CaixaBank Tech Hackathon

La presente documentación describe brevemente el proceso para crear un modelo que pueda predecir si el precio de cierre del IBEX35 será superior o inferior al precio de cierre actual.

Fase 1: Preparación de los datos

El proyecto empieza con la importación de las librerías necesarias, la definición de la ruta y de los ficheros a cargar, la definición de ciertas funciones y la carga de los datasets a utilizar. Una vez cargados, empieza el análisis y tratamiento de los datos disponibles.

Empezando por el dataset "train" se realizan las siguientes acciones clave de transformación:

- Reemplazar valores nulos y quitar el atributo date
- Quitar los registros que tienen 0 volumen de acciones negociadas
- Comprobar la existencia de duplicados
- Comprobar las correlaciones entre atributos

Pasando al dataset de los tweets, aunque se llegó a cierto punto de preparación, no ha sido posible integrar los datos a los demás dataframes. Dentro del tratamiento realizado:

- Se quitaron duplicados, filas vacias y registros que tenían el campo "tweetDate"corrupto
- Se quitaron columnas innecesarias por el análisis
- Se hizo una pequeña limpieza de los tweets con el propósito de poder hacer un análisis de sentimiento, aunque esta última parte no se consiguió.

El tratamiento de los datos de test está hecho de forma parecida a lo del dataset "train" para que se pueda aplicar en ellos el modelo creado y poder entregar los resultados según los requerimientos.

Fase 2: Creación de los modelos de predicción

Pasando en la creación del modelo, se decide construir 2 modelos en 2 versiones:

A) Regresión logística:

i) Un modelo con todas las variables independientes numéricas del dataset "train"

ii) El mismo modelo pero incluyendo solo las variables independientes "Open", "Adj Close", y "Volume".

B) Red neuronal :

i) Un modelo con todas las variables independientes numéricas del dataset "train" (28 nodos en total)

ii) El mismo modelo pero incluyendo solo las variables independientes "Open", "Adj Close", y "Volume".

(20 nodos en total)

Los resultados de bondad de cada uno de los modelos se ven a continuación:

Regresión Logística con todas las variables numéricas independientes

In [386]:
```
# Mirando la bondad del modelo

CR1 = classification_report(y_test, test_predictions1)

print ('\033[1m' + 'Classification Report: Logistic Regression','\033[0m')
print (' ')
print(CR1)
```

Classification Report: Logistic Regression

	precision	recall	f1-score	support
0	0.50	0.57	0.53	720
1	0.51	0.44	0.47	729
accuracy			0.50	1449
macro avg	0.50	0.50	0.50	1449
weighted avg	0.50	0.50	0.50	1449

Regresión Logística solo con las variables independientes “Open”, “Adj Close”, y “Volume”

In [399]:
```
CR2 = classification_report(y_test, test_predictions2)

print ('\033[1m' + 'Classification Report: Logistic Regression','\033[0m')
print (' ')
print(CR1)
```

Classification Report: Logistic Regression

	precision	recall	f1-score	support
0	0.50	0.57	0.53	720
1	0.51	0.44	0.47	729
accuracy			0.50	1449
macro avg	0.50	0.50	0.50	1449
weighted avg	0.50	0.50	0.50	1449

Red neuronal con todas las variables numéricas independientes

In [413]:
```
# Predicción
model_prediction = model.predict(X_test)

#Con valores > 0.5 se define clase 1 => threshold
pred_train = (model.predict(X_train) > 0.5)
pred_test = (model.predict(X_test) > 0.5)


# Comprobar la bondad del modelo
print(classification_report(y_test, pred_test))
```

	precision	recall	f1-score	support
0	0.52	0.36	0.42	706
1	0.53	0.68	0.60	743
accuracy			0.52	1449
macro avg	0.52	0.52	0.51	1449
weighted avg	0.52	0.52	0.51	1449

Red neuronal solo con las variables independientes “Open”, “Adj Close”, y “Volume”

In [427]:
```
# Predicción
model_prediction = model2.predict(X_test)

#Con valores > 0.5 se define clase 1 => threshold
pred_train = (model2.predict(X_train) > 0.5)
pred_test = (model2.predict(X_test) > 0.5)


# Comprobar la bondad del modelo
print(classification_report(y_test, pred_test))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	706
1	1.00	1.00	1.00	743
accuracy			1.00	1449
macro avg	1.00	1.00	1.00	1449
weighted avg	1.00	1.00	1.00	1449

Los resultados indican overfitting

Fase 3: Elección de un modelo para las predicciones

De los modelos anteriores, lo que ha sido elegido para hacer las predicciones fue la red neuronal con todas las variables numéricas independientes porque era el modelo con mayor accuracy sin sobreajuste.