

Grado en Ingeniería Informática  
2021 - 2022

*Trabajo Fin de Grado*

# “Análisis comparativo de algoritmos de aprendizaje automático para la predicción bursátil”

---

Alvaro Andrés Henríquez Pérez

Tutor

David Quintana Montero

Madrid, 2022



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

## **AGRADECIMIENTOS**

Deseo aprovechar esta oportunidad para agradecer a todas las personas que me han apoyado en llevar a cabo este proyecto, y más aún, en los que me han apoyado en mi transcurso en mi carrera.

Quiero agradecer a cada uno de los miembros de mi familia, ya que desde el primer momento que comencé la carrera he tenido su apoyo incondicional, en especial a mis padres por el sacrificio puesto en venirse a España a acompañarme en mi camino de formación, y tenerlos a la mano en todo momento. A mis hermanos, quienes han estado pendiente de mí, y han sido muy importantes para mí para llegar a donde estoy. A mi abuelo, que desde el primer día me apoyó en mis decisiones, el cual siempre tuvo fe en mí, y quien me dejó muchas lecciones y aprendizajes.

A mis amigos que me dejó la universidad, Gerardo, Eduardo, Andrés y Luis, con los cuales he pasado momentos únicos, y en todo momento he recibido su apoyo ante cualquier problema. A Gabriela, quien ha sido durante muchos años mi amiga, a quien agradezco su apoyo y motivación, y quien me ha ayudado a alcanzar todas mis metas. A Juan y Antonio, que, a pesar de la distancia, han estado en todo momento y han sido de gran ayuda.

A todos los profesores que he tenido a lo largo de la universidad, y de los cuales he aprendido durante mi periodo de formación. En especial a mi tutor David Quintana, por dedicar todo su tiempo en mí, y permitirme llevar a cabo este proyecto, su dedicación a sus alumnos es de reconocer, ya que en todo momento a pesar de las circunstancias he recibido su apoyo.

Alvaro Andrés Henríquez Pérez

## **RESUMEN**

El aprendizaje automático es una rama de la inteligencia artificial que se encarga de crear modelos automáticos a partir del uso de datos. Por otro lado, el mercado bursátil es el lugar dónde empresas, personas o instituciones compran y venden activos financieros que se cotizan en la bolsa. Esta compra y venta puede producir una rentabilidad y de esta manera generar beneficio.

Teniendo en cuenta esto, se pretende aplicar algoritmos de aprendizaje automático para la predicción de valores en la bolsa de mercados, y de esta manera, generar modelos que sean capaces de modelar el comportamiento de los activos.

Realizar estas predicciones pueden llegar a ser complicadas ya que no existe un comportamiento fijo de la bolsa, y se puede llegar a decir que su comportamiento es aleatorio. Es por esto, que para abordar el problema se pretende realizar un estudio previo exhaustivo para poder realizar una aproximación que sea lo más correcta posible.

Por otro lado, se pretende realizar una comparativa de distintos algoritmos, de manera que se consiga la mejor configuración para nuestras predicciones. A partir de la mejor configuración, se pretende realizar un estudio estadístico, de manera que se sepa lo bueno o malo del resultado.

Por último, se pretende crear una interfaz que permita a cualquier usuario sin experiencia en el campo de la inteligencia artificial realizar predicciones en el mercado de valores

## **ABSTRACT**

Machine learning is a branch of artificial intelligence that is responsible for creating automatic models from the use of data. On the other hand, the stock market is the place where companies, individuals or institutions buy and sell financial assets that are listed on the stock exchange. This buying and selling can produce a return and thus generate profit.

Taking this into account, we intend to apply machine learning algorithms for the prediction of securities in the stock market, and thus, generate models that are capable of modeling the behavior of assets.

Making these predictions can be complicated because there is no fixed behavior of the stock market, and it can be said that its behavior is random. That is why, to address the problem is intended to conduct a comprehensive preliminary study to make an approximation that is as correct as possible.

On the other hand, it is intended to make a comparison of different algorithms, in order to get the best configuration for our predictions. From the best configuration, it is intended to perform a statistical study, so that we know how good or bad the result is.

Finally, it is intended to create an interface that allows any user without experience in the field of artificial intelligence to make predictions in the stock market.

# ÍNDICE GENERAL

<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1. Motivación.....	1
1.2. Descripción del problema .....	1
1.3. Objetivos .....	2
1.4. Estructura del documento.....	2
<b>2. ESTADO DEL ARTE .....</b>	<b>5</b>
<b>3. MARCO TEÓRICO .....</b>	<b>8</b>
3.1. Inteligencia artificial .....	8
3.2. Algoritmos de Aprendizaje automático .....	9
3.2.1. Redes Neuronales .....	10
3.2.2. Árboles de decisión .....	11
3.2.3. Máquina de Vectores de Soporte .....	13
3.3. Entorno del mercado de valores.....	14
<b>4. DESARROLLO .....</b>	<b>20</b>
4.1 Entorno operacional .....	20
4.2 Extracción de información .....	21
4.2.1 Fuente de datos .....	21
4.2.2 Transformación de los datos.....	23
4.2.2.1 Media móvil .....	23
4.2.2.2 Media móvil exponencial.....	24
4.2.2.3 RSI .....	24
4.2.2.4 Bandas de Bollinger .....	25
4.2.3 Series temporales.....	26
4.3. Metodología .....	27
4.4. Diseño de GUI .....	27
4.4.1. Especificación de casos de uso.....	27
4.4.1.1. Descripción de actores .....	27
4.4.1.2. Descripción de tablas y plantillas de casos de uso .....	27
4.4.1.3. Descripción textual de casos de usos .....	28
4.4.2. Especificación de requisitos .....	30
4.4.2.1. Descripción de tablas y plantillas de requisitos.....	30
4.4.2.2. Descripción textual de requisitos .....	31
4.4.3. Diseño de la aplicación.....	33
4.4.3.1. Arquitectura del sistema .....	34
<b>5. RESULTADOS EXPERIMENTALES .....</b>	<b>36</b>
5.1 Perceptrón multicapa .....	36
5.2 Decission Tree Regressor .....	40

5.3 Random Forest.....	43
5.4 SVR .....	46
5.5 Comparativa de modelos .....	49
<b>6. ANÁLISIS DE RESULTADOS.....</b>	<b>51</b>
<b>7. GESTIÓN DEL PROYECTO .....</b>	<b>52</b>
7.1. Fases del proyecto.....	52
7.2. Planificación.....	52
7.3. Presupuesto .....	53
7.4. Marco regulador.....	55
<b>8. CONCLUSIONES.....</b>	<b>57</b>
<b>BIBLIOGRAFÍA.....</b>	<b>59</b>
<b>ANEXO A. MANUAL DE USUARIO .....</b>	<b>61</b>
<b>ANEXO B. SUMMARY.....</b>	<b>64</b>
1. Introduction .....	64
1.1 Motivation .....	64
1.2 Description of the problem .....	64
1.3 Objectives.....	64
2. State of the art .....	65
3. Theoretical Framework .....	66
3.1 Artificial Intelligence .....	66
3.2 Stock Market Enviroment .....	67
4. Development .....	68
4.1 Data extraction.....	69
4.2 Experimentation.....	71
5. GUI.....	71
6. Result.....	71
7. Conclusion.....	72



## ÍNDICE DE FIGURAS

Figura 1. Programación tradicional contra el aprendizaje automático. Fuente: Elaboración propia .....	8
Figura 2. Proceso de generalización de un modelo de aprendizaje supervisado. Fuente: Elaboración propia .....	9
Figura 3. Estructura de una red neuronal. Fuente: Elaboración propia .....	10
Figura 4. Estructura de un árbol de decisión [11] .....	12
Figura 5. Ejemplo de una tendencia alcista [12] .....	16
Figura 6. Indicadores de tendencia en Tesla. Fuente: Elaboración propia .....	16
Figura 7. Indicador de impulso en la gráfica de Tesla. Fuente: Elaboración propia .....	17
Figura 8. Indicador de volatilidad en la gráfica de Tesla. Fuente: Elaboración propia .....	18
Figura 9. Indicador de volumen en Tesla. Fuente: Elaboración propia .....	18
Figura 10. Método de gráfica de la estructura de datos. Fuente: Elaboración propia .....	22
Figura 11. Representación gráfica de los datos obtenidos. Fuente: Elaboración propia .....	23
Figura 12. Representación gráfica de los datos con indicadores utilizados. Fuente: Elaboración propia .....	26
Figura 13. Plantilla de diagrama de casos de usos .....	28
Figura 14. Diagrama de casos de uso .....	29
Figura 15. Representación del modelo-vista-controlador. Fuente: Elaboración propia .....	34
Figura 16. Error cometido por el perceptrón multicapa en todo el conjunto de test para el mejor modelo .....	37
Figura 17. Error cometido por el árbol de regresión en todo el conjunto de test para el mejor modelo .....	40
Figura 18. Error cometido por el Random Forest en todo el conjunto de test para el mejor modelo .....	43
Figura 19. Error cometido por el SVR en todo el conjunto de test para el mejor modelo .....	46
Figura 20. Diagrama de Gantt. Fuente: Elaboración propia .....	53
Figura 21. Parámetros de predicción para el algoritmo Random Forest Regressor .....	61
Figura 22. Parámetros de predicción para el algoritmo Support Vector Regressor .....	62
Figura 23. Parámetros de predicción para el algoritmo Decision Tree Regressor .....	62
Figura 24. Ejemplo de predicción realizada por la aplicación .....	63





## ÍNDICE DE TABLAS

Tabla 1. Estructura de los datos recogidos por la librería yfinance .....	21
Tabla 2. Plantilla de casos de uso.....	28
Tabla 3. Caso de uso número 1 .....	29
Tabla 4. Caso de uso número 2 .....	30
Tabla 5. Caso de uso número 3 .....	30
Tabla 6. Caso de uso número 4.....	30
Tabla 7. Plantilla de requisitos .....	31
Tabla 8. Requisito funcional número 1 .....	32
Tabla 9. Requisito funcional número 2 .....	32
Tabla 10. Requisito funcional número 3 .....	32
Tabla 11. Requisito funcional número 4 .....	32
Tabla 12. Requisito funcional número 5 .....	33
Tabla 13. Requisito no funcional número 6 .....	33
Tabla 14. Requisito no funcional número 2 .....	33
Tabla 15. Requisito no funcional número 3 .....	33
Tabla 16. Estadístico descriptivo de los datos utilizados .....	36
Tabla 17. Configuración de parámetros del Perceptrón Multicapa .....	37
Tabla 18. Experimentación del Perceptrón Multicapa .....	38
Tabla 19. Configuración de parámetros para Decission Tree Regressor .....	40
Tabla 20. Experimentación de Decission Tree Regressor .....	41
Tabla 21. Configuraciones de Random Forest .....	43
Tabla 22. Experimentación de Random Forest .....	44
Tabla 23. Configuración de SVR .....	46
Tabla 24. Experimentación de SVR .....	47
Tabla 25. Estadístico descriptivo del RMSE obtenido en el conjunto de validación .....	49
Tabla 26. Significación estadística de la diferencia de errores medio cometido en el conjunto de validación .....	49
Tabla 27. Tiempos llevados a cabo para la realización del proyecto.....	53
Tabla 28. Coste total del personal para el desarrollo del proyecto.....	54
Tabla 29. Coste del hardware amortizado .....	54
Tabla 30. Coste total para el desarrollo del proyecto.....	55



# 1. INTRODUCCIÓN

## 1.1. Motivación

La principal motivación para llevar a cabo este proyecto nace de los conocimientos financieros adquiridos en los últimos años, y sumado a lo aprendido en la informática, en específico en el área de inteligencia artificial, surge la idea de realizar una mezcla. Además de esto, han surgido oportunidades de desarrollar modelos de predicción, que a simple vista parecían comportarse de manera correcta, lo que despierta más la atención en profundizar esta área.

Por otro lado, el principal problema que se puede plantear para desarrollar este trabajo es la dificultad de predecir un comportamiento con una componente aleatoria, sin embargo, la búsqueda de ideas nuevas y realizar pruebas de distintos algoritmos puede convertirse en un progreso para la predicción del mercado.

Además de esto, el crear una herramienta de predicción puede llegar a ser de gran utilidad para contrastar resultados de análisis financieros personales. De manera que se convierta en un instrumento de ayuda para cualquier persona que desee operar en el mercado.

Por último, el analizar distintos algoritmos, puede llegar a ser útil para entender en este entorno cuales se adaptan mejor.

## 1.2. Descripción del problema

El mercado financiero es el lugar donde se negocian valores y derivados bajo un coste. Los que podemos encontrar aquí pueden ser acciones, bonos, materias primas, entre otros. A partir de esta compra y venta es posible obtener alguna rentabilidad, de manera, que resulta interesante buscar la manera de predecir su comportamiento para obtener el mayor beneficio.

El comportamiento de este activo suele tener un comportamiento con una componente estocástica, y resulta complicado realizar predicciones, sin embargo, con el paso del tiempo han surgido métodos como el análisis técnico, que busca identificar tendencias estadísticas recogidas de la actividad comercial, como pueden ser el precio y el volumen.

A pesar de que a simple vista el obtener un beneficio realizando operaciones de compra y venta parece sencillo, la realidad es que no, ya que el valor de un activo va definido por muchos factores difíciles de controlar y lleva a cambios constantes en sus precios.

A partir de esto, se han realizado numerosos estudios que intenten modelar el comportamiento de los activos, abordándolos en su mayoría, con modelos de aprendizaje automático, sin embargo, los resultados obtenidos tienden a perder eficiencia con el paso del tiempo.

### 1.3. Objetivos

El objetivo de este trabajo tiene como principal fin realizar un estudio de técnicas de aprendizaje automático enfocado en el ámbito del mercado de valores, en específico, para la predicción del precio de cierre.

Para realizar esto, se pretende en primer lugar realizar los estudios del ETF<sup>1</sup> SPY, de esta manera, se deben analizar los datos, y tomar decisiones de diseño para transformar los datos, de manera que quede una estructura que permita hacer uso de modelos de aprendizaje automático. La decisión de realizar los estudios en base al SPY, es debido a que es un activo el cual tiene como fin agrupar las 500 compañías más grande de Estados Unidos, de esta manera, se realiza un estudio general del comportamiento medio que pueda estar ocurriendo en la bolsa.

Una vez, se han realizado los ajustes en los datos, se pretenden realizar pruebas de distintas técnicas de aprendizaje automático, esto tiene como fin, la obtención de aquellos algoritmos que mejor resultado dan a la hora de predecir el precio de cierre.

Por otro lado, se pretende realizar un contraste de hipótesis, para observar que tan útiles son las predicciones que realiza el algoritmo.

Por último, se pretende crear una interfaz web, que permita a cualquier usuario sin experiencia en la inteligencia artificial, realizar predicciones del precio de una lista de activos, para hacer esto, se presentarán distintos algoritmos con diversos parámetros, de manera que el usuario, coloque aquellos que quiera utilizar, y el programa, realiza la predicción de la acción escogida.

A continuación, se enumeran los objetivos del trabajo de investigación

1. Estudio de los datos, y transformación de estos, para poder hacer uso de modelos de aprendizaje automático para la predicción del cierre diario del ETF SPY
2. Diseñar un sistema para poder realizar pruebas, con distintos algoritmos de aprendizaje automático
3. Realizar un contraste de hipótesis, para saber si los resultados obtenidos, son buenas predicciones o no
4. Diseño de interfaz web, que permite a un usuario realizar predicciones en la bolsa de valores

### 1.4. Estructura del documento

El siguiente documento pretende detallar todas las fases llevadas a cabo para realizar el proyecto. Desde el estudio previo, hasta las implementaciones y desarrollo con su respectivo análisis y conclusiones. El desarrollo de este seguirá la siguiente estructura:

- En primera parte se tiene esta introducción, que pretende dar idea de lo que va a ser el trabajo.

---

<sup>1</sup> exchange-traded fund, es un valor de inversión agrupado que funciona de forma muy parecida a un fondo de inversión

- A continuación, se realiza un análisis del estado de la cuestión, donde se da contexto al área donde se experimenta este problema, sumado a términos del contexto del desarrollo. Por otro lado, se tendrá estudios de distintos trabajos previos como criterio de partida.
- Posteriormente se pretenden declarar los objetivos del trabajo a realizar
- Siguiendo este orden, se pretende describir todo el proceso de desarrollo del trabajo, desde la obtención de los datos, hasta el desarrollo del programa.
- Tras esto, se pretende realizar un apartado donde se expliquen todos los resultados obtenidos, así como un estudio estadístico de ellos.
- Por último, se pretende desarrollar las conclusiones obtenidas durante el desarrollo del proyecto, así como futuras mejoras, y contrastar si se han cumplido los objetivos propuestos en el trabajo.



## 2. ESTADO DEL ARTE

En la predicción del precio de un activo en la bolsa se pueden aplicar numerosas técnicas y diversos métodos para alcanzar este objetivo, ya que se pueden realizar desde predicciones de compra y venta, hasta del precio exacto.

Las predicciones de precio suelen ser difícil ya que se trata de conseguir el precio exacto de un activo sumado a que existen incontables factores que alteran el mercado, sin embargo, existen trabajos que intentan realizar esto, como el artículo publicado en 2012 de Adebisi Ayodele el cuál intenta combinar el análisis técnico y fundamental para realizar predicciones utilizando redes neuronales, en específico el perceptrón multicapa [1]. Los principales indicadores técnicos fueron la apertura, máximo, mínimo, volumen, cierre. En este trabajo se consiguen algunas configuraciones que resultan bastante buenas, como el dataframe de sólo indicadores técnicos, que arrojan un modelo con una raíz del error cuadrático medio (RMSE<sup>2</sup>) de 0,0729, la parte negativa de este trabajo es la mezcla del análisis técnico y fundamental, ya que una medida es en años y la otra en días, lo que resulta confuso a la hora de modelar. Por otro lado, es importante destacar la modelización de los datos la cual se realiza como una serie temporal<sup>3</sup> y esto tiene sentido ya que los datos de un día dependen en cierta medida sobre los datos de los días anteriores.

Por otro lado, los investigadores Rasheed y Ganesh de la universidad de Georgia publican en el mismo año una comparativa de distintos atributos y algoritmos de aprendizaje automático para la predicción de distintas acciones del mercado americano [2]. En este trabajo resulta que el mejor algoritmo es el SVR, se dice que las redes neuronales se comportan bien, sin embargo, los resultados obtenidos no eran satisfactorios, por otro lado, los atributos que mejor funcionaron fueron aquellos que tenían el volumen, otro conjunto de datos que se comportó bien fue una combinación de índices e información de la empresa a predecir.

Otro trabajo que se puede encontrar es el publicado por Subba, Kudipudi y Krishna en el 2019 [3], donde se busca comprar una regresión línea con árboles de regresión para la predicción del precio en la bolsa, resulta que los árboles tienen una gran ventaja, y que los resultados obtenidos tienden a ser buenos.

En el 2020, los investigadores Suryani y Buani hacen uso de redes neuronales para la predicción de la acción ANTM [4], como datos de entrada utilizan únicamente el precio de cierre de la acción y realizan transformaciones en los datos haciendo uso de la media móvil<sup>4</sup>. En este trabajo se obtuvo para el mejor modelo un RMSE de 0,004 y una media de error de 0,0121, estos resultados resultan bastante buenos. Por otro lado, se estudian los resultados de un modelo sin aplicar la media móvil, y resulta estadísticamente significativo mejor el modelo utilizando la media móvil.

Para abordar el problema desde el punto de vista de los datos podemos ver la estructura que se muestran en [5], dónde utilizan indicadores técnicos para modelizar el problema, por una

---

<sup>2</sup> Función que mide la cantidad de error que hay entre dos conjuntos de datos

<sup>3</sup> Consta de unos datos los cuales son medidos en un instante de tiempo determinado, y además tienen un orden cronológico

<sup>4</sup> Es una sencilla herramienta de análisis técnico que suaviza los datos de los precios creando una media de precios constantemente actualizada



parte se tiene la media móvil, el RSI, la media móvil exponencial, todos estos dependiendo del precio de cierre. Por otro lado, existen otros indicadores dependientes de máximo, mínimos y volumen, los utilizados en este ámbito fueron la media del rango verdadero, el indicador Williams %R, o el Oscilador Estocástico %K. También podemos encontrar otros trabajos.



### 3. MARCO TEÓRICO

#### 3.1. Inteligencia artificial

A lo largo de la historia, el ser humano ha intentado replicar su inteligencia, esto se puede constatar por hechos históricos como los silogismos creados por Aristóteles, que buscan describir una parte del funcionamiento de la mente humana, o la primera máquina autocontrolada creada por Ctesibio en el 250 a.C [6]. Sin embargo, todas estas creaciones estaban bastante alejadas al comportamiento de un humano, y es por esto, que en el año 1950 se publica el artículo “Computing Machinery and Intelligence” [7], escrito por Alan Turing, el cual tiene como objetivo evaluar el comportamiento de una máquina y su semejanza con el de un humano, esto recibe el nombre de “Test de Turing”.

La Inteligencia artificial, es por ende un área de la informática, la cual busca desarrollar y replicar la inteligencia del ser humano a través de un computador, este comportamiento intenta imitar principalmente el razonamiento humano.

A pesar de que emular el comportamiento humano puede llegar a ser difícil, se han tenido en los últimos años una cantidad de avances, debido a nuevos conocimientos, y mejoras en la velocidad de cómputo de los ordenadores. Actualmente se tiene como objetivo desarrollar sistemas de IA de amplia aplicabilidad que interactúen de forma segura con los seres humanos y el mundo físico. Para ello, cada vez se reúnen más conceptos y enfoques diferentes: aprendizaje automático, razonamiento simbólico, ciencia cognitiva, psicología del desarrollo, ingeniería de control de robots e interacciones hombre-máquina, entre otros.

En el área de la inteligencia artificial existen muchas técnicas para resolver problemas, sin embargo, una de las más utilizadas es la de aprendizaje automático.

El aprendizaje automático es la ciencia que tiene como objetivo el desarrollo de diversas técnicas que consigan que los ordenadores aprendan. Para realizar esto, la solución al problema no es programada implícitamente, sino que se aplican diversas técnicas para que las máquinas aprendan

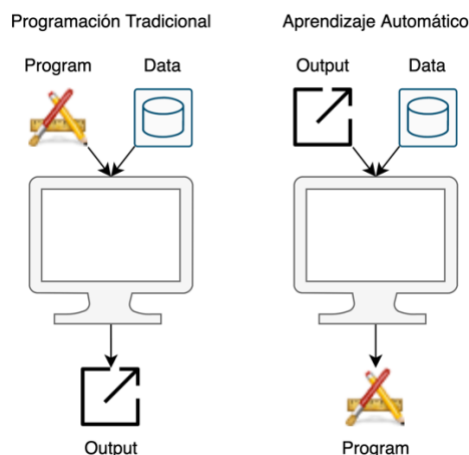


Figura 1. Programación tradicional contra el aprendizaje automático. Fuente: Elaboración propia

Como se muestra en la Figura 1. Se puede observar una comparativa de la programación tradicional y el aprendizaje automático. Donde la primera busca la solución a partir de una instancia<sup>5</sup>. Y, por otro lado, se encuentra el aprendizaje automático, que, a partir de datos de ejemplo, se aplican diversos algoritmos que generan un modelo capaz de dar soluciones para diversas instancias.

En cuanto a los algoritmos, hay que tener en cuenta que su funcionamiento se basa principalmente en modelos matemáticos y estadísticos, es, por esto, que esta área de la informática es multidisciplinar, y cada vez son más los conocimientos que se aplican, como lo pueden ser la neurociencia, la lógica, entre otros.

En el aprendizaje automático existen tres ramas de aprendizaje, una de ellas es el aprendizaje no supervisado, esta técnica utiliza una serie de datos para entrenar el algoritmo, sin tener idea de la relación o comportamiento de estos, el fin de esto es dar explicación a los datos, o buscar una manera de entender su comportamiento

Por otro lado, se tiene el aprendizaje semisupervisado, el cual combina una cantidad de datos etiquetados con una cantidad de datos no etiquetados, por ende, es una mezcla entre el aprendizaje supervisado y no supervisado. Esta técnica ayuda en los entrenamientos en los cuales se tiene una cantidad de datos sin etiquetar y otra parte etiquetada, ya sea por su coste de adquisición o falta de información.

Por último, se tiene el aprendizaje supervisado, el cual es el foco de la investigación, el cuál es una técnica que busca aprender, y generalizar a partir de una serie de datos de entrenamiento que tienen la característica que son pares, con una parte correspondiente a la información de la instancia, y otra para el resultado esperado de esa entrada. Esta salida esperada dependiendo del problema puede ser un valor numérico (problema de regresión) o una etiqueta (problema de clasificación).

El principal objetivo del aprendizaje supervisado es la generalización de una función capaz de generar una salida válida a partir de una entrada, esto debe ser hecho una vez ha visto una serie de ejemplos.

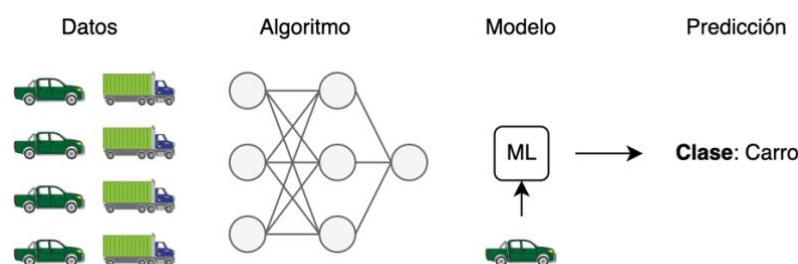


Figura 2. Proceso de generalización de un modelo de aprendizaje supervisado. Fuente: Elaboración propia

### 3.2. Algoritmos de Aprendizaje automático

<sup>5</sup> Especificación exacta de los datos de un problema para un caso particular

### 3.2.1. Redes Neuronales

Entre los algoritmos para el aprendizaje automático se encuentran las redes neuronales, estas, buscan imitar el funcionamiento del cerebro humano, y de esta manera, generar un modelo capaz de solventar el problema.

Una red neuronal se encuentra conformada por neuronas individuales interconectadas entre sí, entre estas conexiones existen pesos que empiezan siendo valores aleatorios, y en el proceso de entrenamiento se van ajustando los valores, de manera que el resultado final tenga una solución con el menor error.

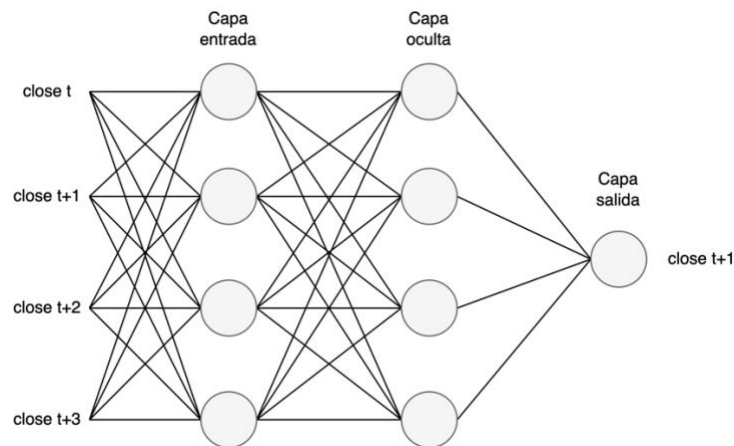


Figura 3. Estructura de una red neuronal. Fuente: Elaboración propia

La principal estructura de las redes neuronales se conforma de tres capas:

- Capa de entrada: La capa de entrada es la primera, y se conforma por los datos de entrada, su número puede ser determinado por el valor de datos de la instancia.
- Capa oculta: La capa oculta viene a continuación de la de entrada, en cuanto al número de neuronas en esta capa puede ser diverso, y no existe ninguna regla para determinar su valor. Esta capa frecuentemente es una, sin embargo, pueden existir varias capas intermedias
- Capa de salida: Para el caso de la capa de salida, esta será la última, y dependerá del problema, para uno de regresión, se acostumbra a tener una neurona como capa de salida, sin embargo, en problemas de clasificación, dependerá del problema

A fin de cuentas, el funcionamiento de las redes de neuronas artificiales busca solventar el problema de la misma manera que lo resolvería el ser humano. La estructura de la red consta de tres capas, y según variaciones, la capa intermedia puede ser una, o varias capas de neuronas.

En el ámbito de las redes neuronales han existido muchos avances en los últimos años, pero remontando a los inicios de esta rama, se tiene que, en 1940 Warren McCulloch, junto a Walter Pitts crean un modelo lineal, capaz de producir salidas positivas o negativas [8]. A este modelo se le llamo neurona, ya que intentaba replicar el funcionamiento en solitario de una neurona. A pesar de que era un gran avance, este modelo no permitía resolver muchas cosas

Durante el paso del tiempo, en el año 1958 se publica por parte de Rosenblatt un artículo, donde da a conocer el perceptrón multicapa [9]. Este modelo, consiste igual que en el

anterior, como unidad básica la neurona, agregado a esto, cada neurona recibe una serie de entradas y pesos, y su funcionamiento consiste en una combinación de una suma ponderada, y si esta suma, llega a cierto valor, dispara la neurona y genera una salida. De esta manera, el funcionamiento se acerca más al comportamiento de una neurona real, y este avance logró un cambio en el mundo de las redes neuronales.

En la librería de scikit-learn [10] se pueden encontrar los siguientes parámetros del modelo perceptrón multicapa:

- **Hidden\_layer\_size:** representa el número de capas ocultas que debe tener la red neuronal y a su vez, el número de neuronas de cada capa
- **Activation:** Función de activación de la neurona, como se mencionó anteriormente, para que la neurona se dispare deberá llegar a cierto valor, por ende, se debe definir que función se encarga de esto
- **Solver:** Método para actualizar los pesos de las neuronas
- **Batch\_size:** Número de iteraciones que se utilizan para actualizar el modelo
- **Learning\_rate\_init:** Valor del learning rate. Este valor, hace referencia a cuanto es el cambio que debe de hacerse en los pesos de las neuronas, una vez se ha entrenado
- **Learnin\_rate:** Como se modifica el learning rate, existen métodos que dejan el valor constante, o, por otro lado, algunos que disminuyen su valor con el pasar de las iteraciones
- **Max\_iter:** Número de iteraciones para entrenar el modelo

### **3.2.2. Árboles de decisión**

Un árbol de decisión es un modelo predictivo cuyo objetivo principal es el aprendizaje inductivo a partir de observaciones y construcciones lógicas [11]. El comportamiento de estos árboles es similar a los sistemas de predicción basados en reglas, que buscan dar una solución a un problema basado en el cumplimiento o no de ciertas condiciones.

La representación física del modelo es mediante un árbol, el cual se compone de la siguiente estructura:

- **Nodo:** El nodo se representa como un atributo de entrada, se puede tener el nodo principal o raíz, del cual salen todos los descendientes, y desde el cual se inicia el proceso de clasificación. A partir de cada nodo, se generan preguntas a estos atributos que resultan en respuestas representadas como nodos hijos
- **Rama:** La rama se constituye como los posibles valores del atributo
- **Hoja:** Las hojas, o también conocidas como nodos finales, son la respuesta final, y se entiende como la solución al problema.

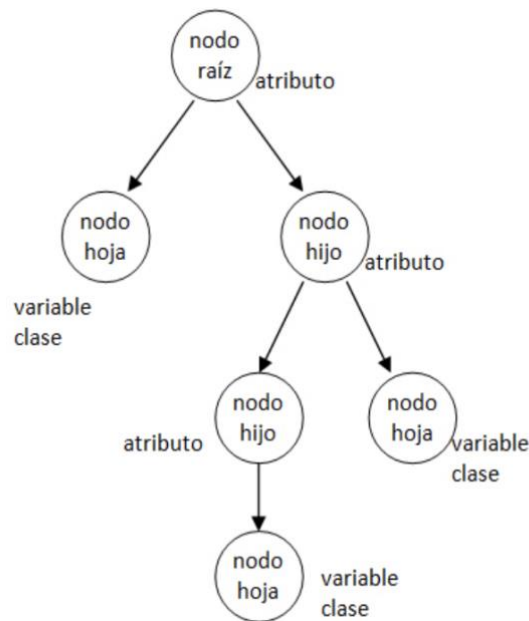


Figura 4. Estructura de un árbol de decisión [11]

Para la construcción del árbol de decisión se plantea una idea muy sencilla, la cual es empezar por un atributo raíz, el cual es seleccionado mediante cálculos matemáticos, pero que siguen la lógica de buscar aquel que divida de la mejor manera las clases<sup>6</sup> o valores<sup>7</sup> que se quieren predecir, a partir de este atributo se generan los nodos hijos o hojas dependiendo del caso. Si se logra un nodo en el cual ya se diferencia del resto, y es un resultado final, se queda como hoja, pero en el caso de que se necesite seguir dividiendo el conjunto, se irán generando los nodos hijos de manera sucesiva.

Uno de los primeros árboles a mencionar es el Decision Tree Regressor, este algoritmo es de los más básicos dentro de la rama de los árboles de decisión, su objetivo es crear un modelo, capaz de predecir un resultado numérico a partir de unos datos de entrada, las decisiones para poder realizar la predicción, es a partir de reglas de decisión inferidas a partir de los datos.

Algunos de los parámetros que se pueden utilizar para el desarrollo del algoritmo son los proporcionados por la librería de scikit-learn [10]:

- **Criterion:** Es la función que se encarga de medir la calidad de la división. Existen algunas funciones que permiten medir esto, como lo pueden ser el error cuadrático medio, el error medio absoluto, o Poisson, que busca reducir la desviación de Poisson para encontrar divisiones
- **Splitter:** Hace referencia a la estrategia utilizada para dividir los nodos, se tienen la opción de mejor, es decir aquel atributo de mayor importancia, y por otro lado random, el cual escoge aleatoriamente la mejor división.
- **Max\_depth:** Este parámetro numérico, permite definir, que profundidad máxima va a tener el árbol, de esta manera, se defina cuantos niveles de profundidad tendrá el árbol

<sup>6</sup> Problema de clasificación

<sup>7</sup> Problema de regresión

- `Min_samples_split`: Se define numéricamente, cuantas muestras se debe tener para dividir un nodo.
- `Min_samples_leaf`: Parámetro que permite establecer, cuantas muestras debe tener una hoja.

Por otro lado, se tiene el Random Forest Regressor, que resulta un poco más complejo, y que es un algoritmo que genera modelos más complejos.

Este algoritmo lo que hace es generar una cantidad variada de árboles de decisión, sin embargo, no es sólo esto, si no que cada árbol, puede contener una cantidad de atributos variada, y, además, una cantidad aleatoria de datos. De esta manera, cada árbol es completamente diferente, y entrenado diferente. El resultado final, constará de una media de todas las salidas de los árboles.

A continuación, se describen algunos de los parámetros que recibe el modelo de scikit-learn [10]

- `N_estimators`: Hace referencia al número de árboles que debe de generar el modelo.
- `Criterion`
- `Max_depth`
- `Min_samples_leaf`
- `Min_samples_split`

Las últimas 4, tienen el mismo significado que la de un árbol de decisión, y fueron enunciadas anteriormente

### **3.2.3. Máquina de Vectores de Soporte**

Este algoritmo es mejor conocido por su nombre en inglés Support Vector Machine (SVM), para el caso de un problema de regresión, este se conoce como Support Vector Regressor (SVR), este algoritmo tiene el objetivo de encontrar un hiperplano en un espacio n-dimensional que clasifique los puntos de datos. Para el caso de un problema de regresión, se busca la mejor línea que ajuste a los valores, y este será el hiperplano que tiene el máximo número de puntos.

La estructura de este algoritmo se basa en las siguientes claves:

- **Hiperplano**: Son aquellos límites que permiten predecir la salida continua
- **Núcleo**: Conjunto de funciones matemáticas que toman datos de entrada y los transforma, esto es utilizado para encontrar un hiperplano en dimensiones superiores
- **Líneas de límites**: Líneas que se encuentran alrededor del hiperplano

En el caso del problema de regresión, el plano que se busca es aquel que tenga el menor error entre el dato a predecir y el plano. La dificultad de este algoritmo es conseguir el plano, y existe un kernel, el cual es la función que se encarga de dividir los datos, sin embargo, esta función, puede ser lineal, cuadrática y hasta un árbol de decisión.



A continuación, se enumeran algunos de los parámetros que ofrece la librería scikit-learn [10].

- Kernel: Es la función que se encarga de dividir los datos, es decir, de pasar los datos de una dimensión a otra.
- Degree: El grado de la función a utilizar cuando el kernel es de tipo poly
- Gamma: Este parámetro hace referencia a la curvatura del plano, es decir, que tanto debería del plano resultante.
- Epsilon: Define un margen de tolerancia cuando no hay penalización de error
- C: Parámetro de regularización

### 3.3. Entorno del mercado de valores

El mercado de valores se refiere en general a una serie de bolsas y otros lugares en los que se compran y venden acciones de empresas públicas. Estas actividades financieras se llevan a cabo a través de bolsas formales institucionalizadas (físicas o electrónicas) y a través de mercados extrabursátiles que operan bajo un conjunto definido de regulaciones.

Aunque los términos "mercado de valores" y "bolsa" se utilizan a menudo indistintamente, el segundo término es realmente un subconjunto del primero. Los operadores del mercado de valores compran o venden acciones en una o más de las bolsas que forman parte del mercado de valores global.

Al momento de operar en la bolsa, existen diversas técnicas, que permiten a los usuarios tener una intuición de la proyección que tendrá el mercado, estas se dividen en análisis fundamental y análisis técnicos.

En cuanto al análisis fundamental, se tiene que es una técnica que tiene como objetivo determinar el valor intrínseco, para realizar esto, se hace un estudio combinado de los estados financieros de la empresa, las influencias externas, los acontecimientos, y las tendencias del sector.

A partir del análisis fundamental se pueden tomar acciones de compra y venta dependiendo del valor obtenido al analizar, en caso de un valor bajo en comparación a la cotización real, es recomendable vender, y en caso contrario comprar.

Es posible realizar dos tipos de análisis:

- Top-down: Se centra en el panorama general, o en cómo la economía global y los factores macroeconómicos impulsan los mercados y, en última instancia, los precios de las acciones. También se fijan en el rendimiento de los sectores o industrias. Estos inversores creen que, si el sector va bien, lo más probable es que las acciones de esas industrias también lo hagan.
- Bottom-up: Enfoque de inversión que se centra en el análisis de valores individuales y resta importancia a los ciclos macroeconómicos y de mercado. En otras palabras, la inversión ascendente suele centrarse en los fundamentos de una empresa concreta,

como los ingresos o los beneficios, frente al sector o la economía en general. El enfoque de inversión ascendente parte de la base de que las empresas individuales pueden obtener buenos resultados incluso en un sector con un rendimiento inferior, al menos en términos relativos.

En cuanto al análisis técnico se tiene que es un método que intenta predecir la tendencia, dirección y precio del mercado. Para realizar esto se hace uso de modelos estadísticos y matemáticos, que son capaces de generar señales de compra y venta.

Este enfoque se basa en tres premisas [12]:

- “Los movimientos del mercado lo descuentan todo”. Esta es la principal base del análisis técnico, y hace referencia a que el mercado se ve afectado por factores como pueden ser, fundamentales, políticos, psicológicos, entre otros, y esto, se ve reflejado en el precio del activo. Es por esto, que el análisis técnico, solo se enfoca en las gráficas, para predecir las tendencias del mercado, ya que, por factores externos, se verá reflejado en la gráfica, y de esta manera el analista sabrá como actuar.
- “Los precios se mueven por tendencias”. Esta premisa hace referencia a que los precios del mercado de valores actúan por tendencias, como se puede observar en la Figura 6, y es una de las principales razones por las cuales se debe de observar el gráfico, ya que permite observar tendencias y de esta manera, saber el rumbo del mercado. Es importante destacar que la tendencia de un mercado no siempre es la misma, y que existen cambios en esta, que el análisis técnico debería ser capaz de resolver.
- “La historia se repite”. En cuanto a la historia, se sabe que el análisis técnico se apoya mucho en la psicología humana, ya que, analizando distintas gráficas del pasado, se observa como tendencias alcista o bajistas han funcionado en el pasado, y es por esto por lo que se asume que también van a funcionar en el futuro. Sin ir muy lejos, se cree que para comprender el futuro hay que estudiar el pasado, o simplemente que el futuro es una repetición del pasado.

Al momento de realizar un análisis técnico no basta con mirar el gráfico y decir que se está formando una tendencia alcista o bajista, si no que el analista se apoya en numerosos indicadores que le permiten saber la tendencia que seguirá el mercado.

En cuanto a estos indicadores existen dos aproximaciones de analistas, ya que en primer lugar podemos encontrar a aquellas personas que usan los indicadores más comunes para estudiar el mercado, y así tomar sus decisiones, pero, por otro lado, existen personas que diseñan sus propios indicadores, a través de análisis matemáticos y estadísticos, los cuales se ajustan a las metas personales, y de esta manera realizar sus predicciones.

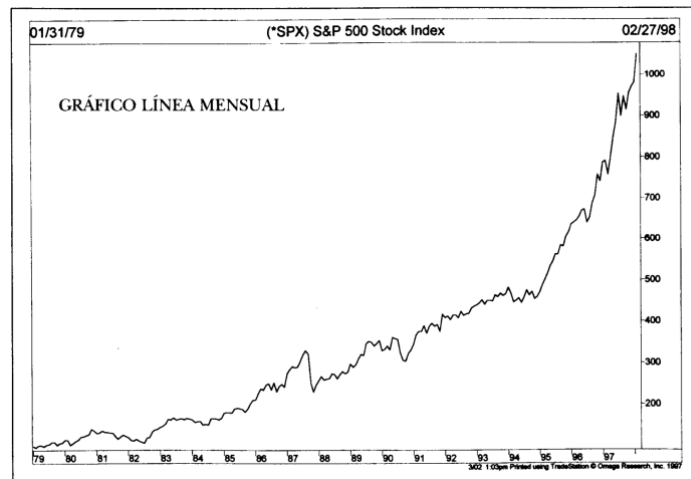


Figura 5. Ejemplo de una tendencia alcista [12]

En cuanto a los indicadores, existen una multitud de estos, sin embargo, no hay manera de decir lo bueno o malo de uno, ya que depende mucho del uso que se le quiere dar, y además de esto, las predicciones que se pueden realizar con estos indicadores dependen mucho de la persona que hace uso de ellos, ya que un indicador como tal, no te dice nada, sin embargo, la interpretación que se le pueda dar depende de cada persona, y a partir de esto, es que se hacen las decisiones en el mercado

A continuación, se pretenden describir los tres tipos de indicadores que existen:

- **Indicadores de tendencia:** Estos indicadores, son utilizados y calculados con información del pasado, por ende, son indicadores retrasados, y el objetivo de estos, es identificar las tendencias del mercado, por otro lado, son indicadores que suavizan mucho el comportamiento del activo, y esto es normal, ya que se tratan de medias



Figura 6. Indicadores de tendencia en Tesla. Fuente: Elaboración propia

En cuanto a los indicadores de tendencias, existen variedad de ellos, y con diversas utilidades, se pueden nombrar algunos como media móvil (MA – Moving average), media móvil exponencial (EMA – Exponential moving average), o media móvil exponencial doble (DEMA – Double exponential moving average). En la figura 6, se puede observar un ejemplo, para la acción Tesla, dónde se muestran la media móvil de 50 en color azul, y, por otro lado, la media móvil exponencial de 200 en color morado. Al número que se refiere, es al número de días previos que representan la media.

- Indicadores de impulso: Estos indicadores tienen como función hacer una medida de que tanto cambia el precio de un activo, de esta manera, se pueden detectar si el mercado está en un estado débil o fuerte, y a partir de esto tomar decisiones de operativa.



Figura 7. Indicador de impulso en la gráfica de Tesla. Fuente: Elaboración propia

En cuanto a estos indicadores, se pueden nombrar el Índice de fuerza relativa (RSI – Relative Strength Index), ratio de cambio (ROC – Rate of change), o convergencia de la media móvil (MACD – Moving average Convergence). En este caso, en la figura 7, se ilustra en la parte inferior de la imagen el RSI.

- Indicadores de volatilidad: La volatilidad es un aspecto muy importante al operar en el mercado, ya que esto hace referencia a que tan brusco son los cambios en los precios de una acción. Estos indicadores son muy importantes en ciertas operativas, y se pueden encontrar algunos como las bandas de Bollinger, la media de rango verdadero (ATR – Average true range), o el Donchian Channel. En la figura 8 se puede observar una representación de las bandas de Bollinger, estas son representadas por las líneas superior, inferior y del medio, en la gráfica de la acción tesla



Figura 8. Indicador de volatilidad en la gráfica de Tesla. Fuente: Elaboración propia

- Indicadores de volumen: Por último, se tienen los indicadores de volumen, estos hacen referencia a como su nombre lo dice, el volumen que entra y sale del mercado, este indicador es importante cuando se quieren observar si existen tendencias de venta o compra en las operaciones. Los indicadores que podemos mencionar en este apartado son la distribución acumulada (A/D – Accumulation-Distribution) o el de oscilador de volumen (VO – Volume Oscillator). En la figura 9, se puede ver en la partide inferior, el indicador VO.



Figura 9. Indicador de volumen en Tesla. Fuente: Elaboración propia



## 4. DESARROLLO

Como se explicó en el apartado anterior del trabajo, se procede en primer lugar al desarrollo de los métodos de estudio de los algoritmos de aprendizaje automático para la predicción del precio del activo SPY. En primer lugar, se procede a la extracción de los datos, así como su preparación, posteriormente, se plantean los algoritmos que se pretenden utilizar, y, por último, el desarrollo de la ingeniería del software, para la aplicación web a desarrollar.

Es importante destacar que la primera parte del trabajo pretende hacer el estudio de una acción en concreto, y su comportamiento con los algoritmos de aprendizaje automático, y en la segunda parte, donde se desarrolla la web, se pretende crear una herramienta, que permita a un usuario, crear modelos, que predigan el precio de una acción en concreto.

Por último, el estudio del activo SPY se hará en la moneda estadounidense, el dólar (\$)

### 4.1 Entorno operacional

Para el desarrollo de este trabajo, existen numerosas herramientas que posibilitan su desarrollo, y con el paso del tiempo, han salido tecnologías que facilitan los procesos de aprendizaje automático.

Una herramienta bastante conocida en el ámbito de algoritmos de aprendizaje automático es Weka [13], esta permite crear numerosos modelos de aprendizaje automático, en esta misma herramienta, se pueden gestionar los datos, probar distintas configuraciones y hacer descargar modelos. Todo esto funciona en el lenguaje de programación Java. El principal problema que se tiene para hacer uso de esta herramienta es la imposibilidad de realizar una comparativa entre todos los modelos, ya que para hacer esto, se debe de realizar un proceso manual de ir probando todos y cada uno, por esta razón, esta herramienta queda descartada para su utilización.

A partir de la idea de querer probar numerosos modelos, de la mejor manera, resulta la idea de realizar el proceso con un lenguaje de programación, entre los más famosos están Python, Java, C++ y Javascript. Sin embargo, en los últimos años, el principal lenguaje líder en proceso de Inteligencia artificial es Python, esto se debe a que este lenguaje tiene numerosas librerías incorporadas que facilitan este proceso. Por esta razón, se pretende realizar las implementaciones en este lenguaje

En cuanto a las librerías que se tienen para realizar modelos predictivos se tienen scikit-learn[10], la cual tiene numerosos algoritmos de regresión y clasificación, además de herramientas que permiten el análisis de datos. Por otro lado, existe TensorFlow [14], librería utilizada exclusivamente para desarrollar redes neuronales. Algunas otras que se pueden mencionar son keras, pyTorch, entre otras. Debido a que en el desarrollo de este trabajo se pretenden probar algunos de los algoritmos más conocidos, se ha decidido hacer uso de scikit-learn, debido a su versatilidad y amplitud de algoritmos

Por último, hay que hablar del centro de ejecuciones, al tratarse de ejecuciones largas, y necesidad de correr a la mayor velocidad posible, queda descartada la opción de usar un

ordenador propio, por ende, se pretende hacer uso de Google Colab, herramienta de Google que permite escribir cuadernos de Jupyter, y ejecutarlos en una máquina en la nube.

## 4.2 Extracción de información

Tras los estudios previos realizados, debemos en primer lugar, obtener los datos con los que se quieren trabajar, y además de esto, hay que realizar una transformación de estos. En el mundo financiero, los datos de un activo se pueden trabajar en series temporales de distintas magnitudes, desde segundos, semanas, o incluso meses. Por ende, en el estudio a realizar se pretende hacer uso de series diarias, de manera que cada fila de datos represente un día en concreto, por ende, el objetivo de la predicción, es decir el precio de cierre que el activo va a tener. Con precio de cierre, se refiere al valor monetario, que el activo tendrá al momento del cierre del mercado.

### 4.2.1 Fuente de datos

Como se ha mencionado anteriormente, se pretende realizar los estudios en base al ETF SPY, para la extracción de los datos se hará uso de la librería en Python `yfinance` [15][16]. Esta es una librería que utiliza la `api`<sup>8</sup> de Yahoo, de manera que permite acceder a distintas series temporales, en el caso de este trabajo, se pretende recolectar una serie temporal diaria, desde que se comenzó a cotizar el ETF, hasta el día 15 de julio del 2022.

La fuente de los datos viene de la siguiente manera:

TABLA 1. ESTRUCTURA DE LOS DATOS RECOGIDOS POR LA LIBRERÍA YFINANCE

Date	Apertura	Máximo	Mínimo	Cierre	Cierre Adj	Volume
1993-01-29	43.97	43.97	43.75	43.94	25.72	1003200
1993-02-01	43.97	44.25	43.96	44.25	25.90	480500
.....	.....	.....	.....	.....	.....	.....
2021-07-14	373.61	379.05	371.04	377.91	377.91	89704800
2021-07-15	382.55	382.25	380.54	385.13	385.13	79016800

El total de instancias recogidas es de 7419, que corresponden al total de días que se ha operado el ETF en el período de estudio. Es importante recordar que la bolsa no abre los fines de semanas, ni días feriados, y, por ende, existen saltos de fecha en los datos, sin embargo, para el caso de estudio, esto no es un problema, ya que, aunque exista un salto, para el modelo significará como el próximo día.

En cuanto a los datos que nos devuelve la librería, se describen su significado a continuación:

---

<sup>8</sup> Application Programming Interface



- Apertura: Precio al que abre el activo, una vez se entra en el horario de apertura del mercado. Se podría entender que este valor es el cierre del día pasado, sin embargo, antes de la apertura existen unas horas de comercio que no están disponibles a todo el público, y, por ende, no se considera como el precio de apertura
- Máximo: Precio máximo que alcanza el activo durante las horas de comercio
- Mínimo: Precio mínimo al que llega el activo durante las horas de comercio
- Cierre: Precio al que cierra el mercado
- Adj Cierre: En inglés conocido como adjusted Cierre, o en español, cierre ajustado. El cierre ajustado es el precio de cierre después de los ajustes por todas las divisiones y distribuciones de dividendos aplicables.
- Volume: Hace referencia al volumen operacional. Es decir, la cantidad de compras y ventas que ocurren durante el período de tiempo.

Estos datos pueden llegar a ser muy complejos, sin embargo, pueden ser graficados de una manera muy sencilla, para esto se muestra en la Figura 10 como se maneja la estructura de estos datos, este método se llama gráfica de velas. En el caso de una vela de color verde, significa que el valor de apertura se encuentra en la parte inferior, y en el caso de una vela roja, en la parte superior. Su representación de colores significa de un aumento del valor para el caso del color verde, y un descenso del precio para el caso del color rojo.

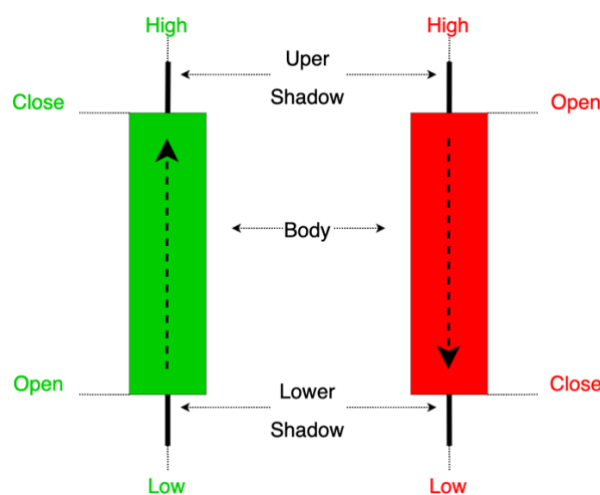


Figura 10. Método de gráfica de la estructura de datos. Fuente: Elaboración propia

Tomando en cuenta esto, se puede observar en la Figura 11 un gráfico de velas de la información obtenida para el desarrollo de este trabajo, donde en la parte inferior a la gráfica también se puede observar el volumen por día.

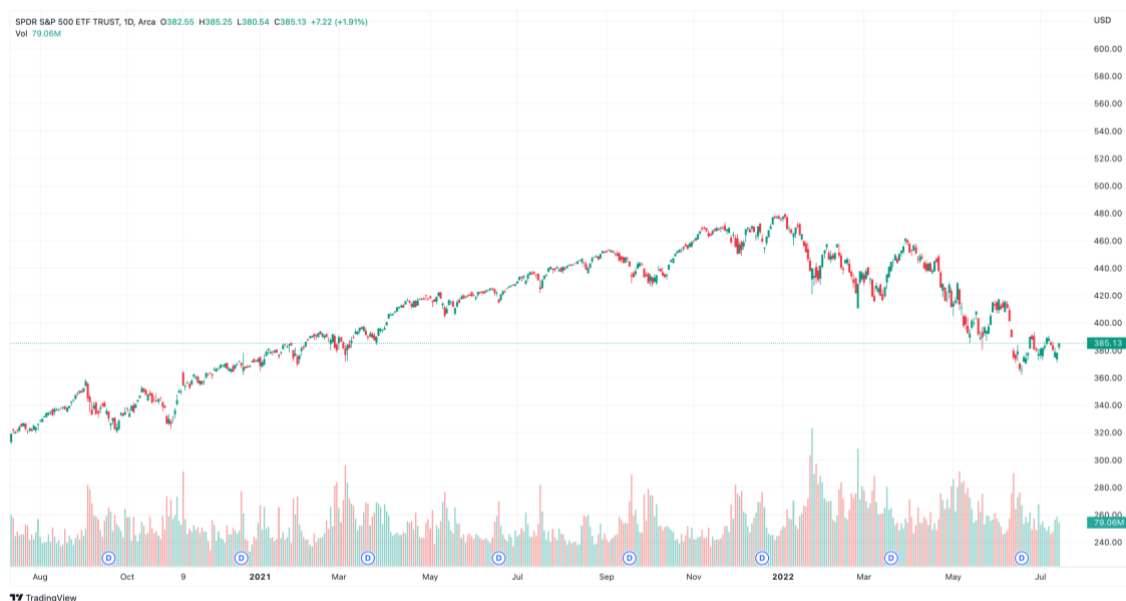


Figura 11. Representación gráfica de los datos obtenidos. Fuente: Elaboración propia

## 4.2.2 Transformación de los datos

Una vez se han obtenido los datos, hay que realizar algunas transformaciones para poder hacer uso de estos. Ya que no basta con tener los datos para poder realizar las predicciones.

Cómo se han visto en trabajos previos, y habiendo definido los tipos de análisis que existen para predecir la bolsa, se tienen el análisis técnico y fundamental. Para el desarrollo de este trabajo se pretende hacer uso único del análisis técnico, y en específico, se pretenden aplicar diversos indicadores técnicos, que permitan dar más información de los datos. Es importante destacar que se utilizará únicamente los datos de Cierre, y los indicadores serán calculados con esta variable. De manera que los datos sobre el Máximo, Mínimo, volumen y Apertura, no serán utilizados

A continuación, se describen los indicadores a utilizar. Es importante destacar que todos los indicadores son calculados con la librería en Python llamada Ta-Lib [17]

### 4.2.2.1 Media móvil

La media móvil es uno de los indicadores técnicos más utilizados, una de las principales razones es que es muy versátil, y dependiendo del inversor, puede hacer uso de distintas medias móviles, ya que puede ser de 10 días, 20 días, o la cantidad deseada. Por otro lado, es un indicador muy sencillo de calcular, y fácil de verificar.

La media móvil, como su primer nombre lo indica se trata de un promedio de una serie de datos. Por otro lado, su segundo nombre hace referencia a que se trata de una media que, en un ejemplo de 20 días, se hace el cálculo de estos 20 y se va trasladando y calculando el valor por cada día. Por ende, si se tiene una serie de 20 datos, sólo se podrá calcular la media móvil de 20 días en una instancia.

$$MA = \frac{1}{n} \sum_{i=0}^{n-1} C_{t-i}$$

$C_t$  = Valor de cierre para el día t

n = Período de días

A partir de las medias móviles, se pretende utilizar las medias móviles de 50 y 200 días. La razón de esto es que son muy buenas y utilizadas en el área del trading, esto se debe a que, según registros del pasado, y experiencias de inversores, ocurre una característica llamada golden cross y death cross, que traducidas al español son cruce dorado y cruce de la muerte. Estas dos son opuestas entre ellas. La primera hace referencia a que si una media de corto plazo (50 días) rompe al alza una de largo alcance (200 días), podría significar un mercado alcista, mientras que su contraparte, hace referencia a que, si una media móvil de período corte rompe a la baja la de larga duración, podría significar un mercado bajista.

#### 4.2.2.2 Media móvil exponencial

A continuación, se pretende utilizar otro indicador de tendencia, pero en este caso, la media móvil exponencial. A diferencia de la media móvil tradicional, esta busca dar mayor importancia y peso a aquellos datos más recientes, de manera, que los datos más antiguos tengan una menor relevancia al momento de los cálculos.

La razón por la cual, se pretende incorporar otra media móvil, es que, dependiendo de situaciones, es recomendable hacer uso de una exponencial o una simple. Y es por esto, que puede resultar importante agregar la información de estos datos.

El cálculo de la media móvil exponencial, consiste en calcular para cada día el valor de EMA, y, por consiguiente, realizar una media de todos esos valores, el cálculo del valor EMA, para un Cierre en concreto, se representa a continuación

$$EMA = \left( C \times \left( \frac{2}{\frac{d+1}{1+d}} \right) \right) + EMA_y \times \left( 1 - \left( \frac{2}{\frac{d+1}{1+d}} \right) \right)$$

C = Cierre del día a calcular

$EMA_y$  = Valor de EMA del día anterior

d = período de días de la media

Para el desarrollo del trabajo se hará uso de la EMA de 200 y 50 días.

#### 4.2.2.3 RSI

Cómo fue mencionado anteriormente, el RSI es un indicador de impulso, en específico, se encarga de medir la magnitud de cambio reciente en los precios de la acción, para evaluar si existe sobrecompra o sobreventa.

Este indicador fue creado J. W. Wilder jr, en el artículo “New concepts in technical trading systems” [18]. En este trabajo, se llega a la conclusión de que el mejor período de días para realizar el cálculo del RSI es de 14 días.

El cálculo de este indicador consta de dos partes, la primera, es a través de la siguiente fórmula

$$RSI = 100 - \left( \frac{100}{1 + \frac{\text{promedio de ganancia}}{\text{promedio de pérdida}}} \right)$$

Para el cálculo de los promedios de ganancia y pérdida, lo que se procede es a calcular la media de ganancia, aquellos días en los que el mercado cerro positivo, y en caso contrario, la media de los cierres negativo es importante destacar que, para el caso de los valores de pérdida, se trataran en valor positivo. Por otro lado, el período de días que se toma en cuenta en este caso será de 14 días, como lo sugiere la documentación sobre este indicador.

En cuanto a los valores del RSI pueden estar entre 100 y 0, sin embargo, lo importante de este indicador es lo que representa. Ya que cuando su valor es mayor de 70, significa que se está en un estado de sobre compra, por ende, puede suponer la iniciación de una tendencia bajista. Por otro lado, cuando el valor está por debajo de 30, significa que se está en un estado de sobreventa, y, por consiguiente, puede suponer el inicio de una tendencia alcista.

#### **4.2.2.4 Bandas de Bollinger**

Las bandas de Bollinger fueron un indicador técnico creado por John Bollinger en 1980 [19], este indicador es construido a partir de una media móvil, junto a dos desviaciones estándar tanto positiva y negativa, y se representa mediante tres valores, la banda superior, inferior, y del medio, de manera que las velas, quedan encerradas entre dos líneas. Es importante destacar que la banda del medio siempre es una media, y que las superiores e inferiores se calculan de otra manera.

El cálculo de las bandas de Bollinger se representa de la siguiente manera

$$UPBAND = MA_{20} + (2 \times \sigma_{20})$$

$$BOLM = MA_{20}$$

$$LOWBOLD = MA_{20} - (2 \times \sigma_{20})$$

$$MA_{20} = \text{Media móvil de 20 días}$$

$$\sigma_{20} = \text{desviación estándar del precio en 20 días}$$

La razón de utilizar 20 días es debido a que las bandas de Bollinger es recomendable utilizar esta cantidad de días, y, por ende, es lo que esta extendido de manera tradicional para el uso de este indicador.

Las bandas de Bollinger son un indicador muy completo ya que se pueden identificar distintos patrones con ella. En primer lugar, se dice que cuanto más se acerquen los precios a la banda superior, más sobrecomprado estará el mercado, y cuanto más se acerquen los precios a la banda inferior, más sobrevendido estará el mercado. Por otro lado, cuando las dos bandas se amplían, se dice que se está en un momento de volatilidad. Sin embargo, cuando estas se contraen, se puede decir que el mercado está en un momento de baja volatilidad.

Sumando todos los indicadores mencionados, la representación de los datos queda de la siguiente manera.



Figura 12. Representación gráfica de los datos con indicadores utilizados. Fuente: Elaboración propia

### 4.2.3 Series temporales

Como fue descrito en el estado de la cuestión, los datos a tratar se pueden representar como un problema de series temporales. Ya que los datos tienen un orden cronológico, y se podría decir que los datos de un día dependen en cierta medida del valor del día anterior.

A partir de esto, el sistema de predicción a montar intentará buscar el valor de cierre un día, partiendo de datos de días anteriores, es decir

$$\text{Cierre}[t-n], \text{Cierre}[t-n-1], \text{Cierre}[t-n-2], \dots, \text{Cierre}[t-1], \text{Cierre}[t] \Rightarrow \text{Cierre}[t+1]$$

Sin embargo, el Sistema no solo toma en cuenta el cierre, sino que también se pretenden introducir ciertos indicadores técnicos, y además de esto, hay que definir, cuantos días previos se tomarán en cuenta.

Para trabajar el problema, se plantea utilizar un retardo de 5 días, de manera que se juntan los indicadores de 5 días previos junto al cierre para predecir el siguiente día. Por ende, cada día vendrá representado con la siguiente información:

- MA200
- MA50

- EMA200
- EMA50
- RSI
- UPBAND
- MIDBAND
- LOWBAND
- CIERRE

De esta manera, la entrada del modelo será un set de 45 datos, constituido por los últimos 5 días, donde para cada día se conforma de los datos mencionados anteriormente.

### **4.3. Metodología**

La metodología para evaluar los modelos resulta en un proceso bastante sencillo que consta en primer lugar dividir en conjunto de datos en tres partes, una para entrenar, la siguiente para evaluar el modelo, al cual se llamará conjunto de test, y, por último, un conjunto de validación, el cual se encarga de realizar la comparativa entre los mejores modelos de cada algoritmo de aprendizaje automático.

Para generar los modelos se pretende realizar un grid search entre una serie de parámetros, y de esta manera probar las posibles configuraciones, con esto, se podría saber entre todas las combinaciones que tan bueno es el modelo.

Para saber que tan bueno es el modelo, se utilizarán dos métodos, el primer será calcular la raíz del error cuadrático medio de las predicciones, de esta manera, se puede saber que tanto error se comete, por otro lado, se hará un contraste de hipótesis, para saber, si el mejor modelo generado, realizar predicciones tan buenas como el resultado real.

### **4.4. Diseño de GUI**

#### **4.4.1. Especificación de casos de uso**

##### **4.4.1.1. Descripción de actores**

Los actores son personas que actúan con nuestro sistema, para este proyecto, solo existirá el Usuario, ya que todas las personas van a tener la posibilidad de acceder a las mismas funciones, por otro lado, no existe ningún tipo de mantenimiento para el funcionamiento del programa, es por esto, que no es necesario ningún otro tipo de actor que intervenga en el sistema

##### **4.4.1.2. Descripción de tablas y plantillas de casos de uso**

Para poder describir los casos de uso, se plantea un diagrama que muestra como los actores interactúan con el sistema, por la sencillez del sistema, solo existirá un actor, y a su vez, los casos de uso que este puede realizar, esto se puede observar en la Figura 13. dónde se pretende describir cada uno de los atributos del diagrama

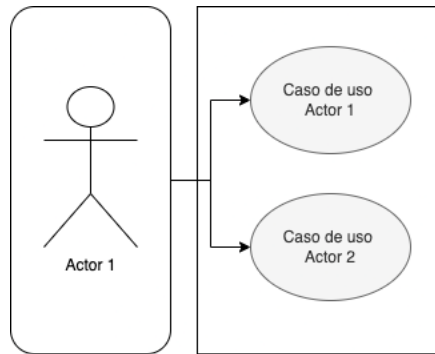


Figura 13. Plantilla de diagrama de casos de usos

- Actor 1: Actor identificado que llevan a cabo diferentes casos de uso en el sistema.
- Caso de uso: Casos de uso identificados llevados a cabo respectivamente por los actores.

En cuanto a los casos de usos, se ha hecho una selección de diversos atributos que permitan describir con exactitud cada uno de los casos, además de esto, se plantea una tabla para recoger estas características, como se puede observar en la Tabla 2.

TABLA 2. PLANTILLA DE CASOS DE USO

CUXXX – Caso de uso: nombre del caso de uso	
Actores	...
Objetivo	...
Precondiciones	...
Postcondiciones	...

- CUXXX: XXX representa el número identificativo del caso de uso
- Actores: Actores implicados en el caso de uso
- Objetivo: Tarea del caso de uso
- Precondiciones: condición que se debe cumplir para poder realizar el caso de uso
- Postcondiciones: Estado en el que queda el sistema al realizar el caso de uso

#### 4.4.1.3. Descripción textual de casos de usos

A continuación, se exponen los casos de usos identificados para el desarrollo del sistema, para esto, se hace uso de las plantillas mencionadas en el apartado anterior.

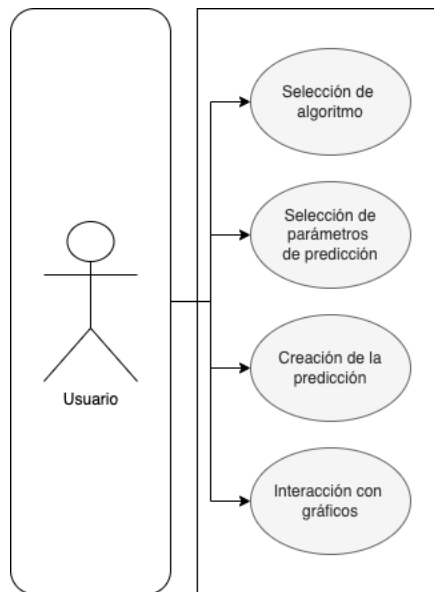


Figura 14. Diagrama de casos de uso

TABLA 3. CASO DE USO NÚMERO 1

CU001 – Caso de uso: selección del algoritmo	
Actores	Usuario
Objetivo	El usuario selecciona una lista de entre los algoritmos que se pueden utilizar, en este caso SVR, Decission Tree Regressor, o Random Forest Regressor
Precondiciones	El usuario debe de estar en la interfaz principal, y selecciona el algoritmo que quiere utilizar
Postcondiciones	Se muestran los posibles parámetros para ese algoritmo



TABLA 4. CASO DE USO NÚMERO 2

CU002 – Caso de uso: selección de parámetros de predicción	
Actores	Usuario
Objetivo	El usuario selecciona los parámetros de la predicción
Precondiciones	El usuario debe de haber seleccionado el algoritmo del modelo
Postcondiciones	Se cambian los parámetros del modelo

TABLA 5. CASO DE USO NÚMERO 3

CU003 – Caso de uso: creación de la predicción	
Actores	Usuario
Objetivo	El usuario pulsa el botón para crear la predicción
Precondiciones	El usuario ha puesto todos los parámetros para la predicción y pulsa continuar
Postcondiciones	Se carga el modelo y se crea la predicción

TABLA 6. CASO DE USO NÚMERO 4

CU004 – Caso de uso: interacción con gráfica	
Actores	Usuario
Objetivo	El usuario puede interactuar con la gráfica de predicción
Precondiciones	El usuario se mueve por la gráfica
Postcondiciones	La gráfica se mueve según el criterio del usuario

#### 4.4.2. Especificación de requisitos

##### 4.4.2.1. Descripción de tablas y plantillas de requisitos

En este apartado identificaremos los requisitos que debe tener nuestro sistema, para su correcto funcionamiento, para recoger esta información, se hará uso de una tabla, como se muestra en la Tabla 7. y a su vez, una serie de atributos, que permitan identificar y detallar cada uno de estos requisitos.

- **Título:** En primer lugar, se tiene el título, el cual posee las siguientes características
  - RS: Hace referencia a requisitos del sistema
  - RY: Puede tomar los valores RF o RNF que hacen referencia a requisitos funcional y requisito no funcional respectivamente
  - XXX: Número identificativo del requisito
- **Fuente:** Hace referencia a cómo ha sido identificado el requisito del sistema, en este caso, vendrán de los casos de uso identificados
- **Verificabilidad:** Indica la medida para comprobar el requisito, es decir, permite saber en qué medida se puede probar el requisito en el sistema, los posibles valores son los siguientes
  - Alta: Permite de manera sencilla, comprobar que el requisito se cumple, normalmente hacen referencia a los requisitos básicos del sistema
  - Media: Es posible verificar que el requisito ha sido implementado, sin embargo, a veces puede ser difícil, ya que dependerá del código fuente, u otros factores
  - Baja: Resulta difícil comprobar si el requisito ha sido implementado, y en algunos casos, resulta difícil
- **Necesidad:** Indica en que grado el requisito debe de ser implementado en el sistema, para este requisito se tienen los siguientes valores
  - Esencial: El requisito debe de obligatoriamente ser implementado en el sistema
  - Deseable: Se debería de implementar el requisito, sin embargo, no es obligatorio
  - Opcional: El requisito se podría implementar, pero no es obligatorio
- **Estabilidad:** Describe la consistencia del requisito a lo largo de la vida del proyecto. Casi todos se establecen para que duren toda la vida del proyecto.
  - Estable: Debe de durar durante toda la vida del sistema
  - Inestable: El requisito puede variar durante la vida del sistema
- **Nombre:** Título del requisito
- **Descripción:** Detalle del requisito

TABLA 7. PLANTILLA DE REQUISITOS

RS-RY-XXX			
Fuente		Verificabilidad	
Necesidad		Estabilidad	
Nombre			
Descripción			

#### 4.4.2.2. Descripción textual de requisitos

TABLA 8. REQUISITO FUNCIONAL NÚMERO 1

RS-RF-001			
Fuente	CU001	Verificabilidad	Alta
Necesidad	Esencial	Estabilidad	Estable
Nombre	Selección de algoritmo de predicción		
Descripción	El usuario podrá seleccionar entre un listado de algoritmos de aprendizaje automático		

TABLA 9. REQUISITO FUNCIONAL NÚMERO 2

RS-RF-002			
Fuente	CU002	Verificabilidad	Alta
Necesidad	Esencial	Estabilidad	Estable
Nombre	Selección de acción		
Descripción	El usuario podrá seleccionar un listado de acciones del mercado de valores para realizar la predicción		

TABLA 10. REQUISITO FUNCIONAL NÚMERO 3

RS-RF-003			
Fuente	CU002	Verificabilidad	Alta
Necesidad	Esencial	Estabilidad	Estable
Nombre	Parámetros de predicción		
Descripción	El usuario podrá rellenar una serie de parámetros del algoritmo de predicción		

TABLA 11. REQUISITO FUNCIONAL NÚMERO 4

RS-RF-004			
Fuente	CU003	Verificabilidad	Alta
Necesidad	Esencial	Estabilidad	Estable
Nombre	Crear predicción		
Descripción	El usuario podrá realizar una predicción una vez rellenado los parámetros		

TABLA 12. REQUISITO FUNCIONAL NÚMERO 5

RS-RF-005			
Fuente	CU004	Verificabilidad	Alta
Necesidad	Esencial	Estabilidad	Estable
Nombre	Gráfica de predicción		
Descripción	El usuario se le generará una gráfica de predicción con una predicción de los próximos 20 días, y una previsión pasada de los últimos 20 días		

TABLA 13. REQUISITO NO FUNCIONAL NÚMERO 6

RS-RNF-001			
Fuente	CU001, CU002	Verificabilidad	Alta
Necesidad	Esencial	Estabilidad	Estable
Nombre	Interfaz web		
Descripción	La interfaz de la página web se realizará mediante HTML y CSS		

TABLA 14. REQUISITO NO FUNCIONAL NÚMERO 2

RS-RNF-002			
Fuente	CU003	Verificabilidad	Alta
Necesidad	Esencial	Estabilidad	Estable
Nombre	Motor de predicción		
Descripción	Los modelos de predicción se realizan mediante el lenguaje de programación python		

TABLA 15. REQUISITO NO FUNCIONAL NÚMERO 3

RS-RNF-003			
Fuente	CU001, CU002	Verificabilidad	Alta
Necesidad	Esencial	Estabilidad	Estable
Nombre	Accesibilidad		
Descripción	El programa se podrá acceder desde el equipo donde se instale		

#### 4.4.3. Diseño de la aplicación

El siguiente apartado tiene como objetivo plasmar todas las decisiones de diseño realizadas para el desarrollo del sistema, de esta manera, se detallan los componentes que intervienen en el sistema.

#### 4.4.3.1. Arquitectura del sistema

Para la arquitectura del sistema, se plantea una estructura de tres capas, conocida como modelo vista controlador, como se muestra en la Figura 15. De esta manera, el sistema queda dividido en tres componentes

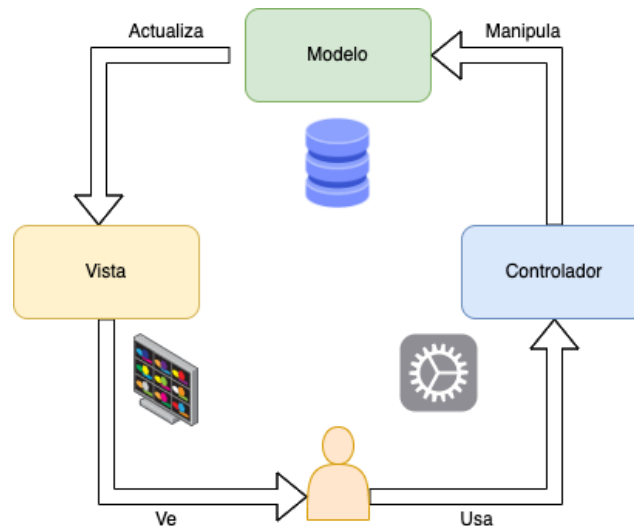


Figura 15. Representación del modelo-vista-controlador. Fuente: Elaboración propia

- **Modelo:** Parte del sistema, encargada de representar la lógica del sistema, en la mayoría de los casos, es el sistema encargado de almacenar la información, en nuestro caso, será la parte del sistema encargada de almacenar y crear el modelo de predicción
- **Vista:** Parte del sistema encargada de representar la información del modelo al usuario, a través de la interfaz gráfica, la vista, tiene acceso al modelo, sin embargo, no puede realizar modificaciones sobre esta.
- **Controlador:** Esta parte se encarga de reaccionar ante las peticiones del usuario, de esta manera, se generan los cambios en el modelo, y en este caso, se envía las peticiones para realizar las predicciones



## 5. RESULTADOS EXPERIMENTALES

En primer lugar, se pretende hacer un estudio de la muestra, como fue mencionado anteriormente, los datos a utilizar son del ETF SPY, los datos recogidos van desde el 29 de enero del 1993 hasta el 2 de noviembre del 2021. El total de instancias tras el cálculo de los indicadores es de 7220, y tras la generación de la serie temporal quedan un total de 7215 instancias. De este total de instancias, se destinarán 4905 para entrenar, 1227 para el conjunto de test y por último 1083 para hacer validación de los modelos

Según los indicadores mencionados anteriormente, a continuación, se muestran los principales estadísticos descriptivos de las variables a utilizar.

TABLA 16. ESTADÍSTICO DESCRIPTIVO DE LOS DATOS UTILIZADOS

	CIERRE	EMA50	EMA200	MA50	MA200	RSI	UP BAND	MID BAND	LOW BAND
Media	154,66	153,28	149,38	153,27	149,18	54,46	158,89	154,11	149,34
Mediana	129,74	129,48	126,90	129,64	127,61	55,30	133,96	130,01	126,1
Máximo	461,90	445,41	419,48	445,46	420,84	87,03	466,99	448,49	442
Mínimo	43,91	45,28	45,21	45,12	45,21	16,70	45,37	44,70	43,38
Desv. Típ	83,40	81,63	77,10	81,7	76,80	11,17	84,96	82,70	80,62

A partir del procesamiento de los datos para aplicar los algoritmos de aprendizaje automático, se plantean algunos de los resultados obtenidos.

### 5.1 Perceptrón multicapa

Para el caso del perceptrón multicapa, se plantean probar distintas configuraciones, realizando un grid search de las configuraciones mencionadas en la Tabla 17, de esta manera, se llegará a la mejor configuración, que será aquella que tenga el menor error.

TABLA 17. CONFIGURACIÓN DE PARÁMETROS DEL PERCEPTRÓN MULTICAPA

Parámetro	Valor	Valor	Valor	Valor	Valor	Valor
Capa Oculta	100	200	3	10	700	(500,200)
Función de activación	Relu			Tanh		
Learning Rate	0.001		0.005		0.004	
Cambio del Learning Rate	Fijo			Adaptativo		
Iteraciones de entrenamiento	2000					

En el caso de esta experimentación, se pueden observar los resultados obtenidos en la Tabla 18 tras 10 ejecuciones por configuración, en esta se puede ver como la red que mejor se ajustó a los datos de test, fue la configuración con un learning rate adaptativo de valor 0,001, así como una capa oculta con 700 neuronas y función de activación relu, además de un máximo de 2000 iteraciones, de manera que se pueda realizar un early stopping, además de esto, es importante destacar que la capa de entrada consta de 45 neuronas, que corresponde a los atributos que se utilizan

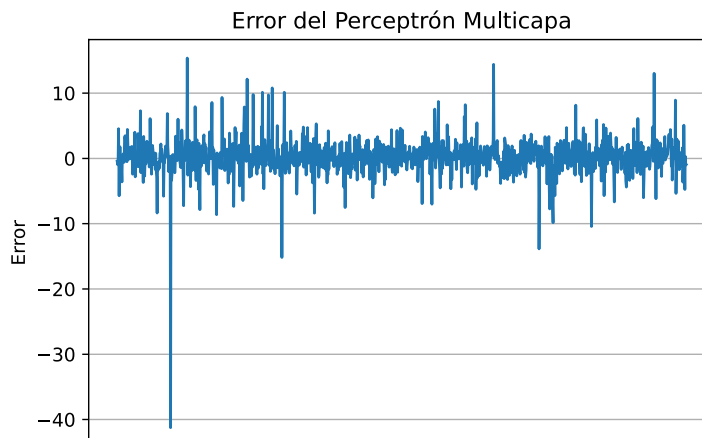


Figura 16. Error cometido por el perceptrón multicapa en todo el conjunto de test para el mejor modelo

Este modelo presentó un error de 2,7449, y se puede observar su comportamiento en la Figura 16.

Por otro lado, hay que comentar que las configuraciones obtenidas, presentan resultados variados, sin embargo, los experimentos con pocas neuronas convergen en errores más altos, esto se puede deber a que, con pocas neuronas, no existen grados de libertad suficiente que dejen al modelo, ajustarse correctamente, y por ende, no logra modelar los datos.



TABLA 18. EXPERIMENTACIÓN DEL PERCEPTRÓN MULTICAPA

Capa Oculta	F. activación	L. rate	L. rate change	RMSE
700	relu	0,001	Adaptive	2,7449
[500, 200]	relu	0,001	Adaptive	3,0494
700	relu	0,005	Constant	3,0848
[500, 200]	relu	0,004	Adaptive	3,0999
200	relu	0,005	Adaptive	3,1266
200	relu	0,001	Adaptive	3,1307
700	relu	0,001	Constant	3,2021
200	relu	0,004	Adaptive	3,3979
[500, 200]	relu	0,004	Constant	3,4445
[500, 200]	relu	0,001	Constant	3,4458
200	tanh	0,005	Adaptive	3,4982
[500, 200]	relu	0,005	Adaptive	3,5952
[500, 200]	tanh	0,001	Constant	3,5960
[500, 200]	tanh	0,001	Adaptive	3,6596
700	tanh	0,004	Adaptive	3,6781
700	relu	0,004	Adaptive	3,7098
100	relu	0,005	Constant	3,7426
100	tanh	0,004	Adaptive	3,7437
700	relu	0,005	Adaptive	3,7946
700	relu	0,004	Constant	3,8041
200	relu	0,001	Constant	3,8386
700	tanh	0,001	Adaptive	3,8610
700	tanh	0,005	Constant	3,9298
100	relu	0,005	Adaptive	4,0468
700	tanh	0,004	Constant	4,0693
100	relu	0,004	Constant	4,0940
700	tanh	0,001	Constant	4,0966
200	tanh	0,001	Adaptive	4,1452
200	tanh	0,001	Constant	4,1722
[500, 200]	tanh	0,004	Adaptive	4,2176
[500, 200]	relu	0,005	Constant	4,2560
100	relu	0,004	Adaptive	4,3700
200	tanh	0,005	Constant	4,4394
200	relu	0,004	Constant	4,4840
200	relu	0,005	Constant	4,4890
200	tanh	0,004	Adaptive	4,5998
100	tanh	0,001	Adaptive	4,6190
200	tanh	0,004	Constant	4,6208
700	tanh	0,005	Adaptive	4,6443
100	tanh	0,004	Constant	4,9107

Capa Oculta	F. activación	L. rate	L. rate change	RMSE
[500, 200]	tanh	0,005	Constant	4,9511
100	relu	0,001	Adaptive	4,9683
[500, 200]	tanh	0,004	Constant	5,1627
10	relu	0,005	Constant	5,1906
100	relu	0,001	Constant	5,2343
100	tanh	0,005	Constant	5,3581
100	tanh	0,005	Adaptive	5,6809
10	tanh	0,004	Adaptive	5,9897
[500, 200]	tanh	0,005	Adaptive	6,1759
10	tanh	0,005	Adaptive	6,4208
10	relu	0,004	Adaptive	7,0099
10	tanh	0,005	Constant	7,2347
10	tanh	0,001	Constant	7,3554
10	relu	0,004	Constant	7,4298
100	tanh	0,001	Constant	7,7797
10	relu	0,001	Constant	7,9671
3	tanh	0,004	Constant	8,2410
3	tanh	0,005	Constant	8,6242
3	relu	0,001	Constant	8,9573
3	tanh	0,004	Adaptive	9,5336
10	relu	0,005	Adaptive	9,9945
10	relu	0,001	Adaptive	10,2330
10	tanh	0,004	Constant	10,6323
3	relu	0,005	Adaptive	11,0504
3	relu	0,005	Constant	12,7348
3	tanh	0,005	Adaptive	13,9150
3	relu	0,004	Adaptive	15,2618
3	relu	0,004	Constant	17,2726
10	tanh	0,001	Adaptive	19,2639
3	tanh	0,001	Adaptive	21,1996
3	tanh	0,001	Constant	24,5074
3	relu	0,001	Adaptive	95,5105

## 5.2 Decision Tree Regressor

En el caso del árbol de decisión, se plantea en la Tabla 19, las configuraciones a probar. Es importante destacar, que se realizara una combinación de todas las posibles opciones, para lograr obtener el mejor resultado

TABLA 19. CONFIGURACIÓN DE PARÁMETROS PARA DECISSION TREE REGRESSOR

Parámetro	Valor	Valor	Valor	
Función de error	Error cuadrático medio	Error absoluto medio	Poisson	
Profundidad Máxima	N/A	30	20	120
Muestras mínimas para dividir	2	4	30	
Técnica para dividir	Best	Random		

En cuanto a este modelo, se puede decir que el que obtuvo mejor resultados, fue la configuración que tenía profundidad máxima de 120 nodos, la estrategia de división utilizada fue Best junto a la función de error cuadrático medio, por último, las muestras mínimas para poder dividir son de 30 instancias.

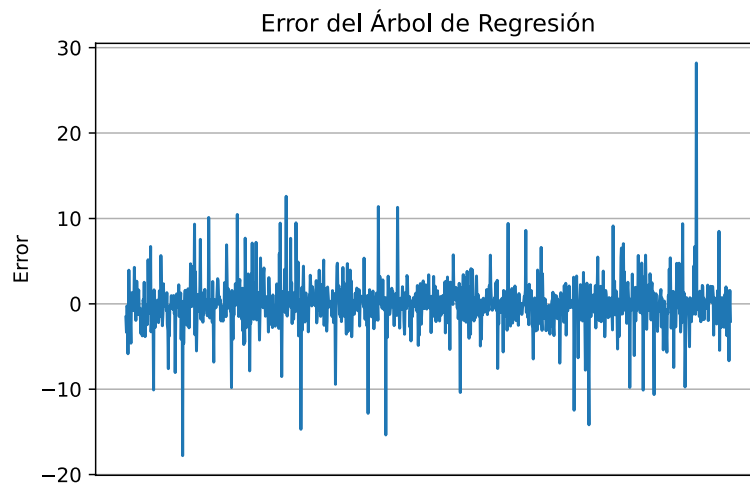


Figura 17. Error cometido por el árbol de regresión en todo el conjunto de test para el mejor modelo

En cuanto al error obtenido, este fue de 2,7485. Y su comportamiento es modelado en la Figura 17. Se puede decir que resulta en una tendencia parecida a la obtenida en las redes neuronales.

Por último, hay que destacar que el comportamiento de los árboles de decisión parece relativamente bueno, sin embargo, falta realizar un contraste de hipótesis, para saber que tan buenos son.

TABLA 20. EXPERIMENTACIÓN DE DECISION TREE REGRESSOR

F. error	Profundidad Maxima	Muestras para dividir	Estrategia de división	RMSE
MSE	120	30	Best	2,7485
MAE	120	30	Best	2,7676
MSE	30	30	Best	2,7712
MAE	None	30	Best	2,7713
MSE	None	30	Best	2,7745
MSE	20	30	Best	2,7873
MAE	30	30	Best	2,7897
MAE	20	30	Best	2,7971
Poisson	None	30	Best	2,9028
MSE	None	30	Random	2,9264
MSE	120	4	Best	2,9538
MSE	None	4	Random	2,9548
MSE	120	2	Random	2,9558
MSE	30	2	Best	2,9562
MSE	120	2	Best	2,9622
MSE	None	2	Best	2,9797
MSE	None	4	Best	2,9893
MAE	20	2	Best	3,0216
MAE	30	2	Best	3,0339
MSE	120	30	Random	3,0348
MAE	None	30	Random	3,0375
MSE	30	4	Best	3,0397
MSE	20	2	Best	3,0425
Poisson	None	2	Best	3,0478
MAE	120	4	Best	3,0486
MAE	None	4	Best	3,0564
MAE	30	4	Best	3,0622
MAE	20	4	Best	3,0672
MSE	30	2	Random	3,0859
MAE	120	2	Best	3,0882
MSE	None	2	Random	3,0929
MAE	None	2	Best	3,1092
MSE	20	4	Best	3,1475
MAE	120	4	Random	3,1810
MSE	20	30	Random	3,1923
MSE	30	4	Random	3,1991
MAE	30	4	Random	3,2122
MAE	30	30	Random	3,2178
Poisson	None	4	Best	3,2785

<b>F. error</b>	<b>Profundidad Maxima</b>	<b>Muestras para dividir</b>	<b>Estrategia de división</b>	<b>RMSE</b>
MAE	20	4	Random	3,2884
MSE	120	4	Random	3,3022
MSE	20	4	Random	3,3290
MAE	None	2	Random	3,3959
MSE	20	2	Random	3,4057
MAE	120	30	Random	3,4233
MSE	30	30	Random	3,5038
MAE	20	2	Random	3,5179
Poisson	None	2	Random	3,5315
MAE	120	2	Random	3,5598
MAE	20	30	Random	3,5658
MAE	None	4	Random	3,6146
MAE	30	2	Random	3,7221
Poisson	None	30	Random	4,0818
Poisson	None	4	Random	4,0960
Poisson	120	30	Random	5,1554
Poisson	120	2	Random	6,2321
Poisson	120	4	Random	12,4908
Poisson	30	30	Random	40,4077
Poisson	30	4	Random	40,5012
Poisson	30	2	Random	42,9419
Poisson	20	4	Random	49,9648
Poisson	20	2	Random	52,8706
Poisson	20	30	Random	54,2065
Poisson	120	30	Best	65,0629
Poisson	120	4	Best	65,2985
Poisson	120	2	Best	65,3004
Poisson	30	30	Best	83,1791
Poisson	30	4	Best	83,1803
Poisson	30	2	Best	83,1805
Poisson	20	2	Best	87,7904
Poisson	20	30	Best	87,7907
Poisson	20	4	Best	87,7907

### 5.3 Random Forest

A continuación, se pretenden diseñar los Random Forest, para realizar esto, se plantea probar todas las posibles configuraciones entre los siguientes parámetros, de esta manera, se consigue el mejor resultado en el conjunto de test.

TABLA 21. CONFIGURACIONES DE RANDOM FOREST

Parámetro	Valor	Valor	
Número de árboles	200	100	
Función de error	Error cuadrático medio	Error absoluto medio	Poisson
Profundidad Máxima	N/A	200	10
Muestras mínimas para dividir	2	60	
Muestras mínimas para ser hoja	1	20	

Tras 10 ejecuciones por configuración se ha conseguido que el mejor modelo que se ajusta al conjunto de test consta de un total de 100 árboles, con función de error absoluto medio, además de esto, se tiene una profundidad máxima de 10, las hojas tienen como mínimo una muestra, y al momento de dividir un nodo, como mínimo tendría que haber 2 muestras. Este modelo presenta un error cuadrático medio de predicción de 2,3212.

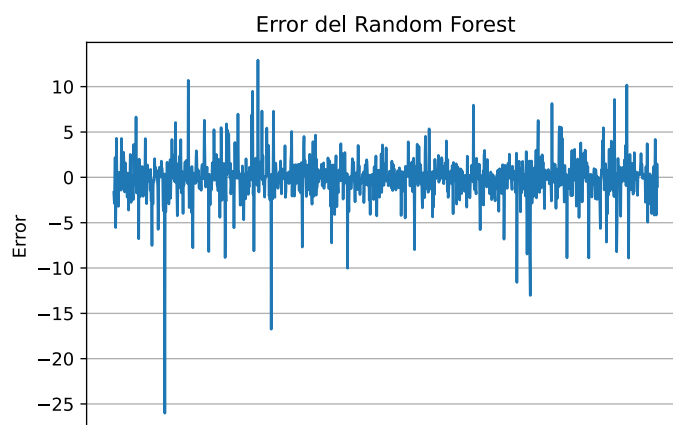


Figura 18. Error cometido por el Random Forest en todo el conjunto de test para el mejor modelo

En la Figura 18, se puede observar la tendencia que sigue el modelo. En este se puede observar, como existen numerosas instancias con error, sin embargo, se mantienen bastante bajas.

TABLA 22. EXPERIMENTACIÓN DE RANDOM FOREST

N. árboles	F. error	Profundidad Maxima	Muestras para dividir	Muestras hojas	Error
100	MAE	10	2	1	2,3212
200	MAE	10	2	1	2,3424
100	MAE	200	2	1	2,3435
200	MSE	None	2	1	2,3468
200	MSE	10	2	1	2,3520
100	MAE	None	2	1	2,3542
200	MAE	200	2	1	2,3567
200	MSE	200	2	1	2,3573
100	MSE	10	2	1	2,3591
100	MSE	None	2	1	2,3632
200	MAE	None	2	1	2,3645
100	MSE	200	2	1	2,3652
100	Poisson	None	2	20	2,4276
100	Poisson	200	2	20	2,4405
200	Poisson	None	2	20	2,4416
100	Poisson	None	60	20	2,4466
100	Poisson	200	60	20	2,4477
200	Poisson	200	2	20	2,4497
200	Poisson	None	60	20	2,4508
200	MSE	10	2	20	2,4527
200	MSE	None	2	20	2,4530
200	Poisson	200	60	20	2,4548
100	MSE	None	2	20	2,4580
100	MSE	10	2	20	2,4587
200	MSE	200	2	20	2,4635
100	MSE	200	2	20	2,4790
200	MSE	None	60	20	2,5236
200	MSE	10	60	20	2,5411
200	MSE	200	60	20	2,5480
100	MSE	10	60	20	2,5501
100	MAE	None	2	20	2,5574
100	MSE	None	60	20	2,5657
100	MSE	200	60	20	2,5665
200	MAE	None	2	20	2,5667
200	MSE	10	60	1	2,5745
200	MSE	200	60	1	2,5755
100	MSE	10	60	1	2,5855
200	MAE	200	2	20	2,5980
100	MSE	200	60	1	2,5983

N. árboles	F. error	Profundidad Maxima	Muestras para dividir	Muestras hojas	Error
200	MSE	None	60	1	2,6064
100	MAE	10	2	20	2,6082
200	MAE	10	2	20	2,6126
100	MAE	200	2	20	2,6268
100	MSE	None	60	1	2,6308
200	MAE	None	60	20	2,8170
100	MAE	200	60	1	2,8273
200	MAE	10	60	20	2,8277
200	MAE	200	60	1	2,8292
200	MAE	None	60	1	2,8296
100	MAE	200	60	20	2,8304
200	MAE	10	60	1	2,8428
100	MAE	None	60	1	2,8487
100	MAE	10	60	1	2,8503
100	MAE	10	60	20	2,8541
200	MAE	200	60	20	2,8591
100	MAE	None	60	20	2,8615
100	Poisson	None	2	1	3,5458
200	Poisson	None	2	1	3,6998
100	Poisson	None	60	1	3,7180
200	Poisson	None	60	1	3,7227
100	Poisson	200	2	1	40,1540
200	Poisson	200	2	1	40,3117
100	Poisson	200	60	1	40,3557
200	Poisson	200	60	1	40,4224
200	Poisson	10	2	20	65,0566
200	Poisson	10	60	20	65,1572
100	Poisson	10	60	20	65,1698
100	Poisson	10	2	20	65,2206
200	Poisson	10	60	1	89,1991
100	Poisson	10	2	1	89,2168
200	Poisson	10	2	1	89,3351
100	Poisson	10	60	1	89,4327



## 5.4 SVR

Por último, se plantea en la Tabla 23, las configuraciones a experimentar para el SVR. En este caso, se plantean probar únicamente dos Kernel, debido a lo complejo que puede llegar a ser este algoritmo.

TABLA 23. CONFIGURACIÓN DE SVR

Parámetro	Valor	Valor	Valor	Valor	Valor	Valor
Kernel	Linear			RBF		
Epsilon	0	0,1	0,5	10	100	1000
C	1	0,05	0,01	0,001	0,0001	
Iteraciones	2000					

En cuanto a los resultados obtenidos, la configuración con el mejor resultado fue realizada con un kernel lineal, y con parámetros de  $\epsilon$  y C, de 0,1 y 0,001 respectivamente. En este experimento se obtuvo un error de 2,3036.

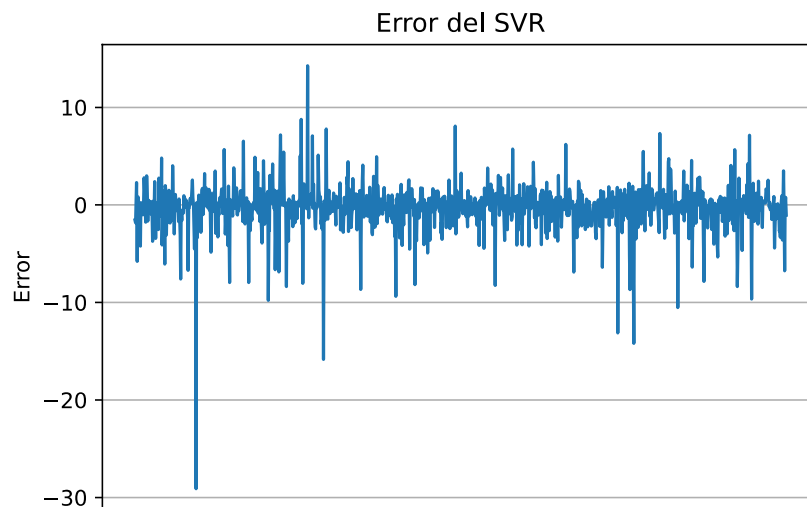


Figura 19. Error cometido por el SVR en todo el conjunto de test para el mejor modelo

En cuanto a la gráfica de error, podemos observar su tendencia. Y en cuanto al resto de experimentos, se puede decir que el kernel RBF, no presenta buenos resultados, ya que no se ajusta de manera adecuada a los datos.

TABLA 24. EXPERIMENTACIÓN DE SVR

Kernel	Epsilon	C	RMSE
Lineal	0,1	0,001	2,3036
Lineal	0,5	0,01	2,3092
Lineal	0,5	0,001	2,3160
Lineal	0	0,05	2,4422
Lineal	0	0,001	2,4751
Lineal	0,1	0,0001	2,5198
Lineal	0,5	0,0001	2,5212
Lineal	0	0,0001	2,5262
Lineal	0,1	0,01	2,5415
Lineal	0	0,01	2,5572
Lineal	0	1	2,6325
Lineal	10	0,01	2,6355
Lineal	0,5	0,05	2,6364
Lineal	10	1	2,7042
Lineal	10	0,001	2,8436
Lineal	0,1	1	2,8765
Lineal	10	0,05	2,9015
Lineal	0,5	1	3,2463
Lineal	10	0,0001	4,1540
Lineal	0,1	0,05	5,7085
RBF	0,5	1	8,9396
RBF	0	1	9,9747
RBF	10	1	10,3128
RBF	0,1	1	10,4247
Lineal	100	1	34,3667
Lineal	100	0,05	34,3667
Lineal	100	0,01	34,3667
Lineal	100	0,001	34,3667
Lineal	100	0,0001	34,3667
RBF	100	1	67,7142
RBF	10	0,05	70,2285
RBF	0,5	0,05	70,9478
RBF	0,1	0,05	71,1069
RBF	0	0,05	71,1521
RBF	100	0,05	86,1714
RBF	10	0,01	93,7120
RBF	100	0,01	93,9093
RBF	0	0,01	94,9025
RBF	0,1	0,01	95,1077
RBF	0,5	0,01	95,6070

Kernel	Epsilon	C	RMSE
RBF	100	0,001	96,3017
RBF	100	0,0001	96,6046
RBF	10	0,001	100,1736
RBF	10	0,0001	100,8584
RBF	0	0,001	101,3960
RBF	0,1	0,001	101,7221
RBF	0	0,0001	102,0648
RBF	0,5	0,001	102,2164
RBF	0,1	0,0001	102,3897
RBF	0,5	0,0001	102,8802
RBF	1000	1	135,8534
RBF	1000	0,05	135,8534
RBF	1000	0,01	135,8534
RBF	1000	0,001	135,8534
RBF	1000	0,0001	135,8534
Lineal	1000	1	189,6632
Lineal	1000	0,05	189,6632
Lineal	1000	0,01	189,6632
Lineal	1000	0,001	189,6632
Lineal	1000	0,0001	189,6632

## 5.5 Comparativa de modelos

Tras haber probado numerosas configuraciones de distintos algoritmos de aprendizaje automático, toca estudiar con el conjunto de validación cual presenta los mejores resultados. Para hacer esto, se pretende para cada técnica, agarrar el mejor resultado del apartado anterior y una vez esto, realizar la experimentación 10 veces, estudiando el error en el conjunto de validación, a partir de esto, se hace un estadístico descriptivo de todas las ejecuciones, la cual se recoge en la Tabla 25.

TABLA 25. ESTADÍSTICO DESCRIPTIVO DEL RMSE OBTENIDO EN EL CONJUNTO DE VALIDACIÓN

Algoritmo	Media	Mediana	Desv. Típica	Máximo	Mínimo
Perceptrón Multicapa	3,4568	3,4022	0,1885	3,7025	3,1870
Random Forest	2,5159	2,5166	0,015267	2,5436	2,4911
Decission Tree	2,8945	2,8946	0,0061	2,9047	2,8871
SVR	2,4255	2,3066	0,3154	3,3099	2,2735

La evaluación de la significatividad de las diferencias de las medianas de los errores de predicción se evaluó formalmente a través de un contraste de Wilcoxon [20]. En ningún caso se pudo rechazar la hipótesis básica de que la capacidad predictiva de los cuatro modelos fuese idéntica

TABLA 26. SIGNIFICACIÓN ESTADÍSTICA DE LA DIFERENCIA DE ERRORES MEDIO COMETIDO EN EL CONJUNTO DE VALIDACIÓN

	Valor Real	Árbol de Regresión	Perceptrón Multicapa	Random Forest
Árbol de Regresión	=			
Perceptrón Multicapa	=	=		
Random Forest	=	=	=	
SVR	=	=	=	=



## 6. ANÁLISIS DE RESULTADOS

En primer lugar, hay que hablar de los resultados obtenidos para el conjunto de test de los algoritmos utilizados. El SVR presenta el menor error con un valor de 2,3036, por otro lado, el Random Forest en segundo lugar un error de 2,3212 y en últimas posiciones el perceptrón multicapa y Decisión Tree Regressor con un error de 2,7449 y 2,7485 respectivamente.

Se puede decir que los resultados obtenidos a simple vista son bajos, y que entre los modelos no presentan una diferencia notoria, sin embargo, esto no nos da información de cómo se comportarán los modelos en acción, es por esto, que se necesita un conjunto de validación para saber entre estos modelos, cuál es el mejor.

En la Tabla 25, se pueden observar los resultados obtenidos para el conjunto de validación donde una vez mas el algoritmo SVR presenta el menor error, con un valor de 2,4255 de media sobre 10 ejecuciones. En segunda posición queda el Random Forest con un valor de 2,5159, seguido del Decission Tree y el Perceptrón multicapa, cuyos errores fueron de 2,8945 y 3,4568 respectivamente.

Tras realizar estas pruebas, se puede decir que los resultados obtenidos parecen relativamente buenos, debido a que el cambio de error entre conjunto de datos no fue tan elevado. Sin embargo, queda por último paso realizar un contraste estadístico entre los valores predichos y los reales, para de esta manera saber la veracidad de los datos.

En la tabla 26, se puede observar los resultados obtenidos en el contraste de Wilcoxon [20], donde se puede observar que todos los modelos no son capaces de rechazar la hipótesis básica y, por ende, las predicciones son estadísticamente iguales.

Es importante destacar que el SVR fue el que tuvo mejor resultados en todas las pruebas realizadas, pero por contraparte, se tiene que es un algoritmo bastante costoso y lento de entrenar, por la complejidad computacional<sup>9</sup> que puede llegar a conseguir. Por otro lado, en cuanto a complejidad, se puede llegar a decir que el Random Forest puede llegar a ser muy costoso, debido a que un modelo puede tener 700 árboles, y que el algoritmo debe de entrenar cada uno de estos por separado, lo que converge en tiempos de ejecución bastante elevados.

En cuanto a la utilización de estos modelos en la vida real, quedaría para estudios posteriores estudiar la rentabilidad de las predicciones en el largo plazo para poder sacar beneficio del mercado de valores. Es importante destacar que estos modelos predictivos creados sean buenos para el corto plazo, y que es muy difícil saber si en un futuro sean asertivos, ya que el mercado al tener un comportamiento tan aleatorio, y funcionar por “temporadas”, hace que quizás la conducta sea muy buena durante los primeros días del entrenamiento, pero en un futuro quizás no sean lo óptimo.

---

<sup>9</sup> Recursos requeridos para resolver el problema

## **7. GESTIÓN DEL PROYECTO**

Este capítulo tiene como objetivo describir toda la planificación llevada a cabo para realizar con éxito del proyecto. Para hacer esto, se plantea una serie de fases identificadas durante el desarrollo del trabajo, y una vez esto, se plantea una planificación del tiempo necesario para llevar a cabo cada una de las fases descritas. Además de esto, se plantea un presupuesto que permite tantear el coste económico del trabajo, y por último, se tiene el marco regulador de esta investigación

### **7.1. Fases del proyecto**

A continuación, se describen las tareas identificadas durante el desarrollo del trabajo, de manera que se pueda realizar una planificación

1. Definición del problema: Esta fase tiene como objetivo identificar el problema que se quiere abordar, así como los objetivos de este.
2. Requisitos del sistema: Esta parte del desarrollo del trabajo tiene como objetivo estudiar cómo se va a abordar el problema, y a su vez, las especificaciones que debe tener el trabajo
3. Diseño: Se describe la estructura del sistema
4. Implementación: Una vez se ha estudiado las necesidades del sistema, y cómo será este, se procede a implementar el sistema
5. Pruebas: Una vez se ha desarrollado el sistema, se procede a realizar diversas pruebas, de manera que se compruebe que todo funcione de manera correcta, y en caso contrario, se procede a realizar las correcciones
6. Experimentación: Para el desarrollo de la comparativa de los algoritmos, se procede a realizar los experimentos, y de esta manera, se consiguen aquellos algoritmos que mejor funcionen
7. Documentación: Fase del proyecto, en la cual, se realiza la documentación necesaria del trabajo
8. Presentación: Proceso en el que se prepara todo para la presentación del trabajo

### **7.2. Planificación**

A partir de las etapas identificadas en el apartado anterior para poder llevar a cabo el proyecto, se procede a realizar una planificación, de manera que quede reflejado todo el proceso llevado a cabo en el trabajo.

En primer lugar, se plantea una tabla, de los tiempos llevados para poder desarrollar el proyecto, y por otro lado, un diagrama de Gantt, de manera que se pueda visualizar el progreso llevado a cabo durante el trabajo

TABLA 27. TIEMPOS LLEVADOS A CABO PARA LA REALIZACIÓN DEL PROYECTO

Tarea	Duración	Comienzo	Fin
Definición del problema	5 días	25/04/2022	30/04/2022
Requisitos del sistema	5 días	01/05/2022	05/05/2022
Diseño	10 días	06/05/2022	16/05/2022
Implementación	14 días	17/05/2022	31/05/2022
Pruebas	5 días	01/06/2022	05/06/2022
Experimentación	19 días	06/06/2022	25/06/2022
Documentación	30 días	01/07/2022	30/07/2022
Presentación	10 días	01/08/2022	10/08/2022

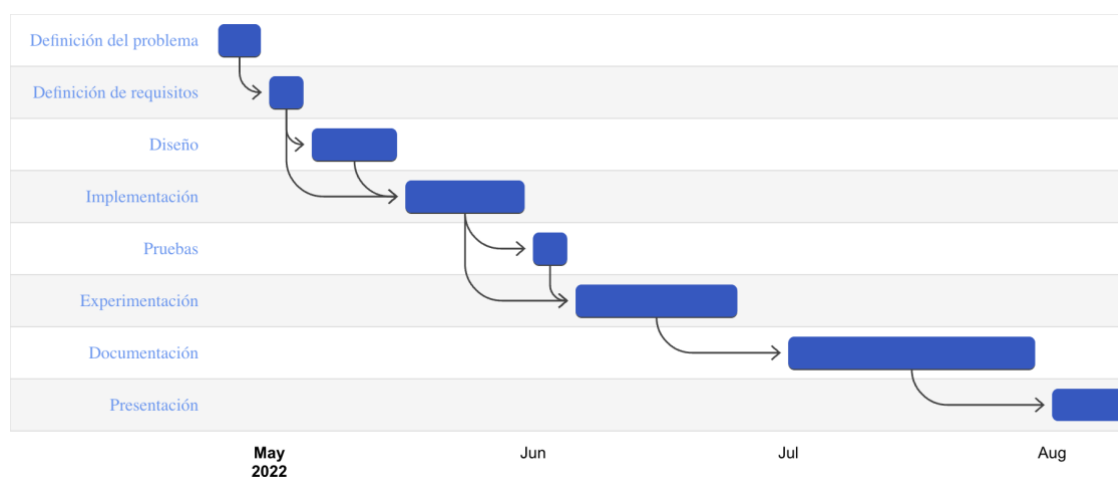


Figura 20. Diagrama de Gantt. Fuente: Elaboración propia

### 7.3. Presupuesto

A continuación, se pretende realizar un análisis de los costes económicos necesarios para el desarrollo de este trabajo.

En primer lugar, hay que describir los costes humanos del proyecto. Esto se conformará de dos personas las cuales son el desarrollador, y el jefe del proyecto, en la siguiente tabla se recoge los costes operacionales



TABLA 28. COSTE TOTAL DEL PERSONAL PARA EL DESARROLLO DEL PROYECTO

Cargo	Precio/hora	Precio Hora	Paga	Seguridad Social	Valor Seguridad Social	Coste
Desarrollador	753	15 €	11.295,0 €	30%	3.885,4 €	15.180,4 €
Jefe de proyecto	60	30 €	1.800,0 €	30%	540,0 €	2.340,0 €
Total						17.520,4

A pesar de que la paga a cada uno de los cargos pueda ser una, es importante agregar la cotización en la seguridad social, es por esto, que a pesar de que lo que ingrese una persona pueda ser una cantidad, siempre hay que agregar el IRPF necesario al total.

A continuación, se pretende declarar los costes de hardware para el desarrollo del proyecto, es importante destacar que estos montos deben de ser amortizados, ya que los equipos pueden ser útiles para otras cosas además del proyecto, se tendrá como base, una duración de 8 meses de proyecto, y una amortización del hardware de 4 años

TABLA 29. COSTE DEL HARDWARE AMORTIZADO

Descripción	Coste	Amortización	Total
Ordenador	1.500,0 €	$\frac{8 \times 1.500,0}{4 \times 12}$	307,69 €
Total			307,67 €

Para el caso de software, no se usará ninguno de coste, y, por ende, se entiende que no habrá ninguno, ya que se utilizaran herramientas open-source, que permiten su uso sin ningún tipo de coste. Por último, se presenta un desglose total del coste del proyecto

TABLA 30. COSTE TOTAL PARA EL DESARROLLO DEL PROYECTO

Descripción	Coste
Personal	17.520,4 €
Hardware	307.67 €
Software	0 €
Total	17.828,07 €

#### 7.4. Marco regulador

En cuanto a los aspectos legales, hay que en primer lugar dejar claro, que esta herramienta no garantiza la obtención de beneficio económico mediante la operación del mercado.

Por otro lado, la información mostrada no se puede considerar bajo ningún efecto como recomendaciones financieras, y tampoco se puede considera que la herramienta desarrollada funciona como asesor financiero.

En cuanto a la operativa en el mercado de valores, se deberán cumplir las regulaciones pertinentes de cada lugar de operativa. En el caso de Estados Unidos, se tiene el organismo SEC<sup>10</sup>, el cual vela que los inversores estén protegidos a la hora de realizar las inversiones, y que todas las personas tengan la misma ventaja a la hora de invertir. Es por esto, que toda la información utilizada para el desarrollo de este proyecto puede ser utilizada, y todas las personas pueden acceder a ella.

La aplicación en cuestión se distribuirá a través del sitio GitHub bajo la siguiente URL <https://github.com/alvaroahp11/TFG>

---

<sup>10</sup> Securities and Exchange Commission



## 8. CONCLUSIONES

Tras la realización del trabajo, se puede concluir que se ha podido estudiar los datos del ETF SPY, los cuales fueron procesados y aplicado técnicas de transformación que permitiesen su uso en algoritmos de aprendizaje automático.

Para las predicciones, se diseñó un sistema que permitiese evaluar distintos parámetros para un modelo de predicción, de manera que se pudiera hacer un grid-search por cada algoritmo, para en una instancia final, evaluar los mejores resultados de cada algoritmo.

En cuanto a los resultados de las predicciones, se realizó un estudio estadístico que permitiese saber si los resultados obtenidos eran significativamente iguales a los valores reales, lo que resultó en que todos los algoritmos de predicción hacían predicciones significativamente iguales a los valores reales.

Por último, se ha podido diseñar una interfaz web, que permite a los usuarios crear modelos predictivos usando algoritmos de aprendizaje automático, para la predicción de un activo en concreto.

Tras el trabajo realizado, quedan muchas mejores e ideas que se pueden hacer sobre este tema. En primer lugar, habría que estudiar cómo se comportan los modelos predictivos en un futuro, ya que quizás no sea capaz de modelar el comportamiento, y, por ende, las predicciones no van a ser tan correctas. Por otro lado, se puede estudiar que rentabilidad económica dan estos modelos, realizando distintas estrategias. Además de esto se podría estudiar otros indicadores técnicos, así como estrategias para dar más información a los datos. Por último, se podría realizar una web más compleja que permita al usuario utilizar otros indicadores técnicos, así como la posibilidad de descargar modelos.



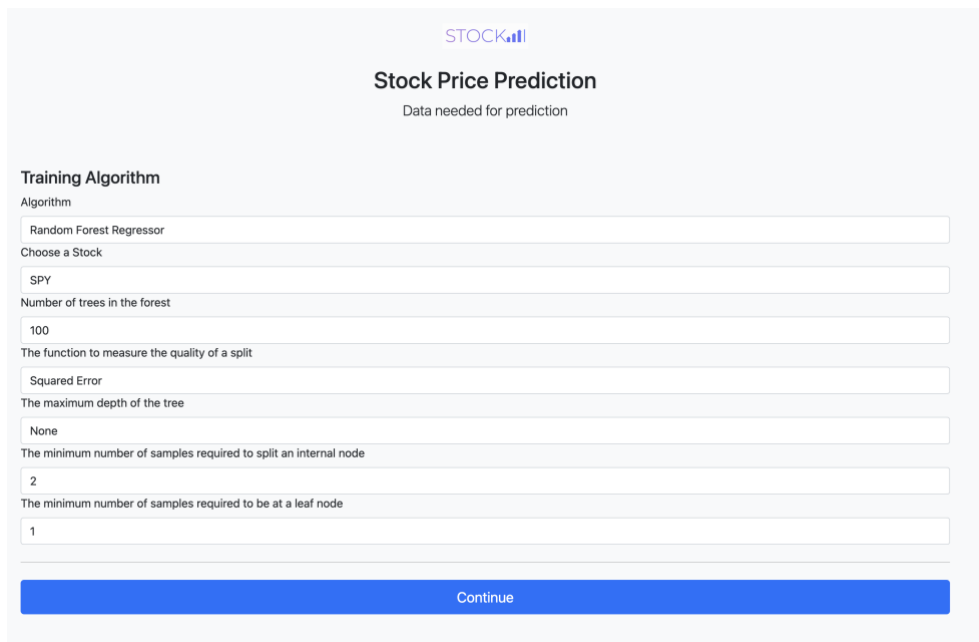
## BIBLIOGRAFÍA

- [1] A. Adebiyi, C. Ayo, M. Adebiyi, and O. S. Otokiti, "Stock price prediction using neural network with hybridized market indicators," vol. 3, pp. 1–9, 2012.
- [2] G. Bonde R. Khaled, 'Extracting the best features for predicting stock prices using machine learning', Proceedings of the 2012 International Conference on Artificial Intelligence, ICAI 2012, τ. 1, σσ. 222–229, 01 2012.
- [3] D. Polamuri, K. Srinivas, A. Mohan, 'Stock Market Prices Prediction using Random Forest and Extra Tree Regression', International Journal of Recent Technology and Engineering, τ. 8, σσ. 1224–1228, 09 2019.
- [4] I. Suryani and D. C. P. Buani, "Stock price prediction using artificial neural network integrated moving average," Journal of Physics: Conference Series, vol. 1641, (1), 2020. Available: <https://www.proquest.com/scholarly-journals/stock-price-prediction-using-artificial-neural/docview/2571033969/se-2>. DOI: <https://doi.org/10.1088/1742-6596/1641/1/012028>.
- [5] Y. Shynkevich, T. M. McGinnity, S. A. Coleman, A. Belatreche, and Y. Li, "Forecasting price movements using technical indicators: Investigating the impact of varying input window length," vol. 264, pp. 71–88, 2017, doi: <https://doi.org/10.1016/j.neucom.2016.11.095>.
- [6] P. Marquis, O. Papini, and H. Prade, Elements for a History of Artificial Intelligence, vol. 1. Cham: Springer International Publishing, 2020, pp. 1–43.
- [7] A. Turing, "Computing Machinery and Intelligence," 1950.
- [8] G. Piccinini, "The First Computational Theory of Mind and Brain: A Close Look at McCulloch and Pitts's 'Logical Calculus of Ideas Immanent in Nervous Activity,'" vol. 141, no. 2, pp. 175–215, Aug. 2004, doi: 10.1023/B:SYNT.0000043018.52445.3e.
- [9] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," vol. 65, no. 6, pp. 386–408, Nov. 1958, doi: 10.1037/h0042519.
- [10] F. Pedregosa κ.ά., 'Scikit-learn: Machine Learning in Python', Journal of Machine Learning Research, τ. 12, σσ. 2825–2830, 2011.
- [11] R. E. Barrientos Martínez et al., "Árboles de decisión como herramienta en el diagnóstico médico," Sep. 2009
- [12] J. J. Murphy, Análisis técnico de los mercados financieros. Barcelona, 2000.
- [13] I. H. Witten, E. Frank, M. A. Hall, and C. Pal, Data Mining. San Francisco: Elsevier Science & Technology, 2016, pp. xxxiii–xxxiii.
- [14] Martín Abadi κ.ά., 'TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems'. 2015.
- [15] R. Araoussi, "yfinance." [www.github.com/rararoussi/yfinance](https://www.github.com/rararoussi/yfinance).

- [16] Yahoo, "Yahoo Finance" <https://finance.yahoo.com/>.
- [17] "Librería TA-Lib (Technical Analysis Library)." [www.ta-lib.org](http://www.ta-lib.org)
- [18] J. W. Wilder jr, New concepts in technical trading systems. Greensboro/N. C: Trend Research, 1978.
- [19] "John Bollinger's Official Bollinger Band Website." <https://www.bollingerbands.com/>.
- [20] D. Rey and M. Neuhäuser, Wilcoxon-Signed-Rank Test. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1658–1659.

## ANEXO A. MANUAL DE USUARIO

La primera interfaz que tiene el usuario es la que se muestra en la figura 16, en esta, se puede observar como principal algoritmo el Random Forest Regressor. En esta se pueden observar los parámetros para este algoritmo, y a su vez, la acción para la cual se hace la predicción. En este caso, se muestra un ejemplo con el SPY.



The screenshot displays a web interface for "Stock Price Prediction". At the top, there is a logo "STOCK" with a bar chart icon. Below the logo, the title "Stock Price Prediction" is centered, followed by the subtitle "Data needed for prediction". The main section is titled "Training Algorithm" and contains a form with the following fields:

- Algorithm: Random Forest Regressor
- Choose a Stock: SPY
- Number of trees in the forest: 100
- The function to measure the quality of a split: Squared Error
- The maximum depth of the tree: None
- The minimum number of samples required to split an internal node: 2
- The minimum number of samples required to be at a leaf node: 1

At the bottom of the form, there is a blue button labeled "Continue".

Figura 21. Parámetros de predicción para el algoritmo Random Forest Regressor

En caso de que el usuario decida escoger otro algoritmo, se ajustará la página para los parámetros de este, para este caso, al escoger SVR, se puede observar los parámetros a escoger para este algoritmo.



STOCK

## Stock Price Prediction

Data needed for prediction

### Training Algorithm

Algorithm

Support Vector Regressor

Choose a Stock

SPY

Specifies the kernel type to be used in the algorithm

Rbf

Degree of the polynomial kernel function only works for poly

3

Kernel coefficient for rbf, poly and sigmoid

Scale

Continue

Figura 22. Parámetros de predicción para el algoritmo Support Vector Regressor

Por último, se muestran las opciones de parámetros, cuando el usuario decide seleccionar Decision Tree Regressor

STOCK

## Stock Price Prediction

Data needed for prediction

### Training Algorithm

Algorithm

Decision Tree Regressor

Choose a Stock

TSLA

The strategy used to choose the split at each node

Best

The function to measure the quality of a split

Squared Error

The maximum depth of the tree

None

The minimum number of samples required to split an internal node

2

The minimum number of samples required to be at a leaf node

1

Continue

Figura 23. Parámetros de predicción para el algoritmo Decision Tree Regressor

En la siguiente figura se muestra los resultados de una predicción, cuando el usuario decide seleccionar continuar. En esta imagen se puede mostrar, como se grafica el período del precio real, de fechas previas, y a continuación, se muestra la predicción que realiza el algoritmo



Figura 24. Ejemplo de predicción realizada por la aplicación

## **ANEXO B. SUMMARY**

### **1. Introduction**

#### **1.1 Motivation**

The main motivation to carry out this project comes from the financial knowledge acquired in recent years and added to what has been learned in computer science, specifically in the area of artificial intelligence, the idea of making a mixture arises. In addition to this, opportunities have arisen to develop predictive models, which at first glance seemed to behave correctly, which arouses more attention to deepen this area.

On the other hand, the main problem that can be posed to develop this work is the difficulty of predicting random behavior, however, the search for new ideas and testing different algorithms can become a progress for market prediction.

In addition to this, creating a prediction tool can be very useful to contrast the results of personal financial analysis. So that it becomes a helpful tool for anyone who wants to operate in the market.

Finally, analyzing different algorithms can be useful to understand which ones are best suited in this environment.

#### **1.2 Description of the problem**

The financial market is the place where securities and derivatives are traded at a cost. Those that we can find here can be stocks, bonds, commodities, among others. From this buying and selling it is possible to get some profitability, so it is interesting to find a way to predict their behavior to get the most profit.

The behavior of this asset is usually random, and it is difficult to make predictions, however, with the passage of time have emerged methods such as technical analysis, which seeks to identify statistical trends collected from trading activity, such as price and volume.

Although at first glance making a profit by buying and selling seems simple, the reality is that it is not, since the value of an asset is defined by many factors that are difficult to control and leads to constant changes in their prices.

Based on this, numerous studies have been carried out in an attempt to model the behavior of assets, mostly using machine learning models, however, the results obtained tend to lose efficiency over time.

#### **1.3 Objectives**

The main objective of this work is to carry out a study of machine learning techniques focused on the field of the stock market, specifically, for the prediction of the closing price.

To do this, it is intended first to conduct studies of the ETF<sup>11</sup> SPY, in this way, you must analyze the data, and make design decisions to transform the data, so that is a structure that allows to make use of machine learning models. The decision to conduct studies based on the SPY, is because it is an asset which aims to group the 500 largest companies in the United States, in this way, a general study of the average behavior that may be occurring in the stock market is performed.

Once the adjustments have been made to the data, we intend to test different machine learning techniques in order to obtain the algorithms that give the best results when it comes to predicting the closing price.

On the other hand, it is intended to perform a hypothesis test, to observe how useful are the predictions made by the algorithm.

Finally, it is intended to create a web interface that allows any user without experience in artificial intelligence, make predictions of the price of a list of assets, to do this, different algorithms will be presented with various parameters, so that the user, place those you want to use, and the program, performs the prediction of the chosen action.

The objectives of the research work are listed below.

1. Study of the data, and data transformation, in order to make use of machine learning models to predict the daily close of the ETF SPY.
2. Design a system to be able to test different machine learning algorithms.
3. Perform a hypothesis test, to know if the results obtained are good predictions or not.
4. Web interface design, which allows a user to make predictions on the stock market.

## **2. State of the art**

In the prediction of the price of an asset in the stock market, numerous techniques and various methods can be applied to achieve this goal, since they can be made from buying and selling predictions, to the exact price.

Price predictions are often difficult, since it is about getting the exact price of an asset, added to that, there are countless factors that change a price, however, there are works that try to do this, such as the article published in 2012 by Adebisi Ayodele, which attempts to combine technical and fundamental analysis, to make predictions using neural networks, specifically, the multilayer perceptron [1]. The main technical indicators were open, high, low, volume, close. In this work we get some configurations that are quite good, as the case of only the technical indicators data, which yield a model with a root mean square error (RMSE<sup>12</sup>) of 0.0729, the negative part of this work is in the models that mix technical and fundamental analysis, since one measure is in years and the other in days, which is confusing when modeling. On the other hand, it is important to highlight the modeling of the data, which is done as a time series<sup>13</sup>, and

---

<sup>11</sup> exchange-traded fund, is a pooled investment security that works much like a mutual fund.

<sup>12</sup> Function that measures the amount of error between two data sets.

<sup>13</sup> It consists of data which are measured at a specific time instant, and also have a chronological order.

this makes sense, since the data of one day depends to some extent on the data of the previous days.

On the other hand, researchers Rasheed and Ganesh from the University of Georgia, published in the same year, a comparison of different attributes and machine learning algorithms, for the prediction of different shares of the American market [2], in this work results, that the best algorithm is the SVR, it is said that neural networks behave well, however, the results obtained, were not satisfactory, on the other hand, the attributes that worked best, were those that had the volume, and, another set of data that behaved well was a combination of indexes, and information of the company to predict.

Another paper published by Subba, Kudipudi and Krishna in 2019 [3], where they seek to buy a line regression, with regression trees, for stock market price prediction, it turns out that trees have a great advantage, and that the results obtained, tend to be good.

In 2020, the researchers Suryani and Buani, make use of neural networks for the prediction of the ANTM stock [4], as input data they use only the closing price of the stock, and perform transformations on the data using the moving average<sup>14</sup>. In this work was obtained for the best model an RMSE of 0.004 and an average error of 0.0121, these results are quite good, in addition to studying the results of a model without applying the moving average, and is statistically significant better model using the moving average.

To approach the problem from the data point of view, we can see the structure shown in [5], where they use technical indicators to model the problem, as for those that use the closing of the asset price are the moving average, the RSI, the exponential moving average. On the other hand, there are other indicators dependent on maximum, minimum and volume, the ones used in this area were the mean of the true range, the Williams indicator %R, or the Stochastic Oscillator %K. We can also find other works.

### **3. Theoretical Framework**

#### **3.1 Artificial Intelligence**

Throughout history, human beings have tried to replicate their intelligence, this can be seen by historical facts such as the syllogisms created by Aristotle, which seek to describe a part of the functioning of the human mind, or the first self-controlled machine created by Ctesibius in 250 BC [6]. However, all these creations were quite far from the behavior of a human, and that is why in 1950 the article "Computing Machinery and Intelligence" [7], written by Alan Turing, was published, which aims to evaluate the behavior of a machine and its resemblance to that of a human, this is called "Turing's Test".

Artificial intelligence is therefore an area of computer science, which seeks to develop and replicate the intelligence of human beings through a computer, this behavior tries to mimic mainly human reasoning.

---

<sup>14</sup> It is a simple technical analysis tool that smoothes the price data by creating a constantly updated price average.

Although emulating human behavior can be difficult, there have been a number of advances in recent years, due to new knowledge and improvements in the computational speed of computers. Currently, the goal is to develop widely applicable AI systems that interact safely with humans and the physical world. To this end, more and more different concepts and approaches are coming together: machine learning, symbolic reasoning, cognitive science, developmental psychology, robot control engineering, and human-machine interactions, among others.

In the area of artificial intelligence there are many techniques to solve problems, however, one of the most used is machine learning.

Machine learning is the science that aims to develop various techniques that make computers learn. To do this, the solution to the problem is not programmed implicitly, but various techniques are applied to make the machines learn

As for the algorithms, we must take into account that its operation is based mainly on mathematical and statistical models, it is, therefore, that this area of computer science is multidisciplinary, and more and more knowledge is being applied, such as neuroscience, logic, among others.

In machine learning there are three branches of learning, one of them is supervised learning, which is a technique that seeks to learn and generalize from a series of training data.

These training data have the characteristic that they are pairs, with one part corresponding to the information of the instance, and another for the expected output of that input. This expected output depending on the problem can be a numerical value (regression problem) or a label (classification problem).

The main goal of supervised learning is the generalization of a function capable of generating a valid output from an input, this should be done once you have seen a number of examples.

### **3.2 Stock Market Environment**

The stock market generally refers to a series of exchanges and other venues where shares of public companies are bought and sold. These financial activities are conducted through formal institutionalized exchanges (physical or electronic) and through over-the-counter markets that operate under a defined set of regulations.

Although the terms "stock market" and "exchange" are often used interchangeably, the latter term is really a subset of the former. Stock market traders buy or sell shares on one or more of the exchanges that are part of the global stock market.

When trading the stock market, there are several techniques that allow users to have an intuition of the projection that will have the market, these are divided into fundamental analysis and technical analysis.

As for the fundamental analysis, it is a technique that aims to determine the intrinsic value, to do this, a combined study of the financial statements of the company, external influences, events, and industry trends is made.

From the fundamental analysis you can take buy and sell actions depending on the value obtained from the analysis, in case of a low value compared to the actual price, it is advisable to sell, and otherwise buy.

Two types of analysis are possible:

- Top-down: Focuses on the big picture, or how the global economy and macroeconomic factors drive markets and ultimately stock prices. They also look at the performance of sectors or industries. These investors believe that if the sector is doing well, stocks in those industries are likely to do well.
- Bottom-up: An investment approach that focuses on the analysis of individual stocks and downplays macroeconomic and market cycles. In other words, bottom-up investing typically focuses on the fundamentals of an individual company, such as revenues or earnings, as opposed to the industry or the economy as a whole. The bottom-up investment approach assumes that individual companies can do well even in an underperforming sector, at least in relative terms.

Technical analysis is a method that tries to predict the trend, direction and price of the market. To do this it makes use of statistical and mathematical models, which are able to generate buy and sell signals.

This approach is based on three premises [12]:

- "Market movements discount everything". This is the main basis of technical analysis, and refers to the fact that the market is affected by factors such as fundamental, political, psychological, among others, and this is reflected in the price of the asset. That is why the technical analysis, only focuses on the graphs, to predict market trends, since, by external factors, will be reflected in the graph, and thus the analyst will know how to act.
- "Prices move by trends". This premise refers to the fact that stock market prices act by trends, as can be seen in Figure 6, and is one of the main reasons why the chart should be observed, since it allows to observe trends and thus, to know the direction of the market. It is important to note that the trend of a market is not always the same, and that there are changes in it, which technical analysis should be able to resolve.
- "History repeats itself. As for history, it is known that technical analysis relies heavily on human psychology, since, analyzing different graphs of the past, it is observed how bullish or bearish trends have worked in the past, and that is why it is assumed that they will also work in the future. Without going too far, it is believed that to understand the future you have to study the past, or simply that the future is a repetition of the past.

When performing a technical analysis it is not enough to look at the chart and say that a bullish or bearish trend is forming, but the analyst relies on numerous indicators that allow him to know the trend that the market will follow.

#### **4. Development**

For the development of this work, there are numerous tools that enable its development, and over time, technologies that facilitate machine learning processes have emerged.

A well-known tool in the field of machine learning algorithms is Weka [13], which allows you to create numerous machine learning models, in this same tool, you can manage the data, try different configurations and download models. All this works in Java programming language. The main problem to make use of this tool is the impossibility of making a comparison between all the models, because to do this, you must perform a manual process of testing each and every one, for this reason, this tool is discarded for use.

From the idea of wanting to test numerous models, in the best way, results the idea of performing the process with a programming language, among the most famous are Python, Java, C++ and Javascript. However, in recent years, the main leading language in artificial intelligence process is Python, this is because this language has numerous built-in libraries that facilitate this process. For this reason, it is intended to carry out the implementations in this language.

As for the libraries that are available for predictive models, there is scikit-learn [10], which has numerous regression and classification algorithms, as well as tools that allow data analysis. On the other hand, there is TensorFlow [14], a library used exclusively to develop neural networks. Some others that can be mentioned are keras, pyTorch, among others. Because in the development of this work we intend to test some of the best known algorithms, it has been decided to use scikit-learn, due to its versatility and breadth of algorithms.

Finally, we have to talk about the execution center, since we are talking about long executions, and the need to run at the highest possible speed, the option of using your own computer is discarded, therefore, we intend to make use of Google Colab, Google tool that allows you to write Jupyter notebooks, and run them on a machine in the cloud.

#### **4.1 Data extraction**

After the previous studies, we must first obtain the data with which we want to work, and in addition to this, we must perform a transformation of these data. In the financial world, the data of an asset can be worked in time series of different magnitudes, from seconds, weeks, or even months. Therefore, in the study to be carried out, we intend to use daily series, so that each row of data represents a specific day, therefore, the objective of the prediction, i.e. the closing price that the asset is going to have. The closing price refers to the monetary value that the asset will have at the time of market closing.

As mentioned above, we intend to perform the studies based on the ETF SPY, for the extraction of the data we will use the Python library yfinance [15][16]. This is a library that uses the Yahoo api<sup>15</sup>, so that it allows access to different time series, in the case of this work, it is intended to collect a daily time series, since the ETF started trading, until July 15, 2022.

The total number of instances collected is 7419, which corresponds to the total number of days that the ETF has been traded in the study period. It is important to remember that the stock market does not open on weekends or holidays, and therefore, there are date breaks in the data, however, for the case study, this is not a problem, since, even if there is a break, for the model it will mean the next day.

---

<sup>15</sup> Application Programming Interface



As for the data returned by the library, its meaning is described below:

- Open: Price at which the asset opens, once it enters the opening hours of the market. It could be understood that this value is the close of the last day, however, before the opening there are some trading hours that are not available to the public, and therefore it is not considered as the opening price.
- High: Maximum price that the asset reaches during trading hours.
- Low: Minimum price at which the asset reaches during trading hours.
- Close: Price at which the market closes
- Adj Close: In English known as adjusted close, or in Spanish, cierre ajustado. The adjusted close is the closing price after adjustments for all applicable splits and dividend distributions.
- Volume: Refers to the operational volume. That is, the amount of purchases and sales that occur during the time period.

Once the data have been obtained, some transformations must be carried out in order to be able to make use of them. It is not enough to have the data to be able to make predictions.

As we have seen in previous works, and having defined the types of analysis that exist to predict the stock market, we have the technical and fundamental analysis. For the development of this work we intend to use only the technical analysis, and specifically, we intend to apply various technical indicators, which allow us to give more information from the data. It is important to emphasize that only close data will be used, and the indicators will be calculated with this variable. So that the data on the high, low, volume and open, will not be used.

The following is a description of the indicators to be used. It is important to note that all the indicators are calculated with the Python library called Ta-Lib [17].

As described in the state of the art, the data to be treated can be represented as a time series problem. Since the data has a chronological order, and it could be said that the data of one day depends to some extent on the value of the previous day.

From this, the prediction system to be mounted will try to find the closing value one day, based on data from previous days, i.e.

$$\text{Close}[t-n], \text{Close}[t-n-1], \text{Close}[t-n-2], \dots, \text{Close}[t-1], \text{Close}[t] \Rightarrow \text{Close}[t+1]$$

However, the system not only takes into account the close, but also intends to introduce certain technical indicators, and in addition to this, it must be defined, how many previous days will be taken into account.

To work the problem, it is proposed to use a 5-day lag, so that the indicators of 5 previous days are put together with the close to predict the next day. Therefore, each day will be represented with the following information:

- MA200
- MA50

- EMA200
- EMA50
- RSI
- UPBAND
- MIDBAND
- LOWBAND
- CLOSE

In this way, the input of the model will be a set of 45 data, constituted by the last 5 days, where for each day it is made up of the data mentioned above.

## 4.2 Experimentation

The methodology to evaluate the models results in a fairly simple process that consists first of dividing the data set into three parts, one for training, the next to evaluate the model, which is called test set, and finally, a validation set, which is responsible for making the comparison between the best models of each machine learning algorithm.

To generate the models, it is intended to perform a grid search among a series of parameters, and in this way test the possible configurations, with this, it could be known among all the combinations how good is the model.

To know how good the model is, two methods will be used, the first one will be to calculate the root mean square error of the predictions, in this way, you could know how much error is made, on the other hand, a hypothesis test will be done, to know, if the best model generated, make predictions as good as the real result.

## 5. GUI

For the architecture of the system, a three-layer structure is proposed, known as the controller view model, as shown in Figure 15. In this way, the system is divided in three components.

- Model: Part of the system, responsible for representing the logic of the system, in most cases, is the system responsible for storing the information, in our case, will be the part of the system responsible for storing and creating the prediction model.
- View: Part of the system in charge of representing the information of the model to the user, through the graphic interface, the view has access to the model, however, it cannot make modifications on it.
- Controller: This part is in charge of reacting to the user's requests, in this way, the changes in the model are generated, and in this case, the requests are sent to make the predictions.

## 6. Result

First of all, it is necessary to talk about the results obtained for the test set of the algorithms used. The SVR presents the smallest error with a value of 2.3036, on the other hand, the Random Forest in second place with an error of 2.3212 and in last positions the multilayer perceptron and Decision Tree Regressor with an error of 2.7449 and 2.7485 respectively.

It can be said that the results obtained at first glance are low, and that between the models do not present a noticeable difference, however, this does not give us information on how the models will behave in action, which is why a set of validation is needed to know between these models, which one is the best.

Table 25 shows the results obtained for the validation set, where once again the SVR algorithm presents the lowest error, with an average value of 2.4255 over 10 runs. In second position is the Random Forest with a value of 2.5159, followed by the Decision Tree and the multilayer Perceptron, whose errors were 2.8945 and 3.4568 respectively.

After performing these tests, it can be said that the results obtained seem relatively good, because the change in error between data sets was not so high. However, the last step is to perform a statistical contrast between the predicted values and the real ones, in order to know the veracity of the data.

Table 26 shows the results obtained in the Wilcoxon test [20], where it can be seen that all models are not able to reject the basic hypothesis and, therefore, the predictions are statistically equal.

Given these results, it can be said that the algorithms used for the predictions with the configurations of the best models have good behavior to model the stock market. However, it is important to emphasize that the SVR was the one that had the best results in all the tests performed, but on the other hand, it is a rather expensive and slow algorithm to train, due to the computational complexity<sup>16</sup> that it can achieve. On the other hand, in terms of complexity, it can be said that the Random Forest can be very expensive, because a model can have 700 trees, and that the algorithm must train each of these separately, which converges in quite high execution times.

As for the use of these models in real life, it would remain for further studies to study the profitability of the predictions in the long term to be able to profit from the stock market. It is important to emphasize that these predictive models created are good for the short term, and that it is very difficult to know if in the future they will be assertive, since the market has such a random behavior, and works by "seasons", that perhaps the behavior is very good during the first days of training, but in the future may not be optimal.

As for the software created, we can say that all the requirements are met and that it works in the right way. Despite being so simple, it fulfills the objective of being a support for an investor, so that it allows him to know the results that a predictive model can have and counterbalance it with personal investment decisions.

## **7. Conclusion**

After the completion of the work, it can be concluded that it has been possible to study the ETF SPY data, which were processed and applied transformation techniques that allow their use in machine learning algorithms.

---

<sup>16</sup> Resources required to solve the problem

For the predictions, a system was designed to evaluate different parameters for a prediction model, so that a grid-search could be made for each algorithm, in order to evaluate the best results of each algorithm in a final instance.

As for the results of the predictions, a statistical study was carried out to find out if the results obtained were significantly equal to the real values, which resulted in all the prediction algorithms making predictions significantly equal to the real values.

Finally, it has been possible to design a web interface that allows users to create predictive models using machine learning algorithms for the prediction of a specific asset.

After the work done, there are many better ideas that can be done on this topic. First of all, it would be necessary to study how predictive models behave in the future, since they may not be able to model the behavior, and therefore the predictions will not be as correct. On the other hand, it is possible to study the economic profitability of these models, carrying out different strategies. In addition to this, other technical indicators could be studied, as well as strategies to give more information to the data. Finally, a more complex website could be created that allows the user to use other technical indicators, as well as the possibility of downloading models.