Fahrenheit y Anders Celsius establecen una escala en función de los puntos de ebullición y congelación del agua (siglo XVIII).

Evangelista Torricelli, estudiante de Galileo crea en 1644 el primer barómetro de mercurio para medir la presión atmosférica. Blaise Pascal y Renes Descartes encuentran una relación de dependencia entre la presión atmosférica y la altura, pues si el aire tiene peso, ejerce menos presión a medida que nos elevemos.

Ya en el nuevo mundo John Campanius Holm durante los años 1644 y 1645 tomó registro de observaciones meteorológicas en Nueva Suecia cerca de Wilmington, Delaware; puede ser considerado como el primer meteorólogo. [28]

John Dalton en 1793 escribe en su “Ensayos y Observaciones Meteorológicas”, que la temperatura afecta la cantidad de vapor de agua que hay en el aire, es lo que hoy conocemos como humedad relativa.

A principios del siglo XX se dan los primeros pasos en la previsión del tiempo utilizando modelos numéricos por parte de Lewis Fry Richardson quien publica "*Weather prediction by numerical process*". Allí se detalla la forma en la cual eliminar variables poco importantes para la aplicación de ecuaciones de dinámica de fluidos, pero aún no se disponía del poder de computo necesario.

Edward Lorenz en los años 60s, expone en su teoría del caos (efecto mariposa) la complejidad y dinamismo de la atmósfera, en la cual pequeñas variaciones en las condiciones iniciales repercuten en grandes diferencias en el futuro, es por eso que se dificulta enormemente la predicción a largo plazo.

Luego comienzan a aparecer los satélites meteorológicos (*TIROS-1)* equipados con instrumentos necesarios para estudiar diversos fenómenos y se fundan diversas organizaciones a nivel mundial vinculadas al clima.

3.2 Problemática actual

El clima afecta la vida humana en múltiples aspectos; si hablamos de la economía de un país, las lluvias pueden salvar plantaciones de la sequía, pero en demasía las pueden arruinar, los huracanes pueden alejar turistas de sitios claves para el turismo. El clima es capaz de incidir en el desenlace de una guerra tal como fue el caso de Napoleón en su intento de conquista a Rusia en 1812 cuando cruzó su frontera con más de 600.000 soldados y 50.000 caballos pretendiendo llegar a Moscú. Los rusos quemaron campos y casas en su retirada, pero como el clima permaneció seco y cálido, las tropas francesas llegaron en setiembre con hambre, agotamiento y sin suministros lo cual provocó más de 20.000 muertes por fatiga y enfermedades. En el invierno de ese año en plena retirada de los franceses el frío terminó de matar a los más débiles, quedando tan solo 100.000 hombres con vida. [28]

En la Segunda Guerra Mundial, los aliados eligieron el 5 de junio (Día D) entre tres posibles días de ese mes, de modo tal que los factores climáticos favorecieran el operativo y fuese exitoso. Para eso, la marea baja debía coincidir con el amanecer, la luna debía ser llena creciente, la visibilidad mínima de 5Km

para que los artilleros navales vean su objetivo; el viento no podía superar los 13 a 19 Km/h en tierra y 21 a 29 Km/h en el mar. Debido a cambios en el clima inesperados, Eisenhower retrasó la invasión un día (6 de junio de 1944). [28]

El clima es algo difícil de pronosticar con certeza, los fenómenos extremos son cada vez más frecuentes e impredecibles. Ciclones, tornados, huracanes, maremotos, lluvias y la erosión del suelo provocan inundaciones, que terminan siendo la causa mayor de mortalidad pues favorecen la aparición y propagación de enfermedades.

Quizás en un futuro cercano, con los avances en la tecnología hasta sea posible controlar el clima y utilizarlo como arma, podemos poner como ejemplo el proyecto HAARP (*Hig Frequency Active Auroral Research Program*). Financiado por la marina y fuerza aérea de EEUU, HAARP surge con el fin de estudiar las características de la ionósfera (parte ionizada de la atmósfera terrestre) y mejorar las radiocomunicaciones para la detección temprana de misiles. Las operaciones se llevaron a cabo en Gakona (Alaska) desde el año 1993 utilizando como dispositivo el IRI (Instrumento de Investigación Ionosférico), un radiotransmisor muy potente de alta frecuencia (180 antenas repartidas en 14 hectáreas) que puede modificar las propiedades electromagnéticas en una zona limitada de la ionósfera. Hasta el año 2008 el presupuesto de HAARP era de 250 millones de dólares y su impacto fue medido mediante instrumentos de radiofrecuencia UHF, VHF y magnetómetros. Los rusos por su parte han trabajado en un proyecto similar llamado Sura, con la salvedad de que es unas cincuenta veces más potente que HAARP (3.6 MW), incluso algunas teorías afirman que son los responsables de una amplia gama de eventos y desastres naturales. [64]

3.3 ¿Qué pasa en Uruguay?

En 1920 el Servicio Meteorológico del Uruguay fue una realidad funcionando en la Facultad de Humanidades y Ciencias que tiempo después, se llamaría Instituto Nacional para la Previsión del Tiempo. Los meteorólogos a cargo, Eleazar Giuffra y Juan María Bergeiro comenzaron a trabajar en técnicas de pronóstico del tiempo basándose en el análisis de mapas sinópticos y estadísticas con amplios

beneficios para la navegación aérea y marítima, así como también al sector agropecuario.

En 1944 se crea dentro del Servicio Meteorológico del Uruguay, la Escuela de Meteorología del Uruguay (EMU) la cual se encarga aún de la formación de recursos expertos en el tema. En 1950, se suma como integrante a la Organización Meteorológica Mundial, prestando servicios al Aeropuerto Internacional de Carrasco, por ese entonces recientemente inaugurado.

En 1970 el Servicio Meteorológico del Uruguay muda su sede a Javier Barrios Amorín 1488 (Montevideo); pasa a denominarse Dirección General de Meteorología para luego en 1979 llamarse Dirección Nacional de Meteorología bajo la dependencia del Ministerio de Defensa.

El 20 de mayo de 2009 mediante el Decreto del Poder Ejecutivo 238/09 se crea el Sistema Nacional de Respuesta al Cambio Climático (SNRCC) a cargo del Ministerio de Vivienda, Ordenamiento Territorial y Medio Ambiente en colaboración con otros ministerios y entidades como el Sistema Nacional de Emergencias (SINAE). Tiene como objetivo la prevención de riesgos, mitigación y adaptación al cambio climático.

El *SNRCC* dio apoyo a la creación del Instituto Uruguayo de Meteorología (INUMET) quien es en nuestro país la autoridad meteorológica y aeronáutica en aplicación de la Convención de Aviación Civil Internacional (OACI). Se creó en base a la Dirección Nacional de Meteorología el 25 de octubre de 2013 bajo la Ley Nº 19158.

Diversos son los organismos que trabajan en conjunto para prevenir y mitigar los riesgos del cambio climático en nuestro país. El Ministerio de Vivienda Ordenamiento Territorial y Medio Ambiente (MVOTMA) elabora planes quinquenales para realojar familias en zonas inundables o contaminadas. El Ministerio de Ganadería Agricultura y Pesca (MGAP) también se encuentra implementando proyectos tales como el “Desarrollo y Adaptación al Cambio Climático” (DACC) apoyado por el Banco Mundial, así como el “Ganaderos

Familiares y Cambio Climático" apoyado con 10 millones de dólares por el Fondo de Adaptación del Protocolo de Kioto. Este último tiene como objetivo crear capacidad a nivel nacional para la adaptación al cambio climático, con miras en familias de productores ganaderos de pequeña escala ubicados en zonas vulnerables.

Sin duda por los eventos extremos que Uruguay está sufriendo últimamente, se constata que el cambio climático es una realidad. Para ello, veamos algunos datos:

- Inundaciones de 1959, 45.000 dejaron evacuados,
- Inundaciones de 2007, tuvieron un impacto económico de 20 millones de dólares para las zonas de Durazno, Soriano y Treinta y Tres.
- En los últimos diez años, la cifra de evacuados a causa de las inundaciones ronda entorno a las 67.000 personas.
- Tornado de Fray Marcos (Florida) de 1970, dejó 11 muertos.
- Tornado de Dolores de 2016, los vientos alcanzaron entre 180 y 330 Km/h. No se predijo y en solo 3 minutos, provocó 5 muertes, cientos de heridos y millonarias pérdidas materiales (30 millones de dólares).
- Se han contabilizado unos 37 tornados entre 1968 y 2017 según el Sistema Nacional de Emergencias (SINAE).

Es cierto que si bien se están haciendo esfuerzos por mejorar la realidad es, que en nuestro país es muy escasa la inversión, muy bajo el presupuesto sumado a que existe una falta de tecnología y personal calificado que hacen aún más difícil la tarea. A la carrera de licenciatura en Ciencias de la Atmósfera, de los 20 alumnos que se inscriben por año, solo consiguen egresar 10. Muchos estudiantes cursan apenas año y medio de carrera pues esto ya los habilita a trabajar como observadores.

INUMET no es ajeno a esto, la falta de personal y recursos ha puesto en evidencias las carencias y conflictos por los que ha atravesado, al punto tal que en el año 2017 fue necesario decretar su esencialidad. Cuenta con 21 estaciones de observación en todo el país, pero sólo tres de ellas funcionan las 24hs

(Aeropuerto de Melilla, Laguna del Sauce y Rocha). Muchas estaciones en las que trabajaban observadores, debido a la falta de personal o por licencias redujeron su trabajo a la mitad (Rivera, Colonia y Paso de los Toros), otras solo están operativas 6 horas al día (Trinidad, Young y Tacuarembó). El caso de Bella Unión es aún peor, tuvo que cerrar porque el personal se jubiló y no se lograron llenar las vacantes.

El INUMET según los entendidos, cuenta con un atraso de 30 años por falta de inversión, sólo se destina el 1% de su presupuesto. El 90% del mismo está destinado a pago de salarios mientras que el 9% que resta a gastos operativos. Existen estaciones automáticas funcionando las 24 horas en Colonia, Laguna del Sauce, Paso de los Toros, Rocha, Rivera y Carrasco; aunque no son las adecuadas.

La mayor carencia del instituto es no disponer de un radar Doppler, dado que hubiese permitido predecir antes de que toque tierra, con entre 10 y 20 minutos el tornado que ocurrió en Dolores. El radar más cercano se encuentra en Buenos Aires en el Aeropuerto de Ezeiza y si bien es utilizado, no tiene alcance a todo el territorio nacional, ni siquiera logra cubrir Montevideo.

Los radares se valen del efecto Doppler para detectar tornados, granizadas y hasta lluvias muy abundantes que pueden dar lugar a posteriores inundaciones. Son útiles para contrastar información satelital con lo que ocurre en tierra, sumado a la gran capacidad de procesamiento que le permite hacer predicciones a corto plazo, la única desventaja es que su precio ronda el millón de dólares. Uruguay es el único país en la región que no tiene un radar de este tipo, tampoco globo sonda para obtención de datos a grandes alturas.

3.5 Principales variables meteorológicas

Cerca del 90% de la humedad de la atmósfera proviene de los océanos, éstos a su vez albergan aproximadamente el 97% del agua de la Tierra. El agua se recicla permanentemente desde el océano hacia el aire mediante la evaporación por calentamiento de la superficie de mar (océanos), el aire caliente sube y se

une a otras moléculas de agua a través de la troposfera. Una vez que el vapor alcanza las capas más frías de la atmósfera se condensa alrededor de pequeñas partículas de polvo hasta volverse lo suficientemente grandes para comenzar a descender en forma de lluvia, nieve o granizo.

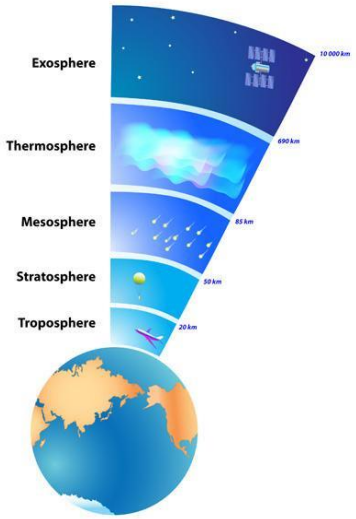


Figura 3 – Capas de la atmósfera terrestre
Fuente: [65]

Las capas superiores de la tropósfera son frías y secas por tanto el vapor de agua no permanece líquido por mucho tiempo, las nubes allí están compuestas de cristales de hielo. Cirrus es el tipo de nube más común que se puede encontrar a una altura de entre 6 y 18 Km, se asemejan a finos filamentos que se estiran por acción del viento, son de color blancas y transparentes. Se forman antes de que llegue un frente frío o tormenta y se mueven generalmente de oeste a este, predicen un sistema de baja presión lo cual da indicios de posibles lluvias en las próximas doce o veinticuatro horas.

Existen otros indicadores de precipitaciones en las nubes medias, por ejemplo, los altocúmulos y altoestratos (de color gris o gris azulado). Mientras que para las capas inferiores los estratos o nimbostratos pueden producir llovizna y niebla; los cumulonimbos producen rayos, tormenta eléctrica, granizo y tornados (color rojo brillante en el radar) generalmente cuando hay alta humedad y fuertes corrientes ascendentes.

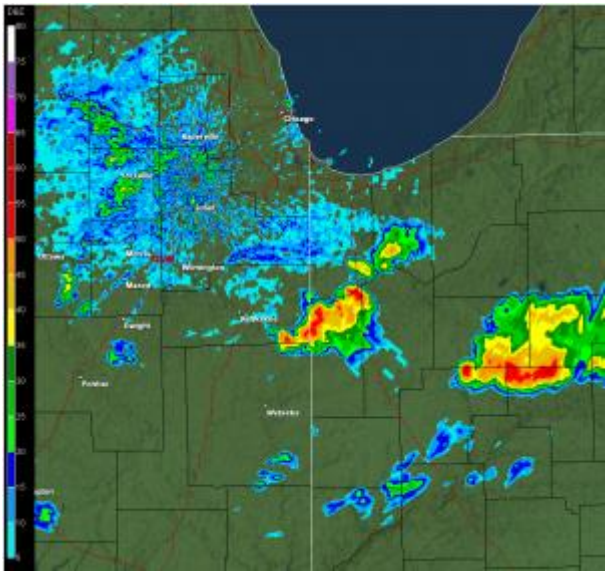


Figura 4 – *Cumulonimbus* en rojo bajo la vista del radar
Fuente: [66]

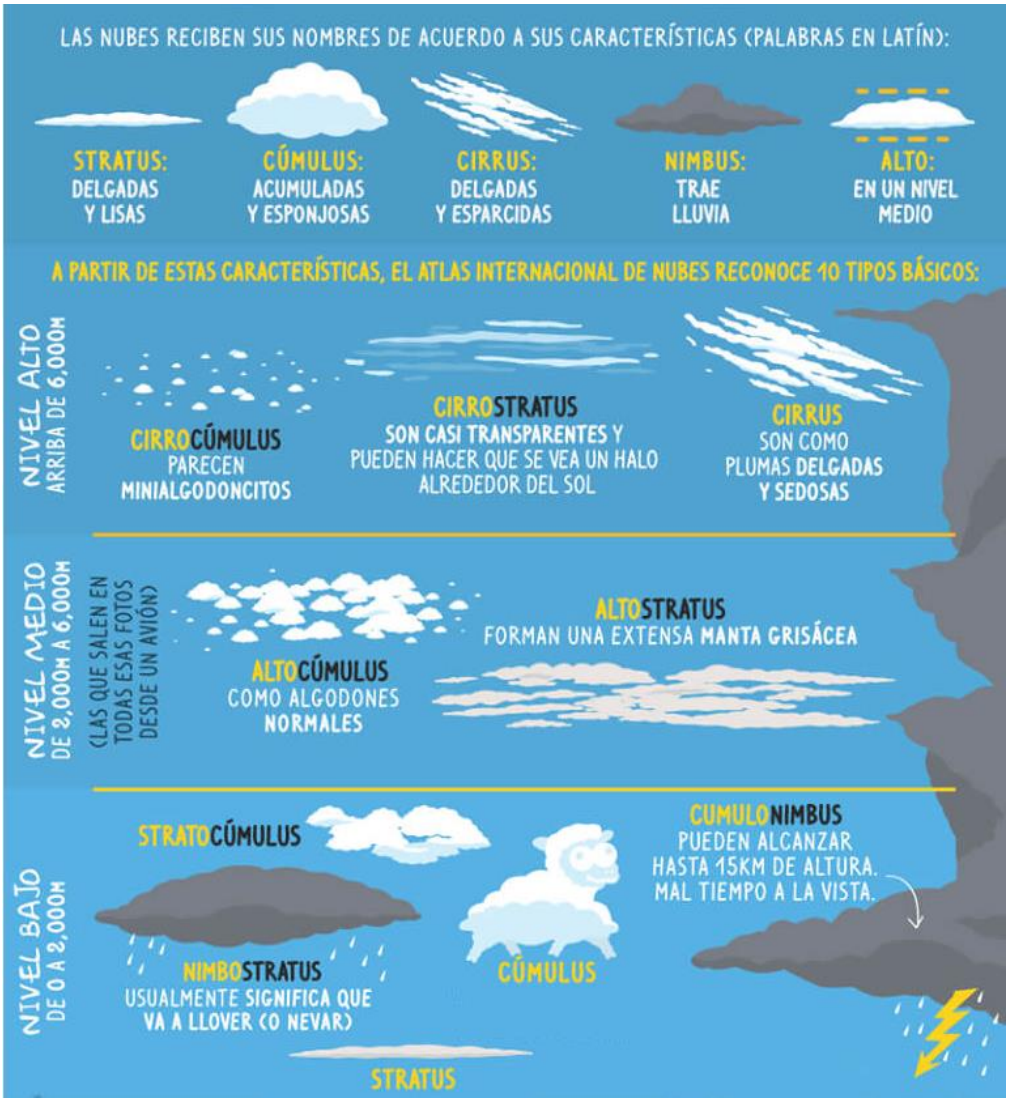


Figura 5 – Clasificación de nubes en la tropósfera
Fuente: [67]

Veamos ahora algunas de las principales variables meteorológicas que son de interés al proyecto y su relación:

PRESIÓN ATMOSFÉRICA

Se mide utilizando barómetros, en la antigüedad estaban llenos de líquido (mercurio) hoy en día se utilizan los aneroides, que miden variaciones/deformaciones sobre celdas de metal hechas generalmente de berilio y cobre.

Las unidades más comunes son el milibar (mb), hectopascal (hPa) y el milímetro de mercurio (mmHg). La presión a nivel del mar para una atmósfera estándar es de 1013.25 mb = 1013.25 hPa = 760 mmHg.

Figura 6 – Barómetro aneroide
Fuente: [71]

La presión disminuye cuanto más alto subamos en la atmósfera, el aire al nivel del suelo soporta más presión debido al peso del aire que está sobre él. Las líneas en espiral en los mapas del tiempo a nivel del mar (cartas sinópticas de superficie) que suelen tener la apariencia de huellas dactilares se llaman isobaras, ellas conectan regiones de igual presión de aire y son trazadas generalmente cada 4 hPa. En el centro encontramos las letras A o B que identifican áreas de alta o baja presión respectivamente. Las zonas de baja presión suelen mostrar buen tiempo, mientras que las de alta presión representan aire ascendente e inestable, lo que trae aparejada lluvia, nieve o algo aún peor. En la cercanía de un centro de baja presión, los vientos circulan en sentido horario en el hemisferio sur, mientras que en los de alta presión lo hacen en sentido anti horario (para el hemisferio norte es al revés). La intensidad del viento puede verse en la medida de cuan juntas o espaciadas estén las isobaras, cuanto más juntas, mayor es la intensidad de los vientos.

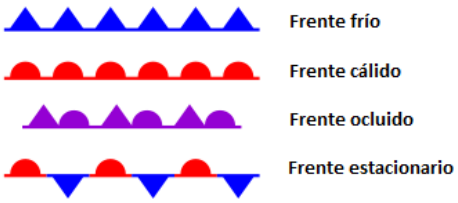


Figura 7 – Simbología carta sinóptica

Los frentes fríos se señalan con líneas dentadas de color azul junto la dirección de movimiento, indican un avance de una masa de aire frío que puede ocasionar lluvias y descensos de temperatura a su paso. Los frentes cálidos señalados por líneas bordeadas de semicírculos en rojo, indican el avance de una masa de aire cálido que puede traer lluvias y ascensos de temperatura. Los frentes ocluidos son una mezcla entre frentes cálido y frío, suelen estar asociados a fuertes lluvias. Por último, los estacionarios establecen un límite entre dos masas de aire (frio y caliente) en el cual ninguna de ellas es lo suficientemente fuerte como para sustituir a la otra, son más comunes en verano.

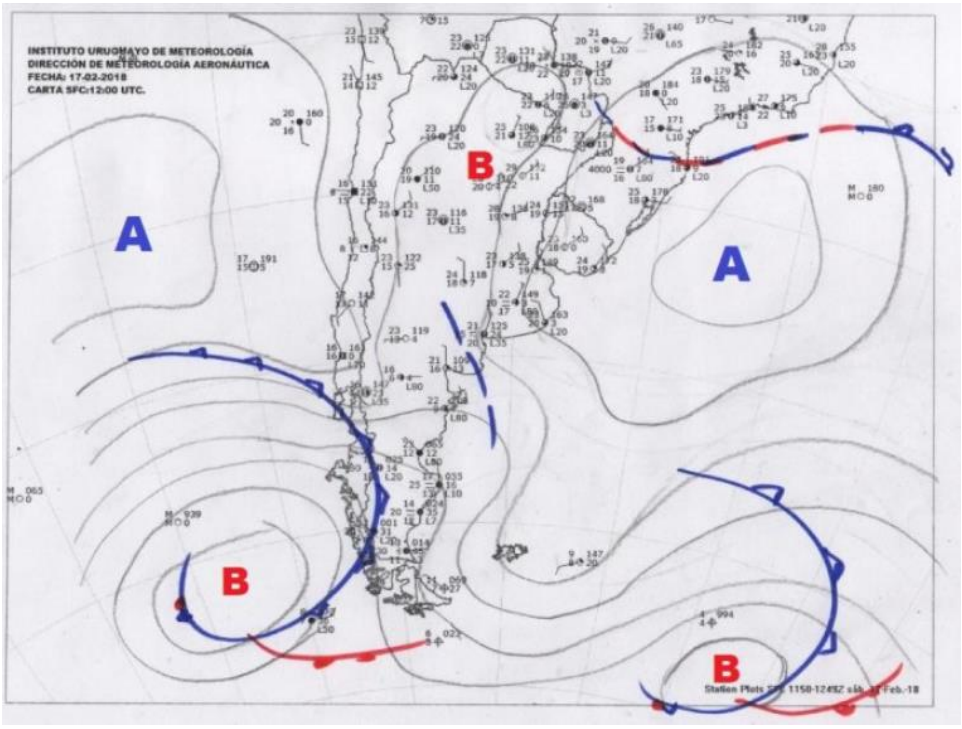


Figura 8 – Carta sinóptica región sur para 17 de febrero de 2018 – INUMET
Fuente: [29]

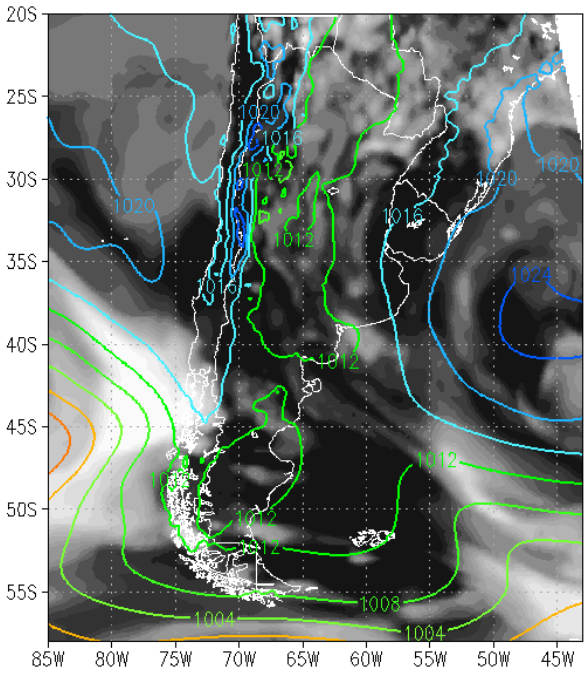


Figura 9 – Mapa isobárico región Uruguay para el 17 de Febrero 2018 – INUMET
Fuente: [29]

VIENTO

El viento es consecuencia del movimiento del aire en una dirección y velocidad específica. Se utilizan anemómetros para medir su velocidad, en meteorología se utilizan los de molinetes con aspas que giran por la fuerza del viento y así es posible medir la cantidad de vueltas que da en un determinado lapso de tiempo.

Figura 10 – Anemómetro de aspas utilizado en nuestro proyecto – Sparkfun
Fuente: [34]

Para determinar hacia donde sopla el viento se debe identificar las fuerzas que afectan el movimiento horizontal del aire:

- **Gradiente de Presión**

  Si a una misma altura tenemos zonas de baja y alta presión, el aire se moverá de la zona de alta presión a la de baja. Cuanto mayor sea la diferencia de presión mayor será el movimiento del aire en este desplazamiento.

- **Coriolis**

  Debido a la rotación de la Tierra sobre su eje, se produce una desviación inercial en los vientos hacia la izquierda en el hemisferio sur y a la derecha en el norte, lo cual hace que el viento tienda ser paralelo a las isobaras. Se opone a la fuerza de gradiente de presión. Fue descubierto en 1835 por el ingeniero y matemático francés Gaspard-Gustave de Coriolis.

Figura 11 – Fuerza Coriolis
Fuente: [69]

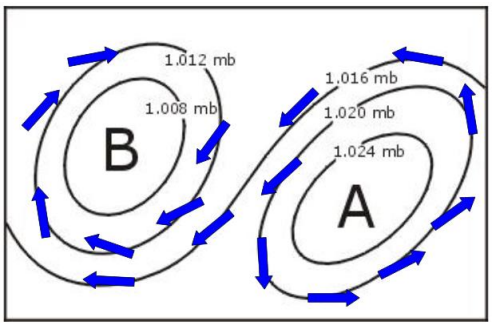


Figura 12 – Efecto de Coriolis en el hemisferio sur para centros de baja y alta presión
Fuente: [70]

- **Centrífuga**

  En isobaras curvas actúa radialmente hacia el exterior, de magnitud pequeña no tiene normalmente incidencia a menos en vientos fuertes en trayectorias curvas acelerando o desacelerando según dicha curvatura.

- **Fricción**

  Producida por el rozamiento del aire con la superficie de la Tierra, actúa aproximadamente hasta los 1000 metros de altura, por arriba de ese nivel puede considerarse insignificante su efecto fluyendo el viento en forma paralela a las isobaras.

La dirección del viento se representa según los puntos cardinales Norte (N), Sur (S), Este (E) y Oeste (O) o bien en grados acimutales del Norte (0 - 360º) siendo 0º Viento Norte, 45º viento Noreste, 90º Viento Leste, 135º Sudeste, 180º Sur, 225º Suroeste, 270º Oeste y 315º Noroeste.

Las unidades de medida utilizadas comúnmente son:

- Nudos (kn en ISO o kt por el inglés knot) donde 1 kt equivale a 1,852 km/h
- Kilómetros por hora (Km/h) donde 1 Km/h equivale a 0,27778 m/s
- Metros por segundo (m/s)
- Beaufort basada en el estado del mar, olas y fuerza del viento

Beaufort (Bft)	Viento km/h	Viento nudos
0	1	1
1	1-5	1-3
2	6-11	4-6
3	12-19	7-10
4	20-28	11-15
5	29-38	16-21
6	39-49	22-27
7	50-61	28-33
8	62-74	34-40
9	75-88	41-47
10	89-102	48-55
11	103-117	56-63
12	118-133	64-71

Figura 13 – Tabla Beaufort con equivalencias en Km/h y nudos

Los símbolos eólicos utilizados para representar dirección y velocidad del viento son los siguientes:

Símbolo	Descripción
	del S
	del SW
	del W
	del NW
	del N
	del NE
	del E
	del SE

Símbolo	Descripción
	0 - 1 Km/h
	2 - 9 Km/h
	10 - 19 Km/h
	20 - 28 Km/h
	29 - 37 Km/h
	38 - 46 Km/h

Símbolo	Descripción
	47 - 56 Km/h
	57 - 65 Km/h
	66 - 74 Km/h
	84 - 93 Km/h
	103 - 111 Km/h
	121 - 130 Km/h

Figura 14 – Simbología dirección e intensidad del viento en Km/h

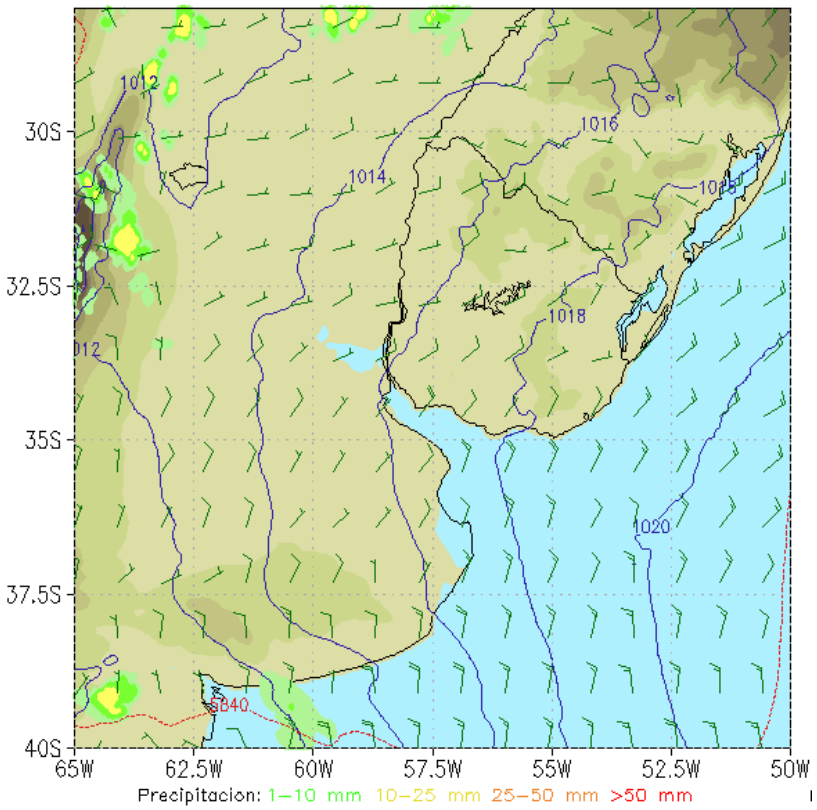


Figura 15 – Mapa presión a nivel medio del mar (hPa), viento a 10 metros (kt) y precipitación acumulada de 6hs (mm) para el 17 de febrero de 2018 – INUMET
Fuente: [29]

HUMEDAD

La humedad relativa es el porcentaje de saturación de un cierto volumen de aire a una temperatura determinada, ésta depende de la presión y temperatura del aire que se analiza. Es un indicativo de la cantidad de vapor de agua que es transportada por el aire, está directamente vinculado al desarrollo de nubes y precipitaciones.

Es una medida porcentual que va desde 0 (seco) a 100% (saturado) y cuanto mayor sea la temperatura de una masa de aire mayor será su capacidad de albergar vapor de agua por tanto la humedad relativa aumenta.

Se utilizan higrómetros, en la antigüedad utilizaban cabello humano o de animales pues son elementos higroscópicos, retienen humedad. A medida que su longitud variaba era registrado en una escala mediante un sistema de amplificador mecánico.

Los higrómetros más modernos usan la temperatura de condensación (punto de rocío) o cambios en la capacitancia para medir diferencias en la humedad. También existen los psicrómetros que miden la diferencia de temperatura entre un termómetro con bulbo seco y otro con bulbo húmedo.

Figura 16 – Higrómetro de tensión de cabello
Fuente: [68]

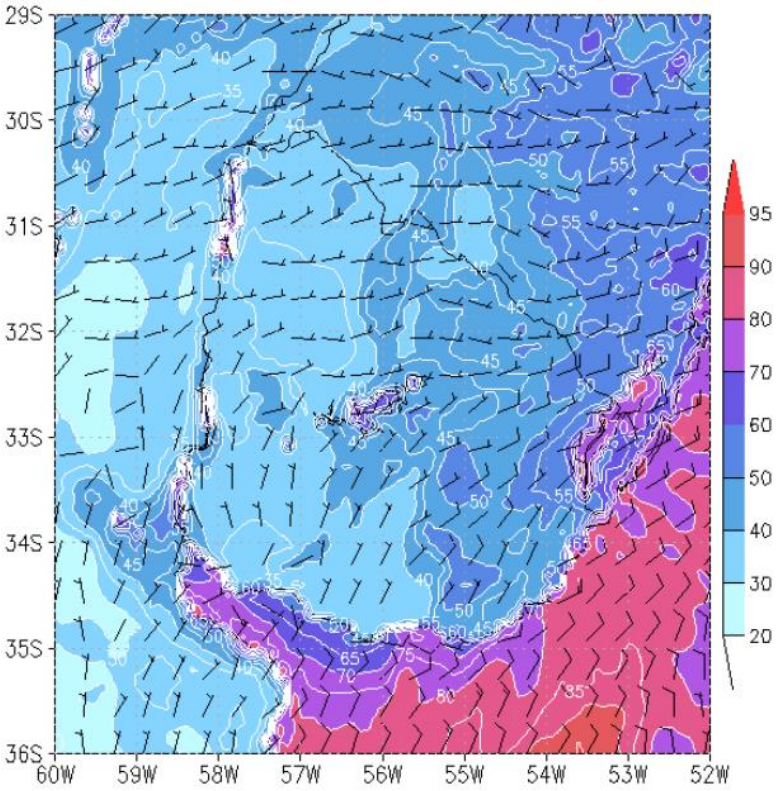


Figura 17 – Mapa de humedad relativa del aire a 2 metros (%) y viento a 10 metros de altura (kt) para el 17 de febrero de 2018 – INUMET
Fuente: [29]

TEMPERATURA

Hace referencia al grado de calor específico del aire en un lugar y momento determinado y su evolución en el tiempo y espacio en distintas zonas climáticas. Se utilizan termómetros para su medición, calibrados en función de múltiples escalas; según el Sistema Internacional de Unidades, la unidad de temperatura es el kelvin (K) cuya escala es la escala kelvin o absoluta. Es muy común también el uso de otras como la centígrada o Celsius o la Fahrenheit. Es una de las variables de mayor importancia en la caracterización del clima, veamos algunos conceptos relacionados:

- **Temperatura máxima**

  Es la mayor temperatura que el aire alcanza en un lugar para un cierto lapso (mensual, diario, anual). Por lo general las temperaturas máximas diarias son alcanzadas en las primeras

horas de la tarde, las máximas mensuales en enero o febrero en el hemisferio sur.

- **Temperatura mínima**

  Es la menor temperatura alcanzada para un determinado lugar en un cierto lapso. Las temperaturas mínimas diarias se suelen alcanzar al amanecer, mientras que las mensuales lo hacen por julio o agosto para el hemisferio sur.

- **Temperatura media**

  Se refiera a promedios estadísticos obtenidos entre las temperaturas mínimas y máximas para un lapso y lugar determinado. Estos datos en conjunto con el promedio de precipitaciones mensuales permiten armar un climograma (gráfico climático) de un determinado lugar.

La temperatura y presión atmosférica son variables que se relacionan en forma inversa, a mayor temperatura menor presión con lo que el aire asciende. Cuanto más caliente el aire, mayor será la inestabilidad incidiendo directamente en las precipitaciones.

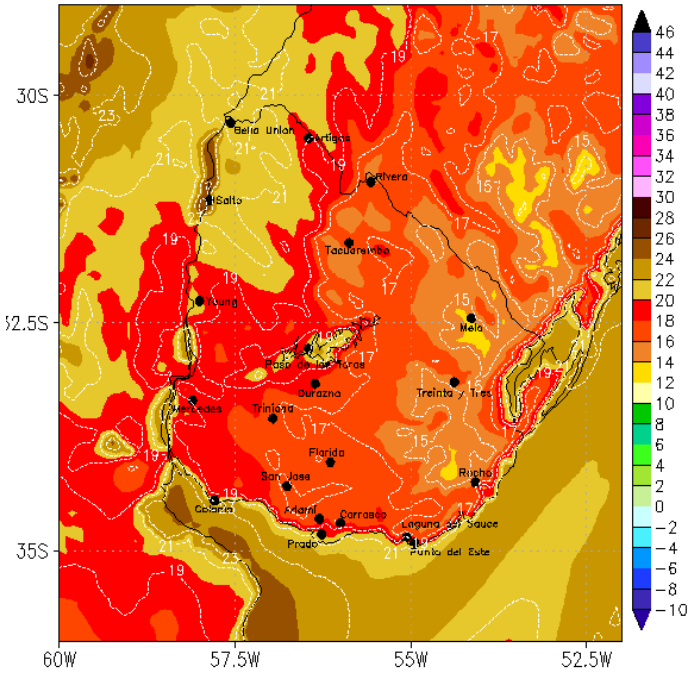


Figura 18 – Mapa de temperatura territorio nacional para el 17 de Febrero de 2018 – INUMET
Fuente: [29]

PRECIPITACIONES

Se denomina precipitación a cualquier forma de meteoro formado por agua que cae a la superficie terrestre desde la atmósfera, llámese llovizna, lluvia, nieve o granizo. En este proyecto nos referiremos a lluvia cuando hablemos de precipitaciones.

La cantidad de precipitación en un punto de la superficie es llamada monto pluviométrico o pluviosidad. Se utilizan pluviómetros generalmente ubicados en las estaciones meteorológicas, que recogen y miden las precipitaciones caídas en el sitio expresadas generalmente en milímetros de altura o mediante el peso del agua que queda atrapada en el depósito.

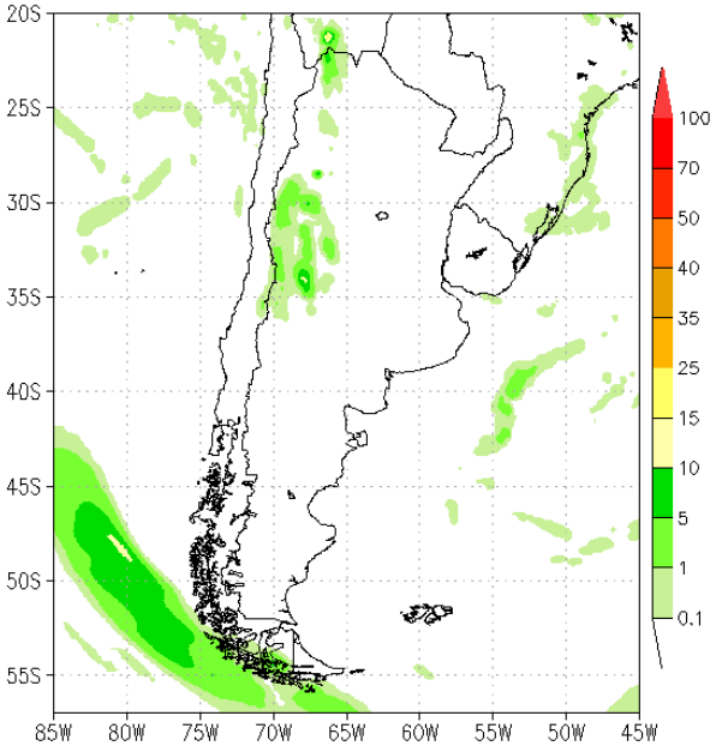


Figura 19 – Mapa de precipitaciones en mm zona sur para el 24 de Febrero de 2018 – INUMET
Fuente: [29]

3.6 Modelos matemáticos utilizados hoy en día para realizar predicciones

El uso de modelos para predecir el clima comenzó en 1922 por el matemático inglés Lewis Fry Richardson, que intentó hacer una previsión numérica sin éxito. Obviamente en esa época no se disponía del volumen de datos y capacidad de cómputo que existe en la actualidad. En 1950 el grupo de meteorólogos compuesto por Jule Charney, Larry Gates, Philip Thompson y John von Neumann logran tener resultado utilizando la ENIAC. Años después, en 1955 aúnan esfuerzos la Fuerza Aérea y la Oficina Meteorológica americana para así invertir en la predicción mediante modelos numéricos.

Hoy en día el trabajo de los meteorólogos consiste en analizar el resultado de modelos matemáticos ejecutados en esas supercomputadoras para un punto dado, para luego en base a su experiencia, emitir un pronóstico. Si estos modelos se aplican a una gran cantidad de puntos (cuadrícula) para distintas capas en la atmósfera se puede obtener una predicción para condiciones futuras

(por ejemplo, los próximos diez minutos). Si luego se utiliza esta información para retroalimentar a los modelos, se pueden hacer predicciones para el futuro tantas horas o días para adelante como se deseen.

Cada uno de los modelos utilizados tiene precisión en algún aspecto a predecir, está en la experiencia de cada meteorólogo elegir los convenientes. Es tarea de distintos especialistas colaborar a mejorar los modelos actuales, para ello incorporan características topográficas tales como montañas, ríos, cantidad de agua en el suelo, bosques, áreas de vegetación (pues reflejan menos la luz solar), los oceanógrafos por su parte incorporan conocimiento sobre el contenido de sal, hielo, temperatura y densidad de los cauces de agua.

Entre los modelos utilizados en la actualidad encontramos:

- ***Global Forecast System* (GFS)**
  El Sistema de Pronóstico Global pertenece a la Administración Nacional Oceánica y Atmosférica de Estados Unidos (NOAA), es un modelo de pronóstico libre, de dominio público y puede ser consultada desde su sitio web. Empresas como *Weather Wunderground*, *AccuWeather*, *The Weather Channel* o *MeteoGroup* lo ofrecen. Permite analizar un conjunto de variables atmosféricas tales como temperatura, viento, precipitaciones, humedad, concentración de ozono (entre otras), aunque su fiabilidad disminuye cuando se intenta pronosticar más de 7 días.
- ***European Centre for Medium-Range Weather Forecasts (*ECMWF*)***
  Es considerado uno de los más fiables, superando a GFS que cometió errores en la dirección e intensidad el huracán Sandy en 2012, permite hacer predicciones a mediano plazo (10 días).
- **UKMET *o United Kingdon Model (*UKMO*)***
  Modelo global utilizado por la Agencia Meteorológica del Reino Unido, es poco utilizado en Uruguay. Se ejecuta cada 12 horas y es capaz de predecir hasta 3 días por delante.
- ***GME***
  Modelo global de la oficina alemana *Deutscher Wetterdienst*.

- ***GEM***

  Modelo global de la Oficina Canadiense de Meteorología.

- ***NOGAPS***

  Modelo global de la *Fleet Numerical Meteorology and Oceanography* Centers norteamericana.

- ***Mesoscale Model 5 (MM5)***

  Quinta generación del modelo de mesoescala tipo euleriano, fue desarrollado en *Pennsylvania State University* en conjunto con el *National Center for Atmospheric Research (NCAR)*. Es un modelo que tiene todos los avances en cuanto a la modelización meteorológica y es elegido por su alta definición en la detección de sistemas atmosféricos de mesoescala, es utilizado además en la *University of California*, *NASA*, Instituto Geofísico del Perú, Universidad de Ciencia y Tecnología *de Hong Kong*, *Institut d'Estudis Espacials de Catalunya*, Universidad Politécnica de Madrid, Observatorio Nacional de Atenas, Servicio Meteorológico Nacional de México, entre otros. Permite realizar simulaciones de fechas anteriores (re análisis), actuales o pronosticar eventos futuros.

- ***Weather Research and Forecast Model (*WRF*)***

  Fue desarrollado por la *NCAR* (*National Center of Atmospheric Research*), la *NOAA*, el *NCEP* (Centro Nacional de Predicción Ambiental), la *AFWA* (Agencia del Tiempo de la Fuerza Aérea) y la Universidad de Oklahoma junto con la *FAA* (Administración Federal de Aviación) americana. Es un modelo de sexta generación de mesoescala, que puede ser aplicado tanto en el pronóstico operativo de tiempo como para investigación, es flexible y eficiente desde el punto de vista computacional además de estar disponible libre y gratuitamente. Es el que utiliza INUMET hoy en día corriendo en la Facultad de Ciencias desde el año 2005. Se estima que es utilizado en más de 150 países y por más de 25.000 usuarios justamente por su flexibilidad, su principal uso está en el estudio del clima regional y el estudio de la generación eólica.

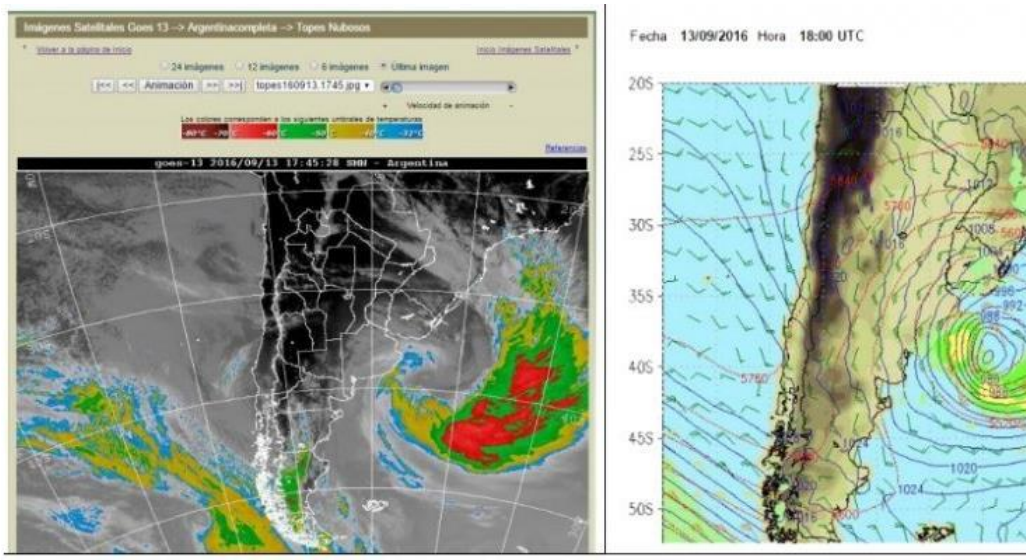


Figura 20 - Comparación imagen satelital en tiempo real *vs* predicción numérica WRF 3 horas previas – INUMET
Fuente: [29]

4. Estado del arte

4.1 Contexto tecnológico

En la década de 1950, hubo una revolución en la predicción del clima. Los avances en la tecnología permitieron simular la atmósfera utilizando modelos dinámicos, lo suficientemente rápidos y precisos para ser utilizados en las predicciones operacionales. Los modelos dinámicos son ahora una parte central de la previsión meteorológica. A partir de leyes físicas básicas, permiten predecir eventos como tormentas antes incluso de que comiencen a formarse.

Un desafío crucial en la próxima década será la integración de simulaciones físicas directas, por un lado y enfoques basados en datos, por el otro. Este enfoque híbrido ofrece muchas oportunidades para la predicción meteorológica, así como para innumerables otros campos.

Los modelos de clima operacional generalmente se ejecutan a una resolución de entre 1 km y 10 km, es decir, todo dentro del mismo kilómetro cuadrado está representado por una sola celda de cuadrícula. Esta resolución es lo suficientemente buena para capturar una amplia gama de fenómenos, pero obviamente no podrá capturar detalles muy localizados.

Es posible realizar este tipo de localización utilizando modelos entrenados en datos históricos, proporcionando un mapeo entre las predicciones a gran escala de la simulación y los efectos a pequeña escala. Esta es un área de investigación activa que podría hacer que los pronósticos sean más útiles para las actividades cotidianas.

Además de predecir el clima a escalas más finas, técnicas similares podrían ayudar a vincular las predicciones meteorológicas con sus impactos más amplios. Muchas cosas se ven afectadas por el clima, ya sea directa o indirectamente; éstas incluyen tráfico, retrasos en los vuelos y admisiones hospitalarias. Si bien algunos efectos pueden no ser fáciles de simular, el uso de

modelos basados en datos podría ayudar a advertir anticipadamente sobre los impactos significativos.

Una cantidad importante de datos se aplica a todas las predicciones meteorológicas, incluidas las observaciones históricas y los factores actuales, como la presión barométrica, la temperatura, la velocidad del viento, los puntos de rocío y más. Los investigadores están aprovechando herramientas informáticas de vanguardia para analizar rápidamente todos estos datos, y la inteligencia artificial (AI) ha comenzado a formar parte de su proceso.

Además, a medida que los sensores más baratos y una mejor conectividad amplían el acceso a internet de las cosas (IOT), es probable que la cantidad de dispositivos y equipos que pueden proporcionar información útil en tiempo real sobre el clima se expanda dramáticamente. Todos los automóviles, camiones, paneles solares, semáforos conectados, teléfonos celulares, sistemas de aire acondicionado domésticos inteligentes, etc. podrían usarse como fuente de información en tiempo real para mejorar el pronóstico.

Un área donde el aprendizaje automático ha logrado un progreso espectacular es la detección de características. Pueden verse ejemplos de esto en aplicaciones que no solo detectan rostros, sino que también le agregan lentes y un bigote en tiempo real.

Actualmente hay mucho interés en aplicar métodos similares a la detección de peligros, especialmente para el seguimiento de tormentas. Los expertos capacitados pueden reconocer las tormentas y rastrear sus caminos a partir de las imágenes meteorológicas; en principio, no hay ninguna razón para que un algoritmo no pueda aprender a hacer lo mismo.

Otra aplicación podría abordar los desafíos planteados por el volumen y la complejidad de los datos cuando se trata de datos de simulaciones físicas. Los campos generados por dichos modelos son altamente multidimensionales; darles sentido es una tarea compleja que requiere muchas "pantallas" de

información. Un algoritmo que pudiera resumir las características principales y ponerlas en conocimiento del pronosticador ayudaría a simplificar esta tarea.

Conceptos, terminología y tecnologías

MACHINE LEARNING

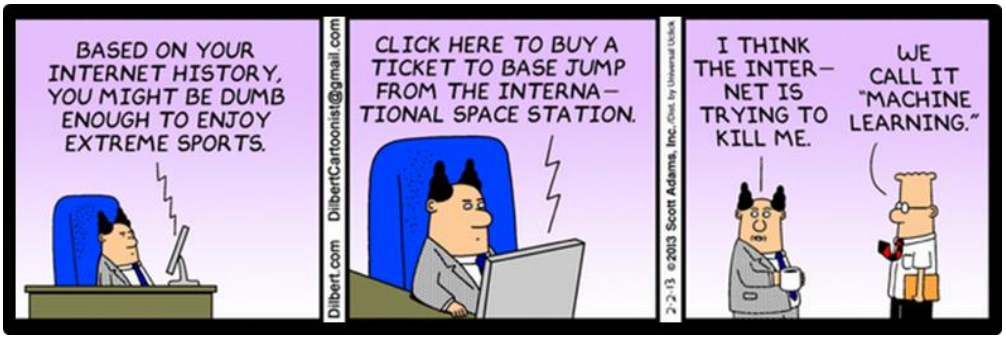


Figura 21 – *Machine Learning* by Dilbert
Fuente: [44]

"Disciplina científica en el ámbito de inteligencia artificial que crea sistemas que aprenden automáticamente." [61] Aprenden identificando patrones complejos en grandes volúmenes de datos, luego generalizan y realizan asociaciones entre ellos. Son capaces de mejorarse en forma autónoma a partir de la experiencia.

Usos de *Machine Learning*

Con el crecimiento *big data*, el aprendizaje automático se ha convertido en una técnica clave para resolver problemas en diferentes áreas. El aprendizaje automático prácticamente cuenta con tantas aplicaciones como imaginemos, pudiéndose adaptar a tantas situaciones como datos con los que contemos. Motores de búsqueda, diagnósticos médicos, reconocimiento del habla y del lenguaje, robótica, entre otras. Veamos algunas de las actividades de nuestro día a día que se ven impulsadas por el *Machine Learning*:

- Detección de rostro. Podemos verlo en nuestras cámaras móviles.
- Reconocimiento facial, de voz o de objetos.
- Buscadores. Para mejorar los resultados y sugerencias de búsqueda.
- Anti-*spam*. Mediante el uso de etiquetas.

- Anti-virus. Para la detección de software malicioso.
- Genética. Por ejemplo, en la clasificación de secuencias de ADN.
- Predicción y pronósticos. De clima, tráfico o para evitar fallos tecnológicos en equipos.
- Comprensión de textos. Se aplica a resúmenes estructurados de noticias o comentarios sobre un tema específico.
- Vehículos autónomos y robots.
- Métodos de optimización más rápidos y flexibles. Se evalúa qué momento es el adecuado para una tarea concreta.
- Análisis de imágenes de alta calidad.
- Análisis de datos económicos. Para operar en el mercado de valores o evitar el fraude en transacciones.
- Análisis de comportamiento de consumo y productividad. Para la identificación de clientes potenciales, prever qué empleados pueden ser más rentables, adaptar servicios a las necesidades del usuario.
- Diagnóstico de enfermedades
- Clasificación de correos para dar respuesta automática a los usuarios

Figura 22 – *Machine Learning* en la vida cotidiana
Fuente: [45]

Técnicas de aprendizaje

El aprendizaje automático utiliza dos tipos de técnicas: aprendizaje supervisado, que entrena un modelo en datos de entrada y salida conocidos para que pueda predecir resultados futuros, y aprendizaje no supervisado, que encuentra patrones ocultos o estructuras intrínsecas en los datos de entrada.

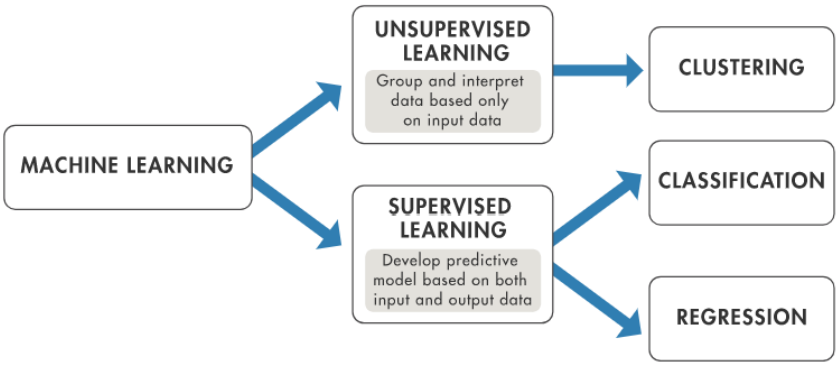


Figura 23 – Técnicas de aprendizaje de Machine Learning
Fuente: [46, Fig. 1]

Aprendizaje supervisado

- **Clasificación**

  La clasificación es una técnica de aprendizaje supervisado mediante la cual los datos se clasifican en categorías relevantes aprendidas previamente. Consta de dos pasos:

  1. El sistema recibe datos de capacitación que ya están categorizados o etiquetados, de modo que puede desarrollar una comprensión de las diferentes categorías.

  2. El sistema se alimenta con datos desconocidos pero similares para la clasificación y, en base a la comprensión que se desarrolló a partir de los datos de capacitación, el algoritmo clasificará los datos sin etiqueta.

Una aplicación común de esta técnica es para el filtrado de spam de correo electrónico. Tenga en cuenta que la clasificación se puede realizar para dos o más categorías. [39]

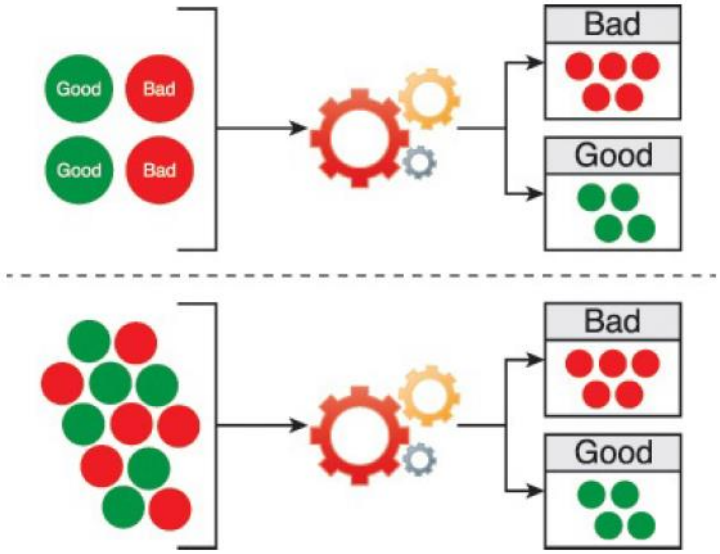


Figura 24 – Aprendizaje supervisado - Clasificación
Fuente: [39, Fig. 8.11]

- **Regresión**

  Este método se utiliza para predecir el valor de un atributo continuo. Consiste en encontrar la mejor ecuación que atraviese de forma óptima un conjunto de puntos (n-dimensiones). Se utiliza cuando la precisión no es crítica y el número de variables es pequeño. Ejemplo: Predecir el precio de una vivienda, dado su tamaño.

Aprendizaje no supervisado

El aprendizaje no supervisado encuentra patrones ocultos o estructuras intrínsecas en los datos. Se utiliza para extraer inferencias a partir de conjuntos de datos de entrada sin respuestas etiquetadas.

- ***Clustering***

  Es la técnica de aprendizaje no supervisada más común. Se utiliza para el análisis exploratorio de datos para encontrar patrones o agrupaciones ocultos en los datos. Sus aplicaciones incluyen análisis de secuencia de genes, estudios de mercado y reconocimiento de objetos. Por ejemplo, si una empresa de telefonía celular desea optimizar las ubicaciones donde construyen las antenas, puede usar el aprendizaje automático para estimar la cantidad de grupos de personas que confían en sus antenas. Un teléfono solo puede hablar con una antena a la vez, por lo que el equipo usa algoritmos de agrupamiento para diseñar la mejor ubicación de antenas celulares para optimizar la recepción de señales para grupos o *clusters* de sus clientes.

Figura 25 – Aprendizaje no supervisado - *Clustering*
Fuente: [46, Fig. 2]

Metodología para construir modelos predictivos

El proceso de construir modelos ha sido desarrollado y perfeccionado por muchos profesionales durante muchos años. Aquí hay un enfoque simple y comprobado para construir modelos exitosos y rentables.

1. **Preparar los datos**. Este paso debe completarse antes de que se lleve a cabo cualquier exploración o análisis significativo. La inversión en comprender el proceso de preparación de datos dentro de su organización

a menudo paga beneficios a largo plazo. Como se necesita acceso a datos cada vez más grandes y más granulares, el conocimiento de qué datos existen y cómo se puede combinar con otras fuentes de datos proporcionará una visión que no era posible hace tan solo unos años.

2. **Realizar análisis de datos exploratorios**. Este es el paso donde se comprenden los datos y comienza a ganar intuición sobre las relaciones entre las variables. Esta exploración se realiza mejor con un experto de dominio, si no tiene esa experiencia. Mediante la minería de datos se descubrirán relaciones y tendencias a través de una exploración compleja. Estos hallazgos se deben revisar junto a un experto de dominio de modo que pueda responder cosas que se conocen desde hace algún tiempo. En los últimos años, las herramientas gráficas han mejorado dramáticamente. Estos productos pueden cargar datos para exploración visual, generalmente a través de una interfaz basada en navegador, y brindan una experiencia altamente interactiva en la exploración de datos. Se ha demostrado que esta tecnología funciona con miles de millones de observaciones, suponiendo que hay suficientes recursos de hardware disponibles. La exploración de datos nunca es completa. Siempre hay más formas de considerar los datos, las interacciones y las relaciones para tener en cuenta, por lo que es necesario observar el principio de suficiencia y la ley de rendimientos decrecientes. La ley de rendimientos decrecientes proviene del campo de la economía y establece que agregar una unidad más de esfuerzo (tiempo en nuestro caso) arrojará menos valor agregado por unidad para cada unidad de esfuerzo sucesiva que ponga en la tarea. En este caso, la percepción y el conocimiento que gana entre las horas 15 y 16 en la exploración de datos es probablemente menor que la percepción que obtuvo entre las horas 2 y 3. El principio de suficiencia reconoce la ley de rendimientos decrecientes y establece un umbral para la pérdida de productividad. Dicho en un lenguaje común, esto es: saber cuándo detener la exploración. La metodología de

desarrollo de software se ha movido para incorporar esta idea a través de procesos ágiles de aprendizaje, de iniciación y mejora continua.

3. **Construir el primer modelo**. La clave para este paso es darse cuenta por adelantado de que el exitoso proceso de creación de modelos implicará muchas iteraciones. Las palabras de Thomas Edison "no he fallado. Acabo de encontrar 10.000 formas que no funcionarán" se ajustan perfectamente a esta situación. Hasta que se construya el primer modelo, no podremos evaluar con precisión cuál será el impacto potencial del mismo. La construcción del primer modelo ayuda a cimentar los criterios de éxito y establecer las expectativas adecuadas para las personas que utilizarán las predicciones del modelo. Siendo optimistas, los próximos modelos deberían mejorar. Construir el primer modelo es una verificación de la realidad para el desempeño y las expectativas futuras. Este primer modelo es, por defecto, el modelo de referencia.

4. **Iterativamente construir modelos**. Esta fase es donde se debe pasar la mayor parte del tiempo. Este paso es un ciclo de retroalimentación donde se construirá un modelo (el retador) y luego se comparará con el modelo referencia usando algunos criterios objetivos que definen el mejor modelo. Si el retador es mejor que el modelo de referencia, entonces debemos evaluar si el modelo desafiante satisface los objetivos del proyecto. Si los objetivos del proyecto no se cumplen, entonces se deberá construir otro modelo. A menudo no hay una evaluación de modelo concreta para determinar cuándo parar, sino más bien es una ventana de tiempo la que obliga al proyecto a finalizar. Supongamos que estamos contratados para proporcionar una lista de clientes para una campaña de marketing. La campaña tiene una fecha límite para proporcionar la lista de clientes para el próximo martes, por lo que la construcción de modelos continuará hasta ese momento.

BIG DATA

Big Data es un campo dedicado al análisis, procesamiento y almacenamiento de grandes colecciones de datos que frecuentemente se originan de fuentes dispares. Las soluciones y prácticas de *Big Data* son típicamente requeridas cuando las técnicas y tecnologías tradicionales de análisis, procesamiento y almacenamiento de datos son insuficientes. Específicamente, *Big Data* aborda distintos requisitos, como la combinación de múltiples conjuntos de datos no relacionados, el procesamiento de grandes cantidades de datos no estructurados y la recopilación de información oculta de forma puntual.

Aunque *Big Data* puede aparecer como una nueva disciplina, se ha estado desarrollando durante años. La gestión y el análisis de grandes conjuntos de datos ha sido un problema de larga data, desde los enfoques intensivos en mano de obra de los primeros esfuerzos censales hasta los cálculos de las primas de seguros. La ciencia de *Big Data* ha evolucionado desde estas raíces.

Además de los enfoques analíticos tradicionales basados en estadísticas, *Big Data* agrega nuevas técnicas que aprovechan los recursos computacionales y los enfoques para ejecutar algoritmos analíticos. Este cambio es importante a medida que los conjuntos de datos continúan volviéndose más grandes, más diversos, más complejos y centrados en la transmisión. Si bien los enfoques estadísticos se han utilizado para las medidas aproximadas de una población a través del muestreo desde épocas bíblicas, los avances en la ciencia computacional han permitido el procesamiento de conjuntos de datos completos, lo que hace innecesario el muestreo.

El análisis de los conjuntos de datos de *Big Data* es un esfuerzo interdisciplinario que combina las matemáticas, las estadísticas, la informática y la experiencia en la materia. Esta mezcla de conjuntos de habilidades y las perspectivas han llevado a cierta confusión en cuanto a lo que comprende el campo de *Big Data* y su análisis, ya que la respuesta que se reciba dependerá de la perspectiva de quien responda la pregunta. Los límites de lo que constituye un problema de *Big Data* también están cambiando debido al panorama cambiante y en constante

avance de la tecnología de *software* y *hardware*. Esto se debe al hecho de que la definición de *Big Data* tiene en cuenta el impacto de las características de los datos en el diseño del entorno de la solución en sí. Hace treinta años, un *gigabyte* de datos podía constituir un problema de *Big Data* y requería recursos informáticos de propósito especial. Ahora, *gigabytes* de datos son comunes y pueden ser fácilmente transmitidos, procesados y almacenados en cualquier dispositivo.

Los datos dentro de los entornos de *Big Data* generalmente se acumulan a través de aplicaciones, sensores y fuentes externas. Los datos procesados por una solución de *Big Data* pueden ser utilizados por las aplicaciones empresariales directamente o pueden ser enviados a un almacén de datos.

Desde la implantación total de Internet y el crecimiento de la población el volumen de datos ha crecido en forma exponencial. *Retail*, bancos, *eCommerce* y todo tipo de negocio incrementa el *Big Data* con millones de datos que se generan en la red. A esto se le suman los dispositivos electrónicos: sensores, coches, cámaras, micrófonos, etc.

Ese volumen de datos ya no puede ser tratado de la forma tradicional y es aquí donde entra en juego *Machine Learning*. Porque no solo se trata de conocer a los usuarios, sino que ahora es necesario poder predecir su comportamiento futuro.

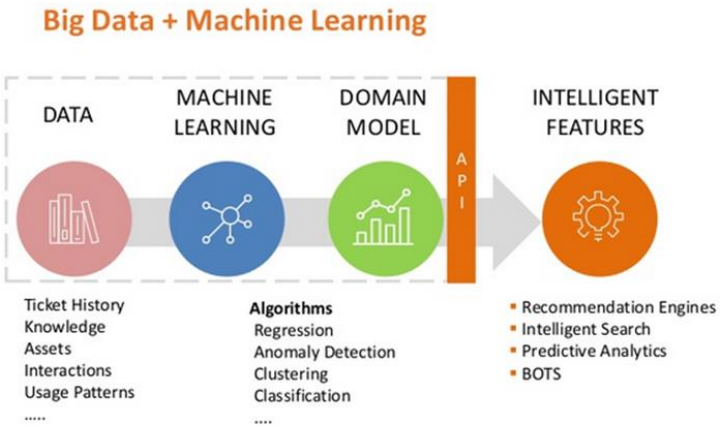


Figura 26 – *Big Data + Machine Learning*
Fuente: [47]

ANÁLISIS DE DATOS

El análisis de datos permite la toma de decisiones basada en datos con respaldo científico, de modo que las decisiones pueden basarse en datos objetivos y no simplemente en la experiencia pasada o la intuición solamente. Hay cuatro categorías generales de análisis que se distinguen por los resultados que producen:

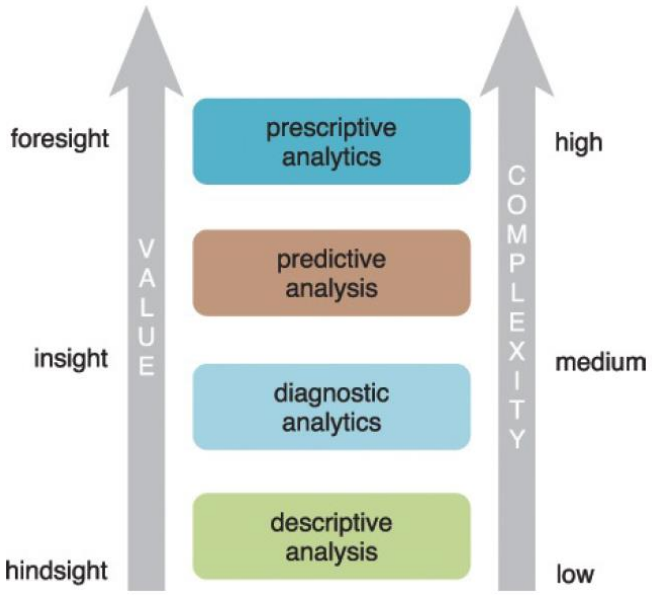


Figura 27 – Tipos de análisis de datos
Fuente: [39, Fig. 1.4]

- **Análisis descriptivo**: Se realizan análisis descriptivos para responder preguntas sobre eventos que ya han ocurrido. Esta forma de análisis contextualiza los datos para generar información. El análisis descriptivo a menudo se lleva a cabo a través de informes *ad-hoc* o tableros. Los informes son generalmente de naturaleza estática y muestran datos históricos que se representan en forma de cuadrículas o gráficos de datos. Las consultas se ejecutan en almacenes de datos operativos desde una empresa, por ejemplo, un sistema de gestión de relaciones con el cliente

(CRM) o un sistema de planificación de recursos empresariales (ERP) [39].

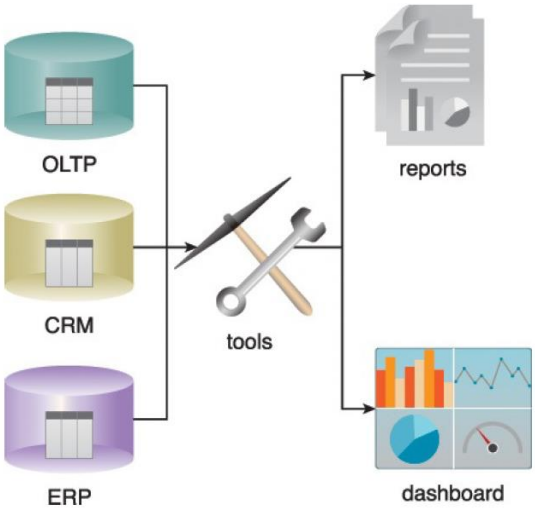


Figura 28 – Análisis descriptivo
Fuente: [39, Fig. 1.5]

- **Análisis de diagnóstico**: La analítica de diagnóstico tiene como objetivo determinar la causa de un fenómeno que ocurrió en el pasado utilizando preguntas que se centran en la razón detrás del evento. El objetivo de este tipo de análisis es determinar qué información está relacionada con el fenómeno para permitir responder preguntas que buscan determinar por qué algo ha ocurrido. El análisis de diagnóstico proporciona más valor que el análisis descriptivo, pero requiere un conjunto de habilidades más avanzado. El análisis de diagnóstico generalmente requiere recopilar datos de múltiples fuentes y almacenarlo en una estructura que se presta para realizar análisis detallados y acumulativos. Los resultados del análisis de diagnóstico se visualizan mediante herramientas de visualización interactiva que permiten a los usuarios identificar tendencias y patrones. Las consultas ejecutadas son más complejas en comparación con las del análisis descriptivo y se realizan en datos multidimensionales almacenados en sistemas de procesamiento analítico [39].

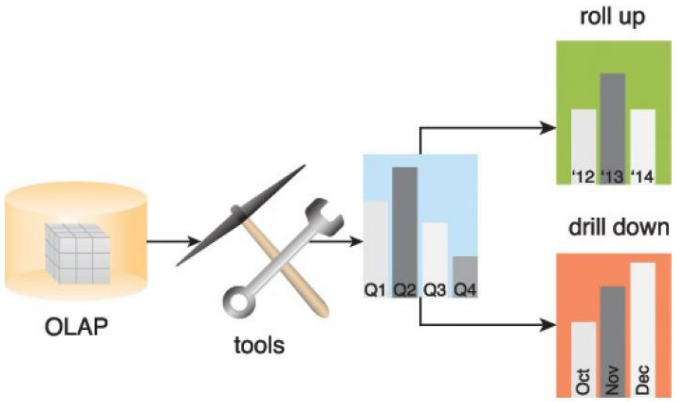


Figura 29 – Análisis de diagnóstico
Fuente: [39, Fig. 1.6]

- **Análisis predictivo**: El análisis predictivo se lleva a cabo en un intento de determinar el resultado de un evento que pueda ocurrir en el futuro. Con el análisis predictivo, la información se mejora con el significado de generar conocimiento que transmita cómo se relaciona esa información. La fuerza y la magnitud de las asociaciones forman la base de los modelos que se utilizan para generar predicciones futuras basadas en eventos pasados. Es importante entender que los modelos utilizados para el análisis predictivo tienen dependencias implícitas en las condiciones bajo las cuales ocurrieron los eventos pasados. Si estas condiciones subyacentes cambian, entonces los modelos que hacen predicciones necesitan ser actualizados. Los análisis predictivos intentan predecir los resultados de los eventos y las predicciones se basan en patrones, tendencias y excepciones que se encuentran en los datos históricos y actuales. Esto puede conducir a la identificación de riesgos y oportunidades. Este tipo de análisis implica el uso de grandes conjuntos de datos compuestos por datos internos y externos y diversas técnicas de análisis de datos. Proporciona un mayor valor y requiere un conjunto de habilidades más avanzado que el análisis descriptivo y de diagnóstico [39].

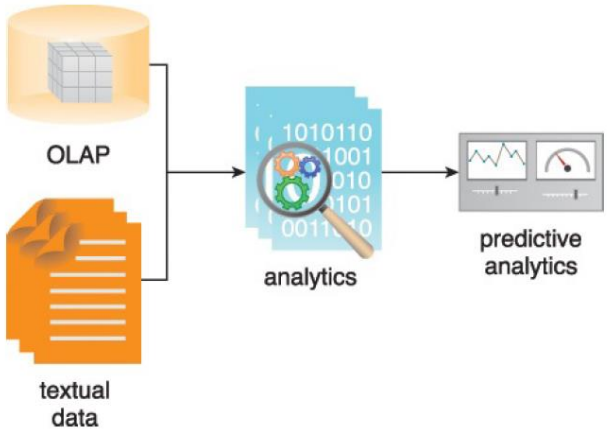


Figura 30 – Análisis predictivo
Fuente: [39, Fig. 1.7]

- **Análisis prescriptivo**: El análisis prescriptivo se basa en los resultados del análisis predictivo prescribiendo las acciones que se deben tomar. El enfoque no es solo qué opción prescrita es mejor seguir, sino por qué. En otras palabras, el análisis prescriptivo proporciona resultados que pueden razonarse porque incorporan elementos de comprensión situacional. Por lo tanto, este tipo de análisis se puede utilizar para obtener una ventaja o mitigar un riesgo. El análisis prescriptivo proporciona más valor que cualquier otro tipo de análisis y, por consiguiente, requiere el conjunto de habilidades más avanzado, así como *software* y herramientas especializadas. Se calculan varios resultados y se sugiere el mejor curso de acción para cada resultado. Este tipo de análisis incorpora datos internos con datos externos. Los datos internos pueden incluir datos de ventas actuales e históricos, información del cliente, datos del producto y reglas comerciales. Los datos externos pueden incluir datos de redes sociales, pronósticos meteorológicos y datos demográficos producidos por el gobierno. El análisis prescriptivo implica el uso de reglas comerciales y grandes cantidades de datos internos y externos para simular resultados y prescribir el mejor curso de acción [39].

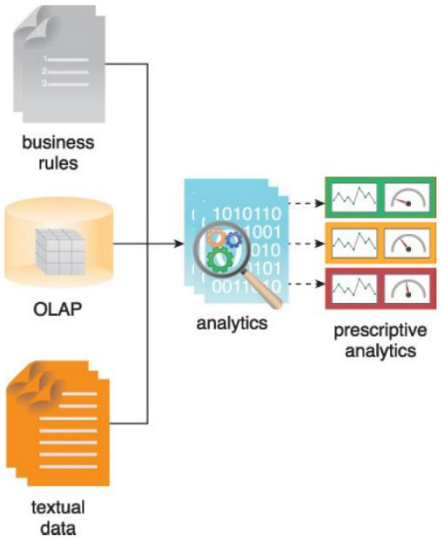


Figura 31 – Análisis prescriptivo
Fuente: [39, Fig. 1.8]

INTERNET DE LAS COSAS

Viene del inglés *Internet of Things*, “es un concepto que se refiere a la interconexión digital de objetos cotidianos con *Internet*. Generalmente, *Internet* de las cosas es la conexión de *Internet* con más “cosas u objetos” que personas” [22].

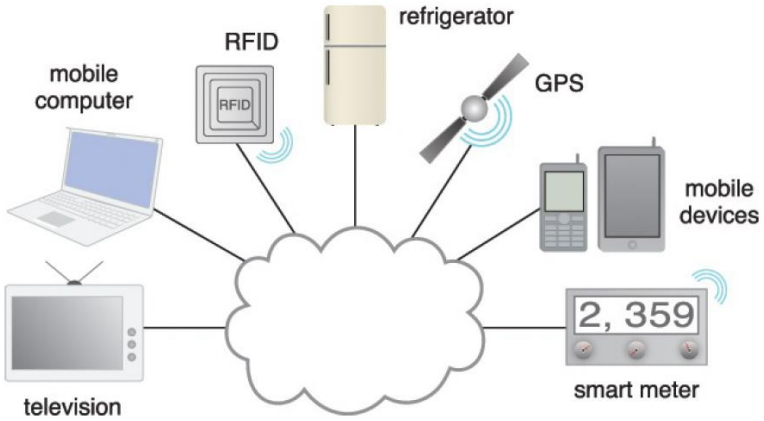


Figura 32 – Internet de las cosas
Fuente: [39, Fig. 2.6]

ARDUINO

Arduino es una plataforma de código abierto utilizada para la construcción de proyectos de electrónica. Consiste en una placa de circuito programable física (a menudo denominada microcontrolador) y una pieza de software, o IDE (Entorno de Desarrollo Integrado) que se ejecuta en la computadora donde es posible escribir código y cargarlo a la placa física [23].

La plataforma Arduino se ha vuelto bastante popular entre las personas que recién comienzan con productos electrónicos, y por una buena razón. A diferencia de la mayoría de las placas de circuitos programables anteriores, Arduino no necesita una pieza de hardware separada (llamada programador) para cargar código nuevo en la placa; simplemente puede usar un cable USB. Además, Arduino IDE utiliza una versión simplificada de C++, por lo que es más fácil aprender a programar [23].

Permite de forma sencilla y económica conectar cualquier cosa a *Internet*. Con un Arduino y un sencillo módulo Ethernet o WiFi podemos conectar a *Internet* sensores para informar, controlar motores o lamparitas desde cualquier parte del mundo o mandar un SMS o email cada vez que se abra la puerta de casa.

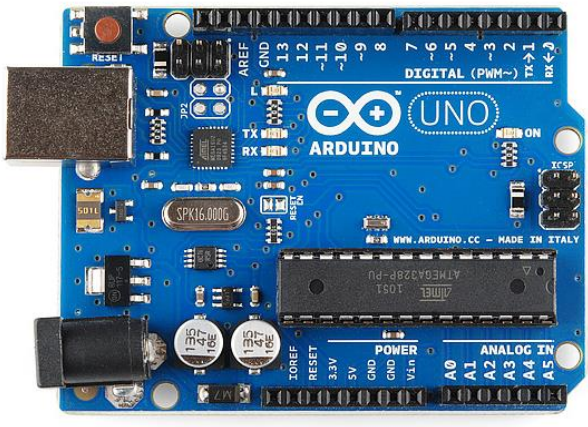


Figura 33 – Placa Arduino Uno
Fuente: [23]

Figura 34 –Arduino IDE
Fuente: [23]

PYTHON

Es un lenguaje de programación interpretado, interactivo y orientado a objetos. Incorpora módulos, excepciones, tipos de datos dinámicos de muy alto nivel y clases. Combina una potencia notable con una sintaxis muy clara. Tiene interfaces para muchas llamadas y librerías del sistema, así como para varios sistemas de ventanas, además es extensible en C o C++. También se puede usar como un lenguaje de extensión para aplicaciones que necesitan una interfaz programable. Es portátil pues se ejecuta en sistemas operativos Linux, Mac y Windows.

Ha surgido en las últimas dos décadas como una herramienta de primera clase para tareas informáticas científicas, incluido el análisis y la visualización de grandes conjuntos de datos. La utilidad de este lenguaje para la ciencia de datos

proviene principalmente del ecosistema grande y activo de paquetes de terceros: NumPy para manipulación de datos homogéneos basados en arreglos, Pandas para manipulación de datos heterogéneos y etiquetados, SciPy para tareas informáticas científicas comunes, Matplotlib para visualizaciones, IPython para la ejecución interactiva y el uso compartido de código, Scikit-Learn para el aprendizaje automático, entre otras herramientas.

CLOUD COMPUTING

Figura 35 – Cloud Computing
Fuente: [49]

“La computación en la nube es un paradigma de tecnología de la información (TI) que permite el acceso ubicuo a grupos compartidos de recursos configurables del sistema y servicios de alto nivel que pueden aprovisionarse rápidamente con un mínimo esfuerzo administrativo, a menudo a través de *Internet*. La computación en la nube se basa en el intercambio de recursos para lograr coherencia y economías de escala, similar a una utilidad pública. Las nubes de terceros permiten que las organizaciones se centren en sus negocios centrales en lugar de gastar recursos en infraestructura y mantenimiento de computadoras. Los defensores señalan que la computación en la nube permite a las empresas evitar o minimizar los costos iniciales de la infraestructura de TI. Los proponentes también afirman que la computación en la nube permite a las empresas tener sus aplicaciones en funcionamiento más rápido, con mejor capacidad de administración y menos mantenimiento, y permite a los equipos de

TI ajustar más rápidamente los recursos para satisfacer la demanda fluctuante e impredecible. Los proveedores de la nube suelen utilizar un modelo de "pago por uso", que puede generar gastos operativos inesperados si los administradores no están familiarizados con los modelos de fijación de precios en la nube.” [24]

“Si bien la gran mayoría de la atención de los medios en los primeros días de la nube se dirigió a los principales proveedores de servicios web IaaS-Amazon de la nube pública, Microsoft y Google, el siguiente punto más estratégico en *Cloud Wars* será el software, particularmente alrededor del potencial de auge para IA, *Machine Learning* y *Blockchain*. Como resultado, las últimas clasificaciones de *Cloud Wars Top 10* revelan algunos cambios importantes en la lista de los proveedores de computación en la nube más poderosos e influyentes del mundo.” [25]

Figura 36 – Cloud Wars
Fuente: [25]

MICROSOFT AZURE

Es un conjunto integral de servicios en la nube que los desarrolladores y los profesionales de TI utilizan para crear, implementar y administrar aplicaciones a través de nuestra red global de centros de datos. Herramientas integradas,

DevOps y un *marketplace* le ayudan a crear de manera eficaz cualquier cosa, desde aplicaciones móviles sencillas hasta soluciones orientadas a Internet. [38]

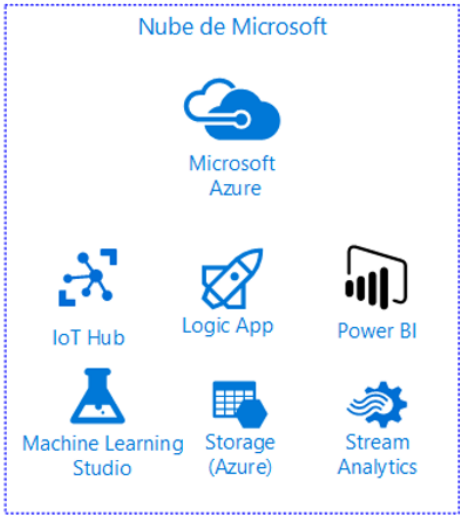


Figura 37 – Servicios de Azure

Algunos de los servicios disponibles:

- ***Logic Apps***: Es una manera de simplificar e implementar integraciones escalables y flujos de trabajo en la nube. Proporciona un diseñador visual para modelar y automatizar el proceso en una serie de pasos denominada flujo de trabajo. Hay muchos conectores en la nube y locales para la integración rápida en servicios y protocolos.
- **Azure *Storage***: Es un servicio en la nube administrado por Microsoft que proporciona almacenamiento altamente disponible, seguro, duradero, escalable y redundante.
- ***Stream Analytics***: Es un motor de procesamiento de eventos totalmente administrado para configurar cálculos analíticos en tiempo real sobre datos de *streaming*. Los datos pueden proceder de dispositivos, sensores, sitios web, fuentes de redes sociales, aplicaciones, sistemas de infraestructura, etc.
- ***Machine Learning Studio*:** Es una herramienta *drag* y *drop* que le permite crear, probar e implementar soluciones de análisis

predictivo en sus datos. *Machine Learning Studio* publica modelos como servicios web que pueden utilizarse fácilmente en aplicaciones personalizadas o herramientas de BI como Excel.

- ***Power BI***: Es un conjunto de aplicaciones de análisis de negocios que permite analizar datos y compartir información. Los paneles de *Power BI* ofrecen a los usuarios una vista de 360 grados con sus métricas más importantes en un mismo lugar. La información se actualiza en tiempo real y está disponible en todos sus dispositivos.
- ***IOT Hub***: Es un servicio totalmente administrado que permite la comunicación bidireccional fiable y segura entre millones de dispositivos IoT y un *back-end* de soluciones. [38]

IBM *CLOUD*

La plataforma informática de nube IBM combina la plataforma como servicio (PaaS) con la infraestructura como servicio (IaaS) e incluye un catálogo enriquecido de servicios en la nube que se pueden integrar fácilmente con PaaS e IaaS para compilar aplicaciones empresariales rápidamente.

IBM *Cloud* (anteriormente Bluemix) tiene despliegues que se ajustan a las necesidades de los usuarios tanto si se trata de un pequeño negocio con planes de crecimiento, como si es una empresa de gran tamaño que precisa de aislamiento adicional. Puede desarrollar en una nube sin límites, donde puede conectar sus servicios privados a los servicios IBM *Cloud* públicos disponibles desde IBM. Se puede acceder a las apps, servicios e infraestructura de IBM *Cloud* y utilizar datos, sistemas, procesos, herramientas de PaaS y de IaaS existentes. Los desarrolladores pueden trabajar en el ecosistema de crecimiento rápido de servicios disponibles e infraestructuras de tiempo de ejecución para crear aplicaciones utilizando enfoques de programación políglotas. [57]

Figura 38 – Servicios de IBM *Cloud* (Bluemix)

Algunos de los servicios disponibles:

- ***Node-RED***: Es una herramienta de desarrollo visual que permite implementar flujos con enfoque IOT permitiendo integrar APIs de terceros y dispositivos de hardware a través de protocolos estándar como REST y MQTT. [35]
- **Cloudant NoSQL DB**: Es un almacén de documentos NoSQL JSON optimizado para manejar grandes cargas de trabajo de lecturas y escrituras concurrentes en la nube; una carga de trabajo típica de aplicaciones web y móviles grandes y de rápido crecimiento. Puede usar Cloudant como un DBaaS totalmente administrado que se ejecuta en IBM Cloud. [57]
- ***Weather Company Data***: Este servicio le permite integrar datos sobre el tiempo de *The Weather Company* a la aplicación IBM Bluemix. Puede recuperar datos del tiempo correspondientes a un área especificada por una geolocalización. Los datos le permiten crear aplicaciones para resolver problemas empresariales reales en los que el tiempo tiene un impacto significativo sobre el resultado. [58]

- ***Data Science Experience* (DSX)**: Es un entorno basado en *cloud*, interactivo y colaborativo donde los científicos de datos pueden utilizar múltiples herramientas para sacar conocimiento de los datos. Los científicos de datos pueden utilizar lo mejor del código abierto, acceder a funcionalidades exclusivas de IBM, desarrollar sus capacidades y compartir sus logros. [57]

4.2 Trabajos de aplicación de *Machine Learning* en el pronóstico del tiempo

En la actualidad existen y se están desarrollando modelos de *Machine Learning* aplicados al pronóstico del tiempo. A continuación, se listan varios trabajos realizados en Uruguay y distintas universidades del mundo, asimismo se describen aplicaciones que han sido desarrolladas para este mismo propósito.

En Uruguay

- **Relevamiento y obtención de datos sobre emergencias en Uruguay para análisis predictivo - Ignacio Chiazzo, Felipe García, Guillermo Leopold**. Uruguay, a pesar de ser uno de los países más estables en materia de desastres naturales, sufre de manera recurrente de inundaciones dentro de su territorio, las cuales repercuten negativamente principalmente en su población, pero también en su economía debido a pérdidas materiales. Por lo tanto, resulta deseable poder predecirlas con la mayor exactitud posible y minimizar sus consecuencias a todo nivel. Los datos públicos que el estado uruguayo expone sobre meteorología y el estado de sus corrientes fluviales no son en todos los casos claros ni consistentes, y en algunos casos, casi inexistentes. Interesa entonces investigar sobre su disponibilidad y la posibilidad de consumirlos y almacenarlos en un *data warehouse* central con una calidad aceptable para su posterior uso en análisis predictivo. Con esas dos premisas, se plantea el siguiente trabajo como proyecto de grado divido en tres objetivos principales a cumplir. En una primera etapa, el objetivo es el de realizar una investigación sobre los datos públicos disponibles y explorar y establecer medios para obtener aquellos no disponibles, tanto históricos

como del presente de manera continua. Como segunda etapa, se propone procesar dichos datos para darles una mayor calidad y almacenarlos de manera adecuada y de que resulten útiles de alguna manera para asistir decisiones. Se logró una base de datos muy amplia, con más de un millón de registros concernientes a condiciones climatológicas que afectan a los eventos de inundación en el período 1983-2014. En la última etapa del proyecto se investigan técnicas de predicción en base a estadística, llamadas de aprendizaje automático, mostrando el potencial de los datos obtenidos y procesados a la hora de monitorear este tipo de desastres. Luego de las pruebas se seleccionó el algoritmo de clasificación SVC con *kernel* RBF por tener mejor precisión y dar menor error en la mayoría de los casos de prueba. Para desplegar geolocalizadamente éstas predicciones y gestionar los datos almacenados, se desarrolla además una pequeña aplicación web con un sistema de información geográfica integrado para mostrar de manera gráfica las posibles inundaciones en el territorio de alguno de los 19 departamentos por separado y con la posibilidad de una ejecución aplicada a todo el país con el fin de obtener un panorama general del territorio nacional. [27]

En el exterior

- **Modelo predictivo. *Machine Learning* aplicado al análisis de datos climáticos capturados por una placa Sparkfun.** El proyecto tiene como objetivo principal el desarrollo de un modelo de predicción que determine en tiempo real la probabilidad de cancelación o retraso de un vuelo en función de las condiciones meteorológicas. Para ello, se utiliza una placa Sparkfun como dispositivo de toma de datos meteorológicos, la plataforma en la nube Azure para desarrollar y ejecutar el modelo y *Power BI* como herramienta de procesamiento y visualización de datos. [31]

- ***A Machine Learning approach to finding Weather regimes and skillful predictor combinations for short-term storm forecasting.*** Un desafío importante para la planificación eficiente del vuelo y la gestión del tránsito aéreo es la predicción precisa de las condiciones meteorológicas que plantean un peligro para la aviación. En apoyo de la visión conjunta de la Oficina de Planificación y Desarrollo (JPDO) de una única fuente autorizada de información meteorológica para todos los usuarios, la Administración Federal de Aviación (FAA) ha ordenado la investigación y el desarrollo para combinar las mejores tecnologías disponibles utilizadas por varios patrocinadores de la FAA convección *nowcast* y productos de pronóstico en una sola predicción de tormenta consolidada para la aviación (CoSPA, Wolfson et al., 2008). Para lograr este objetivo, se necesita una técnica objetiva para comparar la utilidad de varios predictores e identificar un subconjunto que pueda utilizarse en un algoritmo eficiente y hábil para la predicción de tormentas. En este documento, los autores examinan el problema específico de combinar varios modelos de NWP, radar, satélite y campos derivados para pronosticar el inicio de tormentas en un lapso de tiempo de una hora. Para este propósito, se usa un método de aprendizaje automático que crea bosques aleatorios (conjuntos de árboles de decisión débiles y correlacionados débilmente) para clasificar la importancia del predictor y proporcionar un punto de referencia para el rendimiento potencial del algoritmo. Con los datos recopilados durante el verano de 2007, esta técnica sugiere que el mejor conjunto de predictores de iniciación varía según el día, la hora y la ubicación, presumiblemente debido a las diferentes características climáticas. Los bosques aleatorios se utilizan para ayudar a identificar "regímenes" significativos que pueden representar diferentes tipos de convección, ubicaciones geográficas o condiciones sinópticas. Los resultados iniciales sugieren que las predicciones ajustadas a cada régimen se pueden combinar en un algoritmo de estilo Takagi-Sugeno basado en "membresías" de régimen difuso para lograr una mejor simplicidad y rendimiento del algoritmo.

Además, se evalúa la producción de un algoritmo preliminar de predicción de bosque aleatorio para dos estudios de casos. Aunque este trabajo aún se encuentra en sus etapas iniciales, los autores concluyen que este enfoque es prometedor y que la aplicación de una metodología similar a otros elementos del desarrollo de CoSPA puede valer la pena. [20]

- ***Dark Sky*.** Es una aplicación para iPhone, iPad y iPod *touch* que predice el clima. Usando su ubicación precisa, le indica cuándo lloverá y por cuánto tiempo. Por ejemplo: podría decirte que comenzará a llover en 8 minutos, con una lluvia de 15 minutos seguida de un descanso de 25 minutos. ¿Cómo es posible predecir el clima hasta el último minuto? ¿Cuál es el truco? Bueno, la trampa es que solo funciona en un corto período de tiempo: de media hora a una hora en el futuro. Pero, como resulta, este período de tiempo es de crucial importancia. Nuestras vidas están llenas de actividades al aire libre a corto plazo: viajar hacia y desde el trabajo, pasear al perro, almorzar con amigos, deportes al aire libre, etc. ¿Cuantos de nosotros hemos dejado nuestras casas en el momento más inoportuno, atascados en la lluvia porque no teníamos una advertencia previa y sin el conocimiento de cuánto tiempo duraría? *Dark Sky* te da ese conocimiento. Y después de usar el prototipo nosotros mismos, no podemos imaginar la vida sin él. Ahora nos parece tan extraño mirar hacia afuera y ver a la gente atrapada en la lluvia. [21]

- ***Atmospheric Temperature Prediction using Support Vector Machines*.** Este trabajo presenta una aplicación de *Support Vector Machines* (SVMs) para la predicción del clima. Datos de series temporales de la temperatura máxima diaria en una ubicación, se analizan para predecir la temperatura máxima del día siguiente en esa ubicación en función del máximo diario temperaturas durante un lapso de n días previos denominado orden de la entrada. El método de regresión no lineal es adecuado para entrenar el SVM para esta aplicación. Los resultados son comparados con *Multi Layer Perceptron* (MLP) entrenado con algoritmo

de retropropagación y el rendimiento de SVM es encontrado para ser consistentemente mejor. [1]

- ***Bayesian Networks for Probabilistic Weather Prediction*.** En este trabajo se presenta Bayesian *Networks* (BNs) como marco de trabajo para modelar las dependencias espaciales y temporales entre las diferentes estaciones usando un gráfico acíclico dirigido. Este gráfico es aprendido de las bases de datos disponibles y permite derivar un modelo de probabilidad consistente con toda la información disponible. Luego, el modelo resultante se combina con predicciones numéricas atmosféricas que se dan como evidencia para el modelo. Los mecanismos de inferencia eficiente proporcionan las distribuciones condicionales de las variables deseadas en un tiempo futuro deseado. [2]

- ***A hidden Markov model for downscaling synoptic atmospheric patterns to precipitation amounts*.** Los modelos ocultos de Markov no homogéneos (NHMM) proporcionan un marco relativamente simple para simular la precipitación en estaciones de pluviómetros múltiples condicionadas a patrones atmosféricos sinópticos. Sobre la base de los NHMM existentes para las ocurrencias de precipitación, se propone una extensión para incluir también las cantidades de precipitación. El modelo que se describe supone la existencia de patrones climáticos no observados (u ocultos), los estados climáticos, que siguen una cadena de Markov. Los estados climáticos dependen de información sinóptica observable y, por lo tanto, sirven como un enlace entre los patrones atmosféricos a escala sinóptica y la precipitación a escala local. La presencia de estados ocultos simplifica la estructura espacio-temporal del proceso de precipitación. Suponemos que la dependencia temporal de la precipitación se explica completamente por la evolución de Markov del estado del tiempo. La dependencia espacial en la precipitación también puede explicarse parcial o completamente por la existencia de un estado meteorológico común. [3]

- ***A Deep Hybrid Model for Weather Forecasting*.** Se exploran nuevas direcciones con predicciones meteorológicas como un desafío de datos intensivos que implica inferencias a través del espacio y el tiempo. Se estudia específicamente el poder de hacer predicciones a través de un enfoque híbrido que combina de forma discriminatoria modelos predictivos entrenados con una red neuronal profunda que modela las estadísticas conjuntas de un conjunto de variables relacionadas con el clima. Se muestra cómo se puede mejorar el modelo base con interpolación espacial que usa las dependencias espaciales aprendidas de largo alcance. También se deriva un procedimiento de aprendizaje e inferencia eficiente que permite la optimización a gran escala de los parámetros del modelo. Se evalúan los métodos con experimentos sobre datos meteorológicos del mundo real que destacan la promesa de este enfoque. [4]

- ***The Weather Company***

  Pertenece a IBM y es la empresa meteorológica privada más grande del mundo. La compañía ofrece datos meteorológicos precisos, personalizados y procesables para millones de consumidores y miles de empresas a través de *Weather's* API, su división de soluciones comerciales y sus propios productos digitales de *The Weather Channel* (*weather.com*) y *Weather Underground* (wunderground.com). La compañía ofrece hasta 26 mil millones de pronósticos diarios y cuenta con una red de estaciones meteorológicas personales más grande del mundo. Provee servicios que ayudan a mejorar la toma de decisiones y responder al impacto del clima en los negocios. [58]

  Cuenta con tres grandes proyectos:

  - ***Deep Thunder***. Tiene como objetivo mejorar la predicción meteorológica local a corto plazo a través del uso de la computación

de alto rendimiento. Es parte de la iniciativa *Deep Computing* de IBM que también produjo la computadora de ajedrez *Deep Blue*. *Deep Thunder* está destinado a proporcionar predicciones meteorológicas locales de alta resolución, personalizadas para operaciones comerciales específicas sensibles al clima. Por ejemplo, podría usarse para predecir la velocidad del viento en una plataforma olímpica de buceo, o donde habrá inundaciones o líneas eléctricas dañadas con varias horas de anticipación. El proyecto ahora tiene su sede en el Centro de Investigación Thomas J. Watson de IBM en Yorktown Heights, Nueva York. [59]

- ***Predicting Hurricane Damage with Machine Learning (Outage Prediction).*** Cuando ocurre un clima severo y se corta la electricidad, las compañías de servicios públicos deben actuar rápidamente para que los servicios vuelvan a funcionar para mantener felices a sus clientes y reguladores. Las empresas de servicios públicos quieren movilizarse de forma proactiva, eficiente y solo cuando sea necesario, ya que movilizarse por interrupciones relacionadas con el clima puede costar a una empresa millones de dólares por año. La predicción de cortes de energía puede ahorrar tiempo y dinero a los servicios públicos. La Predicción de paradas aprovecha la plataforma líder mundial de pronósticos meteorológicos, permitiendo a las empresas de servicios públicos responder proactivamente a las tormentas y otras condiciones climáticas severas, optimizar los esfuerzos de restauración y minimizar el riesgo y el daño. Esto les ayuda a controlar los costos de movilización y aumentar la satisfacción del cliente. [26]

- ***Weather for Agriculture - Integrative Solutions*.** El clima es uno de los factores más limitantes para la industria agrícola. El clima no solo afecta la forma en que crecen los cultivos, sino también la logística de siembra, cosecha y transporte. Al integrar los modelos de pronóstico del clima en la siembra de cultivos y la cosecha y el transporte, se

pueden tomar mejores decisiones antes de las pérdidas de cosechas debido a los peligros del clima. Los datos meteorológicos y analíticos campo por campo o zona por zona ayudan a los agricultores a tomar decisiones informadas a lo largo del año para maximizar la producción de alimentos, minimizar el impacto ambiental y reducir los costos operativos. [60]

- ***Weather Forecasting in Sudan Using Machine Learning Schemes*.** En esta investigación, se han realizado esfuerzos para examinar la relación de las precipitaciones en Sudán con parámetros importantes como la estación, la dirección del viento, la fecha, la humedad, la temperatura mínima, la temperatura máxima y la velocidad del viento. Se ha intentado averiguar la correlación de la lluvia con estos elementos. Los objetivos de este documento son demostrar: (1) Cómo se puede utilizar la selección de características para identificar las relaciones entre las ocurrencias de lluvia y otras condiciones climáticas y (2) ¿Qué clasificadores pueden proporcionar las estimaciones de precipitación más precisas? Se han utilizado datos meteorológicos mensuales de la Oficina Central de Estadística de Sudán de 2000 a 2012 para 24 estaciones meteorológicas. Para realizar la selección de características y construir modelos de predicción, se utilizaron un grupo de algoritmos de minería de datos. El análisis muestra que las variables fecha, temperatura mínima, humedad y viento afectan la lluvia en Sudán. Como resultado se obtuvieron los mejores 14 algoritmos para construir modelos capaces de predecir lluvia. [5]

- ***Meteonowcasting using Deep Learning Architecture***. El presente documento propone un esfuerzo para aplicar el enfoque de aprendizaje profundo para la predicción de los parámetros meteorológicos, tales como la temperatura, la presión y la humedad de un sitio en particular. Los modelos predictivos implementados se basan en *Deep Belief Network* (DBN) y *Restricted Boltzmann Machine* (RBM). Inicialmente, cada modelo

se entrena capa por capa sin supervisión para aprender las características jerárquicas no lineales de la distribución de entrada del conjunto de datos. Posteriormente, cada modelo se vuelve a entrenar globalmente de forma supervisada con una capa de salida para predecir el resultado apropiado. Los resultados obtenidos son alentadores. Se encuentra que el modelo de pronóstico basado en características puede hacer predicciones con un alto grado de precisión. Esto implica que el modelo puede adaptarse adecuadamente para hacer pronósticos más largos en áreas geográficas más grandes. [6]

- ***RAL - Advancing Weather Analysis and Forecasting Technologies***. RAL ha sido líder en el desarrollo de sistemas inteligentes de predicción meteorológica que combinan datos de modelos numéricos de predicción meteorológica, conjuntos de datos estadísticos, observaciones en tiempo real e inteligencia humana para optimizar los pronósticos en ubicaciones definidas por el usuario. El objetivo de estos sistemas es reducir el error de pronóstico inherente asociado con los modelos de Predicción Numérica del Tiempo (NWP) y simplificar el proceso de previsión para los responsables de la toma de decisiones. Al utilizar el aprendizaje automático para comprender las características de error de los modelos, podemos combinarlos para crear un pronóstico de consenso optimizado. El Sistema de pronóstico integrado dinámico (DICast) se desarrolló en RAL y es un ejemplo de esta tecnología. El sistema DICast combina los datos y las observaciones del modelo de NWP para producir pronósticos ajustados de elementos meteorológicos sensibles, así como otros elementos meteorológicos derivados o personalizados. El sistema está completamente automatizado, se actualiza con la frecuencia necesaria y produce previsiones para extensiones de pronóstico personalizadas y resoluciones temporales. El sistema DICast se ha utilizado como base para numerosos proyectos de RAL, incluidas las áreas de transporte, energía eólica, energía solar y agricultura. [14]

- ***Hurricane Harvey, Forecasting Weather With Machine Learning Artificial Intelligence***. La temporada de huracanes se está calentando para 2017 con la tormenta tropical Harvey, que se convertirá en un huracán de categoría 2 o 3 en el Golfo de México y afectará a Texas. Los comerciantes, los primeros en responder, los pronosticadores por igual se basan en los datos. La tecnología a través del aprendizaje automático de inteligencia artificial se está utilizando para mejorar los pronósticos. Los comerciantes de gas natural y petróleo en los últimos años han visto cómo los servicios meteorológicos tradicionales se han visto afectados por patrones climáticos difíciles e impredecibles. Hedgers y especuladores por igual confían en esta extrapolación de datos. ¿Cómo lo hacemos mejor? Cada vez hay más satélites circulando por la Tierra y tenemos múltiples modelos climáticos que, según nos dicen, son más poderosos, pero muchos con un éxito limitado después de 5 a 10 días. [15]

- ***Machine Learning for Sales Forecasting Using Weather Data*.** WalMart es una compañía con miles de tiendas en 27 países. Es posible encontrar varios artículos sobre los mecanismos tecnológicos utilizados para gestionar la logística y la distribución de los productos. Es la segunda vez que ofrecen un concurso en Kaggle con la intención de encontrar candidatos para entrevistas para trabajos de científicos de datos. Una gran ventaja de este tipo de competencia es que tenemos acceso a los datos de grandes empresas y comprendemos qué problemas están tratando de resolver con modelos probabilísticos. El objetivo de la competencia era crear un modelo que pudiera predecir la cantidad de ventas de algunos productos en tiendas específicas en los días previos y posteriores a tormentas de nieve y tormentas. El ejemplo que dieron en la descripción de la tarea fue la venta de paraguas, que intuitivamente debe ver un aumento antes de una gran tormenta. [16]

- ***A system for airport weather forecasting based on circular regression trees***. Este documento describe un conjunto de herramientas

y un modelo para mejorar la exactitud de los pronósticos meteorológicos aeroportuarios producidos por los productos numéricos de predicción meteorológica (NWP), aprendiendo de las relaciones entre los datos previamente modificados y los observados. Esto se basa en una nueva metodología de aprendizaje automático que permite que las variables circulares se incorporen naturalmente en los árboles de regresión, produciendo resultados más precisos que las metodologías de árbol de regresión circular lineal y previa. El software se ha puesto a disposición del público como un paquete de Python, que contiene todas las herramientas necesarias para extraer el PNT histórico y los datos meteorológicos observados y para generar pronósticos de variables climáticas diferentes para cualquier aeropuerto del mundo. Se presentan varios ejemplos donde los resultados del modelo propuesto mejoran significativamente los producidos por PNT y también por modelos de árboles de regresión previos. [17]

- ***Evaluation of Machine Learning Techniques for Green Energy Prediction*.** Se evalúan las técnicas de aprendizaje automático para la predicción de la energía verde (eólica, solar y de biomasa) en función de las predicciones meteorológicas. El clima está constituido por múltiples atributos: temperatura, cobertura de nubes, velocidad / dirección del viento, que son variables aleatorias discretas. Uno de los objetivos es predecir el clima en función de los datos meteorológicos anteriores. Además, se tiene interés en encontrar la correlación (dependencias para reducir la dimensionalidad del conjunto de datos) entre estas variables, predecir datos faltantes predecir desviaciones en proyecciones meteorológicas (para la programación de trabajos dentro del centro de control verde), encontrar agrupaciones dentro de los datos (constituido por variables estrechamente relacionadas, por ejemplo, PCA que pueden utilizarse para eliminar variables redundantes), clasificación, búsqueda (modelos de regresión de SVM no lineales), capacitación de redes

neuronales artificiales basadas en los datos históricos para que puedan utilizarse para la predicción en el futuro. [18]

- ***Weather Prediction Based on Decision Tree Algorithm Using Data Mining Techniques***. Se Investiga el uso de técnicas de minería de datos para pronosticar atributos como la temperatura máxima y la temperatura mínima. Esto se llevó a cabo utilizando los algoritmos de *Decision Tree* y los datos meteorológicos recopilados entre 2012 y 2015 de las diferentes ciudades. Los enfoques de predicción meteorológica son desafiados por fenómenos climáticos complejos. Los fenómenos meteorológicos tienen muchos parámetros como temperatura máxima, temperatura mínima, humedad y velocidad del viento que son imposibles de enumerar y medir. En los conjuntos de datos disponibles, aplicamos el Algoritmo del Árbol de decisiones para eliminar los datos inapropiados. En general, la temperatura máxima y la temperatura mínima son las principales responsables de la predicción meteorológica. En cuanto al porcentaje de estos parámetros, pronosticamos que se trata de una caída de frío o de calor total o de nieve. Este documento desarrolla un modelo que utiliza el árbol de decisión para predecir fenómenos climáticos como el frío total, el calor total y la caída de la nieve, que pueden ser una información que salve vidas. [19]

- ***Forecasting Weather to Predict Rainfall for Sustainable Agriculture using Machine Learning Techniques***. Hay varios métodos de pronóstico disponibles para predecir las precipitaciones. La técnica de propagación de retorno es uno de los métodos que pueden usarse para predecir la precipitación, ya que esta técnica tiene algunos inconvenientes, la máquina de vectores de soporte se usa para reducir las deficiencias del algoritmo de propagación de retorno y desempeña un papel muy importante en la agricultura sostenible. El rendimiento de la técnica de sistema propuesta se analiza y compara con el algoritmo de propagación de retorno. En este trabajo se examina que la máquina de vectores de

soporte ofrece un mejor rendimiento que la técnica de propagación de retorno. [7]

- ***Machine Learning Techniques for Short-Term Rain Forecasting System in the Northeastern Part of Thailand***. Este artículo presenta la metodología de los enfoques de aprendizaje automático para el sistema de pronóstico de lluvia a corto plazo. El árbol de decisiones, la red neuronal artificial (ANN) y la máquina de vectores de soporte (SVM) se aplicaron para desarrollar modelos de predicción y predicción de climas para pronósticos de lluvia. Los objetivos de esta presentación son demostrar (1) cómo se puede usar la selección de características para identificar las relaciones entre ocurrencias de lluvia y otras condiciones climáticas y (2) qué modelos se pueden desarrollar y desplegar para predecir las estimaciones precisas de lluvia para respaldar las decisiones de lanzar las operaciones de siembra de nubes en la parte nororiental de Tailandia. Conjuntos de datos recopilados durante 2004-2006 del Centro de Investigación Chalermprakiat Royal Rain Making en Hua Hin, Prachuap Khiri Khan, el Centro de Investigación Real de Lluvia Chalermprakiat en Pimai, Nakhon Ratchasima y el Departamento Meteorológico Tailandés (TMD). Un total de 179 registros con 57 características se fusionaron y se combinaron por fecha única. Hay tres partes principales en este trabajo. En primer lugar, se utilizó un algoritmo de inducción de árbol de decisión (C4.5) para clasificar el estado de lluvia en lluvia o sin lluvia. La precisión general del árbol de clasificación alcanza el 94.41% con la validación cruzada de cinco veces. El algoritmo C4.5 también se usó para clasificar la cantidad de lluvia en tres clases: sin lluvia (0-0.1 mm), pocas lluvias (0.1-10 mm) y lluvia moderada (> 10 mm). y la precisión general del árbol de clasificación alcanza el 62.57%. En segundo lugar, se aplicó una ANN para predecir la cantidad de lluvia y se utilizó el error cuadrático medio (RMSE) para medir el entrenamiento y los errores de prueba de la ANN. Se encuentra que la ANN arroja un RMSE menor a 0.171 para las estimaciones de lluvia diarias, en comparación

con la estimación del día siguiente y de los próximos 2 días. En tercer lugar, las técnicas ANN y SVM también se utilizaron para clasificar la cantidad de lluvia en tres clases: sin lluvia, con poca lluvia y con lluvia moderada, como en el caso anterior. Los resultados se lograron en 68.15% y 69.10% de la precisión general de predicción el mismo día para los modelos ANN y SVM, respectivamente. Los resultados obtenidos ilustraron la comparación del poder predictivo de diferentes métodos para la estimación de la precipitación. [8]

- ***Non-Linear Machine Learning Approach to Short-Term Precipitation Forecasting*.** Se emplean modelos efectivos de predicción de precipitación a corto plazo en una gran cantidad de datos meteorológicos espacio-temporales, mediante el uso de una plataforma de algoritmo construida sobre Alibaba Cloud (PAI) 1. Para predecir la cantidad de lluvia en un sitio de observación, se propone extraer las características temporales y geográficas no solo del sitio de observación sino también de sus sitios circundantes. Además, se centra en evaluar el rendimiento de los enfoques de aprendizaje automático no lineal, como las redes neuronales profundas y los modelos de árbol de decisiones con gradiente en la plataforma PAI. Los experimentos exhaustivos muestran que la capacidad de las relaciones complejas entre las características meteorológicas puede describirse por modelos no lineales mejor que los modelos lineales básicos. Se comparan los modelos no lineales con el sistema de predicción de conjuntos ECMWF en tareas de previsión de precipitaciones de 3 horas y se prueba la efectividad de nuestros modelos. [9]

- ***A rainfall forecasting method using Machine Learning models ans its application to the Fukuoka city case.*** En el presente artículo, se intenta derivar métodos óptimos de aprendizaje automático basados en datos para pronosticar una precipitación promedio diaria y mensual de la ciudad de Fukuoka en Japón. Este estudio comparativo se lleva a cabo

concentrándose en tres aspectos: modelado de insumos, elaboración de modelos y técnicas de preprocesamiento. Se realiza una comparación entre el análisis de correlación lineal y la información mutua promedio para encontrar una técnica de entrada óptima. Para el modelado de la lluvia, se propone un nuevo método híbrido multi-modelo y se lo compara con sus modelos constituyentes. Los modelos incluyen la red neuronal artificial, splines de regresión adaptativa multivariante, el vecino k-másrest y la regresión vectorial de apoyo de base radial. Cada uno de estos métodos se aplica para modelar la precipitación diaria y mensual, junto con una técnica de preprocesamiento que incluye el promedio móvil y el análisis de componentes principales. En la primera etapa del método híbrido, los submodelos de cada uno de los métodos anteriores se construyen con diferentes configuraciones de parámetros. En la segunda etapa, los submodelos se clasifican con una técnica de selección de variables y los modelos de mayor clasificación se seleccionan en función del error de validación cruzada de dejar salir uno. La previsión del modelo híbrido se realiza mediante la combinación ponderada de los modelos finalmente seleccionados. [10]

- ***Rainfall Forecasting Using Neural Network: A Survey***. Un pronóstico preciso de las lluvias es muy importante para los países dependientes de la agricultura, como la India. Para analizar la productividad de los cultivos, el uso de los recursos hídricos y la planificación previa de los recursos hídricos, la predicción de las precipitaciones es importante. Las técnicas estadísticas para la predicción de precipitaciones no pueden funcionar bien para las predicciones de precipitaciones a largo plazo debido a la naturaleza dinámica de los fenómenos climáticos. Las Redes Neuronales Artificiales (ANN) se han vuelto muy populares, y la predicción usando ANN es una de las técnicas más utilizadas para la predicción de precipitaciones. Este documento proporciona una encuesta detallada y una comparación de las diferentes arquitecturas de redes neuronales utilizadas por los investigadores para la predicción de precipitaciones. El

documento también discute los problemas al aplicar diferentes redes neuronales para el pronóstico de lluvia anual / mensual / diaria. Además, también presenta diferentes medidas de precisión utilizadas por los investigadores para evaluar el rendimiento de ANN. [11]

- ***Machine Learning Techniques For Rainfall Prediction: A Review***. Debido a la naturaleza dinámica de la atmósfera, las técnicas estadísticas no proporcionan una buena precisión para la predicción de precipitaciones. La no linealidad de los datos de lluvia hace que la red neuronal artificial sea una mejor técnica. El trabajo de revisión y la comparación de diferentes enfoques y algoritmos utilizados por los investigadores para la predicción de lluvias se muestran en forma tabular. La intención de este documento es brindar a los no expertos un fácil acceso a las técnicas y enfoques utilizados en el campo de la predicción de lluvias. [12]

- ***Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting***. El objetivo de *precipitation nowcasting* es predecir la intensidad de lluvia futura en una región local durante un período de tiempo relativamente corto. Muy pocos estudios previos han examinado este crucial y desafiante problema de pronóstico del clima desde la perspectiva del aprendizaje de la máquina. En este trabajo, se formula la predicción de precipitación como un problema de predicción de secuencia espaciotemporal en el que tanto los datos de entrada como la predicción son secuencias espaciotemporales. Al extender el LSTM completamente conectado (FC-LSTM) para que tenga estructuras convolucionales en las transiciones de entrada a estado y de estado a estado, se propone el LSTM convolucional (ConvLSTM) y lo usamos para construir un extremo a otro modelo entrenable para el problema de predicción inmediata de la precipitación. Los experimentos muestran que nuestra red ConvLSTM captura mejor las correlaciones espaciotemporales y supera de manera consistente a FC-LSTM y al

algoritmo ROVER operacional de última generación para predicción de precipitación. [13]

4.3 Aporte de nuestro proyecto en el citado contexto

De todo lo presentado en los *papers* anteriores, podemos concluir que prácticamente no existen tanto en Uruguay como en el mundo, soluciones que consoliden múltiples fuentes de datos para contribuir a la mejora del pronóstico del tiempo.

Si bien se han realizado esfuerzos al respecto, la problemática de Uruguay respecto al pronóstico del clima es que existe escases de recursos (observadores, estaciones), inversión (bajo presupuesto), personal capacitado (egresan pocos meteorólogos anualmente) y tecnología (falta de radar Doppler).

Sin ir más lejos Japón, que tiene algo más del doble de territorio que Uruguay, cuenta con 1300 estaciones meteorológicas contra las 21 convencionales y 17 automáticas que dispone el país. De esas 21 convencionales, solo 3 de ellas funcionan las 24hs, Rocha, Aeropuerto de Carrasco y Laguna del Sauce. Aún hoy cerca de 19 policías ofician de agentes pluviométricos en algunos puntos del país, dado que existen pluviómetros ubicados en comisarías.

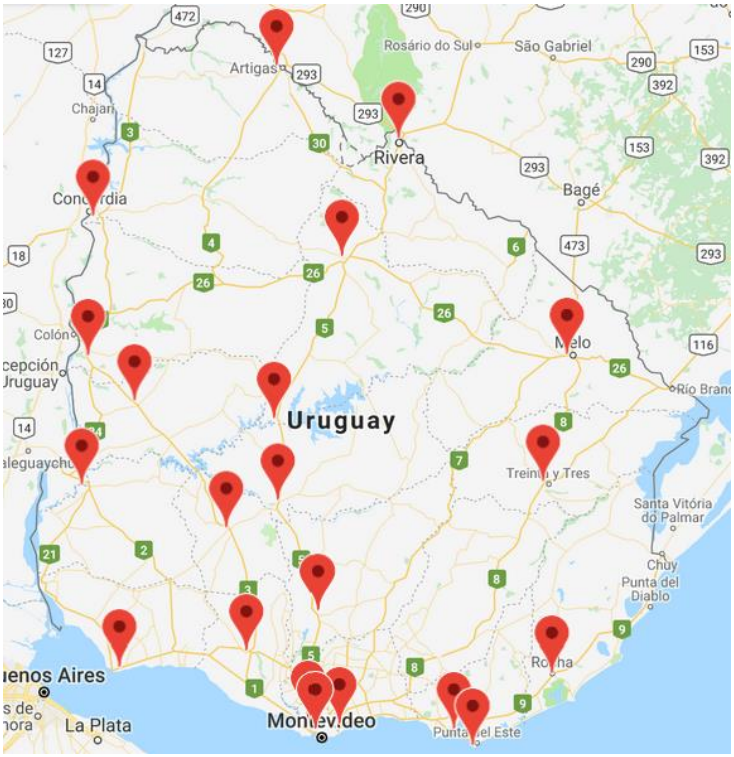


Figura 39 – Estaciones meteorológicas convencionales - INUMET
Fuente: [29]

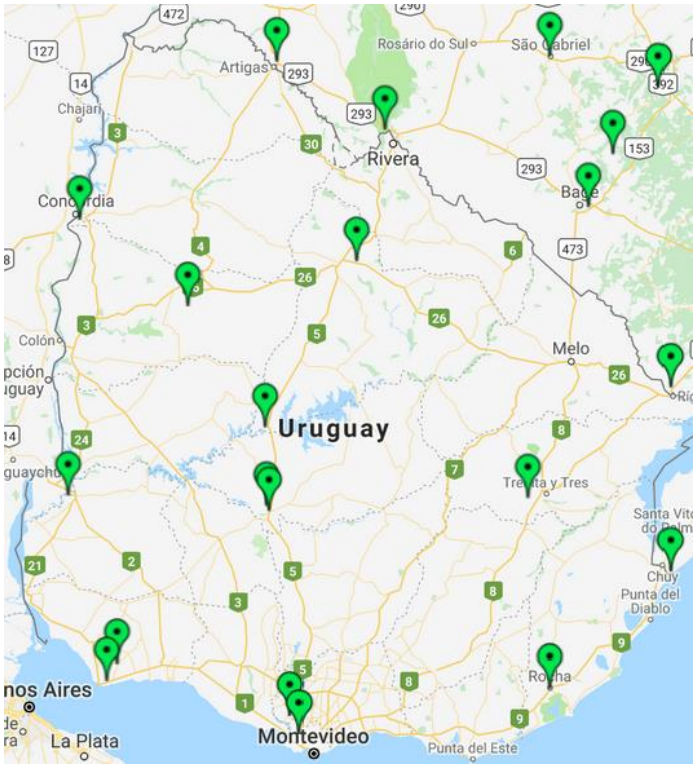


Figura 40 – Estaciones meteorológicas automáticas - INUMET
Fuente: [29]

Por tanto, no solo la recopilación de datos de distintas fuentes es lo que aportaría valor a la situación actual que vive el país, sino que también disponer de estaciones meteorológicas automáticas de bajo costo como las PWS interconectadas en una red que cubran distintas áreas del territorio permitirían sustancialmente enriquecer cualquier modelo y por tanto elevar la posibilidad de obtener mejores pronósticos.

5. Análisis, diseño e implementación de la solución

En este capítulo se detalla la solución propuesta en este trabajo y detalles de su análisis, diseño e implementación.

5.1 Plataforma

5.1.1 Solución de alto nivel y componentes

En el siguiente gráfico se muestra la solución de alto nivel de la aplicación. A la izquierda podemos identificar las diferentes fuentes de datos utilizadas:

- Prototipo de estación meteorológica (PWS)
- Servicios que se consumen desde internet (proveedores de información meteorológica de coordenadas geográficas de interés).

La información proveniente de la PWS es recibida por los servicios *IOT Hub* y *Stream Analytics* para que finalmente se almacene en el repositorio Azure Storage. Por otra parte, las APIs REST se consumen desde aplicaciones *logic apps* y la información obtenida es guardada en el mismo repositorio. El modelo predictivo desarrollado en *Machine Learning studio* se alimenta de la información proveniente de las fuentes descritas y de datos históricos proporcionados por Inumet e IBM. En los tableros desarrollados en Power BI se muestra información en tiempo real proveniente de la mini estación y la predicción de precipitaciones realizada por el modelo. Dicho modelo se expone como un servicio web y puede ser consumido desde otras aplicaciones como Excel, aplicaciones móviles, etc.

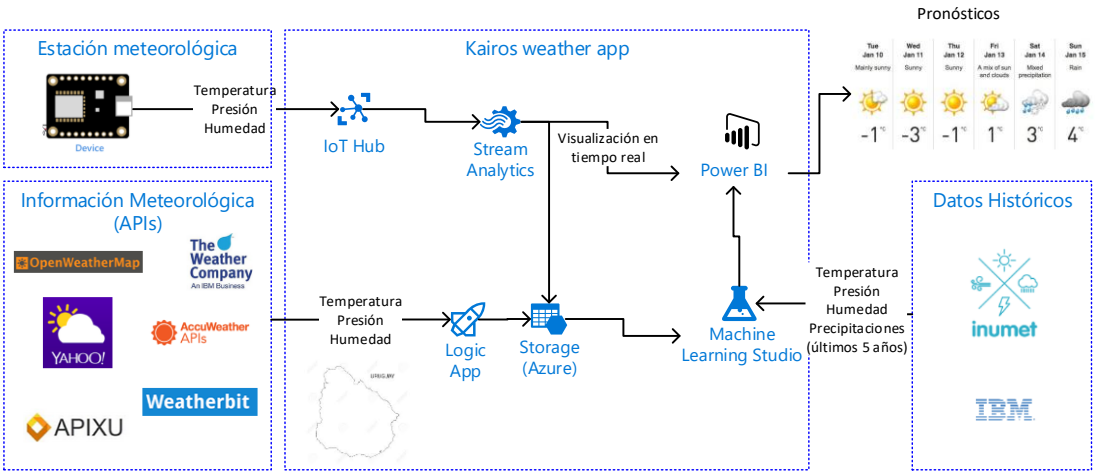


Figura 41 – Diagrama a alto nivel de la solución

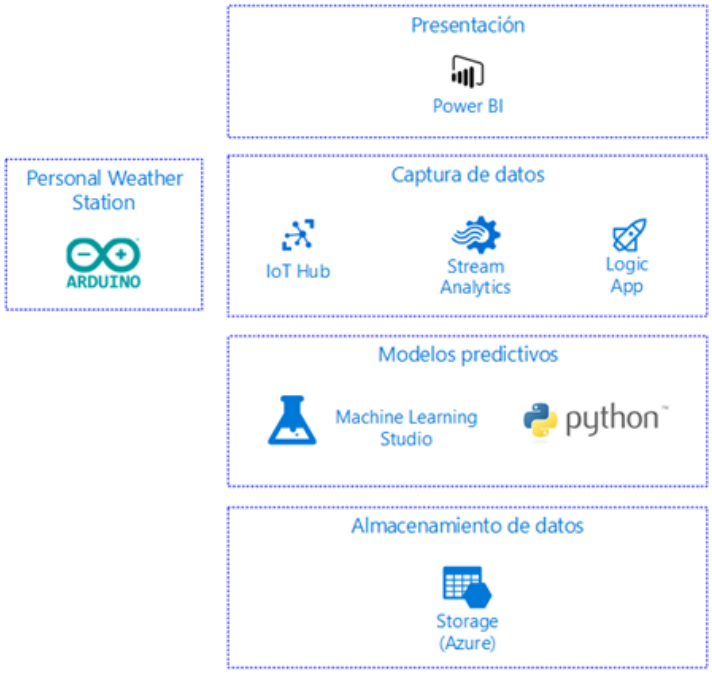


Figura 42 – Componentes de la solución

5.1.2 Descripción de la arquitectura

Para cumplir con este cometido se propone mostrar las estructuras del sistema y documentar las principales decisiones de diseño utilizando diferentes perspectivas o vistas aplicando el enfoque "*Views and Beyond*".

A continuación, se muestran las diferentes vistas del sistema (Módulos, Componentes y Conectores, Asignación).

5.1.2.1 Vistas de Módulos

- **Vista de Descomposición**

  La siguiente es la descomposición del sistema en sus diferentes módulos.

  Representación primaria

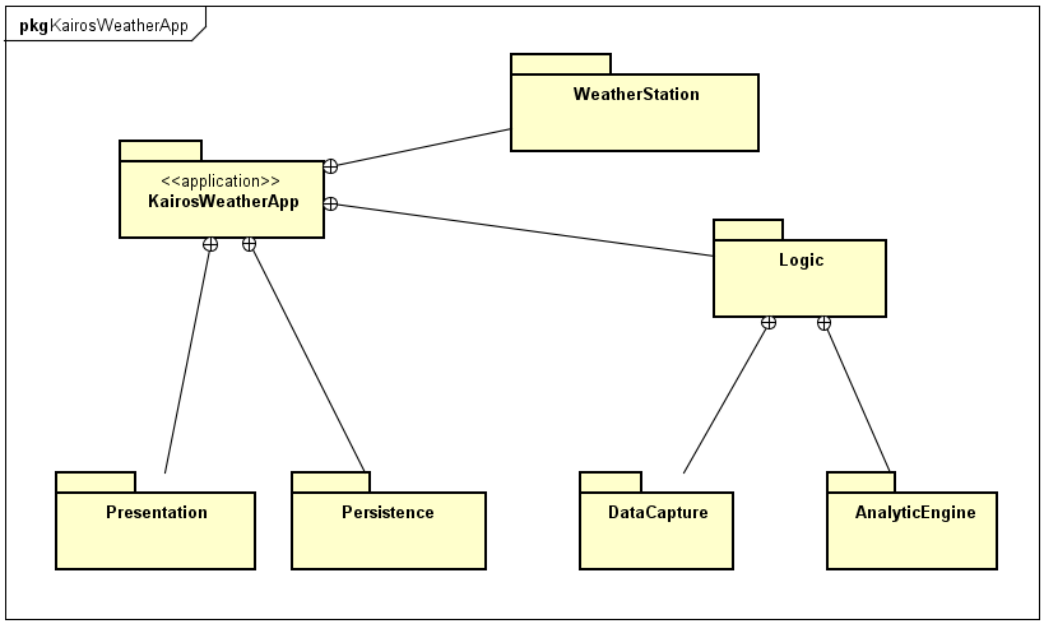


Figura 43 – Diagrama de descomposición del sistema

- **Vista de Uso**

La vista de uso describe las dependencias de usos entre los módulos del sistema.

Representación primaria

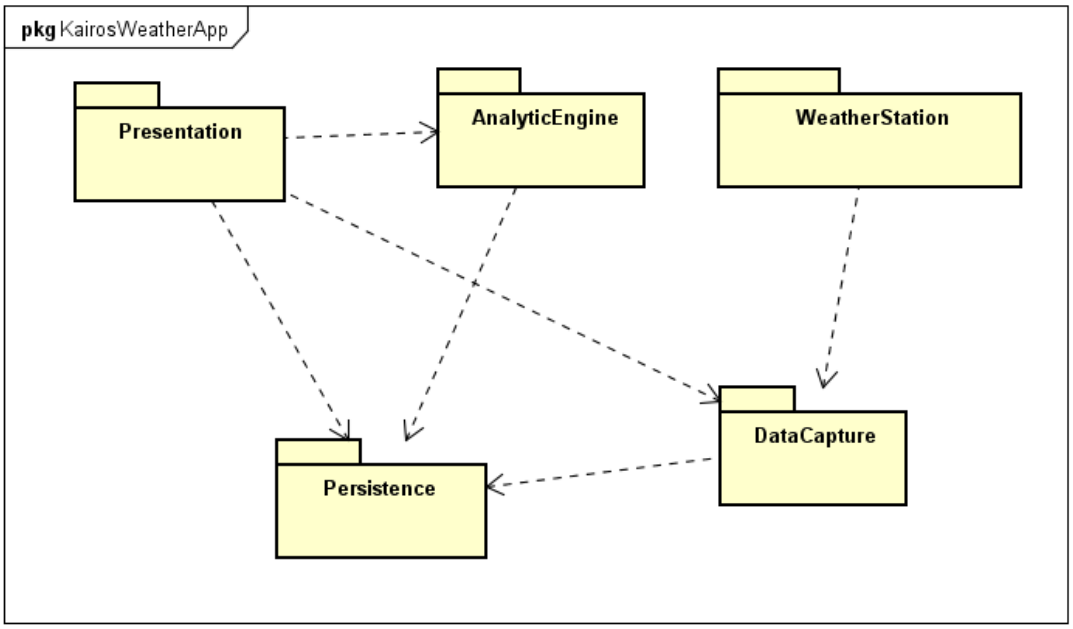


Figura 44 – Diagrama de usos del sistema

- **Catálogo de elementos**

En la siguiente tabla se describen las responsabilidades de cada uno de los módulos que componen el sistema:

Elemento	Responsabilidades
Presentation	Implementa la interfaz con el usuario. Publica paneles con información de la *Personal Weather Station*, resultados del modelo predictivo e informes.
WeatherStation	Implementación de una mini estación meteorológica utilizando Arduino UNO.
DataCapture	Implementa la lógica que procesa la información meteorológica procedente de las fuentes externas (APIs).
AnalyticEngine	Implementa el modelo predictivo de *Machine Learning*.

Persistence	Implementa la persistencia de la información meteorológica en un repositorio escalable y altamente disponible.

5.1.2.2 Vistas de *Layers*

Representación primaria

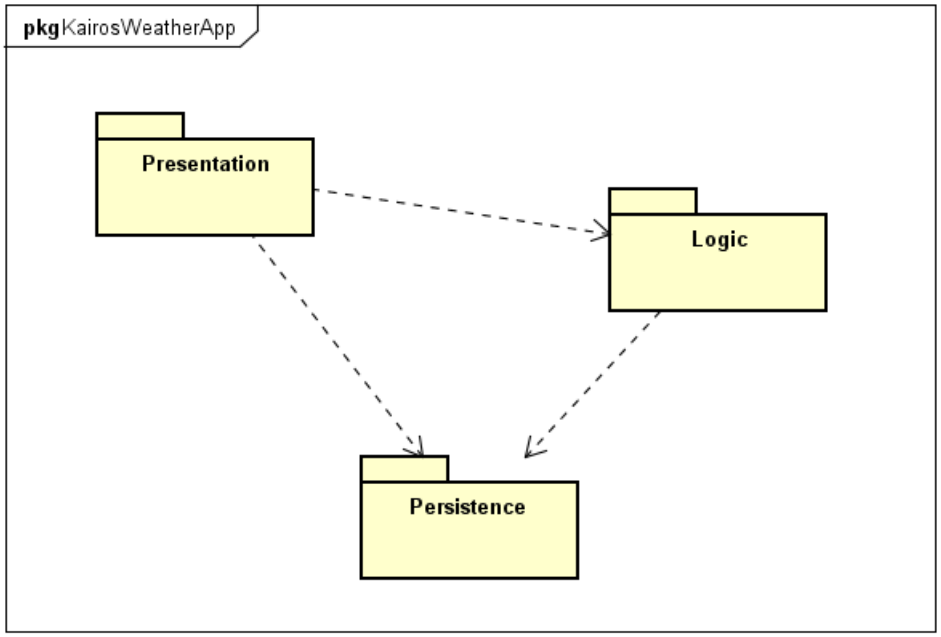


Figura 45 – Diagrama de capas

5.1.2.3 Vistas de Componentes y conectores

Esta sección describe las vistas de componentes y conectores que consideramos relevantes para comunicar la visión del sistema en tiempo de ejecución. En particular se describen los componentes, las formas de conexión y la interacción entre los mismos para explicar la implementación de funcionalidades o de mecanismos claves.

- **Representación primaria**

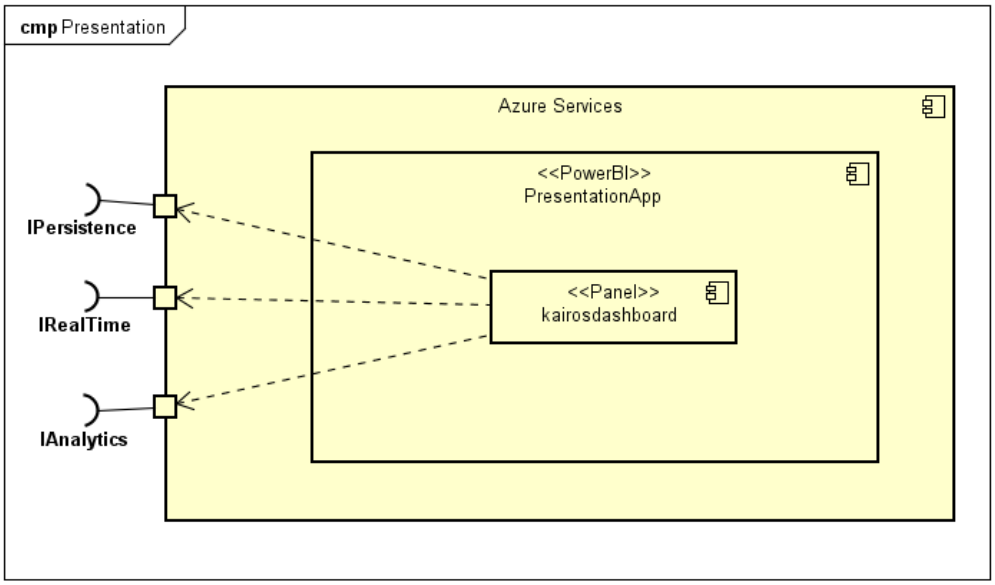


Figura 46 – Diagrama de componentes y conectores de la presentación

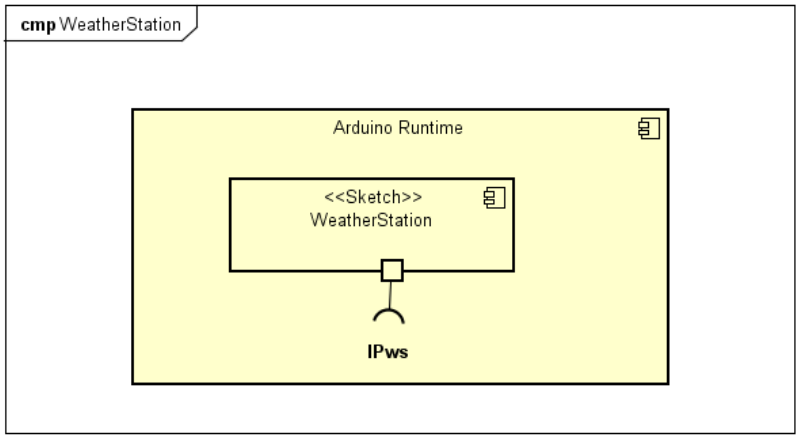


Figura 47 – Diagrama de componentes y conectores de la mini estación

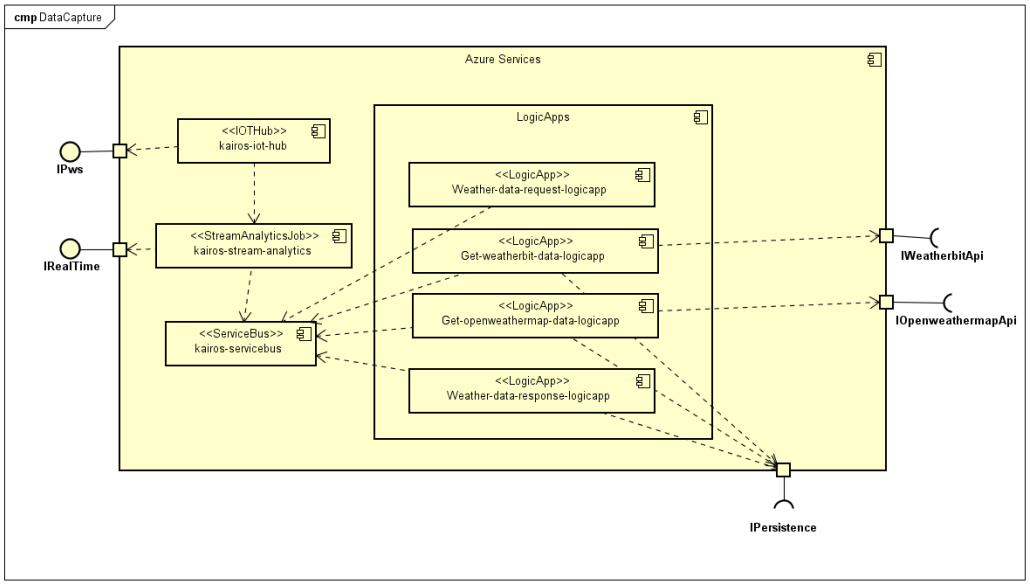


Figura 48 – Diagrama de componentes y conectores de la captura de datos

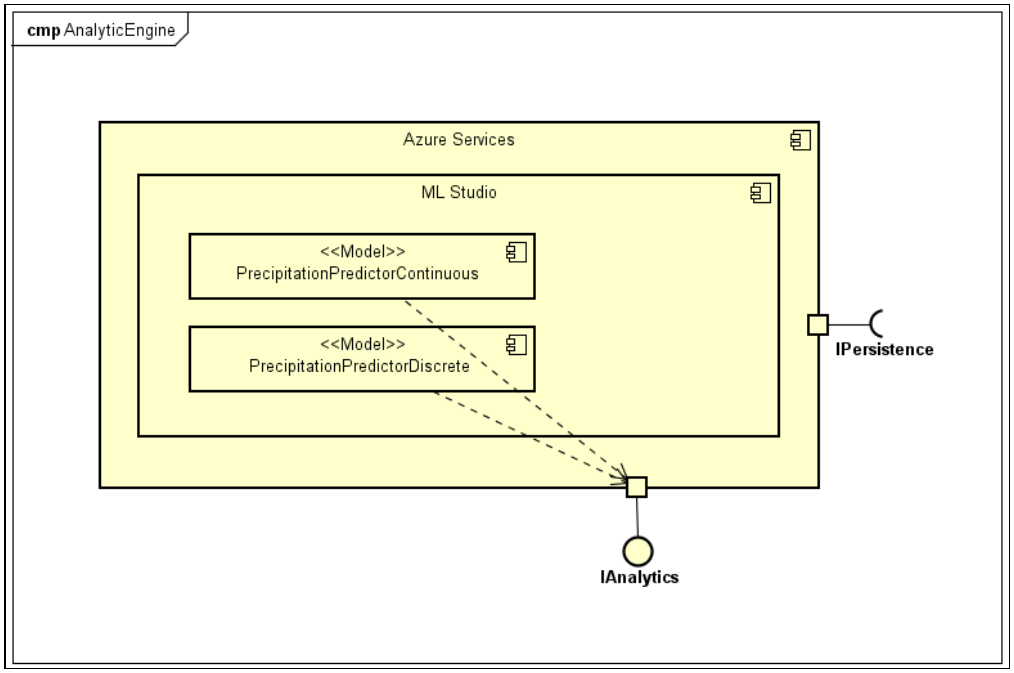


Figura 49 – Diagrama de componentes y conectores del modelo analítico

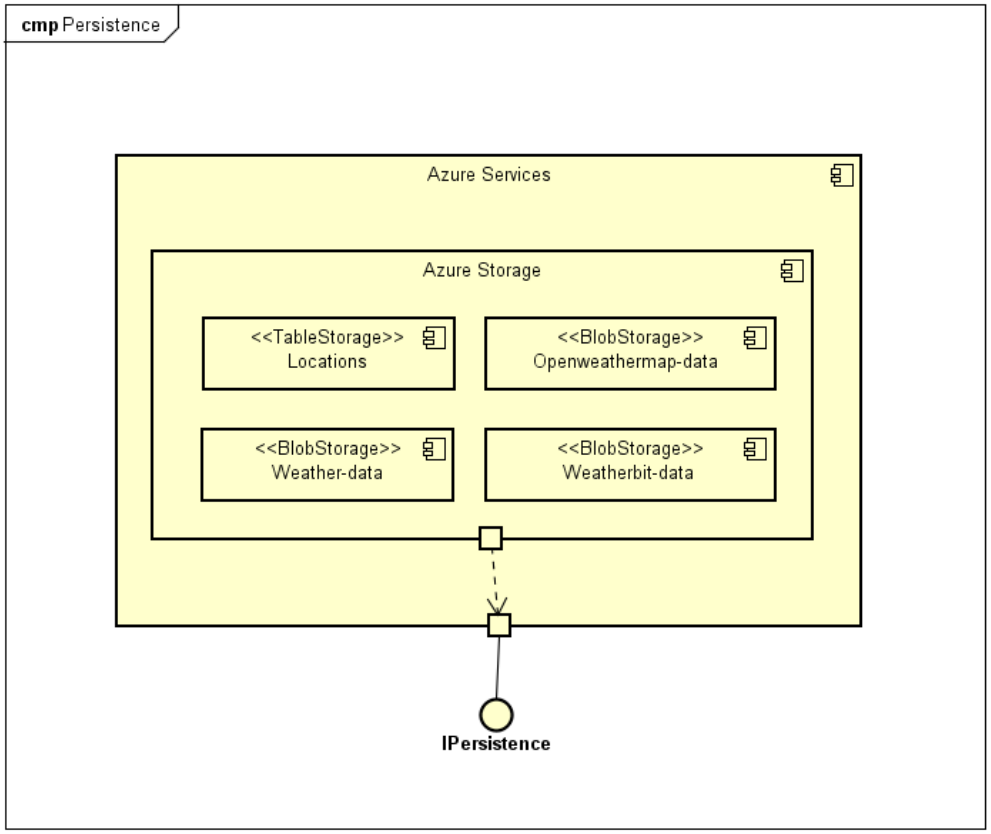


Figura 50 – Diagrama de componentes y conectores de la persistencia

- **Catálogo de elementos**

Componente/conector	Tipo	Descripción
kairosdashboard	Panel de *Power BI*	Panel con información de la *Personal Weather Station*, resultados del modelo predictivo e informes.
WeatherStation	*Sketch* de Arduino	Implementación de una mini estación meteorológica utilizando Arduino UNO.
kairos-iot-hub	*IOT Hub*	Permite la comunicación entre la mini estación y el *core* de la aplicación.
kairos-stream-analytics	*Stream Analytics Job*	Procesa los mensajes provenientes del *IOT Hub* y permite consultarlos en tiempo real.

kairos-servicebus	*Service Bus*	Componente de mensajería para la comunicación entre los demás componentes de la aplicación (Tópicos).
weather-data-request-logicapp	*Logic App*	Aplicación que obtiene las coordenadas geográficas para las cuales se desea obtener datos meteorológicos y publica las solicitudes de información en un tópico.
get-weatherbit-data-logicapp	*Logic App*	Aplicación que resuelve la lógica de invocación a la API de Weatherbit, almacena la respuesta en un repositorio propio y publica la misma en un tópico de salida (formato normalizado). Se inicia cuando recibe la solicitud en el tópico de entrada.
get-openweathermap-logicapp	*Logic App*	Aplicación que resuelve la lógica de invocación a la API de OpenWeatherMap, almacena la respuesta en un repositorio propio y publica la misma en un tópico de salida (formato normalizado). Se inicia cuando recibe la solicitud en el tópico de entrada.
weather-data-response-logicapp	*Logic App*	Aplicación que obtiene los mensajes de respuesta con información meteorológica proveniente de las APIs y de la mini estación y la almacena en el repositorio de datos normalizado.
PrecipitationPredictor Discrete	Modelo de Azure ML	Modelo de *Machine Learning* que realiza predicciones de valores discretos (llueve, no llueve) en base a los parámetros presión, humedad y temperatura.

PrecipitationPredictor Continuous	Modelo de Azure ML	Modelo de *Machine Learning* que realiza predicciones de la cantidad de lluvia en mm en base a los parámetros presión, humedad y temperatura.
Locations	*Table Storage*	Repositorio de coordenadas geográficas de los puntos donde se quiere consultar las variables meteorológicas.
Openweathermap-data	*Blob Storage*	Repositorio de la información obtenida de la API de OpenWeatherMap
Weatherbit-data	*Blob Storage*	Repositorio de la información obtenida de la API de Weatherbit.
Weather-data	*Blob Storage*	Repositorio la información de las APIs y de la mini estación en un formato normalizado.

5.1.2.4 Vistas de Asignación

Esta sección describe los estilos que consideramos relevantes para comunicar la forma como los elementos de software, principalmente los componentes y conectores se relacionan con su entorno. En particular se describe la forma como los componentes se despliegan en los diferentes nodos físicos (servidores).

- **Representación primaria**

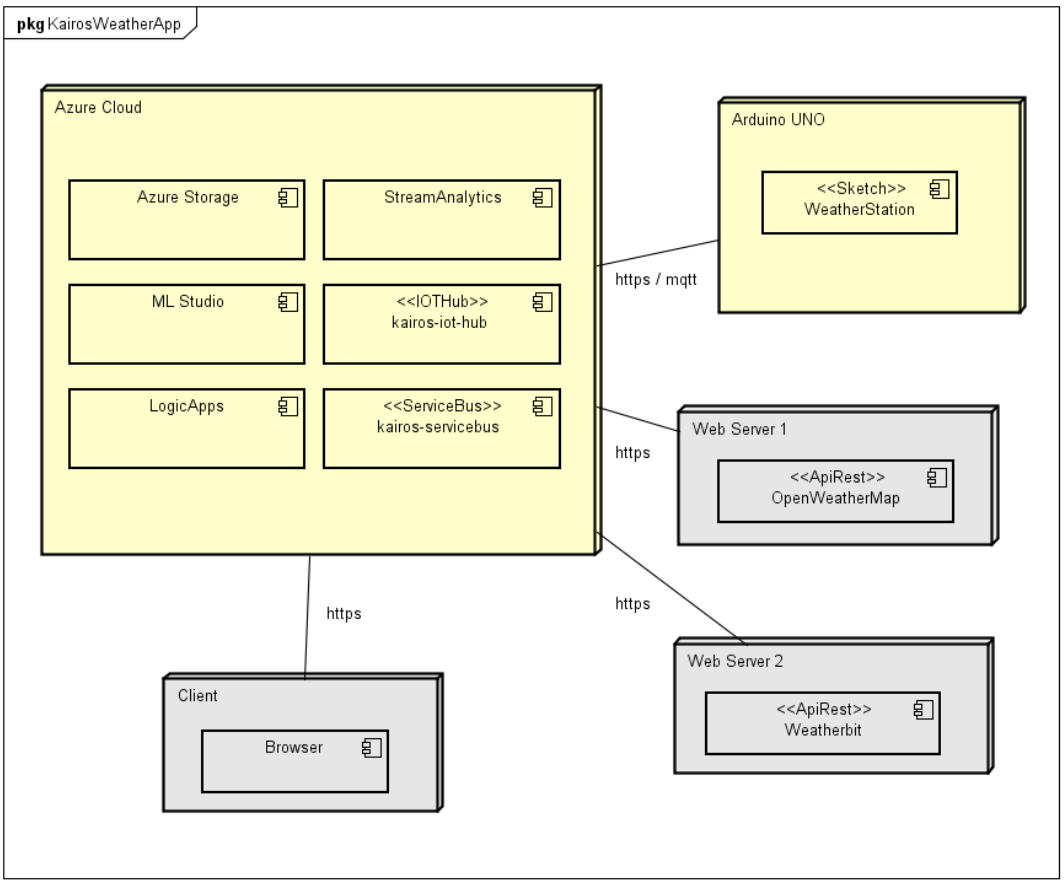


Figura 51 – Diagrama de despliegue

- **Catálogo de elementos**

Nodo	**Descripción**
Client	Equipo del usuario
Azure Cloud	Infraestructura de servicios en la nube de Microsoft
WebServer 1	Servidor web donde se encuentra publicada la API de OpenWeatherMap.
WebServer 2	Servidor web donde se encuentra publicada la api de Weatherbit.
***Arduino* UNO**	Dispositivo de hardware equipado con sensores que implementa la mini estación meteorológica.

Conector	Descripción
https	Protocolo de comunicación utilizado para consumir las APIs. Se podría utilizar para comunicar la mini estación con Azure.
mqtt	Protocolo de comunicación para comunicar la mini estación con Azure.

5.1.3 Primer prototipo en IBM Cloud

Se realizaron pruebas con el objetivo de resolver la problemática de consumir los servicios desde Bluemix (IBM *Cloud*) y de almacenar la información obtenida en un repositorio común para que posteriormente alimente el modelo predictivo.

Si bien comprobamos que es factible su utilización, nos encontramos con algunos problemas en la plataforma (caídas en el *kernel* y elevado tiempo de procesamiento) que nos hicieron tomar la decisión de migrar a Microsoft Azure.

La figura 52 se muestra a nivel general el flujo implementado:

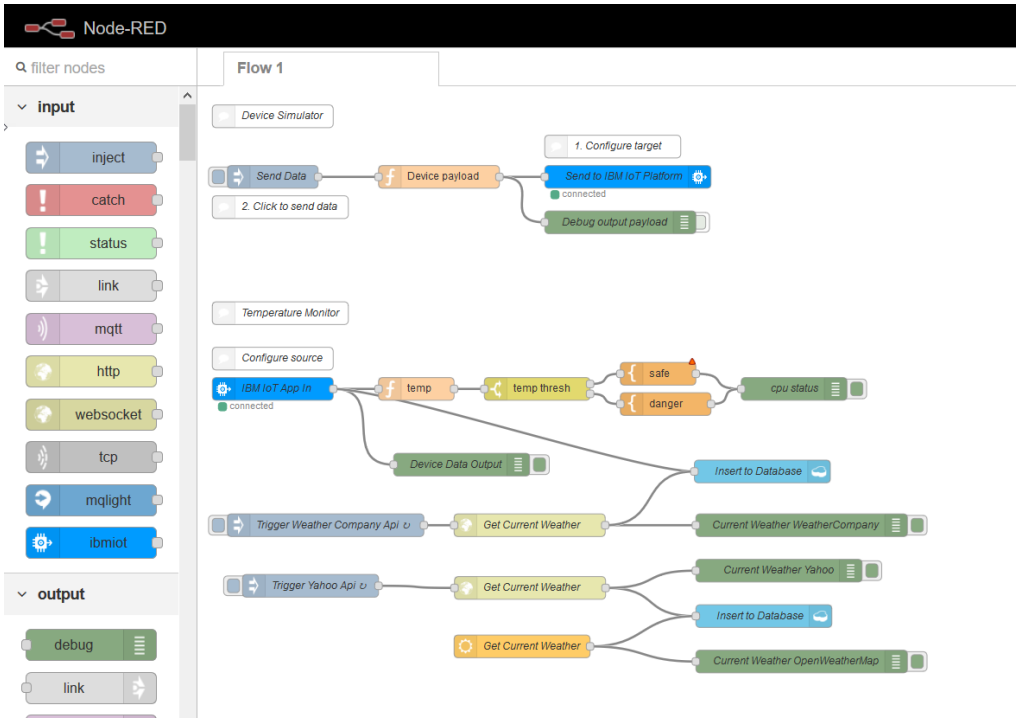


Figura 52 - Flujo de captura de datos implementado en *Node-RED*

El mismo se puede dividir en tres partes de acuerdo al problema que resuelve:

Simulación de una PWS

- Esta parte emula la transmisión de datos meteorológicos desde un dispositivo de hardware a la plataforma de IOT de Bluemix.
- Al presionar el botón del *trigger* gris genera una lectura de datos y es transmitida a la plataforma.

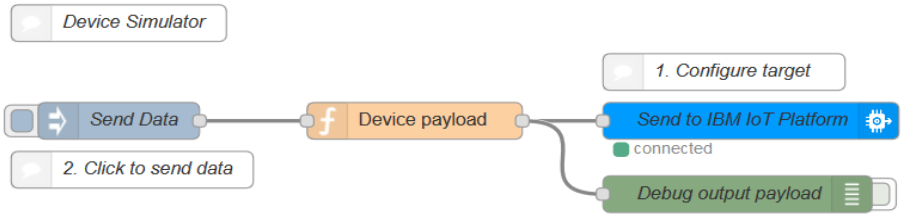


Figura 53 - Flujo que emula el envío de datos desde una estación

Almacenamiento de la información de la PWS

1. A recibir los datos provenientes del dispositivo emulado inserta los mismos en la base de datos.

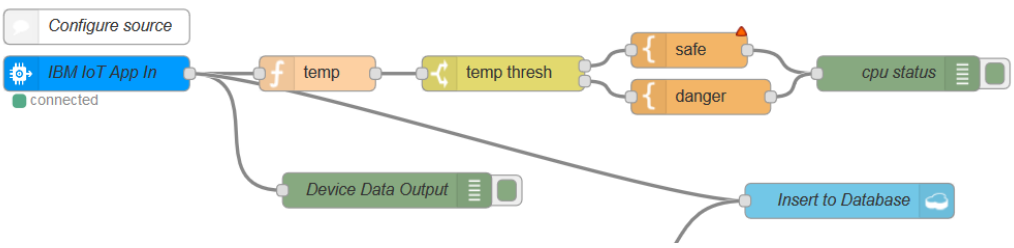


Figura 54 - Procesamiento y almacenamiento de datos meteorológicos

Invocación de tres APIs y almacenamiento de la información

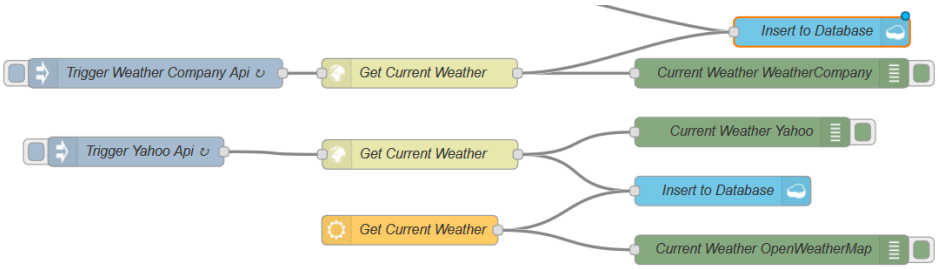


Figura 55 - Invocación de APIs externas y almacenamiento de la información obtenida

1. Al presionar el botón del *trigger* gris realiza una invocación mediante HTTP y consulta la condición actual del tiempo para una coordenada geográfica perteneciente a la ciudad de Montevideo (*The Weather Company*).
2. De manera análoga el segundo *trigger* realiza la misma consulta, pero a la API de Yahoo.
3. La caja naranja realiza una llamada a la API de OpenWeatherMap y devuelve la información si las condiciones cambiaron desde la última lectura realizada.
4. En cualquiera de los tres casos la información obtenida se guarda en la base de datos.

La base de datos utilizada en la prueba se denomina weather-test (no relacional). La misma almacena documentos en formato JSON con toda la información brindada por las fuentes de datos.

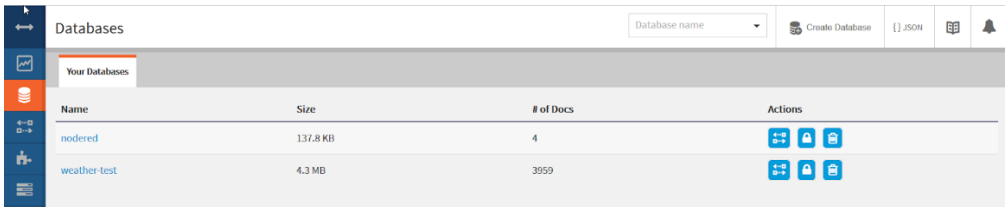


Figura 56 - Base de datos Cloudant NoSQL DB

En las siguientes figuras se muestran ejemplos de lecturas almacenadas:

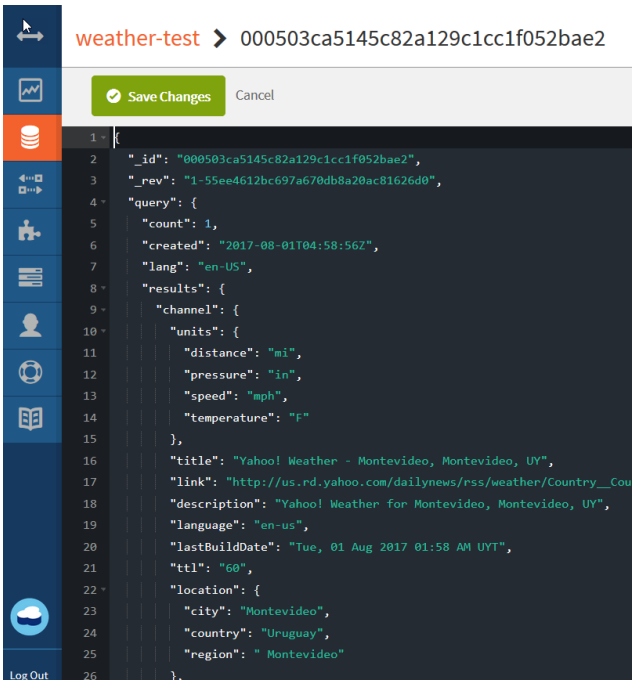


Figura 57 - Información obtenida de la API Yahoo

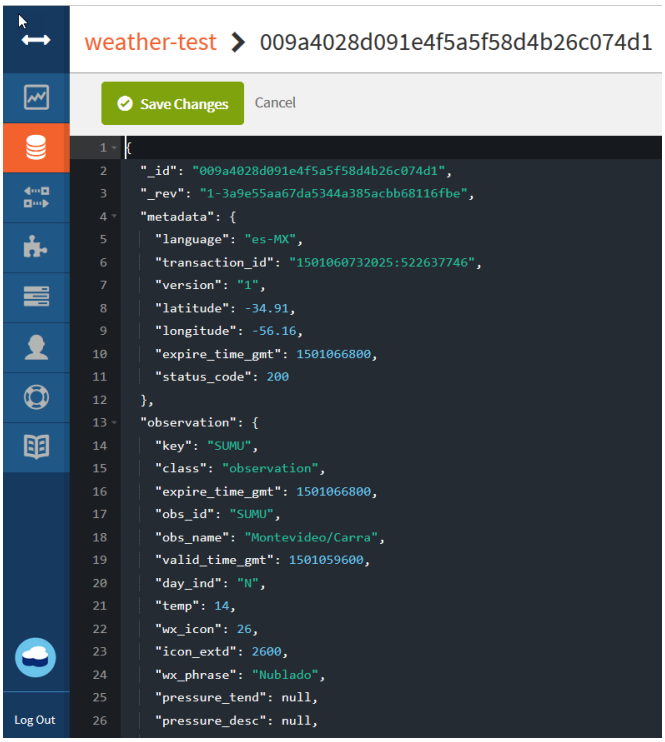


Figura 58 - Información obtenida de la API *The Weather Company*

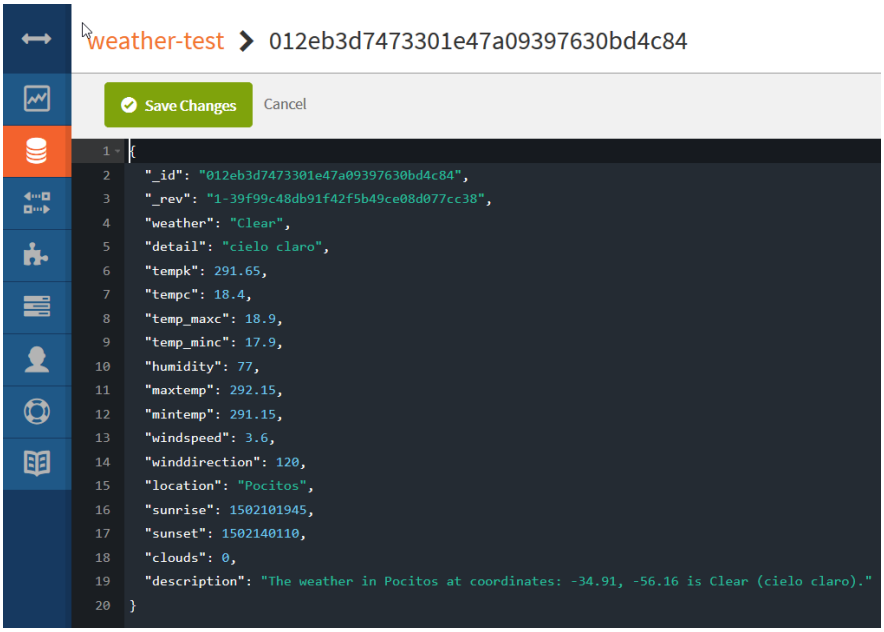


Figura 59 - Información obtenida de la API OpenWeatherMap

Posteriormente a las pruebas de concepto de captura de datos, se realizaron pruebas con la herramienta DSX, puntualmente se creó un modelo predictivo base (*baseline*) con el cual se verificó el desempeño de la herramienta en un contexto bastante básico.

En dichas pruebas pudimos identificar varios problemas (problemas con el *kernel*). Adicionalmente a nuestra experiencia también tuvimos *feedback* de nuestro tutor, ya que el mismo software se estaba utilizando en un curso de maestría que actualmente dicta la universidad.

5.1.4 Aplicaciones de capturas de datos en Azure

Los problemas mencionados con IBM *Cloud* llevaron a tomar la decisión de optar por un cambio en la plataforma de *cloud*. La herramienta elegida fue Microsoft Azure.

Debido al cambio, tuvimos que realizar nuevamente pruebas de concepto de la captura de datos, generación de modelos y pruebas de la plataforma de IoT.

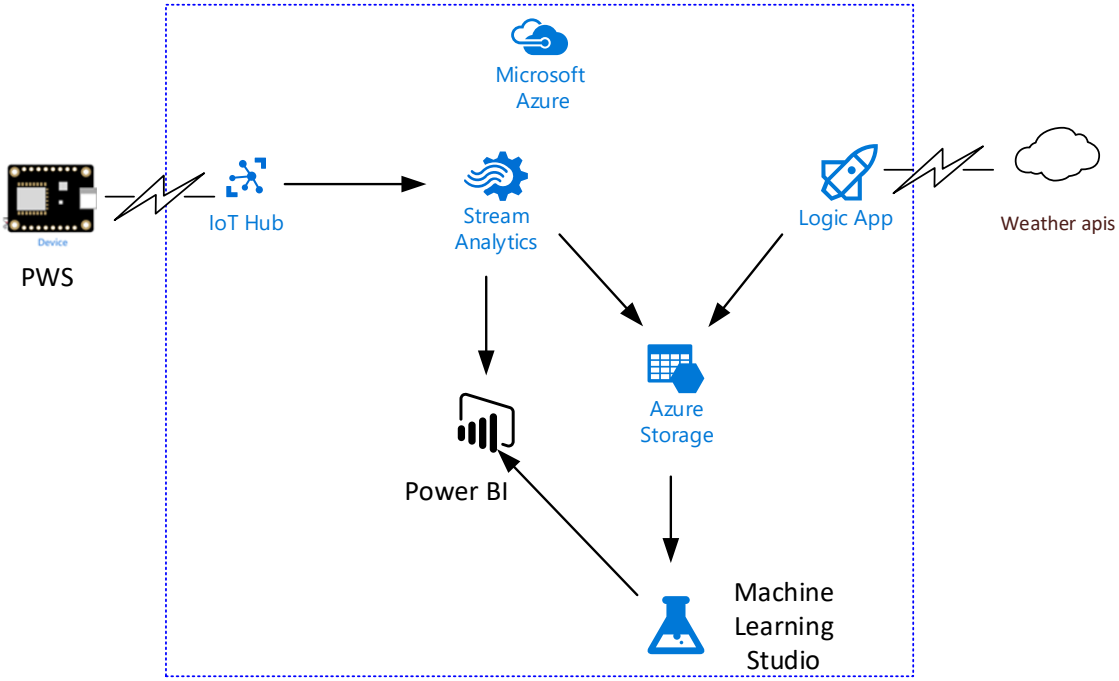


Figura 60 - Servicios de Azure a ser utilizados en la aplicación

Captura de datos de las diversas fuentes

Para la implementación de la captura de datos de las APIs se utilizó *Logic Apps*. Luego de realizar una prueba de concepto y obtener buenos resultados, se diseñó una arquitectura genérica y extensible de forma de poder ir acoplando diferentes fuentes de datos con el menor impacto posible.

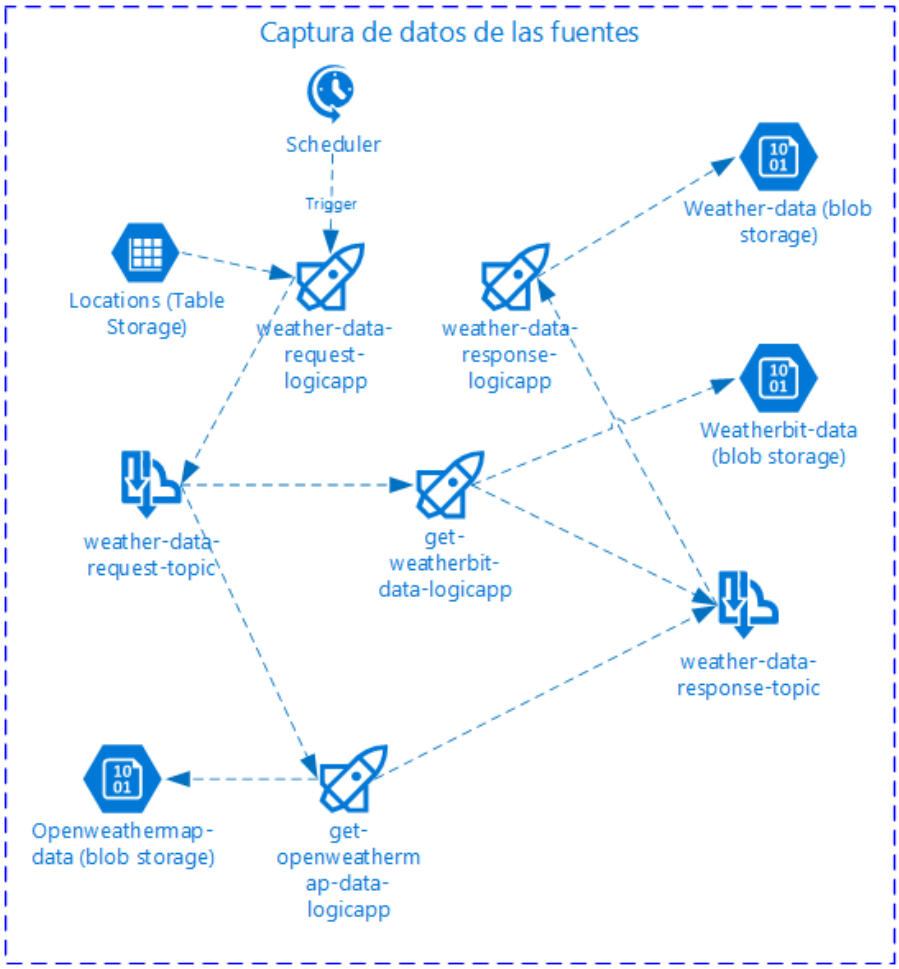


Figura 61 – Arquitectura de aplicación de captura de datos

Para la implementación inicial se tomaron como referencia dos APIs (OpenWeatherMap y Weatherbit).

A continuación, se describen los diferentes componentes que forman parte de la solución propuesta:

***Locations (Table Storage)*:** Es el repositorio de ubicaciones geográficas de las que se desean obtener las mediciones de las variables meteorológicas.

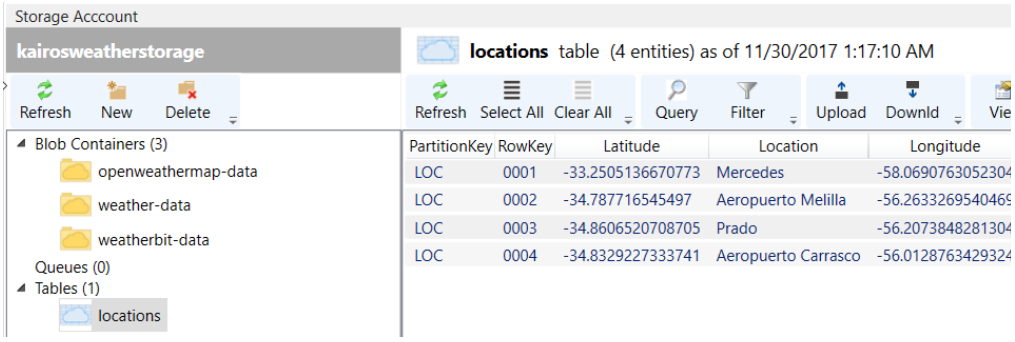


Figura 62 - Tabla de ubicaciones geográficas en Azure *Storage*

Como punto de partida se ingresaron las coordenadas de las cuatro estaciones para las que se solicitaron datos a INUMET.

Weather-data-request-logicapp: Es una aplicación (*logic app*) que se inicia periódicamente cada una hora, obtiene las ubicaciones geográficas de la tabla *locations* e inserta un mensaje por cada ubicación en el tópico Weather-data-request-topic.

Flujo de la aplicación:

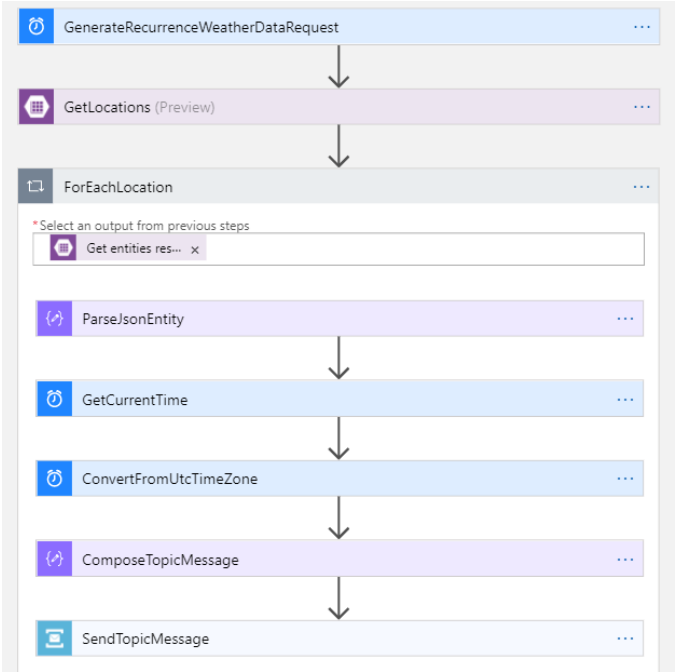


Figura 63 - Aplicación Weather-data-request-logicapp

Weather-data-request-topic: Tópico definido utilizando el servicio *Service Bus* de Azure para almacenar las solicitudes de consulta de datos meteorológicos.

Get-weatherbit-data-logicapp: Es una aplicación (*logic app*) que se inicia al recibir un mensaje desde el tópico Weather-data-request-topic. Obtiene las coordenadas geográficas del mensaje, invoca la API WeatherBit, guarda la respuesta en su repositorio particular y procesa la misma transformándola a un formato normalizado. Finalmente publica un mensaje en el tópico Weather-data-reponse-topic.

Flujo de la aplicación:

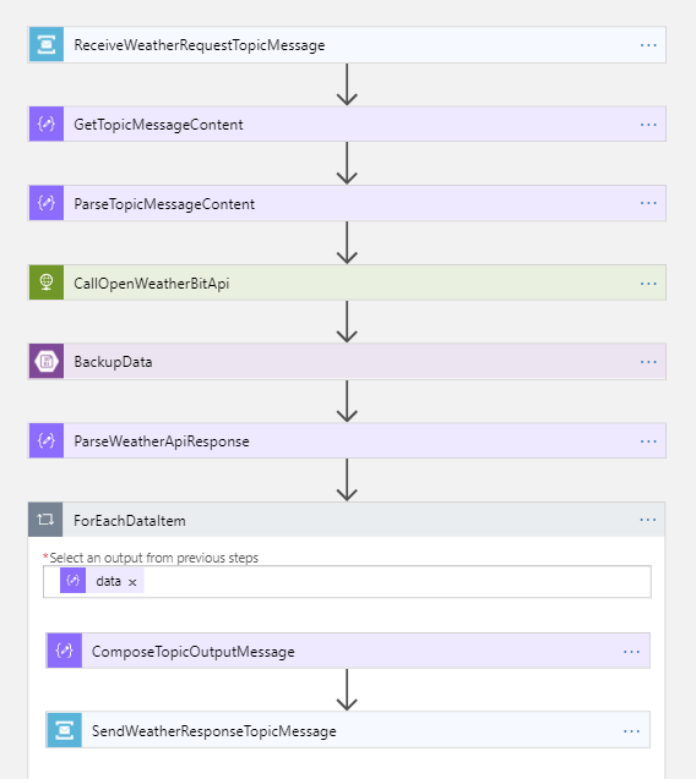


Figura 64 - Aplicación Get-weatherbit-data-logicapp

Weatherbit-data (*blob storage*): Repositorio particular de la API Weatherbit (la idea es respaldar la información en el formato original por si en un futuro requerimos utilizar otra información provista por la API).

Get-openweathermap-data-logicapp: Ídem a la aplicación anterior, pero invocando a la API de OpenWeatherMap.

Flujo de la aplicación:

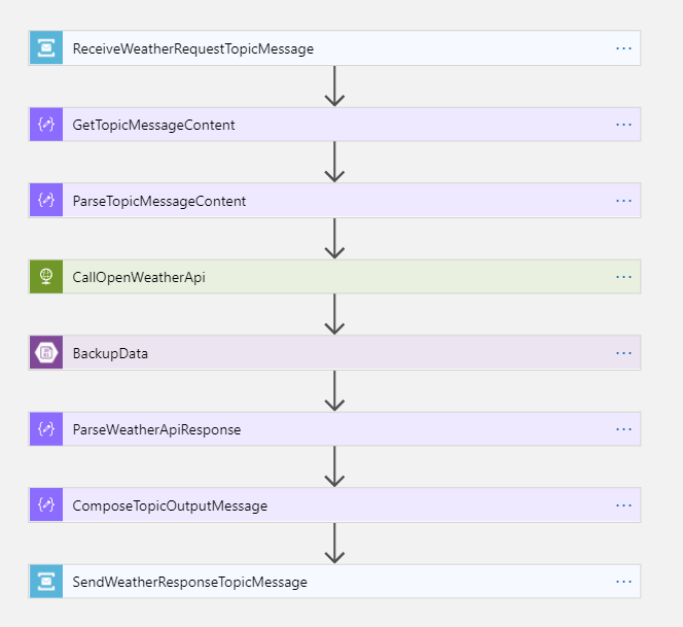


Figura 65 - Aplicación Get-openweathermap-data-logicapp

Openweathermap-data (*blob storage*): Repositorio particular de la API OpenWeatherMap (la idea es respaldar la información en el formato original por si en un futuro requerimos utilizar otra información provista por la API).

Weather-data-response-logicapp: Es una aplicación (*logic app*) que se inicia al recibir un mensaje del tópico Weather-data-response-topic con la información meteorológica obtenida de las diferentes coordenadas geográficas por distintas APIs (dichos mensajes están normalizados) y la almacena en el repositorio consolidado llamado Weather-data.

Flujo de la aplicación:

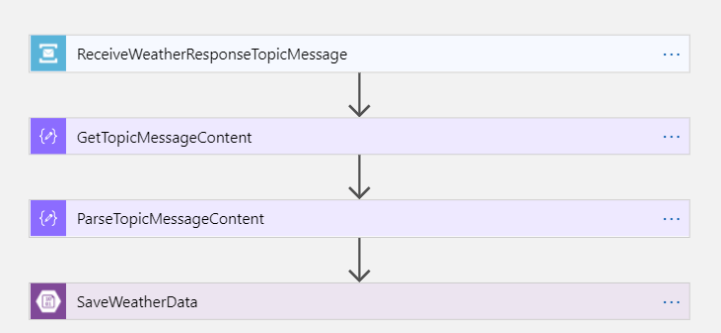


Figura 66 - Aplicación Weather-data-response-logicapp

Weather-data-response-topic: Tópico definido utilizando el servicio *Service Bus* de Azure para almacenar la respuesta obtenida de las APIs en un formato normalizado.

Weather-data (*blob storage*): Repositorio consolidado donde se guarda la información meteorológica proveniente de las diferentes APIs en un formato normalizado.

Visualización de datos en tiempo real

Se generó un panel de prueba para mostrar datos en tiempo real proveniente de una estación meteorológica simulada. Para este desarrollo se usaron los servicios de Azure *IOT Hub*, *Stream Analytics* y *Power B*I.

Se puede acceder al mismo bajo el siguiente *link*:

https://app.powerbi.com/groups/me/dashboards/56062d2b-c96a-4ab4-bda0-d3c441330220

Usuario: kairosweatherapp@itsystemsinnova.com

Contraseña: Kairos01

En la figura 67 se muestran las lecturas en tiempo real enviadas por la estación al IoT Hub.

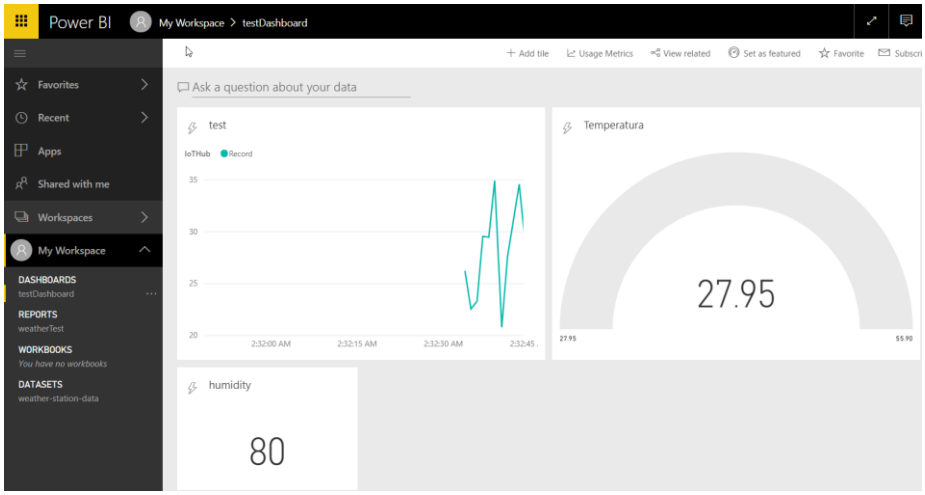


Figura 67 - *Dashboard* de prueba en *Power BI*

En el servicio *Stream Analytics* se configuró como input el *IoT Hub* y como output *Power BI*.

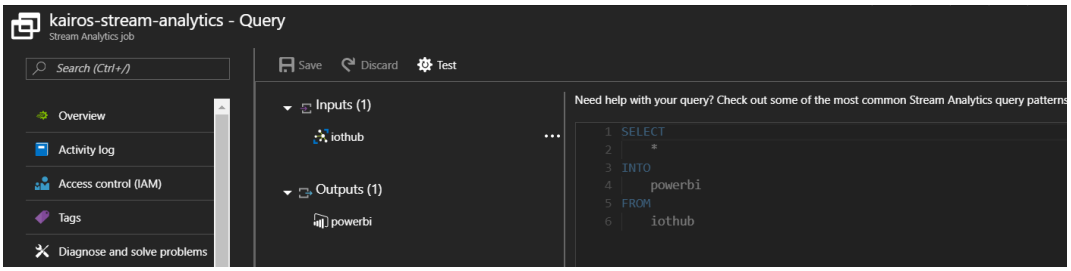


Figura 68 - Configuración de Stream Analytics

Datos generados desde la estación simulada:

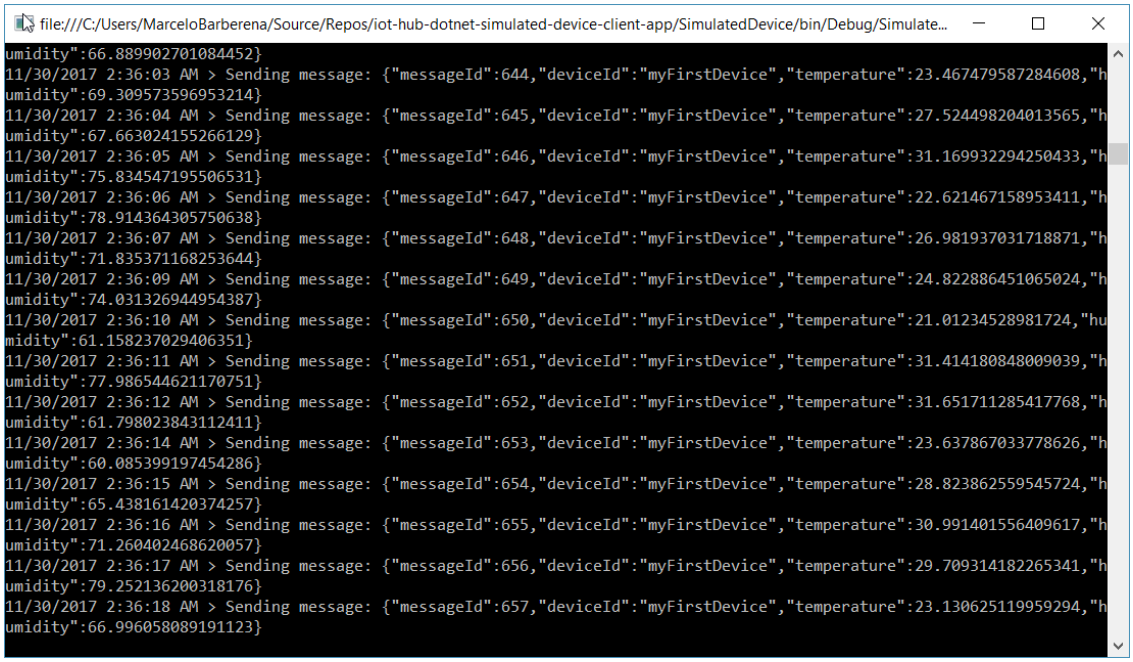


Figura 69 - Datos generados desde la estación simulada

5.1.5 Mini estación meteorológica

Componentes utilizados para la PWS:

Arduino UNO

Figura 70 - Placa Arduino Uno
Fuente: [48]

SparkFun *Weather Shield*

Permite medir variables como Presión barométrica, Humedad, Luminosidad y Temperatura.

Sensores:

- Humedad y Temperatura - HTU21D
- Presión barométrica - MPL3115A2
- Luminosidad - ALS-PT19

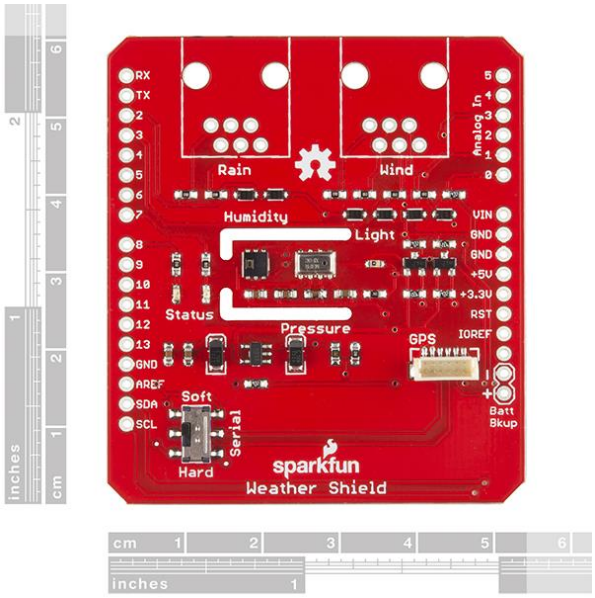


Figura 71 - Placa de sensores Sparkfun *Weather Shield*
Fuente: [32]

Se acoplaron sensores para medir velocidad del viento y dirección (W), pluviómetro (R) mediante conectores RJ11 así como también un GPS para geolocalización.

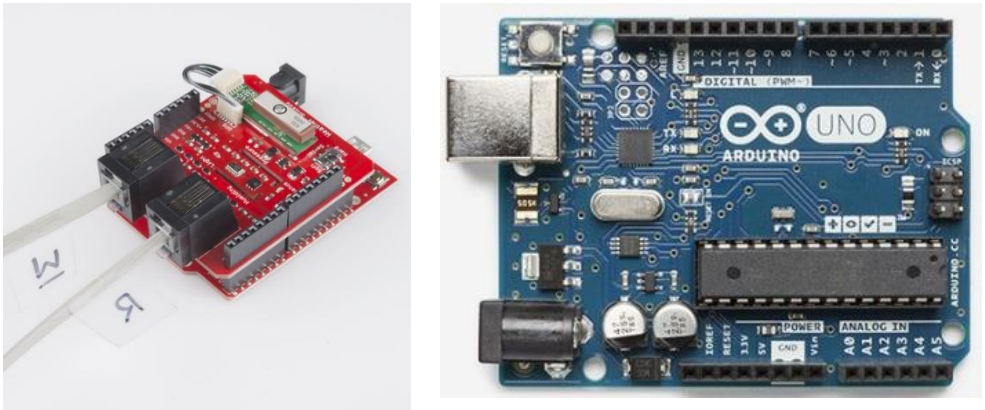


Figura 72 - Integración placa de sensores con *Arduino* Uno
Fuente: [49,48]

La placa de sensores a la que se le añadió un módulo de GPS (GP-735), trabaja junto con una placa Arduino UNO.

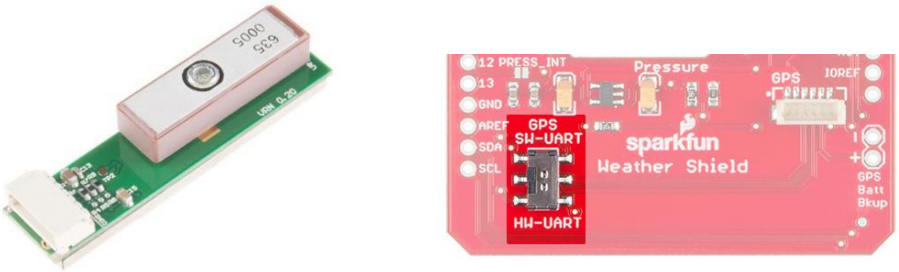


Figura 73 - Integración de GPS a la placa de sensores
Fuente: [51, 49]

Se compraron en plaza algunos componentes de la estación, así como también otros que, si bien no formarán parte de ella, nos han permitido realizar diversas pruebas de concepto que resultaron de utilidad para el prototipo final. La lista completa de componentes es la siguiente:

- Arduino UNO
- *Protoboards*
- Cables hembra/hembra, macho/macho
- Conectores
- Cargador 9V
- Sensor de temperatura y humedad DHT11
- Módulo para tarjeta micro SD
- Módulo reloj RTC DS3231 I2C
- Módulo NodeMCU WiFi ESP8266
- Estaño
- Soldador
- Resistencias
- *Leds*

Se compraron directamente al fabricante los siguientes componentes:

- Placa de sensores Sparkfun *Weather Shield*
- Módulo GPS (GP-735)
- Cable conector JST SH *Jumper* 6 *Wire*
- Conectores RJ11 de 6 pines
- Arduino *Stackable Headers* de 8, 10 y 6 pines.

Una vez obtenida la placa de sensores y el módulo GPS se soldó y acopló la misma a la placa Arduino UNO y se programó el sketch (nombre que

se le designa a un programa Arduino) para configurar, capturar y mostrar la información proveniente de los sensores (WeatherStation.ino).

Se realiza una captura cada cinco minutos del valor de temperatura, humedad, presión atmosférica, junto a la fecha, hora, latitud y longitud proporcionada por el *GPS*.

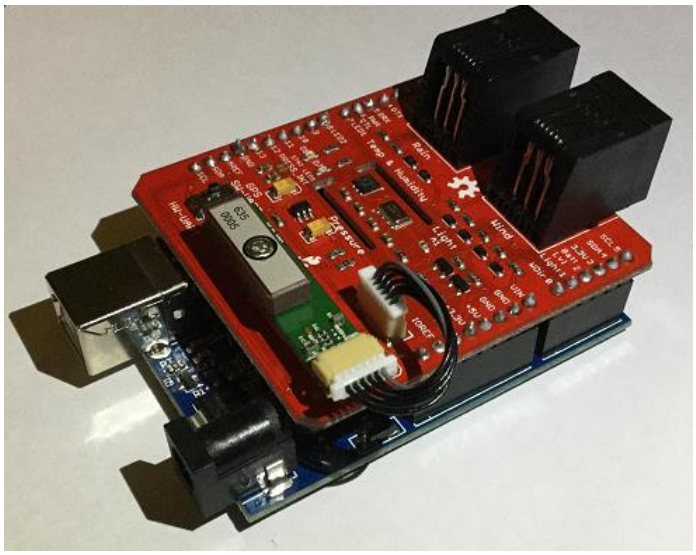


Figura 74 - Ensamblado de estación meteorológica

La información que se puede ver por consola es la siguiente:

```
Humidity=37.2,tempf=71.7,rainin=0.00,dailyrainin=0.00,pressure=101416.00
lat=-34.901321,long=-56.140769,altitude=0.00,sats=5,date=11/25/2017,time=03:55:
```

Figura 75 - Salida sensores estación meteorológica

Los valores obtenidos de la estación fueron cotejados con los publicados en el sitio de INUMET validando su coherencia, pues son muy próximos entre sí para el instante consultado.

El costo de los componentes puede ser consultado en detalle en el Anexo 2 – Mini estación meteorológica.

5.2 Analítica

5.2.1 Análisis de fuentes de datos

Se investigaron varias APIs disponibles que proporcionan información meteorológica para poder determinar si eran de utilidad como fuente de entrada a nuestro modelo predictivo.

OpenWeatherMap **Descripción**: Permite obtener información de las condiciones actuales del clima para una localidad (Nombre, ID Localidad, Coordenadas Geográficas o Código Postal) contando con información proveniente de más de 40.000 estaciones. **Limitaciones**: 60 llamadas por minuto para el plan gratuito. **Actualización de datos**: < 2 horas **Formato**: *JSON, XML, HTML* **Tipo arquitectura**: *REST* **Sitio/*Endpoint***: https://openweathermap.org/api
Weather Company **Descripción**: Permite obtener información de las condiciones actuales del clima para una localidad específica (Coordenadas Geográficas, ID Localidad). **Limitaciones**: 10 llamadas por minuto hasta 10.000 llamadas para el plan gratuito. **Actualización de datos**: Continuamente. **Formato**: *JSON* **Tipo arquitectura**: *REST* **Sitio/*Endpoint***: https://twcservice.mybluemix.net/rest-api/
Yahoo Weather **Descripción**: Permite obtener información de las condiciones actuales del clima para una localidad específica (ID Localidad o Código Postal). **Limitaciones**: 2.000 llamadas por día para el plan gratuito. **Actualización de datos**: - **Formato**: *JSON, XML* **Tipo arquitectura**: *REST* **Sitio/*Endpoint***: https://developer.yahoo.com/weather/
AccuWeather **Descripción**: Permite obtener información de las condiciones actuales del clima para una localidad específica (ID Localidad). **Limitaciones**: 50 llamadas por día para el plan gratuito. **Actualización de datos**: - **Formato**: *JSON*

Tipo arquitectura: *REST* **Sitio/*Endpoint***: https://developer.accuweather.com/
Weatherbit **Descripción**: Permite obtener información de las condiciones actuales del clima para una localidad específica (ID Localidad, Coordenadas Geográficas, Código Postal, Dirección IP, ID Estación) obteniendo información de más de 40.000 estaciones para unas 370.000 ciudades. **Limitaciones**: 45 llamadas por minuto para el plan gratuito. **Actualización de datos**: < 2 horas **Formato**: *HTTP, JSON* **Tipo arquitectura**: *REST* **Sitio/*Endpoint***: api.weatherbit.io https://www.weatherbit.io
Apixu **Descripción**: Permite obtener información de las condiciones actuales del clima para una localidad específica (Nombre Ciudad, Coordenadas Geográficas, Código Postal, Dirección IP). **Limitaciones**: 5.000 llamadas por mes para el plan gratuito. **Actualización de datos**: - **Formato**: JSON, XML **Tipo arquitectura**: **Sitio/*Endpoint***: http://api.apixu.com/v1 https://www.apixu.com/

A continuación, presentamos las mismas, así como también un análisis de las variables relevantes para esta tesis y la medida en la que cada una de ellas satisface la información requerida.

API	OpenWeatherMap	WeatherCompany	Yahoo	Datos Históricos	AccuWeather	Weatherbit	Apixu
Temperatura	✓	✓	✓	✓	✓	✓	✓
Humedad	✓	✓	✓	✓	✓	✓	✓
Dirección del Viento	✓	✓	✓	✓	✓	✓	✓
Velocidad del Viento	✓	✓	✓	✓	✓	✓	✓
Salida Sol	✓		✓			✓	
Puesta Sol	✓		✓			✓	
Nubes	✓			✓		✓	✓
Presión atmosférica		✓	✓	✓	✓	✓	✓
Sensación Térmica		✓					✓
Precipitación				✓	✓	✓	✓
UV		✓			✓		

Figura 76 – Cuadro comparativo de fuentes de datos

5.2.2 Obtención de los datos

Tras un análisis se llegó a la conclusión de la importancia de comenzar tempranamente con la adquisición de datos para alimentarán el modelo predictivo (software) a construir.

Primeramente, evaluamos como plataforma en la nube el producto Bluemix de IBM debido a que nuestro tutor estaba comenzando a trabajar con la tecnología en cursos de postgrado de la Universidad ORT.

En esa línea, tuvimos una breve introducción a la plataforma Bluemix y a la API de *The Weather Company* también de IBM por parte del consultor Diego Aguirre (daguirre@uy.ibm.com). Luego de identificar las fuentes de datos a ser consumidas se decidió realizar una prueba de concepto de la aplicación de captura de datos. De acuerdo a la sugerencia del consultor optamos por implementar la misma utilizando *Node-RED*.

Node-RED es una herramienta de desarrollo visual que permite implementar flujos con enfoque IOT permitiendo integrar APIs de terceros y dispositivos de hardware a través de protocolos estándar como REST y MQTT.

Para ello seleccionamos tres de las APIs descriptas anteriormente, con el objetivo de resolver la problemática de consumir los servicios desde Bluemix y de almacenar la información obtenida en un repositorio común para que posteriormente alimente el modelo predictivo.

Si bien las pruebas de captura de datos y de simulación de una estación resultaron exitosas, cuando comenzamos con las pruebas de DSX identificamos problemas en la plataforma realizando modelos predictivos bastante básicos. Esto último sumado a otros problemas que surgieron en los cursos antes mencionados y a la falta de apoyo por parte de IBM, llevó a la decisión de evaluar la plataforma Azure.

Se volvieron a realizar las pruebas de concepto de captura de datos, simulación de la estación y evaluación de modelos predictivos las que arrojaron resultados exitosos.

IBM proporcionó información histórica en formato csv de cinco años hacia atrás de algunas localidades del interior del país. Se tomó la decisión de utilizar las ubicadas en los departamentos más al sur del territorio nacional como son las de Cerro Colorado (Flores), Mercedes y Tres Bocas (Mercedes).

Se ingresó el 23 de agosto de 2017 el trámite de solicitud en INUMET de información histórica del Sur del país también para cinco años hacia atrás (en forma de medición horaria) para las variables Temperatura, Humedad, Presión Atmosférica y Precipitación.

5.2.3 Análisis de datos exploratorios

- **Estructura del *dataset***

  A continuación, se describe la estructura del conjunto de datos utilizado en el análisis, las transformaciones realizadas y como se resolvió el problema de los datos faltantes. Inicialmente partimos de dos archivos Excel con la siguiente información:

Archivo	Descripción	Tamaño	Cantidad de registros
Inumet-LecturaVariablesPorHora.xlsx	Lecturas por hora de las variables temperatura, presión y humedad en las estaciones meteorológicas “Aeropuerto Carrasco”, “Aeropuerto Melilla”, “Prado” y “Mercedes” en los últimos 5 años.	5653 kb	195840
Inumet-Precipitaciones.xlsx	Lecturas de precipitaciones diarias en todas las estaciones meteorológicas del país en los últimos 5 años	202 kb	2039

Estación	Fecha	HumRelativa[%]	PresAtmMar[hPa]	TempAire[ºC]
Aeropuerto Carrasco	1/1/2012 0:00	83	1016.0	20.0
Aeropuerto Carrasco	1/1/2012 1:00	83	1015.9	19.6
Aeropuerto Carrasco	1/1/2012 2:00	83	1015.2	19.2
Aeropuerto Carrasco	1/1/2012 3:00	83	1015.0	18.8
Aeropuerto Carrasco	1/1/2012 4:00	88	1015.4	18.4
Aeropuerto Carrasco	1/1/2012 5:00	90	1015.9	18.4
Aeropuerto Carrasco	1/1/2012 6:00	94	1016.2	18.0
Aeropuerto Carrasco	1/1/2012 7:00	87	1016.8	20.0
Aeropuerto Carrasco	1/1/2012 8:00	80	1016.7	21.4
Aeropuerto Carrasco	1/1/2012 9:00	79	1016.8	22.4
Aeropuerto Carrasco	1/1/2012 10:00	77	1017.0	23.0
Aeropuerto Carrasco	1/1/2012 11:00	75	1016.8	23.6
Aeropuerto Carrasco	1/1/2012 12:00	75	1016.8	24.0
Aeropuerto Carrasco	1/1/2012 13:00	75	1016.1	24.0
Aeropuerto Carrasco	1/1/2012 14:00	71	1016.0	24.6

Figura 77 - Datos de humedad, presión y temperatura

	Aeropuerto Carrasco	Aeropuerto Melilla	Artigas	Bella Unión	Colonia	Durazno	Florida	Laguna del Sauce	Melo	Mercedes	Paso de los Toros	Paysar
1/1/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/2/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/3/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/4/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/5/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/6/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/7/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/8/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/9/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/10/2012	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/11/2012	3.7	7.0	18.5	2.0	0.8	3.8	6.5	3.0	5.1	9.5	1.0	
1/12/2012	0.0	0.0	0.8	0.4	0.0	0.0	0.0	0.0	4.5	0.5	0.4	
1/13/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/14/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/15/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/16/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/17/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/18/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1/19/2012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Figura 78 - Datos de precipitaciones

Los modelos utilizados requieren que el *dataset* tenga la estructura descripta a continuación.

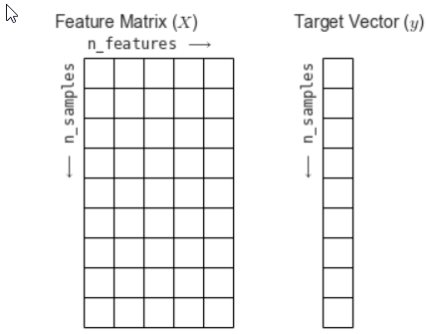


Figura 79 - Estructura de datos requerida por los modelos predictivos

En nuestro caso las *features* corresponden a los valores de temperatura, presión y humedad, las *samples* son el conjunto de mediciones para las estaciones meteorológicas en cierta fecha y el target vector son las cantidades de precipitación o las categorías estimadas (lluvia, no lluvia).

A partir de los dos archivos originales se generó un nuevo archivo con la información consolidada, al que se le agregaron las coordenadas geográficas de las estaciones meteorológicas ya que los métodos tienen una mejor performance con datos numéricos.

Transformaciones aplicadas para normalizar los datos:

- Separar la hora de la fecha
- Convertir las fechas (formato: yyyymmdd) y las horas (formato: hh) a números enteros
- Asignar como separador decimal el punto
- Las etiquetas “s/dato” se sustituyeron por el valor vacío
- Agregar coordenadas geográficas de las estaciones meteorológicas
- Agregar una nueva columna para categorizar (*RAIN*, *NO RAIN*) si el valor de precipitaciones es mayor a cero o cero respectivamente.

El archivo resultante fue el siguiente:

Archivo	Descripción	Tamaño	Cantidad de registros
Inumet-Datos.xlsx	Lecturas por hora de las variables temperatura, presión, humedad y precipitaciones en las estaciones meteorológicas “Aeropuerto Carrasco”, “Aeropuerto Melilla”, “Prado” y “Mercedes” en los últimos 5 años.	62602 kb	195840

Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	0	83	1016.02	20
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	1	83	1015.93	19.6
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	2	83	1015.22	19.2
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	3	83	1015.01	18.8
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	4	88	1015.42	18.4
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	5	90	1015.92	18.4
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	6	94	1016.2	18
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	7	87	1016.8	20
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	8	80	1016.69	21.4
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	9	79	1016.79	22.4
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	10	77	1016.99	23
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	11	75	1016.79	23.6
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	12	75	1016.8	24
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	13	75	1016.1	24
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	14	71	1016	24.6
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	15	71	1016.1	24.6
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	16	75	1015	24
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	17	68	1014.5	24.4
Aeropuerto Carrasco	-34.83292273	-56.01287634	20120101	18	73	1014.3	23.4

Figura 80 - Estructura del *dataset* generado

Luego de generar el Excel con la estructura requerida se guardó en formato csv para importarlo en Python y tomarlo como *input* de los diferentes modelos predictivos.

Al comenzar con el análisis nos enfrentamos a una situación recurrente en este tipo de problemas. En general los *datasets* no están completos, existen mediciones que por diferentes razones no están disponibles (en la jerga se llaman NAs). En la problemática del pronóstico del clima en Uruguay, sabemos que este tipo de faltantes de datos no siempre se da por un problema técnico, sino más bien humano. Puntualmente algunas de las centrales meteorológicas manuales están operativas solo en horarios laborales. Más allá de las razones por las que no tenemos

disponibilidad de datos, tuvimos que aplicar diferentes técnicas para estimar dichos valores de forma que no afecte a la predicción general.

Varias de las técnicas evaluadas para completar los datos faltantes fueron:

- Completar los datos faltantes con ceros
- Completar los datos con el valor anterior
- Completar los datos con el valor medio
- Utilizar árboles de regresión para completar los valores faltantes
- Aplicar técnicas de regresión para estimar dichos valores

- **Análisis cuantitativo inicial**

A continuación, se presenta el análisis cuantitativo inicial del *dataset* mediante estadística descriptiva.

Aplicando el comando *df.describe()* de Python sobre el *dataset*, se obtienen los indicadores estadísticos principales (cantidad de valores, media aritmética, desviación estándar, min, los percentiles y el valor máximo).

	humidity	pressure	temperature
count	195839.000000	195839.000000	195839.000000
mean	71.640644	1015.137888	18.274489
std	17.580853	6.239932	6.511974
min	6.000000	989.700000	-6.600000
25%	59.000000	1010.900000	13.400000
50%	73.000000	1015.020000	18.200000
75%	86.000000	1019.270000	22.900000
max	100.000000	1036.660000	41.200000

Figura 81 - Indicadores estadísticos del *dataset*

- **Variable humedad**

En el siguiente gráfico se muestran las frecuencias para valores de humedad relativa, es decir, que cantidad de lecturas cayeron en determinados rangos de humedad definidos.

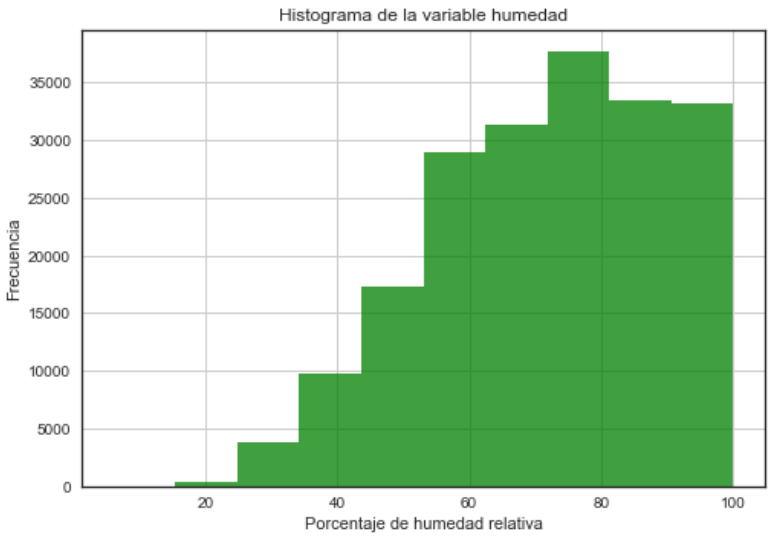


Figura 82 - Histograma de la variable humedad

Humedad relativa (%)			
Desde	Hasta	Frecuencia	Porcentaje
0	15.4	13	0.01%
15.5	24.8	416	0.21%
24.9	34.2	3818	1.95%
34.3	43.6	9786	5.00%
43.7	53	17257	8.81%
53.1	62.4	28949	14.78%
62.3	71.8	31293	15.98%
71.9	81.2	37690	19.25%
81.3	90.6	33411	17.06%
90.7	100	33206	16.96%

Los diagramas de cajas son una presentación visual que describe varias características importantes al mismo tiempo, tales como la dispersión y simetría.

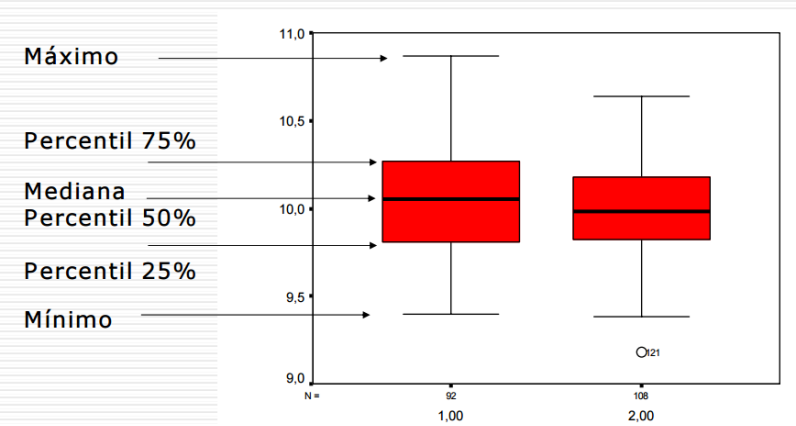


Figura 83 - Interpretación de un diagrama de cajas

Los cuartiles son los tres valores de la variable estadística que dividen a un conjunto de datos ordenados en cuatro partes iguales. Q1, Q2 y Q3 determinan los valores correspondientes al 25%, al 50% y al 75% de los datos. Q2 coincide con la mediana.

En el siguiente gráfico se puede ver que los datos de humedad no son simétricos y se acumulan en el rango 50% - 100 %. Otra observación que se puede realizar, es que la estación de Mercedes registró valores mayores de humedad en el mismo período. Las restantes tres estaciones pertenecen a Montevideo y por lo que puede apreciarse, las medianas tienen valores similares.

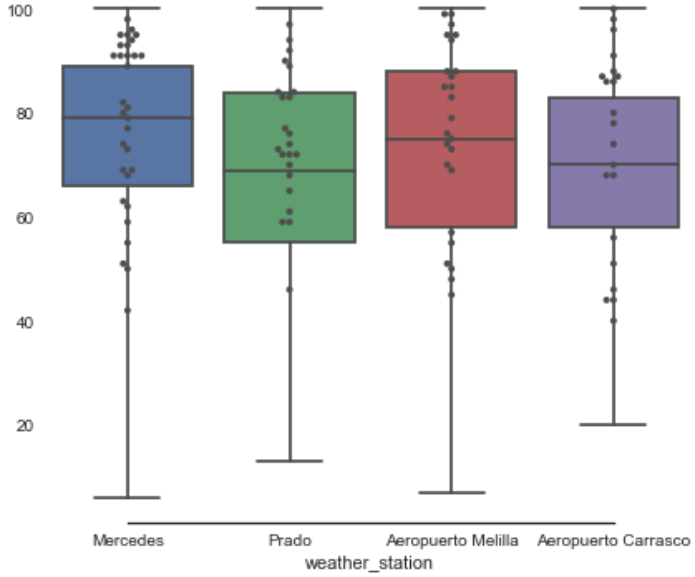


Figura 84 - Diagrama de cajas de la variable humedad

- **Variable presión atmosférica**

En el siguiente gráfico se muestran las frecuencias para valores de la presión atmosférica, es decir, que cantidad de lecturas cayeron en determinados rangos de presión definidos.

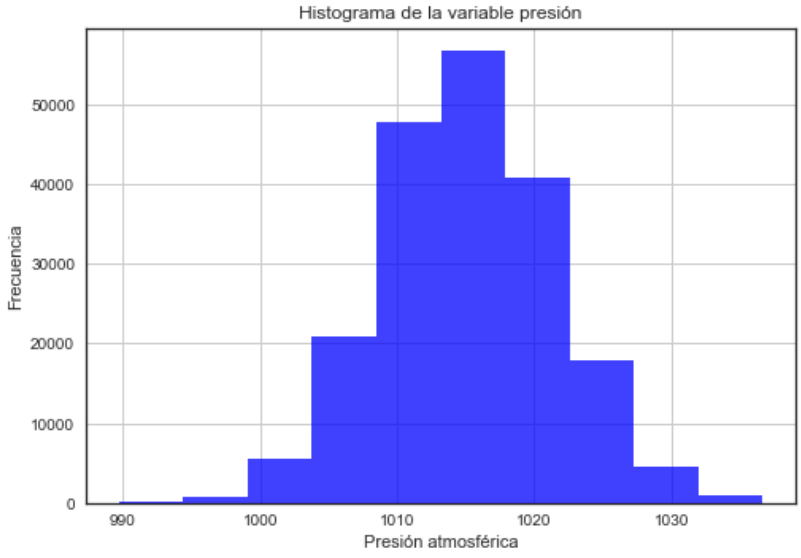


Figura 85 - Histograma de la variable presión

Presión atmosférica (hPa)			
Desde	Hasta	Frecuencia	Porcentaje
990	994.396	68	0.03%
994.395	999.092	793	0.40%
999.093	1003.788	5524	2.82%
1003.789	1008.484	20970	10.71%
1008.485	1013.18	47724	24.37%
1013.19	1017.876	56710	28.96%
1017.877	1022.572	40698	20.78%
1022.573	1027.268	17868	9.12%
1027.269	1031.964	4581	2.34%
1031.965	1036.66	903	0.46%

A continuación, se muestra el gráfico de cajas correspondiente a la presión atmosférica donde puede observarse que la dispersión de los datos es simétrica y en el diagrama anterior la distribución es normal.

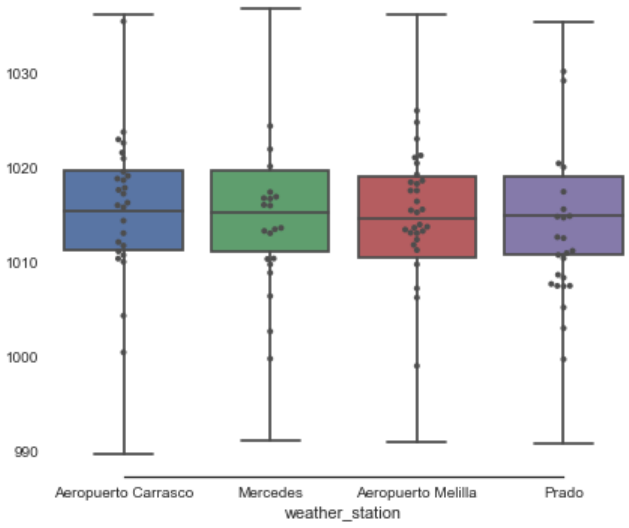


Figura 86 - Diagrama de cajas de la variable presión

- **Variable temperatura**

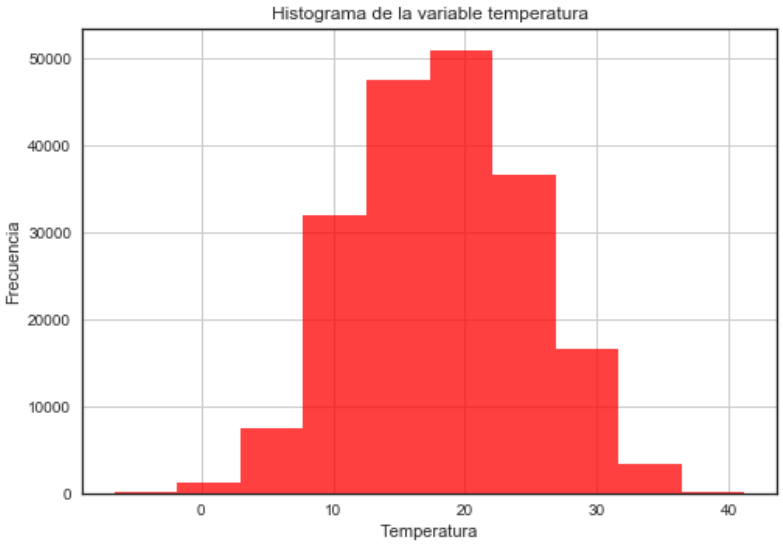


Figura 87 - Histograma de la variable temperatura

Temperatura (C°)			
Desde	Hasta	Frecuencia	Porcentaje
-10	-1.82	64	0.03%
-1.83	2.96	1171	0.60%
2.97	7.74	7385	3.77%
7.75	12.52	32048	16.36%
12.53	17.3	47539	24.27%
17.4	22.08	50911	26.00%
22.09	26.86	36581	18.68%
26.87	31.64	16637	8.50%
31.65	36.42	3308	1.69%
36.43	41.2	195	0.10%

De la misma forma que la presión atmosférica, los datos de temperatura se distribuyen simétricamente y el gráfico anterior muestra que sigue una distribución normal.

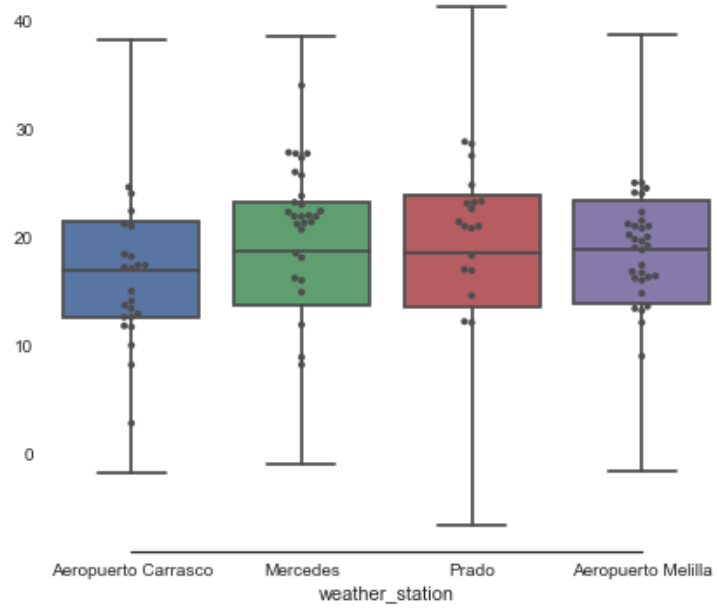


Figura 88 - Diagrama de cajas de la variable presión

También se analizaron las variables de a pares graficándolas con la categorización (*RAIN, NO RAIN*) para ver si se puede inferir alguna relación entre los valores de las mismas con el registro de precipitaciones. Con este objetivo se utilizó el tipo de gráfico *scatter plots*.

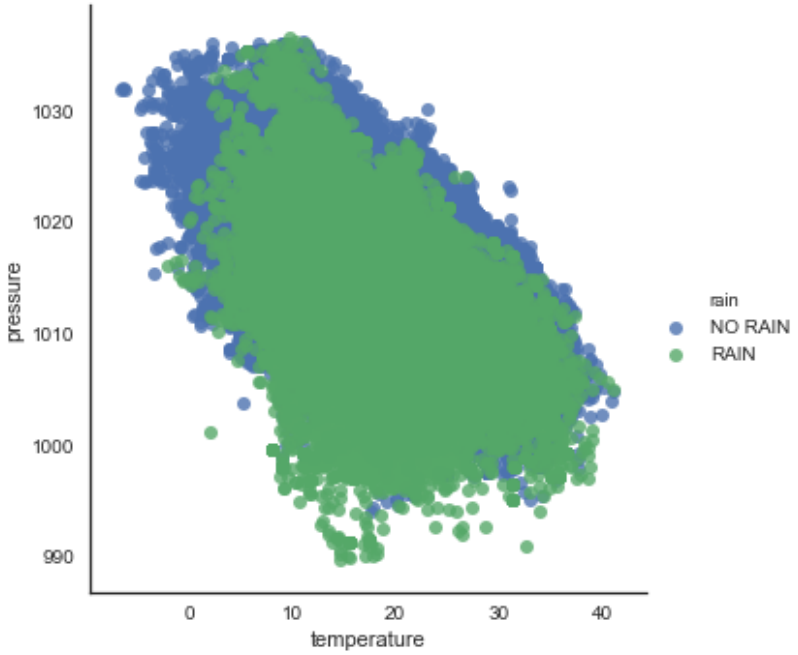


Figura 89 - Diagrama *scatter plot* temperatura presión

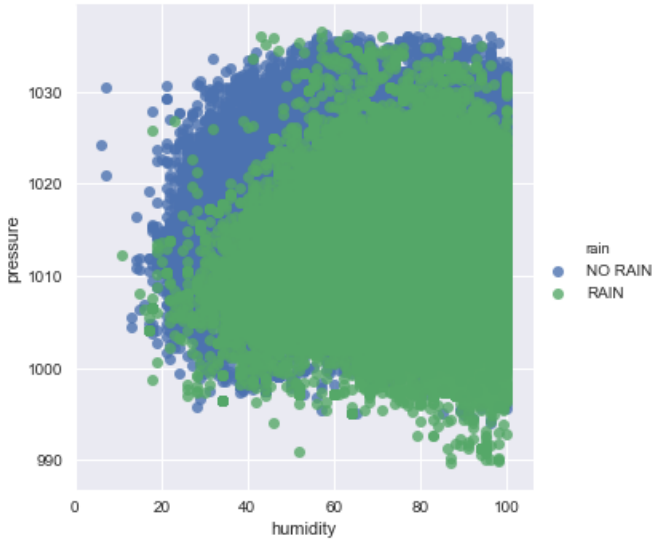


Figura 90 - Diagrama *scatter plot* humedad presión

De estos gráficos no se obtienen demasiadas conclusiones, pero algo que por la naturaleza del *dataset* puede estar afectando el resultado, es que la medición de precipitación es la acumulada del día y a cada hora de ese día se le asigna ese mismo valor.

Y por tanto valores que no representan necesariamente la posibilidad de lluvia se asignan como tal porque a una hora del día llovió.

En el siguiente gráfico se grafican las variables combinadas dos a dos (*scatter plots*) y en la diagonal se muestra la distribución de los valores en referencia al valor de la categoría (*RAIN, NO RAIN*). En dichos casos se puede ver que las distribuciones coinciden con la distribución total de la variable (normal).

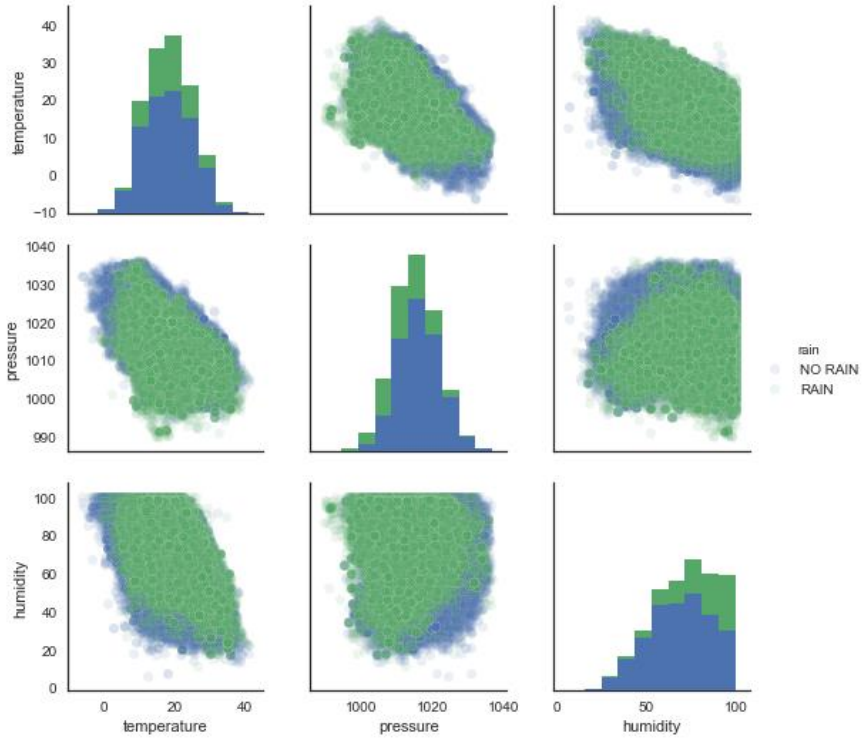


Figura 91 - Diagrama *scatter plots* de todas las variables

Otros gráficos que sí resultaron de utilidad y que en parte confirman cierto conocimiento que se tiene de las variables meteorológicas son los siguientes:

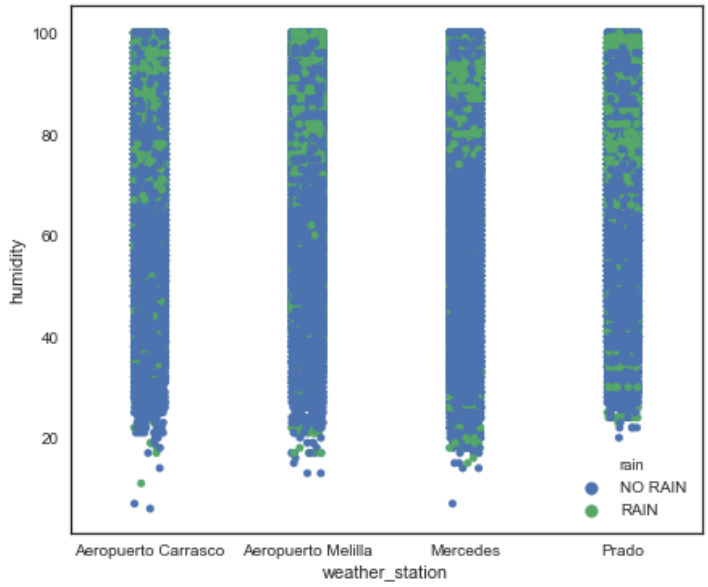


Figura 92 - Diagrama *scatter plots* humedad por estación meteorológica

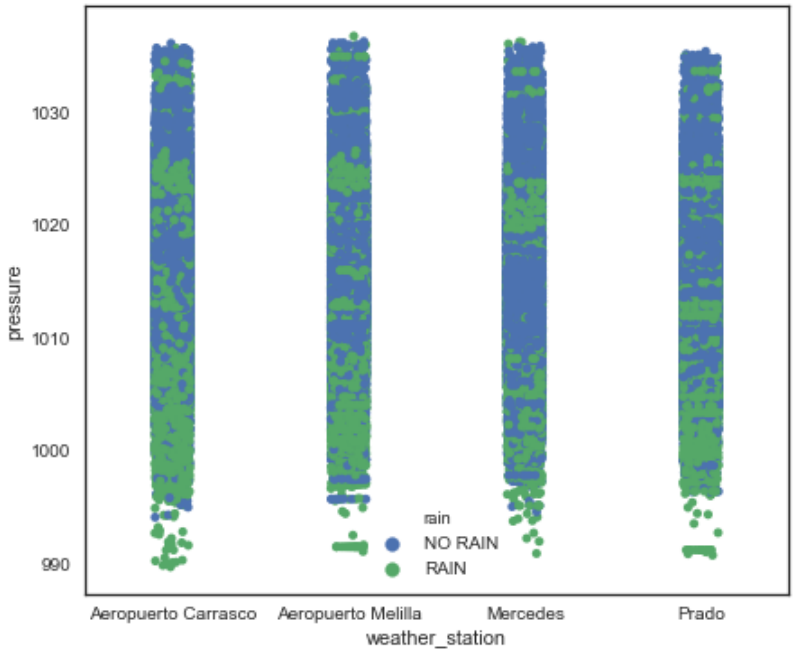


Figura 93 - Diagrama *scatter plots* presión por estación meteorológica

El gráfico no muestra esta relación tan claramente pero como se mencionó en las representaciones anteriores, la precipitación acumulada del día se le asigna a los registros de todas las horas de ese día. Si hay una variación importante de presión al dejar de llover se podría estar catalogando RAIN a una condición de alta presión dentro de las horas siguientes al fenómeno.

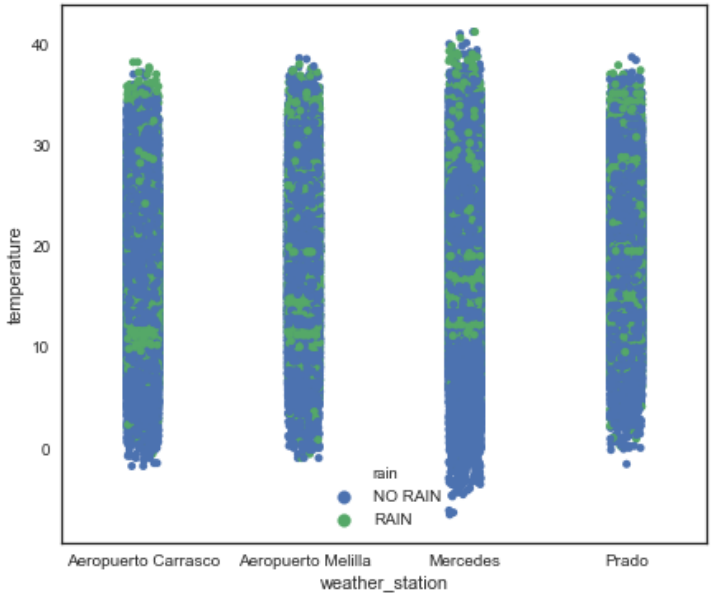


Figura 94 - Diagrama *scatter plots* temperatura por estación meteorológica

En el caso de la temperatura no hay una relación evidente.

En el siguiente gráfico de barras se muestra la cantidad de registros desglosados por estación meteorológica para la categorización (*RAIN, NO RAIN*). A primera vista se puede ver que llovió de forma similar en las estaciones tomadas para este análisis.

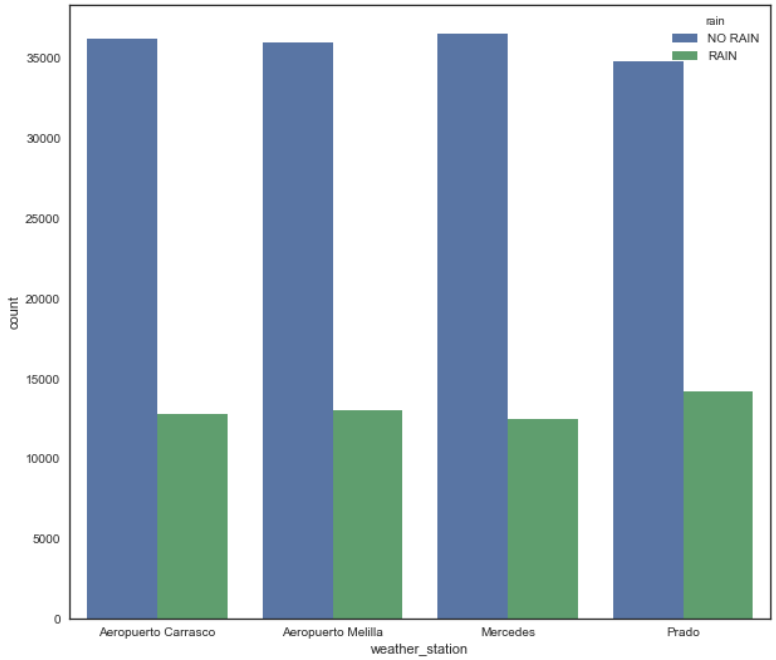


Figura 95 - Diagrama de cantidad registros *RAIN, NO RAIN* por estación

En el gráfico a continuación se puede ver que en las zonas de menor presión y mayor humedad se ubican más puntos rojos que es donde se produjeron precipitaciones, por el contrario, en las zonas de menor humedad y altas presiones se encuentran puntos azules que corresponden a un valor de precipitaciones cero (la intensidad del color no simboliza nada particular).

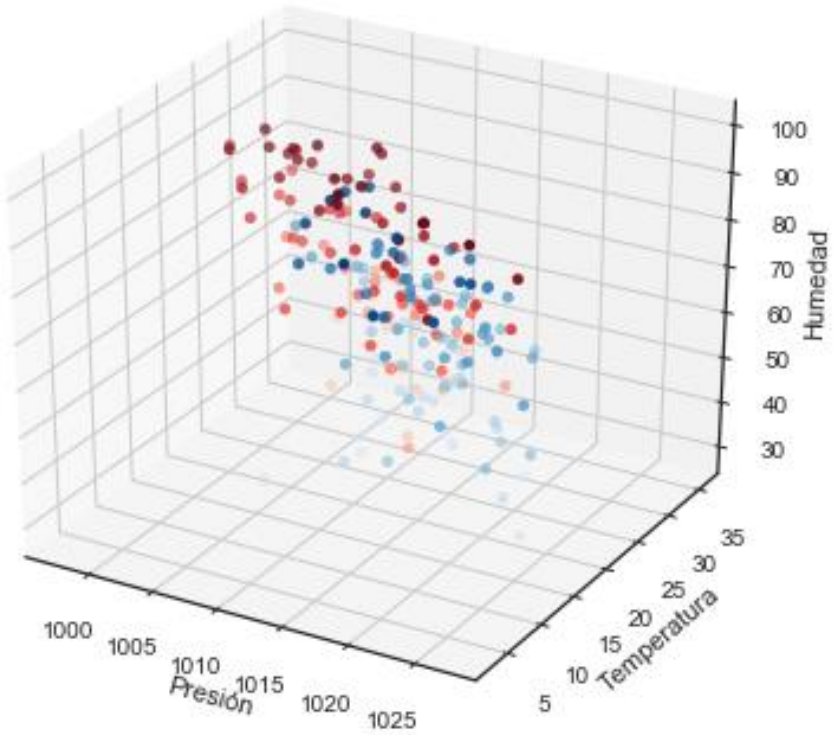


Figura 96 - Diagrama *scatter* 3D de temperatura, humedad y presión

Más allá de ciertas relaciones que se pudieron inferir de los datos, como conclusión del análisis cuantitativo pudimos identificar una mejora en la estructura del *dataset*

Nuevo formato propuesto:

Campo	Descripción	Tipo de dato
Estación	Estación meteorológica en la que se realizó la medición	Texto
Latitud	Latitud de la ubicación geográfica de la estación meteorológica	Decimal
Longitud	Longitud de la ubicación geográfica de la estación meteorológica	Decimal
Fecha	Fecha de la medición	Entero
Humedad0	Medición de la humedad en % a la hora 0	Entero
Humedad1	Medición de la humedad en % a la hora 1	Entero
Humedad2	Medición de la humedad en % a la hora 2	Entero
Humedad3	Medición de la humedad en % a la hora X	Entero
Humedad4	Medición de la humedad en % a la hora 4	Entero
Humedad5	Medición de la humedad en % a la hora 5	Entero
Humedad6	Medición de la humedad en % a la hora 6	Entero
Humedad7	Medición de la humedad en % a la hora 7	Entero
Humedad8	Medición de la humedad en % a la hora 8	Entero
Humedad9	Medición de la humedad en % a la hora 9	Entero
Humedad10	Medición de la humedad en % a la hora 10	Entero
Humedad11	Medición de la humedad en % a la hora 11	Entero
Humedad12	Medición de la humedad en % a la hora 12	Entero

Humedad13	Medición de la humedad en % a la hora 13	Entero
Humedad14	Medición de la humedad en % a la hora 14	Entero
Humedad15	Medición de la humedad en % a la hora 15	Entero
Humedad16	Medición de la humedad en % a la hora 16	Entero
Humedad17	Medición de la humedad en % a la hora 17	Entero
Humedad18	Medición de la humedad en % a la hora 18	Entero
Humedad19	Medición de la humedad en % a la hora 19	Entero
Humedad20	Medición de la humedad en % a la hora 20	Entero
Humedad21	Medición de la humedad en % a la hora 21	Entero
Humedad22	Medición de la humedad en % a la hora 22	Entero
Humedad23	Medición de la humedad en % a la hora 23	Entero
Presion0	Medición de la presión en hPa a la hora 0	Decimal
Presion1	Medición de la presión en hPa a la hora 1	Decimal
Presion2	Medición de la presión en hPa a la hora 2	Decimal
Presion3	Medición de la presión en hPa a la hora 3	Decimal
Presion4	Medición de la presión en hPa a la hora 4	Decimal
Presion5	Medición de la presión en hPa a la hora 5	Decimal
Presion6	Medición de la presión en hPa a la hora 6	Decimal
Presion7	Medición de la presión en hPa a la hora 7	Decimal
Presion8	Medición de la presión en hPa a la hora 8	Decimal
Presion9	Medición de la presión en hPa a la hora 9	Decimal
Presion10	Medición de la presión en hPa a la hora 10	Decimal
Presion11	Medición de la presión en hPa a la hora 11	Decimal
Presion12	Medición de la presión en hPa a la hora 12	Decimal
Presion13	Medición de la presión en hPa a la hora 13	Decimal
Presion14	Medición de la presión en hPa a la hora 14	Decimal
Presion15	Medición de la presión en hPa a la hora 15	Decimal
Presion16	Medición de la presión en hPa a la hora 16	Decimal
Presion17	Medición de la presión en hPa a la hora 17	Decimal
Presion18	Medición de la presión en hPa a la hora 18	Decimal
Presion19	Medición de la presión en hPa a la hora 19	Decimal
Presion20	Medición de la presión en hPa a la hora 20	Decimal
Presion21	Medición de la presión en hPa a la hora 21	Decimal
Presion22	Medición de la presión en hPa a la hora 22	Decimal
Presion23	Medición de la presión en hPa a la hora 23	Decimal
Temperatura0	Medición de la temperatura en C° a la hora 0	Decimal
Temperatura1	Medición de la temperatura en C° a la hora 1	Decimal
Temperatura2	Medición de la temperatura en C° a la hora 2	Decimal
Temperatura3	Medición de la temperatura en C° a la hora 3	Decimal
Temperatura4	Medición de la temperatura en C° a la hora 4	Decimal
Temperatura5	Medición de la temperatura en C° a la hora 5	Decimal
Temperatura6	Medición de la temperatura en C° a la hora 6	Decimal
Temperatura7	Medición de la temperatura en C° a la hora 7	Decimal
Temperatura8	Medición de la temperatura en C° a la hora 8	Decimal
Temperatura9	Medición de la temperatura en C° a la hora 9	Decimal
Temperatura10	Medición de la temperatura en C° a la hora 10	Decimal
Temperatura11	Medición de la temperatura en C° a la hora 11	Decimal
Temperatura12	Medición de la temperatura en C° a la hora 12	Decimal
Temperatura13	Medición de la temperatura en C° a la hora 13	Decimal
Temperatura14	Medición de la temperatura en C° a la hora 14	Decimal
Temperatura15	Medición de la temperatura en C° a la hora 15	Decimal
Temperatura16	Medición de la temperatura en C° a la hora 16	Decimal
Temperatura17	Medición de la temperatura en C° a la hora 17	Decimal
Temperatura18	Medición de la temperatura en C° a la hora 18	Decimal
Temperatura19	Medición de la temperatura en C° a la hora 19	Decimal
Temperatura20	Medición de la temperatura en C° a la hora 20	Decimal
Temperatura21	Medición de la temperatura en C° a la hora 21	Decimal
Temperatura22	Medición de la temperatura en C° a la hora 22	Decimal
Temperatura23	Medición de la temperatura en C° a la hora 23	Decimal
Precipitaciones	Precipitaciones acumuladas en el día	Decimal

Categoría precipitaciones	Categorización que asigna el valor RAIN si Precipitaciones es mayor a cero y NO RAIN en caso contrario	Texto (RAIN, NO RAIN)

Luego de aplicar las transformaciones sugeridas generamos un nuevo *dataset*:

Archivo	Descripción	Tamaño	Cantidad de registros
Inumet-Datos2.xlsx	Lecturas por día de las variables temperatura, presión, humedad (se registra el valor en cada hora del día) y precipitaciones del día en las estaciones meteorológicas “Aeropuerto Carrasco”, “Aeropuerto Melilla”, “Prado” y “Mercedes” en los últimos 5 años.	2840 kb	8160

La justificación de la propuesta de una nueva estructura es para tener los datos de las variables a cada hora del día y que algunos de los métodos utilizados pudieran correlacionar dichos valores con el registro de precipitación diario.

Adicionalmente a los dos *datasets* definidos se identificó otra estructura para realizar pruebas con los modelos predictivos:

Archivo	Descripción	Tamaño	Cantidad de registros
Inumet-Datos3.csv	Lecturas por día de las variables temperatura media, presión media, humedad media (se registra el valor en cada hora del día) y precipitaciones del día en las estaciones meteorológicas “Aeropuerto Carrasco”, “Aeropuerto Melilla”, “Prado” y “Mercedes” en los últimos 5 años.	2840 kb	8160

Este nuevo enfoque busca consolidar las lecturas diarias de cada variable en su media aritmética ya que algunos modelos no pueden correlacionar la información por hora del *dataset* 2. A futuro se podría incluir la desviación estándar, la máxima diferencia diaria para cada variable, la mediana diaria, etc.

Campo	Descripción	Tipo de dato
Estación	Estación meteorológica en la que se realizó la medición	Texto
Latitud	Latitud de la ubicación geográfica de la estación meteorológica	Decimal
Longitud	Longitud de la ubicación geográfica de la estación meteorológica	Decimal
Fecha	Fecha de la medición	Entero
Humedad media	Medición de la humedad en % a la hora 0	Entero
Presion media	Medición de la presión en hPa a la hora 0	Decimal
Temperatura media	Medición de la temperatura en C° a la hora 0	Decimal
Precipitaciones	Precipitaciones acumuladas en el día	Decimal
Categoría precipitaciones	Categorización que asigna el valor RAIN si Precipitaciones es mayor a cero y NO RAIN en caso contrario	Texto (RAIN, NO RAIN)

5.2.4 Separación del *dataset* en *train*, *test* y *validation*

Para la aplicación de los modelos se tomaron los *datasets* y se dividieron en los subconjuntos *train*, *test* y *validation* respetando la siguiente proporción (60% *train*, 20% *test* y 20% *validation*)

La idea es entrenar el modelo de aprendizaje supervisado mediante el conjunto de *train*, luego evaluar la eficiencia mediante el de test y finalmente evaluar el desempeño con un conjunto totalmente distinto que fue separado al comienzo del análisis (*validation*).

La selección de los datos se hace de forma aleatoria.

5.2.5 Construcción de los modelos predictivos

En un comienzo se realizaron capacitaciones online y tutoriales básicos de Azure *Machine Learning*, puntualmente se realizó el mismo ejemplo desarrollado con la herramienta IBM DSX para poder comparar ambas herramientas y verificar si la elegida cumplía con nuestra expectativa.

Adicionalmente se comenzó a explorar los datos provistos por INUMET, a analizar y definir una primera versión del modelo predictivo. Para dicha tarea nos fue muy útil asistir a la electiva “*Machine Learning* para sistemas

inteligentes” ya que nos permitió manejar conceptos teóricos y herramientas para poder encarar dicha tarea.

Los enfoques que definimos para la predicción de las precipitaciones en función de temperatura, humedad y presión fueron dos:

- Predecir un conjunto de estados discretos o categóricos (llueve, no llueve) lo que implica resolver un **problema de clasificación**.
- Predecir un valor concreto (numérico) de la variable precipitación, en este caso estaríamos frente a **un problema de regresión**.

5.2.5.1 Construcción modelo predictivo discreto

- **Primer modelo (*baseline*)**

  En la definición del *baseline* para comenzar con el análisis de las diferentes técnicas, se optó por usar árboles de clasificación y aprendizaje bayesiano naïve. En el segundo se decidió aplicar la variante gaussiana debido a la distribución de las variables que se pudo identificar en el análisis cuantitativo del *dataset*. Cabe aclarar que los algoritmos se ejecutaron asignando a los meta parámetros valores por defecto.

  A continuación, se muestran los resultados obtenidos:

  ***Dataset* 1**

```
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
DecisionTreeClassifier(class_weight=None,             criterion='gini',
max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=1, splitter='best')
Model generated
Evaluating model accuracy and confusion matrix
0.88858251634
[[26495  2217]
 [ 2147  8309]]
             precision    recall  f1-score   support
```

```
    NO RAIN       0.93      0.92      0.92     28712
       RAIN       0.79      0.79      0.79     10456


avg / total       0.89      0.89      0.89     39168

GaussianNB(priors=None)
Model generated
Evaluating model accuracy and confusion matrix
0.764476102941
[[26980  1732]
 [ 7493  2963]]
             precision    recall  f1-score   support


    NO RAIN       0.78      0.94      0.85     28712
       RAIN       0.63      0.28      0.39     10456


avg / total       0.74      0.76      0.73     39168
       ##############################################################
```

Dataset 2

```
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
DecisionTreeClassifier(class_weight=None,             criterion='gini',
max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=1, splitter='best')
Model generated
Evaluating model accuracy and confusion matrix
0.765318627451
[[981 198]
 [185 268]]
             precision    recall  f1-score   support


    NO RAIN       0.84      0.83      0.84      1179
       RAIN       0.58      0.59      0.58       453


avg / total       0.77      0.77      0.77      1632

GaussianNB(priors=None)
Model generated
Evaluating model accuracy and confusion matrix
0.770833333333
[[928 251]
 [123 330]]
             precision    recall  f1-score   support
```

```
    NO RAIN       0.88      0.79      0.83      1179
       RAIN       0.57      0.73      0.64       453


avg / total       0.80      0.77      0.78      1632
##############################################################
```

Para realizar la evaluación cualitativa y cuantitativa de los métodos aplicados y dado que son algoritmos de clasificación vamos a evaluar las métricas exactitud, precisión y *recall*.

Matriz de Confusión:
(Clasificación binaria)

tp: true positives
tn: true negatives
fp: false positives
fn: false negatives

	Positivo (predicho)	Negativo (predicho)
Positivo (real)	tp	fn
Negativo (real)	fp	tn

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Figura 97 - Matriz de confusión

A modo de resumen la exactitud (*accuracy*) indica el porcentaje de elementos que fueron clasificados correctamente por el algoritmo (tomando en cuenta ambas clases).

La precisión indica que porcentaje de los elementos que fueron clasificados en una clase dada, efectivamente pertenecen a esa clase.

Por otra parte, el *recall* indica que porcentaje de los elementos de una clase dada fueron clasificados correctamente por el algoritmo.

	DecisionTreeClassifier			GaussianNB		
	Accuracy	***Precision (RAIN)***	***Recall (RAIN)***	***Accuracy (RAIN)***	***Precision (RAIN)***	***Recall (RAIN)***
Dataset 1	0.88	0.79	0.79	0.76	0.63	0.28
Dataset 2	0.76	0.58	0.59	0.77	0.57	0.73

Basados en las métricas descriptas anteriormente, tomamos el *recall* de la categoría "*RAIN*" como fundamental para decidir sobre algún algoritmo en relación a su desempeño, ya que dicho valor indica que tan bien clasifica como "*RAIN*" los días que efectivamente llovió.

Viendo los resultados podemos concluir que el árbol de clasificación se desempeña mejor con el *dataset* 1 pero que la GaussianNB obtiene buenos resultados sobre el segundo *dataset*. Como posibles puntos de mejora se propone ejecutar dichos algoritmos modificando los meta parámetros y ver si eso genera una mejora en los resultados.

En el árbol de clasificación generado se puede ver que está tomando como primeros atributos la presión y la humedad para realizar las particiones.

Se puede concluir que son atributos que influyen fuertemente en la predicción de las precipitaciones.

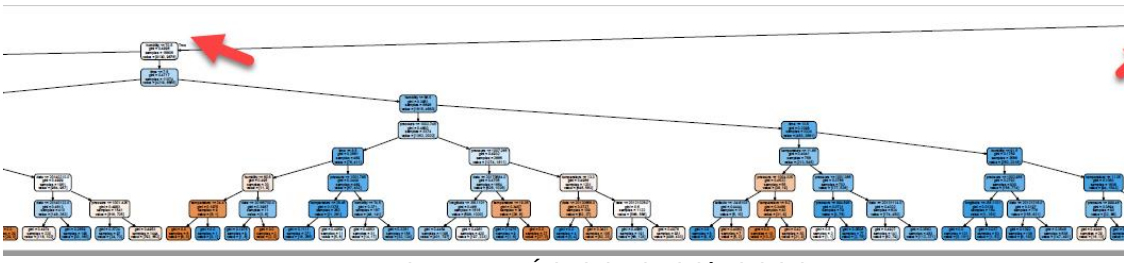


Figura 98 - Árbol de decisión inicial

Si bien el objetivo era analizar el árbol utilizando los parámetros por defecto, igual tuvimos que modificar para que el procesamiento no demorara mucho tiempo para generar el pdf.

- **Ingeniería de atributos**

  En el *dataset* 2 se podría calcular "media" y "dev std" diarias de las variables predictoras y descartarlas de la secuencia diaria (no se realizaron pruebas de lo mencionado).

  Se aplicó PDA al segundo *dataset* para identificar las variables que influyen más en la predicción con el objetivo de reducir la cantidad.

```
pca = PCA()
pca.fit(df_weather[df_weather.columns[list(range(1,75))]])
print(pca.components_)
print(pca.explained_variance_ratio_)
```

```
[[ -1.54398231e-11   1.84230532e-11   9.99999915e-01 ...,   2.49243647e-05
    1.11642027e-06   5.20352404e-05]
 [  2.42595705e-03  -2.94267070e-03   2.03527166e-04 ...,   1.87762341e-02
    2.02303461e-02  -6.58796919e-03]
 [  5.02398132e-03  -5.38994487e-03   8.19532515e-05 ...,  -9.29700097e-02
   -5.23596943e-02  -7.50808124e-02]
 ...,
 [  1.05592925e-01  -1.29706768e-01  -1.67457155e-06 ...,  -1.04272388e-03
    5.21449557e-03  -1.90870277e-02]
 [  2.77567012e-02  -3.21459083e-02  -3.63994142e-07 ...,  -1.88547705e-03
    3.53758067e-03  -1.81661994e-02]
 [  7.81231233e-01   6.24048317e-01  -1.19399820e-07 ...,  -1.82407709e-03
    2.25731400e-03  -2.26705050e-03]]
[  9.99975230e-01   1.02489114e-05   3.00948641e-06 ...,   2.09378143e-10
   1.76531238e-10   8.28633356e-13]
```

- **Técnicas aplicadas**

  Luego de construir el *baseline* de referencia, se realizaron pruebas con otros algoritmos y generamos de forma iterativa nuevos modelos intentando obtener un mejor desempeño en las predicciones.

  Decision Tree Classifier

  Se aplicó el modelo definido en el *baseline* pero cambiando los meta parámetros y posteriormente se verificó el resultado

En este caso se cambió la función utilizada como criterio para evaluar la calidad de una partición de gini a *entropy*.

Como puede verse incrementa en 1% el *recall*.

```
Getting Data from file...
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
DecisionTreeClassifier(class_weight=None,           criterion='entropy',
max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=1, splitter='best')
Model generated
Evaluating model accuracy and confusion matrix
0.895450367647
[[26689  2023]
 [ 2072  8384]]
             precision    recall  f1-score   support


    NO RAIN       0.93      0.93      0.93     28712
       RAIN       0.81      0.80      0.80     10456


avg / total       0.90      0.90      0.90     39168
##############################################################
```

Se modificó el parámetro *presort* pero no produjo cambios sustanciales en el resultado de la predicción.

```
Getting Data from file...
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
DecisionTreeClassifier(class_weight=None,           criterion='entropy',
max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=True, random_state=1, splitter='best')
Model generated
Evaluating model accuracy and confusion matrix
0.895450367647
[[26689  2023]
 [ 2072  8384]]
             precision    recall  f1-score   support

    NO RAIN       0.93      0.93      0.93     28712
       RAIN       0.81      0.80      0.80     10456
```

```
avg / total       0.90      0.90      0.90     39168
#############################################################
```

Random Forest

```
Getting Data from file...
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
RandomForestClassifier(bootstrap=True,               class_weight=None,
criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            n_estimators=10, n_jobs=1, oob_score=False, random_state=1,
            verbose=0, warm_start=False)
Model generated
Evaluating model accuracy and confusion matrix
0.890037785948
[[27895   817]
 [ 3490  6966]]
             precision    recall  f1-score   support

    NO RAIN       0.89      0.97      0.93     28712
       RAIN       0.90      0.67      0.76     10456


avg / total       0.89      0.89      0.88     39168
#############################################################
```

Modificando los parámetros criterion y n_estimators mejora la estimación

```
Getting Data from file...
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
RandomForestClassifier(bootstrap=True,               class_weight=None,
criterion='entropy',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            n_estimators=100,        n_jobs=-1,        oob_score=False,
random_state=1,
            verbose=0, warm_start=False)
Model generated
Evaluating model accuracy and confusion matrix
0.90691380719
[[28036   676]
 [ 2970  7486]]
             precision    recall  f1-score   support

    NO RAIN       0.90      0.98      0.94     28712
       RAIN       0.92      0.72      0.80     10456


avg / total       0.91      0.91      0.90     39168
#############################################################
```

Adaptive Boosting

```
Getting Data from file...
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
          learning_rate=1.0, n_estimators=50, random_state=None)
Model generated
Evaluating model accuracy and confusion matrix
0.777675653595
[[26936  1776]
 [ 6932  3524]]
             precision    recall  f1-score   support


    NO RAIN       0.80      0.94      0.86     28712
       RAIN       0.66      0.34      0.45     10456


avg / total       0.76      0.78      0.75     39168
##############################################################
```

En el siguiente resultado aplicamos el método variando los n_estimators y el random_state.

Si bien la exactitud es de un 0.55, logra clasificar correctamente los días de lluvia en un 82%.

```
Getting Data from file...
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
          learning_rate=2.0, n_estimators=100, random_state=1)
Model generated
Evaluating model accuracy and confusion matrix
0.559410743464
[[13326 15386]
 [ 1871  8585]]
             precision    recall  f1-score   support


    NO RAIN       0.88      0.46      0.61     28712
       RAIN       0.36      0.82      0.50     10456


avg / total       0.74      0.56      0.58     39168
##############################################################
```

Se realizaron pruebas adicionales variando los meta parámetros del algoritmo, pero no se obtuvieron mejores resultados.

Gradient Boosting Classifier

```
Getting Data from file...
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=0.1, loss='deviance', max_depth=3,
              max_features=None, max_leaf_nodes=None,
              min_impurity_split=1e-07, min_samples_leaf=1,
              min_samples_split=2, min_weight_fraction_leaf=0.0,
              n_estimators=100, presort='auto', random_state=None,
              subsample=1.0, verbose=0, warm_start=False)
Model generated
Evaluating model accuracy and confusion matrix
0.801547181373
[[27497  1215]
 [ 6558  3898]]
             precision    recall  f1-score   support


    NO RAIN       0.81      0.96      0.88     28712
       RAIN       0.76      0.37      0.50     10456


avg / total       0.80      0.80      0.78     39168
##############################################################

Getting Data from file...
Data loaded
Exploring the data
Splits the dataset in train (60%), test (20%) and validation (20%)
GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=1.0, loss='deviance', max_depth=3,
              max_features=None, max_leaf_nodes=None,
              min_impurity_split=1e-07, min_samples_leaf=1,
              min_samples_split=2, min_weight_fraction_leaf=0.0,
              n_estimators=500, presort='auto', random_state=None,
              subsample=1.0, verbose=0, warm_start=False)
Model generated
Evaluating model accuracy and confusion matrix
0.901526756536
[[27291  1421]
 [ 2436  8020]]
             precision    recall  f1-score   support

    NO RAIN       0.92      0.95      0.93     28712
       RAIN       0.85      0.77      0.81     10456


avg / total       0.90      0.90      0.90     39168
##############################################################
```

Redes Neuronales

Se comenzaron a realizar pruebas iniciales con redes neuronales básicas con el objetivo de ver si estos modelos predictivos puedan identificar secuencias y otro tipo de relaciones.

Al momento no hemos logrado mejores resultados que con los métodos anteriores, pero es un tema en que debemos seguir profundizando.

Resultados comparativos

Técnica aplicada	Parámetros	Accuracy	Recall (RAIN)
Decision Tree Classifier	Por defecto	0.88	0.79
Gaussian Naive Bayes Classifier	Por defecto	0.76	0.28
Random Forest	Por defecto	0.89	0.67
Random Forest	criterion='entropy' n_estimators=100	0.90	0.72
Adaptive Boosting	Por defecto	0.77	0.34
Adaptive Boosting	n_estimators=100, random_state=1	0.55	0.82
Gradient Boosting Classifier	Por defecto	0.80	0.37
Gradient Boosting Classifier	n_estimators=500 learning_rate=1.0	0.90	0.77
Neural Network	Por defecto		

Luego de aplicar las diferentes técnicas variando los meta parámetros, la predicción de la condición de lluvia se logra predecir de mejor manera utilizando el algoritmo *Adaptive Boosting*.

5.2.5.2 Construcción modelo predictivo continuo

- **Primer modelo (*baseline*)**

  En la definición del *baseline* para el modelo continuo, se optó por usar variantes de árboles de regresión.

  Cabe aclarar que los algoritmos se ejecutaron asignando a los meta parámetros valores por defecto.

  En Azure ML:

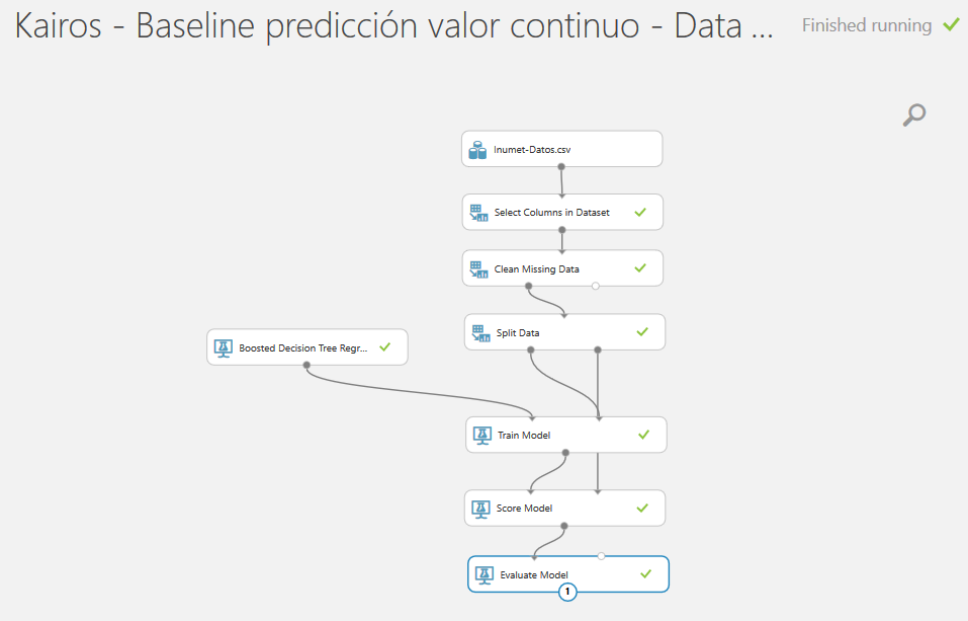


Figura 99 - *Baseline* del modelo continuo

A continuación, se muestran los resultados obtenidos:

Figura 100 - Métricas obtenidas del modelo continuo en Azure ML

En Python:

```
Data loaded
Exploring the data
######################### PCA ###############################
[[-0.98021903 -0.01222081  0.19753812]
 [-0.13622041  0.7657245  -0.62857775]
 [-0.14357805 -0.64305259 -0.75224245]]
Explained variance ratio:  [ 0.82039307  0.14660079  0.03300615]
############################################################
Splits the dataset in train (60%), test (20%) and validation (20%)
#########################################################
```

```
DecisionTreeRegressor(criterion='mse',                    max_depth=None,
max_features=None,
           max_leaf_nodes=None, min_impurity_split=1e-07,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0,                   presort=False,
random_state=None,
           splitter='best')
Model generated
Mean Absolute Error:  4.26969747076
Coefficient of determination:  -0.279972209428
############################################################
Out[302]: 0
```

Analizando el resultado de ambas pruebas se puede constatar que, si bien Azure arrojó un mejor resultado, no es satisfactorio ya que el coeficiente de determinación ML es cercano a uno si el modelo tiene un buen desempeño en la predicción.

De forma análoga se realizaron pruebas con los *datasets* 2 y 3 obteniendo resultados similares.

- **Técnicas aplicadas**

  A continuación, se muestran algunas de las pruebas realizadas para ilustrar el proceso iterativo que es necesario realizar con el objetivo de determinar el modelo que mejor se adapta al tipo de problema a resolver.

  En Azure ML:

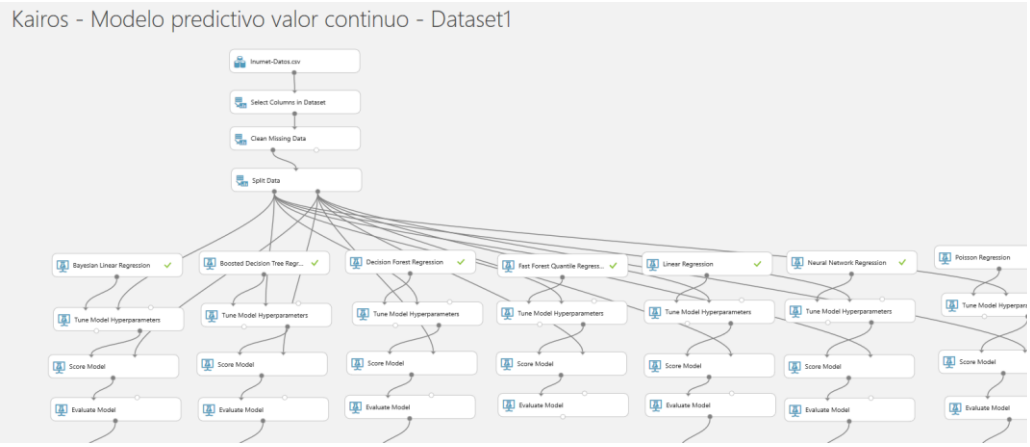


Figura 101 – Modelo continuo de Azure ML

Se aplicaron los siguientes métodos variando los meta parámetros de los mismos:

- Bayesian *Linear Regression*
- *Boosted Decision Tree Regression*
- *Decision Forest Regression*
- *Fast Forest Quantile Regression*
- *Linear Regression*
- *Neural Network Regression*
- *Poisson Regression*

La opción utilizada para variar los meta parámetros fue *entire grid*, que lo que hace es probar con todas las combinaciones de parámetros y seleccionar la que tiene mayor desempeño de acuerdo a una métrica indicada (en nuestro caso coeficiente de determinación).

En Python:

```
Data loaded
Exploring the data
######################### PCA ##############################
[[-0.98021903 -0.01222081  0.19753812]
 [-0.13622041  0.7657245  -0.62857775]
 [-0.14357805 -0.64305259 -0.75224245]]
Explained variance ratio:  [ 0.82039307  0.14660079  0.03300615]
############################################################
Splits the dataset in train (60%), test (20%) and validation (20%)
#########################################################
DecisionTreeRegressor(criterion='mse',                   max_depth=None,
max_features=None,
           max_leaf_nodes=None, min_impurity_split=1e-07,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0,                 presort=False,
random_state=None,
           splitter='best')
Model generated
Mean Absolute Error:  4.28285482279
Coefficient of determination:  -0.296658744474
#########################################################
SGDRegressor(alpha=0.0001, average=False, epsilon=0.1, eta0=0.01,
       fit_intercept=True, l1_ratio=0.15, learning_rate='invscaling',
       loss='squared_loss', n_iter=5, penalty='l2', power_t=0.25,
       random_state=None, shuffle=True, verbose=0, warm_start=False)
Model generated
Mean Absolute Error:  2.52085023667e+14
Coefficient of determination:  -6.00688142963e+26
#########################################################
Ridge(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=None,
   normalize=False, random_state=None, solver='auto', tol=0.001)
Model generated
Mean Absolute Error:  5.10785335556
```

```
Coefficient of determination:  0.0877594151247
#########################################################
BayesianRidge(alpha_1=1e-06,     alpha_2=1e-06,     compute_score=False,
copy_X=True,
       fit_intercept=True, lambda_1=1e-06, lambda_2=1e-06, n_iter=300,
       normalize=False, tol=0.001, verbose=False)
Model generated
Mean Absolute Error:  5.10763824332
Coefficient of determination:  0.0877591251292
#########################################################
Lasso(alpha=0.1, copy_X=True, fit_intercept=True, max_iter=1000,
   normalize=False, positive=False, precompute=False, random_state=None,
   selection='cyclic', tol=0.0001, warm_start=False)
Model generated
Mean Absolute Error:  5.10260250412
Coefficient of determination:  0.0877698632639
#########################################################
ElasticNet(alpha=1.0, copy_X=True, fit_intercept=True, l1_ratio=0.5,
      max_iter=1000, normalize=False, positive=False, precompute=False,
      random_state=None,         selection='cyclic',         tol=0.0001,
warm_start=False)
Model generated
Mean Absolute Error:  5.0785164018
Coefficient of determination:  0.0876704313153
#########################################################
KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='minkowski',
          metric_params=None, n_jobs=1, n_neighbors=2, p=2,
          weights='uniform')
Model generated
Mean Absolute Error:  4.00139782475
Coefficient of determination:  -0.0311345554054
#########################################################
AdaBoostRegressor(base_estimator=None, learning_rate=1.0, loss='linear',
         n_estimators=50, random_state=None)
Model generated
Mean Absolute Error:  6.90952577913
Coefficient of determination:  -0.159950932642
#########################################################
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_split=1e-07, min_samples_leaf=1,
           min_samples_split=2, min_weight_fraction_leaf=0.0,
           n_estimators=10,          n_jobs=1,          oob_score=False,
random_state=None,
           verbose=0, warm_start=False)
Model generated
Mean Absolute Error:  3.97787717817
Coefficient of determination:  0.181453213167
#########################################################
GradientBoostingRegressor(alpha=0.9,           criterion='friedman_mse',
init=None,
             learning_rate=0.1,         loss='ls',         max_depth=3,
max_features=None,
             max_leaf_nodes=None, min_impurity_split=1e-07,
             min_samples_leaf=1, min_samples_split=2,
             min_weight_fraction_leaf=0.0, n_estimators=100,
             presort='auto',      random_state=None,      subsample=1.0,
verbose=0,
             warm_start=False)
Model generated
Mean Absolute Error:  4.52853375482
Coefficient of determination:  0.143558638027
#########################################################
```

```
MLPRegressor(activation='relu',     alpha=0.0001,     batch_size='auto',
beta_1=0.9,
       beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=(100,), learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       nesterovs_momentum=True, power_t=0.5, random_state=None,
       shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1,
       verbose=False, warm_start=False)
Model generated
Mean Absolute Error:  6.19740078874
Coefficient of determination:  0.0500079390252
############################################################
```

También se realizaron pruebas adicionales cambiando los meta parámetros, pero hubo cambio importante en los resultados.

- **Resultados comparativos**

Técnica aplicada	Parámetros	Mean Absolute Error	Coefficient of determination
DecisionTreeRegressor	Por defecto	4.28	-0.29
SGDRegressor	Por defecto	2.52	-6.00
Ridge	Por defecto	5.10	0.08
BayesianRidge	Por defecto	5.10	0.08
Lasso	Por defecto	5.10	0.08
ElasticNet	Por defecto	5.07	0.08
KNeighborsRegressor	Por defecto	4.00	-0.03
RandomForestRegressor	Por defecto	3.97	0.18
GradientBoostingRegressor	Por defecto	4.52	0.14
MLPRegressor	Por defecto	6.19	0.05
Bayesian Linear Regression	Meta parámetros óptimos	5.26	0.06
Boosted Decision Tree Regression	Meta parámetros óptimos	3.57	0.43
Linear Regression Neural Network	Meta parámetros óptimos	5.26	0.06
Neural Network Regression	Meta parámetros óptimos	5.21	0.09
Regression Poisson Regression	Meta parámetros óptimos	4.77	0.09

Para obtener mayor información sobre los modelos de *Machine Learning* que con los que se trabajó, consultar el ANEXO 3 - Modelos de *Machine Learning* utilizados

6. Conclusiones

Es esta sección presentaremos las conclusiones sobre el desarrollo del proyecto, así como también serán mencionados posibles pasos futuros.

Como proyecto de investigación sobre la factibilidad del uso de *Machine Learning* en la predicción del clima, el resultado obtenido fue satisfactorio, cumplió con el alcance y se pudo confirmar que es viable si bien se identificaron oportunidades de mejora en la parte analítica. Se logró predecir tanto en forma discreta (lluvia o no lluvia) como en forma continua (valor), las precipitaciones en función de las variables meteorológicas temperatura, humedad y presión atmosférica.

Se logró formar una buena base de datos con una buena cantidad de registros históricos y actuales para un número considerable de variables meteorológicas, independientemente de que se hayan utilizado solo las cuatro mencionadas para esta tesis. Los resultados obtenidos son mostrados en *Power BI* de modo de consolidar tanto la lectura de la PWS como la invocación al modelo y su resultado predictivo. De todos modos, pretendemos aclarar que aún no consideramos estos resultados como algo realmente aplicable en el corto plazo, sino que deben ser mejorados y utilizados en conjunto con los métodos ya disponibles para la predicción meteorológica.

En cuanto a la experiencia en el tratamiento de datos (depuración), herramientas y lenguajes de programación utilizados, así como en técnicas de aprendizaje automático, el equipo no poseía experiencia alguna, lo cual demandó un mayor esfuerzo por nuestra parte, al mismo tiempo que representó un gran desafío.

Algunos de los problemas encontrados fueron:

- Limitante a la hora de trabajar en la plataforma Bluemix, ya que el tipo de cuenta a la que tuvimos acceso en una primera instancia, fue una cuenta gratuita con 30 días de prueba. Se gestiona posteriormente la obtención de cuentas con dominio @fi365.ort.edu.uy (educativo) para acceder a los beneficios que ofrece IBM para tales fines, accediendo a una cuenta gratuita por 6 meses.

- Entramos en contacto con INUMET para obtener la información histórica solicitada y acordamos la forma en la que nos deberían proporcionar la misma (medio electrónico), se analizaron los datos de modo de esclarecer a que corresponde al concepto “Sur del país” y seleccionaron las localidades que son interés para nuestro proyecto. Se presentó una carta redactada y firmada por alumnos y tutor con el formato indicado en el sitio del Instituto Nacional de Meteorología para obtener dicha información en forma gratuita justificando fines académicos. INUMET si bien nos brindó la información que solicitamos, dio prioridad a las solicitudes pagas y obtuvimos los datos luego de un mes de iniciada la solicitud.

- En el proceso de obtención de datos nos encontramos con un problema al momento de construir el modelo predictivo; la granularidad de los datos obtenidos. Los datos proporcionados por IBM con información histórica de algunos puntos del país estaban en forma diaria mientras que la obtenida mediante las APIs es a razón de dos llamadas por hora, o sea aproximadamente unos 48 datos climáticos por localidad por día, mientras que la proporcionada por INUMET es por hora. Fue necesario uniformizar los datos provenientes de las distintas fuentes mencionadas.

- En cuanto a los componentes de la PWS, se hizo un estudio/consulta en el mercado sobre la facilidad de obtención de los componentes en Montevideo en distintas casas de electrónica y nos encontramos con que no disponían de estos componentes siendo necesario importarlos con una demora de 30 días. Una vez que tuvimos el aval de SinergiaTech procedimos a encargar alguno de ellos directamente con el fabricante para bajar esa demora.

- La antena del módulo de GPS de la estación meteorológica no es muy potente, por tanto, en el interior de un edificio o casa suele tardar o incluso no capturar señal de los satélites necesarios para triangular. Este problema se resolvió al exponer a la intemperie la antena por unos

minutos hasta que logre localizar una cantidad de satélites mínima necesaria para geo posicionar y brindar la latitud y longitud. La PWS capturó datos de los sensores de humedad, temperatura y presión atmosférica en un comienzo y posteriormente se anexó el componente *Weather Meters* de Sparkfun que permitió recoger datos de anemómetro y pluviómetro. El fabricante estuvo sin stock durante cuatro meses y estuvimos aguardando su liberación lo cual retrasó el avance en su construcción.

Como pasos futuros podemos mencionar:

- En cuanto a la analítica se podrá seguir trabajando para obtener resultados más precisos, buscando mejores modelos y disminuyendo el error en cada uno de ellos. Se podrá combinar y trabajar los datos de múltiples formas, ya que los métodos no tienen una única forma de recibirlos. Se podrá también extender el estudio a todo el territorio nacional.

- En cuanto a la PWS, otra posible solución es registrar la estación en sitios como Wunderground y luego consumir la información mediante API para la ubicación específica o ID de estación que registramos.

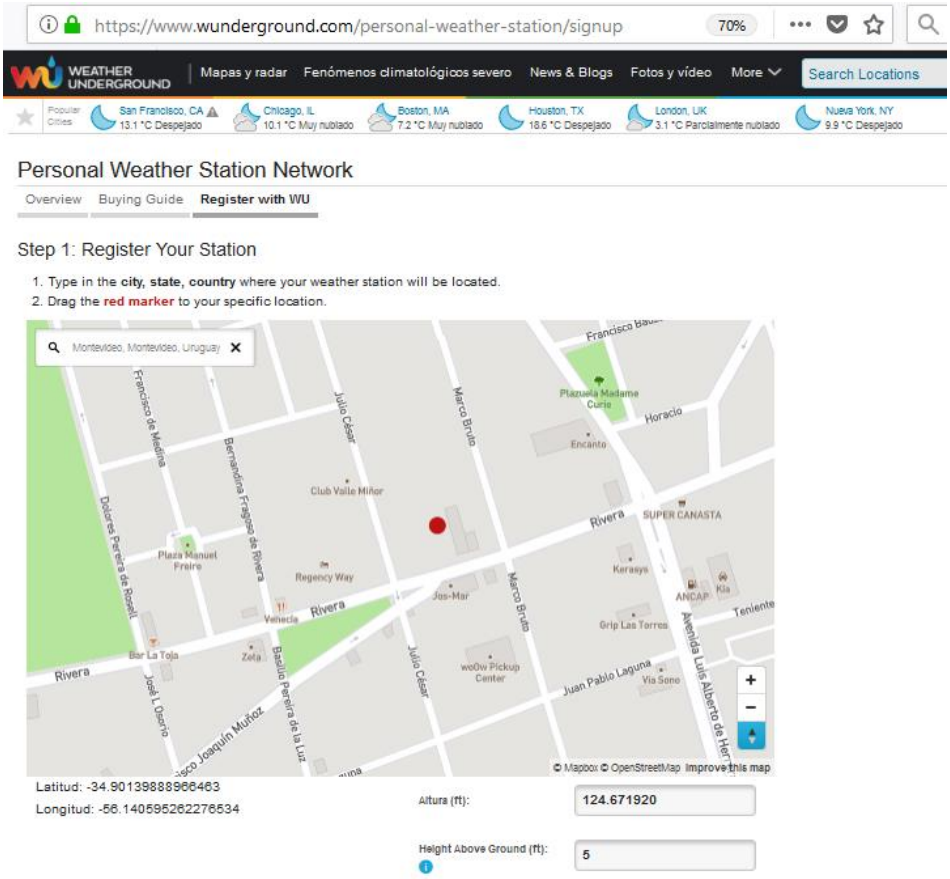


Figura 102 - Registro de PWS en *Wunderground*

Actualmente la estación se encuentra alimentada por un transformador de 9V conectado a corriente; pero podría implementarse como mejora el incorporar alimentación mediante paneles solares. Conjuntamente se podrá agregar una caja aislante y más resistente a los efectos del clima en la cual se proteja a las placas y sensores alojados en ellas.

7. Referencias bibliográficas

[1] Y.Radhika and M.Shashi, “Atmospheric Temperature Prediction using Support Vector Machines,” *International Journal of Computer Theory and Engineering*, vol. 1, no. 1, pp. 55-58, April, 2009. [Online]. Available: https://pdfs.semanticscholar.org/ec5b/f80b2f29526213e6e5623613a6dc23942437.pdf. Accessed on: Jan, 11, 2018.

[2] Antonio S. Cofiño and Rafael Cano and Carmen Sordo and José M. Gutierrez, “Bayesian Networks for Probabilistic Weather Prediction ,” *ECAI 2002. Proceedings of the 15th European Conference on Artificial Intelligence*, IOS Press, pp. 695-700, 2002. [Online]. Available: https://grupos.unican.es/ai/meteo/articulos/ECAI2002.pdf. Accessed on: Jan, 11, 2018.

[3] Enrica Bellone and James P. Hughes and Peter Guttorp, “A hidden Markov model for downscaling synoptic atmospheric patterns to precipitation amounts,” *NRCSE. Technical Report Series*, NRCSE-TRS, no. 21, pp. 1-30. [Online]. Available: http://www.nrcse.washington.edu/pdf/trs21_markov.pdf. Accessed on: Jan, 11, 2018.

[4] Aditya Grover and Ashish Kapoor and Eric Horvitz, “A Deep Hybrid Model for Weather Forecasting”. [Online]. Available: http://aditya-grover.github.io/files/publications/kdd15.pdf . Accessed on: Jan, 11, 2018.

[5] Nazim Osman Bushara and Ajith Abraham, “Weather Forecasting in Sudan Using Machine Learning Schemes ,” *Journal of Network and Innovative Computing*, vol. 2, pp. 309-317, 2014. [Online]. Available: http://www.mirlabs.net/jnic/secured/Volume2-Issue1/Paper32/JNIC_Paper32.pdf. Accessed on: Jan, 11, 2018.

[6] Sanam Narejo and Eros Pasero, “Meteonowcasting using Deep Learning Architecture,” *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 8, no. 8, pp. 16-23, 2017. [Online]. Available: https://thesai.org/Downloads/Volume8No8/Paper_3-MeteoNowcasting_using_Deep_Learning_Architecture.pdf. Accessed on: Jan, 11, 2018.

[7] Dr. S. Baghavathi Priya and A. Muthulakshmi and S. Usha, “Forecasting Weather to Predict Rainfall for Sustainable Agriculture using Machine Learning Techniques ,” *Asian Journal of Research in Social Sciences and Humanities*, vol. 6, no. 6, pp. 1846-1857, June, 2016. [Online]. Available: https://aijsh.com/shop/articlepdf/2016/06/1464758828146.pdf. Accessed on: Jan, 11, 2018.

[8] Lily Ingsrisawang, Supawadee Ingsriswang, Saisuda Somchit, Prasert Aungsuratana, and Warawut Khantiyanan, “Machine Learning Techniques for

Short-Term Rain Forecasting System in the Northeastern Part of Thailand," *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*, vol. 2, no. 5, pp. 1422-1427, 2008. [Online]. Available: http://waset.org/publications/5358/machine-learning-techniques-for-short-term-rain-forecasting-system-in-the-northeastern-part-of-thailand. Accessed on: Jan, 11, 2018.

[9] Binghong Chen, Cong Luo *et al*, "Non-Linear Machine Learning Approach to Short-Term Precipitation Forecasting" [Online]. Available: http://meetings.wmo.int/CBS-16/TECO/Presentations/21-November-Monday/1A1-Non-Linear%20Machine%20Learning%20Approaches%20to%20Short-Term%20precipitation%20forecasting.pdf. Accessed on: Jan, 11, 2018.

[10] S. MONIRA SUMI, M. FAISAL ZAMAN, HIDEO HIROSE, "A RAINFALL FORECASTING METHOD USING MACHINE LEARNING MODELS AND ITS APPLICATION TO THE FUKUOKA CITY CASE," *Int. J. Appl. Math. Comput. Sci* , vol. 22, no. 4, pp. 841-854, 2012 [Online]. Available: http://matwbn.icm.edu.pl/ksiazki/amc/amc22/amc2245.pdf. Accessed on: Jan, 11, 2018.

[11] Mohini P. Darji, Vipul Dabhi, Harshadkumar B Prajapati, "Rainfall forecasting using neural network: A survey," *Computer Engineering and Applications (ICACEA)*
, 2015 [Online]. Available: https://www.researchgate.net/publication/280451881_Rainfall_forecasting_using_neural_network_A_survey.Accessed on: Jan, 11, 2018.

[12] Aakash Parmar, Kinjal Mistree and Mithila Sompura, "Machine Learning Techniques For Rainfall Prediction: A Review," *Conference: 2017 International Conference on Innovations in information Embedded and Communication Systems (ICIIECS)*
, 2017 [Online]. Available: https://www.researchgate.net/publication/319503839_Machine_Learning_Techniques_For_Rainfall_Prediction_A_Review.Accessed on: Jan, 11, 2018.

[13] Xingjian Shi, Zhourong Chen, Hao Wang and Dit-Yan Yeung, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting", pp. 1-9 [Online]. Available: http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting.pdf
.Accessed on: Jan, 28, 2018.

[14] Research Application Laboratory (RAL), "Advancing Weather Analysis and Forecasting Technologies" [Online]. Available: https://ral.ucar.edu/projects/advancing-weather-analysis-and-forecasting-technologies. Accessed on: Jan, 28, 2018.

[15] “Hurricane Harvey, Forecasting Weather With Machine Learning Artificial Intelligence”. [Online]. Available: https://traderscommunity.com/index.php/technology/171-forecasting-weather-with-machine-learning-artificial-intelligence. Accessed on: Jan, 28, 2018.

[16] “Machine Learning for Sales Forecasting Using Weather Data”. [Online]. Available: http://mariofilho.com/machine-learning-sales-forecasting-using-weather-data/. Accessed on: Jan, 28, 2018.

[17] Pablo Rozas Larraondoa,Iˆnaki Inzab, Jose A. Lozano, “A system for airport weather forecasting based on circular regression trees”, 2017 [Online]. Available: https://bird.bcamath.org/bitstream/handle/20.500.11824/749/system-airport-weather-2.pdf?sequence=1&isAllowed=y.Accessed on: Jan, 28, 2018.

[18] Ankur Sahai, “Evaluation of Machine Learning Techniques for Green Energy Prediction”, 2014 [Online]. Available: https://arxiv.org/pdf/1406.3726.pdf .Accessed on: Jan, 28, 2018.

[19] Siddharth S. Bhatkande1, Roopa G. Hubballi, “Weather Prediction Based on Decision Tree Algorithm Using Data Mining Techniques,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 5, pp. 483-487, 2016. [Online]. Available: https://www.ijarcce.com/upload/2016/may-16/IJARCCE%20114.pdf. Accessed on: Jan, 28, 2018.

[20] John K. Williams and D. A. Ahijevych, C. J. Kessinger, T. R. Saxen, M. Steiner and S. Dettling, “A MACHINE LEARNING APPROACH TO FINDING WEATHER REGIMES AND SKILLFUL PREDICTOR COMBINATIONS FOR SHORT-TERM STORM FORECASTING” [Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiNoPK12_vYAhUGFZAKHRvtDu0QFggmMAA&url=https%3A%2F%2Fams.confex.com%2Fams%2Fpdfpapers%2F135663.pdf&usg=AOvVaw0skFfCtBBsdM0HR_b_INdW
.Accessed on: Jan, 28, 2018.

[21] Kickstarter, “Dark Sky - Weather Prediction, Reinvented” [Online]. Available: https://www.kickstarter.com/projects/jackadam/dark-sky-hyperlocal-weather-prediction-and-visuali. Accessed on: Jan, 28, 2018.

[22] Wikipedia, “Internet de las cosas” [Online]. Available: https://es.wikipedia.org/wiki/Internet_de_las_cosas. Accessed on: Feb, 12, 2018.

[23] Sparkfun, “What is an Arduino?” [Online]. Available: https://learn.sparkfun.com/tutorials/what-is-an-arduino. Accessed on: Feb, 12, 2018.

[24] Wikipedia, “Cloud computing” [Online]. Available: https://en.wikipedia.org/wiki/Cloud_computing. Accessed on: Feb, 17, 2018.

[25] Forbes, “Cloud computing vendors Top 5” [Online]. Available: https://www.forbes.com/sites/bobevans1/2017/11/07/the-top-5-cloud-computing-vendors-1-microsoft-2-amazon-3-ibm-4-salesforce-5-sap/#ce57d1b6f2eb. Accessed on: Feb, 17, 2018.

[26] Outage Prediction (The Weather Company), “Proactively respond to storms and other severe weather conditions.” [Online]. Available: https://business.weather.com/products/outage-prediction. Accessed on: Feb, 21, 2018.

[27] Ignacio Chiazzo, Felipe García, Guillermo Leopold, “Relevamiento y obtención de datos sobre emergencias en Uruguay para análisis predictivo”, Universidad de la República (Uruguay), Facultad de Ingeniería, INCO, 2016.

[28] Sears, Kathleen, in *Weather 101: From Doppler Radar and Long-Range Forecasts to the Polar Vortex and Climate Change, Everything You Need to Know about the Study of Weather*: Amazon, 2017.

[29] Inumet, “Ley Nº 19.158” [Online]. Available: https://inumet.gub.uy/reportes/institucional/LEY_19158.pdf. Accessed on: Jan, 26, 2018.

[30] El Observador, “Meteorología a ciegas” [Online]. Available: https://www.elobservador.com.uy/meteorologia-ciegas-n1016707. Accessed on: Jan, 26, 2018.

[31] Diego Iribarren Baró, “MODELO PREDICTIVO. MACHINE LEARNING APLICADO AL ANÁLISIS DE DATOS CLIMÁTICOS CAPTURADOS POR UNA PLACA SPARKFUN”, 2016 [Online]. Available: https://repositorio.comillas.edu/xmlui/bitstream/handle/11531/14322/TFG000958.pdf?sequence=1&isAllowed=y.Accessed on: Jan, 28, 2018.

[32] Sparkfun, “SparkFun Weather Shield” [Online]. Available: https://www.sparkfun.com/products/12081. Accessed on: Jan, 28, 2018.

[33] Sparkfun, “Arduino Yun” [Online]. Available: https://www.sparkfun.com/products/12053. Accessed on: Jan, 28, 2018.

[34] Sparkfun, “Weather Meters” [Online]. Available: https://www.sparkfun.com/products/8942. Accessed on: Jan, 28, 2018.

[35] Node-RED, “Flow-based programming for the Internet of Things” [Online]. Available: https://nodered.org/. Accessed on: Jan, 28, 2018.

[36] Network Technology, “Using Amazon Machine Learning to Predict the Weather” [Online]. Available: https://arnesund.com/2015/05/31/using-amazon-machine-learning-to-predict-the-weather/. Accessed on: Jan, 28, 2018.

[37] Inumet, [Online]. Available: http://www.meteorologia.com.uy/. Accessed on: Jan, 26, 2018.

[38] Microsoft, “Portal Azure” [Online]. Available: https://portal.azure.com. Accessed on: Oct, 17, 2017.

[39] Thomas Erl, Wajid Khattak, and Paul Buhler, *Big Data Fundamentals Concepts, Drivers & Techniques,* Boston, (Massachusetts), USA: Prentice Hall, 2015.

[40] Jared Dean, *Big Data, Data Mining, and Machine Learning*, Hoboken, (New Jersey), USA: WILEY, 2014.

[41] Tom M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[42] Scott Hartshorn, *Machine Learning with Random Forest and Decision Trees.*

[43] Michael Taylor, *Neural Networks A Visual Introduction For Beginners*, 2017.

[44] Dilbert, “Dilbert by Scott Adams” [Online]. Available: http://dilbert.com/search_results?terms=machine+learning Accessed on: Jan, 26, 2018

[45] Pinterest, “Pinterest” [Online]. Available: https://www.pinterest.es/pin/27303141469284489 Accessed on: Jan, 26, 2018

[46] MathWorks, “Machine Learning (Aprendizaje automático)” [Online]. Available: https://la.mathworks.com/discovery/aprendizaje-automatico.html Accessed on: Jan, 26, 2018

[47] Slideshare, “How Big Data and Machine Learning Are Transforming ITSM” [Online]. Available: https://es.slideshare.net/SunViewSoftware/webinar-how-big-data-and-machine-learning-are-transforming-itsm Accessed on: Jan, 26, 2018

[48] Arduino.cl, “Arduino UNO R3” [Online]. Available: http://arduino.cl/arduino-uno/ Accessed on: Jan, 26, 2018

[49] Sparkfun, “Weather Shield Hookup Guide V12” [Online]. Available: https://learn.sparkfun.com/tutorials/arduino-weather-shield-hookup-guide-v12 Accessed on: Jan, 26, 2018

[50] Cloudcomputingmary.blogspot.com, “Computación en la Nube” [Online]. Available: http://cloudcomputingmary.blogspot.com.uy/ Accessed on: Jan, 26, 2018

[51] Sparkfun, “GPS Receiver - GP-735 (56 Channel)” [Online]. Available: https://www.sparkfun.com/products/13670 Accessed on: Jan, 26, 2018

[52] Machine Learning Mastery, “Supervised and Unsupervised Machine Learning Algorithms” [Online]. Available: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ Accessed on: Jan, 26, 2018

[53] Chris Smith, *Decision Trees and Random Forests: A Visual Introduction for Beginners*, Blue Windmill Media, 2017

[54] Wikimedia Commons, “File:Titanic Survival Decision Tree SVG.png” [Online]. Available: https://commons.wikimedia.org/wiki/File:Titanic_Survival_Decison_Tree_SVG.png Accessed on: Jan, 26, 2018

[55] Jake VanderPlas, Python Data Science Handbook, O’Reilly, 2016

[56] Scikitlearn, “SVM: Maximum margin separating hyperplane” [Online]. Available: http://scikit-learn.org/stable/auto_examples/svm/plot_separating_hyperplane.html#sphx-glr-auto-examples-svm-plot-separating-hyperplane-py Accessed on: Jan, 26, 2018

[57] IBM, “IBM Cloud” [Online]. Available: https://www.ibm.com/cloud/ Accessed on: Jan, 26, 2018

[58] IBM, “The Weather Company” [Online]. Available: http://www.theweathercompany.com/ Accessed on: Jan, 26, 2018

[59] IBM, “Deep Thunder” [Online]. Available: http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepthunder/ Accessed on: Jan, 26, 2018

[60] The Weather Company, “Weather for Agriculture - Integrative Solutions” [Online]. Available: https://business.weather.com/industry-solutions/agriculture Accessed on: Jan, 26, 2018

[61] CleverData, “¿Qué es Machine Learning?” [Online]. Available: http://cleverdata.io/que-es-machine-learning-big-data/ Accessed on: Jan, 26, 2018

[62] Wikiquote, “Theophrastus” [Online]. Available: https://en.wikiquote.org/wiki/Theophrastus. Accessed on: Feb, 12, 2018.

[63] Pinterest, “Pinterest” [Online]. Available: https://www.pinterest.com.mx/pin/495325659001964022/. Accessed on: Feb, 12, 2018.

[64] Wikipedia, “High Frequency Active Auroral Research Program” [Online]. Available: https://es.wikipedia.org/wiki/High_Frequency_Active_Auroral_Research_Program. Accessed on: Feb, 20, 2018.

[65] Dreamstime, “Earth’s atmosphere Layers” [Online]. Available: https://www.dreamstime.com/stock-images-earth-s-atmosphere-layers-image22603834. Accessed on: Feb, 20, 2018.

[66] Universities Space Research Association, “Cumulonimbus Calvus on Radar” [Online]. Available: http://epod.usra.edu/blog/2006/06/cumulonimbus-calvus-on-radar.html. Accessed on: Feb, 20, 2018.

[67] Curiosfera, “Qué son y cómo se forman las nubes” [Online]. Available: http://www.curiosfera.com/nubes-que-son-formacion-tipos/. Accessed on: Feb, 20, 2018.

[68] Wikipedia, “Higrómetro” [Online]. Available: https://es.wikipedia.org/wiki/Higr%C3%B3metro. Accessed on: Feb, 20, 2018.

[69] SlidePlayer, “Global Winds and Jet Stream” [Online]. Available: http://slideplayer.com/slide/10534785/. Accessed on: Feb, 20, 2018.

[70] Biblioteca virtual de desarrollo sostenible y salud ambiental, “La estructura dinámica de la atmósfera” [Online]. Available: http://www.bvsde.paho.org/cursoa_meteoro/lecc3/lecc3_1.html. Accessed on: Feb, 20, 2018.

[71] Wikipedia, “Barómetro aneroide” [Online]. Available: https://es.wikipedia.org/wiki/Bar%C3%B3metro_aneroide. Accessed on: Feb, 20, 2018.

ANEXO 1 – Planificación del Proyecto

A continuación, se detallan las tareas realizadas y aquellas que aún están pendientes.

Actividad / Hito	Fecha Inicio	Fecha Fin	Responsable
Definición de material de referencia de ML (1)	17/05/2017	31/01/2018	Sergio Yovine
Capacitación en ML(2)	18/05/2017	31/01/2018	Natalie Gnoza, Marcelo Barberena
Capacitación software modelo predictivo (3)	18/05/2017	31/01/2018	Natalie Gnoza, Marcelo Barberena
Investigar modelos numéricos utilizados actualmente para meteorología (4)	01/06/2017	31/01/2018	Natalie Gnoza, Marcelo Barberena
Definir los algoritmos de aprendizaje a ser utilizados para el problema (5)	01/06/2017	31/01/2018	Sergio Yovine Natalie Gnoza, Marcelo Barberena
Elaboración primer informe de avance	06/08/2017	10/08/2017	Sergio Yovine Natalie Gnoza, Marcelo Barberena
Entrega Primer informe de avance	**10/08/2017**	**10/08/2017**	Natalie Gnoza, Marcelo Barberena
Obtener fuente de datos para prueba de concepto de modelo predictivo (6)	22/07/2017	31/01/2018	Natalie Gnoza, Marcelo Barberena
Prueba de concepto de modelo predictivo - versión 1 ... N	17/11/2017	01/03/2018	Sergio Yovine Natalie Gnoza, Marcelo Barberena
Evaluación e informe de fiabilidad del modelo predictivo	17/11/2017	01/03/2018	Sergio Yovine Natalie Gnoza, Marcelo Barberena
Investigación ensamblado de componentes Arduino (7)	01/07/2017	01/03/2018	Natalie Gnoza, Marcelo Barberena
Construcción de prototipo de PWS	01/11/2017	01/03/2018	Natalie Gnoza, Marcelo Barberena
Elaboración segundo informe de avance	25/11/2017	30/11/2017	Sergio Yovine Natalie Gnoza, Marcelo Barberena
Segundo informe de avance	**30/11/2017**	**30/11/2017**	**Natalie Gnoza, Marcelo Barberena**
Versión final del modelo predictivo (mayor volumen y fuentes de datos)	01/12/2017	01/03/2018	Natalie Gnoza, Marcelo Barberena
Entrega final	**01/03/2018**	**01/03/2018**	**Natalie Gnoza, Marcelo Barberena**
Defensa	**-**	**-**	**Natalie Gnoza, Marcelo Barberena**

(1) Se definió material de referencia de modo de comenzar a tener idea global sobre técnicas de analítica y arquitectura de software de analítica.

- ***Big Data Fundamentals. Concepts, Drivers & Techniques***.
  Thomas Erl, Wajid Khattak, Paul Buhler
- ***Big Data, Data Mining and Machine Learning*.**
  Value Creation for Business Leaders and Practitioners.
  Jared Dean
- ***Python Data Science Handbook***
  Jake VanderPlas

(2) Se dio comienzo a la investigación en función de la bibliografía recomendada en el punto anterior.

(3) Se dio comienzo a la investigación en función de la bibliografía y las herramientas definidas.

(4) (5) Se comenzaron a investigar los distintos algoritmos utilizados hoy en día en proyectos similares para la construcción del modelo, entre ellos tenemos:

- ***Multiclass decision forest***
- ***Multiclass decision jungle***
- ***Multiclass logistic regression***
- ***Multiclass neural network***
- ***Random Forest***

Una vez analizados los algoritmos se configurarán los parámetros de entrada necesarios, se comparará la precisión y porcentaje de acierto de cada uno de ellos de modo de determinar el que más se ajuste al problema planteado, así como también se medirá el tiempo necesario de entrenamiento del modelo.

El enfoque que tendremos que definir que aplique a la predicción de las Precipitaciones en función de Temperatura, Humedad y Presión Atmosférica puede estar entre:

- Predecir un valor concreto (numérico) de la variable Precipitación, en este caso estaríamos frente a un **Problema de Regresión**.

- Predecir un conjunto de estados discretos o categóricos, **Problemas de Clasificación** entre los que encontramos:

  - **Binarios** – "Lloverá hoy?" – Respuesta: {Si, No}
  - **Ordenada –** "Probabilidad de lluvias hoy?" – Respuesta {Baja, Media, Alta}

También se analizarán otros algoritmos dentro de los grupos principales en los que se pueden categorizar las soluciones aportadas por *Machine Learning*:

- **Modelos Lineales**
- **Modelos de Árbol**
- **Redes Neuronales**

(6) Tras un análisis se llegó a la conclusión de la importancia de comenzar tempranamente con la adquisición de datos para alimentarán el modelo predictivo (*software*) a construir.

(7) Se comenzó en forma temprana con la investigación de los componentes necesarios para la construcción de la PWS pues será una fuente más en la fase de adquisición de datos para nuestro modelo.

- ***Arduino* UNO**
- ***Sparkfun Weather Shield***
- ***GPS***
- **Conectores (pines y RJ11)**
- **Estaño**
- **Soldador**
- **Batería**
- ***Software* Arduino (IDE)**

Ver información de precios y detalle de componentes en la sección Anexo 2 – Mini estación meteorológica.

La mayoría de los componentes fueron adquiridos en plaza en casas de electrónica con gran facilidad a excepción de la placa de sensores (Sparkfun *Weather Shield*), GPS y algunos conectores que compramos directamente al fabricante en USA. Esta estrategia no solo nos permitió conseguir componentes compatibles entre sí, sino que también logró bajar la demora en la importación de 30 días (casas de electrónica) a tan solo 10 días mediante Miami Box.

El día 27 de Setiembre a las 9 hs se realizó una reunión en SinergiaTech con uno de los socios directores Max Patissier (max.patissier@sinergiatech.com), de modo de contar con asesoramiento de equipos idóneos en el área electrónica para la elección y ensamblado de la PWS.

Se realizaron también consultas directamente a los fabricantes de la placa Sparkfun de modo de obtener información de componentes adicionales, así como anticiparnos a problemas de incompatibilidad entre ellos. Consultamos también sobre las posibles formas de resolver la conectividad de modo que la PWS pueda transmitir los datos recogidos por los diferentes sensores tanto sea vía WiFi, Ethernet, xBee, etc.

Otros temas vinculados:

- Uno de los integrantes del equipo cursó el semestre pasado la electiva *Machine Learning* para Sistemas inteligentes, que significó un gran aporte para el desarrollo del proyecto.

- El pasado 12 de julio a las 19hs asistimos a la conferencia sobre IBM Cloud POV realizada en el Auditorio Campus Centro de ORT, en la que se presentaron los temas:
  - **El punto de vista *Cloud* para IBM**
  - ***Analytics* en *Cloud***
  - **El desafío de la Arquitectura Híbrida**
  - **Bluemix público, dedicado y local**

Fue importante para comenzar a consolidar conocimiento sobre la plataforma en la que comenzamos a trabajar, si bien luego se optó tras diversos inconvenientes por utilizar Azure de Microsoft.

- El día 23 de octubre a las 19:30 hs asistimos a la conferencia de Técnicas de *Machine Learning* para el procesamiento del habla, realizada en el auditorio de la Universidad ORT a cargo de Agustín Gravano, profesor adjunto del Departamento de Computación de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires e investigador adjunto en CONICET.

- El día 9 de noviembre de 9 a 16hs asistimos al evento *Disruptive Day* organizado por la UM, la Escuela de Negocios IEEM y otras tantas empresas, donde se brindaron charlas y trataron temas vinculados directamente con nuestro proyecto, a saber “Computación cognitiva aplicada a *Commerce* y *Marketing*”, “*Internet of Things*”, “*Big Data*: presente y futuro”, “Watson en su Negocio” entre otras.

ANEXO 2 – Mini estación meteorológica

En esta sección se presentan algunos de los costos (en EUR y USD) asociados a la mini estación meteorológica.

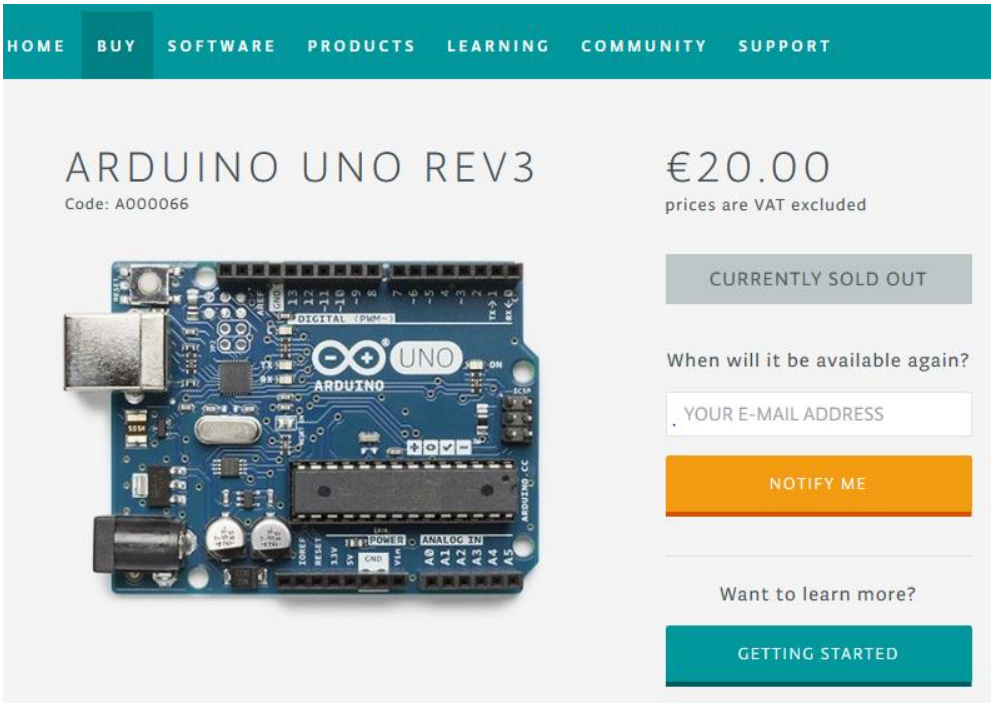
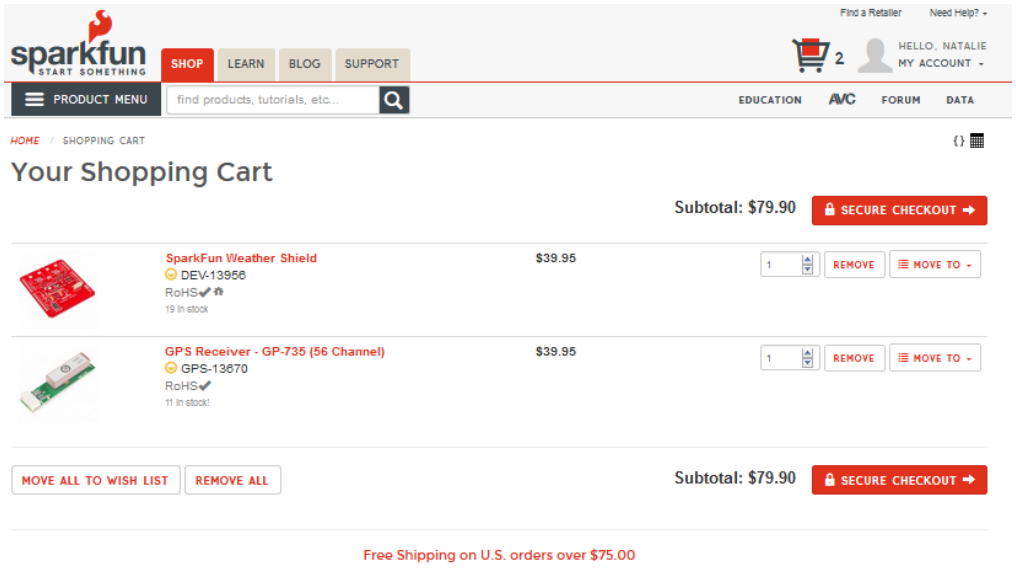


Figura 103 - Placa Arduino Uno
Fuente: [48]



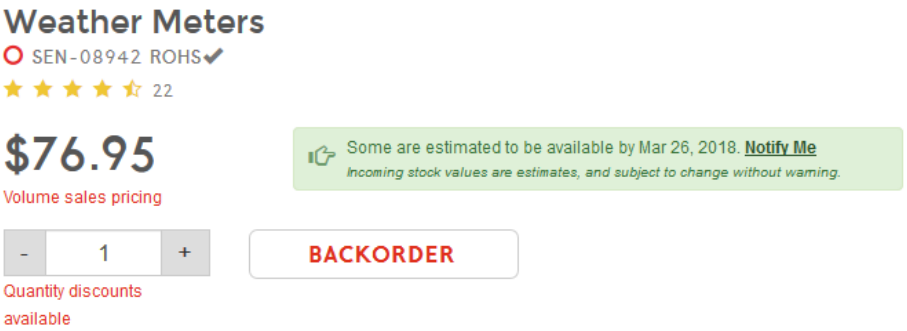


Figura 104 - Lista de compra de componentes de la PWS
Fuente: [51] [34]

ANEXO 3 - Modelos de *Machine Learning* utilizados

En la figura 105 se muestra los diferentes algoritmos de *machine learning* existentes agrupados según su tipo.

En este apartado solo se describen aquellos que fueron utilizados en el proyecto para realizar las predicciones de la variable meteorológica precipitaciones (lluvia).

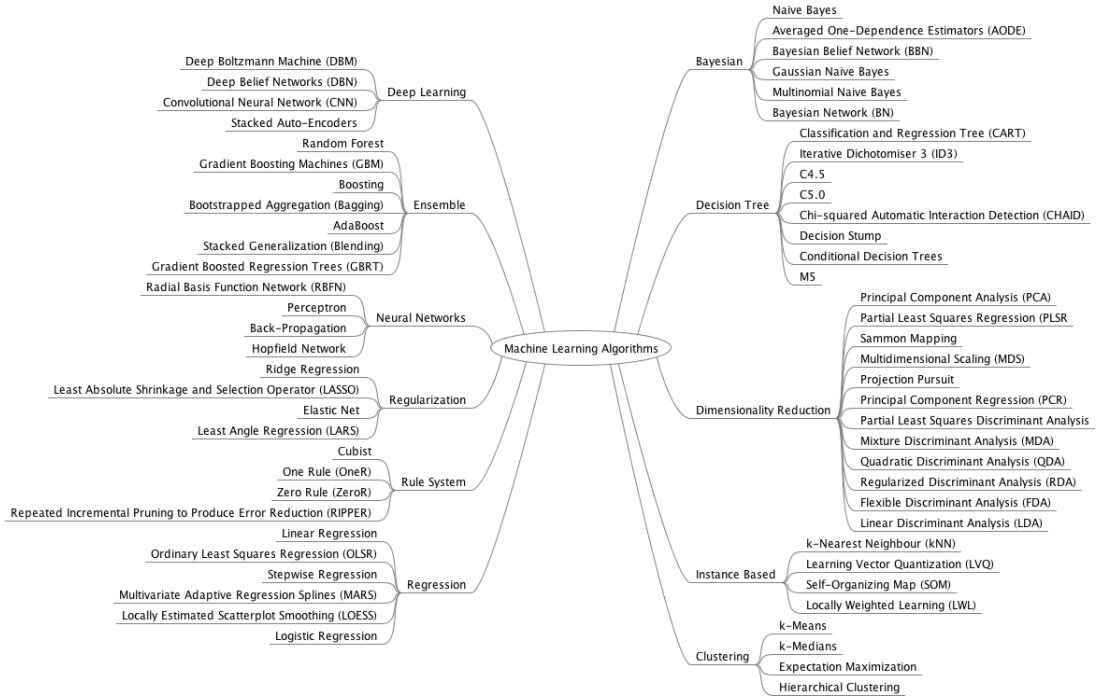


Figura 105 – Modelos de *Machine Learning*
Fuente: [52]

Classification and Regression Tree (CART)

El algoritmo utiliza un árbol de decisiones (como modelo predictivo) para pasar de las observaciones sobre un elemento (representadas en las ramas) a conclusiones sobre el valor objetivo del artículo (representado en las hojas). Es uno de los enfoques de modelado predictivo utilizados en estadística, minería de datos y aprendizaje automático. Los modelos de árbol donde la variable objetivo puede tomar un conjunto discreto de valores se denominan árboles de clasificación; en estas estructuras de árbol, las hojas representan etiquetas de

clase y las ramas representan conjunciones de características que conducen a esas etiquetas de clase. Los árboles de decisión donde la variable objetivo puede tomar valores continuos (típicamente números reales) se llaman árboles de regresión.

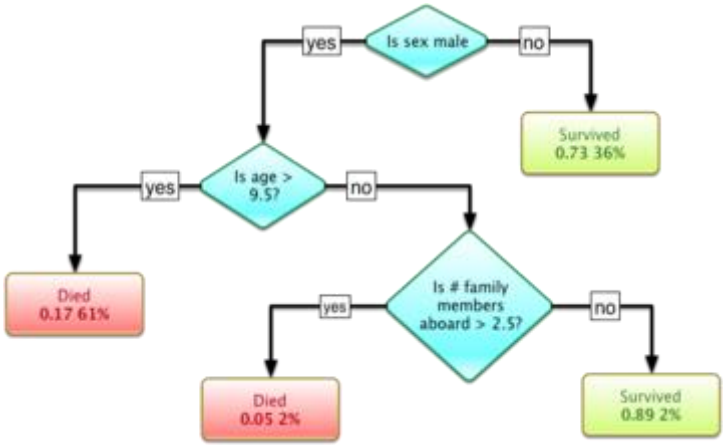


Figura 106 – Árbol de clasificación
Fuente: [54]

Los árboles de decisión se crean dividiendo primero los datos en dos grupos. Este proceso de división binaria se repite en cada rama (capa). El objetivo es seleccionar una pregunta binaria que divida mejor los datos en dos grupos homogéneos en cada rama del árbol, de modo que se minimice el nivel de entropía de datos en el siguiente. La entropía es un término matemático que explica la medida de la varianza en los datos entre diferentes clases. En términos simples, queremos que los datos en cada capa sean más homogéneos que en el último. Por lo tanto, queremos elegir un algoritmo 'codicioso' que pueda reducir el nivel de entropía en cada capa del árbol. Uno de esos algoritmos codiciosos es el *Iterative Dichotomizer* (ID3), inventado por J.R. Quinlan. Esta fue una de las tres implementaciones del árbol de decisiones desarrolladas por Quinlan, de ahí el “3”. ID3 aplica la entropía para determinar qué pregunta binaria preguntar en cada capa del árbol de decisión. En cada capa, ID3 identifica una variable (convertida en una pregunta binaria) que producirá la menor entropía en la siguiente capa.

Ya sea ID3 u otro algoritmo, el proceso de división de datos en particiones binarias, conocido como partición recursiva, se repite hasta que se cumple un

criterio de detención. Este punto de detención puede basarse en varios criterios, como:

- Cuando todas las hojas contienen menos de 3-5 elementos
- Cuando una rama produce un resultado que coloca todos los elementos en una hoja binaria

Una advertencia de los árboles de decisión es su susceptibilidad al sobreajuste. La causa del sobreajuste, en este caso, son los datos de entrenamiento. Teniendo en cuenta las reglas de clasificación que existen en los datos de capacitación, un árbol de decisión es preciso en el entrenamiento de la primera ronda de datos. Sin embargo, ese modelo de árbol de decisión producido por los datos de entrenamiento puede no aplicarse a los datos de prueba debido a la existencia de nuevas reglas de clasificación. En este caso, los datos de entrenamiento y/o los datos de prueba no son representativos de todo el conjunto de datos. Además, dado que los árboles de decisión se forman a partir de dividir repetidamente los puntos de datos en dos particiones, un ligero cambio en la forma en que se dividen los datos en la parte superior o en la mitad del árbol puede alterar drásticamente los resultados finales. Esto puede producir un árbol diferente por completo. Desde la primera división de los datos, el algoritmo codicioso selecciona las preguntas binarias que mejor dividen los datos en dos grupos homogéneos. Sin embargo, como un niño sentado frente a una caja de *cupcakes*, el algoritmo codicioso no tiene en cuenta las repercusiones futuras de sus acciones a corto plazo. La pregunta binaria que se aplica para dividir inicialmente los datos no garantiza la precisión con respecto al resultado final. Por el contrario, una división inicial menos eficiente puede producir un resultado más preciso. En resumen, los árboles de decisión son altamente visuales y efectivos para clasificar un único conjunto de datos. Sin embargo, los árboles de decisión pueden ser inflexibles y vulnerables al sobreajuste.

¿Cómo un árbol de decisión elige sus particiones?

Una cosa interesante acerca de cada árbol de decisión es cómo se genera el umbral para cada decisión. Se eligen específicamente para maximizar la ganancia de información en cada paso. Eso significa que para cualquier criterio que se elija, se dividirá en la mejor ubicación dados los datos en esa rama. La pregunta obvia es "¿Qué es lo mejor?".

La solución más comúnmente utilizada es el criterio "Gini" o el criterio de "Entropía".

- **Criterio “Gini”**

  La ecuación para la impureza de Gini es:

$$Gini = 1 - \sum_j p_j^2$$

  Donde p es la probabilidad de tener una clase de datos dada en su conjunto de datos.

- **Criterios de entropía**

  Un criterio alternativo para determinar las divisiones es la entropía.

  La ecuación de entropía es

$$Entropy = \sum_j -p_j * log_2(p_j)$$

  Para entropía, al igual que los criterios de Gini, cuanto menor sea el número, mejor, siendo el mejor una entropía de cero.

Overfitting

Los algoritmos del árbol de decisiones siempre clasificarán perfectamente su conjunto de datos de entrenamiento, a menos que adopte un método de detención que lo impida. Esto significa que, si es necesario, el algoritmo continuará dividiéndose hasta que cada subconjunto sea un solo ejemplo y 100% puro. Por un lado, esto es fantástico porque el algoritmo aprende a clasificar muy bien el conjunto de datos de tu entrenamiento. Esto se debe a que a medida que el algoritmo continúa dividiéndose, el árbol continúa creciendo y se vuelve más y más preciso en los datos de entrenamiento.

El factor decisivo, sin embargo, es que en algún momento también comienza a ser menos precisa en los datos de prueba nuevos, y eso no es bueno en absoluto. Este problema se conoce como sobreajuste. El sobreajuste se produce cuando un algoritmo se entrena tan estrechamente en ejemplos específicos que no puede comprender correctamente y trabajar con los nuevos ejemplos que se le presentan. El hecho es que cuando un algoritmo de árbol de decisión divide y divide su conjunto de datos, el árbol que crea es muy específico para ese conjunto de datos, lo que significa que puede tener dificultades para trabajar con los datos nuevos que se le presentan. La capacidad de trabajar con nuevos ejemplos de prueba y clasificarlos correctamente se denomina técnicamente "generalizar", y los árboles que no se ajustan bien no se pueden generalizar bien.

Tipos de árboles de decisión

El término análisis de árbol de regresión y clasificación (CART) es un término general usado para referirse a ambos procedimientos anteriores, presentado por primera vez por Breiman et al. Los árboles utilizados para la regresión y los árboles utilizados para la clasificación tienen algunas similitudes, pero también algunas diferencias, como el procedimiento

utilizado para determinar dónde dividir. Algunas técnicas, a menudo llamadas métodos de conjunto, construyen más de un árbol de decisión:

- **Árboles potenciados (*Boosted trees*)**, Construye incrementalmente un conjunto entrenando cada nueva instancia para enfatizar las instancias de entrenamiento previamente mal modeladas. Un ejemplo típico es AdaBoost. Estos pueden usarse para problemas de tipo de regresión y clasificación.
- **Los árboles de decisión agregados (*Bootstrap aggregated trees*)**, un método de conjunto temprano, construye árboles de decisiones múltiples al volver a muestrear los datos de entrenamiento con reemplazo, y votar los árboles para una predicción de consenso.
- **Un clasificador de bosque aleatorio (*Random forest*)** es un tipo específico de agregación de *bootstrap*.
- **Bosque de rotación (*Rotation forest*):** cada árbol de decisión se entrena primero aplicando el análisis del componente principal (PCA) en un subconjunto aleatorio de las características de entrada.

Ventajas

- Simple de entender e interpretar. Las personas pueden comprender los modelos del árbol de decisiones después de una breve explicación. Los árboles también se pueden mostrar gráficamente de una manera que es fácil de interpretar por los no expertos.
- Capaz de manejar datos numéricos y categóricos. Otras técnicas suelen estar especializadas en analizar conjuntos de datos que tienen solo un tipo de variable. (Por ejemplo, las reglas de relación solo se pueden usar con variables nominales, mientras que las redes neuronales solo se pueden usar con variables numéricas o categorías convertidas a valores 0-1).

- Requiere poca preparación de datos. Otras técnicas a menudo requieren normalización de datos.
- Como los árboles pueden manejar predictores cualitativos, no es necesario crear variables ficticias.
- Utiliza un modelo de caja blanca. Si una situación dada es observable en un modelo, la explicación de la condición se explica fácilmente por la lógica booleana. Por el contrario, en un modelo de caja negra, la explicación de los resultados suele ser difícil de entender, por ejemplo, con una red neuronal artificial.
- Posibilidad de validar un modelo usando pruebas estadísticas. Eso hace posible dar cuenta de la fiabilidad del modelo.
- Enfoque no estadístico que no hace suposiciones sobre los datos de entrenamiento o los residuos de predicción; por ejemplo, sin supuestos de distribución, independencia o varianza constante
- Se desempeña bien con grandes conjuntos de datos. Se pueden analizar grandes cantidades de datos utilizando recursos informáticos estándar en un tiempo razonable.
- Refleja la toma de decisiones humanas más de cerca que otros enfoques. Esto podría ser útil al modelar las decisiones / comportamientos humanos.
- Robusto contra la colinealidad, particularmente potenciar
- En la selección de características. Las características irrelevantes adicionales serán menos utilizadas y pueden eliminarse en ejecuciones posteriores.

Desventajas

- Los árboles no tienden a ser tan precisos como otros enfoques.
- Los árboles pueden ser no robustos. Un pequeño cambio en los datos de entrenamiento puede resultar en un gran cambio en el árbol, y por lo tanto un gran cambio en las predicciones finales.
- Se sabe que el problema de aprender un árbol de decisión óptimo es NP completo en varios aspectos de optimalidad e incluso para

conceptos simples. En consecuencia, los algoritmos prácticos de aprendizaje del árbol de decisiones se basan en heurísticas tales como el algoritmo codicioso en el que se toman decisiones localmente óptimas en cada nodo. Tales algoritmos no pueden garantizar devolver el árbol de decisión óptimo globalmente. Para reducir el efecto codicioso de la optimalidad local, se propusieron algunos métodos, como el árbol de distancia de información dual (DID).

- Se pueden crear árboles demasiado complejos que no se generalizan bien a partir de los datos de capacitación. (Esto se conoce como sobreajuste) Mecanismos como la poda son necesarios para evitar este problema (con la excepción de algunos algoritmos como el enfoque de Inferencia Condicional, que no requiere poda).

Random Forest

En lugar de luchar por la división más eficiente en cada ronda de partición recursiva, una técnica alternativa es construir múltiples árboles y combinar sus predicciones para seleccionar un camino óptimo de clasificación. Esto implica una selección aleatoria de preguntas binarias para desarrollar múltiples árboles de decisión diferentes, conocidos como bosques aleatorios. En la industria, a menudo escuchará a las personas referirse a este proceso como 'agregación de *bootstrap'* o 'embolsado'.

La clave para entender los bosques aleatorios es comprender el muestreo *bootstrap*. Hay poco uso compilando cinco o diez modelos idénticos. Debe haber alguna variación y es por eso que el muestreo *bootstrap* se basa en el mismo conjunto de datos, pero extrae una variación diferente de los datos en cada turno. Por lo tanto, en el crecimiento de bosques aleatorios, primero se corren múltiples copias de los datos de entrenamiento a través de cada uno de los árboles. Los resultados de cada árbol son luego comparados y votados para crear un árbol óptimo para producir el modelo final o lo que se conoce como la 'clase final'. Sin embargo, un inconveniente de usar bosques aleatorios es que sacrificamos la

simplicidad visual y la facilidad de interpretación que viene con un solo árbol de decisión y en su lugar devuelve una técnica de caja negra.

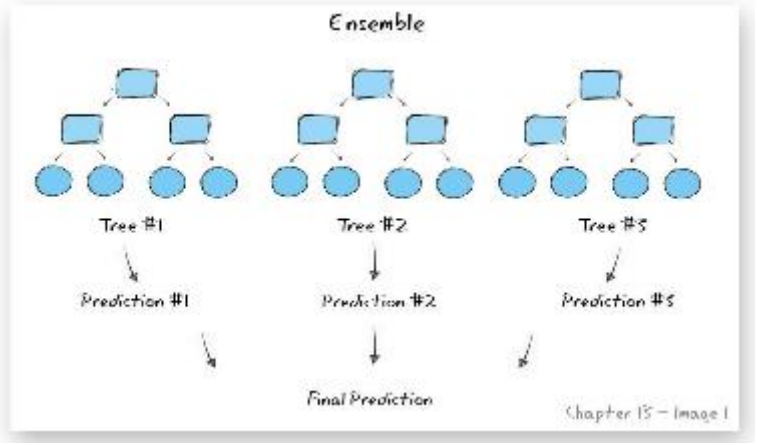


Figura 107 – *Random Forest*
Fuente: [53]

Ventajas

- Los bosques aleatorios tienden a ser más precisos que un solo árbol de decisión. Esto se debe a que reducen el sobreajuste, que es un problema común en los árboles de decisión.
- Los bosques aleatorios tienden a ser más estables que los árboles de decisión. Esta estabilidad significa que los bosques aleatorios a menudo son más consistentes en las predicciones cuando se realizan pequeños ajustes al conjunto de datos de capacitación.
- Los bosques aleatorios son menos susceptibles al impacto negativo de la búsqueda codiciosa porque usan múltiples árboles para predecir. Esto significa que un algoritmo de bosque aleatorio a menudo predecirá con mayor precisión que un algoritmo de árbol de decisión.

Desventajas

- Los bosques aleatorios son más difíciles de comprender e interpretar. Puede ser un desafío entender qué hacen 100 o 1000 de árboles aleatorios.

- Los bosques aleatorios pueden ser muy costosos desde el punto de vista computacional. En otras palabras, se necesita una gran cantidad de poder de computadora para ejecutar de manera suficiente un gran algoritmo de bosque aleatorio. También son más lentos para ejecutarse, lo que significa que lleva más tiempo ver los resultados, realizar ajustes y ejecutar el algoritmo nuevamente.

Gradient Boosting

Otra variante de los árboles de decisión múltiple es la popular técnica del aumento de gradiente. En lugar de seleccionar combinaciones de preguntas binarias al azar, el aumento de gradiente selecciona preguntas binarias que mejorarán la precisión de predicción para cada árbol nuevo. Por lo tanto, los árboles de decisión se desarrollan secuencialmente, ya que cada árbol se crea utilizando información derivada del árbol de decisión anterior. La forma en que esto funciona es que los errores incurridos en los datos de entrenamiento se registran y luego se aplican a la siguiente ronda de datos de entrenamiento. En cada iteración, los pesos se agregan a los datos de entrenamiento en función de los resultados de la iteración anterior. Se aplica una mayor ponderación a las instancias que se predijeron incorrectamente a partir de los datos de entrenamiento y las instancias que se predijeron correctamente recibieron menos peso. Luego, se comparan los datos de entrenamiento y prueba y se vuelven a registrar los errores para informar la ponderación en cada ronda posterior. Las iteraciones anteriores que no funcionan bien y que tal vez se clasifican erróneamente los datos se pueden mejorar mediante iteraciones adicionales. Este proceso se repite hasta que haya un bajo nivel de error. El resultado final se obtiene a partir de un promedio ponderado de las predicciones totales derivadas de cada modelo. Si bien este enfoque mitiga el problema del sobreajuste, lo hace con menos árboles que el enfoque de ensacado. En general, cuantos más árboles agregue a un bosque aleatorio, mayor será su capacidad para frustrar el sobreajuste. Por el contrario, con el aumento de gradiente, demasiados árboles pueden causar un ajuste excesivo y se debe tener precaución a medida que se agrega cada nuevo árbol.

Ventajas

- Manejo natural de datos de tipo mixto (= características heterogéneas)
- Poder de predicción
- Robustez a valores atípicos en el espacio de salida

Desventajas

- La escalabilidad, debido a la naturaleza secuencial de impulsarlo, difícilmente puede ser paralelizada.

***Adaptive Boosting* (AdaBoost)**

Funciona eligiendo un algoritmo base (por ejemplo, árboles de decisión) y mejorándolo iterativamente al tomar en cuenta los casos incorrectamente clasificados en el conjunto de entrenamiento.

En AdaBoost se asignan pesos iguales a todos los ejemplos de entrenamiento y se elige un algoritmo base. En cada paso de iteración, se aplica el algoritmo base al conjunto de entrenamiento y aumenta los pesos de los ejemplos incorrectamente clasificados. Itera n veces, cada vez aplicando el algoritmo base en el conjunto de entrenamiento con pesos actualizados. El modelo final es la suma ponderada de los resultados de los n algoritmos base. AdaBoost en conjunto con árboles de decisión se ha mostrado sumamente efectivo en varios problemas de *Machine Learning*.

Ventajas

- Muy simple de implementar
- Selección de características que da como resultado un clasificador relativamente simple
- Buena generalización

Desventajas

- Solución subóptima

- Sensible a datos ruidosos y atípicos

Naive Bayes

Los clasificadores Naive Bayes están basados en métodos de clasificación bayesianos. Estos se basan en el teorema de Bayes, que es una ecuación que describe la relación de las probabilidades condicionales de cantidades estadísticas. En la clasificación Bayesiana, nos interesa encontrar la probabilidad de que una etiqueta tenga algunas características observadas. El teorema de Bayes nos dice cómo expresar esto en términos de cantidades que podemos calcular más directamente:

$$P(L \mid \text{features}) = \frac{P(\text{features} \mid L)P(L)}{P(\text{features})}$$

Si estamos tratando de decidir entre dos etiquetas, llamémoslas L1 y L2, entonces una forma de tomar esta decisión es calcular la razón de las probabilidades posteriores para cada etiqueta:

$$\frac{P(L_1 \mid \text{features})}{P(L_2 \mid \text{features})} = \frac{P(\text{features} \mid L_1)}{P(\text{features} \mid L_2)} \frac{P(L_1)}{P(L_2)}$$

Todo lo que necesitamos ahora es algún modelo mediante el cual podamos calcular P (*features* | L) para cada etiqueta. Tal modelo se denomina modelo generativo porque especifica el proceso aleatorio hipotético que genera los datos. Especificar este modelo generativo para cada etiqueta es la parte principal del entrenamiento de dicho clasificador bayesiano. La versión general de dicho paso de capacitación es una tarea muy difícil, pero podemos simplificarla mediante el uso de algunas suposiciones sobre la forma de este modelo. Aquí es donde entra el "ingenuo" en "ingenuo Bayes": si hacemos suposiciones muy ingenuas sobre el modelo generativo para cada etiqueta, podemos encontrar una aproximación del modelo generativo para cada clase, y luego proceder con la clasificación bayesiana. Los diferentes tipos de clasificadores Bayes ingenuos se

basan en diferentes suposiciones sobre los datos, y examinaremos algunos de ellos a continuación:

Gaussian Naive Bayes

Quizás el clasificador ingenuo de Bayes más fácil de entender es el gaussiano. En este clasificador, la suposición es que los datos de cada etiqueta se extraen de una distribución Gaussiana simple.

Una forma extremadamente rápida de crear un modelo simple es suponer que los datos se describen mediante una distribución gaussiana sin covarianza entre las dimensiones. Este modelo se puede ajustar simplemente encontrando la media y la desviación estándar de los puntos dentro de cada etiqueta, que es todo lo que necesita para definir dicha distribución. El resultado de esta suposición gaussiana ingenua se muestra en la siguiente figura:

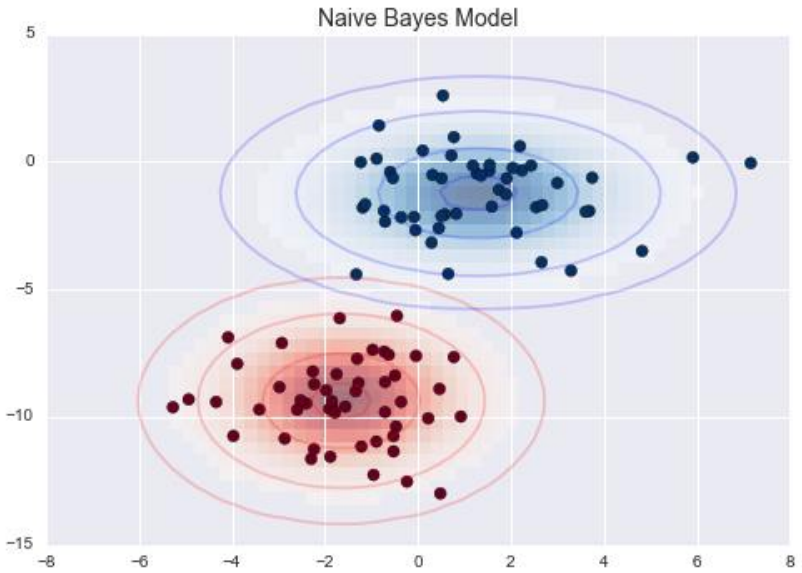


Figura 108 – Gaussian Naive Bayes
Fuente: [55, Fig. 5.39]

Las elipses aquí representan el modelo generativo gaussiano para cada etiqueta, con una mayor probabilidad hacia el centro de las elipses. Con este modelo

generativo para cada clase, tenemos una receta simple para calcular la probabilidad P (*features* | L) para cualquier punto de datos, y así podemos calcular rápidamente la relación posterior y determinar qué etiqueta es la más probable para un punto dado.

Multinomial **Naive Bayes**

La suposición gaussiana recién descrita no es de ninguna manera la única suposición simple que podría usarse para especificar la distribución generativa para cada etiqueta. Otro ejemplo útil es el ingenuo *multinomial* Bayes, donde se supone que las características se generan a partir de una distribución *multinomial* simple. La distribución *multinomial* describe la probabilidad de observar recuentos entre una serie de categorías, por lo que Bayes ingenuo *multinomial* es más apropiado para las características que representan recuentos o tasas de recuento. La idea es exactamente la misma que antes, excepto que en lugar de modelar la distribución de datos con el gaussiano más adecuado, modelamos la distribución de datos con una distribución *multinomial* óptima.

Bernoulli Naive Bayes

Bernoulli Naive Bayes se usa en los datos que se distribuyen de acuerdo con las distribuciones de Bernoulli multivariantes. Es decir, múltiples características pueden estar allí, pero se supone que cada una es una variable de valores binarios (Bernoulli, booleana). Por lo tanto, requiere que las características sean binarias.

Ventajas

- Debido a que los clasificadores bayesianos ingenuos hacen suposiciones tan estrictas sobre los datos, generalmente no funcionarán tan bien como un modelo más complicado.
- Dicho esto, tienen varias ventajas:
    - Son extremadamente rápidos tanto para el entrenamiento como para la predicción.
    - Proporcionan una predicción probabilística directa.
    - A menudo son muy fáciles de interpretar.

    - Tienen muy pocos parámetros (si los hay).
- Estas ventajas significan que un clasificador bayesiano ingenuo suele ser una buena opción. como una clasificación de referencia inicial.
- Si funciona adecuadamente, felicitaciones: tiene un clasificador muy rápido y muy interpretable para su problema. Si no funciona bien, entonces puede comenzar a explorar modelos más sofisticados, con algunos conocimientos básicos de lo bien que deben funcionar.
- Los clasificadores Naive Bayes tienden a funcionar especialmente bien en una de las siguientes situaciones:
    - Cuando las suposiciones ingenuas coinciden con los datos (muy raros en la práctica)
    - Para categorías muy bien separadas, cuando la complejidad del modelo es menos importante
    - Para datos de muy alta dimensión, cuando la complejidad del modelo es menos importante
- Los dos últimos puntos parecen distintos, pero en realidad están relacionados: a medida que crece la dimensión de un conjunto de datos, es mucho menos probable que dos puntos se encuentren próximos (después de todo, deben estar cerca en cada dimensión única para estar cerca en general). Esto significa que los *clusters* en grandes dimensiones tienden a estar más separados, en promedio, que los *clusters* en las dimensiones bajas, suponiendo que las nuevas dimensiones en realidad agregan información. Por esta razón, los clasificadores simplistas como el ingenuo Bayes tienden a funcionar tan bien o mejor que los clasificadores más complicados a medida que crece la dimensionalidad: una vez que tienes suficientes datos, incluso un modelo simple puede ser muy poderoso.

Desventajas

- No sirve para problemas de regresión
- Se hacen varias suposiciones sobre la distribución y la independencia de los datos.

Neural Networks

Una red neuronal, también conocida como red neuronal artificial, es un tipo de algoritmo de aprendizaje automático inspirado en el cerebro biológico.

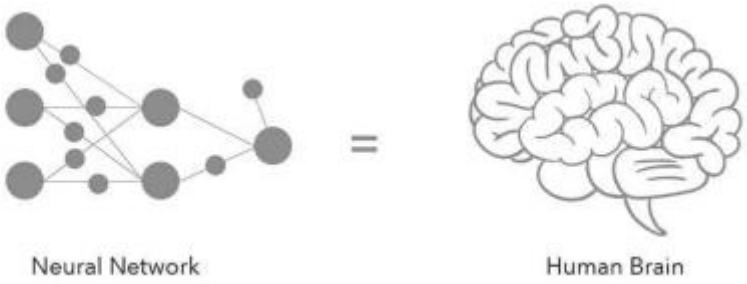


Figura 109 – Redes neuronales
Fuente: [43]

Las redes neuronales son parte de lo que se llama *Deep Learning*, que es una rama del aprendizaje automático que ha demostrado ser valiosa para resolver problemas difíciles, como el reconocimiento de cosas en imágenes y el procesamiento del lenguaje.

Características de las redes neuronales

- Las redes neuronales siempre se crean para resolver un tipo específico de problema
- Una red neuronal tiene tres secciones básicas, o partes, y cada parte está compuesta de "nodos"
    - Capa de entrada (*Input Layer*)
    - Capa(s) oculta(s) (*Hidden Layer*(s))
    - Capa de salida

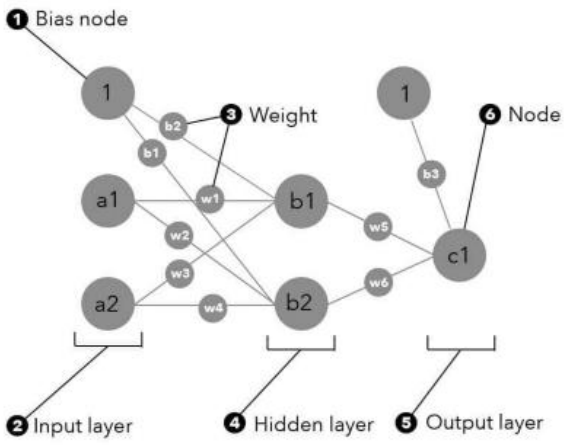


Figura 110 – Estructura de una red neuronal
Fuente: [43]

- Las redes neuronales se construyen de dos maneras:
  - ***Feedforward***: con una red neuronal *feedforward*, las señales viajan solo de una manera, desde la entrada hasta la salida. Este tipo de redes son más sencillas y se usan mucho en el reconocimiento de patrones. Una red neuronal convolucional (CNN o ConvNet) es un tipo específico de red de *feedforward* que a menudo se utiliza en el reconocimiento de imágenes.
  - ***Feedback* (o redes neuronales recurrentes, RNN):** con una RNN, las señales pueden viajar en ambas direcciones y puede haber bucles. Las redes de comentarios son más poderosas y complejas que las CNN, y siempre están cambiando. A pesar de esto, las RNN han sido menos influyentes que las redes de *feedforward*, en parte porque los algoritmos de aprendizaje para redes recurrentes son (al menos hasta la fecha) menos potentes. Sin embargo, las RNN son extremadamente interesantes y tienen un espíritu mucho más cercano a la forma en que funcionan nuestros cerebros que las redes de *feedforward*.

Funcionamiento de la red neuronal

- **Etapa 1 - *Forward Propagation***

Cuando la entrada ingresa a la red, se mueve de una capa a la siguiente hasta que pasa a través de la capa de salida. Un total de dos funciones matemáticas se usan repetidamente para hacer que todo esto sea posible.

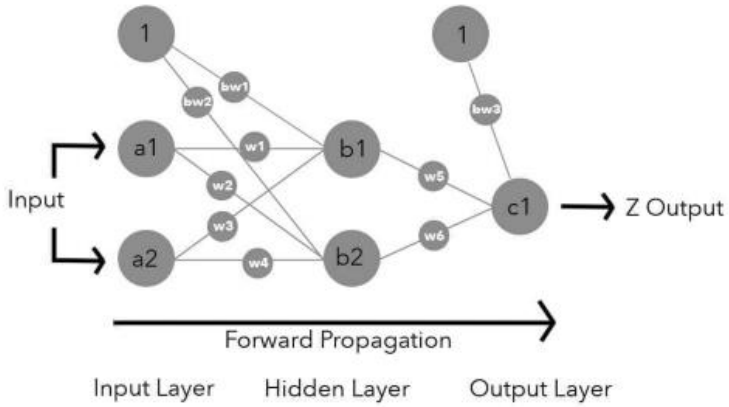


Figura 111 – Propagación hacia adelante
Fuente: [43]

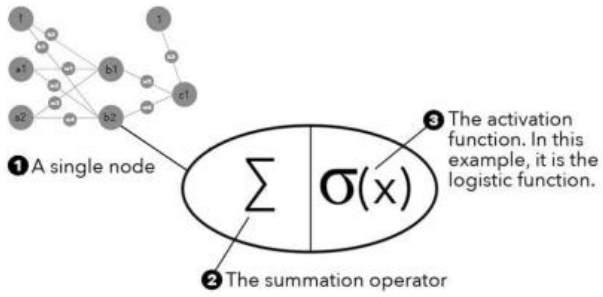


Figura 112 – Operadores de un nodo de la red
Fuente: [43]

Dentro de una red neuronal, el operador (de suma) suma todas las entradas de un nodo para crear una entrada neta.

Cuando se trata de calcular la entrada neta de un nodo, el operador de suma se ve de la siguiente manera:

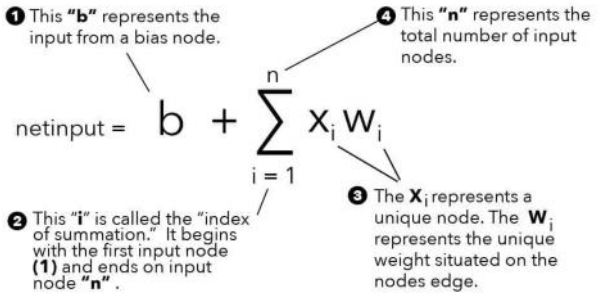


Figura 113 – Cálculo de valor de entrada neto
Fuente: [43]

Una función de activación recibe la salida del operador de suma y la transforma en la salida final de un nodo. En un nivel alto, una función de activación esencialmente "aplasta" la entrada y la transforma en un valor de salida que representa cuánto debe contribuir un nodo (es decir, cuánto debe disparar un nodo). A modo de ejemplo, a continuación, puede ver el gráfico de la función de activación logística (también llamada Sigmoid), que aplasta su entrada para crear una salida entre 0 (cero) y 1.

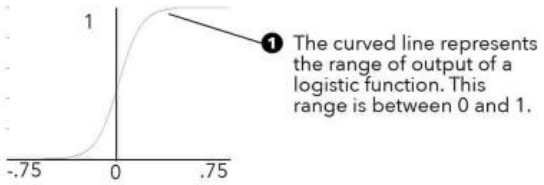


Figura 114 – Función de activación
Fuente: [43]

Hay muchos tipos de funciones de activación para elegir, y una única red a menudo hará uso de múltiples tipos. Por ejemplo, una red puede usar la función logística para los nodos de capa oculta, pero usar una función diferente para los nodos de salida, como la función de softmax. Esto se debe a que las funciones difieren en rendimiento y también tienen fortalezas y debilidades únicas.

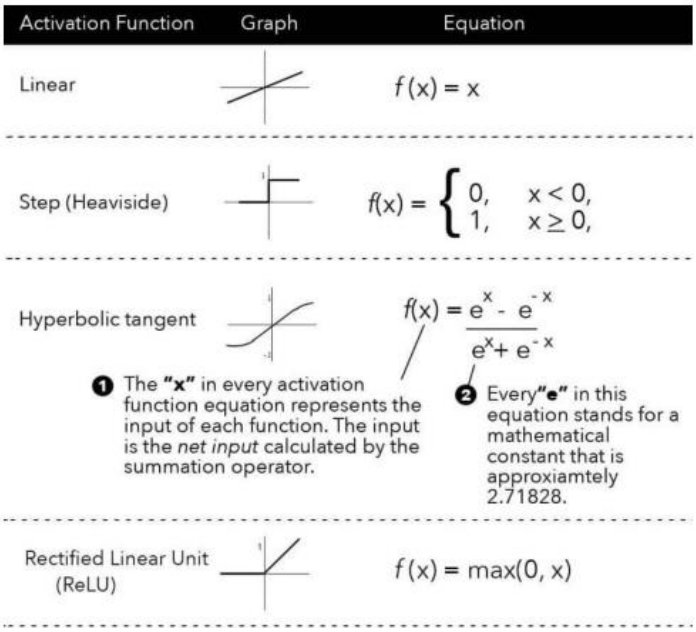


Figura 115 – Tabla de funciones de activación
Fuente: [43]

- **Etapa 2 - Cálculo del error total**

  El siguiente paso es calcular el error total de la red, lo que permitirá a la red ajustar sus pesos y aprender. El error total es la diferencia entre el rendimiento real de una red y el valor objetivo.

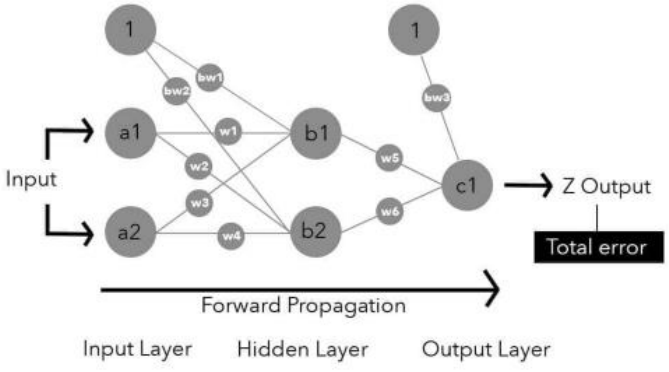


Figura 116 – Error total de la red
Fuente: [43]

Dependiendo de dónde se encuentre la red en el proceso de capacitación, esta etapa podría ser la etapa final. Sin embargo, siempre será la etapa final para cualquier red que entrene con éxito. Una vez que se completa esta etapa y se calcula el error total, pueden ocurrir dos cosas: la red ha

convergido exitosamente, es decir, ha alcanzado un error aceptablemente bajo y ha alcanzado un mínimo global o un mínimo local aceptable que es lo suficientemente cercano al global mínimo. En este punto, la red deja de entrenar.

La red no ha logrado converger. En este caso, no se ha descubierto un mínimo, y la red continúa en la Etapa 3 y continúa entrenando. Este proceso se repite hasta que la red converge.

- **Función de costo**

  Dentro de una red neuronal, una función de costo transforma todo lo que ocurre dentro de la red en un número que representa el error total de la red. Básicamente, es una medida de cuán equivocada es una red. En un nivel más técnico, mapea un evento o valores de una o más variables en un número real. Este número real representa un "costo" o "pérdida" asociado con el evento o los valores.

  Hay muchos tipos de funciones de costos para elegir. Las opciones populares incluyen el error cuadrático medio, el error cuadrado, el error cuadrático medio y la suma de errores cuadrados. Otras funciones de costos incluyen la Entropía Cruzada, Exponencial, Distancia Hellinger y la Divergencia Kullback-Leibler. No profundizaremos en cada función de costos, pero detengámonos para examinar los tres primeros.

  1. ***Mean Squared Error*** **(MSE)**. La función *Mean Squared Error* toma la suma de todos los errores de salida al cuadrado en una red y los promedia. En otras palabras, el MSE mide la diferencia entre la producción objetivo y la producción real de ejemplos de capacitación en una red. La ecuación de MSE es la siguiente:

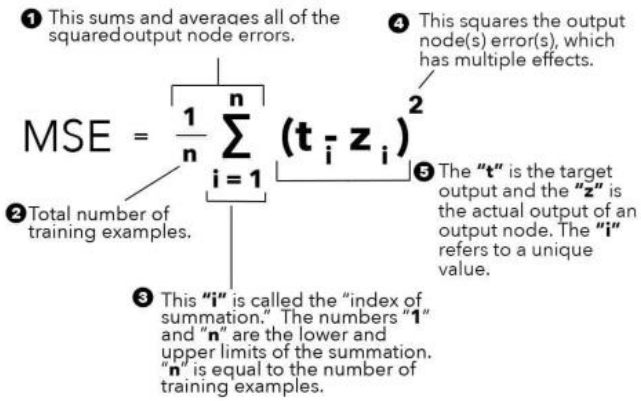


Figura 117 – Error cuadrático medio
Fuente: [43]

2. **Error cuadrado (SE)** - La función de Error al cuadrado es idéntica a la MSE, excepto que se multiplica por 1/2, no 1 / n.

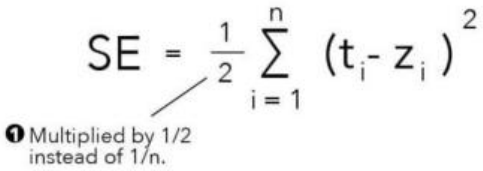


Figura 118 – Error cuadrado
Fuente: [43]

3. ***Root Mean Square*** **(RMS):** *Root Mean Square* realiza los mismos cálculos que el MSE, con la única diferencia de que cuadra la respuesta.

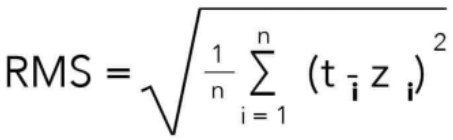


Figura 119 – Media cuadrática
Fuente: [43]

- **Etapa 3 - Cálculo de los gradientes**

  Ahora, vamos a descubrir cómo se extiende este error a través de cada peso en la red para que podamos ajustar los pesos para minimizar el error. Para hacer esto, calcularemos el error de cada peso en la red.

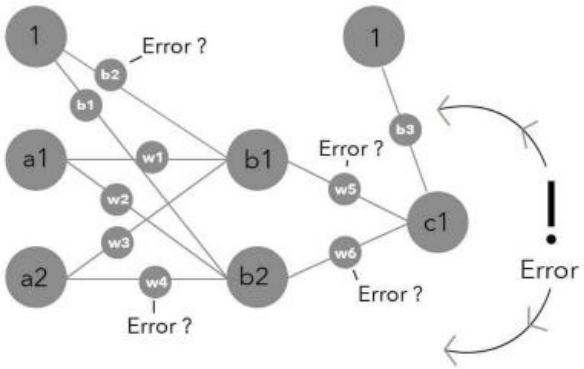


Figura 120 – Cálculo de gradientes
Fuente: [43]

- **Etapa 4 - Verificación de los gradientes**

  La verificación de gradiente es un procedimiento muy simple que permite que los gradientes analíticos que calculamos se comprueben manualmente para determinar su precisión. Esto se logra utilizando una estimación numérica de las derivadas parciales, que puede parecer compleja, pero es bastante sencilla. Una vez que se ha verificado con éxito un gradiente, se desactiva la verificación de gradiente.

- **Etapa 5 - Actualización de los pesos**

  Un gradiente es la pendiente del error local para un peso específico. En otras palabras, es el error individual de un peso específico y nos dice cuánto un cambio en un peso específico afecta el error total.

  El gradiente descendente es un método de optimización que nos ayuda a encontrar la combinación exacta de ponderaciones para una red que minimizará el error de salida. Es la forma en que giramos los diales en nuestra red y los sintonizamos para que estemos satisfechos con su salida.

  Con un descenso gradual, descendemos por la pendiente de los gradientes para encontrar el punto más bajo, que es donde el error es menor. Recuerde, los gradientes son los errores de los pesos individuales,

y el objetivo de una red es minimizar estos errores. Al descender por los gradientes, estamos tratando activamente de minimizar la función de costos y llegar al mínimo global. Nuestros pasos están determinados por la pendiente de la pendiente (el gradiente en sí) y la tasa de aprendizaje. La velocidad de aprendizaje es un valor que acelera o ralentiza la rapidez con la que aprende un algoritmo. Técnicamente, es un conjunto de hiperparámetros en la etapa previa que determina el tamaño del paso que toma un algoritmo cuando se mueve hacia un mínimo global. La mayoría de los casos de estudio que encontrará en línea utilizan tasas de aprendizaje entre 0.0001 y 1.

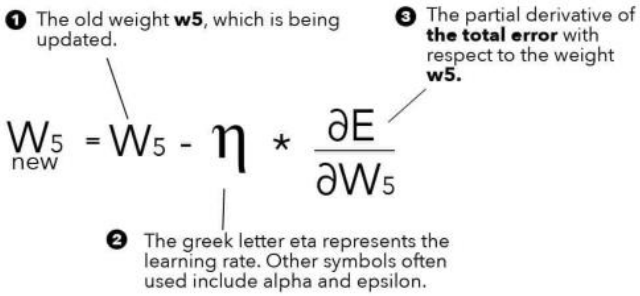


Figura 121 – Reajuste de pesos
Fuente: [43]

- **Ventajas**

  - Una red neuronal puede realizar tareas que un programa lineal no puede
  - Almacenamiento de información en toda la red: la información se almacena en toda la red, no en una base de datos. La desaparición de algunas piezas de información en un solo lugar no impide que la red funcione.
  - Capacidad de trabajar con conocimiento incompleto: después del entrenamiento la red neuronal puede producir resultados incluso con información incompleta. La pérdida de rendimiento aquí depende de la importancia de la información faltante.

- Tener tolerancia a fallas: la corrupción de una o más celdas de la red neuronal no impide que genere salida. Esta característica hace que las redes sean tolerantes a fallas.
- Tener una memoria distribuida: Para que la red neuronal pueda aprender, es necesario determinar los ejemplos y entrenar la red de acuerdo con el resultado deseado mostrando estos ejemplos. El éxito de la red es directamente proporcional a las instancias seleccionadas, y si el evento no se puede mostrar a la red en todos sus aspectos, la red puede producir resultados falsos
- Corrupción gradual: una red se ralentiza con el tiempo y sufre una degradación relativa.
- Capacidad para hacer aprendizaje automático: las redes neuronales artificiales aprenden eventos y toman decisiones en base a eventos similares.
- Capacidad de procesamiento en paralelo: las redes neuronales artificiales tienen una fuerza numérica que puede realizar más de un trabajo al mismo tiempo.

- **Desventajas**

  - Dependencia del hardware: las redes neuronales artificiales requieren procesadores con potencia de procesamiento paralelo, de acuerdo con su estructura. Por esta razón, la realización del equipo es dependiente.

  - Comportamiento inexplicable de la red: este es el problema más importante de ANN. Cuando ANN produce una solución de prueba, no da una pista sobre por qué y cómo. Esto reduce la confianza en la red.

  - Determinación de la estructura de red adecuada: no existe una regla específica para determinar la estructura de las redes neuronales artificiales. La estructura de red adecuada se logra a través de la experiencia y la prueba y error.

- Dificultad para mostrar el problema a la red: las ANN pueden trabajar con información numérica. Los problemas deben traducirse en valores numéricos antes de ser introducidos en la ANN. El mecanismo de visualización que se determinará aquí influirá directamente en el rendimiento de la red. Esto depende de la habilidad del usuario.

- La duración de la red es desconocida: la red se reduce a un cierto valor del error en la muestra significa que la capacitación se ha completado. Este valor no nos da resultados óptimos.

Support Vector Machines

Una máquina de vectores de soporte construye un hiperplano o un conjunto de hiperplanos en un espacio dimensional alto o infinito, que se puede usar para la clasificación, la regresión u otras tareas. Intuitivamente, se logra una buena separación por el hiperplano que tiene la mayor distancia a los puntos de datos de entrenamiento más cercanos de cualquier clase (llamado margen funcional), ya que, en general, cuanto mayor es el margen menor es el error de generalización del clasificador.

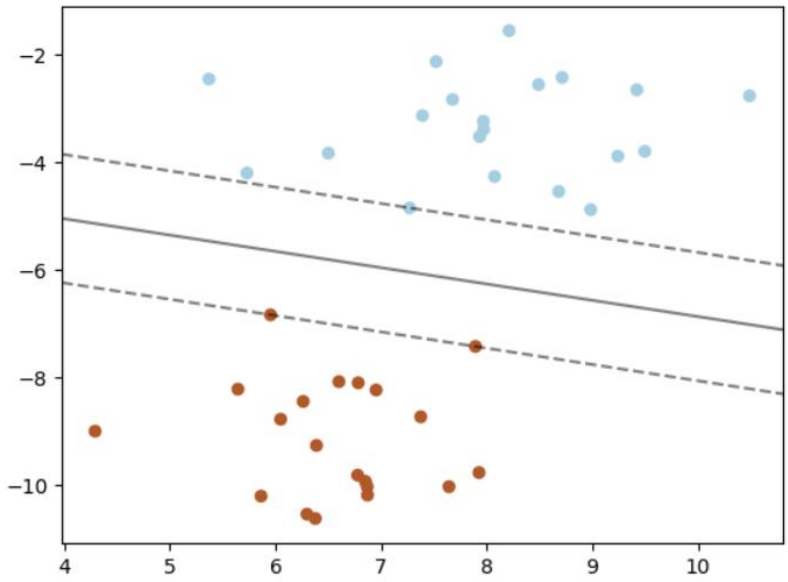


Figura 122 – *Support Vector Machines*
Fuente: [56]

- **Ventajas**

  - Efectivo en espacios de alta dimensión.
  - Sigue siendo efectivo en casos donde el número de dimensiones es mayor que el número de muestras.
  - Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamados vectores de soporte), por lo que también es eficiente desde el punto de vista de la memoria.
  - Versátil: se pueden especificar diferentes funciones del *kernel* para la función de decisión. Se proporcionan núcleos comunes, pero también es posible especificar *kernels* personalizados.

- **Desventajas**

  - Si la cantidad de funciones es mucho mayor que la cantidad de muestras, evite el ajuste excesivo al elegir las funciones *kernel* y el término de regularización es crucial.
  - Las SVM no proporcionan estimaciones de probabilidad directamente, estas se calculan utilizando una costosa validación cruzada de cinco veces

ANEXO 4 – Cartas compromiso y presentación INUMET

UNIVERSIDAD ORT
Uruguay

Facultad de Ingeniería
Bernard Wand-Polak
Cláudio Williman 1451
11.100 Montevideo, Uruguay
Tel 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

CARTA COMPROMISO

Montevideo, 21 de Agosto de 2017.

Dra. (Phd.) Madeline Renom

Presidente
Instituto Uruguayo de Meteorología
Javier Barrios Amorin 1488. Montevideo.

Por medio de esta Carta de Compromiso y con nuestra firma que luce al pie, nosotros Marcelo Barberena (CI 2.988.169-5) y Natalie Gnoza (3.486.952-5) nos comprometemos en nombre de Universidad ORT Uruguay a utilizar los datos suministrados gratuitamente por el Instituto Uruguayo de Meteorología solamente para fines curriculares (Proyecto de Grado), asegurando que los mismos no sean comercializados ni entregados a terceras personas sin previa autorización escrita del Instituto Uruguayo de Meteorología (INUMET).

Nos comprometemos, al mismo tiempo, a dejar registrado en los materiales donde se utilice esta información que dichos datos meteorológicos han sido suministrados por el Instituto Uruguayo de Meteorología.

Natalie Gnoza

Marcelo Barberena

FACULTAD DE INGENIERÍA "BERNARD WAND POLAK" - UNIVERSIDAD ORT Uruguay

Sergio Yovine (Tutor)

UNIVERSIDAD ORT
Uruguay

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

CARTA PRESENTACION

Montevideo, 21 de Agosto de 2017.

Dra. (Phd.) Madeline Renom

Presidente
Instituto Uruguayo de Meteorología
Javier Barrios Amorin 1488. Montevideo.

Por medio de esta carta, se pone en conocimiento que Marcelo Barberena (CI 2.988.169-5) y Natalie Gnoza (CI 3.486.952-5), estudiantes ambos de Universidad ORT Uruguay se encuentran cursando el proyecto de grado de la carrera Ingeniería de Sistemas bajo la tutoría del Ph.D. Sergio Yovine.

El proyecto tiene como fin realizar un trabajo de Investigación + Desarrollo para crear un prototipo (herramienta predictiva) que pueda ser útil para mejorar las predicciones meteorológicas en Uruguay.

A partir de variables como Temperatura, Humedad y Presión Atmosférica se pronosticará con cierto grado de precisión el valor de la variable Precipitación (lluvia).

Por tanto será necesario disponer de la información histórica de las variables mencionadas que, tal como indica la carta de compromiso, los datos proporcionados por el Instituto Uruguayo de Meteorología serán utilizados únicamente con fines académicos.

Saludan atte.

Natalie Gnoza

Marcelo Barberena

FACULTAD DE INGENIERIA "BERNARD WAND POLAK" ORT UNIVERSIDAD ORT Uruguay

Sergio Yovine
TUTOR

INFORME DE LOS CORRECTORES