TRABAJO FIN DE MÁSTER

BIG DATA Y BUSINESS ANALYTICS

Predicción del movimiento de valores financieros aplicando análisis del sentimiento en redes sociales

Alumno: Víctor Fiz Real

Fecha: 18 de septiembre de 2021

DNI: 52908947T

Tutores: Carlos Ortega y Santiago Mota

Índice de contenidos

1. Introducción: La predicción en el mercado

La predicción de valores de mercado ha sido un área de investigación muy activa a lo largo del tiempo. La hipótesis del mercado eficiente afirma que los precios de algunos valores en el mundo financiero están muy influenciados por información externa y siguen un patrón aparentemente aleatorio. Aunque esta hipótesis está aceptada a escala mundial, muchos investigadores y economistas aun intentan elaborar modelos que estimen cómo reaccionan estas variaciones a los distintos estímulos externos.

En este trabajo se pone a prueba una hipótesis basada en la premisa del comportamiento emocional de los inversores en la economía. Se da por hecho que la conducta, los estados de ánimo y las emociones influyen en su toma de decisiones a la hora de entrar en una posición o de salir de ella, formando una correlación entre sentimiento de mercado y sentimiento público.

La idea de que el precio de determinados activos está influenciado por representantes, ya sea porque estén respaldados por una base bastante sólida o porque sean influencias en el *Mass Media,* no es nueva y ha sido estudiada en análisis de todo tipo de valores: acciones, bonos, criptomonedas… Con el aumento de personas influyentes y de usuarios de redes sociales en general, la información acerca del sentimiento público se ha vuelto abundante, ya que es una herramienta perfecta para publicar y difundir emociones públicamente y cada aportación tiene un efecto sobre la opinión pública conjunta.

En el tercer apartado se explica más concretamente el dominio de estudio que se va a seguir.

1.1. Twitter

Twitter es un servicio de microblogging creado hace 15 años que permite que los usuarios compartan sus ideas o pensamientos en tiempo real, así como imágenes o contenido original entre otros. Permite además que dichos usuarios sigan a otros en función de lo que quieran leer en su “feed”.

Cada tuit está caracterizado por estar compuesto por un máximo de 280 caracteres (antes eran 140). De media, cada segundo se publica en torno a 6000 tuits, que se corresponde con unos 500 millones de tuits al día (aunque la mayoría está atribuida a “Cháchara sin sentido”). Esta cantidad es abrumadoramente grande, pero en la imagen mostrada a continuación se puede observar que Twitter tiene menos del 20% de usuarios que la red social más usada, Facebook.

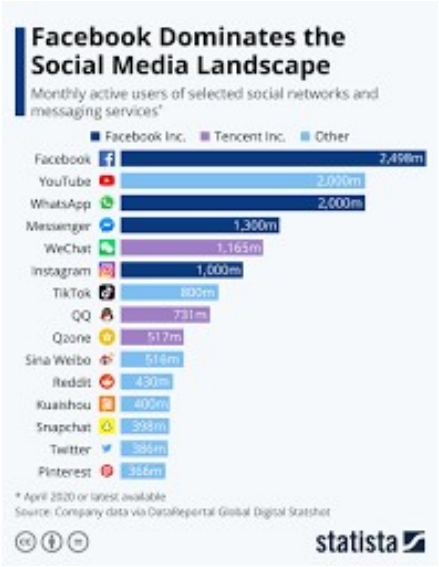

Ilustración 1: Top de redes sociales por usuarios activos mensualmente. Fuente: *statista*

No obstante, la forma de transmisión de la información de Twitter y el estilo propio que ha adaptado la red social han hecho que sea un portal de expresión de opiniones bastante apropiado en lo relativo a nuestro estudio. Dentro de esta categoría se ha considerado Reddit también, ya que comparte una estructura similar y se considera apropiada para el análisis de opinión.

1.2. Criptomonedas

Las criptomonedas, cuya primera aparición fue en 2009 con el Bitcoin, son un medio digital de intercambio caracterizado por emplear en las transacciones criptografía fuerte y estas son grabadas en el *Blockchain* (cadena de bloques). Esta cadena de bloques conforma el historial de transacciones de una o varias criptomonedas y otorga ciertas propiedades a estos activos digitales, tales como:

Es descentralizado (no hay autoridad central que pueda influir sobre sus propiedades directamente).

- Se mantienen todas las unidades y sus propietarios.
- El propio protocolo decide si se deben crear nuevas unidades monetarias.
- El único modo de garantizar la propiedad de una unidad es de manera criptográfica (hash).
- El sistema permite la transacción de unidades entre dos usuarios, pero solo se pueden efectuar si se puede probar el actual propietario de estas unidades.
- De la misma forma, si se efectúan dos transacciones idénticas, solo se tendrá en cuenta una.

El empleo de la tecnología *Blockchain* para las criptomonedas se respalda principalmente en la primera característica, que otorga independencia por parte de cualquier entidad y/o gobierno. Cuenta asimismo con otras ventajas frente a las transacciones típicas de los bancos, como que el mercado de criptodivisas está abierto 24 horas al día 7 días a la

semana, que las transacciones son casi instantáneas, que las transacciones son marcadas por los usuarios y por los *miners*, que sea una red de transacciones transparente y segura, ya que todos los nodos poseen el historial de transacciones (complica enormemente un ataque informático). No obstante, cuenta con otras desventajas como el exceso de privacidad; el anonimato otorgado por el sistema evita que tanto particulares como organismos puedan ser identificados y acusados por haber ejercido actividades ilegitimas.

De entre todas las criptomonedas, la primera y más popular a lo largo de estos años ha sido el Bitcoin. Fue creada en el año 2009 por uno o varios desarrolladores bajo el seudónimo de Satoshi Nakamoto y desde entonces ha sido la moneda mejor valorada. Emplea un sistema de prueba de trabajo para garantizar la seguridad en las transacciones, evitar la falsificación y otorgar recompensa por bloques a los *miners*. Por el contrario, este sistema es el responsable de un gasto energético y la consecuente contaminación desproporcionados a nivel global.

Ilustración 2: Criptomonedas según su capitalización de mercado actualmente. Fuente: *www.statisticsanddata.org*

El valor del Bitcoin y del resto de criptomonedas ha atraído a lo largo de los últimos años mucha atención en el campo del desarrollo de modelos de predicción, y las redes sociales han sido un muestrario perfecto. Según Rick Burgess, esto se debe a:

- El valor del Bitcoin está determinado casi en exclusiva por la demanda del mercado, ya que la oferta monetaria a pesar de ser inflacionaria es predecible y no está respaldada por ningún valor físico (a diferencia de otros tipos de dinero como el oro que está respaldado por su propio valor o el dinero fiat, que está respaldado por cada gobierno o autoridad central).

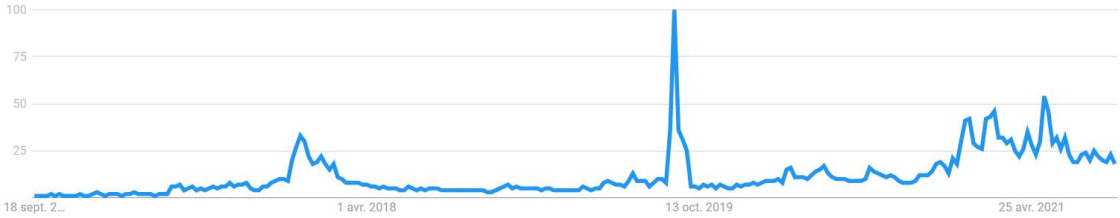

Ilustración 3: Proporción de búsquedas de "Bitcoin" desde el 18 de septiembre de 2016 hasta la actualidad en España. Fuente*: Google Trends*

Ilustración 4: Interés por comunidad autónoma en el mismo periodo. Fuente: *Google Trends*

- El comercio de Bitcoin está bastante ligado a la misma región demográfica que los usuarios de redes sociales, lo cual indica que su opinión es por lo general, documentada.
- Hasta hace relativamente poco, los principales usuarios eran particulares y no grandes instituciones.
- Los efectos favorables y adversos del transcurso de la moneda son retrasmitidos en primer lugar y principalmente en redes sociales.

Por lo expuesto anteriormente, Bitcoin será el activo empleado en el presente proyecto, ya que su popularidad y la volatilidad inherente a la opinión pública, lo convierten en un valor que satisface con creces las necesidades para este estudio.

2. Objetivos

Como se ha mencionado anteriormente, la predicción de valores de mercado ha sido un ámbito del estudio de muchas empresas de inversión y bancos y la finalidad de este trabajo es elaborar una solución probando que existe una relación entre aspecto emocional de los inversores en redes sociales y el valor del activo, teniendo en cuenta como parámetro el comportamiento emocional de los inversores y de la opinión popular para posteriormente productivizarse en versiones más específicas.

Dentro de los objetivos particulares del proyecto, se plantean la obtención efectiva de información de una red social (en este caso Twitter) sobre el Bitcoin, que estos datos sean transformados de forma efectiva para poder ser tratados como datos estructurados. Por último, del análisis inicial se espera obtener más de una característica de interés a parte del precio, y se desea averiguar cómo influyen en función de cuantas se empleen en el modelado.

3. Metodología

En este apartado se explicarán con detalle todos los pasos seguidos a lo largo del proyecto, comentando el algoritmo llevado a cabo junto con las tecnologías empleadas.

El proyecto tiene una división muy clara en cuatro partes: en un principio la extracción de los datos relevantes para el trabajo, seguido de un análisis descriptivo de los datasets antes de ser procesados. En tercer lugar, los datasets serán preprocesados mediante algoritmos de Machine Learning de proceso de lenguaje natural y finalmente combinados y empleados sobre un modelo de Deep Learning sobre un conjunto de validación.

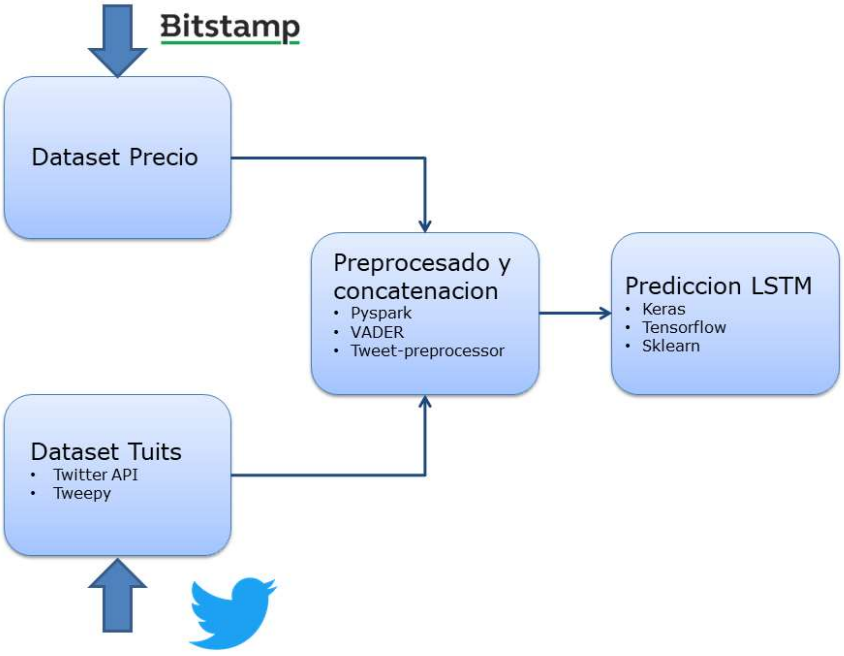

Ilustración 5: Esquema del proceso

La totalidad del proyecto está programada sobre Python, ya sea sobre un script o en dos ambientes distintos de Jupyter Notebooks, uno estándar de Anaconda y otro adaptado a Keras y Tensorflow, para el entrenamiento y desarrollo de modelos de Machine Learning y Deep Learning.

3.1. Extracción de la información

La extracción de información de open data generalmente es considerada trivial en un proyecto de análisis de datos, pero en este caso ha sido necesario recurrir a librerías específicas de extracción de información en redes sociales.

La API de Twitter es una plataforma creada por la misma compañía específicamente para desarrolladores y permite diversas funciones como widgets de implementación de tuits en páginas webs, publicación automática de tuits o retuits (bots), pero la de mayor interés es que permite hacer una recopilación de tuits en función de diversos parámetros como términos de búsqueda, el periodo de búsqueda (año, mes y día), el idioma y el límite, ya sea en tuits o en número de páginas. Ya que todo el código del proyecto está realizado en Python, la API de Twitter se emplea en conjunto con la librería tweepy, que habilita las funcionalidades de la plataforma en el script de Python empleado, llamado Mina_Tuits.py.

Una de las principales preocupaciones de esta parte del proyecto han sido los derechos de autor en cada uno de los tuits, ya que Twitter marca distinta normativa en función de la finalidad de los tuits extraídos. La finalidad declarada a la empresa es puramente académica ya que inicialmente el objetivo principal es la realización de este proyecto y generalmente cuando no se incluye metadata en la captura de tuits que contenga coordenadas de gps que puedan relacionar la información con la identidad de los usuarios que utilizan la plataforma.

En este caso la información extraída no supone ningún conflicto con la empresa o los usuarios ya que la finalidad es académica y en el programa de extracción se elimina toda la metadata y se conserva exclusivamente el contenido del tuit y la fecha exacta de publicación, los datos indispensables para el proyecto.

Para el empleo de la API de Twitter tan solo es necesario hacer una petición declarando las intenciones para el uso y proceso de los tuits y poco tiempo después permiten al usuario hacer uso de la API según el tipo de cuenta seleccionado. La versión Sandbox, que se corresponde con la más básica y gratuita, diseñada para testeo y estudios académicos, es la escogida para la realización del trabajo, ya que a priori es suficiente y las versiones premium y empresariales, están enfocadas a fines comerciales y quedan fuera de alcance en cuanto a presupuesto para este proyecto.

Una vez la identidad ha sido verificada la propia API genera unas claves de consumidor y de acceso con sus respectivas contraseñas y son almacenadas en un archivo csv por columnas para posteriormente ser leídas por la script de Python y permitir la búsqueda. Por la privacidad de la cuenta, las claves no son publicadas juntamente con el proyecto, pero no obstante el dataset resultado si será publicado y el programa está escrito de tal manera que se puede replicar con la clave personal de cualquier usuario con acceso a la API.

```
# Importar credenciales de usuario de twitter para acceder a la API

access_tweepy = pd.read_csv(r"C:\Users\vfizr\OneDrive\Documents\TFM\Login_twitter_API.csv")

consumer_key = access_tweepy["key"][0]
consumer_password = access_tweepy["key"][1]
access_token = access_tweepy["key"][2]
access_password = access_tweepy["key"][3]

authenticate = tweepy.OAuthHandler(consumer_key,consumer_password)
authenticate.set_access_token(access_token,access_password)

api = tweepy.API(authenticate)
```

Ilustración 6: Setup tweepy. Fuente: *Mina_Tuits.py*

Después de haber introducido la identificación personal se puede realizar la búsqueda mediante la función tweepy.Cursor(). Ya que la información extraída se pretende adjuntar al trabajo se graban el contenido de los tuits junto con la fecha en la que se crearon en un archivo csv. Los parámetros de la búsqueda serán los tuits que contengan la palabra bitcoin contenidos en el mes de noviembre de 2017 en lengua inglesa.

Por contrapartida, el dataset del precio del Bitcoin es considerablemente más fácil de obtener ya que casi cualquier plataforma de intercambio permite descargar el historial del precio del Bitcoin según distintos intervalos de tiempo. En este caso se tendrá en cuenta el precio de la criptomoneda por horas. Por supuesto, los datos ligados a los valores mercantiles no poseen ningún derecho de autor ya que los mercados permiten que se puedan emplear para el estudio ya que es parte de la base del comercio, para analizar los valores más competitivos.

3.2. Análisis descriptivo

Después de haber realizado la extracción de datos de Twitter se obtienen 3 datasets distintos que se almacenan por separado ya que las búsquedas se han realizado en paralelo y son un poco voluminosos. En esta parte del trabajo se pretende realizar un análisis descriptivo sobre ambos datasets, mostrando los aspectos más relevantes de la forma más gráfica posible. Para una correcta representación y evaluación de los datos de estudio, el análisis descriptivo de los conjuntos será representado en este apartado como preámbulo a los resultados.

Para la manipulación de los datasets, se va a hacer uso a lo largo de todo el proyecto de la librería Pandas, que permite una fácil modificación de datasets y guardarlos y cargarlos en archivos csv.

En primer lugar, se cargan los datasets correspondientes a los tuits y se concatenan en uno único que contiene todos los tuits ordenados por la fecha de publicación, y posteriormente se conserva la columna correspondiente a los Tweets:

```
In [3]: # Se extrae solamente la columna que contienen los tuits para el análisis descriptivo

df = pd.DataFrame(TwDF,columns=[1])
df.columns=["Tweets"]

df.head()
```

Out[3]:

	Tweets
0	RT @Forbes: The Failure of SegWit2x Shows Bitc...
1	RT @mindstatex: Lots of love from unknown mine...
2	RT @FernandoHuamanX: Warning: Built-in Keylogg...
3	RT @LevelNetwork: Join our telegram. All infor...
4	RT @realsheepwolf: $DIGAF: FLOAT=16M, THE "ONL...

Ilustración 7: Tuits capturados

La función para calcular la longitud del dataset ofrece la cantidad de tuits recopilados, que ofrece una dimensión de la cantidad de datos de nuestro estudio.

```
In [4]: # Número total de documentos (tuits) recopilados

len(df)
```

```
Out[4]: 2564350
```

Ilustración 8: Número total de tuits

Por otra parte, se cargan los datos de los precios de Bitcoin en un dataset aparte, que como está estructurado por horas, tiene una dimensión mucho menor.

```
In [15]: Price_DF=pd.read_csv(r'C:\Users\vfizr\OneDrive\Escritorio\test\BitCoinPrice.csv',
                              error_bad_lines=False,engine = 'python',header = 0)
Price_DF.head()
```

Out[15]:

	Date	Close Price
0	10/30/17 0:00	6123.21
1	10/30/17 1:00	6131.35
2	10/30/17 2:00	6114.17
3	10/30/17 3:00	6153.11
4	10/30/17 4:00	6151.09

Ilustración 9: Carga del dataset de precios de Bitcoin por fecha

De la misma manera se puede obtener la cantidad de elementos o de horas que se han tenido en cuenta:

```
In [32]: len(Price_DF)
```

```
Out[32]: 672
```

Ilustración 10: Número total de medidas del precio

El tamaño de este dataset es mucho más reducido, de hecho, son aproximadamente algo menos de 28 días, que se corresponde con el periodo de estudio. De esta manera, no es necesario realizar ningún proceso de limpieza. Por el contrario, cada uno de los datasets que contienen los tuits contiene en torno a 850000 elementos, lo cual supone un coste en capacidad computacional bastante más grande. No obstante, están incluidos los retuits y no aportan ningún tipo de información al análisis descriptivo, ya que interesa a priori evaluar solamente los elementos únicos. Estos serán identificados buscando duplicados, pero, sin embargo, se conservarán los retuits en el análisis principal del dataset ya que el retuit simboliza la empatía y el apoyo de un usuario a un tercero, lo cual puede ser interpretado como que ambos usuarios manifiestan la misma emoción.

```
In [5]: import numpy as np

        # Para ver el numero de retweets se emplea la función numpy.duplicated

        np.sum(df.duplicated())

Out[5]: 1331185
```

Ilustración 11: Número de retuits en el dataset

La cantidad de retuits presentes en el dataset alcanza casi el 52% del total estudiado, lo cual permite sacar la conclusión de que los usuarios tienden a retuitear más que publicar su propio contenido. Se debe tener en cuenta que los usuarios que se están teniendo en cuenta no están siendo discriminados por el número de seguidores que tengan, y escogiendo una muestra distinta con usuarios con más seguidores, considerados influencias en las redes sociales, se reduciría la cantidad de retuits presentes, quedándose con una mayor cantidad de contenido original. Una vez concretado que no se necesitan para el análisis descriptivo se eliminan los duplicados con la función de pandas pd.drop_duplicates() y se obtiene un nuevo dataset de algo más de 1.2 millones de tuits.

A continuación, se define una función de limpieza que elimina los hashtags que contengan la palabra bitcoin tanto en mayúscula como en minúscula y se reemplaza por la palabra en minúscula y sin hashtag. También se sustituyen por espacios en blanco el símbolo de RT junto con todos los hashtags restantes que contengan caracteres alfanuméricos y símbolos y las menciones a otros usuarios, ya que todo esto no aporta información. Por último, los saltos de línea, los hipervínculos y la puntuación son eliminados y el texto se transforma a minúsculas:

```
In [80]: df.head(10)
```

Out[80]:

	Tweets	Tweets_cleaned
0	RT @Forbes: The Failure of SegWit2x Shows Bitc...	the failure of segwit2x shows bitcoin is di...
1	RT @mindstatex: Lots of love from unknown mine...	lots of love from unknown miners. miners a...
2	RT @FernandoHuamanX: Warning: Built-in Keylogg...	warning built-in keylogger found in mantis...
3	RT @LevelNetwork: Join our telegram. All infor...	join our telegram. all information about a...
4	RT @realsheepwolf: $DIGAF: FLOAT=16M, THE "ONL...	$digaf float=16m, the "only" exchange on ...
5	RT @haydentiff: @BryceWeiner My luggage likes ...	my luggage likes your your luggage.ðÿ¯bit...
6	RT @cryptodrivers: As Bitcoin becomes popular,...	as bitcoin becomes popular, i encounter a c...
7	RT @techreview: A crucial feature of Bitcoin i...	a crucial feature of bitcoin is its securit...
10	RT @bravenewcoin: The Bitcoin, Ethereum & Bloc...	the bitcoin, ethereum & blockchain supercon...
11	RT @ToneVays: In light of winning the #NO2X ba...	in light of winning the battle, spending r...

Ilustración 12: Proceso de limpieza análisis descriptivo

Para poder realizar un análisis descriptivo sobre un conjunto formado por textos se ha optado por hacer análisis de sentimiento sobre los tuits limpios. Para tal tarea se ha empleado la librería **TextBlob**, que aproxima el análisis del sentimiento mediante técnicas de procesamiento del lenguaje. Cuenta con la ventaja de que se trata de un modelo pre-entrenado cargado de la base de datos NLTK, encargada del procesamiento natural del lenguaje, y permite obtener dos parámetros principalmente, uno de sentimiento subdividido en tres emociones simples (positiva, neutra y negativa) y subjetividad (objetivo o subjetivo). En el modelo normalizado la polaridad va de más negativa (-1) a más positiva (1) pasando por el 0 que se corresponde con un tuit neutral, que generalmente habla de otra cosa y a pesar de haber mencionado la palabra de interés, no muestra ninguna opinión (al menos no una que el algoritmo haya detectado).

Para representar en un scatter plot la polaridad o sentimiento y la subjetividad de cada tuit se debe tomar una muestra del dataset, ya que representar en una gráfica de dispersión más de un millón de puntos no aporta información porque es difícil de distinguir. La muestra escogida es de 10000 puntos al azar sobre el dataset original y son representados gracias a las librerías seaborn y matplotlib, de representación gráfica.

```
In [86]: import seaborn as sns

figure(figsize=(16, 10), dpi=80)
xdataSample, ydataSample =  df_sample["Polarity"],df_sample["Subjectivity"]
sns.scatterplot(x = xdataSample, y=ydataSample)
plt.title("Sentiment Analysis Scatter Plot",fontsize=28)
plt.xlabel("Polarity",fontsize=20)
plt.ylabel("Subjectivity",fontsize=20)
plt.show()
plt.show()
```

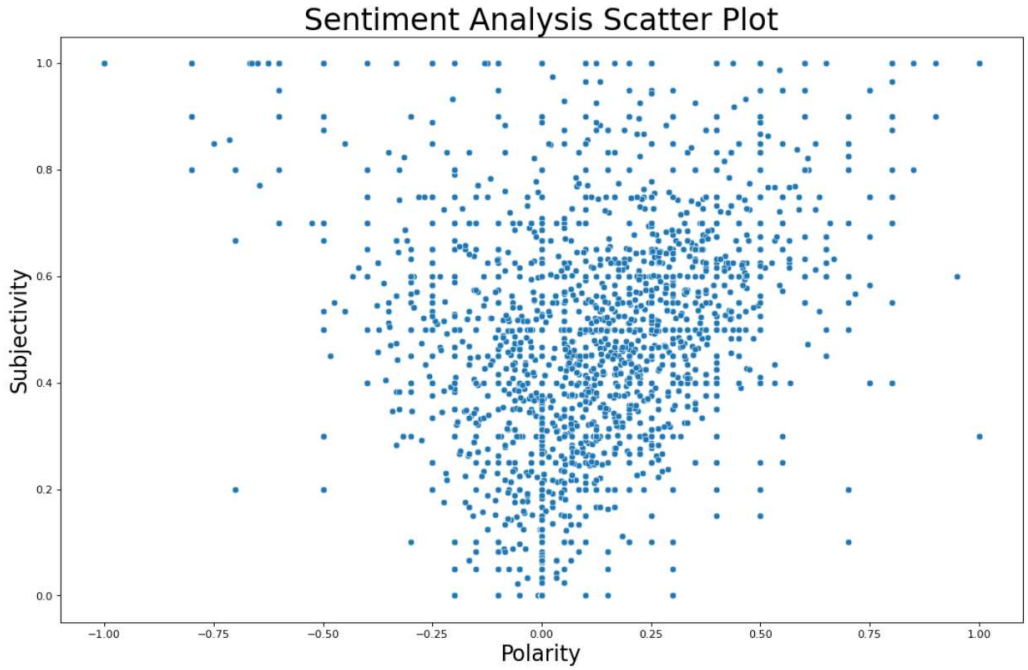

Ilustración 13: Polaridad-Subjetividad de los tuits capturados

Se puede observar que la mayoría de la dispersión está concentrada en el centro, lo cual indica que la mayoría de los tuits son bastante neutros por estar concentrados en torno a la polaridad nula y son algo subjetivos. De hecho, obviando los outliers, se observa que el algoritmo ha otorgado la calidad de tuit objetivo a un número muy reducido de la muestra y que, por lo general, tiene una polaridad cercana al cero. Por el contrario, a medida que los tuits aumentan en subjetividad se polarizan radicalmente, llegando a alcanzar para algunos tuits la subjetividad absoluta y polaridades de todos los rangos.

Para finalizar con el análisis descriptivo de los tuits obtenidos se obtiene un gráfico de barras definiendo la clasificación de sentimiento en opinión positiva, negativa y neutra.

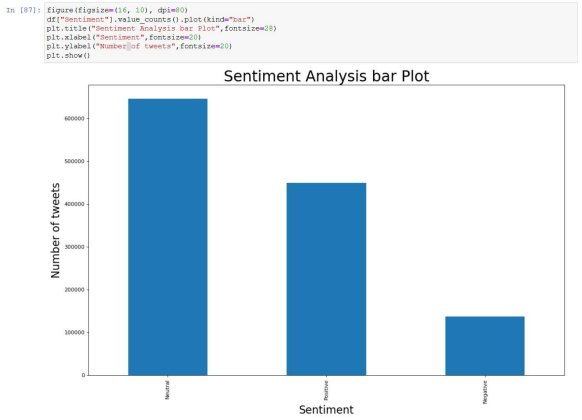

Ilustración 14: Análisis de sentimiento de los tuits

Como se había predicho en las observaciones relativas al gráfico, algo más del 50% de los tuits han sido calificados como neutros por el algoritmo de textblob y del resto, la mayoría son opiniones positivas.

Finalmente, respecto al dataset relativo al precio del Bitcoin, se muestra la gráfica de la evolución del valor de la moneda en dólares a lo largo del tiempo, y será utilizada como plantilla para las predicciones futuras.

```
In [33]: figure(figsize=(16, 10), dpi=80)
         plt.plot(Price_DF["Close Price"], color = "black", label = "Precio BTC")

         plt.title("Hourly Bitcoin Price NOV/2017",fontsize=28)
         plt.xlabel("Time (hours)",fontsize=20)

         plt.ylabel("Price (dollars)",fontsize=20)

         plt.show()
```

Ilustración 15: Evolución del Bitcoin en el mes de noviembre 2017

3.3. Preprocesado

En el análisis descriptivo se ha realizado una primera aproximación del dataset de tuits capturados mediante librerías de procesado natural del lenguaje, que para la representar los aspectos más fundamentales de los datos es suficiente para obtener conclusiones. No obstante, de este punto en adelante, se pretende llegar a un nivel de complejidad superior para acelerar el procesado de los datos y uso de modelos de Deep Learning, teniendo en cuenta principalmente que constituye una base sólida para permitir productivizar el proyecto en una versión futura.

Es por ello por lo que el núcleo de procesamiento del código gira en torno al empleo de **Pyspark**, la librería adaptada a Python de Apache Spark. Spark es un entorno de trabajo de tipo abierto (open-source) en local o en paralelo (mediante el uso de clústeres). Dentro del entorno, más concretamente se va a emplear el uso de Spark SQL que es un módulo empleado para el procesamiento de datos estructurados. Ofrece soporte SQL para Spark y delimita el proceso de demanda de datos almacenados en fuentes externas y en RDDs (Resilient Distributed Dataset), que son los dataframes empleados por la herramienta.

La aplicación usada, no obstante, es Pyspark, que se corresponde con la adaptación al lenguaje Python, sobre el que se desarrolla el proyecto, y la principal ventaja es que se incorporan las consultas de Hadoop Hive, que permiten una velocidad de computación de hasta 100 veces superior sobre el procesamiento de datos con librerías estándar como Pandas. Por otro lado, en Spark no hay forma a priori de realizar representaciones gráficas de los resultados obtenidos, pero son cubiertas por las librerías de Python. Es por esto, que la funcionalidad de las dos herramientas juntas permite satisfacer todas las necesidades de un proyecto de análisis de datos.

La herramienta Spark SQL permite trabajar sobre un entorno local, que emplea la capacidad computacional de la propia máquina empleada para la realización de tests y también permite la computación en paralelo mediante el uso de clústeres, para una finalidad industrial.

La carga de los datasets se realiza como en el apartado anterior, con la librería **Pandas,** concatenados en un único dataset de tuits y otro del precio. A continuación, los dataframes son transformados a RDDs de Spark para realizar un proceso de limpieza similar al del apartado anterior.

Para este proceso de limpieza se emplea una librería pre-entrenada para el procesamiento de tuits, llamada **tweet-preprocessor**. Al haber sido creada específicamente para el procesado de tuits según la jerga empleada en idioma inglés, se esperan unos mejores resultados que con el proceso de limpieza manual llevado a cabo anteriormente, que es más genérico.

A continuación, se debe realizar el análisis de sentimiento sobre los tuits limpios resultado del paso anterior, y esto es llevado a cabo mediante la librería **vaderSentiment.** VADER (Valence Aware Dictionary and sEntiment Reasoner) es un diccionario (lexicon) y una herramienta de análisis de sentimiento basada en reglas, que está armonizada con los sentimientos expresados en redes sociales. Esta librería es mucho más apropiada para el análisis de sentimiento que se desea seguir en este proyecto y otorga un parámetro denominado “compound”, que suma los valores de los sentimientos de cada palabra del lexicón, y está normalizado también entre -1 y 1.

El resultado en este punto serán dos datasets, uno que contiene la evolución del precio del bitcoin por horas y otro que contiene todos los tuits ya procesados con sus respectivas horas exactas de publicación y el análisis de sentimiento correspondiente. En este punto, los formatos de los dos datasets, que por provenir de fuentes distintas no coinciden, deberán ser unificados para la posterior concatenación. Un factor a tener en cuenta es que para cada medida de la evolución del precio del Bitcoin (cada hora) hay más de un tuit publicado, para lo cual se ha decidido hallar una media del análisis de sentimiento para los tuits contenidos en cada una de las franjas horarias e igualar el número de documentos de los dos datasets.

Finalmente, la concatenación de ambos datasets bajo el mismo umbral temporal será almacenado para su uso en el punto siguiente en formato csv.

3.4. Modelo de predicción

Una vez se obtiene el dataset definitivo, se procede a evaluar cual es la mejor manera de simular la evolución del precio de un activo, en este caso el Bitcoin, en función del análisis del sentimiento de los usuarios que opinan sobre ese tema en las redes sociales.

Para poder escoger bien el modelo es necesario identificar bien el problema que se desea resolver. En este caso el modelo debe realizar de forma precisa una predicción sobre la evolución del precio del Bitcoin, que en esencia es la predicción de una serie temporal.

Para este problema, hay una variedad de modelos de machine learning y Deep learning apropiados que en otros estudios citados en las referencias han obtenido resultados bastante fieles a la evolución de la moneda. Entre ellos se encuentran la regresión lineal, el k-Nearest Neighbors, ARIMA, Prophet, LSTM…

El **LSTM** (Long Short-Term Memory) es una RNN (Recurrent Neural Network) empleada en el campo del Deep Learning. A diferencia de las redes tradicionales, estos modelos tienen conexiones de retroalimentación y fueron introducidas inicialmente para ofrecer una solución donde las RNN tradicionales fallaban. Han sido empleadas en muchos campos con unos resultados bastante buenos, pero muestran resultados excelentes para la predicción sobre series temporales.

Generalmente están compuestos por una célula, una puerta de entrada, otra de salida y una de olvido. La célula recuerda los valores sobre periodos de tiempo arbitrarios y las tres puertas regulan el flujo de información que pasa a través de la célula.

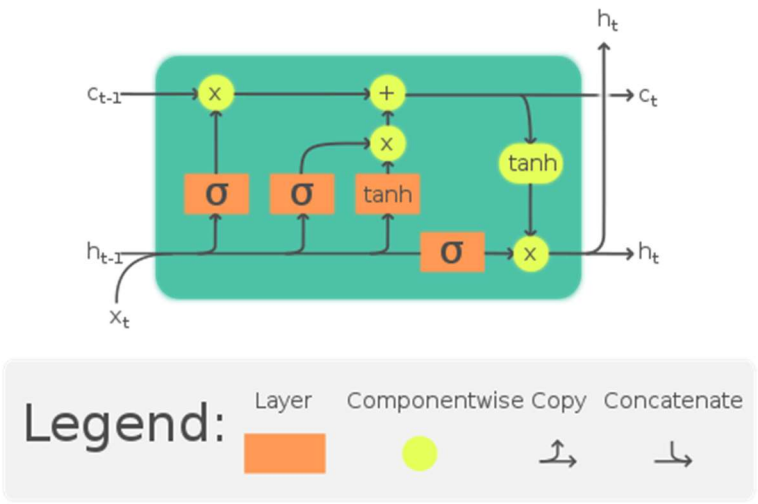

Ilustración 16: Típico esquema de una célula LSTM

Existen diversos modelos de LSTM: el clásico, el stacked, CNN, bidireccional, etc. Pero también depende del número de características que se tengan en cuenta en la elaboración del modelo.

Para el análisis de este modelo se debe cambiar de entorno de Jupyter Notebook a uno que esté adaptado para **tensorflow**, **Keras** y **SciPy**, ya que son las dos librerías con los modelos de Deep Learning necesarios para este apartado del trabajo. Para la elección de un optimizador, se empleará una librería derivada de Keras, llamada **elephas**.

De forma análoga a los apartados anteriores, se carga el csv de los datos preprocesados y se extraen las columnas de la polaridad y el precio del Bitcoin por horas y se almacenan en un nuevo dataset.

Para empezar a definir el modelo LSTM se debe definir una función que desfase arbitrariamente los valores en función del tiempo. Esta función crea columnas adicionales que contienen los valores desfasados de las características, desde t, t-1, t-2, …, t-n, siendo n un valor definido previamente.

A partir de este punto, el proceso se divide en un análisis mono característica y otro multi característica, que serán programados de la forma más parecida posible para estimar la variación. Se ha estimado que la muestra temporal estará entre 20 días y un mes, por lo que se considera conveniente escoger un tiempo de entrenamiento (fit) de una semana, es decir, de 168 horas. El conjunto de prueba empleado para validación estará compuesto por el resto del dominio, que ocupará la mayoría de la predicción.

El desfase temporal que se dará de entrada al modelo será de tres horas para sendos casos y por tanto lo único que variará serán las características. El dataset formado por las características desfasadas más la de output (el precio en este caso) será la entrada que debe introducirse en el modelo de predicción LSTM.

El modelo será secuencial para adaptarse a la serie temporal y contará con una única capa LSTM ya que se busca una aproximación y evaluar los dos modelos. Se empleará el optimizador Adam, que es sencillo de implementar y no tiene unos requerimientos de memoria excesivos, es computacionalmente eficiente y apropiado para problemas no estacionarios. Teniendo en cuenta que el programa puede ser reproducido en paralelo o en local se considera más apropiado buscar unos requisitos poco exigentes para realizar la primera aproximación. El error escogido es el MAE (Mean Absolute Error) que mide la magnitud media de los errores en un set de predicciones sin considerar su dirección. Adicionalmente, una vez realizado el entrenamiento del modelo se medirá también el RMSE (Root Mean Square Error), que también mide la magnitud media del error y está definida como la diferencia entre la predicción y la observación actual.

Ambos errores se encuentran en el dominio desde cero hasta infinito y ofrecen una medida del error relativa a la dimensión de los parámetros. Además, están orientadas inversamente, es decir, cuanto menor sea el resultado mejor. Por el contrario, como el RMSE tiene en cuenta el cuadrado del error antes de que se haya aplicado la media, otorga mucho peso a los valores puntuales con un error muy grande, mientras que un error constante da un resultado más suave.

Para evaluar el modelo se realizarán distintas pruebas variando los parámetros (capas ocultas, epochs y tamaño de muestra) y de entre ellos se escogerá uno como solución. Teniendo en cuenta que es una serie temporal por horas, los tamaños de la muestra se probarán tomando valores representativos, como, por ejemplo, medio día, un día entero, etc. Todas las conclusiones serán expuestas en el siguiente apartado de resultados.

Finalmente, se representarán gráficamente los resultados con sus valores de precisión y las conclusiones.

Con ánimo de medir la dispersión de las características y dar otro punto de vista respecto a la predicción multicaracterística, se realizará adicionalmente una aproximación por regresión lineal simple, haciendo uso de la librería Pyspark.

4. Resultados y conclusiones

El primer paso fue la extracción de los tuits mediante la API de Twitter. La cuenta para desarrolladores de Twitter estándar cuenta con una capacidad muy limitada de demandas al servidor a la hora, lo cual ralentizó considerablemente la captura de tuits (fue incorporada una excepción en el código que permitía realizar nuevas demandas cada 10 minutos hasta que finalmente era permitido de nuevo) y, por tanto, los tuits fueron recolectados simultáneamente en tres documentos csv distintos durante más de una semana.

A pesar de haber expuesto el análisis descriptivo en la parte de metodología, con ánimo de dar al lector una perspectiva general de los datos que se manejan en el proyecto, se parte de la base de la extracción de tuits en bruto para un posterior preprocesado de la información de forma más precisa gracias a las herramientas adaptadas para textos provenientes de redes sociales. No obstante, es necesario remarcar que el principal motivo por el cual se requiere a Spark como herramienta de computación de alta velocidad es porque en el análisis descriptivo, queda demostrada la ineficiencia de Pandas para lidiar con datasets extensos. La representación final de dispersión de la polaridad y la subjetividad se debe realizar con una muestra de 10000 datos que se corresponde con una representación de algo menos del 1%.

Para la ejecución del preprocesado, se decide, por simplicidad, compilar el programa entero en una máquina local. Esto supone modificar previamente los parámetros de sistema de Pyspark para aumentar la capacidad en el momento de introducir el dataset completo, quedando en evidencia por segunda vez que se trata de una semilla demasiado grande para una prueba (es necesario hacer reparticiones para abaratar el coste computacional). No obstante, se realiza el preprocesado del dataset formado por los tuits brutos y el del precio del Bitcoin según se ha explicado en la metodología y se obtienen los siguientes resultados:

```
+-------------+-------+-----------------+
|     DateTime|  Price|  Cleaned_BTC_Time|
+-------------+-------+-----------------+
|11/3/17 12:00|7321.09|2017-11-3 12:00:00|
|11/4/17 16:00|7352.03|2017-11-4 16:00:00|
|11/5/17 18:00|7537.33|2017-11-5 18:00:00|
|11/2/17 23:00|7029.98|2017-11-2 23:00:00|
|11/3/17 17:00| 7290.6|2017-11-3 17:00:00|
+-------------+-------+-----------------+
only showing top 5 rows
```

Ilustración 17: Tabla limpieza precio

```
+-------------------+--------------------+--------------------+
|           DateTime|               Tweet|       CleanedTweets|
+-------------------+--------------------+--------------------+
|Tue Nov 07 12:36:...|ðŸ"¥ ðŸ"¥ The Bos...|The Boss of Bitco...|
|Wed Nov 08 08:25:...|Electrum nodes ca...|Electrum nodes ca...|
|Wed Nov 08 13:03:...|The November 16th...|The November 16th...|
|Wed Nov 08 16:49:...|RT @RandyHilarski...|Bitcoin News SEC ...|
|Mon Nov 06 18:22:...|Have we seen the ...|Have we seen the ...|
+-------------------+--------------------+--------------------+
only showing top 5 rows
```

Ilustración 18: Tabla limpieza tuits

	DateTime	P_Neg	P_Neu	P_Pos	P_Comp	Price
0	2017-10-31 05:00:00	0.028733	0.893319	0.075142	0.098390	6158.76
1	2017-10-31 06:00:00	0.033298	0.866806	0.099268	0.133432	6105.90
2	2017-10-31 07:00:00	0.032960	0.868896	0.096757	0.129065	6094.36
3	2017-10-31 08:00:00	0.032679	0.884061	0.083261	0.112910	6125.13
4	2017-10-31 09:00:00	0.035034	0.868187	0.096220	0.130149	6165.00
5	2017-10-31 10:00:00	0.035643	0.860689	0.103108	0.135687	6170.77
6	2017-10-31 11:00:00	0.031788	0.864757	0.102949	0.157358	6233.74
7	2017-10-31 12:00:00	0.030206	0.880162	0.089151	0.125337	6201.03
8	2017-10-31 13:00:00	0.031716	0.867542	0.099453	0.141733	6332.34
9	2017-10-31 14:00:00	0.034077	0.862787	0.102122	0.134227	6363.13

Ilustración 19: Resultado de la concatenación de datasets

El dataset resultante contiene todas las modificaciones para pasar a la parte del modelado. Los dos análisis que se van a realizar van enfocados sobre las características utilizadas. En el primero se emplea además del precio, la puntuación de sentimiento (que se corresponde con la polaridad compuesta) y en el segundo se tendrán en cuenta todos. A continuación, se muestra la representación gráfica en función del tiempo de las características que se han analizado en ambos casos:

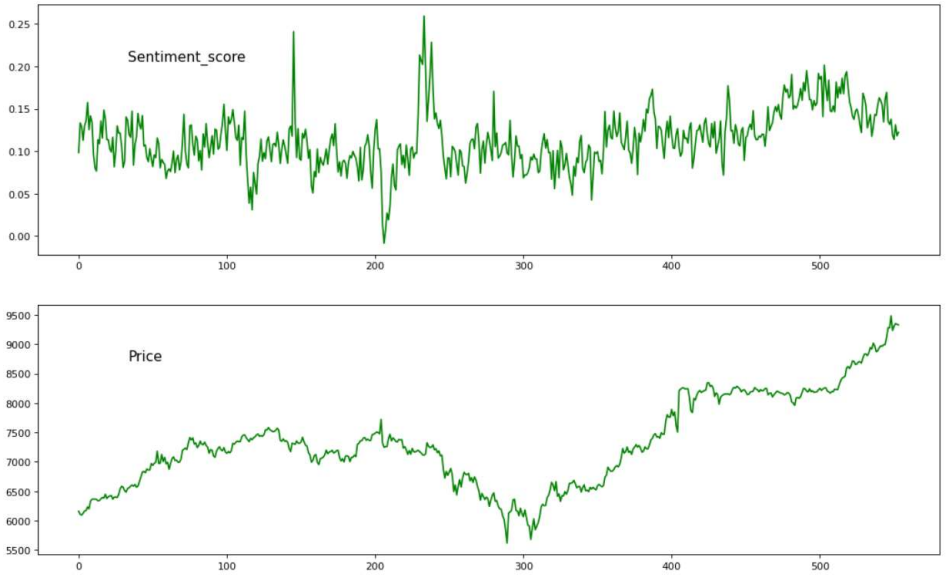

Ilustración 20: Características primer análisis

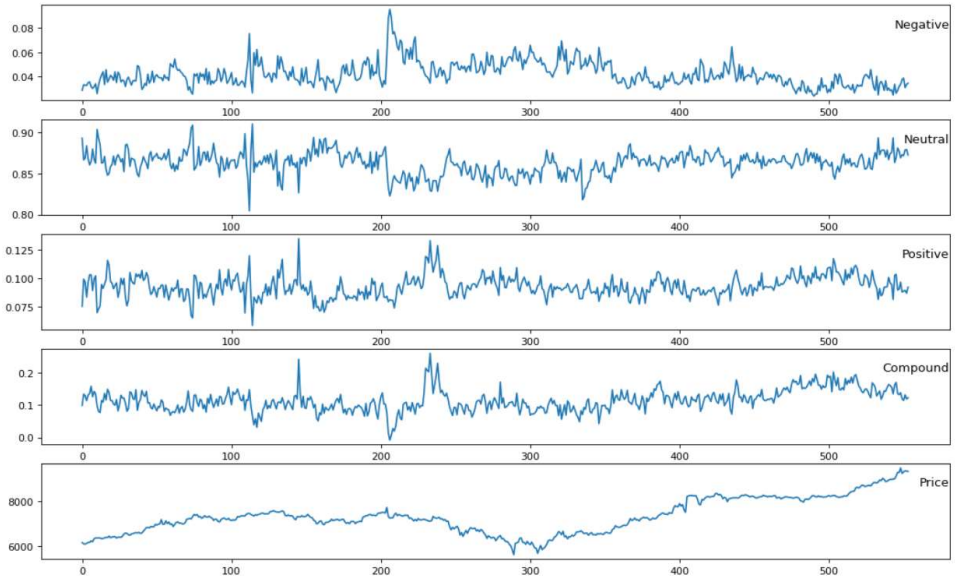

Ilustración 21: Características segundo análisis (multicaracterísticas)

Las condiciones del modelo de predicción no han cambiado entre un caso y otro, pues uno de los objetivos es analizar la diferencia entre emplear la característica de sentimiento, a priori, más importante, y emplearlas todas. Para escoger un modelo preciso se han realizado las siguientes iteraciones con los parámetros del modelo LSTM, aunque se ha mantenido el número de Epochs ya que se pueden obtener conclusiones

gráficamente representando errores de media, y el parámetro que determina la precisión del modelo es el RMSE:

Neuronas	Epochs	Batch size	RMSE mono	RMSE multi
10	50	12	618.9	222
50	50	12	215.8	80.2
100	50	12	176.7	178.7
50	50	6	231.1	235.1
50	50	18	257.5	88.6
50	50	24	258.5	96.9
50 (dropout)	50	18	222	81.2

Tabla 1: RMSE del LSTM para distintos casos

Inicialmente se han modificado las neuronas de la única capa oculta, obteniéndose que cuando el número es muy reducido los dos errores se disparan y cuando es muy alto tienden a igualarse con un error menor. De la misma manera, el batch size, tomado por conveniencia en intervalos con una diferencia de 6 horas, también ha mostrado resultados óptimos para valores intermedios (12 y 18 horas). De entre estos se ha escogido al azar el batch size mayor, y se ha recalculado la predicción realizando un Dropout, que elimina algunas capas aleatoriamente (en este caso el 20% de las 50) para evitar el sobre entrenamiento, ya que es quizá el aspecto más negativo de las redes neuronales LSTM. En teoría al sacrificar neuronas se obtienen peores resultados, pero el resultado de esta simulación es ligeramente optimista.

A continuación, se muestran los errores de entrenamiento y de validación para cada uno de los análisis:

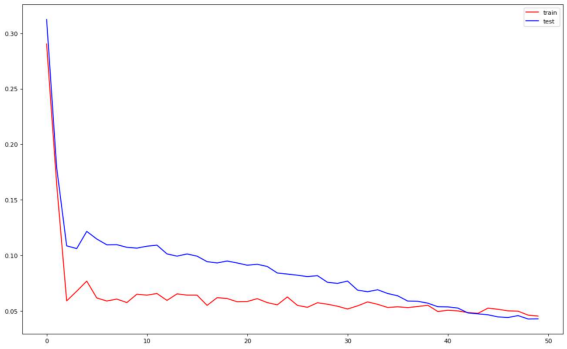

Ilustración 22: MAE monocaracterística

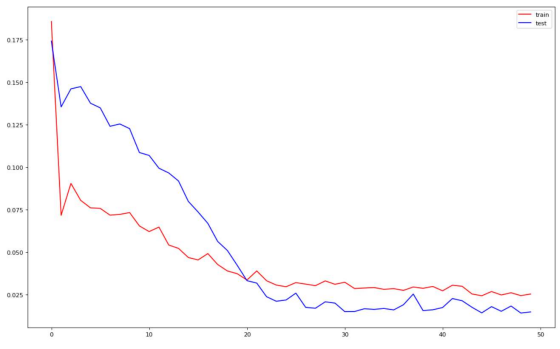

Ilustración 23: MAE multicaracterística

Se observa que el número de Epochs necesarios para converger en el primer caso es mucho menor, ya que pasados los primeros 10 el error disminuye paulatinamente a costa de sobre entrenarse, mientras que en el segundo tarda más, queda en torno al número 40. Es necesario apuntar que la implementación del módulo elephas que se pretendía adjuntar para obtener un optimizador más apropiado ha resultado imposible por incompatibilidades con el entorno de trabajo empleado, y ha sido necesario emplear uno predefinido.

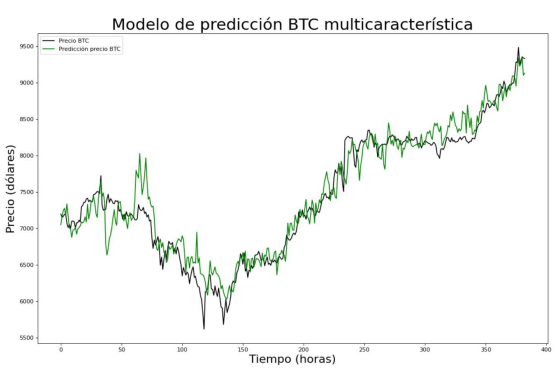

Ilustración 24: Predicción monocaracterística

Ilustración 25: Predicción multicaracterística

Se aprecia con bastante nitidez que el análisis teniendo en cuenta todas las características es más exacto que el primero, y obtener un error de RMSE inferior a 100 en el análisis multicaracterística cuando el precio es dos órdenes de magnitud superior demuestra un modelo muy preciso.

A la vista de estos resultados, se puede afirmar que: el LSTM es un buen modelo para predecir series temporales, en este caso de valores monetarios, y más importante, que hay una relación fuerte de influencia entre la opinión popular que los usuarios de Twitter expresan en las redes sociales. Esto está corroborado por el análisis de regresión lineal multicaracterística del último notebook, en el que se obtiene un coeficiente de determinación del 98% y un RMSE del 2% sobre las características normalizadas.

Uno de los puntos más importantes de este trabajo de cara a sus aplicaciones empresariales es que el código es muy fácil de modificar, para estudiar la influencia del análisis del sentimiento en redes sociales sobre el activo que se quiera.

5. Trabajo futuro

Durante el desarrollo del proyecto, se han observado determinados aspectos que quedan pendientes de estudio y son recogidos en este apartado, con ánimo de facilitar la productivización de esta herramienta. En primer lugar, se han incluido en el script de extracción de tuits funciones para filtrar la búsqueda en función de lo deseado. Un posible objetivo podría ser evaluar el peso de las influencias del tema, filtrando usuarios por un número mínimo de seguidores, por ejemplo.

De cara a la productivización, hay dos aspectos que sin duda deben ser tenidos en cuenta para futuras versiones: el empleo de bases de datos SQL y procesamiento en paralelo para incrementar la capacidad de la herramienta, y una aplicación de esta que funcione en tiempo real (con finalidad financiera, de actualidad, social, etc.).

Por último, merece la pena destacar que, por limitaciones de extensión del propio trabajo, se deben estudiar otros modelos de predicción que tienen una eficacia probada sobre series temporales, o incluso otros tipos de modelos LSTM. También se considera emplear otros activos y otras redes sociales de referencia, como Reddit.

6. Referencias

1. Germán Cheuque Cerda, Juan Reutter de La Maza (2019). Bitcoin Price Prediction Through Opinion Mining. *Pontificia Universidad Católica de Chile Santiago, Chile*
2. Anshul Mittal, Arpit Goel. Stock Prediction Using Twitter Sentiment Analysis. *Stanford University*
3. Eileen McNulty (2014). Bitcoin and Big Data: Can We Predict the Future Value of Virtual Currency? *Dataconomy*
4. Venkata Sasank Pagolu, Kamal Nayan Reddy Challa, Ganapati Panda, Babita Majhi (2016). Sentiment Analysis of Twitter Data for Predicting Stock Market Movements. *International conference on Signal Processing, Communication, Power and Embedded System*
5. Favio Vázquez (2018). Deep Learning with Apache Spark. *https://towardsdatascience.com*
6. Luis Gascó (2021). Text Mining. *UNED*
7. Jason Brownlee (2017). Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras. *https://machinelearningmastery.com*
8. Jason Brownlee (2016). The 5 Step Lifecycle for Long Short-Term Memory Models in Keras. *https://machinelearningmastery.com*
9. Diederik P. Kingma, Jimmy Ba (2017). Adam: A Method for Stochastic Optimization. *Cornell University*

7. Anexo

7.1. Scripts y notebooks

Ejecutados en orden:

1. Mina_Tuits.py
2. Analisis_descriptivo.ipynb
3. preprocesado.ipynb
4. Serie_temporal_LSTM.ipynb
5. LR_multicaracteristica.ipynb

7.2. Datasets adjuntos

1. tuits1.csv, tuits2.csv, tuits3.csv
2. Bitstamp_BTCUSD.csv
3. results.csv