e

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](#_Toc106477188)

[2 OBJETIVOS 8](#_Toc106477189)

[3 ESTADO DEL ARTE 9](#_Toc106477190)

[3.1 MACHINE LEARNING 9](#_Toc106477191)

[3.2 DISPOSITIVOS IoT 12](#_Toc106477192)

[3.3 ANTECEDENTES DE MACHINE LEARNING APLICADO A IOT 14](#_Toc106477193)

[Caso 1 14](#_Toc106477194)

[Caso 2 16](#_Toc106477195)

[Caso 3 18](#_Toc106477196)

[Caso 4 19](#_Toc106477197)

[4 METODOLOGÍA 21](#_Toc106477198)

[5 IMPLEMENTACIÓN Y DESARROLLO 24](#_Toc106477199)

[5.1 Recolección de datos 24](#_Toc106477200)

[5.2 Selección de Features o atributos para el modelo de entrenamiento 26](#_Toc106477201)

[5.3 Algoritmos implementados 27](#_Toc106477202)

[5.4 Inicialización del modelo 32](#_Toc106477203)

[5.5 Entrenamiento del modelo 33](#_Toc106477204)

[5.6 Predicción de datos 33](#_Toc106477205)

[5.7 Creación de la base de datos 34](#_Toc106477206)

[6 RESULTADOS Y DISCUSIÓN 34](#_Toc106477207)

[7 CONCLUSIONES 35](#_Toc106477208)

[8 REFERENCIAS BIBLIOGRÁFICAS 36](#_Toc106477209)

[Anexo 1: Instalación del entorno 38](#_Toc106477210)

[Anexo 2: Librerías usadas 41](#_Toc106477211)

**ÍNDICE DE FIGURAS**

[Figura 2: Metodología usada en el caso 1 15](#_Toc106440758)

[Figura 3: Metodología del proyecto 21](#_Toc106440759)

**ÍNDICE DE ILUSTRACIONES**

[Ilustración 1: Resultados RMSE y MAE del Caso 2 (3) 16](#_Toc106477177)

[Ilustración 2: Resultados R2 del Caso 2 (3) 17](#_Toc106477178)

[Ilustración 3: Resultados MSE, RMSE, MAE y RMAE del Caso 3 18](#_Toc106477179)

[Ilustración 4: Metodología del caso 4 19](#_Toc106477180)

[Ilustración 5: Representación de predicción usando XGBoost 23](#_Toc106477181)

[Ilustración 6: Fragmento de código del proceso de Web Scrapping 24](#_Toc106477182)

[Ilustración 7: Tabla de datos de la web Weather Underground (9) 25](#_Toc106477183)

[Ilustración 8; Fragmento de datos del csv 26](#_Toc106477184)

[Ilustración 9: Fragmento de código de la lectura del CSV 26](#_Toc106477185)

[Ilustración 10: Fragmento de código de la selección de los Features 26](#_Toc106477186)

**ÍNDICE DE TABLAS**

[Tabla 1: Modelos y algoritmos usados en el caso 1 16](#_Toc106039054)

# INTRODUCCIÓN

# OBJETIVOS

# ESTADO DEL ARTE

Para proceder a realizar un análisis del proyecto realizado, previamente 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

Actualmente, 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 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 estos últimos años no se ha empezado a escuchar de verdad. Esto se ha 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.

**¿Cómo funciona el Machine Learning?**

El funcionamiento del Machine Learning se puede resumir de la siguiente manera: el usuario recolecta una serie de datos conocidos como input y partir de estos el ordenador debe ser capaz de encontrar patrones o relaciones entre ellos. Lo que hace el ordenador, una vez tiene el conjunto de datos, es escoger una fórmula matemática que se aplica al conjunto de datos o input de entrenamiento, el cual también pertenece a otro conjunto de datos y devuelve los outputs deseados. Lo realmente interesante es que la misma fórmula puede ser aplicable a otros datos distintos del conjunto de entrenamiento teniendo en cuenta que los nuevos tienen que ser parecidos a los datos de entrenamiento.(2)

El principal objetivo que debe buscar el Machine Learning es la optimización. Nuestra máquina debe ser capaz de escoger el algoritmo que mejor se ajuste y para ello debe elegir la fórmula que relacione los datos y minimice el error, el cual debe ser medido según diferentes métricas dependiendo de las características del modelo.

Las métricas más usadas para medir la efectividad de la predicción de un modelo son: (3)

-MSE/RMSE: mide el error cuadrado promedio de las predicciones del modelo. Por cada punto, se encarga de calcular la diferencia cuadrada entre las predicciones y el objetivo para posteriormente promediar esos valores. Cuanto mayor sea este valor, peor es el modelo. La diferencia entre RMSE y MSE es que la primera es la misma fórmula pero con raíz cuadrada.

-MAE/RMAE: realiza un cálculo como un promedio de diferencias absolutas entre valores objetivo y predicciones. Es una puntuación lineal, es decir, todas las diferencias individuales se ponderan por igual en el promedio. Por ejemplo, la diferencia entre 10 y 0 será el doble de la diferencia entre 5 y 0.

-R^2: está relacionada con la MSE. No importa si los valores de salida son muy grandes o muy pequeños, el R² siempre estará entre -∞ y 1. Cuando R² es negativo, significa que el modelo es peor que predecir la media.

**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, 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 no deseado, hace uso de 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.

## 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 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 sí con un proceso llamado Machine to Machine (máquina a máquina). En este proceso, los dos dispositivos se comunican 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, 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 aspiradoras: estos tienen gran cantidad de sensores que les permiten realizar la limpieza 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, además 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:

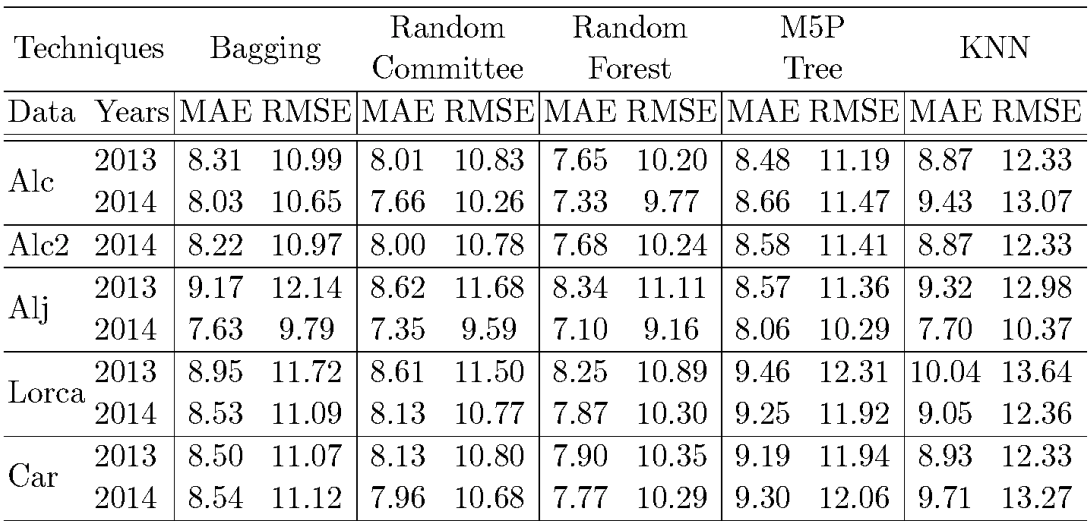


Ilustración 1: Resultados RMSE y MAE del Caso 2 (3)

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 cuanto 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:

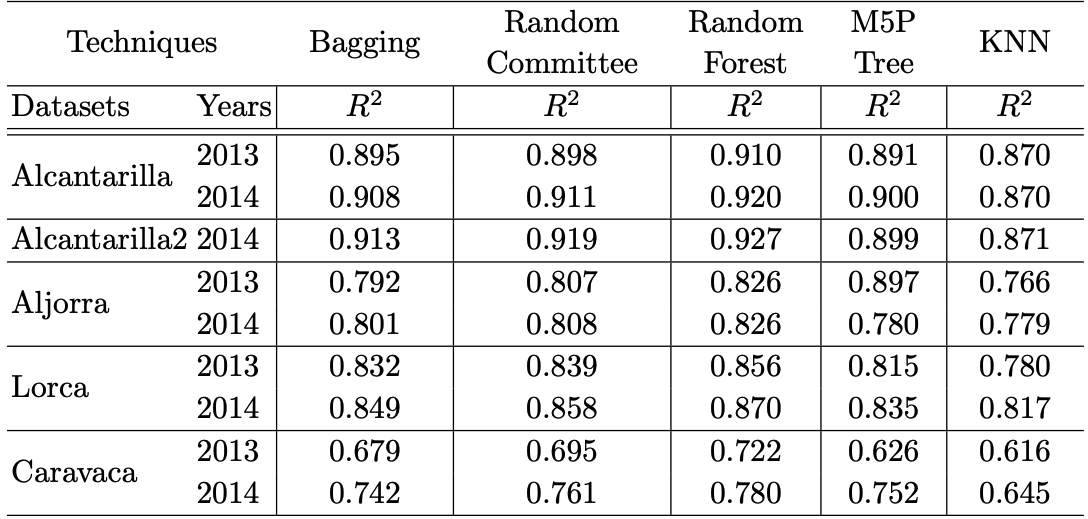


Ilustración 2: Resultados R2 del Caso 2 (3)

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 podría 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 basó 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 quería predecir es la velocidad del viento. En la siguiente tabla, se pueden ver los resultados de los algoritmos que se compararon. Se puede ver que para realizar esta comparación se han usado como indicadores MSE, RMSE, MAE y RMAE.

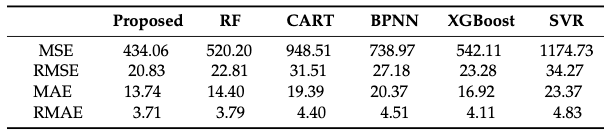


Ilustración 3: Resultados MSE, RMSE, MAE y RMAE del Caso 3

En la tabla se muestra que el algoritmo que mejor resultado consiguió fue el que el articulo propuso, 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.

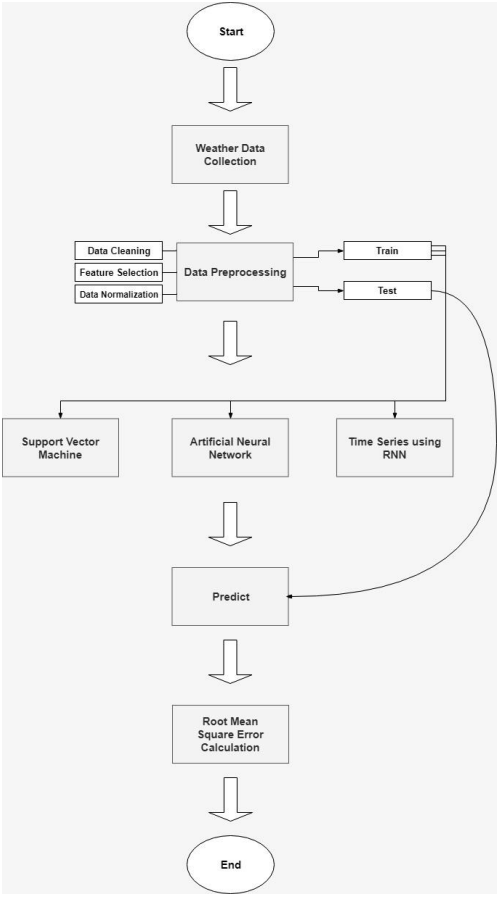


Ilustración 4: Metodología del caso 4

Se puede ver en la imagen, que lo primero que se hizo fue recolectar datos relacionados con el clima con lo necesario para poder tener un modelo de datos sólido y así conseguir una buena predicción. Posteriormente, realizaron 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 pasaron a los algoritmos, en este caso, los que usaron 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

Partiendo de la base de que el significado de Metodología es según Wikipedia *“conjunto de procedimientos racionales utilizados para alcanzar el objetivo”*(6)*,* este proyecto se realiza en función de una serie de procedimientos para llegar a un objetivo final. En el siguiente gráfico se muestra los pasos que se han ido llevando a cabo para el desarrollo de todo el análisis. Posteriormente, se dará una explicación de cada uno de ellos.

Inicio

Creación de base de datos

Recolección de datos

Limpieza de datos inválidos

Selección de los Features o atributos

Preprocesado

Creación Front y Back

División de datos en Train y Test

Inicializar algoritmo

mo

Entrenamiento de datos

Realizar predicción

Creación de modelo

Pintar modelo en Front

Figura 3: Metodología del proyecto

**Parte izquierda del flujo**

En la parte izquierda del flujo se realizan los pasos necesarios para crear un modelo entrenado y poder posteriormente usarlo para obtener predicciones. Los paso son:

-Recolección de datos: el primer paso de la metodología que se ha seguido, ha consistido en realizar un análisis de algunas páginas web para poder recolectar datos y conseguir un modelo sólido que contenga la suficiente información para obtener una buena predicción.

-Preprocesado: en este paso se realizan dos acciones, que son las siguientes:

* Limpieza de datos: en ocasiones, cuando revisas el modelo de datos, encuentras que hay algunos atributos que para algún día no tienen elementos o son incorrectos. Cuando esto ocurre se debe realizar una limpieza, o bien eliminarlos o modificarlos manualmente para que sea un conjunto consistente.
* Selección de los Features o atributos: uno de los requisitos de los algoritmos de Machine Learning es que se le debe indicar cuáles van a ser los atributos o características donde debe buscar el patrón que aprender. Por ejemplo, presión atmosférica, día del mes o mes de año.

-División de datos en Train y Test: en este paso se divide el conjunto de datos en un conjunto llamado Train, que se usa para entrenar el modelo. También, en un conjunto Test, que se utiliza para una vez entrenado el modelo, pasárselo y contractar si los datos que predice son correctos. Normalmente, la división suele ser de un 80% para crear el modelo de entrenamiento y de un 20% para poder contractar los resultados con el conjunto de Test.

-Inicializar algoritmo: en este paso se inicializa el algoritmo, pasándole por parámetros la configuración que necesite.

-Entrenamiento de datos: una vez se ha inicializado el algoritmo, se realiza el entrenamiento, que no es otra cosa que crear un modelo a partir de los datos que se han recolectado y aprende un patrón para poder realizar la posterior predicción.

-Realizar predicción: se le pasa algoritmo un conjunto de datos que no ha sido previamente entrenado y debe devolver un resultado, que tienen que ser comprobado con una métrica de error, por ejemplo Mean Absolute Error (MSE).

-Creación de modelo: se crea un modelo para poder realizar predicciones de datos, cargándolo previamente sin necesidad de hacer un entrenamiento de ellos cada vez que se haga una predicción.

**Parte derecha del flujo**

En la parte derecha del flujo se realizan los pasos para crear un Back y un Front donde poder recrear los datos. Los pasos son:

-Creación de base de datos: este paso consiste en crear una base de datos donde almacenar información, que posteriormente será tratada por el Back y representada por el Front.

-Creación Front y Back: se crea un Back con PHP, donde se realizan todas las llamadas a la base de datos y se ejecutan todas las funciones que se pueden manejar desde el Front, este último creado mediante el uso de HTML, CSS y JavaScript

-Pintar modelo en Front: una vez se han realizado todos los pasos de la metodología desde el JavaScript del Front, se hace un solicitud al PHP del Back, que a su vez hace una solicitud a Python y devuelve una predicción con los datos que ha recogido del formulario del Front. Estos datos se ilustran creando una representación de una línea de tiempo actual y una línea de predicción. En la siguiente imagen se puede ver la representación para un algoritmo XGBoost:

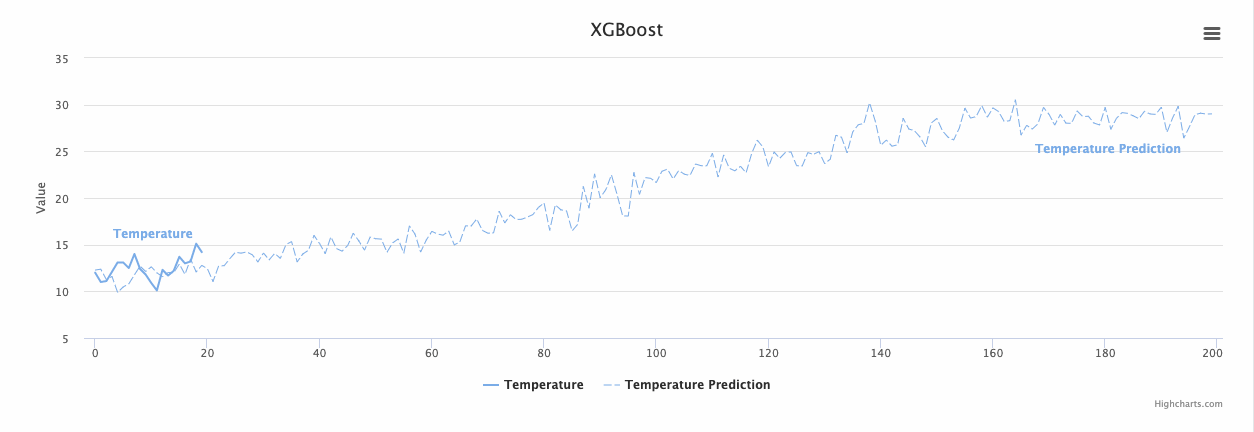


Ilustración 5: Representación de predicción usando XGBoost

# IMPLEMENTACIÓN Y DESARROLLO

En esta sección se va a realizar un análisis de toda la implementación y desarrollo del proyecto. Se explicará cómo se recolectaron los datos, que algoritmos se han usado o que proceso se ha realizado para crear el desarrollo de la interfaz. Además también se irán mostrando pequeños fragmentos de código donde se va creando la funcionalidad.

## Recolección de datos

El primer paso de todos fue recolectar los suficientes datos para crear un Dataset. Este conjunto de datos debe ser enorme ya que cuanto más grande mejor será el aprendizaje. También debe ser limpio, no puede contener errores y en el caso de contenerlos deberán eliminarse o modificarse en un posterior paso de preprocesado.

Para crear el Dataset se extrajeron datos de la página web Weather Underground (9). Para llevar a cabo dicha extracción se realizó un proceso de Web Scrapping[[1]](#footnote-1) mediante el uso del lenguaje de programación Python y su librería Selenium. Cabe destacar que todos estos datos que se recogen en esta página son recogidos mediante el uso de dispositivos IoT.

El intervalo de tiempo para el que se recogieron los datos fue desde el 1 de Enero de 2013 hasta el 31 de Abril de 2022.

En la siguiente imagen podemos ver el fragmento de código principal usado para realizar la extracción.



Ilustración 6: Fragmento de código del proceso de Web Scrapping

La función main() inicializa el web driver y declara mediante una lista de strings los años y los meses que van a ser extraídos.

Posteriormente podemos ver la variable “lista”, esta va a ser la que indique que campos va a tener el Dataset. Estos serán: año, mes, día, temperatura, humedad, velocidad del viento, presión y precipitación.

El funcionamiento del proceso consistirá en ir abriendo página por página para cada mes y año, cambiar la temperatura a Celsius y por ultimo ir obteniendo los datos de la tabla que se muestra a continuación y volcarlos en un CSV.

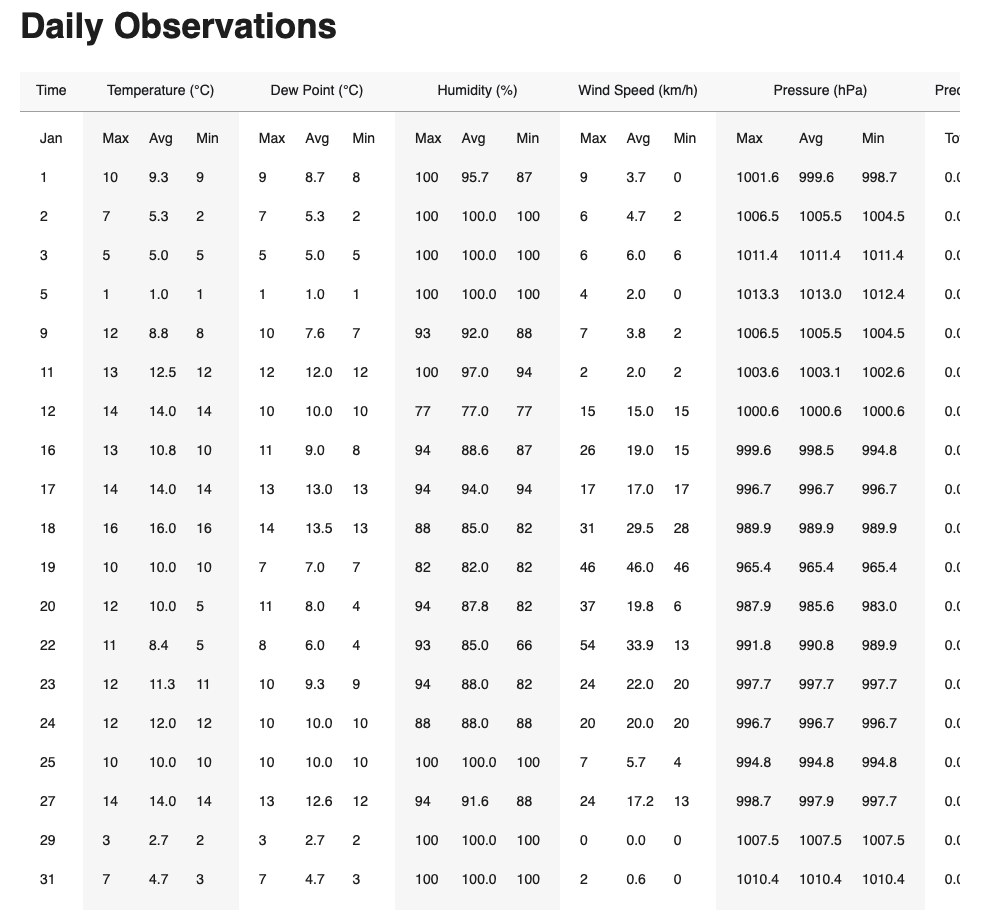


Ilustración 7: Tabla de datos de la web Weather Underground (9)

Todos los campos del csv son de tipo numérico, ya que tanto el campo año, mes y día se han guardado en el CSV como número, siendo el año 2013-2022, el mes 01-12 y el día 1-31. En la siguiente imagen se puede ver un pequeño ejemplo de algunos valores del CSV:

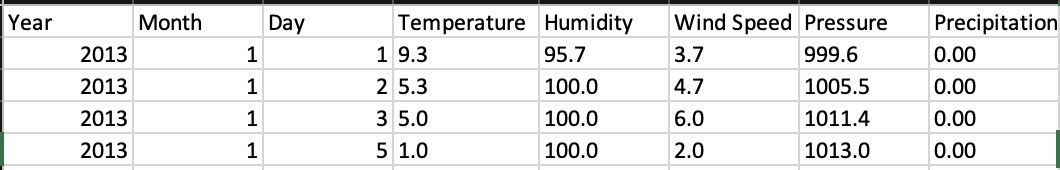


Ilustración 8; Fragmento de datos del csv

En cuanto al código, en la siguiente imagen se puede ver cómo se leen los datos del CSV en una variable y como posteriormente se realiza un eliminado de filas repetidas.

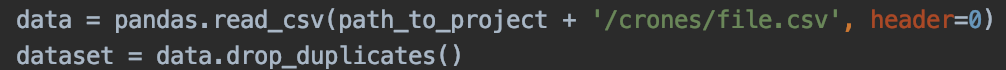


Ilustración 9: Fragmento de código de la lectura del CSV

## Selección de Features o atributos para el modelo de entrenamiento

Una vez se ha creado el CSV con los datos muestrales, se van a seleccionar las Features que se usaran para crear posteriormente el modelo de entrenamiento.

Como se comentaba en el apartado **introducción,** en este proyecto se va a realizar un proceso de Machine Learning para predecir la temperatura ambiental a partir de los datos que se han recogido. Para recordar, estos datos son: año, mes, día, temperatura, humedad, velocidad del viento, presión y precipitación. Con lo cual, se van a usar todos los campos (excepto el campo de temperatura) como Features de los diferentes algoritmos.

En la siguiente imagen se puede ver como se seleccionan las Features mediante código:

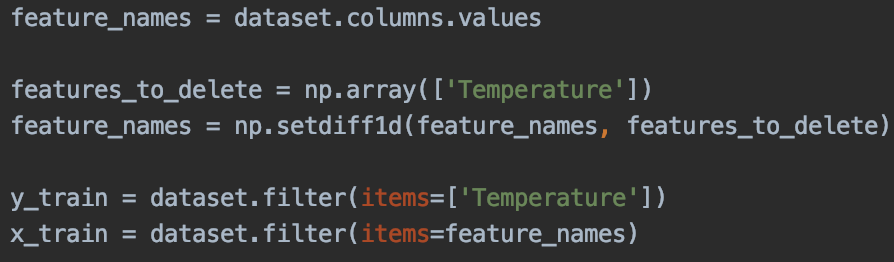


Ilustración 10: Fragmento de código de la selección de los Features

En la primera línea se puede ver que se extrae de la variable “dataset” (creada en la sección anterior) las columnas del CSV.

En la dos posteriores líneas se crean dos variables nuevas “features\_to\_delete” y “feature\_names”. En la primera se guarda un array con el campo temperatura. En la segunda se elimina mediante la función np.setdiff1d[[2]](#footnote-2) el campo temperatura y se guardan los demás.

## Algoritmos implementados

En esta sección se van a describir los algoritmos que se han usado para realizar las predicciones.

**XGBoost** (10)

XGBoost o Extreme Gradient Boosting es uno de los algoritmos de Machine Learning que se clasifica dentro de los de tipo supervisado y es actualmente uno de los más usados. Es bastante conocido por conseguir muy buenas predicciones. Puede igualar o incluso mejorar los resultados de modelos más complejos. Como su propio nombre indica, este algoritmo usa el principio del boosting, que no es otra cosa que ir creando de forma secuencial varios modelos de predicción “débiles”, para que cada uno de esos modelos use los resultados del modelo anterior para generar otro modelo más “fuerte”, con mejores resultados predictivos y con mayor estabilidad en los resultados.

A continuación, se van a mostrar los fragmentos de código más importantes de la librería, que se usan para inicializar el algoritmo, entrenar el modelo y realizar una predicción:

Con los dos siguientes dos comandos, se importa la librería al proyecto y se inicializa el algoritmo:

*import xgboost as xgb*

*xgb.XGBRegressor(“Parámetros de configuración”=””)*

Después de inicializarlo, se realiza el entrenamiento con el comando:

*xg\_reg.fit(“Parámetros”)*

Para finalizar, con el comando de abajo se consigue una muestra de predicción:

*preds = xg\_reg.predict(“Parámetros”)*

Siendo preds el resultado.

**Linear Regression** (11)

El modelo de regresión lineal es probablemente una de las técnicas de regresión más usada en Machine Learning para realizar predicciones, ya que es uno de los métodos más simples. Una de sus principales ventajas es la facilidad con la que se pueden interpretar los resultados.

El funcionamiento consiste en que se busca la relación entre los diferentes puntos de datos y se dibuja una línea recta a través de ellos. Esta línea se puede utilizar para predecir datos futuros.

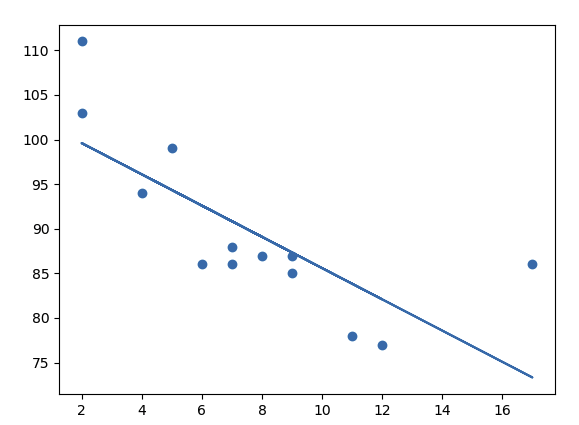
****

Ilustración 11: Ejemplo línea y puntos Regresión Lineal

Para inicializar el algoritmo en Python se usa el siguiente comando:

*from sklearn.linear\_model import LinearRegression*

*lr\_reg = LinearRegression(“Parámetros de configuración”=””)*

Después de inicializarlo, se realiza el entrenamiento con el siguiente comando:

*lr\_reg.fit(“Parámetros”)*

Para finalizar, con el siguiente comando se consigue una muestra de predicción:

*preds = lr\_reg.predict(“Parámetros”)*

Siendo preds el resultado.

**Random Forest**

**Support Vector Regression**

**Logistic Regression**

**Stochastic Gradient Descent**

**LASSO Regression**

**Gausian Naïve Bayes**

**Decision Tree Classifier**

**Robust Regression RANSAC**

## Inicialización del modelo

Para finalizar, en la dos últimas líneas se crean otras dos nuevas variables nuevas, en la primera se guardan todos los elementos de la columna temperatura del Dataset y en la segunda se guardan todos las demás columnas.

## Entrenamiento del modelo

## Predicción de datos

## Creación de la base de datos

## Creación del Front y el Back

## Representación de los datos

# 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. **Burkov, Andriy.** The Hundred-Page Machine Learning Book. 2019.

3. **sitiobigdata.** sitiobigdata.com. [En línea] https://sitiobigdata.com/2018/08/27/machine-learning-metricas-regresion-mse/#:~:text=Ahora%2C%20es%20muy%20importante%20entender,una%20funci%C3%B3n%20que%20no%20disminuye..

4. **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.

5. **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.

6. **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.

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

8. **Wikipedia. [En línea] https://es.wikipedia.org/wiki/Metodolog%C3%ADa.**

**9. Weather Underground. [En línea] https://www.wunderground.com/.**

**10. Wikipedia. [En línea] https://es.wikipedia.org/wiki/Web\_scraping.**

# 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:

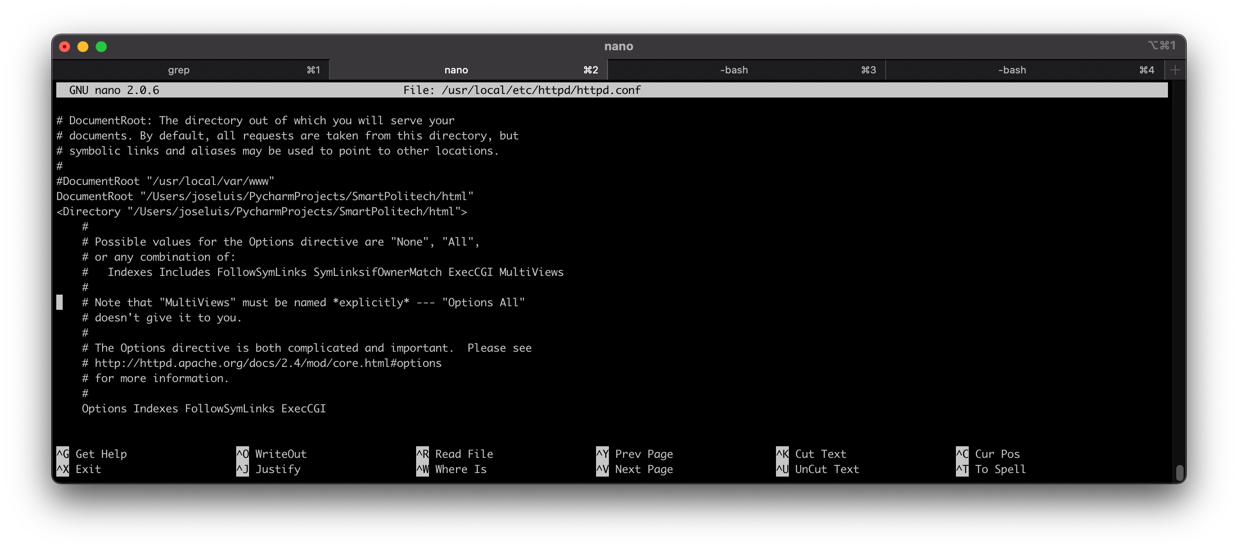
nano /usr/local/etc/httpd/httpd.conf

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 línea se indica 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**

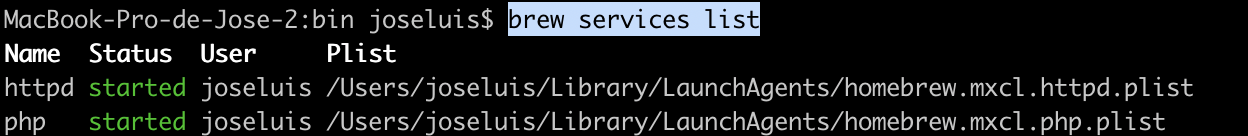
Interrupción de httpd: **brew services stop httpd**

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

# 

1. Técnica utilizada mediante programas de software para extraer información de sitios web. (10) [↑](#footnote-ref-1)
2. Encuentre la diferencia de conjunto de dos arreglos. Devuelve los valores únicos en ar1 que no están en ar2. [↑](#footnote-ref-2)