UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería de Software

Trabajo Fin de Grado

Modelos predictivos aplicados a IoT

<Nombre y Apellidos del Autor>

<Convocatoria, Año>

UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería de Software

Trabajo Fin de Grado

Modelos predictivos aplicados a IoT

Autor: José Luis Pérez García

Tutor: Andrés Caro Lindo

Co-Tutor/es: José Carlos Sancho Núñez

**Tribunal Calificador**

Presidente: <Nombre y Apellidos>

Secretario: <Nombre y Apellidos>

Vocal: <Nombre y Apellidos>

**RESUMEN**

**ÍNDICE GENERAL DE CONTENIDOS**

[1 INTRODUCCIÓN 7](#_Toc105934571)

[2 OBJETIVOS 8](#_Toc105934572)

[3 ESTADO DEL ARTE 9](#_Toc105934573)

[**3.1** **MACHINE LEARNING** 9](#_Toc105934574)

[**1.1** **DISPOSITIVOS IoT** 14](#_Toc105934575)

[**3.2** **ANTECEDENTES DE MACHINE LEARNING APLICADO A IOT** 16](#_Toc105934576)

[Caso 1 16](#_Toc105934577)

[Caso 2 18](#_Toc105934578)

[Caso 3 20](#_Toc105934579)

[Caso 4 21](#_Toc105934580)

[4 METODOLOGÍA 23](#_Toc105934581)

[5 IMPLEMENTACIÓN Y DESARROLLO 24](#_Toc105934582)

[5.1 XGBoost 24](#_Toc105934583)

[5.2 Linear Regression Múltiple 24](#_Toc105934584)

[5.3 Linear Regression Múltiple Stats Model 24](#_Toc105934585)

[5.4 Ramdom Forest 24](#_Toc105934586)

[5.5 Support Vector Regression 24](#_Toc105934587)

[5.6 Logistic Regression 24](#_Toc105934588)

[5.7 Stochastic Gradient Descent 24](#_Toc105934589)

[5.8 LASSO Regression 24](#_Toc105934590)

[5.9 Gausian Naïve Bayes 25](#_Toc105934591)

[5.10 Decision Tree Classifier 25](#_Toc105934592)

[5.11 Robust Regression RANSAC 25](#_Toc105934593)

[6 RESULTADOS Y DISCUSIÓN 26](#_Toc105934594)

[7 CONCLUSIONES 27](#_Toc105934595)

[8 REFERENCIAS BIBLIOGRÁFICAS 28](#_Toc105934596)

[Anexo 1: Instalación del entorno 30](#_Toc105934597)

[Anexo 2: Librerías usadas 33](#_Toc105934598)

**ÍNDICE DE FIGURAS**

[Figura 1: Beneficios del Machine Learning en el ámbito empresarial. Iberdrola 10](#_Toc105625520)

[Figura 2: Metodología usada en el caso 1 16](#_Toc105625521)

**ÍNDICE DE ILUSTRACIONES**

[Ilustración 1: Resultados RMSE y MAE del Caso 2 (3) 18](#_Toc105875044)

[Ilustración 2: Resultados R2 del Caso 2 (3) 19](#_Toc105875045)

[Ilustración 3: Resultados MSE, RMSE, MAE y RMAE del Caso 3 20](#_Toc105875046)

[Ilustración 4: Metodología del caso 4 21](#_Toc105875047)

**ÍNDICE DE TABLAS**

[Tabla 1: Modelos y algoritmos usados en el caso 1 16](#_Toc105875048)

# INTRODUCCIÓN

# OBJETIVOS

# ESTADO DEL ARTE

Para proceder a realizar un análisis del proyecto realizado, antes se va a introducir en que consiste el Machine Learning y en que consiste IoT. Posteriormente, se hará un resumen de otros estudios que se han realizado anteriormente en cuanto al trabajo conjunto de ambos.

## **MACHINE LEARNING**

A día de hoy podemos encontrar muchos ejemplos en Machine Learning en nuestro día a día, pero desconocemos que lo es, por ejemplo, coches que conducen solos (son ya muy famosos los coches de la marca Tesla), asistentes que traducen instantáneamente de un idioma a otro (la cámara en el traductor de Google) o sugerencias de compra personalizadas (por ejemplo el mail que nos llega de Amazon con el mensaje “Te podría interesar”).

**¿Qué es el Machine Learning?**

El Machine Learning o Aprendizaje Automático se conoce por ser una disciplina del campo de la Inteligencia Artificial que mediante el uso de algoritmos, proporciona a los ordenadores la capacidad de poder identificar patrones en grandes conjuntos de datos y con ello poder elaborar predicciones. A esto también se le conoce como análisis predictivo. Este aprendizaje dota a los ordenadores de la posibilidad de poder realizar tareas específicas de forma autónoma. Así, de esta manera no sería necesario que un usuario los programe. (1)

El término Machine Learning fue usado por primera vez en el 1959. No obstante hasta en estos últimos años no se ha empezado a escuchar de verdad. Esto ha sido debido al gran aumento en la capacidad de computación y al uso de los conjuntos masivos de datos. De hecho, en el mundo del Big Data el Machine Learning es una parte fundamental.

**Algoritmos de Machine Learning**

Los algoritmos de Machine Learning se dividen en tres categorías, siendo las dos primeras las más usadas:

Aprendizaje supervisado: este tipo de algoritmos necesitan un aprendizaje previo basado que está basado en un sistema de etiquetas asociadas a datos que permiten poder tomar decisiones o hacer predicciones. Por ejemplo, un detector de spam que se usa para etiquetar un mail como spam hace uso patrones que ha aprendido del histórico de correos (remitente, relación entre texto e imágenes, palabras clave en el asunto, etc.).

Aprendizaje no supervisado: estos algoritmos no requieren de un conocimiento previo. Analizan los datos (incluso desordenados) con el objetivo de encontrar patrones que permitan organizarlos de alguna forma. Por ejemplo, en marketing es muy usado para extraer patrones de grandes conjuntos de datos que provienen de las redes sociales y así poder crear campañas de publicidad altamente segmentadas.

Aprendizaje por refuerzo: el objetivo de este tercer tipo de algoritmos es que este aprenda de la experiencia. De forma que sea capaz de tomar la mejor decisión ante diferentes situaciones mediante un proceso de prueba y error en el que se recompensan las decisiones correctas. Por ejemplo en la actualidad lo podemos encontrar en reconocedores faciales, clasificación de secuencias de ADN o diagnósticos médicos.

**Beneficios del Machine Learning en el ámbito empresarial**

En el siguiente gráfico se pueden ver los beneficios que las empresas obtienen las al usar Machine Learning, además después de dicho gráfico se puede ver una explicación de cada cosa.

Figura 1: Beneficios del Machine Learning en el ámbito empresarial. [Iberdrola](https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico)

Impulsa la innovación: El aprendizaje automático se sirve de los datos masivos, los analiza y busca nuevas soluciones.

Reduce costes: Dado que con el aprendizaje automático se pueden automatizar tareas, esto hace que se las empresas puedan ahorrar capital humano. Además les permite optimizar tiendas gracias a los datos de navegación y flujo de clientes.

Mejora la relación con el cliente: Tener un Chatbot online las 24 horas, los 7 días de la semana respondiendo permite recoger datos para profundizar en el conocimiento del consumidor

Mejora la segmentación de anuncios: Con los algoritmos de Machine Learning se pueden predecir los contenidos que son más eficaces para cada objetivo.

Mejora la segmentación del público objetivo: Gracias a los algoritmos de aprendizaje no supervisado se pueden encuentrar algunos patrones en los datos que recompilan las empresas sobre los consumidores.

Predice tendencias: Mediante un análisis de los hábitos de compra, los algoritmos de Machine Learning pueden predecir los productos que tendrán más demanda y en que momento se pueden subir o bajar precios.

**Aplicaciones prácticas del Machine Learning**

El Machine Learning es uno de los fundamentos básicos de la transformación digital. Actualmente se está utilizando para el desarrollo de nuevas soluciones en diferentes campos, entre los que podemos encontrar:

Recomendaciones: le permite por ejemplo poder hacer sugerencias personalizadas de compra en algunas plataformas online. La forma más sencilla de explicar cómo funciona esto sería la siguiente: se encarga de analizar nuestro historial de compras y lo compara con las tendencias de compra de otros usuarios con gustos iguales o parecidos.

Vehículos inteligentes: actualmente como ya se señalaba anteriormente en el documento, ya podemos encontrar algunos prototipos de la marca Tesla con funciones de conducción autónoma. Además también, gracias al aprendizaje automático, estos vehículos podrán ajustar las configuraciones internas como son por ejemplo la temperatura, música, inclinación del respaldo, etc. Esto de acuerdo a las preferencias del conductor e, incluso, mover el volante solos para reaccionar a cualquier situación extrema que surja del entorno.

Redes sociales: Twitter, por ejemplo, se sirven de algoritmos de Machine Learning para reducir de forma drástica el spam que se publica por los usuarios de esta red. Mientras Facebook, lo usa para poder detectar noticias falsas o contenidos no permitidos en retransmisiones de videos en directo, los cuales son bloqueados en el acto de forma automática.

Procesamiento de Lenguaje Natural: mediante la comprensión de la voz humana, los asistentes virtuales como Alexa, Google o Siri pueden traducir de forma instantánea un idioma a otro. Además, pueden reconocer la voz del usuario e incluso analizar sus sentimientos.

Búsquedas: los buscadores como por ejemplo Google o Bing usan el aprendizaje automático para mejorar sus resultados en función de su eficacia, midiendo esta, a través de los clics de los usuarios.

Medicina: se está utilizando el Machine Learning para detectar en la mayor brevedad y con la mayor antelación el cáncer de mama, ya que su detección temprana aumenta las probabilidades de curación También se utiliza con una alta eficacia para detectar neumonía y algunas enfermedades de la retina que pueden provocar ceguera en el futuro.

Ciberseguridad: los nuevos ya usan el Machine Learning para mejorar el escaneado, de esta forma se acelera la detección y se mejorara la habilidad de reconocer anomalías.

## **DISPOSITIVOS IoT**

IoT, Internet de las cosas o Internet of Things está muy presente en la actualidad, y cada vez hay más fabricantes que crean dispositivos de este tipo.

**¿Qué es IoT o Internet de las cosas?**

IOT es una red de **interconexión digital entre dispositivos**, personas e Internet que hace posible compartir datos, lo que permite que se pueda capturar información sobre el uso y el rendimiento de los dispositivos para así poder detectar patrones y hacer por ejemplo recomendaciones con las que mejorar la experiencia del usuario.

Por ejemplo, IOT es la conexión entre tu smartphone y los dispositivos Smart que tienes en casa que pueden ser: un aire acondicionado o una Raspberry Pi (que controla la programación de tu televisor) o también puede ser un sensor de temperatura (que puede almacenar datos históricos de la temperatura de un lugar).

El termino IOT se empieza a usar cuando los dispositivos (no solo ordenadores) empiezan acceder a la red, para obtener información que necesitan y así poder dar sus servicios.

**Cómo funciona el IoT**

Los dispositivos IoT se conectan entre si con un proceso llamado Machine to Machine (máquina a máquina). En este proceso, los dos dispositivos se comunican entre sí utilizando cualquier tipo de conectividad como por ejemplo Wifi o Bluetooth. De esta manera se consigue que puedan realizar un trabajo sin necesidad de que haya un usuario de por medio.

Una vez se han conectado los dispositivos, estos generan gran cantidad de datos que se almacenan en una plataforma IoT, la cual recolecta y procesa y posteriormente analiza los datos. Gracias a estos datos el usuario puede sacar conclusiones de hábitos y preferencias de él mismo.

**Influencia de los dispositivos IoT en nuestra vida actual**

Durante nuestro día a día nos podemos encontrar con una enorme cantidad de dispositivos que forman parte del Internet de las cosas. Se van a describir a continuación algunos ejemplos:

Vehículos autónomos: como ya se habló anteriormente en la sección de Machine Learning, cada vez se tiene más presente el futuro de los coches con conducción autónoma (lo cual, no es otra cosa que coches que conducen solos). Conforme pasa el tiempo, estos tienen más tecnología, y se basan en el uso de sensores para poder realizar dicha conducción autónoma.

Robots aspiradora: estos tienen gran cantidad de sensores que les permiten realizar una limpieza del sin chocarse con ningún obstáculo.

Smart home: los dispositivos smart pueden ser desde sensores de presencia que encienden las luces automáticamente hasta sensores de temperatura o humedad (son en estos en los que se ha basado el estudio) que recogen constantemente datos que luego pueden ser analizados por el usuario.

## **ANTECEDENTES DE MACHINE LEARNING APLICADO A IOT**

En este punto se van a describir algunos artículos académicos publicados en la web de trabajos previos en los que se ha usado un sistema de Machine Learning aplicado a dispositivos IoT. Todos los artículos se basan en datos relacionados con el clima (contaminación del aire, radiación del sol y temperaturas). Se ha comprobado que son teóricos y no tienen una aplicación o implementación donde ver resultados en tiempo real. Existen diferentes tecnologías con las que alcanzar buenos resultados, pero se puede afirmar que todos los artículos dan como útil el uso del Machine Learning aplicado a IoT.

Caso 1**: Predicción de la energía solar diaria para sistemas fotovoltaicos con una distribución de tiempo variable** (2)

En este artículo se realizó un estudio para conseguir un pronóstico lo más preciso de la producción de energía en sistemas fotovoltaicos. Para ello se usaron 12 algoritmos diferentes. En la siguiente tabla se muestra cada modelo con los algoritmos que incluye:

|  |  |
| --- | --- |
| **Modelo** | **Algoritmos** |
| Linear Regression | Multivariate Linear Regression (MLR), Least Absolute Shrinkage and Selection Operator (LASSO), Seasonal Auto-Regressive Integrated Moving Average with exogenous input variables (SARIMAX) |
| Support Vector Regression | Support Vector Machine (SVM) |
| Ensemble Learning | Random Forest(RF), Gradient Boosting regression (GB) |
| Deep Learning | Artificial Neural Network (ANN), Long Short-Term Memory (LSTM) |
| Physical Models | Physical Model |
| Benchmark Models | Diurnal Persistence (DP), Clear Sky Persistence (CSP) |

Tabla 1: Modelos y algoritmos usados en el caso 1

Como punto de partida, la metodología que se siguió se puede ver en el siguiente gráfico:

Figura 2: Metodología usada en el caso 1

Se usó un conjunto de datos o Dataset que contenía las horas de claridad diarias desde Febrero de 2014 a Febrero de 2017.

El Dataset a parte de lo mencionado, también tenía los siguientes datos, considerados como atributos: presión del aire, presión media al nivel del mar, temperatura ambiente, temperatura en el punto de rocío, precipitación, velocidad del viento por zona, velocidad del viento meridional, variación de la nubosidad, cielo despejado (GHI), radiación solar, ángulo cenital, ángulo acimutal y seno y coseno de la hora del día.

Una vez se siguieron los pasos que se indican en metodología de la **Figura 2**, se obtuvo el siguiente resultado:

* Perspectiva técnica: los resultados fueron favorables. Se pudo comprobar que los modelos basados en Ensemble Learning (ANN y LSTM) fueron los que más bajo Mean Absolute Error (MAE) consiguieron, seguidos de los modelos de Deep Learning (RF y GB) que también obtuvieron un buen pronóstico.
* Perspectiva económica: los resultados fueron negativos. El modelo que mejor funcionó fue el Modelo Físico, seguido de los modelos de Deep Learning (RF y GB).

Caso 2**: Predicción de la polución del aire en ciudades inteligentes mediante algoritmos de Machine Learning** (3)

Este artículo se realizó en Murcia, y tuvo como finalidad realizar una predicción del nivel de ozono en la región. Para ello se usaron 6 algoritmos diferentes que fueron: Bagging, Random Committe, Random Forest, Decision Tree, K Nearest Neightbors, y Hieralchical cluster.

Como punto de partida, la metodología que se siguió es la misma que la de **Figura 2**

Se usaron cinco conjuntos de datos obtenidos en diferentes puntos o ciudades. Los datos contenían los siguiente atributos: el promedio por hora de elementos químicos (NO, NO2, SO2, NOx, PM10, C6H6, tolueno (C7H8), xileno (XIL)) y los parámetros climáticos: temperatura, humedad relativa, dirección, velocidad del viento, presión atmosférica y radiación solar. Todos estos datos se recogieron cada día durante los años 2013–2014. En la siguiente tabla se recoge los resultados obtenidos para cada Dataset y algoritmo:

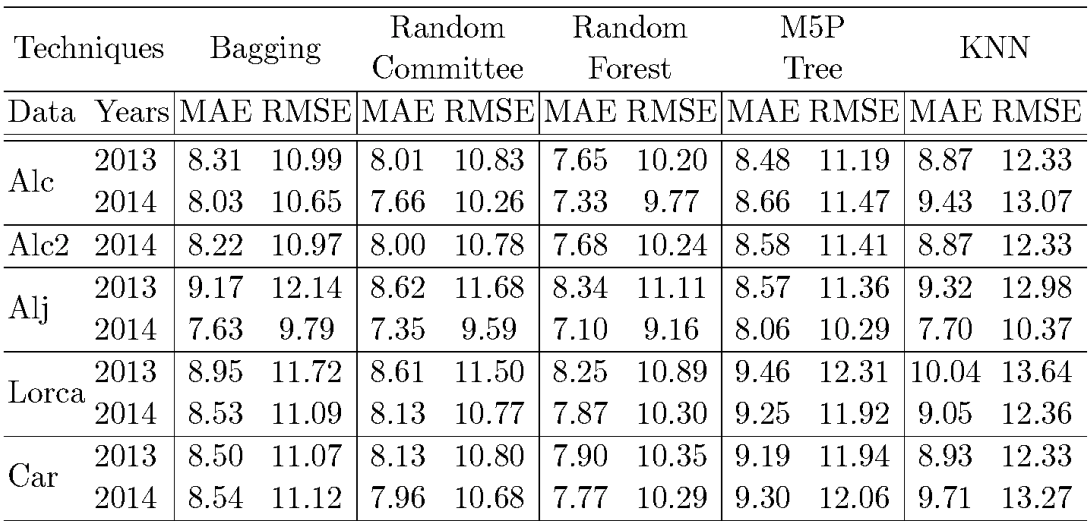


Ilustración 1: Resultados RMSE y MAE del Caso 2 (3)

Como se puede observar en la tabla de arriba, el algoritmo que dio mejor resultado fue el Random Forest ya que tenía el RMSE y MAE más bajos.

En cuando a la relación con R2, se estableció un umbral de 0,75 y se consideró como buen resultado todo el que lo sobrepasara. En la siguiente tabla podemos ver los resultados:

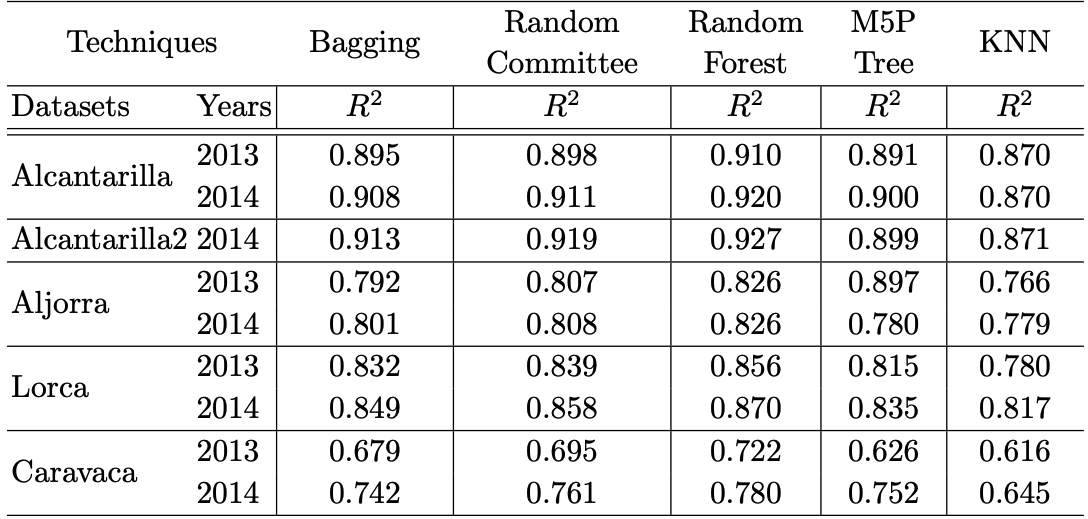


Ilustración 2: Resultados R2 del Caso 2 (3)

En este caso, lo sobrepasaron todos los algoritmos, ya que todos dieron un R2 por encima del 0,80.

Caso 3**: Comparación de modelo XGBoost con otros modelos para predecir energía eólica a corto plazo** (4)

En este tercer artículo, se realizó un estudio para conseguir una predicción de la energía eólica que se puede generar teniendo en cuenta los problemas que surgen debido al método de obtención. Uno de estos principales problemas a los que se enfrenta esta energía es a la aleatoriedad del viento, esto producido por variedad de factores como el terreno, la estación del año (no es lo mismo verano que invierno), la presión del aire, la temperatura, etc.

El estudio se basa en la comparación de los resultados de un algoritmo propio (creado por las mismas personas que realizan el estudio) usando como base XGBoost, con otros algoritmos como Random Forest (RF), Classification and Regression Trees (CART), Back Propagation Neural Network (BPNN), XGBoost y Support Vector Regression (SVR).

El modelo de datos contenía los siguientes atributos: día de la semana (0-6), día del año (0-365), día del mes (1-31), mes del año (1-12), hora del día (0-23), minuto del día (0-1339), valor de la energía eólica de las 24h anteriores, valor de la energía eólica de las 48h anteriores, velocidad del viento, dirección del viento, temperatura, humedad y presión.

El valor que se quiere predecir es la velocidad del viento. En la siguiente tabla, podemos ver los resultados de los algoritmos que se han comparado. Se puede ver que para realizar esta comparación se han usado como indicadores MSE, RMSE, MAE y RMAE.

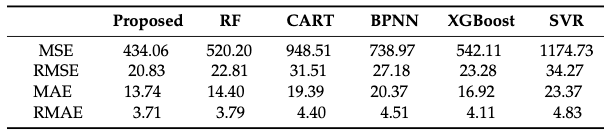


Ilustración 3: Resultados MSE, RMSE, MAE y RMAE del Caso 3

Se puede ver en la tabla, que el algoritmo que mejor resultado ha conseguido es el que el articulo ha propuesto, consiguiendo valores más bajos en todos los indicadores.

Caso 4**: Predicción del tiempo utilizando técnicas de aprendizaje automático** (5)

En este último artículo se realizó un estudio para conseguir una predicción de la temperatura ambiental.

La metodología que se usó para llevar a cabo este estudio fue la que se puede ver la siguiente imagen.

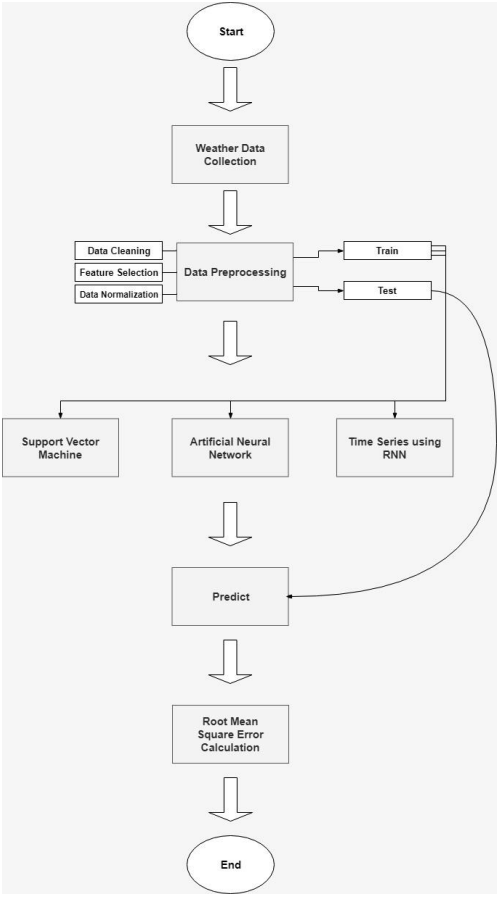


Ilustración 4: Metodología del caso 4

Se puede ver en la imagen, que lo primero que se hace es recolectar datos relacionados con el clima con lo necesario para poder tener un modelo de datos sólido y así conseguir una buena predicción. Posteriormente, realizan el preprocesado de los datos que consiste en la limpieza de datos no válidos, selección de los mejores features o atributos y normalización de los datos. Después de esto, estos datos se le pasan a los algoritmos, en este caso, los que usan son Support Vector Machine (SVM), Artificial Neural Network (ANN) y Time Series using RNN.

Con respecto al conjunto de datos, los atributos que contenía son los siguientes: temperatura del aire a 2 metros de altura sobre la superficie terrestre, presión atmosférica a nivel de estación meteorológica, presión atmosférica reducida al nivel medio del mar, humedad relativa a una altura de 2 metros sobre la superficie terrestre, dirección media del viento a una altura de 10-12 metros sobre la superficie de la tierra, nubosidad total, visibilidad horizontal, temperatura del punto de rocío a una altura de 2 metros sobre la superficie de la tierra. Este conjunto de datos contiene datos desde 2006 a 2018.

Después del análisis de todos los modelos, se puede ver en la siguiente tabla los resultados usando como referencia el indicador de error RMS:

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Tiempo de predicción** | **RMS** |
| Support Vector Machine (SVM) | 8 semanas | 6,67 |
| Artificial Neural Network (ANN) | 8 semanas | 3,1 |
| Time Series using RNN | 8 semanas | 1,41 |

Fijándonos en la tabla podemos ver que el algoritmo que mejor resultado dio fue Time Series using RNN, con error cuadrático medio o RMS de 1,41.

# METODOLOGÍA

# IMPLEMENTACIÓN Y DESARROLLO

En esta sección se van a describir los algoritmos usados para hacer las predicciones.

## XGBoost

¿Qué es XGBoost?

Conocido como eXtreme Gradient Boosting, se trata de una librería o algoritmo

## Linear Regression Múltiple

<https://realpython.com/linear-regression-in-python/>

## Linear Regression Múltiple Stats Model

https://realpython.com/linear-regression-in-python/

## Ramdom Forest

penalized linear regression

## Support Vector Regression

M5p -> Predicion tree -> <https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/>

Gradient Boosting -> xgboost -> revisar proyecto telecoming

random forest -> https://towardsdatascience.com/random-forest-in-python-24d0893d51c0

## Logistic Regression

SVM -> <https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python>

Para regresión: <https://medium.com/pursuitnotes/support-vector-regression-in-6-steps-with-python-c4569acd062d>

## Stochastic Gradient Descent

Stochastic Gradient Descent -> <https://scikit-learn.org/stable/modules/sgd.html>

## LASSO Regression

eARTH/bagEARTH (py-earth) -> http://mehdidc.github.io/pyearth-doc/

## Gausian Naïve Bayes

Naïve Bayes -> <https://scikit-learn.org/stable/modules/naive_bayes.html>

## Decision Tree Classifier

https://scikit-learn.org/stable/modules/generated/sklearn.isotonic.IsotonicRegression.html

## Robust Regression RANSAC

<https://machinelearningmastery.com/make-predictions-scikit-learn/>

# RESULTADOS Y DISCUSIÓN

# CONCLUSIONES

# REFERENCIAS BIBLIOGRÁFICAS

1. **Iberdrola.** Descubre los principales beneficios del Machine Learning. [Online] 2019. https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico.

2. **Visser, Lennard, AlSkaif, Tarek y van Sark, Wildfried.** Elveiser.com. [En línea] 2022. https://reader.elsevier.com/reader/sd/pii/S0960148121015688?token=280D13A6F4479E4B92F62354AF166484F7451982D4F5B6F488C28139DAFC90A4CD62B2552EBE9D7905CDD16085AA1BFB&originRegion=eu-west-1&originCreation=20220605172019.

3. **Martínez, Raquel, y otros.** www.jucs.org. [En línea] 2018. https://www.jucs.org/jucs\_24\_3/air\_pollution\_prediction\_in/jucs\_24\_03\_0261\_0276\_espana.pdf.

4. **Zheng, Huan y Wu, Yanghui.** www.researchgate.net. [En línea] https://www.researchgate.net/publication/334711620\_A\_XGBoost\_Model\_with\_Weather\_Similarity\_Analysis\_and\_Feature\_Engineering\_for\_Short-Term\_Wind\_Power\_Forecasting.

5. **Singh, Siddharth, y otros.** deliverypdf.ssrn.com. [En línea] https://deliverypdf.ssrn.com/delivery.php?ID=190081025071014118094007117025066121042037029042091050024118019086002125092082010075126060062111031022046118098094122106010098049015010026004066095115099094071080108023077048127111007017087072100003093115020102.

**Libros**

Burkov, Andriy, 2019: *The hundred-page machine learning book*. ISBN 9781999579517

<https://realpython.com/linear-regression-in-python/>

<http://webapplayers.com/>

<https://wpbeaches.com/installing-configuring-apache-on-macos-using-homebrew/>

# Anexo 1: Instalación del entorno

Para poder ejecutar nuestro proyecto, vamos a necesitar instalar apache como servidor. Pero, ¿qué es apache? Apache es un software de servidor web gratuito y de código abierto para plataformas Unix(Linux, Mac por ejemplo) con el cual se ejecutan el 46% de los sitios web de todo el mundo. Es mantenido y desarrollado por la Apache Software Foundation.

Le permite a los propietarios de sitios web o a un particular en su propio servidor o pc servir contenido en la web o de forma local, de ahí el nombre de servidor web. Es uno de los servidores web más antiguos y confiables. Su primera versión fue lanzada hace más de 20 años, en 1995.

Cuando alguien quiere visitar un sitio web, ingresa un nombre de dominio en la barra de direcciones de su navegador. Luego, el servidor web envía los archivos solicitados actuando como un repartidor virtual.

Sabiendo que es Apache, proseguimos con los pasos para realizar la instalación, en esta documentación el proyecto se ha realizado sobre un Mac, con lo cual los pasos para la instalación van a ser en lenguaje de consola de Mac OS

El comando de instalación que vamos a usar es:

**brew install httpd**

Una vez instalado vamos a indicar en el archivo httpd.conf cuál va a ser la ruta que debe abrir.

Primero abrimos el archivo para editarlo con el siguiente comando:

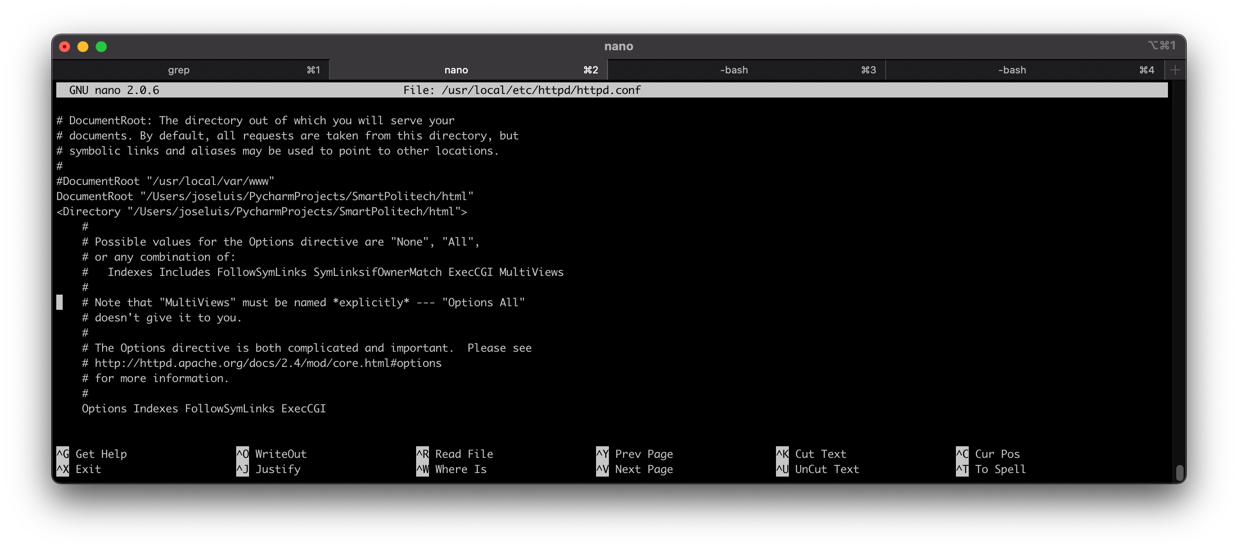
**nano /usr/local/etc/httpd/httpd.conf**

Bajamos hasta la linea donde pone:

# DocumentRoot: The directory out of which you will serve your

# documents. By default, all requests are taken from this directory, but

# symbolic links and aliases may be used to point to other locations.



Justo debajo de esta linea indicaremos donde se encuentra nuestra carpeta con el main principal. Por defecto aparecerá #DocumentRoot “/usr/local/var/www”, con lo cual, tendríamos que cambiar “/usr/local/var/www” por nuestra ruta. En mi caso será “/Users/joseluis/PycharmProjects/SmartPolitech/html”, quedando la linea así:

**#DocumentRoot “/Users/joseluis/PycharmProjects/SmartPolitech/html”**.

Una vez hecho esto, debajo de esta linea hay otra que también debemos modificar:

<Directory “/usr/local/var/www”> y haremos lo mismo, sustituiremos “/usr/local/var/www” por nuestra ruta, en mi caso quedando así:

**<Directory “/Users/joseluis/PycharmProjects/SmartPolitech/html”>**

Hecho esto, bajamos un poco hasta donde encontraremos el fragmento de código siguiente:

<IfModule dir\_module>

DirectoryIndex index.html

</IfModule>

Si nuestro main, se llama index.html, lo dejaremos como está, sin por el contrario se llama de otra forma, deberemos cambiarlo. En mi caso se llama main.html, por lo que mi fragmento de código quedaría así:

<IfModule dir\_module>

DirectoryIndex main.html

</IfModule>

Una vez hemos terminado, pulsamos control + X para guardar los cambios y salir del archivo. Ya tenemos configurado el archivo de configuración para poder ejecutar nuestro servidor local con el proyecto.

Ahora tenemos que levantar el servidor, para ello usaremos los siguientes comandos:

Inicio de httpd: **brew services start httpd**

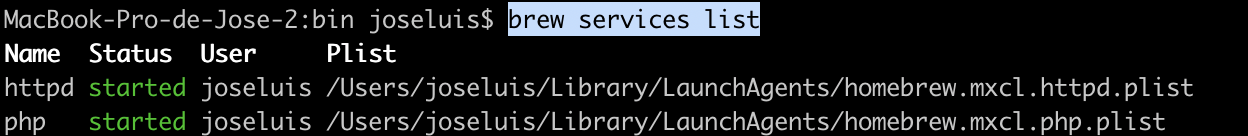
Interrupción de httpd: **brew services stop httpd**

Reinicio de httpd: **brew services restart httpd**

Para comprobar que se ha iniciado el servidor podemos usar el siguiente comando

**brew services list**

Este comando mostrará todos los servicios iniciados en el sistema y su estado.

****

Cómo podemos ver en la captura anterior tenemos servicio httpd en estado started.

# Anexo 2: Librerías usadas

Durante el Proyecto, se han usado varias librerías en los diferentes lenguajes con lo que se ha programado:

Librería XGBoost -> se instala con el comando pip3 install xgboost

Librería Sklearn -> se instala con el comando pip3 install sklearn

Librería Statsmodels -> se instala con el comando pip3 install statsmodels

Librería Bartpy -> se instala con el comando pip3 install bartpy

# 