

Introducción

- Estado del arte
- Objetivo

Herramientas y tecnologías empleadas

- Herramientas
- librerías y tecnologías

LSTM

- Introducción
- Primeras pruebas
 1. Modelo simple
 2. Método de la ventana
- Mejorando los datos de entrada
- Usando indicadores técnicos

Referencias

Estado del arte

< texto >

Objetivo:

El objetivo de este trabajo es poder analizar el uso de diferentes técnicas de ML¹ para el problema de predecir el valor de una acción en la bolsa. Para ello se experimentara con tipo de red que son las LSTM, las cuales han dado buenos resultados en la resolución de este tipo de problema. Por ello, se probaran diferentes tipos de configuraciones, enfoques, análisis y modificación de los datos de entrada, etc, con el objetivo de mejorar la exactitud de las predicciones y poder ver el potencial de este tipos de redes.

Herramientas y tecnologías empleadas

Como lenguaje de programación para el desarrollo de las distintas pruebas se ha optado por el lenguaje Python. El código desarrollado con este lenguaje es sencillo, elegante y fácil de leer. Esto permite que la estructura sea muy parecida al lenguaje humano y al lenguaje matemático, permitiendo que sea idóneo como un pseudocódigo.

Python se caracteriza por ser un lenguaje puente entre el mundo empresarial y científico. Esto resulta idóneo a la hora de creación de códigos entendibles de rápido aprendizaje como los que son necesarios en proyectos de ML.

Herramientas:

Como herramienta principal para el desarrollo de las pruebas y el entrenamiento de las distintas redes LSTM se ha usado la herramienta proporcionada por Google llamada Google Colab o también llamado **Colaboratory**. Esta herramienta se basa en la tecnología de Jupyter Notebook², la cual permite ejecutar código estructurado basado en entradas y salidas usando celdas. De esta manera se pueden crear documentos que contengan además del código a ejecutar, textos, gráficos, etc, que facilitan la creación de documentos científicos y educativos.

Con Google Colab se pueden ejecutar estos documentos directamente en la nube en servidores optimizados y dedicados a esta tarea. Esto facilita a la hora de diseñar y ejecutar diferentes modelos de ML no depender del hardware que se posee. De esta manera se puede agilizar todo el proceso, sin tener que preocuparnos por si se tienen los recursos suficientes, pudiendo centrarse uno en el desarrollo y las pruebas.

Dentro de esta herramienta de Google podemos activar el uso de una GPU, podemos compartir código fácilmente a otras personas y que puedan también modificar este, ya viene instalada con las librerías más usadas en datascience y además permite instalar las librerías que nos hagan falta y podemos enlazarla con nuestra cuenta de Google Drive, lo que permite leer documentos como CSV fácilmente, pero también se pueden escribir y guardar los documentos, imágenes, etc que nos haga falta.

Librerías y tecnologías:

- **TA-Lib:**
- **Jupyter Notebook:** es una aplicación web de código abierto que permite crear y compartir documentos que contienen código ejecutable, texto descriptivo, imágenes y fórmula matemáticas. Esta pensada para el desarrollo científico permitiendo crear documentos que son fáciles de entender, dando mayor facilidad para un desarrollo conjunto y compartir de documentos a un público mayor.
- **Keras:**

¹ ML: Machine Learning

² Jupyter Notebook: https://es.wikipedia.org/wiki/Proyecto_Jupyter

- **Panda:**
- **Numpy**
- **Matplotlib:**

LSTM

LSTM son las siglas en ingles de **L**ong **S**hort-**T**erm **M**emory. Este tipo de red es una evolución de las RNN (redes neuronales recurrentes), las cuales se usan para encontrar patrones en un conjunto de datos. El problema con las RNN es que no guardan relación entre los datos, no pueden hacer uso del contexto para poder predecir el siguiente valor. Esto se solventa con las LSTM, las cuales pueden hacer uso de una memoria, de tal forma que el valor en el tiempo t , se ve afectado por el valor $t-1$. Estas redes nacen para el análisis lingüístico, donde se necesita conocer las letras anteriores para predecir las siguientes. Es por esto que estas redes funcionan muy bien en la predicción de los valores de bolsa, ya que permiten hacer uso de los valores anteriores para poder predecir los futuros.

Primeras pruebas:

Para la primera prueba se ha optado por un dataset simple, para poder ver que resultado da y como se puede ir mejorando usando otras técnicas o un dataset mas completo.

Datos de entrada:

Se ha escogido los valores del petroleo Brent. En esta primera entrada, tenemos los precios precios de cierre de todos los días desde 1987-05-20 hasta el 2020-04-14.

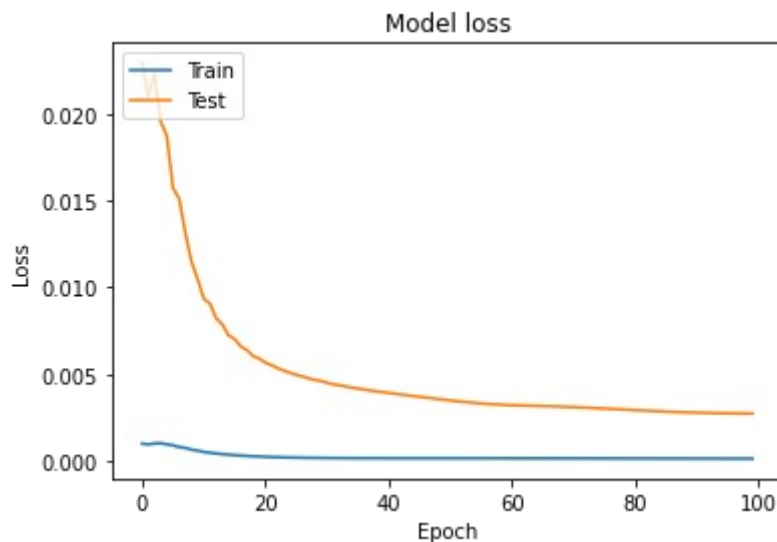
Primeras pruebas

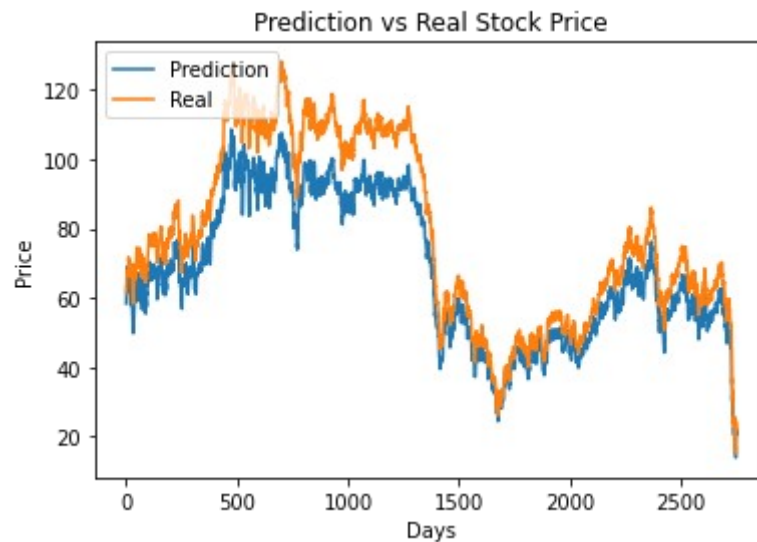
1. Modelo simple

El primer modelo es un modelo simple, usando un único valor para intentar predecir el valor del día siguiente. La idea de hacer este primer experimento con un modelo simple es poder ver el potencial de los modelos LSTM, donde con una red simple y datos simples se puede obtener un buen resultado.

A continuación tenemos el resultado de como evoluciona el entrenamiento frente al test:

ECM = 133.06





Aquí podemos ver una grafica del precio de los datos de validación, donde se pueden ver los precios reales frente a la predicción:

A continuación calculamos la predicción que tiene la red para la fecha 2020-06-12, teniendo en cuenta que los valores usados para predecirlo nunca los ha visto ni en el entrenamiento ni en la validación. El valor que obtenemos es:

Predicción: 42.37542

Real: 38.54

Diferencia: 3,83542

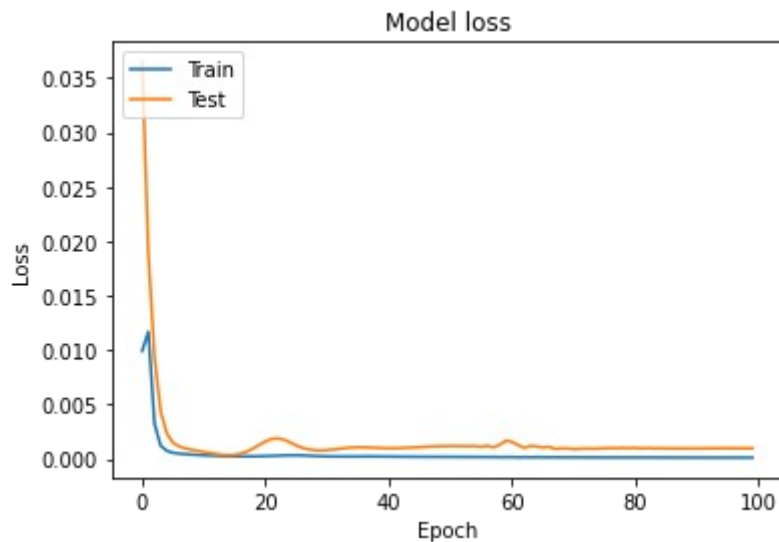
Conclusión:

Para ser un modelo simple da un primer buen resultado, dejándonos ver el potencial de este tipo de redes para predecir valores de bolsa.

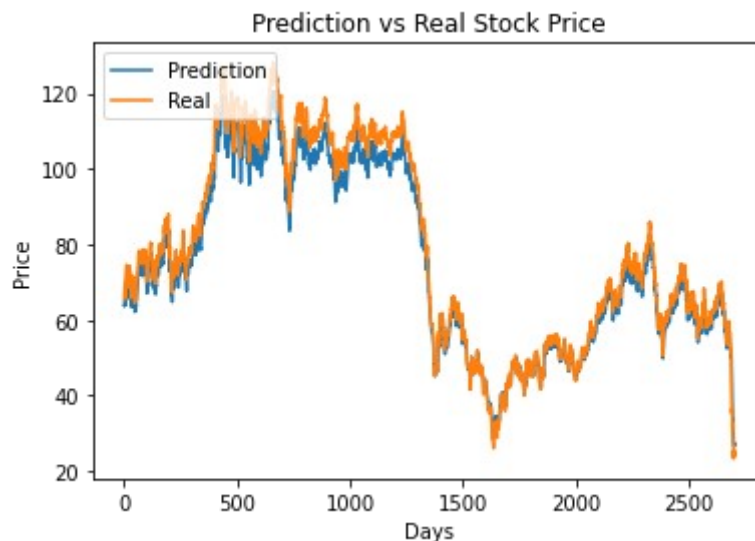
2. Método de la ventana:

Para esta segunda prueba se ha procedido a usar el método de la ventana, que consiste en usar más de un valor de entrada para intentar predecir un valor futuro. Para ello, se ha usado los 40 días anteriores para predecir un valor. El resultado sería:

ECM = 17.47



La gráfica de los datos reales vs predicción:



A continuación calculamos la predicción que tiene la red para la fecha 2020-06-12, teniendo en cuenta que los valores usados para predecirlo nunca los ha visto ni en el entrenamiento ni en la validación. El valor que obtenemos es:

Predicción: 39.169594

Real: 38.54

Diferencia: 0.629594

Conclusión:

Vemos que el modelo usando el método de la ventana ha mejorado considerablemente del modelo simple. Tanto el error medio cuadrático (ECM) es menor como el valor predicho para la fecha 2020-06-12.

Mejorando los datos de entrada:

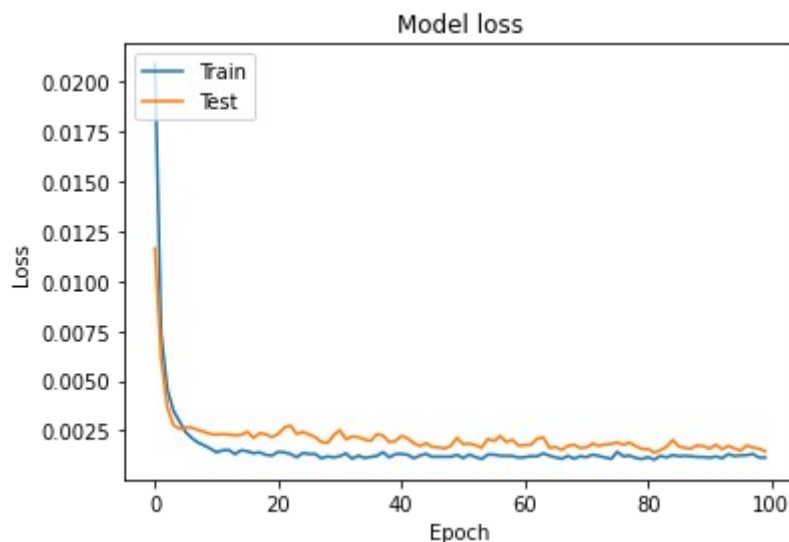
Ahora para mejorar los resultados obtenidos, vamos a proceder a mejorar la calidad de los datos de entrada. Para ello vamos a usar dataset distinto, este es el del valor MSFT, que es el valor de la empresa Microsoft en el mercado NASDAQ³. Este dataset cuenta con fecha, valor de apertura, mayor valor del día, menor valor del día, precio de cierre y volumen de las operaciones.

timestamp	open	high	low	close	volume
2019-01-02	99.5500	101.7500	98.9400	101.1200	35329345
2018-12-31	101.2900	102.4000	100.4400	101.5700	33173765
2018-12-28	102.0900	102.4100	99.5200	100.3900	38169312
2018-12-27	99.3000	101.1900	96.4000	101.1800	49498509
2018-12-26	95.1400	100.6900	93.9600	100.5600	51634793
2018-12-24	97.6800	97.9700	93.9800	94.1300	43935192

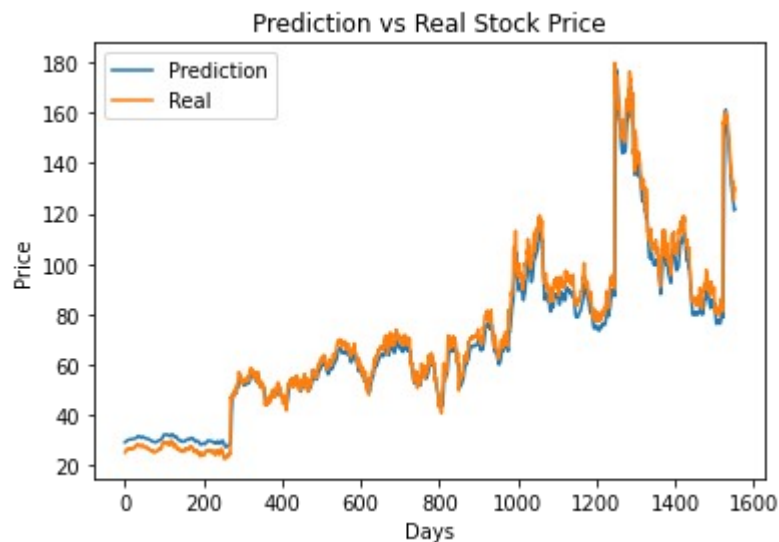
Primer experimento

En este primer experimento, no se van a tratar los datos de entrada, simplemente se van a usar como se indica en el apartado anterior. A continuación se puede apreciar como ha ido el entrenamiento de la red para los datos antes mencionados.

ECM = 35.69



Se puede observar como ha sido la predicción frente al valor real en la siguiente grafica.



3 NASDAQ: <https://es.wikipedia.org/wiki/NASDAQ-100>

Ahora procedemos a realizar la misma prueba que la anteriores, intentar predecir el precio del valor para la fecha 2020-06-22 usando valores que nunca ha visto la red, solo los necesarios para cumplir con la configuración de la red.

Predicción: 192.58609

Real: 200.57

Diferencia: 7.98391

Usando indicadores técnicos

Observando los resultados obtenidos hasta el momento, se puede ver que todavía la red no consigue obtener el patrón con los datos de entrada, lo que nos da una alta tasa de error. Para intentar mejorar esto, se van a calcular varios indicadores técnicos sobre los datos de entrada, para así poder facilitar el aprendizaje de la red y mejorar sus resultados.

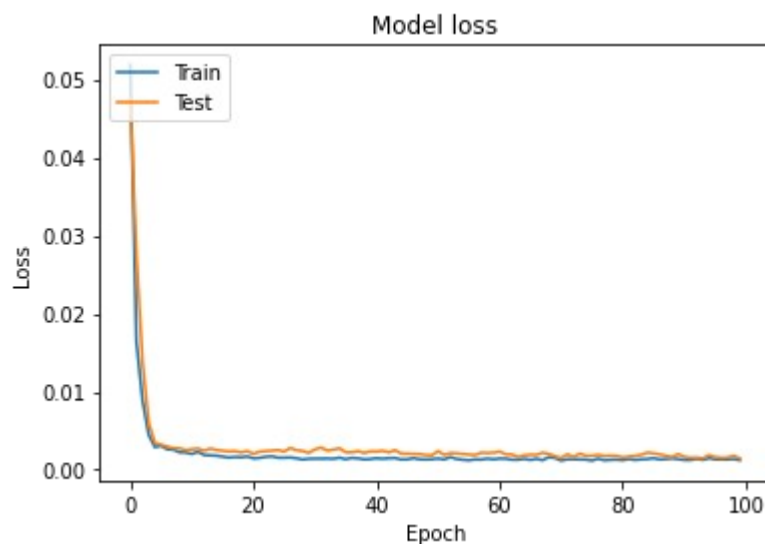
Primero se probara con los indicadores más populares, los cuales son MACD, RSI, Soportes y resistencias, Bandas de Bollinger, Medias Móviles y Estocástica.

RSI:

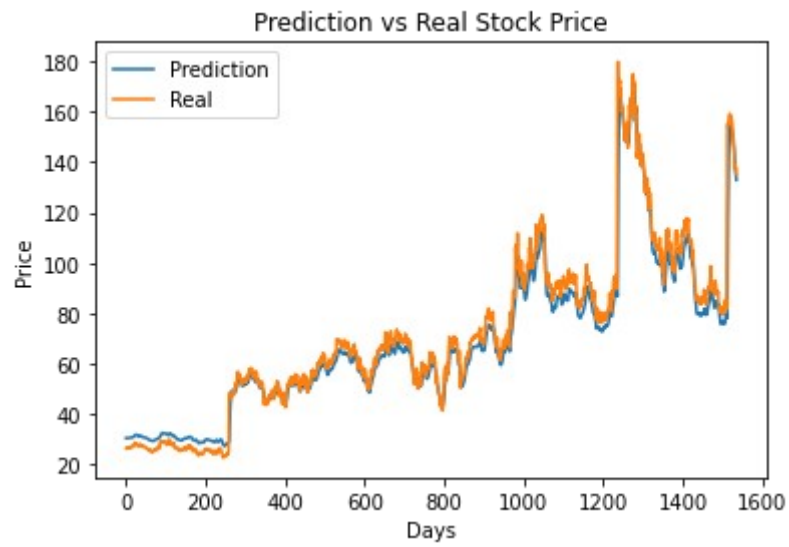
El RSI (Relative Stength Index) es uno de los indicadores más utilizados en todos los mercados financieros. Su objetivo es definir con la mejor precisión posible la fuerza de una tendencia en curso mediante la utilización de los volúmenes de negociación. De esta forma, se puede saber si una tendencia esta empezando o si, por el contrario, está finalizando.

Para ver como cambia el entramiento se ha ejecutado una nueva iteraccion sin usar el campo rsi, lo que da como resultado:

ECM=36,76

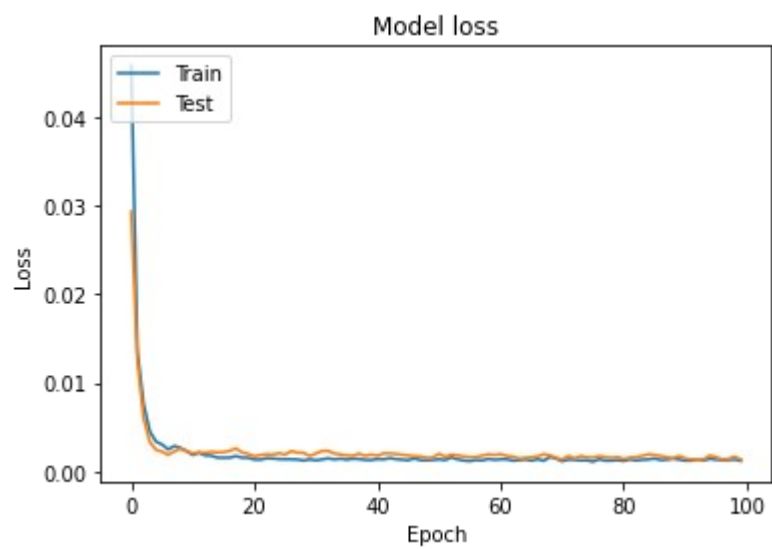


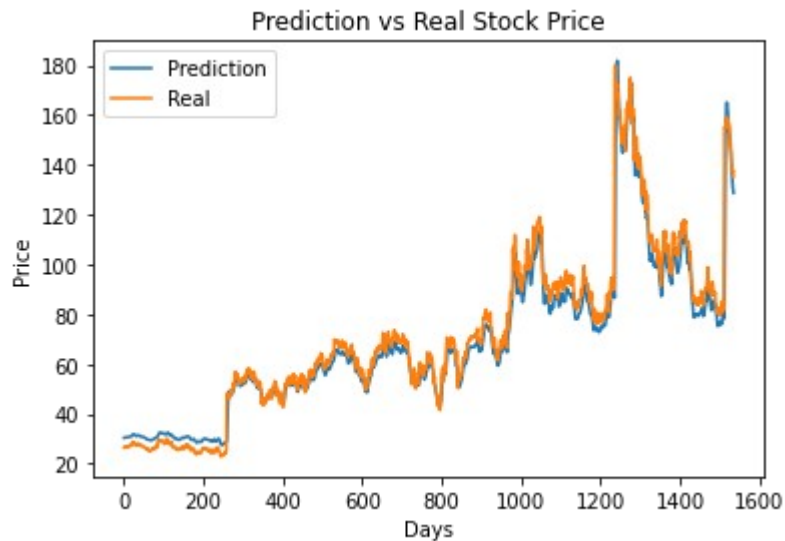
La gráfica de la predicción vs el real:



Ahora, haciendo uso del campo calculado del RSI, tenemos como resultado:

ECM= 35.46





Como podemos ver, se ha conseguido una mejora en el ECM de 1,3.

Vamos a realizar la misma prueba que en los apartados anteriores, donde se va a intentar predecir el valor para la fecha 2020-06-22 usando valores que la red nunca ha visto, y solo los necesarios para poder cumplir con la configuración de la red. El resultado:

Predicción: 193.474

Real: 200.57

Diferencia: 7,096

MACD:

Soportes y resistencias:

Bandas de Bollinger:

Medias Móviles

Estocástica:

Bibliografía:

Jupyter Notebook: https://es.wikipedia.org/wiki/Proyecto_Jupyter
<https://jupyter.org/>

Error cuadrático medio (ECM): https://es.wikipedia.org/wiki/Error_cuadr%C3%A1tico_medio

Artículo hablando sobre el optimizador Adam: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>

Método de la ventana: <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>

NASDAQ: <https://es.wikipedia.org/wiki/NASDAQ-100>

Valores más populares: <https://www.estrategia-bolsa.es/indicadores-tecnicos-utilizados-bolsa.html>
<https://www.tradingybolsaparatropes.com/blog/5-indicadores-imprescindibles-para-invertir-en-bolsa-en-2019>

Librerías:

TA-Lib: <https://github.com/mrjbq7/ta-lib>