M5_U4_Carlos_Ramirez_Martin

December 15, 2023

[128]:
```
# Configuración de SparkSession
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, FloatType, StringType,␣
 ↪IntegerType, DateType
from pyspark.sql.functions import mean, round, std, max, min
import numpy as np
import pandas as pd
from scipy import stats



spark = SparkSession.builder.master("local").appName("Análisis Exploratorio").
 ↪getOrCreate()
spark
```

[128]:
```
<pyspark.sql.session.SparkSession at 0xffff5a0fc090>
```

[129]:
```
# Importo el fichero localizado en la misma ruta del projecto
data = spark.read.options(inferShema='True', delimiter= ',', header=True).
 ↪csv('work/CSV_stocks_2021.csv')
data.printSchema()
```

```
root
 |-- ticker: string (nullable = true)
 |-- open: string (nullable = true)
 |-- high: string (nullable = true)
 |-- low: string (nullable = true)
 |-- close: string (nullable = true)
 |-- volume: string (nullable = true)
 |-- dividends: string (nullable = true)
 |-- stock splits: string (nullable = true)
 |-- date: string (nullable = true)
 |-- ccy: string (nullable = true)
```

Al importar el fichero e indicarle en la configuración que realizara la inferencia de datos de forma automática **inferSchema=True**, observo como las columnas han sido inferidas a datos de tipo String, cuando realmente nos encontramos con diversos tipos de datos. Por este motivo, volveré a

cargar los datos infiriendo manualmente en el tipo de dato para cada columna.

[130]:
```
# Defino el Schema manualmente
custom_schema = StructType([
    StructField("ticker", StringType(), True),
    StructField("open", FloatType(), True),
    StructField("high", FloatType(), True),
    StructField("low", FloatType(), True),
    StructField("close", FloatType(), True),
    StructField("volume", IntegerType(), True),
    StructField("dividends", FloatType(), True),
    StructField("stock splits", IntegerType(), True),
    StructField("date", DateType(), True),
    StructField("ccy", StringType(), True)])

data = spark.read.schema(custom_schema).option("delimiter", ",").
 ↪option("header", "true").csv('work/CSV_stocks_2021.csv')
data.printSchema()
```

```
root
 |-- ticker: string (nullable = true)
 |-- open: float (nullable = true)
 |-- high: float (nullable = true)
 |-- low: float (nullable = true)
 |-- close: float (nullable = true)
 |-- volume: integer (nullable = true)
 |-- dividends: float (nullable = true)
 |-- stock splits: integer (nullable = true)
 |-- date: date (nullable = true)
 |-- ccy: string (nullable = true)
```

1 Contraste de Hipótesis A/B

En este ejercicio vamos se va a analizar la cotización de cierre para el valor bursatil **HON**. *(Empresa multinaciona estadounidense que opera en varias áreas como la industria aeroespacial, la fabricación de productos para el control de seguridad y automatización).*

Para ello vamos a dividir los 252 periodos en dos partes y compararlos entre si: - Un **Perido Inicial** que inicia el 4 de Enero del 2021 y finaliza el 30 de Junio del 2021. - Un **Periodo Final** que inicia el 01 de Julio del 2021 y finaliza el 30 de Diciembre del 2021

[131]:
```
# Filtro los datos por el valor HON
data_hon = data.filter(data.ticker == 'HON').orderBy('date')
data_hon.show()
print("Número filas:",data_hon.count())
```

```
+------+---------+---------+---------+---------+-------+---------+------------+-
```

```
---------+---+
|ticker|     open|     high|      low|    close| volume|dividends|stock splits|
date|ccy|
+------+---------+---------+---------+---------+-------+---------+------------+-
---------+---+
|   HON|207.45502|209.42142|206.14735|209.12645|1406400|      0.0|
0|2020-12-31|USD|
|   HON| 209.2641|209.43124|202.89297|204.45625|2328900|      0.0|
0|2021-01-04|USD|
|   HON|203.50256|206.65863|203.50256| 204.9577|2172100|      0.0|
0|2021-01-05|USD|
|   HON|205.93106|210.38495|205.71475|208.69385|2747900|      0.0|
0|2021-01-06|USD|
|   HON|209.31328|210.41446|207.25839|209.03798|2057300|      0.0|
0|2021-01-07|USD|
|   HON|209.22478|209.82452|204.18097|206.50131|3278900|      0.0|
0|2021-01-08|USD|
|   HON|204.61356|206.09819|204.33827|204.85936|2938900|      0.0|
0|2021-01-11|USD|
|   HON|204.36778|205.96056|201.81146|205.37065|2498800|      0.0|
0|2021-01-12|USD|
|   HON|204.78072|205.11499| 202.9323|203.54189|2145100|      0.0|
0|2021-01-13|USD|
|   HON|204.54475|206.25552| 203.6992|205.10518|3661500|      0.0|
0|2021-01-14|USD|
|   HON| 204.0433|204.36777|201.75246|202.50952|3887500|      0.0|
0|2021-01-15|USD|
|   HON| 204.7414|205.25266|202.98146|203.28625|2656300|      0.0|
0|2021-01-19|USD|
|   HON|204.42676|205.16415|203.19777|204.58408|2452400|      0.0|
0|2021-01-20|USD|
|   HON|203.38457| 204.3186|201.65414|201.78195|2705100|      0.0|
0|2021-01-21|USD|
|   HON|200.82826|201.02492|198.00648|198.85204|3502700|      0.0|
0|2021-01-22|USD|
|   HON|197.95732|199.15681|196.73816|198.47841|4737700|      0.0|
0|2021-01-25|USD|
|   HON| 200.4153|201.32968|197.60335|  197.682|2201900|      0.0|
0|2021-01-26|USD|
|   HON|194.78159|197.38707|193.22813|196.03026|4108600|      0.0|
0|2021-01-27|USD|
|   HON| 197.2494|201.89995|196.27605|199.43211|3731700|      0.0|
0|2021-01-28|USD|
|   HON|193.20847|197.56404| 191.2814|192.08762|4635100|      0.0|
0|2021-01-29|USD|
+------+---------+---------+---------+---------+-------+---------+------------+-
---------+---+
only showing top 20 rows
```

```
Número filas: 252
```

Como podemos observar, el dataframe inicia en la **fecha 31-12-2020**. COmo este valor está fuera de nuestro análisis *(Se centra en los resultados del 2021)* vamos a proceder a eliminarlo más adelante. Cuando transformemos el datafrade de **Spark a Pandas**.

[132]:
```
# Transformo los datos en dataframe de pandas
pandas_data_hon = data_hon.toPandas()

# Elimino la primera fila ya que corresponde al año 2020
pandas_data_hon = pandas_data_hon.drop(pandas_data_hon.index[0])
print(pandas_data_hon['date'][1])

# Usando Numpy extraigo la columna open y la conversitmos en un array␣
 ↪unidemensional
hon_array = np.array(pandas_data_hon['open']).ravel()
#print(hon_array)
```

```
2021-01-04
```

[133]:
```
# Divido el array en 2 para comparar dos periodos y determinar si se ha␣
 ↪producido un cambio significativo
hon_array_split = np.array_split(hon_array, 2)

hon_array_0 = hon_array_split[0].astype(float)
hon_mean_0 = np.mean(hon_array_0)

hon_array_1 = hon_array_split[1].astype(float)
hon_mean_1 = np.mean(hon_array_1)

print("Media del Periodo Inicial: ", hon_mean_0)
print("Media del Periodo Final: ", hon_mean_1)
print("-"*50)

# Aplicamos la t de student a las 2 muestras
hon_ttest_result = stats.ttest_ind(hon_array_0, hon_array_1, equal_var=False)
p_value = hon_ttest_result.pvalue
print("P-Value:", p_value)
```

```
Media del Periodo Inicial:  213.36190856449187
Media del Periodo Final:  219.47718298339845
--------------------------------------------------
P-Value: 7.464256120088e-07
```

1.1 Resultado del Análisis

El análisis de dos periodos de cotización de las acciones de **Honeywell International Inc.** revela un cambio de **($6.12)**, entre la media del periodo inicial **($213.36)** y la media del periodo final

($219.48). Este aumento en la media sugiere un **incremento en el valor promedio de las acciones de Honeywell durante el segundo periodo**. Sin embargo, para determinar la significancia estadística de este cambio, es esencial considerar el P-Value obtenido a través de la prueba t de Student.

La prueba t de Student confirma la significancia estadística de esta diferencia, evidenciada por un P-Value bajo **(7.46)**. Esta información sugiere un rendimiento positivo en el valor de las acciones durante el segundo periodo, proporcionando a inversores y analistas una base sólida para considerar ajustes en sus estrategias de inversión.