



UNIVERSIDAD DE JAÉN  
*Escuela Politécnica Superior de Jaén*

## Trabajo Fin de Grado

# **DESARROLLO DE UNA PLATAFORMA WEB PARA EL ANÁLISIS Y PREDICCIÓN DE SERIES TEMPORALES CON DEEP LEARNING**

**Alumno: Rafael Javier Megales Anguita**

Tutor: Prof. D. Antonio Jesús Rivera Rivas

Prof. D<sup>a</sup>. M<sup>a</sup> Dolores Pérez Godoy

Dpto: Informática

**Junio, 2018**





Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Informática

Don Antonio Jesús Rivera Rivas y Doña M<sup>a</sup> Dolores Pérez Godoy , tutores del Proyecto Fin de Carrera titulado: Desarrollo de una plataforma web para el análisis y predicción de series temporales con Deep Learning, que presenta Rafael Javier Megales Anguita, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2018

El alumno:

Rafael Javier Megales Anguita

Los tutores:

Fdo: Antonio Jesús Rivera Rivas

Fdo: M<sup>a</sup> Dolores Pérez Godoy



## Índice de contenido

|       |                                                                    |    |
|-------|--------------------------------------------------------------------|----|
| 1     | Introducción y motivación .....                                    | 11 |
| 2     | Series temporales .....                                            | 15 |
| 2.1   | ¿Qué es una serie temporal? .....                                  | 15 |
| 2.2   | Componentes de las series temporales.....                          | 15 |
| 2.3   | Caracterización de series temporales .....                         | 19 |
| 2.4   | Estudio de la estacionariedad de una serie temporal .....          | 20 |
| 2.5   | Análisis de series temporales.....                                 | 21 |
| 2.6   | Objetivos del análisis de series temporales.....                   | 21 |
| 3     | Predicción de series temporales.....                               | 23 |
| 3.1   | ¿Qué es la predicción de series temporales?.....                   | 23 |
| 3.2   | Ejemplos de predicción de series temporales.....                   | 23 |
| 3.3   | Análisis exploratorio de datos para predicción de series .....     | 24 |
| 3.4   | Preprocesamiento de una serie temporal .....                       | 25 |
| 3.4.1 | Eliminación de la tendencia.....                                   | 25 |
| 3.4.2 | Ajuste de la estacionalidad.....                                   | 26 |
| 3.4.3 | Homogeneización de la varianza .....                               | 26 |
| 4     | Técnicas estadísticas para la predicción de series temporales..... | 28 |
| 4.1   | Media móvil .....                                                  | 28 |
| 4.2   | Autorregresión.....                                                | 29 |
| 4.3   | Autocorrelación .....                                              | 30 |
| 4.4   | ARIMA.....                                                         | 31 |
| 5     | Predicción de series temporales con Deep Learning.....             | 33 |
| 5.1   | Estructuras de las redes neuronales .....                          | 34 |
| 5.2   | Arquitecturas de aprendizaje profundo .....                        | 35 |
| 5.2.1 | Redes convolucionales .....                                        | 35 |
| 5.2.2 | Redes neuronales recurrentes .....                                 | 36 |
| 5.3   | Redes LSTM ( Long Short-Term Memory) .....                         | 37 |
| 5.3.1 | Parámetros de una red LSTM .....                                   | 38 |
| 5.3.2 | Entrenamiento de una red LSTM .....                                | 39 |
| 6     | Análisis.....                                                      | 40 |
| 6.1   | Análisis de requerimientos .....                                   | 40 |
| 6.1.1 | Requerimientos funcionales .....                                   | 40 |

|        |                                               |    |
|--------|-----------------------------------------------|----|
| 6.1.2  | Requerimientos no funcionales .....           | 41 |
| 6.2    | Planificación .....                           | 41 |
| 6.3    | Casos de uso .....                            | 43 |
| 7      | Diseño .....                                  | 46 |
| 7.1    | Preprocesamiento .....                        | 47 |
| 7.1.1  | Homogeneización de datos .....                | 47 |
| 7.1.2  | Valores nulos o erróneos .....                | 47 |
| 7.1.3  | Análisis y limpieza de datos .....            | 48 |
| 7.1.4  | Transformación de los datos .....             | 49 |
| 7.1.5  | Normalización de datos .....                  | 50 |
| 7.2    | Predicción .....                              | 50 |
| 7.2.1  | ARIMA .....                                   | 51 |
| 7.2.2  | LSTM (Long Short-Term Memory) .....           | 52 |
| 7.3    | Obtención y visualización de resultados ..... | 59 |
| 7.3.1  | Obtención y visualización de resultados ..... | 59 |
| 7.3.2  | Resultados mediante gráficos .....            | 59 |
| 7.3.3  | Resultados mediante tablas .....              | 60 |
| 7.3.4  | Resultados mediante archivos csv .....        | 61 |
| 8      | Implementación .....                          | 63 |
| 8.1    | Esquema de la plataforma web .....            | 63 |
| 8.2    | Estructuras de datos usadas .....             | 63 |
| 8.2.1  | DataFrame .....                               | 63 |
| 8.2.2  | Arrays de numpy .....                         | 64 |
| 8.3    | Codificación .....                            | 64 |
| 8.3.1  | Tecnologías usadas .....                      | 64 |
| 8.3.2  | Bibliotecas usadas .....                      | 66 |
| 8.4    | Estructura del proyecto .....                 | 70 |
| 8.5    | Fichero principal .....                       | 70 |
| 9      | Pruebas de la plataforma web .....            | 72 |
| 10     | Manual de usuario .....                       | 77 |
| 10.1   | Iniciar la plataforma .....                   | 77 |
| 10.2   | Registro / Inicio de sesión .....             | 78 |
| 10.3   | Dashboard .....                               | 79 |
| 10.3.1 | Creación de un proyecto .....                 | 79 |
| 10.3.2 | Cargar/Descargar de un proyecto .....         | 79 |
| 10.4   | Imputación y limpieza del proyecto .....      | 80 |

|       |                                                                       |    |
|-------|-----------------------------------------------------------------------|----|
| 10.5  | Análisis del proyecto .....                                           | 81 |
| 10.6  | Ajustar estacionalidad y tendencia .....                              | 83 |
| 10.7  | Transformaciones .....                                                | 83 |
| 10.8  | ARIMA.....                                                            | 83 |
| 10.9  | LSTM Univariante con regresión aplicando diferenciación .....         | 84 |
| 10.10 | LSTM Univariante con regresión aplicando el método de la ventana..... | 84 |
| 10.11 | LSTM Univariante con regresión aplicando pasos de tiempo.....         | 85 |
| 10.12 | LSTM Univariante con memoria entre batches .....                      | 85 |
| 10.13 | LSTM Univariante con número de observaciones estándar .....           | 86 |
| 10.14 | LSTM Univariante con número de observaciones personalizado.....       | 87 |
| 10.15 | LSTM multivariantes .....                                             | 88 |
| 11    | Conclusiones y trabajos futuros .....                                 | 90 |
| 11.1  | Conclusiones.....                                                     | 90 |
| 11.2  | Trabajos futuros .....                                                | 91 |

## Índice de ilustraciones

|                                                                                                                                                                        |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Ilustración 1.1 Número de trabajos de series temporales por área de conocimiento .....                                                                                 | 12 |
| Ilustración 1.2 Número de trabajos de series temporales a lo largo del tiempo .....                                                                                    | 12 |
| Ilustración 2.1 Número mensual de pasajeros de una aerolínea a lo largo del tiempo.....                                                                                | 15 |
| Ilustración 2.2 Componentes de las series temporales.....                                                                                                              | 17 |
| Ilustración 2.3 Número de ventas de un producto de champú a lo largo del tiempo .....                                                                                  | 18 |
| Ilustración 2.4 Series temporales sobre las condiciones climatológicas de Beijing, China....                                                                           | 18 |
| Ilustración 2.5 Ejemplos de series temporales.....                                                                                                                     | 19 |
| Ilustración 3.1 Serie temporal original y serie temporal con diferenciación .....                                                                                      | 26 |
| Ilustración 4.1 Serie temporal en azul y su respectiva media móvil en rojo.....                                                                                        | 28 |
| Ilustración 4.2 Serie temporal y valores esperados .....                                                                                                               | 29 |
| Ilustración 4.3 Serie temporal y su diagrama de autocorrelación .....                                                                                                  | 31 |
| Ilustración 5.1 Diagrama temporal de las redes neurales .....                                                                                                          | 35 |
| Ilustración 5.2 Representación de una celda de memoria en LSTM .....                                                                                                   | 38 |
| Ilustración 6.1 Diagrama de Gantt del proyecto .....                                                                                                                   | 42 |
| Ilustración 6.2 Diagrama de casos de uso .....                                                                                                                         | 43 |
| Ilustración 7.1 Diagrama del sistema completo .....                                                                                                                    | 46 |
| Ilustración 7.2 Valores de un conjunto de datos .....                                                                                                                  | 48 |
| Ilustración 7.3 Características generales de un fichero de datos .....                                                                                                 | 48 |
| Ilustración 7.4 Limpieza de datos .....                                                                                                                                | 49 |
| Ilustración 7.5 Resultados del test de Dickey Fuller para la serie temporal de apertura de los<br>valores de las acciones de APPLE .....                               | 50 |
| Ilustración 7.6 Resultados modelo ARIMA para la serie de tiempo de apertura de los valores<br>de las acciones de APPLE .....                                           | 52 |
| Ilustración 7.7 Resultados LSTM para regresión utilizando el método de la ventana para la<br>serie de tiempo de los valores de apertura de las acciones de APPLE ..... | 54 |
| Ilustración 7.8 Resultados LSTM usando regresión con pasos de tiempo para la serie<br>temporal de las acciones de APPLE.....                                           | 55 |
| Ilustración 7.9 Resultados LSTM usando memoria entre batches para las valores de<br>apertura de las acciones de APPLE .....                                            | 56 |



|                                                                                                                                                                      |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Ilustración 7.10 Resultados LSTM Univariante con instantes de tiempo personalizados para la serie temporal de los valores de apertura de las acciones de APPLE ..... | 57 |
| Ilustración 7.11 Resultados LSTM Multivariante para la serie temporal de los valores de las acciones de APPLE.....                                                   | 58 |
| Ilustración 7.12 Ejemplo de gráficos de la plataforma web .....                                                                                                      | 59 |
| Ilustración 7.13 Tablas que aparecen en la plataforma .....                                                                                                          | 60 |
| Ilustración 7.14 Fichero de salida con valores de muestra y predicciones, así como diversos estimadores.....                                                         | 61 |
| Ilustración 7.15 Conjunto de datos parcial antes y después de limpiar .....                                                                                          | 62 |
| Ilustración 8.1 Estructura del proyecto.....                                                                                                                         | 70 |
| Ilustración 8.2 Ejemplo implementación rutas en Flask .....                                                                                                          | 71 |
| Ilustración 9.1 Dashboard principal de la plataforma web.....                                                                                                        | 72 |
| Ilustración 9.2 Limpieza de datos del fichero de prueba .....                                                                                                        | 74 |
| Ilustración 9.3 Gráficos sobre nuestro fichero de prueba.....                                                                                                        | 74 |
| Ilustración 9.4 Vista de análisis de la plataforma web.....                                                                                                          | 75 |
| Ilustración 9.5 Vista de predicción de la plataforma web .....                                                                                                       | 76 |
| Ilustración 10.1 Iniciando la plataforma .....                                                                                                                       | 77 |
| Ilustración 10.2 Vista inicial de la plataforma.....                                                                                                                 | 78 |
| Ilustración 10.3 Vista registro en la plataforma .....                                                                                                               | 78 |
| Ilustración 10.4 Vista inicio de sesión en la plataforma.....                                                                                                        | 78 |
| Ilustración 10.5 Dashboard principal.....                                                                                                                            | 79 |
| Ilustración 10.6 Eliminación de columnas innecesarias .....                                                                                                          | 80 |
| Ilustración 10.7 Estadísticas generales del fichero .....                                                                                                            | 80 |
| Ilustración 10.8 Imputación y limpieza de un fichero.....                                                                                                            | 81 |
| Ilustración 10.9 Ejemplo vista análisis .....                                                                                                                        | 82 |
| Ilustración 10.10 Ejemplo vista análisis 2 .....                                                                                                                     | 88 |



## 1 Introducción y motivación

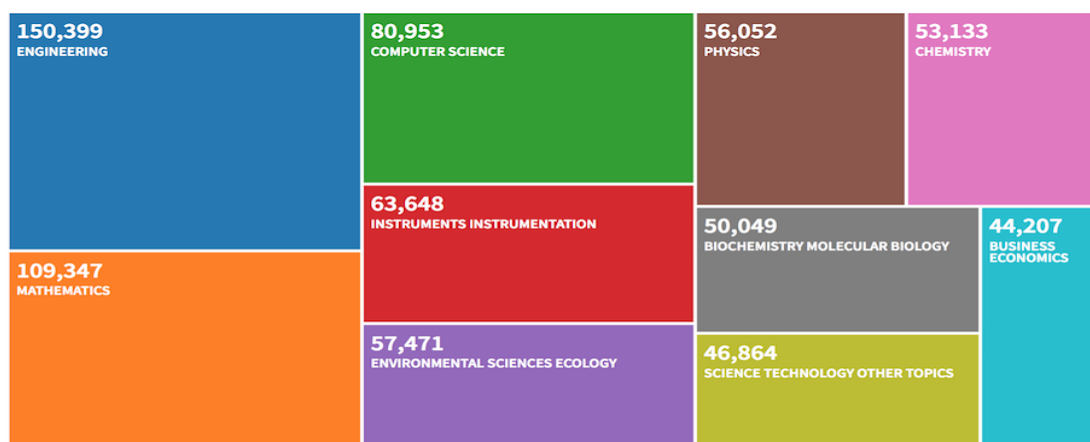
Una serie temporal es una secuencia de valores medida regularmente a lo largo del tiempo. El análisis de series temporales busca comprender el pasado y poder hacer una predicción sobre cómo será el futuro, aborda métodos que ayudan a interpretar todo tipo de datos, extrayendo información representativa, tanto referente a los orígenes o relaciones subyacentes como a la posibilidad de extrapolar y predecir su comportamiento futuro.

Actualmente, la predicción de series temporales se usa en muchos campos, desde la medicina hasta la economía. En la industria eólica, por ejemplo, la predicción de la velocidad del viento juega un papel vital en la programación óptima en el control de la generación y conversión de la energía eólica. El objetivo principal es reducir las emisiones de gases de efecto invernadero y construir una sociedad renovable. El informe de viento global, publicado por el Consejo Global de Energía Eólica (GEWC) en 2017, declaró que el mercado mundial de la energía eólica en 2016 fue más de 54.6 GW, lo que generó una capacidad global total instalada de casi 487 GW, generada principalmente por China, EE.UU y la India.

Otro ejemplo podría ser el éxito de las campañas de retención en los mercados, como la industria de las telecomunicaciones, que a menudo depende de la predicción precisa de rumores potenciales. Ser capaz de identificar ciertos patrones de comportamiento que conducen a la rotación es importante, ya que permite a la organización hacer arreglos para la retención de manera oportuna. Además, investigaciones previas han demostrado que la decisión de dejar a un operador por otro, a menudo está influenciada por el círculo social del cliente. Por lo tanto, las características que representan el estado de rotación de sus conexiones suelen ser buenos predictores de la deserción.

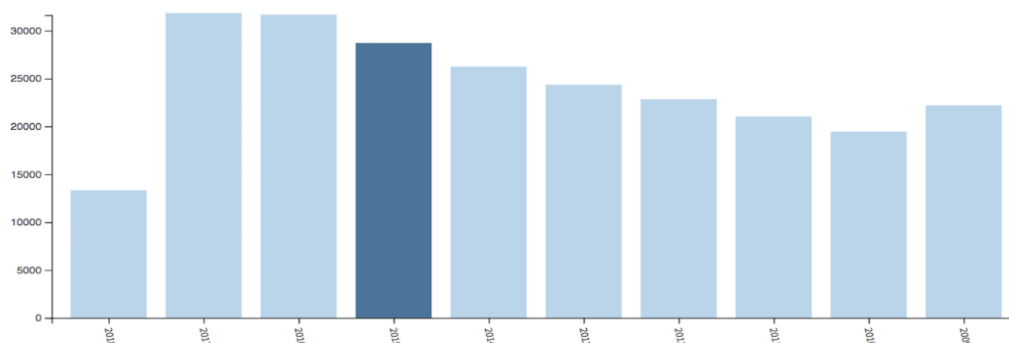
Aquí solo se han comentado varios ejemplos, el análisis de las series temporales se encuentra en todos los campos, tanto en ingeniería, matemáticas, medicina, química, economía como en muchos otros.

A continuación se muestra una imagen de la base de datos *Web of Science* en la que se detalla el número de trabajos acerca de series temporales por área de conocimiento.



**Ilustración 1.1** Número de trabajos de series temporales por área de conocimiento

En la siguiente ilustración se muestra una gráfica en la que se puede observar la evolución que presenta dicho campo, presentando un más que evidente incremento a lo largo del tiempo.



**Ilustración 1.2** Número de trabajos de series temporales a lo largo del tiempo

Si el Machine Learning se centra en lograr un aprendizaje automático, el Deep Learning se orienta hacia el entendimiento de los datos que va teniendo, suponiendo un acercamiento al modo de pensar humano, buscando imitar las características de nuestro sistema nervioso.

El aprendizaje profundo, en inglés Deep Learning, se puede considerar la punta de lanza de la inteligencia artificial. Los sistemas basados en Deep Learning están proliferando y ya son muchos los avances producidos en este campo. No son pocas las ocasiones en las que sus resultados copan los titulares de revistas científicas, y

no sólo eso, sino que también, poco a poco se van introduciendo en nuestro día a día, a través de software que permiten contar con asistentes de voz, o avanzando en el perfeccionamiento de un coche autónomo, por ejemplo.

El TFG consiste en el desarrollo de una plataforma web para el análisis y la predicción de series temporales. Esta plataforma contendrá una serie de módulos entre los que se incluyen: módulo de preprocesamiento y análisis de los datos, módulo de predicción y módulo para la obtención y visualización de los resultados. En cada uno de estos módulos se estudiarán los métodos más importantes utilizados en Deep Learning para el tratamiento de series temporales, incorporando al sistema los más representativos.

Este trabajo fin de grado trata sobre el desarrollo de una plataforma web para la predicción de series temporales con Deep Learning. Para el desarrollo de dicha herramienta, se ha realizado una etapa de investigación y estudio, ya que en la carrera no se estudian dichos conocimientos, tan sólo algunos conceptos estudiados en asignaturas como Estadística. El desarrollo de la plataforma se ha realizado en Python, un lenguaje un poco desconocido para mi, pero a la vez muy potente, versátil y con una curva de aprendizaje muy leve.

A continuación se describe la estructura de la memoria perteneciente a este trabajo fin de grado, la cual está formada por once capítulos:

- El capítulo 1, donde se encuentra la introducción y en el que se ha explicado la motivación y los objetivos del TFG.
- El capítulo 2, introduciendo el término de serie temporal, así como la definición de sus componentes y la caracterización de las series.
- El capítulo 3, dónde hablamos de la importancia de la predicción de series temporales.
- El capítulo 4, donde introducimos las técnicas estadísticas clásicas para la predicción de series temporales, presentado modelos autorregresivos, de media móvil y ARIMA.
- El capítulo 5, el cual está dedicado a estudiar las estructuras, arquitecturas y redes de Deep Learning para la predicción de series temporales.

- El capítulo 6, dedicado al análisis del proyecto, realizando un análisis de requisitos, así como una planificación del proyecto.
- El capítulo 7, dedicado al diseño del proyecto, dónde estudiamos los distintos módulos que forman la plataforma.
- El capítulo 8, describiendo cómo se ha implementado el proyecto, las tecnologías usadas y la estructura interna del mismo.
- El capítulo 9, en el cual se realiza un recorrido por la plataforma a modo de prueba.
- El capítulo 10, dedicado al diseño de un manual de usuario, en el cual se describe de forma exhaustiva los pasos y posibilidades de la plataforma.
- El capítulo 11, dónde se expondrán las conclusiones que nos ha aportado realizar este T.F.G.

## 2 Series temporales

### 2.1 ¿Qué es una serie temporal?

Una **serie temporal** (o simplemente una serie de tiempo) es una secuencia de  $N$  observaciones ordenadas y equidistantes cronológicamente sobre una característica, la cual la convierte en una serie univariante o escalar, o sobre varias características, multivariante vectorial, de una unidad observable en diferentes momentos.

Dada su naturaleza, resulta muy fácil interpretar una serie temporal dibujada en un gráfico:

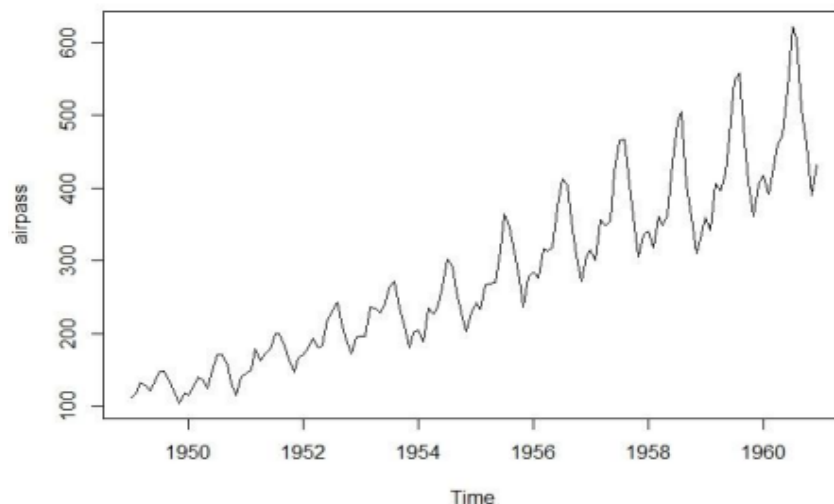


Ilustración 2.1 Número mensual de pasajeros de una aerolínea a lo largo del tiempo

### 2.2 Componentes de las series temporales

El estudio descriptivo de series temporales se basa en la idea de descomponer la variación de una serie en varias componentes básicas. Este enfoque no siempre resulta ser el más adecuado, pero es interesante cuando en la serie se observa cierta tendencia o cierta periodicidad. Hay que resaltar que esta descomposición no es en general única.

Este enfoque descriptivo consiste en encontrar componentes que correspondan a una tendencia a largo plazo, un comportamiento estacional y una parte aleatoria.

Las componentes o fuentes de variación que se consideran habitualmente son las siguientes:

- **Tendencia:** Se puede definir como un cambio a largo plazo que se produce en relación al nivel medio, o el cambio a largo plazo de la media. La tendencia se identifica con un movimiento suave de la serie a largo plazo.
- **Efecto Estacional:** Muchas series temporales presentan cierta periodicidad o dicho de otro modo, variación de cierto periodo (anual, mensual ...). Por ejemplo, el paro laboral aumenta en general en invierno y disminuye en verano. Estos tipos de efectos son fáciles de entender y se pueden medir explícitamente o incluso se pueden eliminar del conjunto de los datos, desestacionalizando la serie original.
- **Componente Aleatoria:** Una vez identificados los componentes anteriores y después de haberlos eliminado, persisten unos valores que son aleatorios. Se pretende estudiar qué tipo de comportamiento aleatorio presentan estos residuos, utilizando algún tipo de modelo probabilístico que los describa.

De las tres componentes, las dos primeras componentes son determinísticas, mientras que la última es aleatoria. La ecuación que representa la serie temporal se denota:

$$X_t = T_t + E_t + I_t$$

$T_t$  es la tendencia,  $E_t$  denota la componente estacional, lo que constituye la parte determinística e  $I_t$  es la parte aleatorio o ruido.

En la siguiente figura se muestran cuatro series donde se pueden interpretar visualmente cada una de las componentes mencionadas anteriormente:



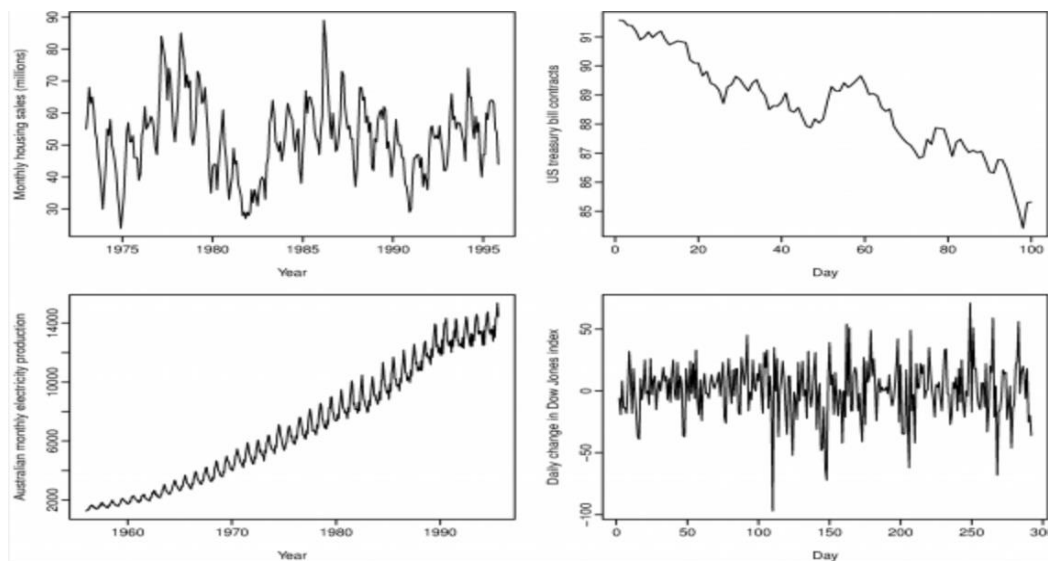


Ilustración 2.2 Componentes de las series temporales

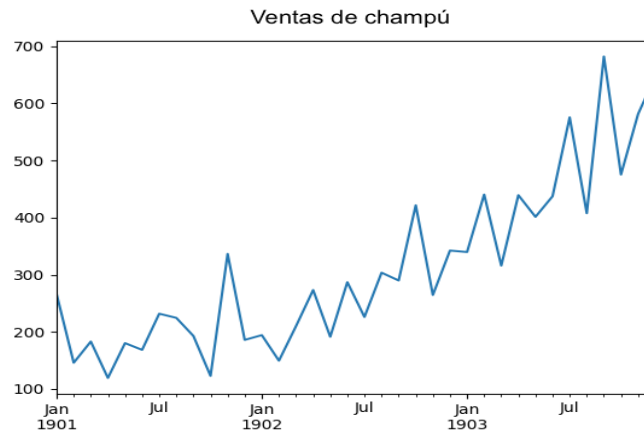
En la ilustración 2.2 se puede observar; arriba a la izquierda: venta de casas (mensual), muestra comportamiento estacional y cíclico. Arriba a la derecha: resultados del mercado de Chicago durante cien días consecutivos, muestra una clara tendencia a la baja. Abajo a la izquierda: producción mensual de electricidad en Australia, obvia componente estacional y de tendencia. Abajo a la derecha: el cambio diario del estado de Dow Jones no sigue ningún patrón reconocible.

Las series temporales son usadas en estadística, reconocimiento de patrones, econometría, matemáticas financieras, previsión meteorológica, predicción de terremotos, astronomía, ingeniería de comunicaciones, y básicamente cualquier campo de ciencia aplicada o ingeniería que involucre mediciones temporales.

En una serie temporal, llamamos frecuencia al número de mediciones anuales, dependiendo del número de variables una serie temporal puede ser:

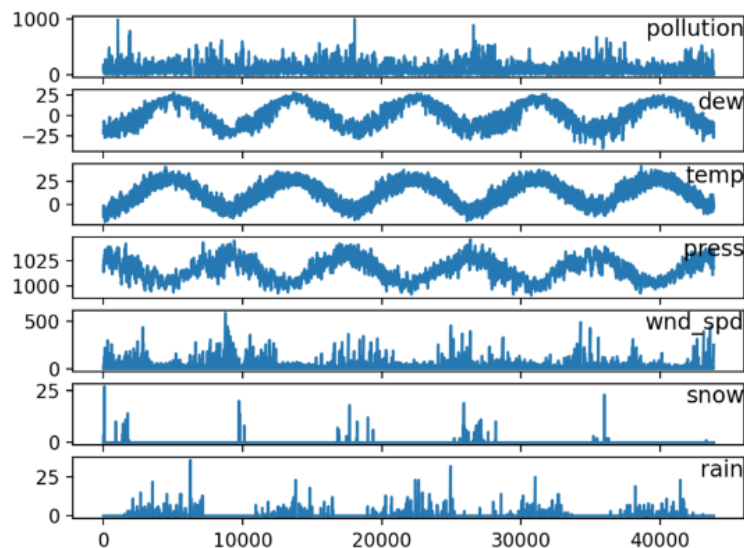
- **Monovariable:** para cada medición se recoge una sólo muestra o variable. Es lo más común y sobre lo que nos vamos a centrar en este trabajo.
- **Multivariable:** cada medición recoge datos de dos o más fuentes.

Un ejemplo de serie temporal monovariable podría ser el número de ventas de un producto de champú a lo largo del tiempo, cuya serie temporal se representa en la imagen siguiente:



**Ilustración 2.3 Número de ventas de un producto de champú a lo largo del tiempo**

Mientras que un ejemplo de serie temporal multivariable podría ser sobre las condiciones climatológicas de Beijing, China, las cuales dependen de diversas variables como son: polución, lluvia, año, mes, día, hora, presión, temperatura, etc.



**Ilustración 2.4 Series temporales sobre las condiciones climatológicas de Beijing, China**

## 2.3 Caracterización de series temporales

Las series temporales se pueden clasificar en estacionarias o en no estacionarias:

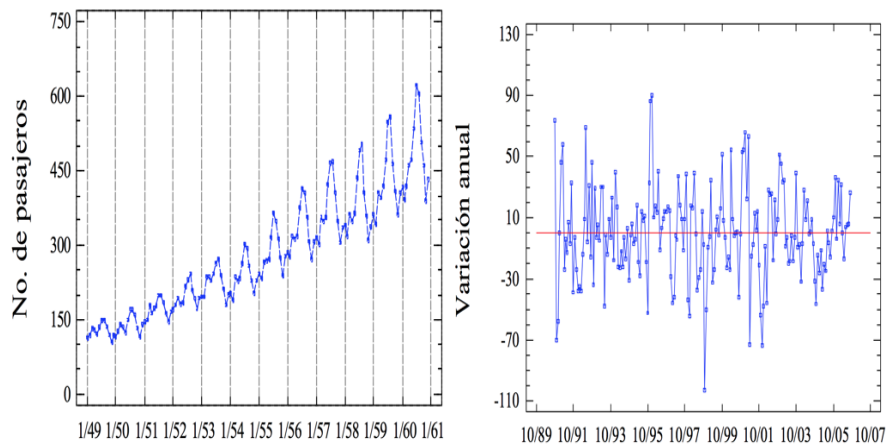


Ilustración 2.5 Ejemplos de series temporales

- **Estacionarias:** Se dice que una serie es estacionaria cuando es estable, esto es, cuando la media y la variabilidad son constantes a lo largo del tiempo. Esto se refleja gráficamente en que los valores de la serie tienden a oscilar alrededor de una media constante y la variabilidad con respecto a esa media también permanece constante en el tiempo. Como resumen podríamos decir que es una serie constante/estable a lo largo del tiempo
- **No estacionarias:** Son series en las cuales la media y/o variabilidad cambian en el tiempo. Los cambios en la media determinan una tendencia a crecer o decrecer a largo plazo, por lo que la serie no oscila alrededor de un valor constante.

Lo bueno de que las series sean estacionarias, es que podemos obtener predicciones más fácilmente, como la media es constante, podemos estimarla con todos los datos, y utilizar este valor para predecir una nueva observación.

En la ilustración 2.4 se puede observar en la figura de la izquierda: una serie no estacionaria; Número mensual de pasajeros de avión. USA.

Mientras que en la figura de la derecha: una serie estacionaria; Variaciones anuales de la media de los pluviómetros peninsulares entre Octubre de 1990 y Septiembre 2006.

## 2.4 Estudio de la estacionariedad de una serie temporal

Como se ha dicho anteriormente, lo bueno de que las series sean estacionarias, es que podemos obtener predicciones más fácilmente, como la media es constante, podemos estimarla con todos los datos, y utilizar este valor para predecir una nueva observación.

Sin embargo, hay diversos factores que impiden que una serie sea estacionaria, como puede ser la tendencia, la estacionalidad o la varianza no homogénea:

- Las series no estacionarias pueden mostrar una tendencia, es decir, que la media crezca o decrezca a lo largo del tiempo.
- Las series no estacionarias pueden mostrar cambios de varianza.
- Las series no estacionarias pueden presentar efectos estacionales, es decir, que el comportamiento de la serie sea parecido en ciertos tiempos periódicos en el tiempo.

Hay diversos métodos para comprobar la estacionalidad de una serie:

- El método más simple es **graficando** la serie, a simple vista se puede observar si una serie es estacionaria o no, aunque hay ocasiones en las que no resulta tan fácil comprobarlo y se deben realizar otro tipo de comprobaciones.
- **Diagrama de autocorrelación (ACF)**: Es una herramienta que mide el grado de dependencia entre observaciones próximas en una sucesión de valores. En dicho diagrama, la dependencia temporal debe tender a

cero de forma rápida. En caso contrario, de que el crecimiento sea lento, es debido a que la serie no es estacionaria.

- **Test de raíz unitaria:** Como su propio nombre indica, es un test , un test de hipótesis estadísticas de estacionariedad. Uno de los más populares es el Augmented Dickey-Fuller (ADF). Es un simple modelo autorregresivo:

$$y_t = py_{t-1} + u_t$$

Donde  $y_t$  es la variable de interés,  $t$  es el índice de tiempo,  $p$  un coeficiente, y  $u_t$  es el término de error, La raíz unitaria está presente si  $p = 1$ . En este caso, el modelo no sería estacionario.

Una vez que hemos analizado nuestra serie temporal y hemos comprobado que es estacionaria, podemos aplicarle modelos predictivos. Ya que al no serlo, los modelos contruidos en base a la serie temporal no serían los mejores.

## 2.5 Análisis de series temporales

El análisis de series temporales implica desarrollar modelos que capturen o describan mejor una serie temporal para comprender las causas subyacentes. Este campo de estudio busca el por qué detrás de un conjunto de datos, lo que a menudo implica hacer suposiciones sobre la forma de los datos.

El objetivo principal del análisis de series temporales es desarrollar modelos matemáticos que brinden descripciones plausibles a partir de datos de muestra.

## 2.6 Objetivos del análisis de series temporales

**Descripción:** Cuando se estudia una serie temporal, lo primero que se tiene que hacer es graficarla y considerar las medidas descriptivas básicas.

Así, se tiene que considerar:

- Si los datos presentan forma creciente (tendencia).
- Si existe influencia de ciertos periodos de cualquier unidad de tiempo (estacionalidad).
- Si aparecen outliers (observaciones extrañas o discordantes).

**Predicción:** Cuando se observan los valores de una serie, se pretende normalmente no sólo explicar el pasado, sino también predecir el futuro.

### 3 Predicción de series temporales

El propósito del análisis de series temporales es generalmente doble: comprender o modelar los mecanismos estocásticos que dan lugar a una serie observada y predecir o pronosticar los valores futuros de una serie en función de los valores de la misma.

#### 3.1 ¿Qué es la predicción de series temporales?

Cuando se observan los valores de una serie, se pretende normalmente no sólo explicar el pasado, sino también predecir el futuro.

La previsión de series temporales es un área importante del aprendizaje automático. Es importante ya que hay muchos problemas de predicción que dependen de una componente de tiempo. Estos problemas se descuidan porque es esta componente del tiempo la que hace que los problemas de series de tiempo sean más difíciles de manejar.

#### 3.2 Ejemplos de predicción de series temporales

Hay casi un suministro interminable de problemas de pronóstico de series de tiempo. A continuación hay 10 ejemplos de una variedad de industrias para hacer que las nociones de análisis de series de tiempo y pronósticos sean más concretos.

- Pronosticar el rendimiento de maíz en toneladas por estado cada año.
- Pronosticar el precio de cierre de una acción cada día.
- Pronóstico de la tasa de natalidad en todos los hospitales de una ciudad cada año.
- Pronóstico de ventas de productos en unidades vendidas cada día para una tienda.
- Pronosticar el número de pasajeros a través de una estación de tren todos los días.
- Pronóstico del desempleo para un estado cada trimestre.
- Pronosticar la demanda de utilización en un servidor cada hora.

- Pronosticar el tamaño de la población de conejos en un estado en cada temporada de cría.
- Pronosticar el precio promedio de la gasolina en una ciudad cada día.

### 3.3 Análisis exploratorio de datos para predicción de series

Al pronosticar, es importante entender su objetivo. Hay que responder a muchas preguntas para ayudar a ampliar los detalles del problema.

Por ejemplo:

- ¿Cuántos datos tiene disponibles y puede reunirlos todos juntos? Más datos a menudo son más útiles, ofreciendo una mayor oportunidad para el análisis de datos exploratorios, pruebas y ajuste de modelos y fidelidad del modelo.
- ¿Cuál es el horizonte temporal de predicciones que se requiere? ¿Corto, mediano o largo plazo? Los horizontes temporales más cortos a menudo son más fáciles de predecir con mayor confianza.
- ¿Se pueden actualizar los pronósticos con frecuencia a lo largo del tiempo o deben hacerse una vez y permanecer estáticos? La actualización de pronósticos a medida que la nueva información está disponible a menudo da como resultado predicciones más precisas.
- ¿A qué frecuencia temporal se requieren los pronósticos? Con frecuencia, las previsiones se pueden realizar a frecuencias más bajas o más altas, lo que le permite aprovechar el muestreo descendente y el muestreo ascendente de los datos, lo que a su vez puede ofrecer beneficios al modelar.

Los datos de series de tiempo, a menudo requieren limpieza, escalado e incluso transformación. Por ejemplo:

- **Frecuencia:** Quizás los datos hayan sido proporcionados a una frecuencia demasiado alta para modelar o está espaciada de forma desigual a lo largo del tiempo, lo que requiere un remuestreo para su uso en algunos modelos.
- **Outliers:** Valores atípicos, corruptos o extremos que deben ser identificados y manejados.



- **Desaparecidos:** Se refieren a datos faltantes que deban ser interpolados o imputados.
- **Normalización:** Consiste en realizar un cambio de escala en los datos del rango inicial de modo que todos los valores estén en el rango de 0 y 1.

### 3.4 Preprocesamiento de una serie temporal

Un concepto importante comentado anteriormente es la estacionalidad. Una serie temporal es estacionaria si las variables aleatorias que lo componen poseen la misma función de distribución, probabilidad, con independencia del momento de tiempo considerado.

En otras palabras, una serie temporal es estacionaria cuando no tiene tendencia ni estacionalidad y la varianza permanece constante a lo largo del tiempo. En una serie así, ni la media ni la varianza ni las autocorrelaciones dependen del tiempo.

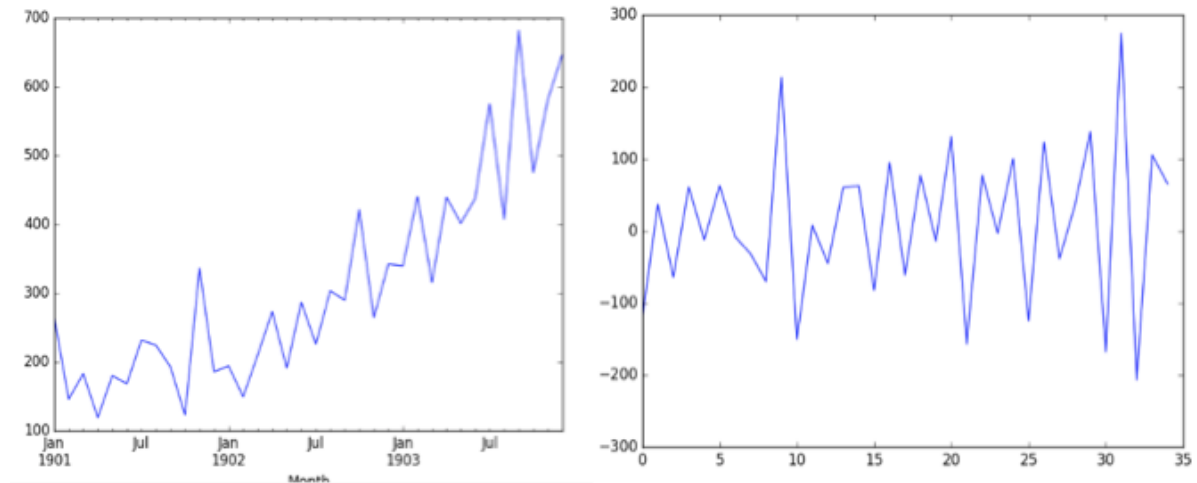
En este proyecto, se realiza una etapa de preprocesamiento con el fin de conseguir que la serie sea estacionaria, por lo que se hace uso de las técnicas descritas a continuación.

#### 3.4.1 Eliminación de la tendencia

Como hemos comentado anteriormente, la tendencia es un componente clave en las series temporales, por lo que muchas veces es necesario un tratamiento especial para corregir dicha componente.

A la hora de eliminar la tendencia, se aplica diferencias regulares, esto consiste en aplicar a cada valor de la serie la diferencia del valor inmediatamente anterior de la siguiente forma:

$$X_i = X_i - X_{i-1}$$



**Ilustración 3.1** Serie temporal original y serie temporal con diferenciación

### 3.4.2 Ajuste de la estacionalidad

Al igual que la tendencia, la periodicidad es otro componente muy importante a la hora de trabajar con series temporales. Se utiliza un método parecido al comentado en el apartado 3.4.1 Eliminación de tendencia, ya que aplica el mismo método, diferenciación. Consiste en aplicar tantas diferencias estacionales como sean necesarias hasta que la estacionalidad queda eliminada. Son como las regulares, sin embargo, en lugar del restar el valor inmediatamente anterior, restamos el valor correspondiente a la misma fecha del año anterior.

$$X_i = X_i - X_{i - frecuencia(ts)}$$

Otro mecanismo para ajustar la estacionalidad consiste en restar la media. Los promedios móviles en sí mismos tienen un efecto suavizante sobre las variaciones estacionales.

### 3.4.3 Homogeneización de la varianza

Se trata de un método clásico para corregir sesgos en la distribución de errores y para corregir varianzas desiguales a partir de un modelo matemático que depende de un parámetro lambda.

Un método clásico para corregir estos sesgos es la transformación de Box-Cox.

Para la serie original  $X$  de longitud  $N$ :

$$X = x_0 + x_1 + x_2 + \dots + x_{n-1}$$

La transformación Box-Cox de un parámetro es la siguiente:

$$x_i(\lambda) = \begin{cases} \frac{X_i^\lambda - 1}{\lambda}, \lambda \neq 0 \\ \ln(x_i), \lambda = 0 \end{cases}$$

donde  $i=0,1,2,\dots,N-1$ .

Esta transformación solo tiene un parámetro  $\lambda$ . Si el valor de  $\lambda$  es igual a 0, se lleva a cabo la transformación logarítmica de la serie inicial, si el parámetro es distinto de cero, la transformación es potencial. En el caso en el que  $\lambda$  sea igual a uno, la ley de distribución de la serie inicial permanece inalterada, aunque la serie cambia, ya que se resta una unidad a cada uno de sus valores.

## 4 Técnicas estadísticas para la predicción de series temporales

### 4.1 Media móvil

El suavizado promedio móvil es una técnica efectiva en el pronóstico de series de tiempo que se puede usar para la preparación de datos, la ingeniería de características e incluso directamente para hacer predicciones. Es una técnica aplicada a series de tiempo para eliminar la variación de grano fino entre los pasos de tiempo. La intención de dicha técnica es eliminar el ruido y exponer mejor la señal de los procesos causales subyacentes. Los promedios móviles son un tipo de suavizado simple y común utilizado en el análisis de series temporales y el pronóstico de series de tiempo.

El cálculo de una media móvil implica crear una nueva serie donde los valores están compuestos por el promedio de observaciones en bruto en la serie de tiempo original.

Este promedio móvil requiere que se especifique un tamaño de ventana denominado ancho de ventana. La parte "móvil" en la media móvil se refiere al hecho de que la ventana definida por el ancho de la ventana se desliza a lo largo de la serie temporal para calcular los valores promedio en la nueva serie.

Hay dos tipos principales de promedio móvil que se utilizan:

- **Promedio móvil centrado:** El valor en el tiempo (t) se calcula como el promedio de las observaciones en bruto en, antes y después del tiempo (t).
- **Promedio móvil final:** El valor en el tiempo (t) se calcula como el promedio de las observaciones en bruto en y antes del tiempo (t).

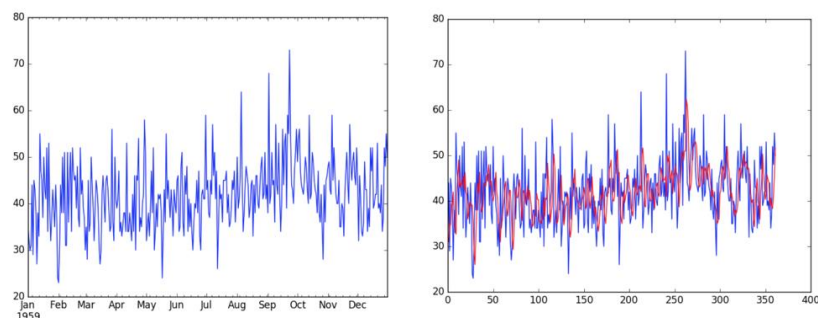


Ilustración 4.1 Serie temporal en azul y su respectiva media móvil en rojo

## 4.2 Autorregresión

La autorregresión es un modelo de serie temporal que utiliza observaciones de pasos de tiempo anteriores como entrada a una ecuación de regresión para predecir el valor en el siguiente paso de tiempo.

Es una idea muy simple que puede dar como resultado pronósticos precisos sobre una serie de problemas de series de tiempo.

Un modelo de regresión, como la regresión lineal, modela un valor de salida basado en una combinación lineal de valores de entrada.

$$Y = b_0 + b_1 * x_1$$

Donde Y es la predicción,  $b_0$  y  $b_1$  son coeficientes encontrados al optimizar el modelo en los datos de entrenamiento, y X es un valor de entrada.

Esta técnica se puede utilizar en series de tiempo donde las variables de entrada se toman como observaciones en pasos de tiempo anteriores, llamadas variables de retardo.

Por ejemplo, podemos predecir el valor para el siguiente paso de tiempo ( $t + 1$ ) dadas las observaciones en los dos últimos pasos de tiempo ( $t-1$  y  $t-2$ ):

$$Y(t + 1) = b_0 + b_1 * x(t - 1) + b_2 * x(t - 2)$$

Debido a que el modelo de regresión usa datos de la misma variable de entrada en pasos de tiempo previos, se lo denomina autorregresión.

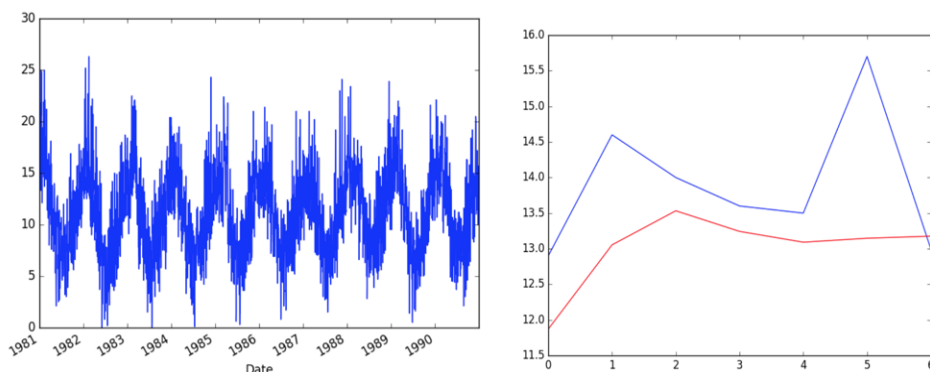


Ilustración 4.2 Serie temporal y valores esperados

En la ilustración 4.2 se muestra a la izquierda la serie temporal correspondiente a las temperaturas mínimas recogidas diariamente desde 1981 hasta 1990, mientras que en la figura de la derecha, se realiza un gráfico de valores esperados (azul) frente a valores predichos (rojo), para una semana en concreto.

### 4.3 Autocorrelación

Un modelo de autorregresión supone que las observaciones en pasos de tiempo anteriores son útiles para predecir el valor en el siguiente paso, a esta relación entre variables se denomina correlación. Si dichas variables cambian en la misma dirección se denomina correlación positiva, en cambio, si las variables se mueven en direcciones opuestas a medida que los valores cambian se denomina correlación negativa.

Se pueden usar medidas estadísticas para calcular la correlación entre la variable de salida y los valores en los pasos de tiempo previos en diferentes rezagos. Cuanto más fuerte sea la correlación entre la variable de salida y una variable retrasada específica, más peso puede poner ese modelo de autorregresión sobre esa variable al modelar.

De nuevo, debido a que la correlación se calcula entre la variable y sí mismo en los pasos de tiempo anteriores, se denomina autocorrelación. También se denomina correlación serial debido a la estructura secuenciada de los datos de series de tiempo.

Las estadísticas de correlación también pueden ayudar a elegir qué variables de demora serán útiles en un modelo y cuáles no.

Es posible que una serie temporal no pueda ser predecible si las variables de desfase muestran baja o nula correlación.

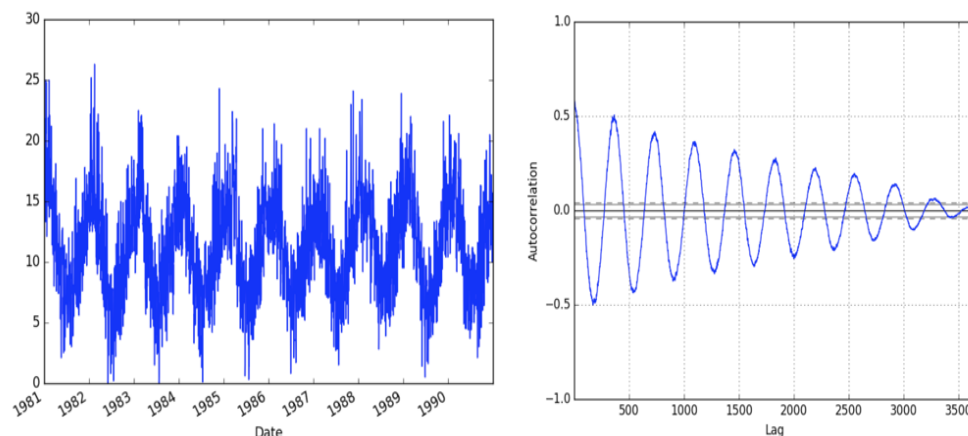


Ilustración 4.3 Serie temporal y su diagrama de autocorrelación

Respecto a la ilustración 4.3, en la figura de la izquierda se muestra la serie temporal correspondiente a las temperaturas mínimas recogidas diariamente desde 1981 hasta 1990, mientras que en la figura de la derecha, mediante un simple test de autocorrelación podemos ver una oscilación en la correlación positiva y negativa a medida que los valores de temperatura cambian en las temporadas de verano en invierno.

#### 4.4 ARIMA

ARIMA (Autoregressive Integrated Moving Average) es un método que se utiliza para el pronóstico de series de tiempo, es muy popular y utilizado. Es una clase de modelo que captura un conjunto de diferentes estructuras temporales estándar en datos de series temporales, por lo que se utiliza para analizar y pronosticar los datos de las series temporales. Es un método simple y poderoso para hacer pronósticos de series de tiempo.

El acrónimo es descriptivo y significa lo siguiente:

- **AR: Autoregresión.** Un modelo que usa la relación dependiente entre una observación y un número de observaciones rezagadas.
- **I: Integrado.** El uso de la diferenciación de observaciones en bruto (por ejemplo, restando una observación de una observación en el paso de tiempo anterior) para hacer que la serie temporal sea estacionaria.

- **MA: Promedio móvil.** Un modelo que usa la dependencia entre una observación y un error residual de un modelo de promedio móvil aplicado a observaciones rezagadas.

Cada una de estos componentes se especifica explícitamente en el modelo como parámetro. Se utiliza una notación estándar de ARIMA ( $p, d, q$ ) donde los parámetros se sustituyen por valores enteros para indicar rápidamente el modelo ARIMA específico que se está utilizando.

Los parámetros del modelo ARIMA se definen de la siguiente manera:

- **p:** El número de observaciones de retraso incluidas en el modelo, también llamado orden de retraso.
- **d:** El número de veces que las observaciones son diferenciadas, también llamado el grado de diferenciación.
- **q:** El tamaño de la ventana de promedio móvil, también llamado orden de promedio móvil.

Se construye un modelo de regresión lineal que incluye el número y tipo de términos especificados, y los datos se preparan mediante un grado de diferenciación para hacerlo estacionario, es decir, para eliminar las estructuras estacionales y de tendencia que afectan negativamente al modelo de regresión.

Se puede usar un valor de 0 para un parámetro, que indica que no se debe usar ese elemento del modelo. De esta forma, el modelo ARIMA se puede configurar para realizar la función de un modelo ARMA, e incluso un modelo AR, I o MA simple.



## 5 Predicción de series temporales con Deep Learning

El Deep Learning es sin duda el área de investigación más popular dentro del campo de la inteligencia artificial. La mayoría de nuevas investigaciones que se realizan trabajan con modelos basados en técnicas de Deep Learning.

El aprendizaje profundo o Deep learning es parte de un conjunto más amplio de métodos de aprendizaje automático basados en asimilar representaciones de datos. Una observación (por ejemplo, una imagen) puede ser representada en muchas formas (por ejemplo, un vector de píxeles), pero algunas representaciones hacen más fácil aprender tareas de interés (por ejemplo, "¿es esta imagen una cara humana?") sobre la base de ejemplos, y la investigación en esta área intenta definir qué representaciones son mejores y cómo crear modelos para reconocer estas representaciones.

Aquí nos centramos en el mundo de la economía y el business intelligence, el deep learning se está aplicando en los campos del customer centricity, finanzas o mercados. En transformación digital, está muy ligado a los progresos en proyectos relacionados con el Internet de las cosas y, en medicina, con los campos de diagnóstico, genética y muy especialmente en el descubrimiento de fármacos, que son los más beneficiados.

## 5.1 Estructuras de las redes neuronales

Las redes neuronales se pueden estructurar de diversas maneras, según el número de capas, según el tipo de conexiones y por último según el grado de conexión.

Según el número de capas:

- **Redes neuronales monocapas:** Se corresponde con la red neuronal más sencilla ya que se tiene una capa de neuronas que proyectan las entradas a una capa de neuronas de salida donde se realizan diferentes cálculos.
- **Redes neuronales multicapa:** son una generalización de las anteriores, existiendo un conjunto de capas intermedias entre la entrada y la salida (capas ocultas). Este tipo de red puede estar total o parcialmente conectada.

Según el tipo de conexión:

- **Redes neuronales no recurrentes:** En esta red la propagación de las señales se produce en un sentido solamente, no existiendo la posibilidad de realimentaciones. Lógicamente estas estructuras no tienen memoria.
- **Redes neuronales recurrentes:** Esta red viene caracterizada por la existencia de lazos de realimentación. Estos lazos pueden ser entre neuronas de diferentes capas, neuronas de la misma capa o, más sencillamente, entre una misma neurona. Esta estructura estudia principalmente la dinámica de sistemas no lineales.

Según el grado de conexión:

- **Redes neuronales totalmente conectadas,** en este caso todas las neuronas de una capa se encuentran conectadas con las de la capa siguiente (redes no recurrentes) o con las de la anterior (redes recurrentes).
- **Redes parcialmente conectadas,** son en las que no se da la conexión total entre neuronas de diferentes capas.

## 5.2 Arquitecturas de aprendizaje profundo

El aprendizaje profundo está formado por redes profundas de varias patologías. Las redes neurales llevan cierto tiempo en escena, pero el desarrollo de numerosas capas de redes ha hecho que utilizarlas sea más práctico.

Añadir más capas significa que hay más interconexiones y ponderaciones entre y dentro de capas.

El número de arquitecturas y algoritmos que se utilizan en el aprendizaje profundo es amplio y variado. En la siguiente sección exploramos dos arquitecturas de aprendizaje profundo que han cubierto los últimos 20 años. Notablemente, LSTM y CNN son dos de los enfoques más antiguos, pero también dos de los más usados en diferentes aplicaciones. En la siguiente ilustración<sup>1</sup> se muestran las distintas redes neurales a través del tiempo.

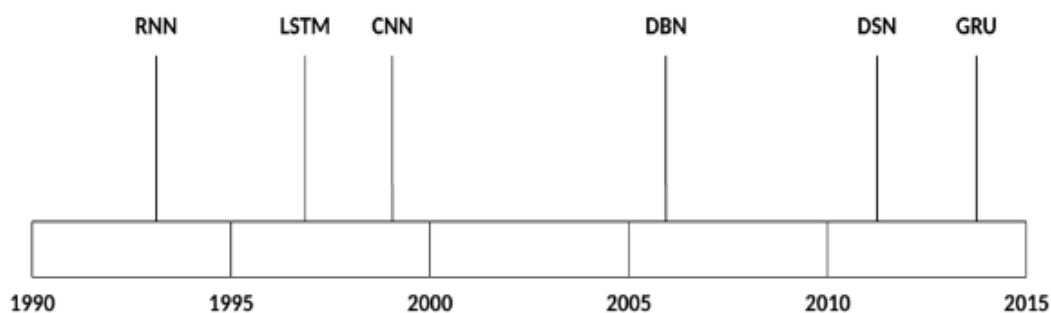


Ilustración 5.1 Diagrama temporal de las redes neurales

### 5.2.1 Redes convolucionales

Son muy eficaces para procesar datos visuales y otros datos bidimensionales, son muy similares a las redes neuronales ordinarias como el perceptrón multicapa; se componen de neuronas que tienen pesos y sesgos que pueden aprender. Cada neurona recibe algunas entradas, realiza un producto escalar y luego aplica una función de activación. Al igual que en el perceptrón multicapa también vamos a tener una función de pérdida o costo sobre la última capa, la cual estará totalmente

<sup>1</sup> <https://www.ibm.com/developerworks/ssa/library/cc-machine-learning-deep-learning-architectures/index.html>

conectada. Lo que diferencia a las redes neuronales convolucionales es que disponen de una serie de unidades o neuronas especiales que realizan operaciones de convolución y pooling, entre otras, procesando la información gráfica de manera espacial.

### 5.2.2 Redes neuronales recurrentes

Estas redes pueden usar su memoria interna para procesar secuencias arbitrarias de entradas. Esto los hace aplicables a tareas tales como reconocimiento de escritura, predicción de series temporales o reconocimiento de voz.

Los seres humanos no comenzamos nuestro pensamiento desde cero cada segundo, sino que los mismos tienen una persistencia. Las redes neuronales prealimentadas tradicionales no cuentan con esta persistencia, y esto es una deficiencia importante. Las redes neuronales recurrentes abordan este problema. Son redes con bucles de retroalimentación, que permiten que la información persista.

Una red neuronal recurrente puede ser conceptualizada como una red con múltiples copias de ella misma, en las que cada una de ellas pasa un mensaje a su sucesor. Esta naturaleza en forma de cadena revela que las redes neuronales recurrentes están íntimamente relacionadas con las secuencias y listas; por lo que son ideales para trabajar con este tipo de datos. En los últimos años, ha habido un éxito increíble aplicando redes neuronales recurrentes a una variedad de problemas como: reconocimiento de voz, modelado de lenguaje, traducción, subtítulos de imágenes y para la predicción de series temporales.

Las redes de memoria de largo plazo a corto plazo, generalmente llamadas **LSTMs**, son un tipo especial de redes neuronales recurrentes, capaces de aprender dependencias a largo plazo, tienen una estructura como cadena, pero el módulo de repetición tiene una estructura diferente, debido a que en lugar de tener una sola capa de red neuronal, tiene cuatro, que interactúan de una manera especial permitiendo tener una memoria a más largo plazo. Es por esto que nos centramos en las de este último tipo en la elaboración de este trabajo fin de grado.

### 5.3 Redes LSTM ( Long Short-Term Memory)

Las LSTM son un tipo de red neuronal recurrente, que comúnmente se componen de una celda, una puerta de entrada, una puerta de salida y una puerta de olvido. La célula es la responsable de “recordar” los valores en intervalos de tiempo arbitrarios; de ahí la palabra “memoria” en LSTM. Cada una de las tres puertas se pueden considerar como una neurona artificial “convencional”, como en una red.

A mediados de los años 90, los investigadores alemanes Sepp Hochreiter y Juergen Schmidhuber propusieron una variación de la red recurrente con las denominadas unidades de memoria a largo plazo, o LSTM, como una solución al problema del gradiente de fuga.

Las LSTM ayudan a preservar el error que se puede volver a propagar a través del tiempo y las capas. Al mantener un error más constante, permiten que las redes recurrentes continúen aprendiendo durante muchos pasos de tiempo (más de 1000), abriendo así un canal para vincular causas y efectos de forma remota. Este es uno de los desafíos centrales para el aprendizaje automático y la IA, ya que los algoritmos se enfrentan con frecuencia a entornos en los que las señales de recompensa son dispersas y diferidas, como la vida misma.

Las LSTM contienen información fuera del flujo normal de la red recurrente en una celda cerrada. La información puede almacenarse, escribirse o leerse desde una celda, al igual que los datos en la memoria de una computadora. La célula toma decisiones sobre qué almacenar y cuándo permitir las lecturas, escrituras y borrados, a través de puertas que se abren y cierran.

Esas puertas actúan sobre las señales que reciben, y de forma similar a los nodos de la red neuronal, bloquean o transmiten información en función de su fuerza e importación, que filtran con sus propios conjuntos de ponderaciones. Esos pesos, como los pesos que modulan los estados de entrada y ocultos, se ajustan a través del proceso de aprendizaje de redes recurrentes. Es decir, las células aprenden cuándo permitir que los datos entren, salgan o se eliminen a través del proceso

iterativo de hacer conjeturas, volver a propagar el error y ajustar los pesos mediante el descenso del gradiente.

El siguiente diagrama<sup>2</sup> ilustra cómo los datos fluyen a través de una celda de memoria y están controlados por sus puertas:

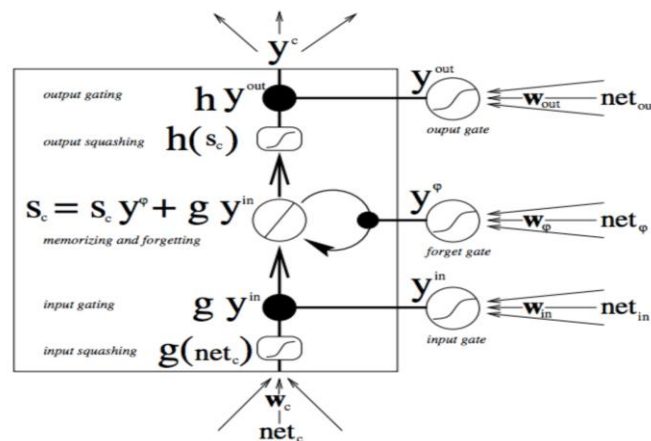


Ilustración 5.2 Representación de una celda de memoria en LSTM

En la ilustración superior, comenzando desde abajo, las flechas triples muestran donde la información fluye hacia la celda en múltiples puntos. Esa combinación de la entrada actual y el estado de celda pasado se alimenta no solo a la celda en sí, sino también a cada una de sus tres puertas, que decidirá cómo se manejará la entrada.

Los puntos negros son las puertas, que determinan respectivamente si se permite la entrada nueva, se borra el estado actual de la celda y / o se deja que ese estado afecte la salida de la red en el paso actual.  $s_c$  es el estado actual de la celda de memoria, y  $g_{y_{in}}$  es la entrada actual.

### 5.3.1 Parámetros de una red LSTM

En cualquier red neuronal, existen una serie de hiperparámetros que el programador debe ajustar.

<sup>2</sup> <https://deeplearning4j.org/lstm.html>

En concreto, los hiperparámetros que se pueden ajustar en las redes neuronales son los siguientes:

- **Número de épocas:** el aprendizaje en este estudio se realiza por épocas. La red se hace evolucionar durante un período de tiempo (época) en el que se le van pasando distintos batch del conjunto de datos. Una vez llegado al final de una época la red ya habrá entrenado con todos los datos y habrá alcanzado un determinado error que se intentará reducir en las siguientes épocas.
- **Tamaño del batch:** La entrada de la red es un vector de números. El lote de estos datos se conoce con el nombre de batch, es un parámetro importante de la red puesto que puede determinar el tiempo de ejecución de la red, pero también puede variar la calidad de las predicciones de la misma.
- **Número de neuronas:** El número de neuronas que debe tener la red no es un dato que se sepa de anteaño, depende de los datos de entrada. Por ello es necesario realizar una pequeña experimentación con el número de neuronas a emplear para optimizar el funcionamiento de la red.
- **Tamaño de ventana:** el tamaño de ventana se refiere a la longitud del vector de entrada de la red, es decir, al número de datos que tomará la red y que evaluará para predecir los datos futuros.

### 5.3.2 Entrenamiento de una red LSTM

Entrenar una red es un proceso que modifica el valor de los pesos asociados a cada neurona con el fin de que la red pueda, a partir de unos datos presentados en la entrada, generar una salida. En el caso de aprendizaje supervisado, se tiene un conjunto de datos pero no se conoce la función o relación matemática que representa; al propagar hacia delante cada uno de estos patrones, se obtiene una respuesta en la salida de la red, la cual se compara con la salida deseada, permitiendo obtener el error del desempeño de la red.

## 6 Análisis

El sistema sobre el que se sustenta la página web tiene tres partes diferenciadas una de otra, como son el preprocesamiento, la predicción y la obtención de resultados. Para el preprocesamiento se utilizan diferentes técnicas para preparar los datos con el fin de elaborar las predicciones. En la segunda, dispondremos de los métodos más usuales para realizar diferentes predicciones según el modelo y los datos ajustados anteriormente, y en la última etapa tendremos la visualización de resultados, la cual es la etapa final del proceso.

### 6.1 Análisis de requerimientos

El propósito de este software es el análisis y predicción de series temporales. Este apartado consistirá en lista una serie de requisitos para explicar de manera cómoda y sencilla cómo funciona el sistema.

#### 6.1.1 Requerimientos funcionales

Los requerimientos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que este reaccionará a entradas particulares del usuario. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer. En nuestro sistema son los siguientes:

- Proporcionar una herramienta para el procesamiento y la predicción de series temporales.
- Proporcionar distintos métodos para preprocesar los datos de manera que las predicciones sean las más acertadas posibles.
- Proporcionar una herramienta para crear modelos predictivos , con métodos como ARIMA o basado en LSTM.
- Mostrar mensajes de error si los datos son introducidos de forma errónea.
- Proporcionar herramientas para visualizar los resultados.
- Proporcionar herramientas para guardar los resultados.
- Proporcionar herramientas para guardas los proyectos realizados.



### 6.1.2 Requerimientos no funcionales

Los requisitos no funcionales se tratan de requisitos normalmente relacionados con aspectos técnicos o legales: comunicaciones, interfaces con otros sistemas, fiabilidad, entorno de desarrollo, portabilidad, etc.

Suelen especificar las necesidades software, hardware, interfaz. A continuación, detallaremos las necesidades de las tres partes que componen nuestra aplicación. En nuestro sistema encontramos los siguientes:

- El proceso debe ser lo más automático posible, de modo que un nuevo usuario pueda realizar predicciones de manera cómoda y sencilla.
- Las salidas y las entradas de texto deben de ser sencillas e intuitivas.
- En cada vista se añadirá la respectiva información sobre lo que se puede realizar en la misma para una mejor uso de la aplicación.
- Permitir la posterior edición e inclusión de nuevos métodos en el sistema. El código debe ser claro para facilitar el trabajo de terceras personas sobre el paquete.
- Los nombres de las funciones y objetos deben describir a primera vista la funcionalidad de las mismas.

## 6.2 Planificación

En la realización de este trabajo fin de grado se han empleado aproximadamente unos 6 meses, desde enero hasta junio de 2018.

Realizar una planificación del tiempo que vamos a emplear es importante, ya que, en caso de un proyecto real nos ayudaría a estimar el presupuesto y los recursos que se van a emplear.

| Tarea                            | Fecha Inicio | Duración | Fecha Fin  |
|----------------------------------|--------------|----------|------------|
| <b>Estudio series temporales</b> | 30-01-2018   | 9        | 7-02-2018  |
| <b>Estudio deep learning</b>     | 9-02-2018    | 5        | 14-02-2018 |
| <b>Instalación software</b>      | 15-02-2018   | 1        | 16-02-2018 |
| <b>Aprendizaje software</b>      | 17-02-2018   | 5        | 22-02-2018 |

|                                       |            |     |            |
|---------------------------------------|------------|-----|------------|
| <b>Diseño esqueleto web</b>           | 24-03-2018 | 4   | 28-03-2018 |
| <b>Módulo preprocesamiento</b>        | 1-03-2018  | 5   | 6-03-2018  |
| <b>Módulo predicción</b>              | 7-03-2018  | 68  | 24-05-2018 |
| <b>Módulo obtención de resultados</b> | 29-05-2018 | 9   | 2-06-2018  |
| <b>Experimentación</b>                | 29-05-2018 | 9   | 2-06-2018  |
| <b>Redacción memoria</b>              | 4-03-2018  | 107 | 20-06-2018 |

A continuación el diagrama de Gantt correspondiente:

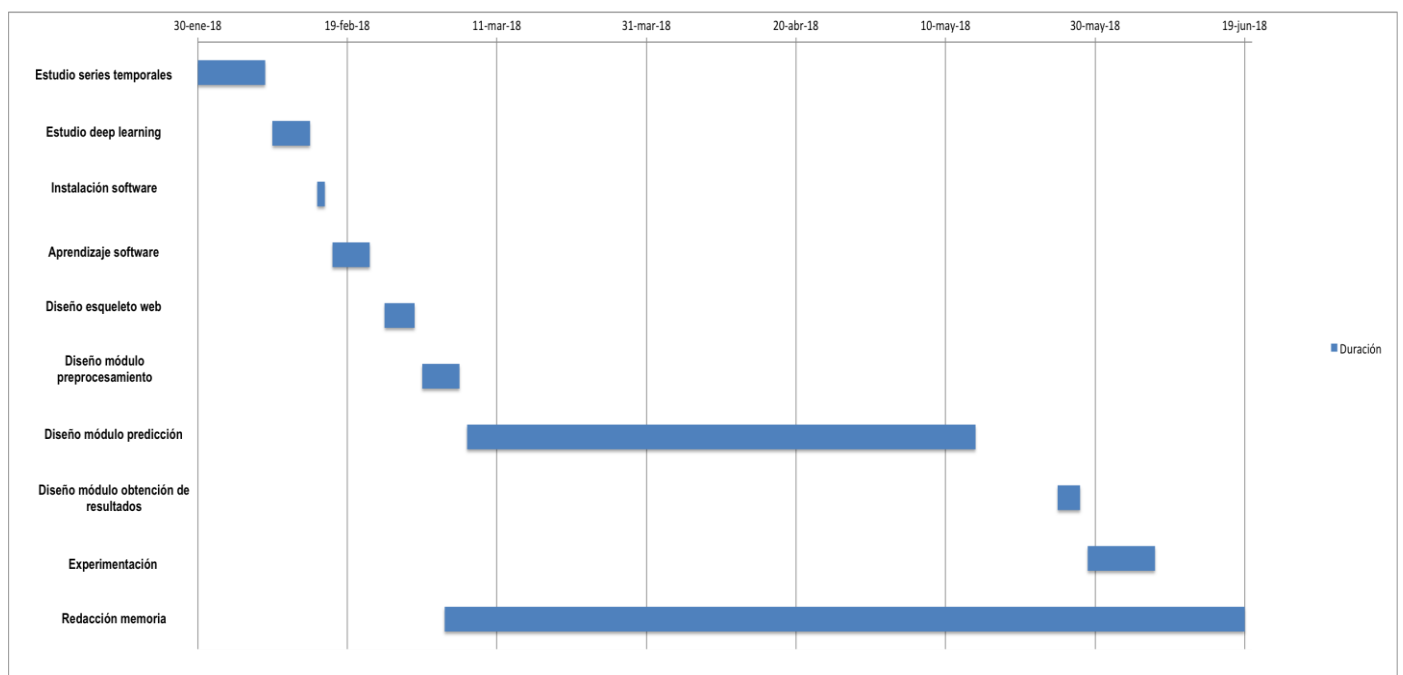


Ilustración 6.1 Diagrama de Gantt del proyecto

### 6.3 Casos de uso

La técnica de los casos de uso nos permite entender y describir los requisitos. Los casos de uso son requisitos funcionales que ponen el acento en el uso del producto.

Los casos de uso describen como debe comportarse el sistema desde el punto de vista del usuario, sin especificar cómo debe hacerlo internamente. Es un formato simple para que los usuarios y desarrolladores puedan trabajar juntos.

Identificamos las siguientes partes en el proceso:

- Loguearse
- Crear proyecto
- Preprocesar proyecto
- Modelar proyecto
- Predecir proyecto
- Visualizar resultados
- Descargar resultados

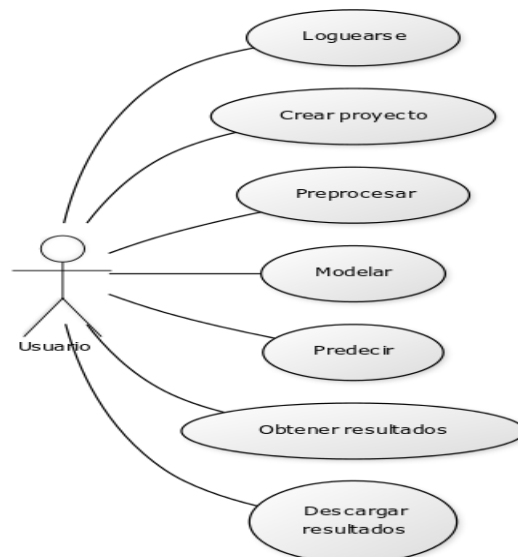


Ilustración 6.2 Diagrama de casos de uso

| Caso de uso   | Loguearse                                                                                                                                                                         |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Actor         | Usuario                                                                                                                                                                           |
| Precondición  | El usuario quiere iniciar sesión                                                                                                                                                  |
| Postcondición | El usuario puede crear un proyecto                                                                                                                                                |
| Secuencia     | <ol style="list-style-type: none"><li>1. El usuario introduce su nombre de usuario</li><li>2. El usuario introduce su contraseña</li><li>3. Usuario hace click en login</li></ol> |
| Excepciones   | El usuario introduce erróneamente su contraseña/usuario                                                                                                                           |

| Caso de uso   | Crear proyecto                                                                                                                                            |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Actor         | Usuario                                                                                                                                                   |
| Precondición  | El usuario tiene un conjunto de datos que quiere predecir.                                                                                                |
| Postcondición | El usuario crea un proyecto y puede empezar la etapa de preprocesado.                                                                                     |
| Secuencia     | <ol style="list-style-type: none"><li>1. Introduce nombre del proyecto</li><li>2. Introduce descripción del proyecto</li><li>3. Sube el archivo</li></ol> |
| Excepciones   | Proyecto con mismo nombre creado anteriormente.                                                                                                           |

| Caso de uso   | Preprocesar                                             |
|---------------|---------------------------------------------------------|
| Actor         | Usuario                                                 |
| Precondición  | El usuario debe de haber subido un fichero al proyecto. |
| Postcondición | El usuario obtiene un conjunto de datos preprocesado.   |
| Excepciones   | El fichero está corrupto/ tiene algún carácter extraño. |

| Caso de uso   | Modelar                                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Actor         | Usuario                                                                                                                                       |
| Precondición  | El usuario debe de tener un conjunto de datos preprocesado.                                                                                   |
| Postcondición | El usuario obtiene un modelo sobre el que realizar las predicciones.                                                                          |
| Secuencia     | <ol style="list-style-type: none"><li>1. Usuario elige una serie temporal a trabajar</li><li>2. Usuario modela el conjunto de datos</li></ol> |
| Excepciones   | Parámetros introducidos erróneamente                                                                                                          |

| Caso de uso   | Predecir                                                                                                                                                                |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Actor         | Usuario                                                                                                                                                                 |
| Precondición  | El usuario tiene un modelo sobre el que realizar las predicciones.                                                                                                      |
| Postcondición | El usuario obtiene resultados sobre dichas predicciones.                                                                                                                |
| Secuencia     | <ol style="list-style-type: none"><li>1. El usuario elige los parámetros a configurar.</li><li>2. El usuario elige el tipo de predicción que quiere realizar.</li></ol> |
| Excepciones   | Parámetros introducidos erróneamente                                                                                                                                    |

| Caso de uso   | Visualizar resultados                                                       |
|---------------|-----------------------------------------------------------------------------|
| Actor         | Usuario                                                                     |
| Precondición  | El usuario tiene un modelo sobre el que realizar las predicciones.          |
| Postcondición | El usuario obtiene gráficos y resultados sobre las predicciones realizadas. |

## 7 Diseño

En este apartado se detalla el esquema y el funcionamiento del proyecto. El mismo consta de tres módulos diferenciados, entre los que se encuentran el módulo de preprocesamiento, donde el usuario puede elegir el tipo de imputación y limpieza de los datos, así como técnicas para ajustar la estacionalidad, tendencia y varianza, el de predicción, dónde se presentan los diferentes métodos para realizar las predicciones, en este caso aplicando ARIMA o LSTM, y el módulo de obtención y visualización de resultados, dónde el usuario puede ver mediante gráficos, tablas y ficheros de datos los resultados de las etapas anteriores.

A continuación se muestra un diagrama en el que se engloba lo comentado anteriormente:

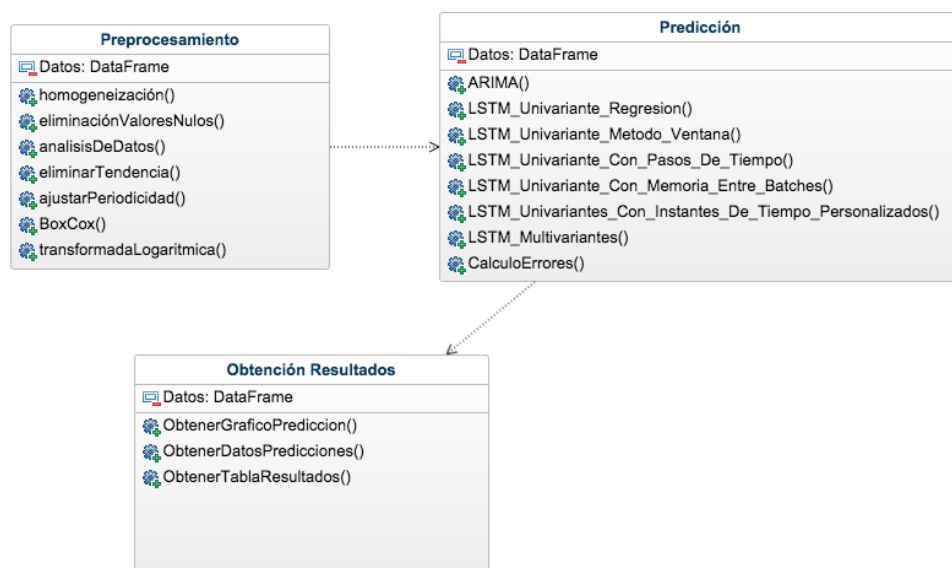


Ilustración 7.1 Diagrama del sistema completo

## 7.1 Preprocesamiento

El sistema cuenta con una etapa de preprocesamiento, en la cual se realizan varias tareas de imputación y limpieza de datos. Es importante que los datos tengan un formato apropiado para poder trabajar con ellos correctamente, como estar indexados por un índice temporal, o no contener un pie de referencia al final del fichero, indicando de donde se ha extraído la información, por ejemplo. A continuación se detallan las operaciones que se pueden realizar en esta etapa de preprocesamiento.

### 7.1.1 Homogeneización de datos

La homogeneización de datos tal y como se ha mencionado anteriormente, es una tarea importante para que los datos tengan un formato correcto y estándar. Por ejemplo, para el uso de fechas, unidades, métricas, etc.

### 7.1.2 Valores nulos o erróneos

Es importante la sustitución de valores nulos o erróneos (missing values u outliers). A la hora de recoger la información, se pueden producir errores en la medición, ruido, o de cualquier otro tipo. En cualquier caso, estos datos anómalos pueden provocar errores a la hora de realizar nuestros análisis, por lo que es de vital importancia tratarlos. Existen diversas formas de tratarlos, los más usual es sustituir esos datos anómalos o faltantes por valores calculados gracias a los demás datos, como pueden ser la media, la mediana, la moda, o por algún valor precalculado usando un modelo en particular. Otra opción, pero menos recomendable es directamente eliminar la información asociada a dicho valor. (En nuestro caso, la fila del fichero que presenta un valor erróneo).

El sistema provee de herramientas para realizar dichas sustituciones, como pueden ser sustituir los valores anómalos por valores estimados, como pueden ser la media, la moda o la mediana.

|   | one       | two       | three     |
|---|-----------|-----------|-----------|
| a | 1.552024  | -0.080851 | 0.805559  |
| b | NaN       | NaN       | NaN       |
| c | -1.136328 | 1.657122  | -1.402466 |
| d | NaN       | NaN       | NaN       |
| e | 1.587220  | -0.478010 | 1.416437  |

|   | one       | two       | three     |
|---|-----------|-----------|-----------|
| a | 1.552024  | -0.080851 | 0.805559  |
| c | -1.136328 | 1.657122  | -1.402466 |
| e | 1.587220  | -0.478010 | 1.416437  |

Ilustración 7.2 Valores de un conjunto de datos

En la ilustración superior, a la izquierda podemos ver un ejemplo de valores faltantes, denominados NaN(Not a number), y en la derecha el mismo conjunto de datos pero con dichos valores eliminados.

### 7.1.3 Análisis y limpieza de datos

Es importante tener un resumen de los datos para posteriormente, en función de los mismos, decidir qué estrategia de las mencionadas anteriormente es la adecuada para nuestro conjunto de datos, por lo que el sistema muestra justo al cargar los datos información acerca de estos, como son los valores máximos y mínimos, el valor medio, la media, desviación típica, etc. El sistema muestra en modo de tabla el resumen anterior para que el usuario nada más introducir los datos tenga una vista general de los datos sobre los que se está tratando.

Mostrando número de valores, media, desviación,máximo,mínimo y percentiles por serie temporal:

|       | Open               | High               | Low                | Close              | Volume             |
|-------|--------------------|--------------------|--------------------|--------------------|--------------------|
| count | 250.0              | 250.0              | 251.0              | 250.0              | 250.0              |
| mean  | 125.27396000000002 | 126.11319999999995 | 124.60282868525897 | 125.40408000000005 | 30487642.796       |
| std   | 17.294411449597003 | 17.27041538441391  | 17.15767157765103  | 17.207486093681933 | 14201408.102509603 |
| min   | 96.75              | 97.65              | 96.42              | 96.67              | 11475922.0         |
| 25%   | 110.465            | 111.5625           | 109.68             | 111.03750000000001 | 22100800.0         |
| 50%   | 117.95             | 118.525            | 117.45             | 118.08             | 26473697.5         |
| 75%   | 142.35750000000002 | 143.24249999999998 | 141.505            | 142.27             | 33978014.5         |
| max   | 156.01             | 156.65             | 155.05             | 156.1              | 112340318.0        |

Ilustración 7.3 Características generales de un fichero de datos



En la ilustración superior se puede observar una captura de imagen del preprocesamiento de la plataforma, los valores comentados anteriormente para cada serie de tiempo del fichero, en este caso representa los valores de apertura, más altos, más bajos, de cierre y de volumen de las acciones de APPLE.

El sistema detecta también la presencia de valores nulos o anómalos, comentados anteriormente, y brinda al usuario la capacidad de eliminarlos o sustituirlos por otros valores del modelo. Como se puede observar en la imagen a continuación:

Valores nulos del fichero : 4

Los nombres de las columnas del fichero: **Date - Open - High - Low - Close - Volume -**

Elija el tipo de imputación que quiere realizar, puede elegir entre sustituir los valores nulos por la media, la mediana o la moda:

✓ Mean value

Median value

Mode value

Drop Nan

Analysis

Submit

tipo de imputación, puede comenzar con el análisis:

Ilustración 7.4 Limpieza de datos

#### 7.1.4 Transformación de los datos

La transformación de los datos es otra opción que presenta el sistema, mediante diferentes técnicas explicadas anteriormente, como la diferenciación o el ajuste según el modelo, permite ajustar determinados componentes de las series de tiempo, como pueden ser la periodicidad, la tendencia, la varianza, etc. Es importante realizar un ajuste adecuado de dichas características para que el modelo sea lo más representativo posible y se ajuste cómodamente a nuestras exigencias.

El sistema permite realizar un ajuste de la periodicidad y la tendencia por medio de la diferenciación, el cual resulta muy eficaz, y mediante ajuste de modelo, en la cual toma como entrada los valores de la etapa anterior y realiza un ajuste de la tendencia.

Después de realizar las transformaciones comentadas anteriormente, se muestra el test de Dickey Fuller, una herramienta comentada anteriormente para

estudiar aspectos clave de nuestra serie temporal, como la periodicidad por ejemplo, para comprobar la periodicidad de la misma y si da pie a usar modelos de minería de datos como ARIMA.

Resultados Test Dickey-Fuller para la serie de tiempo Open

|                            |                     |
|----------------------------|---------------------|
| Test Statistic             | -0.5407825023394939 |
| P-value                    | 0.8838093928415338  |
| # Lags Used                | 0.0                 |
| Number of observation used | 250.0               |
| Critical value (1%)        | -3.456780859712     |
| Critical value (5%)        | -2.8731715065600003 |
| Critical value (10%)       | -2.572968544        |

**Ilustración 7.5 Resultados del test de Dickey Fuller para la serie temporal de apertura de los valores de las acciones de APPLE**

### 7.1.5 Normalización de datos

La normalización es un cambio de escala de los datos del rango original de modo que todos los valores estén dentro del rango de 0 y 1. La normalización requiere que sepa o pueda estimar con precisión los valores observables mínimo y máximo. Se pueden estimar estos valores a partir de los datos disponibles. Si su serie está subiendo o bajando, estimar estos valores esperados puede ser difícil y la normalización puede no ser el mejor método para usar en su problema. Si un valor a escalar está fuera de los límites de los valores mínimo y máximo, el valor resultante no estará en el rango de 0 y 1.

Este proceso se lleva a cabo a la hora de realizar las predicciones con Deep learning, ya que resulta más cómodo y proporciona mejores resultados que los datos estén normalizados.

## 7.2 Predicción

La fase de predicción de datos es uno de los componentes principales del sistema, dónde se le permite al usuario realizar predicciones, ya sea usando un modelo econométrico ARIMA o mediante LSTM.

Esta fase tiene como entrada los datos preprocesados en la etapa anterior y devolverá unos resultados que recogerá la etapa de visualización y obtención de resultados para mostrársela al usuario.

Queda detallado a continuación el funcionamiento de cada una de las opciones previstas por la plataforma para la elaboración de las predicciones, así como de una imagen del sistema real donde se pueden visualizar los resultados.

### 7.2.1 ARIMA

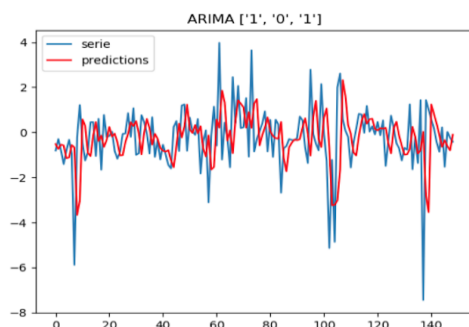
Respecto a **ARIMA**, el cual hemos comentado anteriormente, para realizar la predicción es necesario que la serie sea periódica, es decir, se ajuste en el tiempo y sea estacionaria, una vez satisfecho dicho requerimiento, se permite aplicar el modelo e introducir manualmente los parámetros de configuración para el ARIMA , los cuales son:

- **p**: El número de observaciones de retraso incluidas en el modelo, también llamado orden de retraso.
- **d**: El número de veces que las observaciones son diferenciadas, también llamado el grado de diferenciación.
- **q**: El tamaño de la ventana de promedio móvil, también llamado orden de promedio móvil.

Estos valores son configurables por el usuario, y los puede introducir manualmente desde teclado. La salida que produce es una gráfica en la que se representa la predicción realizada y una tabla con los diferentes valores que se obtienen en el modelo ARIMA. Adicionalmente, el sistema guarda en un fichero en formato csv los valores esperados y predichos para que el usuario se los pueda descargar y trabajar en cualquier momento con dicha salida obtenida.

**Resumen modelo Arima:**

|              | pvals                 | coeff                 | conf_lower           | conf_higher          | tvalues              |
|--------------|-----------------------|-----------------------|----------------------|----------------------|----------------------|
| const        | 0.9760374683472186    | -0.002975473700711861 | -0.1969312466623101  | 0.1909802992608864   | -0.03006778917323598 |
| ar.L1.D.Open | 5.331751647518709e-05 | -0.25760279013321036  | -0.38035329654813776 | -0.13485228371828295 | -4.113157702758964   |



| Errores                                    | Valor              |
|--------------------------------------------|--------------------|
| Test RMSE (Root Mean Squared Error)        | 1.7375173357280564 |
| Test MSE (Mean Squared Error)              | 3.0189664919555232 |
| Test MAPE (Mean Absolute Percentage Error) | 440.5480102348515  |

**Ilustración 7.6 Resultados modelo ARIMA para la serie de tiempo de apertura de los valores de las acciones de APPLE**

En la ilustración superior se puede observar la tabla resultante que genera el modelo ARIMA, el gráfico de predicción (en rojo) sobre la serie temporal (en azul), muestra también los errores MSE, RMSE y MAPE.

## 7.2.2 LSTM (Long Short-Term Memory)

En este apartado, detallamos los tipos de LSTM implementadas en nuestra plataforma. Cada una de las cuales funciona de una forma distinta y tiene como entrada distintos parámetros, ya que se han utilizado diversos métodos a la hora de realizar las predicciones, como el método de la ventana, o si son univariantes (se usa una variable para predecirse a si misma) o multivariantes (se usan diferentes variables para predecir otra, incluido así misma).

### 7.2.2.1 Red LSTM Univariante para regresión

Podemos expresar casi cualquier problema como un problema de regresión, por ejemplo; dada la cantidad de pasajeros este mes, ¿cuál es el número de pasajeros el próximo mes?

La plataforma modifica el conjunto de datos para convertir el conjunto de datos original en un conjunto de datos con una columna de más, el instante a predecir, en el ejemplo comentado anteriormente: la primera columna que contendría el número

de pasajeros de este mes y la segunda columna que contendría el número de pasajeros del mes siguiente ( $t + 1$ ), a pronosticar.

El usuario también puede configurar más datos de la red, como el porcentaje del conjunto de datos que se usa como entrenamiento y como test, el número de épocas que el proceso va a llevar a cabo, el número de neuronas, el tamaño del lote así como el número de lags.

Posteriormente, una vez recibidos los parámetros introducidos por el usuario, el sistema escala los datos. Esto es debido a que a las redes neuronales les resulta más fácil operar con un número en el rango  $(-1,1)$  como podría ser 0.53, que con 20000442.75, por ejemplo.

Después de esto, el sistema realiza los cálculos necesarios e inicia la predicción, la salida resultante es un gráfico en el que se muestra la serie original, la parte que ha usado como entrenamiento y la predicción realizada. También muestra, al igual que con ARIMA, errores MSE, RMSE y MAPE, también se le da la opción al usuario de descargar los ficheros de predicciones para que después pueda manipularlos o almacenarlos.

#### 7.2.2.2 Red LSTM Univariante para regresión utilizando el método de la ventana

También podemos expresar el problema para que se puedan usar múltiples pasos de tiempo recientes para hacer la predicción para el siguiente paso.

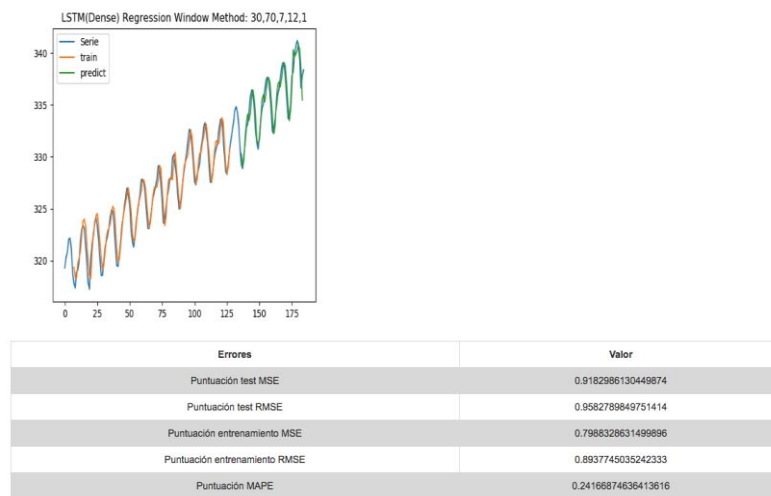
Esto se llama una ventana y el tamaño de la ventana es un parámetro que se puede ajustar para cada problema.

Por ejemplo, dado el tiempo actual ( $t$ ) que queremos predecir el valor en la próxima vez en la secuencia ( $t + 1$ ), podemos usar la hora actual ( $t$ ), así como las dos veces anteriores ( $t-1$ ) y ( $t-2$ ) como variables de entrada. Cuando se expresa como un problema de regresión, las variables de entrada son  $t-2$ ,  $t-1$ ,  $t$  y la variable de salida es  $t + 1$ .

Al igual que antes, el usuario puede configurar los datos de la red que desee, como el número de épocas, la longitud de entrenamiento y test, el número de pasos

de tiempo previos para usar como valores de entrada para predecir el próximo período de tiempo, el número de neuronas o el tamaño del batch.

Mostrando gráfico predicción por LSTM:



**Ilustración 7.7 Resultados LSTM para regresión utilizando el método de la ventana para la serie de tiempo de los valores de apertura de las acciones de APPLE**

Como podemos observar en la ilustración superior, las predicciones realizadas en verde, resultan muy buenas, ya que se ajustan a los valores de la serie original. Para este ejemplo concreto hemos utilizado los siguientes parámetros:

- 30 épocas
- 70 % del conjunto como entrenamiento
- Hemos usado 7 instantes de tiempo anteriores para calcular el siguiente
- 12 neuronas
- Tamaño de batch 1

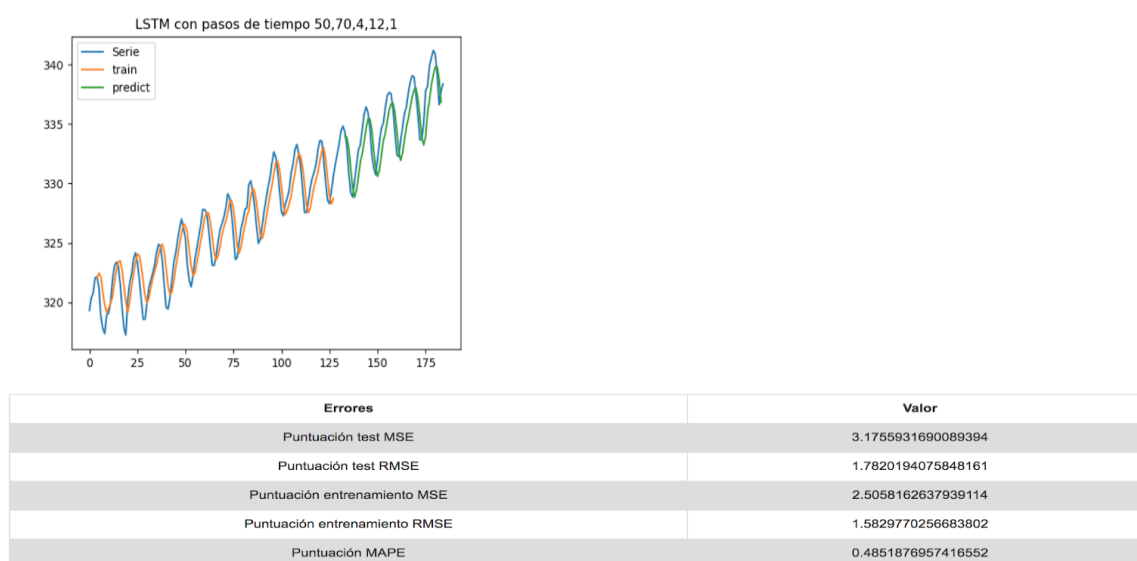
#### 7.2.2.3 Red LSTM Univariante para regresión usando pasos de tiempo

Algunos problemas de secuencia pueden tener un número variado de pasos de tiempo por muestra. Por ejemplo, puede tener medidas de una máquina física que conducen a un punto de fallo. Cada incidente sería una muestra de las observaciones que conducen al evento, las cuales serían los pasos de tiempo, y las variables observadas, las características.

Los pasos del tiempo proporcionan otra forma de expresar nuestro problema de series de tiempo. Al igual que en el ejemplo de la ventana anterior, podemos tomar pasos de tiempo anteriores en nuestra serie temporal como entradas para predecir el resultado en el siguiente paso.

En lugar de redactar las observaciones anteriores como funciones de entrada separadas, podemos usarlas como pasos de tiempo de la función de entrada única, que de hecho es un encuadre más preciso del problema.

Mostrando gráfico predicción por LSTM:



**Ilustración 7.8 Resultados LSTM usando regresión con pasos de tiempo para la serie temporal de las acciones de APPLE**

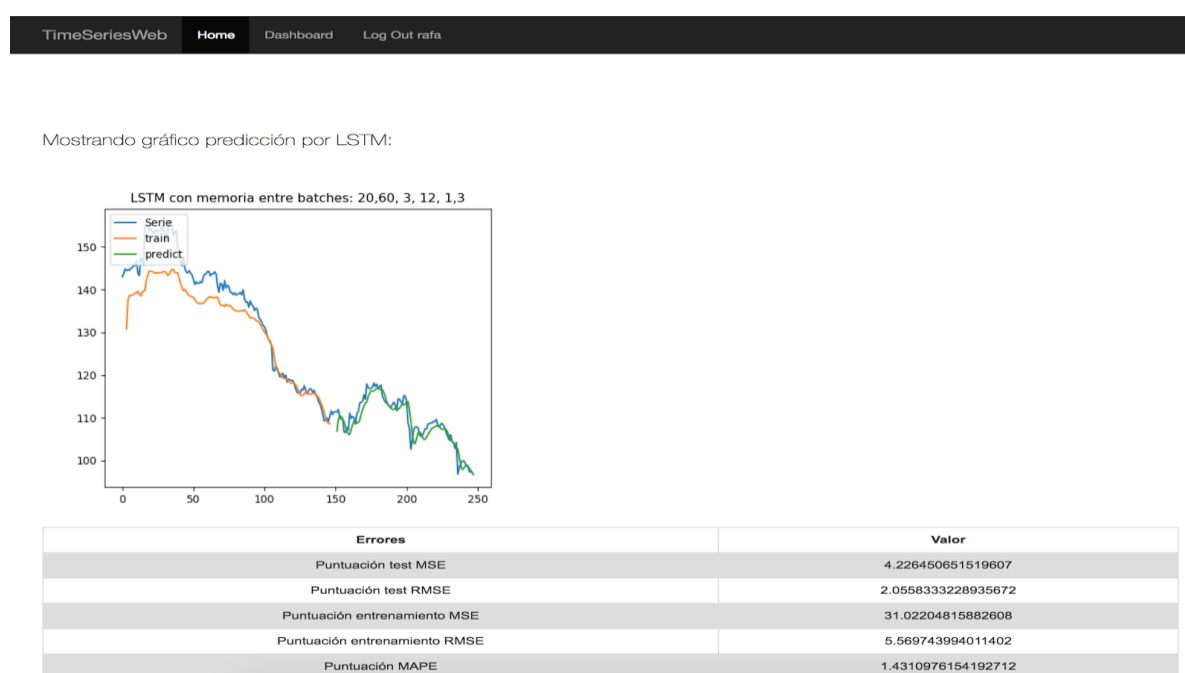
Como podemos observar en la ilustración superior, las predicciones realizadas en verde, resultan muy buenas, ya que se ajustan a los valores de la serie original. Para este ejemplo concreto hemos utilizado los siguientes parámetros:

- 50 épocas
- 70 % del conjunto como entrenamiento
- Hemos usado 4 instantes de tiempo anteriores para calcular el siguiente
- 12 neuronas
- Tamaño de batch 1

#### 7.2.2.4 Red LSTM Univariante con memoria entre batches

Las redes LSTM tienen memoria, la cual es capaz de recordar en largas secuencias. Normalmente, el estado dentro de la red se restablece después de cada lote de entrenamiento cuando se ajusta el modelo, así como también, de cada llamada para predecir o evaluar el modelo.

Se puede obtener un control más preciso sobre cuándo se borra el estado interno de la red LSTM haciendo que la capa sea "con estado". Esto significa que se puede construir el estado en toda la secuencia de entrenamiento e incluso mantener ese estado si es necesario para hacer predicciones.



**Ilustración 7.9 Resultados LSTM usando memoria entre batches para los valores de apertura de las acciones de APPLE**

Como podemos observar en la ilustración superior, las predicciones realizadas en verde, resultan muy buenas, a pesar de que en la etapa de entrenamiento, en naranja, le hayan costado al principio. Para este ejemplo concreto hemos utilizado los siguientes parámetros:

- 20 épocas
- 60 % del conjunto como entrenamiento
- Hemos usado 3 instantes de tiempo anteriores para calcular el siguiente



- 12 neuronas
- Tamaño de batch 1
- Número de bucles: 3

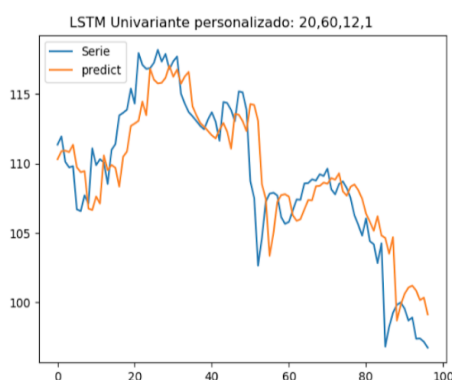
#### 7.2.2.5 Red LSTM Univariante con instantes de tiempo personalizados

En esta configuración de la red LSTM, a diferencia de las anteriores, donde definíamos un valor para usar los “x” instantes de tiempos anteriores, se pueden elegir manualmente qué valores anteriores queremos usar para predecir la salida. Por lo que es más flexible ya que si conocemos mejor cómo funcionan los datos, quizás nos interese más usar unos instantes de tiempo concretos, en vez de una ventana en la que se cogían todos en dicho intervalo como se hacía en los ejemplos anteriores.

Elija la serie temporal con la que se va a trabajar y visualizar resultados:

Open

Mostrando gráfico predicción por LSTM Univariante personalizada:



| Errores              | Valor              |
|----------------------|--------------------|
| Puntuación test MSE  | 6.9784484          |
| Puntuación test RMSE | 2.6416752          |
| Puntuación MAPE      | 1.8352469429373741 |

**Ilustración 7.10 Resultados LSTM Univariante con instantes de tiempo personalizados para la serie temporal de los valores de apertura de las acciones de APPLE**

Configuración usada para el ejemplo de la imagen superior:

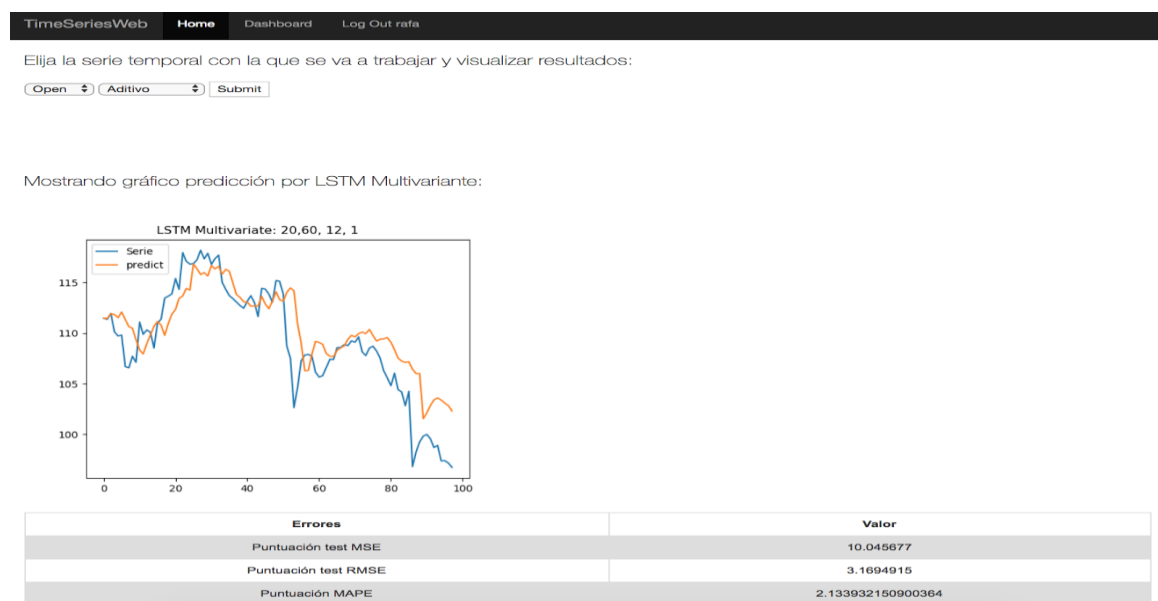
- 20 épocas
- 60 % del conjunto como entrenamiento
- Hemos usado los instantes de tiempo anteriores  $t-3$  y  $t-1$  para calcular  $t$

- 12 neuronas
- Tamaño de batch 1

#### 7.2.2.6 Red LSTM Multivariante

A diferencia de las redes LSTM comentadas anteriormente, donde, con instantes de tiempo anteriores de la misma variable pretendíamos predecir el siguiente instante de tiempo, con esta configuración, pretendemos predecir una variable “x” haciendo uso de ella misma (o no) y de otras variables que dependan de ella, por eso a este tipo de redes se les llama multivariantes.

En la plataforma están implementadas de manera que, una vez cargado el conjunto de datos de trabajo, introducimos la variable a predecir, qué variables queremos usar para la predicción y qué instantes de tiempo de las variables de entrada queremos usar para predecir la salida.



**Ilustración 7.11 Resultados LSTM Multivariante para la serie temporal de los valores de las acciones de APPLE**

Configuración usada para el ejemplo de la imagen superior:

- 20 épocas
- 60 % del conjunto como entrenamiento
- Hemos usado los instantes de tiempo anteriores t-2 para calcular t
- 12 neuronas

- Tamaño de batch 1
- Variable a predecir: "Open"
- Variables usadas: "Open, High, Low, Close".

## 7.3 Obtención y visualización de resultados

Para poder comprobar las predicciones y las transformaciones realizadas anteriormente, es importante poder visualizar los resultados correctamente, para ello se proporcionan varias formas de visualizarlos, mediante tablas, gráficos y ficheros csv.

### 7.3.1 Obtención y visualización de resultados

Respecto a la parte de análisis de datos, se han tenido en cuenta que estamos en una plataforma web, por lo que es importante proporcionar al usuario retroalimentación cada vez que realice una acción, por lo que cada vez que se realiza una transformación de datos, el sistema muestra al instante en forma de gráfica o tabla la transformación realizada. Así mismo, cada vez que se realiza una predicción, después de la etapa de cómputo, el sistema muestra inmediatamente en forma de gráfica y tabla las predicciones realizadas, además, guarda las predicciones en formato csv para que posteriormente el usuario pueda descargar los datos y usarlos con los fines que estime oportuno.

### 7.3.2 Resultados mediante gráficos

Como se ha podido ver anteriormente en las ilustraciones adjuntadas, el sistema muestra las predicciones realizadas mediante gráficos, la cual es una manera muy sencilla y cómoda de que el usuario pueda interpretar los resultados.

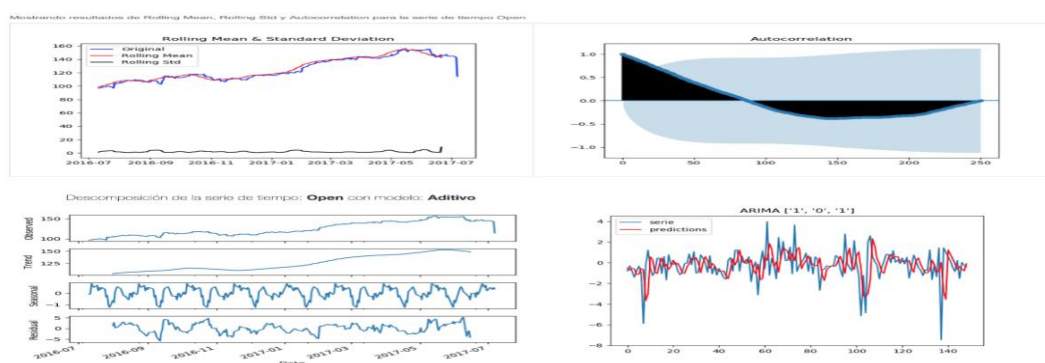


Ilustración 7.12 Ejemplo de gráficos de la plataforma web

### 7.3.3 Resultados mediante tablas

Al igual que con el apartado anterior de gráficos, el sistema muestra tablas con errores de las predicciones calculadas, o información sobre el conjunto de datos, así como resultados de test realizados, para que el usuario los pueda visualizar cómodamente.

Mostrando número de valores, media, desviación, máximo, mínimo y percentiles por serie temporal:

|       | CO2                |
|-------|--------------------|
| count | 185.0              |
| mean  | 328.49767567567574 |
| std   | 5.883286339717931  |
| min   | 317.25             |
| 25%   | 323.61             |
| 50%   | 328.31             |
| 75%   | 332.87             |
| max   | 341.19             |

Mostrando primeros 10 elementos del archivo:

| Month               | CO2    |
|---------------------|--------|
| 1965-01-01 00:00:00 | 319.32 |
| 1965-02-01 00:00:00 | 320.36 |
| 1965-03-01 00:00:00 | 320.82 |
| 1965-04-01 00:00:00 | 322.06 |
| 1965-05-01 00:00:00 | 322.17 |
| 1965-06-01 00:00:00 | NaT    |
| 1965-07-01 00:00:00 | 321.2  |
| 1965-08-01 00:00:00 | 318.81 |
| 1965-09-01 00:00:00 | 317.82 |
| 1965-10-01 00:00:00 | 317.37 |

Resultados Test Dickey-Fuller para la serie de tiempo CO2

|                            |                    |
|----------------------------|--------------------|
| Test Statistic             | 0.3713581866933478 |
| P-value                    | 0.9803916591629375 |
| # Lags Used                | 11.0               |
| Number of observation used | 180.0              |
| Critical value (1%)        | -3.4672111510631   |
| Critical value (5%)        | -2.877734766803841 |
| Critical value (10%)       | -2.575403364197531 |

| Errores                       | Valor              |
|-------------------------------|--------------------|
| Puntuación test MSE           | 4.235204843712967  |
| Puntuación test RMSE          | 2.0579613319285097 |
| Puntuación entrenamiento MSE  | 3.219706873865506  |
| Puntuación entrenamiento RMSE | 1.7943541662295952 |
| Puntuación MAPE               | 0.5653444328996845 |

**Ilustración 7.13** Tablas que aparecen en la plataforma

En la ilustración superior aparecen a modo de ejemplo, las diferentes tablas que se pueden visualizar en la página web, empezando por la primera, la cual es una tabla a modo de características generales del conjunto de datos subido por el usuario, la segunda, en la cual se muestran los primeros valores del conjunto de datos, para visualizar si tiene valores nulos o no, la tercera, resultados del test de Dickey Fuller para un conjunto de datos concreto, y por último, tabla que muestra los errores calculados para predicción de series temporales.

### 7.3.4 Resultados mediante archivos csv

Es importante de que la plataforma deje a disposición del usuario los ficheros con los que va trabajando, por lo que, se le permite al usuario descargar los ficheros modificados, tanto por la limpieza e imputación, como por el preprocesamiento o el para la predicción. Para este último paso, el archivo muestra los valores de prueba y esperados, junto al cálculo de diversos errores, como MSE, RMSE o MAPE, como se puede observar en la imagen siguiente.

| MAPE               | MSE                   | RMSE                | SMAPE               | observaciones        | predicciones         |
|--------------------|-----------------------|---------------------|---------------------|----------------------|----------------------|
| 55.187692713732886 | 0.0829736051409418    | 0.2880513932286074  | 0.43878380537866446 | -0.8100000000000023  | -0.521948607713949   |
| 59.10578637813613  | 0.1880090636334426    | 0.4336001195034921  | 1.0894669255908778  | -0.3000000000001137  | -0.7336001195035035  |
| 48.154503222190655 | 0.06931291094496783   | 0.2632734527918982  | 0.43587775063608747 | -0.8100000000000023  | -0.5467265472081041  |
| 142.28681411610296 | 0.6856586786132257    | 0.8280450945529632  | 0.5846104649763413  | -1.4099999999999966  | -0.5819549054470334  |
| 28.596695484488187 | 0.1078506542190175    | 0.32840623352643217 | 0.77445082013299    | -0.8199999999999932  | -1.1484062335264253  |
| 70.31642127366136  | 0.61109521024727      | 0.7817257896777297  | 1.740295909502097   | -0.3299999999999983  | -1.111725789677728   |
| 67.56097349459718  | 0.14672133108809712   | 0.3830422053613637  | 0.4850800580358262  | -0.9500000000000028  | -0.5669577946386392  |
| 775.0196452918509  | 27.123421148975748    | 5.208015087245403   | 0.7788508821634708  | -5.879999999999996   | -0.6719849127545934  |
| 93.999408536626    | 11.87701989309341     | 3.4463052524542004  | 9.66566456362305    | -0.21999999999999886 | -3.6663052524541992  |
| 139.95695762703662 | 17.962835840845372    | 4.238258585887059   | 3.2654758160373683  | 1.2099999999999935   | -3.0282585858870656  |
| 181.79096352911387 | 1.0912635520593554    | 1.0446356073097238  | 1.3986322795162733  | -0.4699999999999989  | 0.5746356073097248   |
| 529.2942278419757  | 2.3752221986698863    | 1.5411755898241726  | 0.7620580308417553  | -1.25                | 0.2911755898241725   |
| 4.826721210146718  | 0.0019467620530404294 | 0.04412212656978842 | 0.4824186072905219  | -0.8700000000000045  | -0.914122126569793   |
| 142.66702685605475 | 2.264057678668496     | 1.5046785964678622  | 2.1992044054204323  | 0.4500000000000029   | -1.0546785964678593  |
| 416.67391336918104 | 0.3505847917256681    | 0.5921020112494705  | 0.7289421292352523  | 0.4500000000000029   | -0.14210201124946759 |

| pvals                 | coeff                 | conf_lower           | conf_higher          | tvalues              |
|-----------------------|-----------------------|----------------------|----------------------|----------------------|
| 0.9760374683472186    | -0.002975473700711861 | -0.1969312466623101  | 0.1909802992608864   | -0.03006778917323598 |
| 5.331751647518709e-05 | -0.25760279013321036  | -0.38035329654813776 | -0.13485228371828295 | -4.113157702758964   |

Ilustración 7.14 Fichero de salida con valores de muestra y predicciones, así como diversos estimadores

En cuanto a la generación del archivo tras el proceso de limpieza e imputación, la comparación entre el archivo original y el modificado aparece en la siguiente imagen, el usuario puede descargarse ambos ficheros, tanto el original como el modificado:

| Date      | Open   | High   | Low    | Close  | Volume   | Date       | Open   | High   | Low    | Close  | Volume     |
|-----------|--------|--------|--------|--------|----------|------------|--------|--------|--------|--------|------------|
| 7-Jul-17  |        | 144.75 | 142.90 | 144.18 | 19201712 | 2017-07-06 | 143.02 | 143.5  | 142.41 | 142.73 | 24128782.0 |
| 6-Jul-17  | 143.02 | 143.50 | 142.41 | 142.73 | 24128782 | 2017-07-05 | 143.69 | 144.79 | 142.72 | 144.09 | 21569557.0 |
| 5-Jul-17  | 143.69 | 144.79 | 142.72 | 144.09 | 21569557 | 2017-07-03 | 144.88 | 145.3  | 143.1  | 143.5  | 14277848.0 |
| 3-Jul-17  | 144.88 | 145.30 | 143.10 | 143.50 | 14277848 | 2017-06-30 | 144.45 | 144.96 | 143.78 | 144.02 | 23024107.0 |
| 30-Jun-17 | 144.45 | 144.96 | 143.78 | 144.02 | 23024107 | 2017-06-29 | 144.71 | 145.13 | 142.28 | 143.68 | 31499368.0 |
| 29-Jun-17 | 144.71 | 145.13 | 142.28 | 143.68 | 31499368 | 2017-06-28 | 144.49 | 146.11 | 143.16 | 145.83 | 22082432.0 |
| 28-Jun-17 | 144.49 | 146.11 | 143.16 | 145.83 | 22082432 | 2017-06-27 | 145.01 | 146.16 | 143.62 | 143.73 | 24761891.0 |
| 27-Jun-17 | 145.01 | 146.16 | 143.62 | 143.73 | 24761891 | 2017-06-23 | 145.13 | 147.16 | 145.11 | 146.28 | 35439389.0 |
| 26-Jun-17 | 147.17 |        | 145.38 | 145.82 | 25692361 | 2017-06-22 | 145.77 | 146.7  | 145.12 | 145.63 | 19106294.0 |
| 23-Jun-17 | 145.13 | 147.16 | 145.11 | 146.28 | 35439389 | 2017-06-21 | 145.52 | 146.07 | 144.61 | 145.87 | 21265751.0 |
| 22-Jun-17 | 145.77 | 146.70 | 145.12 | 145.63 | 19106294 | 2017-06-20 | 146.87 | 146.87 | 144.94 | 145.01 | 24900073.0 |
| 21-Jun-17 | 145.52 | 146.07 | 144.61 | 145.87 | 21265751 |            |        |        |        |        |            |
| 20-Jun-17 | 146.87 | 146.87 | 144.94 | 145.01 | 24900073 |            |        |        |        |        |            |
| 19-Jun-17 | 143.66 | 146.74 | 143.66 |        |          |            |        |        |        |        |            |

**Ilustración 7.15 Conjunto de datos parcial antes y después de limpiar**

## 8 Implementación

En este apartado se detalla el funcionamiento de la plataforma web, así como el tipo de estructuras de datos usadas en el proyecto. Al ser una página web y utilizar Flask, un microframework en Python, el esquema resulta sencillo, ya que está pensado para proyectos de no muy larga extensión.

### 8.1 Esquema de la plataforma web

El esquema que sigue la página web es el que se encuentra en la figura 7.1, comentado anteriormente. Como se ha explicado en los apartados anteriores, el sistema consta de tres módulos: preprocesamiento, predicción y obtención de resultados. Cada uno realiza sus operaciones propias de dicho módulo y manda al siguiente los resultados de su etapa.

### 8.2 Estructuras de datos usadas

En este apartado se detallan las estructuras de datos usadas para el funcionamiento interno de la plataforma web.

#### 8.2.1 DataFrame

Los tipos de datos usados a lo largo del proyecto para trabajar con los ficheros que sube el usuario son del tipo DataFrame.

El concepto de DataFrame proviene del mundo del software estadístico utilizado en la investigación empírica; se refiere a datos "tabulares": una estructura de datos que representa casos (filas), cada uno de los cuales consiste en un número de observaciones o medidas (columnas). Alternativamente, cada fila puede tratarse como una observación única de múltiples "variables". En cualquier caso, cada fila y cada columna tienen el mismo tipo de datos, pero el tipo de datos de fila ("registro") puede ser heterogéneo (una tupla de diferentes tipos), mientras que el tipo de datos de columna debe ser homogéneo. Los marcos de datos generalmente contienen algunos metadatos además de los datos; por ejemplo, nombres de columnas y filas.

Por lo que al cargar el fichero de datos. El sistema automáticamente lo transforma al tipo de dato DataFrame, el cual es un tipo de dato sencillo de utilizar y entender, y que además, se utiliza mucho en el mundo del software estadístico, da

igual el lenguaje que se esté utilizando, R o Python. El usuario puede elegir con qué columna (serie temporal) del DataFrame trabajar y modificar una manteniendo intacta las demás, si así lo desea.

### **8.2.2 Arrays de numpy**

A la hora de realizar las predicciones con las redes neuronales, es necesario que se divida el conjunto de datos original en una parte de entrenamiento, y otra parte de test, es por ello que se usa este tipo de estructura, ya que además, es la forma de entrada clásica para los datos de la red neuronal.

Las predicciones obtenidas por el modelo también son devueltas en un array de numpy, por lo que es muy importante saber manejar bien estas estructuras.

## **8.3 Codificación**

En esta parte se detalla la implementación y el código que se ha elaborado para realizar este proyecto, así como las tecnologías y las bibliotecas usadas.

### **8.3.1 Tecnologías usadas**

Se detallan los lenguajes de programación así como las tecnologías usadas para la elaboración total del proyecto.

#### **8.3.1.1 Python**

Python ha sido elegido como el lenguaje de programación utilizado casi en la totalidad del proyecto, usando la versión 3 de Python. Es un lenguaje de programación interpretado basado en scripts. Este lenguaje está preparado para cualquier tarea, desde aplicaciones Windows a servidores de red. Es un lenguaje de programación que soporta la orientación a objetos y la programación imperativa. En los últimos años se ha hecho muy popular por:

- Tener una cantidad de librerías que contienen datos y funciones incorporados al propio lenguaje para realizar tareas sin tener que implementarlas desde cero.



- Su gran calidad de sintaxis. Un bloque de código se crea a través de indentaciones que obliga al desarrollador a garantizar una legibilidad notoria.
- Es un lenguaje interpretado. Esto quiere decir que no se debe compilar el código antes de su ejecución. Aunque en ciertos casos, cuando se ejecuta por primera vez, se producen unos bytecodes que sirven para acelerar la compilación implícita que realiza el intérprete cuando se ejecuta el código.
- Python es un lenguaje interactivo. Dispone de línea de comandos en la que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible
- Las herramientas para deep learning son nativas en esta plataforma y funcionan muy bien con las librerías comentadas en el apartado de *Bibliotecas y métodos usados*.

#### 8.3.1.2 Flask

Flask ha sido un componente principal de este proyecto, ya que todo el back-end está implementado en Python, opté por una herramienta que me pudiera generar páginas webs dinámicas de una manera relativamente sencilla. Flask es un microframework de Python, además de incluir un servidor web de desarrollo sin necesidad de instalar un componente extra como Apache, por ejemplo. También soporta cookies seguras y usa el motor Jinja2 para las vistas. Por lo que fue a mi parecer, es la herramienta más adecuada para la elaboración de dicho proyecto.

#### 8.3.1.3 JavaScript

Esta tecnología ha sido usada en el proyecto para poder crear funciones que mejoren la interfaz cliente/servidor. JavaScript es un lenguaje de programación interpretado, orientado a objetos, imperativo y dinámico. Es un lenguaje independiente de la plataforma. El programa creado puede ser ejecutado tanto en Linux, Windows, etc. Muchos confunden JavaScript con Java, pero son lenguajes distintos. JavaScript tiene la ventaja de poder ser incorporado a cualquier página y

ser utilizado sin la necesidad de instalar otro programa. Se puede decir que JavaScript es un lenguaje basado en prototipos mientras que Java es un lenguaje más orientado a objetos. Todos los navegadores actuales interpretan el código JavaScript que permite interactuar con una página que provee a JavaScript una implementación del DOM.

#### 8.3.1.4 Bootstrap

Bootstrap ha permitido crear páginas web responsive que se adapten a diferentes dispositivos. Es un framework para el front-end de código abierto para aplicaciones web. Esta herramienta contiene plantillas de HTML y CSS, como extensiones de JavaScript, la cual nos permite crear una web de manera rápida y sencilla ya que no es necesario crear todos estos ficheros. Otra de las ventajas de este recurso es que es compatible por la mayoría de los navegadores ya que se ha estandarizado.

### 8.3.2 Bibliotecas usadas

Para la realización de este trabajo fin de grado, se han usado diferentes bibliotecas de Python, para graficar, trabajar con formatos csv y las bibliotecas propias de deep learning.

Entre ellas se encuentran: Matplotlib, Statsmodels, Pandas y Keras, entre otras más básicas como Numpy:

#### 8.3.2.1 Keras

Keras es una librería de Python que de una manera limpia y sencilla, proporciona la creación de una gama de modelos de Deep Learning encima de otras librerías como TensorFlow o CNTK. Fue desarrollado por un ingeniero de Google y su código fue liberado bajo licencia del MIT, que utiliza cuatro principios:

- **Modularidad:** un modelo se puede entender como una secuencia o un gráfico solo. Todas las preocupaciones de un modelo de aprendizaje

profundo son componentes discretos que se pueden combinar de maneras arbitrarias.

- **Minimalismo:** la biblioteca proporciona lo suficiente para lograr un resultado, sin lujos y maximizando la legibilidad.
- **Extensibilidad:** los nuevos componentes son fáciles de agregar y usar dentro del marco, destinados a que los investigadores prueben y exploren nuevas ideas.
- **Python:** no hay archivos de modelo separados con formatos de archivo personalizados. Todo es nativo en Python.

La ventaja de este modelo de programación, es que permite expresar redes neuronales de una manera muy modular, considerando un modelo de una secuencia o un sólo grafo. Una vez construido dicho modelo, se puede configurar el proceso de aprendizaje, la compilación de dicho modelo utiliza librerías numéricas como TensorFlow o CNTK, comentadas anteriormente. Una vez definido el modelo, se puede entrenar con datos, mediante una simple llamada a la función *fit()*. Una vez entrenado, se puede evaluar y generar predicciones sobre nuevos datos, mediante una llamada a la función *predict()*.

#### 8.3.2.2 Matplotlib

También se ha usado Matplotlib, la cual es una biblioteca de trazados 2D python que produce figuras de calidad de publicación en una variedad de formatos impresos y entornos interactivos en todas las plataformas. Puede generar gráficos, histogramas, espectros de potencia, gráficos de barras, diagramas de errores, diagramas de dispersión, etc.

Es muy cómoda de usar, ya que permite representar casi cualquier tipo de gráfico de una manera muy sencilla. Mediante una simple llamada a la funciones *plot()* y *show()* nos permite dibujar y mostrar por la pantalla el gráfico requerido. Dicho gráfico también se puede almacenar en nuestro sistema, como archivo PNG por ejemplo, para posteriormente, poder generar reportes.

El modo de utilización de dicha biblioteca para la web consiste en almacenar la imagen PNG en un buffer de memoria que posteriormente se manda a la vista HTML para mostrarla al usuario.

#### 8.3.2.3 Statsmodel

Otra biblioteca usada es Statsmodel, un módulo de Python que proporciona clases y funciones para la estimación de muchos modelos estadísticos diferentes, así como para realizar pruebas estadísticas y explorar datos estadísticos. Hay una lista extensa de estadísticas de resultados la cual está disponible para cada estimador.

Entre los métodos utilizados se encuentran:

- `Seasonal_decompose`: descompone una serie de tiempo en sus cuatro componentes principales.
- `ACF`: muestra la correlación de una serie temporal.
- `Mean squared error`: calcula el error cuadrático medio.
- `MAPE`: calcula el porcentaje de error medio absoluto.
- `ARIMA`: la biblioteca también provee el modelo ARIMA sobre el que realizar predicciones.
- `Adfuller`: Calcula el test de Dickey – Fuller.

#### 8.3.2.4 Pandas

Pandas proporciona un amplio soporte para trabajar con datos de series de tiempo. Generalmente cuando trabajamos con series de tiempo realizamos un amplio abanico de tareas, cómo: convertir fechas, estandarizar el tiempo de acuerdo a la zona horaria, crear secuencias a determinados intervalos o frecuencias, identificar datos faltantes, desplazar las fechas hacia atrás o hacia adelante por un determinado valor, calcular resúmenes agregados de valores a medida que el tiempo cambia, etc. Pandas nos brinda las herramientas para poder realizar estas y muchas otras tareas en forma muy sencilla.

Respecto a los métodos disponibles, Pandas proviene de una buena cantidad de métodos preparados para usar.

Principalmente esta biblioteca se ha usado para trabajar con el tipo de dato DataFrame, el cual resulta muy cómodo para trabajar con series temporales, ya que resulta muy fácil importarlo y exportarlo desde/hacia formato Excel/csv y también porque su representación matricial se adecúa muy bien a las series temporales, al representar por columnas las distintas series que dispongamos en el fichero.

Algunos de los métodos utilizados en esta biblioteca:

- Read\_csv: Lee y carga en un objeto el fichero csv que entra como parámetro.
- To\_csv: Escribe en un fichero csv el dataframe.
- Dropnan: Elimina los valores nulos de un dataframe.
- Df.index: Asigna un index a un dataframe.
- To\_datetime: Convierte una fecha a formato “date time”.
- To\_Series: Convierte un dataframe a un objeto de tipo Serie.
- DatraFrame: Crea un objeto de tipo dataframe.

#### 8.3.2.5 Scikit-Learn

Finalmente, Scikit-Learn, que es una biblioteca de aprendizaje de máquina de software libre para el lenguaje de programación de Python. Cuenta con varios algoritmos de clasificación, regresión y agrupación, incluyendo máquinas de vectores de soporte, bosques aleatorios, aumento de gradiente, k-means y DBSCAN, y está diseñado para inter-operar con las bibliotecas numéricas y científicas de Python, NumPy y SciPy. Se han utilizado herramientas para realizar transformaciones en los conjuntos de datos, como puede ser Box-Cox, MSE, etc.

También permite definir un imputer, para realizar de manera más cómoda la imputación de los datos.

## 8.4 Estructura del proyecto

Al usar Flask, el proyecto mantiene una estructura bastante simple, como se puede ver en la siguiente imagen:

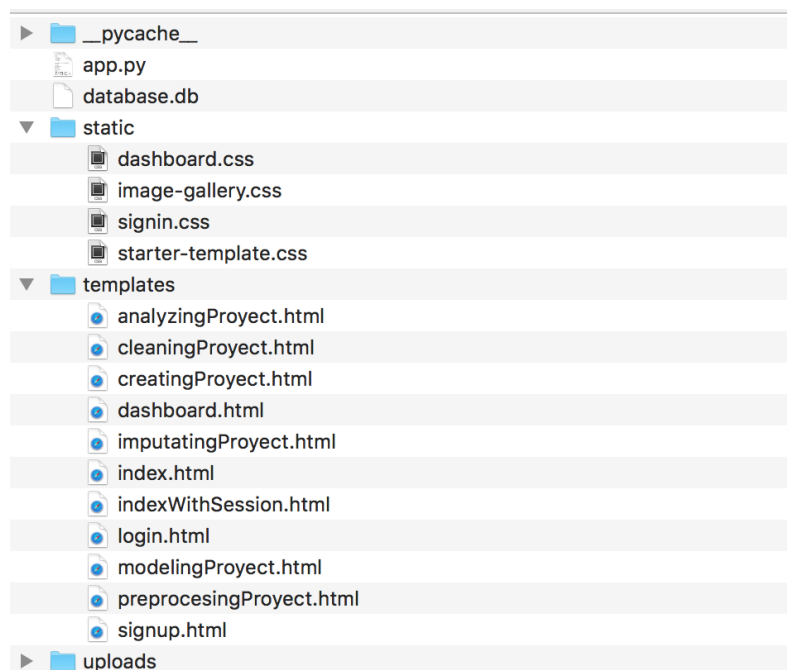


Ilustración 8.1 Estructura del proyecto

Tal y como se puede ver en la imagen superior, el proyecto está organizado en carpetas, las cuales son *static*, *templates* y *uploads*, estas carpetas son para los estilos *css*, las vistas *HTML* y los archivos que el usuario decida subir a la página web. Fuera de estas carpetas, y en el directorio raíz del proyecto se encuentran dos archivos muy importantes, el *app.py* y el *database.db*, los cuales gestionan las vistas de la página web y la base de datos correspondientemente.

## 8.5 Fichero principal

El fichero principal del proyecto es el *app.py*, en el cual está presente la lógica que controla las vistas y el código relacionado con el preprocesamiento, predicción y visualización de resultados. Es una de las comodidades que presenta Flask a la hora de construir aplicaciones web con objetivo científico.

Flask controla las vistas por medio de rutas, por lo que, cuando le llega una dirección, se activa la función implementada en dicha ruta, es decir, si el usuario se

dirige a la ventana login, la cual tiene una ruta /login definida por el programador, es la función a la que está definida dicha ruta la que se ejecuta mientras el usuario permanece en la vista.

Un ejemplo sencillo para la vista login:

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()

    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user:
            if check_password_hash(user.password, form.password.data):
                login_user(user, remember=form.remember.data)
                return redirect(url_for('dashboard', name=current_user.username))

        return '<h1>Invalid username or password</h1>'

    return render_template('login.html', form=form)
```

Ilustración 8.2 Ejemplo implementación rutas en Flask

Es obligatorio que, cada función asociada a cada ruta devuelva una vista HTML, en el *return* devolvemos la vista que el usuario va a visualizar así como variables que van a ser usadas en dicha vista, como por ejemplo el nombre de usuario, o una imagen de una gráfica.

## 9 Pruebas de la plataforma web

En esta sección se realiza un proyecto paso a paso en la plataforma, a modo de prueba, para que se pueda hacer un recorrido por todas las herramientas de la plataforma.

Vamos a trabajar con un fichero sobre las acciones de APPLE. El fichero se encuentra en formato csv, y debido a errores en la captura de los datos, hay valores anómalos/nulos en el conjunto de datos con los que tendremos que trabajar más adelante. Vamos poco a poco creando el proyecto.

Una vez iniciada sesión el sistema, al usuario se le presenta la siguiente vista, donde puede elegir si crear un proyecto nuevo o trabajar sobre uno creado anteriormente.

TimeSeriesWeb Home Dashboard Log Out rafa

### Plataforma para el análisis de series temporales

[Nuevo proyecto](#)

Nombre del proyecto

Descripción del proyecto

Seleccione dataset desde su equipo:

Seleccionar archivo Ningún archivo seleccionado

Enviar

Proyectos creados anteriormente

| Nombre proyecto     | Descripción proyecto                      | Fichero                              | Enlace                   | Descarga archivos         |
|---------------------|-------------------------------------------|--------------------------------------|--------------------------|---------------------------|
| FicheroAerolinea    | Fichero sobre los datos de una aerolínea  | international-airline-passengers.csv | <a href="#">Analizar</a> | <a href="#">Descargar</a> |
| FicheroPrueba       | Descripción Prueba                        | daily-minimum-temperatures.csv       | <a href="#">Analizar</a> | <a href="#">Descargar</a> |
| Fichero Nacimientos | Trabajando con los nacimientos de un país | daily-total-female-births.csv        | <a href="#">Analizar</a> | <a href="#">Descargar</a> |
| Apple               | FicheroApple                              | aapple.csv                           | <a href="#">Analizar</a> | <a href="#">Descargar</a> |
| FicheroPrueba2      | Descripcion                               | serie2.csv                           | <a href="#">Analizar</a> | <a href="#">Descargar</a> |

En esta vista puede cargar un fichero en formato csv, o cargar/descargar un proyecto creado anteriormente

**Ilustración 9.1 Dashboard principal de la plataforma web**

En la imagen superior aparece el dashboard principal, donde el usuario puede elegir crear un proyecto nuevo, o descargar/analizar uno creado anteriormente. En este caso, creamos uno nuevo para esta prueba.

Nada más crear el proyecto, el sistema nos pregunta si queremos eliminar alguna columna de datos innecesaria, por si esos valores no nos interesa estudiarlos



o hay algunos datos alfanuméricos, o por cualquier otra razón, en nuestro caso, no tenemos ningún problema y seguimos con la tarea de limpieza.

En la siguiente vista se nos muestran las características generales de nuestro fichero, como número de muestras, el valor medio, la desviación típica, y los percentiles, así como el valor mínimo y máximo para cada serie de tiempo que hay en el archivo.

TimeSeriesWeb
Home
Dashboard
Log Out rafa

## Plataforma para el análisis de series temporales

Preprocesando el proyecto:

Nombre proyecto: ProyectoPrueba2  
Nombre Archivo : apple.csv

**Mostrando características del fichero:**

Número de valores nulos del fichero: 4  
Estadísticas generales del dataset: **apple.csv**

Mostrando número de valores, media, desviación,máximo,mínimo y percentiles por serie temporal:

|       | Open               | High               | Low               | Close              | Volume             |
|-------|--------------------|--------------------|-------------------|--------------------|--------------------|
| count | 250.0              | 250.0              | 251.0             | 250.0              | 250.0              |
| mean  | 125.27396000000002 | 126.11319999999995 | 124.6028268525897 | 125.40408000000005 | 30487642.796       |
| std   | 17.294411449597003 | 17.27041538441391  | 17.15767157765103 | 17.207486093681933 | 14201408.102509603 |
| min   | 96.75              | 97.65              | 96.42             | 96.67              | 11475922.0         |
| 25%   | 110.465            | 111.5625           | 109.68            | 111.03750000000001 | 22100800.0         |
| 50%   | 117.95             | 118.525            | 117.45            | 118.08             | 26473697.5         |
| 75%   | 142.35750000000002 | 143.24249999999998 | 141.505           | 142.27             | 33978014.5         |
| max   | 156.01             | 156.65             | 155.05            | 156.1              | 112340318.0        |

Pulse para ir al preprocesamiento de datos:

Preprocesar datos

**Ilustración 9.2 Características generales del fichero**

Una vez pulsado en el botón de preprocesar, nos muestra una vista dónde aparecen el número de valores nulos del fichero, así como de distintos métodos de imputación para eliminarlos, como hemos comentado anteriormente en secciones previas de este documento, así como los primeros y últimos diez elementos del archivo, para comprobar que se ha cargado satisfactoriamente en el sistema.

En este caso, voy a optar por sustituir los valores nulos por la media de cada serie temporal.

TimeSeriesWeb

Home

Dashboard

Log Out rafa

# Plataforma para el análisis de series temporales

Imputación del proyecto con nombre:

Nombre proyecto: ProyectoPrueba2

Nombre Archivo : apple.csv

Valores nulos del fichero : 4

Los nombres de las columnas del fichero: **Date - Open - High - Low - Close - Volume -**

Elija el tipo de imputación que quiere realizar, puede elegir entre sustituir los valores nulos por la media, la mediana o la moda, o directamente eliminarlos:

✓ Mean value

Median value

Mode value

Drop Nan

Submit

tipo de imputación, puede comenzar con el análisis:

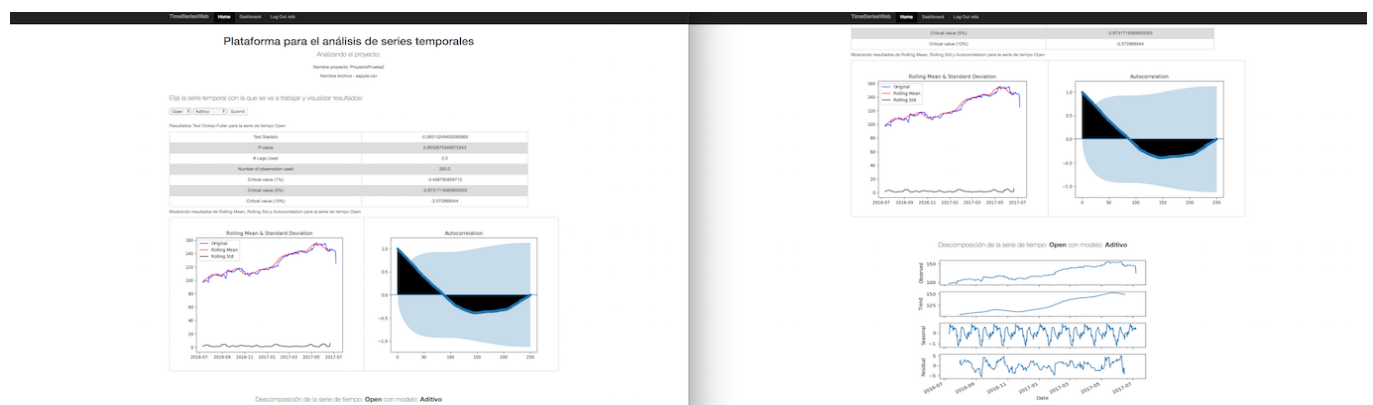
Mostrando primeros 10 elementos del archivo:

| Date                | Open   | High   | Low    | Close  | Volume     |
|---------------------|--------|--------|--------|--------|------------|
| 2017-07-07 00:00:00 | nan    | 144.75 | 142.9  | 144.18 | 19201712.0 |
| 2017-07-06 00:00:00 | 143.02 | 143.5  | 142.41 | 142.73 | 24128782.0 |
| 2017-07-05 00:00:00 | 143.69 | 144.79 | 142.72 | 144.09 | 21569557.0 |
| 2017-07-03 00:00:00 | 144.88 | 145.3  | 143.1  | 143.5  | 14277848.0 |
| 2017-06-30 00:00:00 | 144.45 | 144.96 | 143.78 | 144.02 | 23024107.0 |
| 2017-06-29 00:00:00 | 144.71 | 145.13 | 142.28 | 143.68 | 31499368.0 |
| 2017-06-28 00:00:00 | 144.49 | 146.11 | 143.16 | 145.83 | 22082432.0 |
| 2017-06-27 00:00:00 | 145.01 | 146.16 | 143.62 | 143.73 | 24761891.0 |
| 2017-06-26 00:00:00 | 147.17 | nan    | 145.38 | 145.82 | 25692361.0 |
| 2017-06-23 00:00:00 | 145.12 | 145.16 | 143.00 | 143.00 | 14100000.0 |

## Ilustración 9.2 Limpieza de datos del fichero de prueba

Una vez realizada la limpieza, pulsando sobre el botón de análisis nos lleva a la vista principal de la plataforma, donde aparecen diferentes opciones de análisis, dónde nos permite trabajar con las distintas series temporales de las que dispone el archivo.

Al principio nos muestra el test de Dickey-Fuller para que podamos comprobar la periodicidad de la serie, así como un diagrama de autocorrelación, la representación de la serie junto a su media móvil y desviación estándar y una gráfica donde podemos ver cada uno de los componentes de la misma.



### Ilustración 9.3 Gráficos sobre nuestro fichero de prueba

Un poco más abajo, nos presenta las opciones de análisis de las que disponemos así como herramientas para ajustar la tendencia y la periodicidad, como la diferenciación y por ajuste de modelo, o para ajustar la varianza mediante transformación Box-Cox.

También nos aparecen las opciones comentadas en la etapa de predicción, dónde podemos aplicar ARIMA o LSTM y una breve descripción sobre los parámetros configurables de las redes.

TimeSeriesWeb
Home
Dashboard
Log Out rafa

Proporcionamos tres ajustes de transformadas; Box - Cox, raíz cuadrada y logarítmica:

Box Cox
Submit

Proporcionamos dos maneras de eliminar la tendencia y ajustar la periodicidad, por diferencias o por ajuste de modelo:

Aplicar diferenciación
Submit

No se ha detectado periodicidad, por lo que ARIMA no está disponible

Aplicar Long Short-Term Memory Univariantes:

LSTM Univalente con regresión aplicando diferenciación
Épocas,% entrenamiento
Submit

Aplicar Long Short-Term Memory Univariantes con número de observaciones/salidas personalizadas:

LSTM Univariantes estándar
Épocas,% entrenamiento
N\_Entradas
N\_Salidas
Submit

Aplicar Long Short-Term Memory Multivariantes con número de observaciones/salidas y variables a utilizar personalizadas:

LSTM Multivariantes
Épocas,% entrenamiento
N\_Entradas
N\_Salidas
Variable a predecir
Variables no usar para pre
Submit

Tipos de red y sus características:

| Tipo de red                                                      | Características                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LSTM Univariante con diferenciación                              | Red LSTM a la cual se le aplica diferenciación, aunque no es necesario, ya que las redes neuronales "aprenden" la tendencia.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| LSTM Univariante con regresión aplicando el método de la ventana | Expresa el problema para que se puedan usar múltiples pasos de tiempo recientes para hacer la predicción para el siguiente paso. Esto se llama una ventana y el tamaño de la ventana es un parámetro que se puede ajustar para cada problema. Por ejemplo, dado el tiempo actual (t) que queremos predecir el valor en la próxima vez en la secuencia (t + 1), podemos usar la hora actual (t), así como las dos veces anteriores (t-1) y t-2) como variables de entrada. Cuando se expresa como un problema de regresión, las variables de entrada son t-2, t-1, t y la variable de salida es t + 1. |
| LSTM Univariante con pasos de tiempo                             | Los pasos del tiempo proporcionan otra forma de expresar nuestro problema de series de tiempo. Podemos tomar pasos de tiempo anteriores en nuestra serie temporal como entradas para predecir el resultado en el siguiente paso.                                                                                                                                                                                                                                                                                                                                                                      |
| LSTM Univariante con memoria entre batches                       | Configuración en la que una capa LSTM antes de cada capa LSTM posterior devuelve la secuencia.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| LSTM Multivariante                                               | Las redes neuronales son capaces de modelar modelos con múltiples variables de entradas. Se realiza un pronóstico de series de tiempo multivariante                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

Tipos de red y parámetros de entrada:

| Tipo de red                                                      | Formato valores de entrada                                                                 |
|------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| LSTM Univariante con diferenciación                              | Épocas, LongitudEntrenamientoTest, Look Back, neuronas, tamaño del batch                   |
| LSTM Univariante con regresión aplicando el método de la ventana | Épocas, LongitudEntrenamientoTest, Look Back, neuronas, tamaño del batch                   |
| LSTM Univariante con pasos de tiempo                             | Épocas, LongitudEntrenamientoTest, Look Back, neuronas, tamaño del batch                   |
| LSTM Univariante con memoria entre batches                       | Épocas, LongitudEntrenamientoTest, Look Back, neuronas, tamaño del batch, número de bucles |
| LSTM Univariante estándar/personalizado                          | Épocas, LongitudEntrenamientoTest, neuronas, tamaño batch/número entradas/número salidas   |
| LSTM Multivariante                                               | Épocas, LongitudEntrenamientoTest, neuronas, tamaño batch/número entradas/número salidas   |

Explicación parámetros deep learning:

| Épocas                                                                 | Longitud entrenamiento/test                                                               | Look Back                                                                                                              | Neuronas                                                           | Batch                                                                | Número entradas                              | Número salidas                              |
|------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|----------------------------------------------------------------------|----------------------------------------------|---------------------------------------------|
| Representa las iteraciones sobre el conjunto completo de entrenamiento | Es el porcentaje del fichero que se usa como entrenamiento, el resto será usado como test | Es el número de pasos de tiempo previos para usar como variables de entrada para predecir el próximo periodo de tiempo | Es el número de neuronas que se van a definir para la red neuronal | Define el número de muestras que van a propagarse a través de la red | Define el número de observaciones de entrada | Define el número de observaciones de salida |

Ilustración 9.4 Vista de análisis de la plataforma web

La página nos advierte de que no se puede aplicar ARIMA porque la serie no es periódica, por lo que aplicamos diferenciación para corregirlo y realizamos la predicción, cuyos resultados se pueden ver en la imagen 11.6.

También utilizamos una red LSTM Multivariante utilizando las demás variables del archivo, por lo que, si estamos analizando los valores de apertura de las acciones de APPLE, utilizamos los valores anteriores de dicha serie, junto con las demás variables del conjunto de datos, como son los valores de cierre, el máximo y el mínimo (Close, High, Low). Los resultados aparecen en la siguiente ilustración:

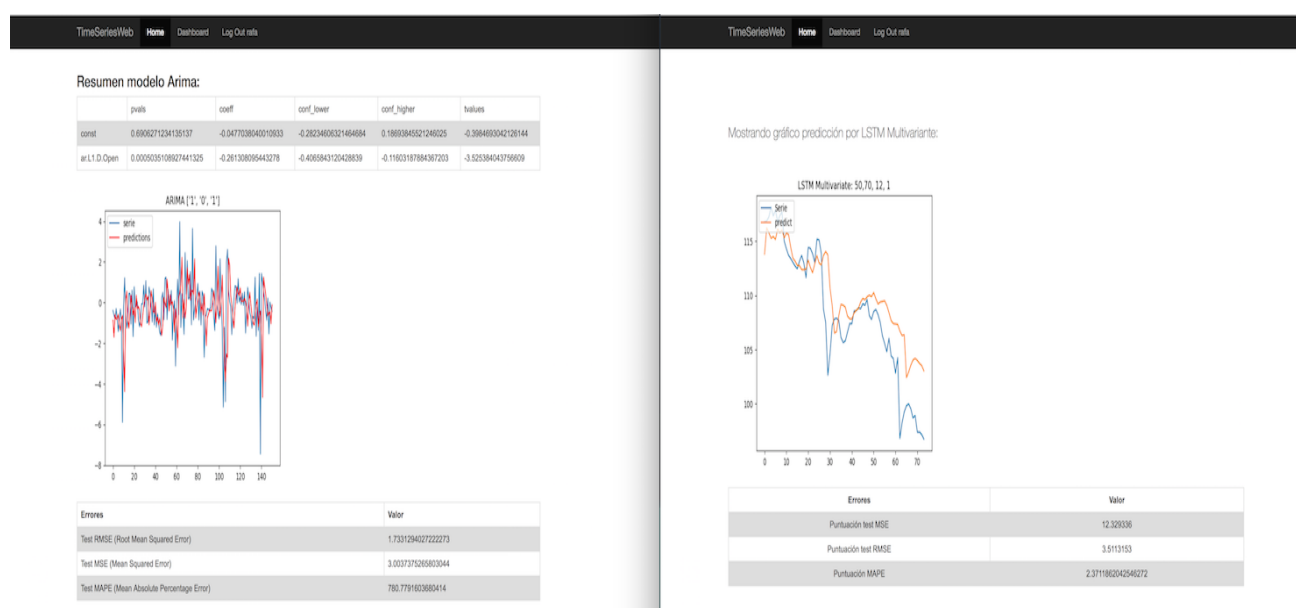


Ilustración 9.5 Vista de predicción de la plataforma web

## 10 Manual de usuario

En este apartado se detalla el manual de referencia para el uso de la plataforma web. A continuación se describen los pasos y las herramientas que proporciona la plataforma para su uso.

### 10.1 Iniciar la plataforma

Para iniciar la plataforma, basta con navegar hasta usando la terminal hasta la carpeta donde se encuentra el proyecto y levantar el servidor ejecutando la siguiente instrucción: `usuarioPrueba:$ python app.py`

Esta acción hace que se levante el servidor e inicie las bibliotecas necesarias para el correcto funcionamiento del mismo. Una vez realizada esta acción, nos dirigimos al navegador e introducimos la dirección: `localhost:5000`

```

/Users/rafaelmegales/anaconda3/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The pandas.core.datetools module is deprecated and will be removed in a future version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
/Users/rafaelmegales/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
/Users/rafaelmegales/anaconda3/lib/python3.6/site-packages/flask_sqlalchemy/__init__.py:794: FSADeprecationWarning: SQLAlchemy_TRACK_MODIFICATIONS adds significant overhead and will be disabled by default in the future. Set it to True or False to suppress this warning.
  'SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and '
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
/Users/rafaelmegales/anaconda3/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The pandas.core.datetools module is deprecated and will be removed in a future version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
/Users/rafaelmegales/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
/Users/rafaelmegales/anaconda3/lib/python3.6/site-packages/flask_sqlalchemy/__init__.py:794: FSADeprecationWarning: SQLAlchemy_TRACK_MODIFICATIONS adds significant overhead and will be disabled by default in the future. Set it to True or False to suppress this warning.
  'SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and '
* Debugger is active!
* Debugger PIN: 237-206-192

```

Ilustración 10.1 Iniciando la plataforma

## 10.2 Registro / Inicio de sesión

Para registrarse, una vez cargada la página principal que se muestra en la figura 10.2, hacemos click en la barra superior, donde pone *Sign Up*. Nos aparece el formulario que se muestra en la figura 10.3, donde introducimos nuestro usuario y contraseña. Una vez registrados, nos dirigimos a la ventana de inicio de sesión haciendo click en *Login*, figura 10.4, e introducimos nuestros datos.



Ilustración 10.2 Vista inicial de la plataforma

The screenshot shows a registration form titled 'Registrarse'. It contains three input fields: 'usuario' with the value 'rafa', 'email' with the value 'rafa@ujaen.es', and 'contraseña' with masked characters. Below the fields is a blue button labeled 'Registrarse'.

Ilustración 10.3 Vista registro en la plataforma

The screenshot shows a login form titled 'Inicie sesión'. It contains two input fields: 'usuario' with the value 'rafa' and 'contraseña' with masked characters. Below the fields is a checkbox labeled 'Recuérdame' and a blue button labeled 'Entrar'.

Ilustración 10.4 Vista inicio de sesión en la plataforma



## 10.3 Dashboard

El dashboard es la página inicial que se carga cuando el usuario inicia sesión. En esta vista se pueden realizar varias operaciones, crear un proyecto nuevo o cargar la ventana de análisis de uno creado anteriormente, así como descargar sus datos.

The screenshot shows the 'Plataforma para el análisis de series temporales' dashboard. At the top is a navigation bar with 'TimeSeriesWeb', 'Home', 'Dashboard', and 'Log Out rafa'. The main content area has a title 'Plataforma para el análisis de series temporales' and a subtitle 'Nuevo proyecto'. Below this are two input fields: 'Nombre del proyecto' and 'Descripción del proyecto'. A message 'Seleccione dataset desde su equipo:' is followed by a file selection button 'Seleccionar archivo' (which shows 'Ningún archivo seleccionado') and an 'Enviar' button. Below this is a table titled 'Proyectos creados anteriormente' with columns: 'Nombre proyecto', 'Descripción proyecto', 'Fichero', 'Enlace', and 'Descarga archivos'. The table lists six projects. At the bottom, a note states: 'En esta vista puede cargar un fichero en formato csv, o cargar/descargar un proyecto creado anteriormente'.

| Nombre proyecto     | Descripción proyecto                      | Fichero                              | Enlace                   | Descarga archivos         |
|---------------------|-------------------------------------------|--------------------------------------|--------------------------|---------------------------|
| FicheroAerolinea    | Fichero sobre los datos de una aerolinea  | international-airline-passengers.csv | <a href="#">Analizar</a> | <a href="#">Descargar</a> |
| FicheroPrueba       | Descripción Prueba                        | daily-minimum-temperatures.csv       | <a href="#">Analizar</a> | <a href="#">Descargar</a> |
| Fichero Nacimientos | Trabajando con los nacimientos de un país | daily-total-female-births.csv        | <a href="#">Analizar</a> | <a href="#">Descargar</a> |
| Apple               | FicheroApple                              | aapple.csv                           | <a href="#">Analizar</a> | <a href="#">Descargar</a> |
| FicheroPrueba2      | Descripcion                               | serie2.csv                           | <a href="#">Analizar</a> | <a href="#">Descargar</a> |
| ProyectoPrueba2     | DescripcionPrueba                         | aapple.csv                           | <a href="#">Analizar</a> | <a href="#">Descargar</a> |

**Ilustración 10.5 Dashboard principal**

### 10.3.1 Creación de un proyecto

Para crear un proyecto, es necesario que el nombre del proyecto a introducir no sea uno que exista anteriormente creado por el mismo usuario, una descripción y un archivo de datos en formato csv, separado por comas.

### 10.3.2 Cargar/Descargar de un proyecto

Para cargar un proyecto, diríjase a la tabla de proyectos creados anteriormente, figura 10.5, y haga click en el botón *Analizar*, le llevará a la ventana de análisis de dicho proyecto.

Para descargar un proyecto, diríjase a la tabla de proyectos creados anteriormente, figura 10.5, que aparece en el la ventana del dashboard y haga click en el botón *Descargar*.

## 10.4 Imputación y limpieza del proyecto

Para realizar la imputación y limpieza de un proyecto, debe crear un proyecto nuevo antes, no es posible limpiar o imputar un proyecto creado anteriormente, debe empezar desde la fase inicial.

Una vez creado el proyecto, la plataforma le preguntará si quiere trabajar con todos los datos del fichero, en caso de que hubiera un subconjunto de datos con los que no quisiera trabajar. Esta vista aparece en la ilustración siguiente:

TimeSeriesWeb
Home
Dashboard
Log Out rafa

Plataforma para el análisis de series temporales

Hola rafa, está creando el proyecto:

**El proyecto con las siguientes características:**

Nombre: ProyectoPrueba3

Nombre Archivo : aapple.csv

Descripción: DescripcionPrueba3

¿Desea trabajar con todos los datos o quiere eliminar alguna columna innecesaria?

Date
Submit
Preprocess

En esta vista puede decidir si quiere eliminar alguna columna innecesaria de su conjunto de datos.

**Ilustración 10.6 Eliminación de columnas innecesarias**

También aparecen las estadísticas generales del fichero:

TimeSeriesWeb
Home
Dashboard
Log Out rafa

Plataforma para el análisis de series temporales

Preprocesando el proyecto:

Nombre proyecto: ProyectoPrueba3

Nombre Archivo : aapple.csv

**Mostrando características del fichero:**

Número de valores nulos del fichero: 4

Estadísticas generales del dataset: **aapple.csv**

Mostrando número de valores, media, desviación, máximo, mínimo y percentiles por serie temporal:

|       | Open               | High               | Low                | Close              | Volume             |
|-------|--------------------|--------------------|--------------------|--------------------|--------------------|
| count | 250.0              | 250.0              | 251.0              | 250.0              | 250.0              |
| mean  | 125.27396000000002 | 126.11319999999999 | 124.60282668525897 | 125.40408000000005 | 30487642.796       |
| std   | 17.294411449597003 | 17.27041538441391  | 17.15767157765103  | 17.207486093681933 | 14201408.102509603 |
| min   | 96.75              | 97.65              | 96.42              | 96.67              | 11475922.0         |
| 25%   | 110.465            | 111.5625           | 109.68             | 111.03750000000001 | 22100800.0         |
| 50%   | 117.95             | 118.525            | 117.45             | 118.08             | 26473697.5         |
| 75%   | 142.35750000000002 | 143.24249999999998 | 141.505            | 142.27             | 33978014.5         |
| max   | 156.01             | 156.65             | 155.05             | 156.1              | 112340318.0        |

Pulse para ir al preprocesamiento de datos:

Preprocesar datos

**Ilustración 10.7 Estadísticas generales del fichero**



Una vez hecho click en *preprocesar datos*, le aparecerá la ventana de imputación y limpieza donde se le informa acerca del número de valores nulos que aparecen en el fichero y de cómo quiere realizar su imputación, eliminándolos completamente, o sustituyéndolos por valores del modelo como la media, moda o mediana. Seleccionando y haciendo click en *Submit* realiza la operación deseada.

The screenshot shows the 'TimeSeriesWeb' application interface. At the top, there is a navigation bar with 'Home', 'Dashboard', and 'Log Out rafa'. The main content area displays the following information:

- Valores nulos del fichero : 4
- Los nombres de las columnas del fichero: **Date - Open - High - Low - Close - Volume**
- Elija el tipo de imputación que quiere realizar, puede elegir entre sustituir los valores nulos por la media, la mediana o la moda, o directamente eliminarlos:
- A dropdown menu is set to 'Mean value' and a 'Submit' button is visible.
- Una vez elegido el tipo de imputación, puede comenzar con el análisis:
- An 'Analysis' button is present.

Below this, it says 'Mostrando primeros 10 elementos del archivo:' followed by a table of the first 10 data points:

| Date                | Open   | High   | Low    | Close  | Volume     |
|---------------------|--------|--------|--------|--------|------------|
| 2017-07-07 00:00:00 | nan    | 144.75 | 142.9  | 144.18 | 19201712.0 |
| 2017-07-06 00:00:00 | 143.02 | 143.5  | 142.41 | 142.73 | 24128782.0 |
| 2017-07-05 00:00:00 | 143.69 | 144.79 | 142.72 | 144.09 | 21569557.0 |
| 2017-07-03 00:00:00 | 144.88 | 145.3  | 143.1  | 143.5  | 14277848.0 |
| 2017-06-30 00:00:00 | 144.45 | 144.96 | 143.78 | 144.02 | 23024107.0 |
| 2017-06-29 00:00:00 | 144.71 | 145.13 | 142.28 | 143.68 | 31499368.0 |
| 2017-06-28 00:00:00 | 144.49 | 146.11 | 143.16 | 145.83 | 22082432.0 |
| 2017-06-27 00:00:00 | 145.01 | 146.16 | 143.62 | 143.73 | 24761891.0 |
| 2017-06-26 00:00:00 | 147.17 | nan    | 145.38 | 145.82 | 25692361.0 |
| 2017-06-23 00:00:00 | 145.13 | 147.16 | 145.11 | 146.28 | 35439389.0 |

Ilustración 10.8 Imputación y limpieza de un fichero

## 10.5 Análisis del proyecto

Para realizar el análisis del proyecto, haga click en el botón de *Análisis* en la imagen 10.8.

Aparecerá un desplegable con la serie temporal a trabajar. Haciendo click en *Submit* cargará todos los datos respectivos a ella: Test de Dickey Fuller, media móvil y desviación estándar, diagrama de autocorrelación, descomposición temporal de la serie y se habilitarán las opciones para realizar transformaciones y predicciones con nuestro conjunto de datos.

En la siguiente imagen se muestra un ejemplo de la vista:

TimeSeriesWeb
Home
Dashboard
Log Out rafa

## Plataforma para el análisis de series temporales

Analizando el proyecto:

Nombre proyecto: ProyectoPrueba3

Nombre Archivo : aapple.csv

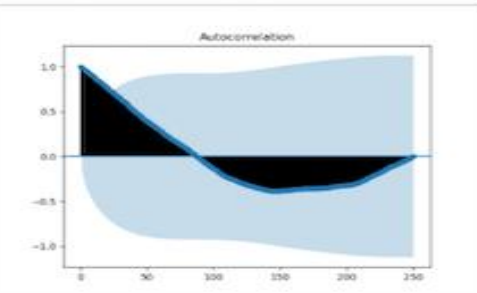
Elija la serie temporal con la que se va a trabajar y visualizar resultados:

Open    Aditivo    Submit

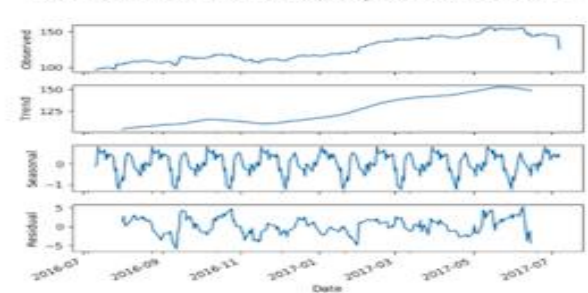
Resultados Test Dickey-Fuller para la serie de tiempo Open

|                            |                      |
|----------------------------|----------------------|
| Test Statistic             | -0.06013249402085865 |
| P-value                    | 0.9532675346975243   |
| # Lags Used                | 0.0                  |
| Number of observation used | 250.0                |
| Critical value (1%)        | -3.456780859712      |
| Critical value (5%)        | -2.8731715065600003  |
| Critical value (10%)       | -2.572968544         |

Mostrando resultados de Rolling Mean, Rolling Std y Autocorrelation para la serie de tiempo Open

Descomposición de la serie de tiempo: **Open** con modelo: **Aditivo**



Proporcionamos tres ajustes de transformadas: Box - Cox, raíz cuadrada y logarítmica:

Box Cox    Submit

Proporcionamos dos maneras de eliminar la tendencia y ajustar la periodicidad, por diferencias o por ajuste de modelo:

Aplicar diferenciación    Submit

No se ha detectado periodicidad, por lo que ARIMA no está disponible

Aplicar Long Short-Term Memory Univariantes:

LSTM Univariante con regresión aplicando diferenciación    Épocas,% entrenamien    Submit

Aplicar Long Short-Term Memory Univariantes con número de observaciones/salidas personalizadas:

LSTM Univariantes estándar    Épocas,% entrenamien    N\_Entradas    N\_Salidas    Submit

Aplicar Long Short-Term Memory Multivariantes con número de observaciones/salidas y variables a utilizar

Ilustración 10.9 Ejemplo vista análisis

## 10.6 Ajustar estacionalidad y tendencia

Para ajustar la estacionalidad y la tendencia, nos dirigimos al apartado dónde nos menciona las herramientas para ajustar la tendencia y estacionalidad que aparece en la figura 10.9.

Podemos elegir entre diferenciación y ajuste de modelo, una vez elegido una opción hacemos click en el botón *Submit* para aplicar los cambios.

## 10.7 Transformaciones

Las transformaciones de datos están destinadas a eliminar el ruido y mejorar la señal en el pronóstico de series de tiempo. En la plataforma se encuentran:

- Homogeneización de varianza mediante una transformación usando el método Box – Cox.
- Transformación cuadrática para una serie de tiempo que tiene tendencia de crecimiento cuadrático.
- Transformación logarítmica para eliminar las tendencias mas extremas.

Para realizar cualquiera de estas transformaciones, nos dirigimos al apartado de transformadas, que se muestra en la figura 10.9, y pulsamos el botón submit después de haber elegido una de las opciones anteriores.

## 10.8 ARIMA

Para realizar una predicción con ARIMA es necesario que exista periodicidad, por lo que debe aplicar una de las transformaciones explicadas en el apartado 10.3.5.

Una vez satisfecho el requerimiento anterior, se nos permite realizar una predicción mediante el modelo econométrico ARIMA. Para ello nos dirigimos al apartado de ARIMA y nos aparece una caja de texto donde introducir datos.

Los datos que se proporcionan son los valores de  $p, d, q$  explicados en el apartado 4.4.

Ejemplo de introducción de datos (separados por comas): 1,0,1.

Posteriormente, se cargará una vista en la que aparece una tabla de resultados de ARIMA, otra tabla de resultados de errores en las predicciones junto con una gráfica.

### 10.9 LSTM Univariante con regresión aplicando diferenciación

Para realizar una predicción con este tipo de LSTM, se debe dirigir al primer desplegable “*Aplicar Long Short-Term Memory Univariantes*” y seleccionar “*LSTM Univariante con regresión aplicando diferenciación*”, el cual se encuentra en la figura 10.9.

En la caja para introducir datos se deben introducir, siguiendo este orden, los siguientes parámetros separados por comas:

- Número de épocas
- Longitud del conjunto de entrenamiento
- Lags
- Número de neuronas
- Tamaño del batch

Posteriormente, se cargará una vista donde se muestra la predicción realizada y una tabla de resultados de errores en las predicciones.

### 10.10 LSTM Univariante con regresión aplicando el método de la ventana

Para realizar una predicción con este tipo de LSTM, se debe dirigir al primer desplegable “*Aplicar Long Short-Term Memory Univariantes*” y seleccionar “*LSTM Univariante con regresión aplicando el método de la ventana*” el cual se encuentra en la figura 10.9.

En la caja para introducir datos se deben introducir, siguiendo este orden, los siguientes parámetros separados por comas:

- Número de épocas
- Longitud del conjunto de entrenamiento
- Lags
- Número de neuronas

- Tamaño del batch

Posteriormente, se cargará una vista donde se muestra la predicción realizada y una tabla de resultados de errores en las predicciones.

### 10.11 LSTM Univariante con regresión aplicando pasos de tiempo

Para realizar una predicción con este tipo de LSTM, se debe dirigir al primer desplegable “*Aplicar Long Short-Term Memory Univariantes*” y seleccionar “*LSTM Univariante con regresión pasos de tiempo*” el cual se encuentra en la figura 10.9.

En la caja para introducir datos se deben introducir, siguiendo este orden, los siguientes parámetros separados por comas:

- Número de épocas
- Longitud del conjunto de entrenamiento
- Lags
- Número de neuronas
- Tamaño del batch

Posteriormente, se cargará una vista donde se muestra la predicción realizada y una tabla de resultados de errores en las predicciones.

### 10.12 LSTM Univariante con memoria entre batches

Para realizar una predicción con este tipo de LSTM, se debe dirigir al primer desplegable “*Aplicar Long Short-Term Memory Univariantes*” y seleccionar “*LSTM Univariante con regresión con memoria entre batches*” el cual se encuentra en la figura 10.9.

En la caja para introducir datos se deben introducir, siguiendo este orden, los siguientes parámetros separados por comas:

- Número de épocas
- Longitud del conjunto de entrenamiento
- Lags
- Número de neuronas

- Tamaño del batch
- Número de bucles

Posteriormente, se cargará una vista donde se muestra la predicción realizada y una tabla de resultados de errores en las predicciones.

### 10.13 LSTM Univariante con número de observaciones estándar

Para realizar una predicción con este tipo de LSTM, se debe dirigir al primer desplegable “*Aplicar Long Short-Term Memory Univariantes con número de observaciones/salidas personalizadas*” y seleccionar “*LSTM Univariantes estándar*” el cual se encuentra en la figura 10.9.

La entrada para la configuración de esta LSTM contiene tres cajas para introducir los parámetros.

En la primera caja, separados por comas y siguiendo este orden, se introducen:

- Número de épocas
- Longitud del conjunto de entrenamiento
- Número de neuronas
- Tamaño del batch

En la segunda caja, un número de instantes de tiempo anterior para tomar como entrada, por ejemplo: 1, 2, 3,4, etc.

En la tercera caja, el instante de tiempo a predecir: 1 (el cual sería  $t+1$ ), 2( el cual sería  $t+2$ ), etc.

Posteriormente, se cargará una vista donde se muestra la predicción realizada y una tabla de resultados de errores en las predicciones.

### 10.14 LSTM Univariante con número de observaciones personalizado

Para realizar una predicción con este tipo de LSTM, se debe dirigir al primer desplegable “*Aplicar Long Short-Term Memory Univariantes con número de observaciones/salidas personalizadas*” y seleccionar “*LSTM Univariantes personalizado*” el cual se encuentra en la siguiente figura 10.9:

La entrada para la configuración de esta LSTM contiene cinco cajas para introducir los parámetros.

En la primera caja, separados por comas y siguiendo este orden, se introducen:

- Número de épocas
- Longitud del conjunto de entrenamiento
- Número de neuronas
- Tamaño del batch

En la segunda caja, un número de instantes de tiempo anterior para tomar como entrada, por ejemplo:

- -1, -3, -5 ( lo que representaría que se usan los instantes de tiempo  $(t-1)$ ,  $(t-3)$  y  $(t-5)$ ).
- -1 ( lo que representaría que se usan los instantes de tiempo  $t-1$ ).

En la tercera caja, el instante de tiempo a predecir: 1 (el cual sería  $t+1$ ), 2 ( el cual sería  $t+2$ ), etc.

## 10.15 LSTM multivariantes

Para realizar una predicción con este tipo de LSTM, se debe dirigir al primer desplegable “Aplicar Long Short-Term Memory Multivariantes con número de observaciones/salidas y variables a utilizar personalizadas” y seleccionar “LSTM Multivariantes” el cual se encuentra en la siguiente figura:

Aplicar Long Short-Term Memory Multivariantes con número de observaciones/salidas y variables a utilizar personalizadas:

LSTM Multivariantes    Épocas,% entrenamiento,    N\_Entradas    N\_Salidas    Variable a predecir    Variables no usar para pre    Submit

Tipos de red y sus características:

| Tipo de red                                                      | Características                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LSTM Univariante con diferenciación                              | Red LSTM a la cual se le aplica diferenciación, aunque no es necesario, ya que las redes neuronales “aprenden” la tendencia.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| LSTM Univariante con regresión aplicando el método de la ventana | Expresa el problema para que se puedan usar múltiples pasos de tiempo recientes para hacer la predicción para el siguiente paso. Esto se llama una ventana y el tamaño de la ventana es un parámetro que se puede ajustar para cada problema. Por ejemplo, dado el tiempo actual ( $t$ ) que queremos predecir el valor en la próxima vez en la secuencia ( $t + 1$ ), podemos usar la hora actual ( $t$ ), así como las dos veces anteriores ( $t-1$ ) y $t-2$ ) como variables de entrada. Cuando se expresa como un problema de regresión, las variables de entrada son $t-2$ , $t-1$ , $t$ y la variable de salida es $t + 1$ . |
| LSTM Univariante con pasos de tiempo                             | Los pasos del tiempo proporcionan otra forma de expresar nuestro problema de series de tiempo. Podemos tomar pasos de tiempo anteriores en nuestra serie temporal como entradas para predecir el resultado en el siguiente paso.                                                                                                                                                                                                                                                                                                                                                                                                    |
| LSTM Univariante con memoria entre batches                       | Configuración en la que una capa LSTM antes de cada capa LSTM posterior devuelve la secuencia.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| LSTM Multivariante                                               | Las redes neuronales son capaces de modelar modelos con múltiples variables de entradas. Se realiza un pronóstico de series de tiempo multivariante                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

Tipos de red y parámetros de entrada:

| Tipo de red                                                      | Formato valores de entrada                                                                     |
|------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| LSTM Univariante con diferenciación                              | Épocas, Longitud, Entrenamiento, Test, Look Back, neuronas, tamaño del batch                   |
| LSTM Univariante con regresión aplicando el método de la ventana | Épocas, Longitud, Entrenamiento, Test, Look Back, neuronas, tamaño del batch                   |
| LSTM Univariante con pasos de tiempo                             | Épocas, Longitud, Entrenamiento, Test, Look Back, neuronas, tamaño del batch                   |
| LSTM Univariante con memoria entre batches                       | Épocas, Longitud, Entrenamiento, Test, Look Back, neuronas, tamaño del batch, número de bucles |
| LSTM Univariante estándar/personalizado                          | Épocas, Longitud, Entrenamiento, Test, neuronas, tamaño batch/número entradas/número salidas   |
| LSTM Multivariante                                               | Épocas, Longitud, Entrenamiento, Test, neuronas, tamaño batch/número entradas/número salidas   |

Explicación parámetros deep learning:

| Épocas                                                                 | Longitud entrenamiento/test                                                               | Look Back                                                                                                              | Neuronas                                                           | Batch                                                                | Número entradas                              | Número salidas                              |
|------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|----------------------------------------------------------------------|----------------------------------------------|---------------------------------------------|
| Representa las iteraciones sobre el conjunto completo de entrenamiento | Es el porcentaje del fichero que se usa como entrenamiento, el resto será usado como test | Es el número de pasos de tiempo previos para usar como variables de entrada para predecir el próximo periodo de tiempo | Es el número de neuronas que se van a definir para la red neuronal | Define el número de muestras que van a propagarse a través de la red | Define el número de observaciones de entrada | Define el número de observaciones de salida |

Hola rafa ! Descarga todos los ficheros [clickando aquí](#)

Ilustración 10.10 Ejemplo vista análisis 2



La entrada para la configuración de esta LSTM contiene tres cajas para introducir los parámetros.

En la primera caja, separados por comas y siguiendo este orden, se introducen:

- Número de épocas
- Longitud del conjunto de entrenamiento
- Número de neuronas
- Tamaño del batch

En la segunda caja, un número de instantes de tiempo anterior para tomar como entrada, por ejemplo: 1, 2, 3,4, etc.

En la tercera caja, el instante de tiempo a predecir: 1 (el cual sería  $t+1$ ), 2( el cual sería  $t+2$ ), etc.

En la cuarta caja, el nombre exacto de la variable que queremos predecir.

En la quinta caja, el nombre exacto de las variables que no queremos usar para predecir (vacío si queremos usar todas).

Posteriormente, se cargará una vista donde se muestra la predicción realizada y una tabla de resultados de errores en las predicciones.

## 11 Conclusiones y trabajos futuros

En este capítulo se describen las conclusiones extraídas una vez terminado el proyecto y las futuras mejoras que se podrían incluir.

### 11.1 Conclusiones

La elección de este trabajo fin de grado fue llevado a cabo al haber tenido una breve introducción a la ciencia de datos en las asignatura de *Tecnologías de la Información* y en *Sistemas de Información para las decisiones estratégicas* que se cursan en el 4º curso de esta titulación.

Al cursarlas me di cuenta de cómo la informática, en especial el campo de la inteligencia artificial, está tan relacionado con el mundo del business intelligence, y de cómo es de importante el manejo de la información para tener una ventaja competitiva sobre los demás, gracias a la transformación de la información en conocimiento.

Este proyecto me ha permitido entender qué es el Deep Learning, y cómo funcionan las redes neuronales, ya que antes de realizar este T.F.G no tenía ningún conocimiento tanto práctico como teórico de estos conceptos.

Esto me ha hecho progresar en diferentes campos, como lo son:

- El aprendizaje de un nuevo lenguaje de programación, *Python*.
- Mejorar sobre desarrollo web, en especial *Flask* y *Bootstrap*.
- Conocer numerosos recursos para la predicción de series temporales.
- Profundizar en el campo de la *Inteligencia Artificial*.

## 11.2 Trabajos futuros

El siguiente paso es subir la plataforma web al servidor SIMIDAT (grupo de investigación de *Sistemas Inteligentes y Minería de Datos de la Universidad de Jaén*).

Para ello se realizará una instalación de la plataforma en un contenedor Docker, y posteriormente se subirá al servidor SIMIDAT, para que el proyecto sea totalmente accesible.

## Bibliografía

**Brownlee, Jason.** 2018. *Introduction to Time Series Forecasting with Python, How to Prepare Data and Develop Models To Predict the Future.*

**Brownlee, Jason.** 2018. *Machine Learning Mastery with Python, Understand Your Data, Create Accurate Models and Work Projects End-To-End.*

**Brownlee, Jason.** 2018. *Long Short-Term Memory Networks With Python, Develop Sequence Prediction Models with Deep Learning.*

**Brownlee, Jason.** 2016. *Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras.* <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>

**Brownlee, Jason.** 2016. *Hot to Scale data for LSTM networks in Python.* <https://machinelearningmastery.com/how-to-scale-data-for-long-short-term-memory-networks-in-python/>

**Brownlee, Jason.** 2017. *Time series prediction with Deep learning in Python with Keras.* <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>

**Brownlee, Jason.** 2017. *How to Grid Search ARIMA Model Hyperparameters with Python.* <https://machinelearningmastery.com/grid-search-arima-hyperparameters-with-python/>

**Jain, Aarshay.** 2016. *A comprehensive beginner's guide to create Time Series Forecast (with Codes in Python).* <https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/>

**Lopez Briega, Raul E.** 2016. *Series de tiempo con Python.* <https://relopezbriega.github.io/blog/2016/09/26/series-de-tiempo-con-python/>

**Toledo, Juan C.** 2016. *Análisis de Series de Tiempo, Minicurso práctico en Python.* [http://www.escuela-verano.otrasenda.org/wp-content/uploads/2015/06/curso\\_series.pdf](http://www.escuela-verano.otrasenda.org/wp-content/uploads/2015/06/curso_series.pdf)

**Kadamati, Srini.** 2015. *Cleaning CSV Data Using the CommandLine and csvkit.* <https://www.dataquest.io/blog/data-cleaning-command-line/>

**Johansen, Anders E.** 2015. *Flask4scientists, Uploading of files.* [http://hplgit.github.io/web4sciapps/doc/pub/.\\_web4sa\\_flask016.html](http://hplgit.github.io/web4sciapps/doc/pub/._web4sa_flask016.html)

**Johansen, Anders E.** 2015. *Flask4scientists, Using Flask for Scientific Web Applications.* [http://hplgit.github.io/web4sciapps/doc/pub/.\\_web4sa\\_flask000.html#\\_\\_\\_sec0](http://hplgit.github.io/web4sciapps/doc/pub/._web4sa_flask000.html#___sec0)

**Kerim, Daniel.** 2015. *Comprehensive Guide to Time Series Analytics, Visualization and Prediction with Python.* <http://dacadatay.com/data-science/comprehensive-guide-time-series-analytics-visualization-prediction-python/>

**González-Avella, J.C.** 2017. *Análisis de series temporales usando redes neuronales recurrentes*. <https://www.apsl.net/blog/2017/06/14/analisis-de-series-temporales-usando-redes-neuronales-recurrentes/>

**Torres, Jordi.** 2017. *Primeros pasos en Keras*. <http://jorditorres.org/primeros-pasos-en-keras/>