Los eventos globales y su influencia en el mercado de valores, usando técnicas de Big Data

Autor: Damián Gil González

14 de octubre de 2021

Damián Gil González

Resumen

Este estudio se ha enfocado en dos pilares. El primero es en la predicción de la fluctuación del valor de las acciones de distintas compañías usando modelos de *machine learning*. El segundo, aplicar un análisis sentimental a las noticias que están relacionadas con estas compañías, para poder predecir subidas o bajadas del precio de las acciones. Siendo el objetivo, usar estos dos estudios a la vez para mejorar los resultados de los pronósticos. Para ello, se usará herramientas y técnicas, usualmente aplicadas en el mundo del *big data*.

1. Introducción

Aplicar técnicas de *ML* para predecir valores futuros es algo que en la actualidad es frecuentemente visto. No es nada nuevo intentar predecir el valor de la ación de la empresa X para poder sacar beneficios en la compra-venta de acciones.

Por otro lado, algo que cada vez es más notable es la fluctuación de la bolsa según el sentimiento grupal del momento. Este sentimiento es controlado a través de las noticias, eventos y redes sociales. Esto permite que se pueda visualizar el sentido de la fluctuación del mercado de valores si las noticias, relacionadas con alguna empresa en particular, son positivas o negativas. En este estudio se intentará determinar si es cierto que existe esta relación, para ello se aplicará un análisis sentimental a noticias y eventos.

Por último, demostrada ya la relación de la opinión social y el mercado de valores, se aplicará modelos de *machine learning* para mejoras las predicciones de la fluctuación del mercado de valores.

2. Métodos de Predicción

Este apartado se va a aplicar métodos de clasificación y regresión de machine learning a nuestro caso de estudio. Para ello se va a hacer primeramente un tratado de datos, para más adelante aplicar los métodos de clasificación/regresión de la librería *MLlib*.

2.1. Preprocesado de datos

Veamos paso por paso como modificamos el data set original hasta convertirlo finalmente en el input de los modelos. Tomaremos como referencia los siguientes artículos [1]-[4]-[2].

- A través de la aplicación de *Yahoo Finance* obtenemos el data set con los datos relacionados con el valor de las acciones de la empresa deseada. En la figura 1a se puede apreciar el formato y el código python usado es el siguiente:

```
!pip install pytrends

msft = yf.Ticker(company)
hist = msft.history(start=start_date, end=end_date)
spark.createDataFrame(hist.reset_index())
```

- En este paso vamos a crear un nuevo tipo de valor a partir de los parámetros *Low* y *High*. Usando la ecuación 1, creamos una columna en nuestro data set que representa la fluctuación max/min del precio de las acciones. Quedando pues, el data set igual al de la figura 1b.

$$LH_{percentege} = \frac{LowPrice - HighPrice}{LowPrice} \tag{1}$$

- Este último paso es añadir la columna denominada *Balance*. Esto es tan solo aplicar la ecuación 2. Quedando tal que así nuestro data frame, figure 1c.

$$Balance = ClosePrice - OpenPrice \tag{2}$$

- Finalmente aplicamos un *MaxMin Scaling* para asegurarnos que todos los valores son positivos, ya que algunos métodos no permiten valores negativos como inputs. Además, añadimos la columna más importante de este proceso, esta columna se denomina *rise* y sus valores son 1 o 0. El valor igual a 1 significa que el valor de las acciones de un día en concreto, al final del día, aumentaron su valor. Esto se traduce en un valor de la columna *Balance* positivo. El caso opuesto, valor igual a 0, significa un balance negativo. Obviamente esta columna solo será usada por los métodos de clasificación. Esto se aprecia en la figura 1d. Para la creación de esta nueva columna se ha usado el siguiente código:

```
def rise_or_not(open_, close_):
  if (close_ - open_) > 0:
    return 1
  else:
    return 0
rise_or_not_UDF = udf(lambda x,y :rise_or_not(x,y))

df_rise = df.withColumn("rise", rise_or_not_UDF(col("Open"), col("Close")))
```

- Para que nuestros modelos de la librería *MLlib* puedan trabajar con el data set se debe crear el vector denso a la vez que se le carga esta información al pipeline del modelo. Aquí un ejemplo con uno de los modelos de clasificación:

```
modelo_nb = NaiveBayes(smoothing=1.0, modelType="multinomial")
labelIndexer = StringIndexer(inputCol="rise", outputCol="label")
vecAssembler = VectorAssembler(inputCols=inputs, outputCol="features")
pipeline = Pipeline(stages=[labelIndexer, vecAssembler, modelo_nb])
```

y otro para los modelos de regresión:

```
vectorAssembler= VectorAssembler(inputCols = inputs , outputCol = 'features')
data_set_dense_vector = vectorAssembler.transform(data_set)
```

Figura 1: Preprocesamiento del data set del histórico del precio de las acciones.

Como vemos, se posee varias variables que pueden ser usadas como inputs para los modelos. Se debe de tener en cuenta que muchas de estas variables tienen una gran correlación. Los parámetros añadidos en el preprocesado están formados por la combinación de las variables originales. Para hacer una elección correcta se muestra la figura 2, en la cual se aprecia perfectamente cuáles tienen una correlación demasiado alta entre sí. Se aprecia dos variables que aún no han aparecido en este estudio, *exp* y *balance exp*. Por el momento se obviará su presencia. Las variables que están dentro de un recuadro rojo serán las elegidas como inputs(a excepción de las dos comentadas):

```
inputs =  ["Open", "LH", "balance", "Volume"]
```

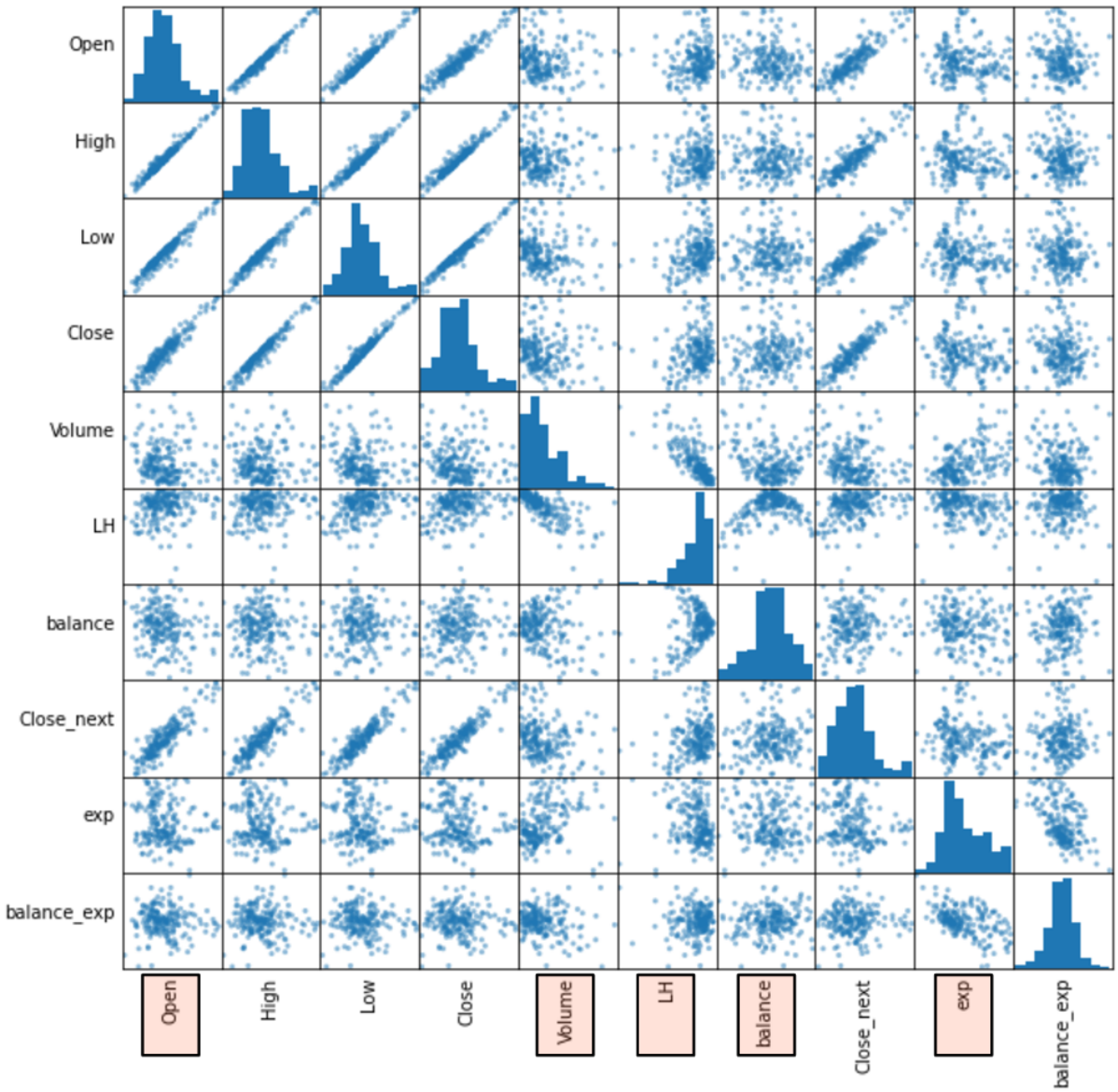

Figura 2: Las variables del data set representadas entre sí.

2.2. Métodos de Clasificación

Para poder determinar la precisión de los modelos con los que se harán las predicciones se seguirá dos caminos. Haremos uso del área bajo la *curva ROC* y de la *precisión*. La definición de *precision* es:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3}$$

Recordar que la *curva ROC* relaciona la sensibilidad de nuestro modelo con los fallos optimistas (clasificar los negativos como positivos). Generalmente, si aumentamos el *recall* (ver Eq 4), el modelo tenderá a ser más optimista e introducirá más falsos positivos en la clasificación. El punto clave de todo esto es que, el mejor de los casos, la curva se acerque lo máximo posible a la esquina superior izquierda de la gráfica, de manera que el hecho de aumentar la sensibilidad (*Recall*) no haga que nuestro modelo introduzca más falsos positivos, vease la figura 3.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4}$$

Veamos el contexto en el cual estamos y desarrollemos el por qué de la importancia de estos dos valores. Es crucial que el número de falsos positivos sea mínimo. En otras palabras, reducir los casos en los cuales se pronostica un aumento del precio de la acción, siendo finalmente una bajada del precio.

Entendamos la siguiente dos situaciones:

- Si el modelo predice que el precio de la acción va a aumentar, y no vendemos por ello para obtener un buen beneficio. Pero resulta que el valor de la acción disminuye (falso positivo), por desgracia estamos teniendo perdidas.
- El modelo nos predice que el valor de la acción disminuirá, por lo tanto vendemos para no estar en perdidas. Pero resulta que al final el valor de la acción aumenta (falso negativo), hemos obtenido posiblemente un beneficio menor, pero no negativo.

Por lo tanto, para reducir las perdidas, nos interesa un modelo **con la mayor *precisión* posible, a costa de un *recall* menor.**

Dicho esto, si visualizamos la imagen 3 podremos ver un ejemplo de dos modelos usados con nuestro data set. Es fácil distinguir que le modelo de *regresión logística* es superior en este caso al de *Naive Bayes* por el comportamiento explicado anteriormente. En pocas palabras, el area bajo la curva es mayor en el primer modelo comentado, siendo más óptimo para clasificar en este caso.

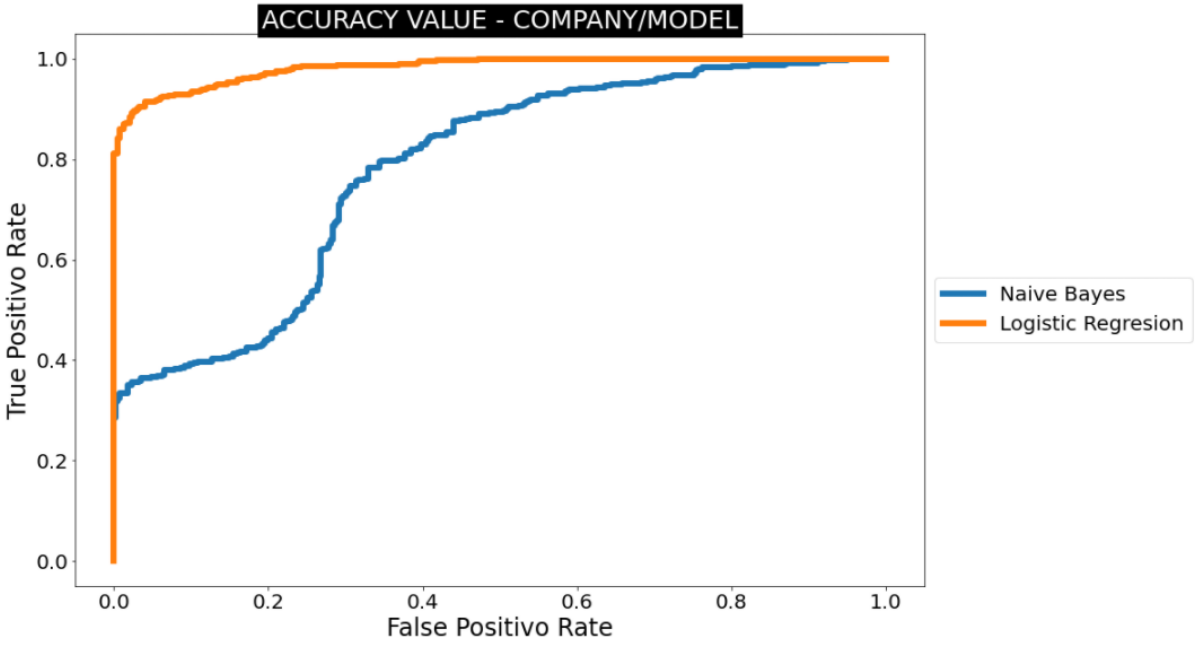

Figura 3: Curva ROC

Para poder obtener una clara comparación entre los modelos, y como se comportan para distintos data set (distintas compañías) se muestra la figura 4. En ella es notable que modelos como el *decision tree* es superior a todos, ya que siempre está próximo al uno. El que peor opera es el *Naive Bayes* en casi todos los casos.

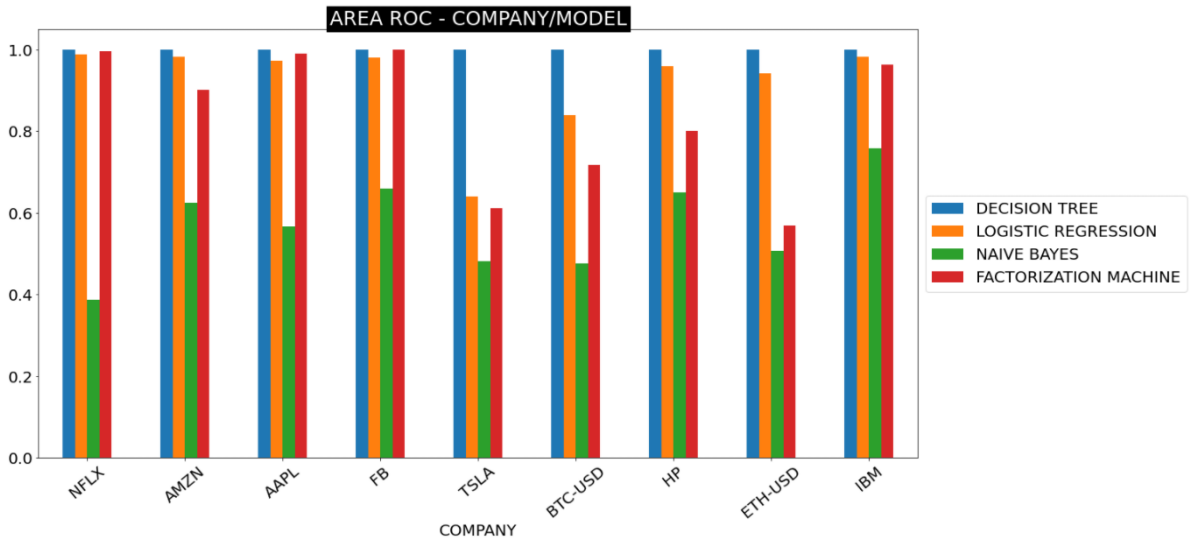

Figura 4: Área bajo la *curva ROC* para distintos modelos y compañías.

Como se ha comentado anteriormente, otra forma de determinar la eficacia del modelo es representando su *accuracy*. En la tabla 1 se muestran los resultados para algunas compañías, señalando en rojo los valores de precisión alarmantemente bajos, y en color verde las predicciones mejores efectuadas.

Modelo/Compañias	NFLX (%)	AMZN (%)	AAPL (%)	FB (%)	TSLA (%)	BTC-USD (%)	HP (%)
Decision Tree	0.998	0.998	1.0	0.997	0.998	1.0	0.996
Logistic Regression	0.935	0.957	0.895	0.905	0.516	0.628	0.895
Naive Bayes	0.327	0.600	0.370	0.751	0.322	0.396	0.332
Factorization Machine	0.860	0.658	0.593	0.700	0.564	0.559	0.529

Cuadro 1: Resultados de los modelos de clasificación para las distintas empresas.

En la figura 5 se presenta el plot de la tabla anterior. De nuevo la conclusión es clara, *decision tree* aporta las mejores predicciones, contrario al modelo de *Naive Bayes*. No es sorprendente que las dos formas de evaluar nuestros modelos coincidan.

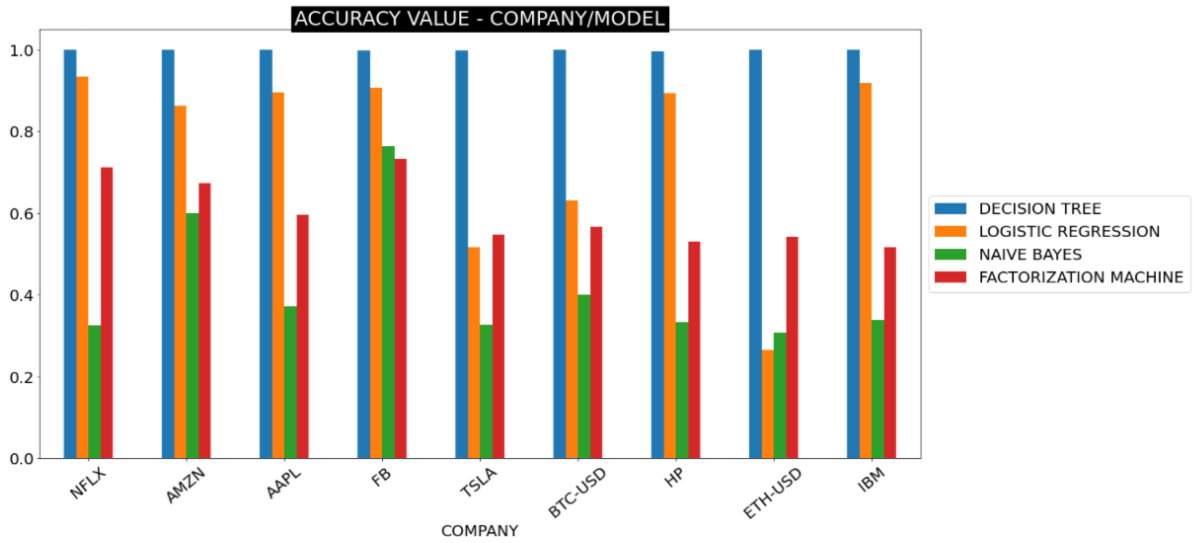

Figura 5: *Accuracy* para distintos modelos y compañías.

2.3. Métodos de Regresión

En este apartado se mostrará los resultados obtenidos al intentar predecir el valor de cierre (*Close*) de las acciones en cada día. La división de los data sets de prueba y de test se ha hecho algo distinto a lo que normalmente se hace. Él data set de prueba se ha construido desde el inicio en el tiempo hasta una cierta fecha en concreto, digamos 2 años por ejemplo. Él data set de test parte de la fecha de los dos años hacia adelante. Véase el código para aclarar esta cuestión:

```
def split_df(df_scale_spark):
  df_scale_pandas = df_scale_spark.orderBy("Date").toPandas()
  proporcion_train = int(df_scale_spark.count() * 0.7)
  stock_df_train = spark.createDataFrame(df_scale_pandas.iloc[0:proporcion_train])
      .orderBy("Date")
  stock_df_test = spark.createDataFrame(df_scale_pandas.iloc[proporcion_train:]).
      orderBy("Date")
  return stock_df_train, stock_df_test
```

Por último, mostramos, en la figura 6 los resultados de los distintos modelos en la predicción del valor de las acciones para diferentes empresas. Para evaluar los modelos se usará el parámetro R_2.

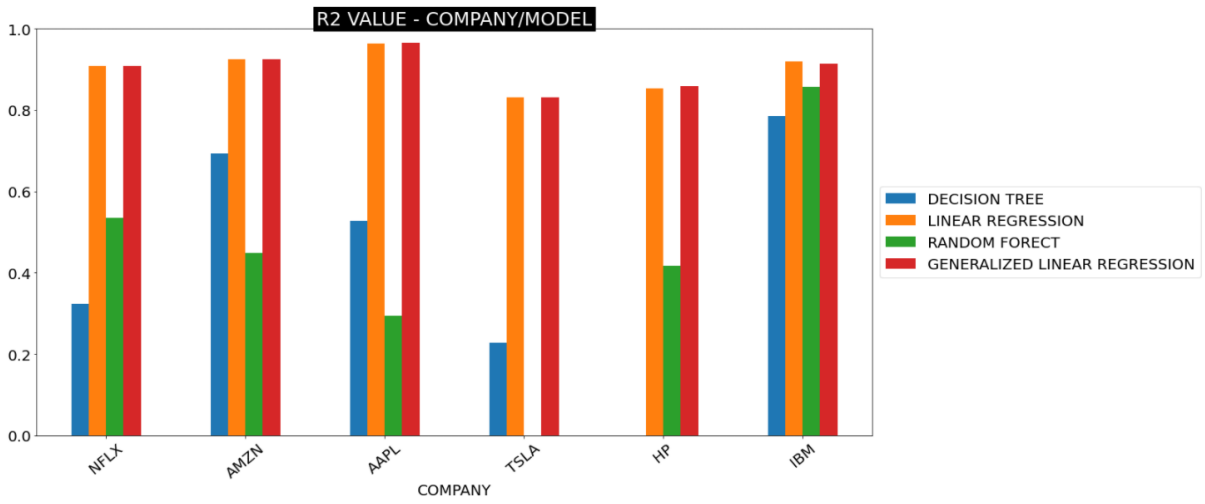

Figura 6: R_2 para distintos modelos y compañías.

Es destacable la fluctuación que sufre la calidad de los resultados de los distintos métodos para el set de empresas. Si ponemos atención en el modelo de *decision tree*, este fluctúa entre resultados aceptables (empresa **IBM**) y valores tan pésimos, llegando a ser el valor de R_2 negativo (empresa **HP**). Por otra parte, modelos tales como el *Generalized linear Regression* o el mismo *Linear Regresión* funcionan bien en todos los casos.

Para ver más detalle la precisión de cada modelo se muestra la tabla 2, donde podemos apreciar los variados resultados de los modelos para la empresa **Neflix**.

Modelo	R_2	$RMSE$
Lineal Regresion	0.908	8.50
Decision Tree Regresion	0.323	23.0
Random Forect Regresion	0.534	19.0
Generalized Lineal Regresion	0.909	8.43

Cuadro 2: Resultados de los modelos de regresión para la compañía **Neflix**.

3. Análisis sentimental

Se puede decir que la apertura del mercado de valores a todo tipo de público ha provocado que se manifiesten en estos últimos años unos fenómenos que hace 30 años no se apreciaban. A esto se le une la posibilidad de obtener información por múltiples canales o fuentes, provocando que el estar al tanto de eventos y noticias de actualidad sea algo sencillo y habitual.

Se tiene que tener en cuenta que, por nuestra naturaleza, el pensar y actuar como grupo nos insta seguridad y confianza. Si añadimos dinero y desconocimiento del tema, se nos presenta un escenario parecido al que vemos en el mercado de valores en estos momentos. En el cual, el valor de las acciones puede desplomarse o llegar a altos históricos dependiendo del sentimiento grupal. Este sentimiento de manada es algo tan sensible que puede variar según el tono, positivo o negativo, de una noticia.

Dicho lo anterior, en esta sección se va a intentar estudiar si el mercado y el sentimiento de los usuarios es tan fuerte como hemos comentado. Para ello, vamos a realizar en primera instancia una "data mining". Este va a consistir en hacer una recogida de noticias automática. Acto seguido, con los titulares y resumen de dichas noticias se procederá a una "sentiment analysis".

Lo primero de todo es definir nuestras fuentes de noticias y eventos.

3.1. Data Set: Noticias

La fuente de noticias usada procede de la página web https://finnhub.io/. Usando la API para Python es fácil obtener los artículos que están relacionados con alguna compañía en un rango de tiempo determinado. Tener en cuenta que las palabras clave que acepta la API usada son los tiques de las empresas que se pueden encontrar en el mercado de valores. El plan gratuito solo permite recopilar eventos de un año atrás. Por lo tanto, nos enfocaqueremos por el momento en tan solo usar datos en este rango de tiempo. Para cada día obtenemos una cierta cantidad de noticias, un ejemplo de lo obtenido es la figura 7.

```
Date: 2020-09-05 -- Company: AMZN -- N_news: 28
Date: 2020-09-06 -- Company: AMZN -- N_news: 23
Date: 2020-09-07 -- Company: AMZN -- N_news: 26
Date: 2020-09-08 -- Company: AMZN -- N_news: 43
Date: 2020-09-09 -- Company: AMZN -- N_news: 54
Date: 2020-09-10 -- Company: AMZN -- N_news: 93
Date: 2020-09-11 -- Company: AMZN -- N_news: 64
Date: 2020-09-12 -- Company: AMZN -- N_news: 34
Date: 2020-09-13 -- Company: AMZN -- N_news: 46
Date: 2020-09-14 -- Company: AMZN -- N_news: 143
Date: 2020-09-15 -- Company: AMZN -- N_news: 124
Date: 2020-09-16 -- Company: AMZN -- N_news: 138
Date: 2020-09-17 -- Company: AMZN -- N_news: 134
Date: 2020-09-18 -- Company: AMZN -- N_news: 96
Date: 2020-09-19 -- Company: AMZN -- N_news: 26
```

Figura 7: Ejecución de la obtención de los artículos de la compañía Amazon (AMZN). Se puede apreciar la fecha y el número de noticias de aquel día (N_{news})

3.2. Sentiment Analysis

Acto seguido hacemos el análisis sentimental de las crónicas. Para ello se usará la librería *NLTK* en Python. La idea es analizar las noticias de un día entero y añada la puntuación de negatividad, neutralidad, positividad y el campo compuesto (el más importante de todos) para cada una de ellas. El campo compuesto es tan solo la suma de los tres campos comentados anteriormente. Un ejemplo de código es el siguiente:

```
%sh
pip install nltk
pip install --upgrade pip
python -m nltk.downloader all

vader = SentimentIntensityAnalyzer()
# Iterate through the headlines and get the polarity scores using vader
scores = df_news['text'].apply(vader.polarity_scores).tolist()
```

Dando como resultado la figura 8.

	date	text	neg	neu	pos	compound
0	2011-04-23	Elon Musk: I'll Put a Man on Mars in 10 Years	0.000	1.000	0.0	0.000
1	2011-05-01	The Missing Secrets Of Nikola Tesla	0.306	0.694	0.0	-0.296
2	2011-09-27	Tesla, Meyl, and Jackson's Wireless Aetheric P...	0.000	1.000	0.0	0.000
3	2011-09-27	Aetherometry (Aeather science)	0.000	1.000	0.0	0.000
4	2011-09-27	[Video] Tesla transverse and longitudinal wave...	0.000	1.000	0.0	0.000

Figura 8: Data frame del resultado de aplicar análisis sentimental a los títulos de las noticias.

Para tener un valor unificado en una fecha concreta se hará la media del campo *compuesto* con todas las noticias de ese día. Finalmente nos queda un data frame parecido al de la figura 9.

date	avg(compound)_news
2020-09-12T00:00:00.000+0000	0.258628125
2020-09-11T00:00:00.000+0000	0.3020151515151515
2020-09-20T00:00:00.000+0000	0.18543076923076923
2020-09-06T00:00:00.000+0000	0.05844090909090909
2020-09-14T00:00:00.000+0000	0.3259678832116788
2020-09-16T00:00:00.000+0000	0.3588664179104478

Figura 9: Data frame del análisis sentimental realizado para las noticias de la empresa *Amazon*.

Teniendo ya la opinión/sentimiento de una empresa en concreto por día, se puede representar con el valor de las acciones para ver comportamientos comunes. En la figura 10 se aprecia el valor de las acciones de *Amazon* en este último año, y el sentimiento proveniente de las noticias para esta misma compañía. En primera instancia uno no sabría sacarte ninguna conclusión interesante. Quizás nos estemos pasando algo por alto, además de que los datos de los sentimientos tiene demasiado ruido para una representación clara.

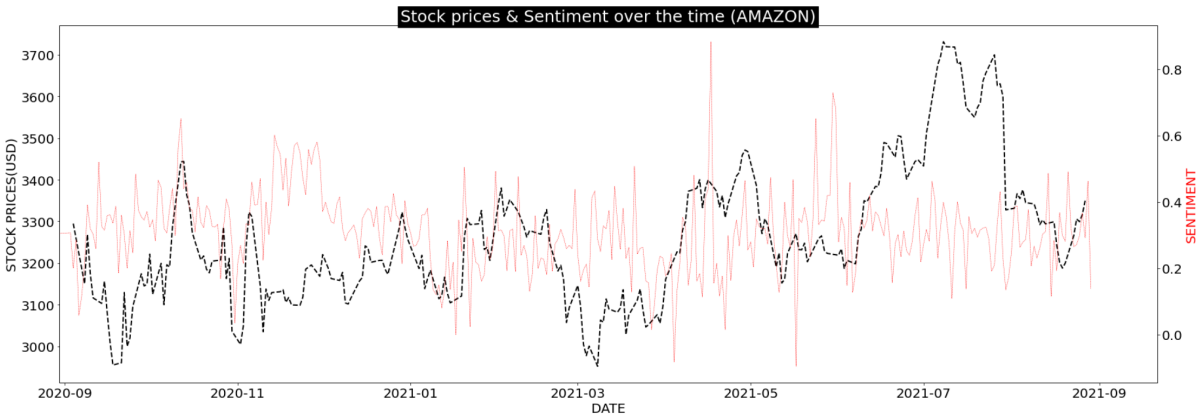

Figura 10: Dataframe del análisis sentimental realizado para las noticias de la empresa Tesla.

3.3. Popularidad de una empresa

Uno de los motivos por los cuales ha sido un fracaso el primer intento de hacer un análisis sentimental, es por el hecho de que no hemos tenido en cuenta como de popular es la compañía a lo largo del tiempo. Cuando hablamos de popularidad nos referimos con que frecuencia la empresa es *googleada* en internet. Para obtener dichos datos tan solo hace falta usar la librería *Pytrends* que obtienen los datos de Google Trend [3]. Combinando los valores de popularidad/impacto con los valores del análisis sentimental, quizás, se clarifique un poco la situación.

Por último, podríamos llegar a pensar en un día en concreto sale una noticia con gran impacto, es notorio que ese día se diferencia de los demás. Es decir, los días de gran popularidad provocan un gran impacto en comparación con algún otro día en el cual haya aparecido una noticia no llamativa. No es pues, una locura, intensificar estos días de alta popularidad de la empresa. Esto se puede hacer fácilmente elevando el valor de la popularidad de los días por alguna potencia impar(para respetar los signos).

En la figura 11a es la unificación de los dos tipos de datos comentados. Se empieza a poder intuir alguna tendencia algo más clara de lo que habíamos obtenido anteriormente. Además si aplicamos la intensificación de las noticias se aprecia finalmente una figura mucho más limpia, véase la figura 11b.

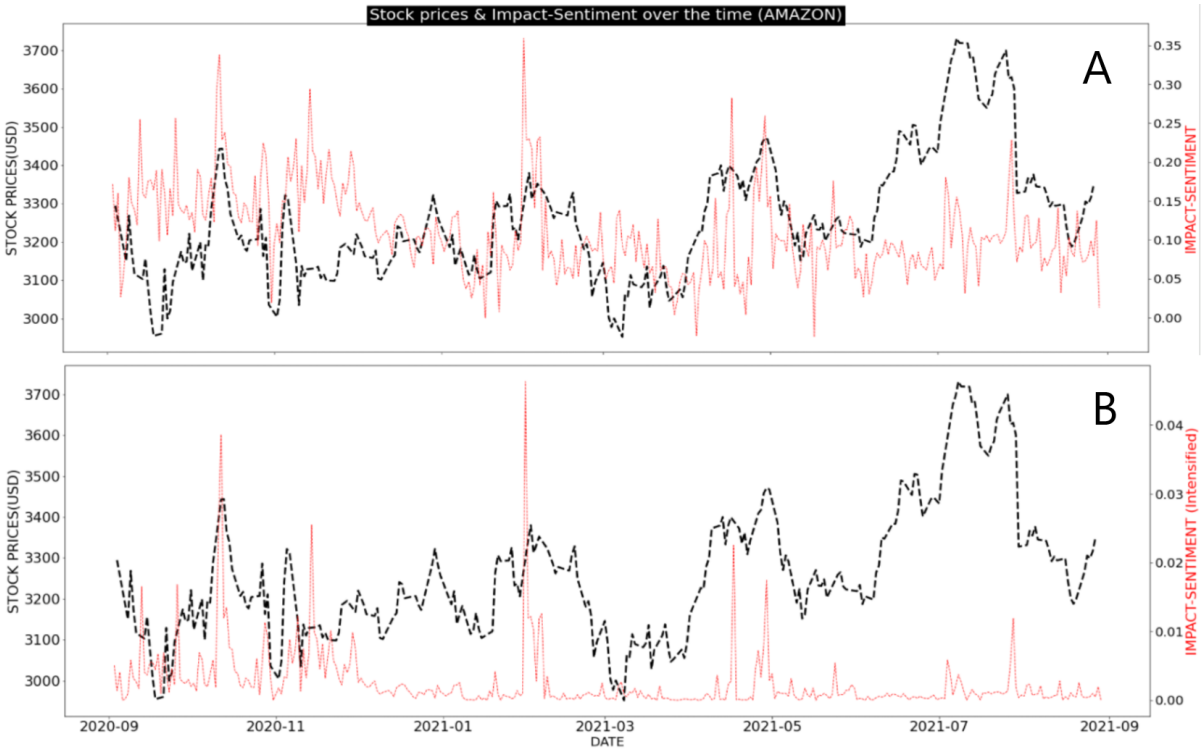

Figura 11: **A**) Representación de la popularidad combinada con la opinión a lo largo del tiempo. Se aprecia que los picos más altos de la línea roja (Popularidad-Sentimiento) es cuando es más positivo el ambiente y es cuando suben el valor de las acciones. **B**) Se intensifica el valor de Popularidad-Sentimiento en un factor 3, provocando picos muchos más claros donde el valor de las acciones suben posteriormente.

3.4. Delay de las noticias

Los seres humanos no leemos una noticia, actuamos en consecuencia y nos olvidamos por completo de ella. Para empezar no todo el mundo se entera de un evento al mismo tiempo, la propagación de la información de boca en boca puede hacer que un sentimiento se vaya propagando en un grupo cercano de personas. Es decir, hay un *delay* entre cuando apareció la noticia y la consecuencia de esta.

La negatividad o positividad en el ambiente económico perdura un cierto tiempo. Tanto es así que una bajada o subida del valor de una acción puede durar meses o minutos, según como haya calado la noticia en la comunidad.

Para nuestro caso, decimos pues que el sentimiento del mercado de días pasados afecta a nuestro presente. De esta forma podemos decir que la noticia de x días anteriores afecta en el sentimiento de hoy de alguna forma. Matemáticamente se traduce en lo siguiente [3]:

$$Sentimiento_{dia=y} = \sum_{x=0}^{x=y} Sentimiento_{dia=x} \cdot e^{-\frac{x}{\tau}} \tag{5}$$

Lo que viene a decir la Eq-5 es que si queremos saber el valor del sentimiento del día, por ejemplo 4, tenemos que hacer el sumatorio del sentimiento de los días anteriores teniendo en cuenta la distancia en el tiempo (la exponencial soluciona dicho aspecto). En la exponencial podemos encontrar el parámetro τ, denominado tiempo característico, que viene a significar un cierto valor que caracteriza el tiempo de reacción del sistema. En nuestro caso, el valor de esta variable determinará como de duradera es la influencia de una noticia. Las noticias más alejadas en el tiempo, es decir, mayores que τ, perderán casi toda su influencia.

Dicho esto último, hacemos de nuevo un procesamiento de los datos aplicando la Eq-5 a estos. Como resultado obtenemos la figura 12. En ella podemos observar que los picos ya no son tan abruptos y la representación de Popularidad-Sentimiento con delay sigue se complementa con el valor de la acción de *Amazon* en el tiempo. En esta imagen usamos $\tau = 7$, siendo una semana el tiempo característico del sistema. Este valor se ha escogido porque es el que mejor muestra la relación entre el valor de las acciones y el sentimiento-popularidad.

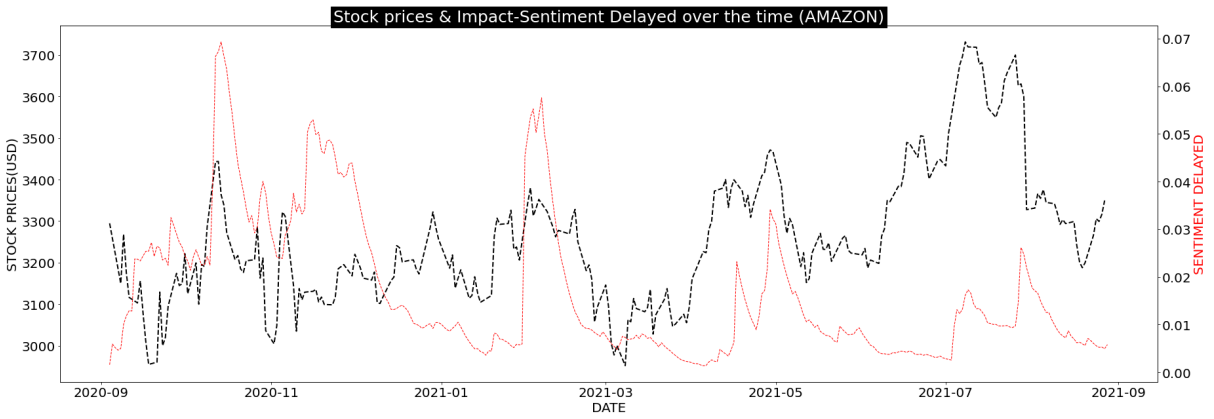

Figura 12: Representación de Popularidad-Sentimiento con delay aplicando la Eq-5. Usando como tiempo característico igual a 7 días.

Al observar la imagen se aprecia que la idea de que las noticias (que provocan estados de ánimo grupales) siguen afectando a los días posteriores de esta, no es tan descabellada del todo. De hecho ya sabemos que tiene bastante lógica, además de que el resultado es casi inmejorables. Es notable que cuando existe aparece noticias positivas para la empresa, el positivismo contagia a los inversores, que acto seguido procuran recaudar acciones de la empresa, ya que determinan que es una buena inversión. Pasa lo mismo cuando salen noticias negativas, por eso la dos curvan convergen en las subidas y bajadas.

3.5. Fuente de Eventos: Reddit

Tras realizar varios análisis usando los datos solo provenientes de noticias, no siempre salen resultados aceptables como los anteriores. Para diferentes compañías no es suficiente el análisis sentimental realizado. De hecho se puede mejorar dicho análisis usando otra fuente de eventos, *Reddit*. Añadiendo esta página web como fuente para nuestro data set al que se le aplicará el análisis sentimental, podemos ver una mejor resolución para nuestro objetivo. El código usado para obtener los datos de esta fuente es el siguiente:

```
%!pip install psaw

from psaw import PushshiftAPI
api = PushshiftAPI()
api.search_submissions(after=start_time, before=end_time, subreddit =
    palabra_clave, filter=["title", "subreddit"])
```

El flujo de nuestro estudio se puede ver en la imagen 13. Se aprecia que tenemos varias fuentes de datos que se le aplicará el análisis sentimental, para después poder realizar *plots* que nos permitan finalizar con unas conclusiones claras y sencillas.

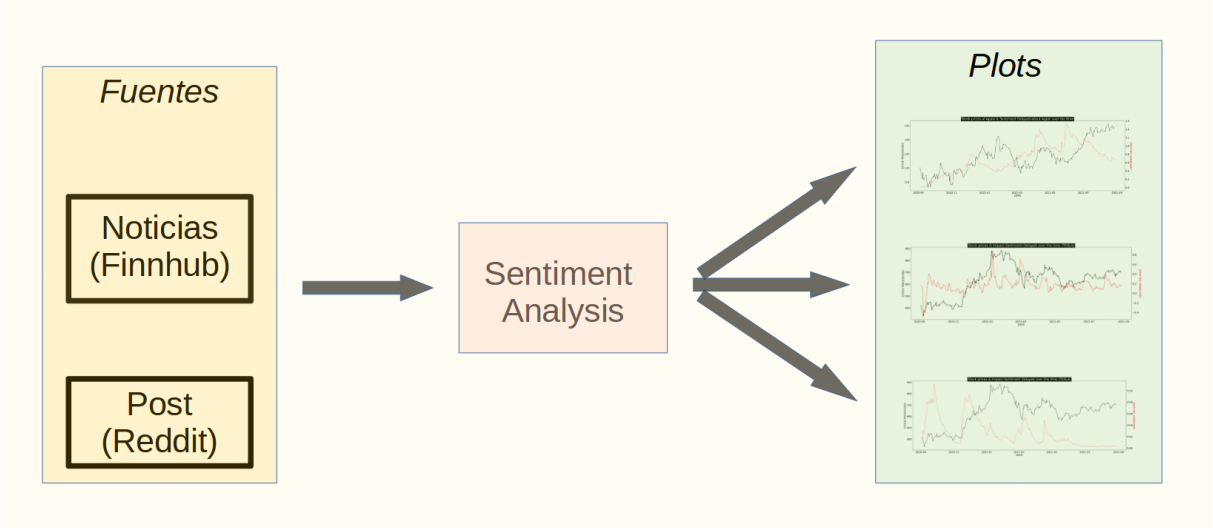

Figura 13: Flujo del análisis

Aplicando esto último, veamos como mejora el análisis de la empresa *Tesla*. En primer lugar veamos que resultado se obtiene sin usar la segunda fuente de datos. En la figura 14 no se aprecia una forma elegante del comportamiento de las acciones de *Tesla* y el sentimiento del mercado. Se podría decir que esta compañía no es tan conocida como *Amazon* y por eso las noticias no impactan tanto en ella. Veremos que, *Tesla* es mucho más nombrada en los foros y por ello el comportamiento del flujo de las acciones se explica mucho mejor añadiendo esta fuente de datos.

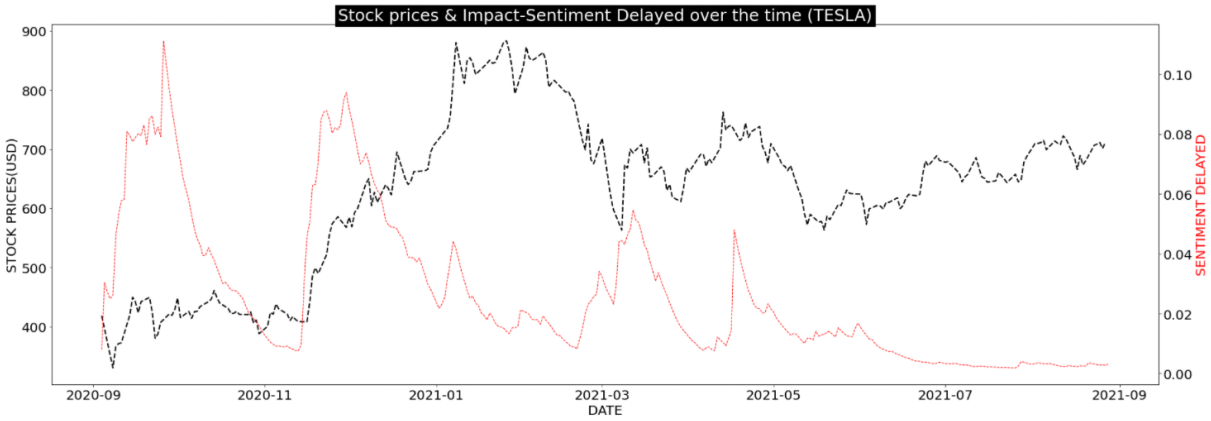

Figura 14: Representación del análisis sentimental usando como fuente de datos *Finnhub*. $\tau = 7$.

En la siguiente representación, se usará las dos fuentes como ya se ha comentado anteriormente. Dicho eso, es notable que el sentimiento producido por los post de *Reddit* combinado con las noticias del momento representa de una forma más fiel a las variaciones del precio de las acciones de la compañía de *Elon Musk*. Todo esto se puede ver en la figura 15. Algo que se debe de comentar es como se debe de apreciar el gráfico anterior.

Para llegar a la conclusión de que realmente existe una correlación entre el sentimiento del mercado y este, uno debe enfocarse no es los valores en sí, sino en el comportamiento de las curvas (fluctuaciones). Por lo tanto, al aumentar el sentimiento positivo, se apreciará un aumento en el valor de las acciones (y viceversa), que no es del todo proporcional al incremento del valor de la primera variable comentado anteriormente.

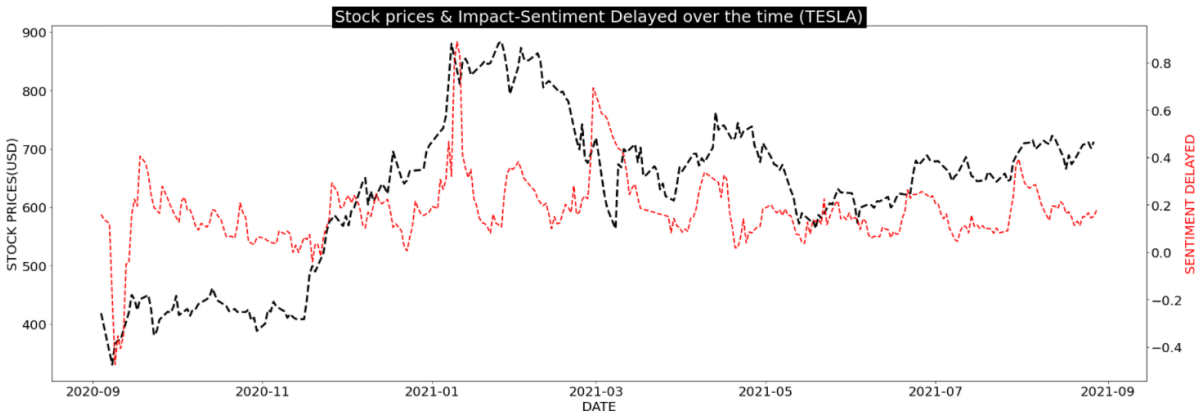

Figura 15: Representación del análisis sentimental usando como fuente de datos *Reddit*. $\tau = 7$.

4. Métodos predictores usando el análisis sentimental

Ya hemos visto en la sección 2 como podemos pedir con mas o menos acierto la fluctuación de la bolsa. Teniendo en cuenta el análisis sentimental realizado anteriormente, es fácil adivinar el siguiente paso. Utilicemos los resultados de la sección 3.2 y los modelos de predicción para mejorar los resultados.

Teniendo en mente el data frame de la figura 1, podemos añadir una columna que contenga el sentimiento del mercado para cada fecha concreta. Usando esto último y aplicando el *delay* explicado en el apartado 3.4, el data set nuevo está listo para ser introducido a los diferentes modelos predictores.

Tras varios intentos se observó una mejoría muy pobre en las predicciones. Esto se debe a que lo que realmente importa es la diferencia del sentimiento del mercado entre un día en concreto, y el día anterior a este. Por lo tanto, se debe añadir una nueva columna, esta columna almacena el balance entre el sentimiento de 'hoy' y el sentimiento de 'ayer', usando la ecuación 6.

$$balance_{exp}^{\text{day n}} = exp^{\text{day n}} - exp^{\text{day n-1}} \tag{6}$$

Finalmente nos queda el data frame de la figura 16. Usamos de nuevo la representación de las variables entre sí, que se puede encontrar en la imagen 2. Induce a la deducción de que es recomendable usar las mismas variables que se emplearon como inputs de los modelos de ML, pero añadiendo una variable extra, *balance exp*.

	Open	High	Low	Close	Volume	LH	balance	Close_next	exp	rise	balance_exp
1	0.274220790578358	0.2626374698090168	0.2881118814960581 5	0.2619505065982430 4	0.3021210553543714	0.8208703716791674	0.5475623713324813	0.1618430102973436	0.6157231012014657	1	0.1981124139857619
2	0.478471606809701 47	0.4782387128467293 6	0.3627435966268097 7	0.2863009871367765 4	0.456050416215962	0.3794040589547738	0	0.2107485025063673 4	0.3757895537425492	0	0.0558482840682414 4
3	0.690746268656715 8	0.6490233533121579	0.5231353970925099	0.5337670562676733	0.7888350656069228	0.3847872298228997	0.0544029563426483	0.4396251035184 36	0	0	0.6624926955679101
4	0.283582089552238 5	0.3561301842402464 5	0.3136390459420370 6	0.2538810800246063	0.5458190283591214	0.6039212237918549	0.4966878277670439	0.4062558336789168	0.2035292599908327	1	0.5615996190074478
5	0.354017898787313 27	0.4206488863952140 3	0.3802419321459447	0.4062558336789168	0.4393547476837699	0.6227426330234046	0.7119254022865 24	0.2863009871367765 4	0.3107100556771356 7	1	0.1970462031219368 8
6	0.491343283582089 45	0.5172115692568502	0.2907880592567013 7	0.4396251035184 36	0.8618139491134834	0	0.3914815291403105	0.2538810800246063	0.0697583235929066 5	0	1.0000000000000002
7	0.318137243470149 04	0.2778885817281642	0.2724663526523487 5	0.1937495536648343 7	0.3618609791656868	0.7176122549051744	0.2237342456457417 4	0.2619505065982430 4	0.5955922607942317	0	0.2711706035709973

Figura 16: Data Set añadiendo los resultados del análisis sentimental.

En lo que queda se usará los modelos de predicción, específicamente el de clasificación. Para ver la comparación de usar o no usar el análisis sentimental se muestra la tabla 3. En ella se aprecia la mejoría, en todos los modelos, de la predicción al usar la información adicional proporcionada por las noticias. Se a de comentar que en la tabla de resultados no se pueden observar algunos modelos ya usados anteriormente. Esto se debe a sin usar el análisis sentimental, el modelo ya es perfecto siendo su Accuracy = 1.

Modelo	Accuracy - Análisis Sentimental	Accuracy - Sin Análisis Sentimental	% Mejora
Logistic Regression	0.96	0.90	**6.66**
Naive Bayes	0.82	0.78	**5.19**
Factorization Machine	0.72	0.62	**16.13**

Cuadro 3: Resultados de los modelos de clasificación para la empresa *Amazon*, haciendo uso del análisis sentimental y no haciendo uso de este.

En la figura 17 muestra lo comentado de una manera gráfica. Para ello, se ha trabajado con los datos de las empresas *Amazon* y *Tesla*. Además se hace uso de los datos resultantes de aplicar el modelo *Factorization Machine*, para el caso en el que se emplea el análisis sentimental (**SA**) y en el que no (**NO-SA**).

El gráfico de barras resalta la diferencia de precisión del modelo usando y no el analisis sentimental. Como era de esperar, la precisión es más alta para cuando si se hace uso de este analisis.

El gráfico de puntos nos revela algo muy curioso. El valor del *recall* es mayor en los casos en los que no se hace uso del *SA*. Es decir, hemos mejorado la precisión del modelo, pero hemos disminuido el *recall* (ver Eq 4). En otras palabras, nos encontremos con un mayor número de situaciones en los cuales en modelo prevé una subida del precio, pero realmente baja el valor de la acción.

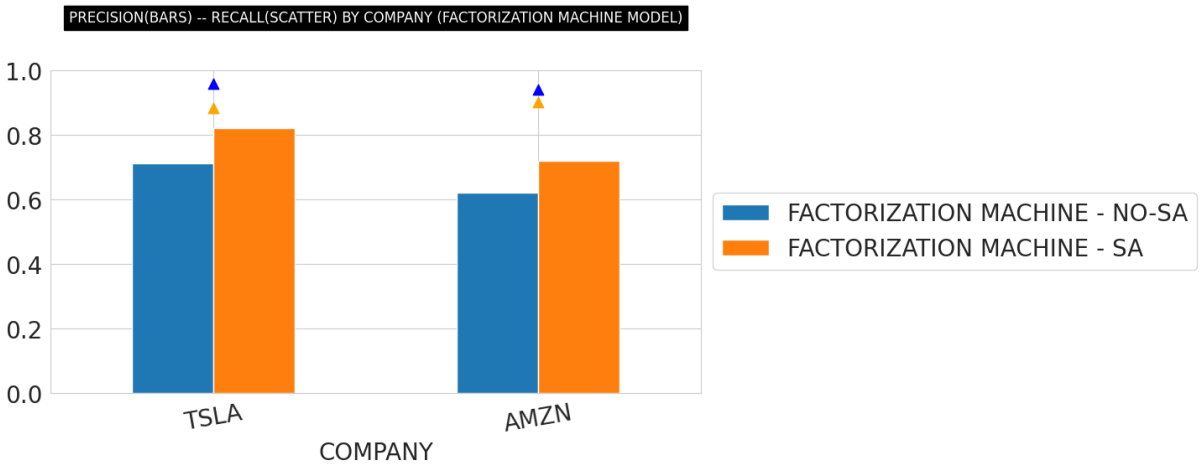

Figura 17: Modelo de clasificación: *Factorization machine*. Predicción usando el análisis sentimental (**SA**) y no usándolo (**NO-SA**).

Como se comentó en el apartado 2.2, si aumentamos la **precisión** del modelo, el **recall** disminuye. Además para nuestra situación, es interesante un modelo con la mayor *precisión* posible, a costa de un *recall* menor. Esto es buena noticia pues, ya que hemos conseguido un aumento de la precisión notable en muchos de los modelos.

5. Conclusiones

A lo largo de este estudio, se ha ido presentando los distintos pasos que han sufrido los datos crudos hasta la obtención del data set usado como input a las distintas técnicas de ML. Se ha intentado justificar la lógica de cada manipulación y la necesidad de añadir nuevas variables al set de datos. Finalmente se ha construido una fuente de datos consistente, lista para ser introducida en los modelos de predicción.

El primer tipo de predicción mostrado ha sido el de clasificación, sección 2.2. En este apartado se descubre la relación entre el *recall* y la *precision*. Llegando a la conclusión de que para el caso se estudio que estamos

abordando, nos interesa un modelo con mayor *precision* aunque esto provoque un *recall* menor. Tras ese análisis se muestran los resultados de distintos modelos de clasificación para diferentes empresas. Para sacar conclusiones fácilmente de hace uso de la figura 5. En ella, se llega a la conclusión de que modelos tales como **decision tree** y **logistic regression** son superiores a los demás. Se puede decir que en media, se obtienen buenos resultados de clasificación para todas las empresas.

Acto seguido se muestran los resultados de los métodos de regresión, sección 2.3. Modelos tales como **linear regression** y **generalized linear regression** sobresalen de los demás. El modelo con menor rendimiento es sin duda el de **decision tree**, véase la figura 6.

Finalizando con los modelos de predicción, se empieza a desarrollar el proceso de usar el sentimiento del mercado como indicador de posibles cambios alcistas o bajistas, sección 3. Además de la carga y transformación de datos estructurados elaborada en la sección 2.1, se ha tenido que realizar otro proceso similar para la fuente de datos de noticias. Después del análisis sentimental realizado al set de noticias en el apartado 3.2. Me he visto obligado a modificar el data set de la figura 8 para transformarlo en un set de datos valiosos, adecuado para introducirlos en los modelos y usarlo en las representaciones que se pueden apreciar a lo largo del trabajo. Dichas modificaciones se aplican desde el apartado 3.3 hasta la sección 3.5.

Se puede concluir que realmente **existe una correlación entre el sentimiento de las noticias y las fluctuaciones de las acciones**. Esto puede llegar a ser alarmante, ya que la manipulación de noticias puede provocar fluctuaciones artificiales en la bolsa de valores. Indicando pues, que ya no solo vale analizar objetivamente la empresa y su futuro como en el pasado, sino también la opinión pública de esta empieza a formar parte de la ecuación.

Ya demostrado la relación entre el sentimiento y la bolsa, se puede usar para mejorar nuestras predicciones. Esto mismo es lo que se intenta hacer en la sección 4. Tras un primer intento de añadir el análisis sentimental como variable para predecir la subida o bajada, se obtuvo resultados realmente insatisfactorios. Tras un análisis minucioso de las variables que se usaron para ejecutar las predicciones, se llegó a una conclusión crucial. Lo que realmente importa no es el sentimiento de cada día, sino el sentimiento relativo a los días pasados. Aplicando esto último al data set (dicho proceso está desarrollado en el apartado), se obtienen resultados alentadores. En la tabla 3 se aprecia una mejoría de las predicciones, llegando a ser de hasta un $16{,}13\,\%$. Por contrapartida, el *recall* disminuye al aumentar la *precision*, algo que era de esperar.

Como conclusión final, puedo atreverme a decir que los resultados son muy reveladores y totalmente exitosos. Se ha demostrado el vínculo, cada vez más notable, entre el sentimiento del mercado y el mercado en sí. Esta acentuación de la importancia de la 'opinión' se da por la revolución tecnológica que se está viviendo en estos momentos. Como sociedad, nos encontramos continuamente conectados al mundo, provocando que cualquier evento o noticia se propague a miles de personas en tan solo segundos. Es totalmente necesario tener en cuenta las noticias relacionadas o que pueden afectar a las compañías, al realizar operaciones en bolsa. Llegando a ser una variable clave en la toma de decisiones, en lo relativo a operar en el mercado de valores.

Referencias

[1] Chen, S.T., Gao, T.H., He, Y.Q. and Jin, Y.F. (2019) Predicting the Stock Price Movement by Social Media Analysis. Journal of Data Analysis and Information Processing, 7, 295-305. https://doi.org/10.4236/jdaip.2019.74017

[2] Jiang Xianyaa, Hai Moa, Li Haifenga, Jiang Xianyaa, Hai Moa, Li Haifen ,
Stock Classification Prediction Based on Spark,
2019.

[3] Selene Yue Xu (UC Berkeley),
Stock Price Forecasting Using Information from Yahoo Finance and Google Trend .

[4] Mazhar Javed Awan, Mohd Shafry Mohd Rahim, Haitham Nobanee, Ashna Munawar ,Awais Yasin and Azlan Mohd Zain ,
Social Media and Stock Market Prediction:A Big Data Approach ,
2020.