00 INTRODUCCIÓN

Este notebook contiene la **opción B del ejercicio 3**:

- Solo tienes que completar una de las dos opciones (A o B)
- Tienes 3 horas en total. Se aconseja que escojas la opción en los primeros 5-10 minutos, ya que los ejercicios son largos
- Las distintas secciones del ejercicio incluyen una estimación del tiempo que se espera que te lleven, tenlo en cuenta.
- Si te atascas en alguna pregunta o no te resta mucho tiempo, deja indicado por escrito (usando una celda de tipo texto, por ejemplo) cómo la resolverías.
- Se valorarán los comentarios, orden y limpieza del código, no solo su funcionalidad.
- Las preguntas son en su gran mayoría independientes. Al final de las secciones de ingesta y transformación se te proporcionarán los datasets completamente transformados para que puedas seguir con los siguientes ejercicios hayas completado los anteriores o no. Por lo que, de nuevo, evita quedarte atascado en uno en concreto.

0001 Importa librerías

Puedes importar aquí las librarías habituales que creas necesitar, o hacerlo luego según las vayas necesitando.

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

01 Ingesta - 60 mins

Vamos a ingestar los datasets que utilizaremos. El primero requiere bastante trabajo:

0101 - Movimientos Warren Buffet

Primero, vamos a crear un dataset a partir de los datos de esta web (https://dataroma.com/m/m_activity.php?m=BRK&typ=b&L=1&o=a) usando técnicas de web scrapping.

IMPORTANTE

*A lo largo de este ejercicio (0101) encontrarás ayuda en forma de código y comentarios para completarlo empleando las librerías **BeautifulSoup y requests**. Sin embargo, tienes total libertad para realizarlo con otras librerías, o en general, obviando el código propuesto si consideras que no te resulta útil.*

010101 Familiarízate con la web - 5 mins

Comprueba el formato y tipo de información que contiene la web. Verás que:

- en esta url (https://dataroma.com/m/m_activity.php?m=BRK&typ=b&L=1&o=a) puedes encontrar solo las operaciones de compra a lo largo de distintos trimestres
- en esta url (https://dataroma.com/m/m_activity.php?m=BRK&typ=s) puedes encontrar solo las operaciones de venta
- aquí (https://dataroma.com/m/managers.php) hay un directorio con otras firmas (no solo la de Warren Buffet)

010102 Beautiful Soup - 5mins

Usaremos Beautiful Soup para crear un objecto html con la información de la página web. En este caso se muestra con el link de operaciones de compra. Emplea 5 minutos para inspeccionar la "sopa" y familiarizarte con el formato.

In [ ]:
```
import requests
from bs4 import BeautifulSoup
import json
```

In [ ]:
```
# Se envía una petición utilizando un agente
url = "https://dataroma.com/m/m_activity.php?m=BRK&typ=b"
headers = {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:
55.0) Gecko/20100101 Firefox/55.0',}
html = requests.get(url, headers=headers).content
soup = BeautifulSoup(html, 'html.parser')
print(soup.prettify())
```

010103 Extrae valores de la fila i - 5 mins

Fíjate bien en cómo podemos extraer los valores de una fila de la tabla de la web original:

In [ ]:
```
# Nombre de la empresa
soup.find_all(class_="stock")[0]
```

In [ ]:
```
# Hay dos valores de compra por cada fila: el de actividad y el de cambio d
e participaciones
len(soup.find_all(class_="hist")), len(soup.find_all(class_="buy"))
```

In [ ]:
```
# Valores de compra 1: actividad
soup.find_all(class_="buy")[0]
```

In [ ]:
```
# Valores de compra 2: shares
soup.find_all(class_="buy")[1]
```

010104 Dataframe fila_i - 10 mins

Define la función "obtener_df_i_compras" que, a partir de la sopa soup y el número de fila fila_i extrae los valores de la fila_i de la web de compras a un dataframe con una única fila y las columnas:

- "Operation Type": Puede ser igual a "Buy" o "Sell" (el link del que estamos extrayendo la información ahora contiene solo compras)
- "Stock Abv": abreviatura del nombre de la compañía
- "Stock": nombre de la compañía
- "Activity": información sobre la cuantía de la operación
- "Shares": numero de participaciones de la operación

In [ ]:
```
def obtener_df_i_compras(soup, fila_i):
    """
    Define esta función
    """
    return df
```

In [ ]:
```
obtener_df_i_compras(soup, 2)
```

010105 Dataframe df - 10 mins

Crea la función "obtener_df_compras" que extrae todas las filas de la url de compras a un dataframe. Solo necesita como parámetro el identificador del inversor del que queremos saber las compras

Ten en cuenta que la url funciona tal que: url = "https://dataroma.com/m/m_activity.php?m={}&typ=b".format(inversor (https://dataroma.com/m/m_activity.php?m={}&typ=b".format(inversor)).

En el caso de Warren Buffet, su groupo de inversión es: "BRK".

In [ ]:
```
def obtener_df_compras(inversor):
    """
    Define esta función
    """
    return df
```

In [ ]:
```
inversor = "BRK"
obtener_df_compras(inversor)
```

010106 Adaptación a ventas - 10 mins

Adapta el código para crear la función "obtener_df_operaciones", que extrae todas las filas y funciona tanto con la página de compras como con la de ventas. Requiere como parametros, por tanto, tanto el nombre del inversor como el tipo de operación (compra o venta).

In [ ]:
```
def obtener_df_operaciones(inversor, tipo_operacion):
  """
  Define esta función
  """
  return df
```

In [ ]:
```
obtener_df_operaciones(inversor, "Venta")
```

In [ ]:
```
obtener_df_operaciones(inversor, "Compra")
```

010107 Incorpora todas las páginas - 15 mins

Hasta ahora estamos consiguiendo solo la información de la primera página. Esta url https://dataroma.com/m/m_activity.php?m=BRK&typ=b&L=2&o=a (https://dataroma.com/m/m_activity.php?m=BRK&typ=b&L=2&o=a) nos lleva a la segunda página, por ejemplo.

Modifica el código para crear la función obtener_df_operaciones_completo que importe los datos de todas las páginas, no solo de la primera como hasta ahora.

In [ ]:
```
def obtener_df_operaciones_completo(inversor, tipo_operacion):
  """
  Define esta función
  """
  return df
```

In [ ]:
```
obtener_df_operaciones_completo(inversor, "Venta")
```

In [ ]:
```
obtener_df_operaciones_completo(inversor, "Compra")
```

0102 - Importa varios inversores

Importemos ahora los mismos datos de actividad inversora pero para todos los inversores disponibles: ["AKO", "AIM", "AP", "GFT", "psc", "LMM", "oaklx", "fairx", "OCL", "ic","ARFFX", "DJCO", "TGM", "tci", "SA", "DAV", "AC", "CAS", "tp", "abc", "GLRE", "MAVFX", "WP", "AM", "SP", "DODGX", "FE", "FPPTX", "ca", "GC", "CCM", "ENG", "GA", "CM", "aq", "SSHFX", "hcmax", "LUK", "JIM", "EC", "CAAPX", "GLC", "KB", "mc", "HC", "LT", "MPGFX", "LLPFX", "MVALX", "SAM", "PI", "TF", "PC", "DA", "pcm", "FFH", "pzfvx", "OFALX", "RVC", "SEQUX", "BAUPOST", "LPC", "FPACX", "FS", "TA", "MKL", "GR", "T", "TFP", "TWEBX", "VFC", "VA", "vg", "WVALX", "BRK", "cc", "YAM"]

In [ ]:
```
raw_url = "https://raw.githubusercontent.com/JotaBlanco/cnmv_public/main/data/investment_activity.csv"
df_inversores = pd.read_csv(raw_url)
df_inversores = df_inversores.rename(columns={"Share Change": "Shares"})
df_inversores
```

0103 - Importa coches usados

Importa ahora este dataset completamente distinto, que contiene el precio de venta de diversos coches usados según sus características.

In [ ]:
```
raw_url = "https://raw.githubusercontent.com/JotaBlanco/cnmv_public/main/data/coches_usados_esp.csv"
df_coches = pd.read_csv(raw_url, sep=";")
df_coches = df_coches.dropna(subset=["months_old", "power", "kms", "price"]).reset_index(drop=True)
df_coches = df_coches.drop(columns=["ID"])
df_coches
```

02 Transformaciones y limpieza - 35 mins

Realicemos ahora ciertos cambios de formato para obtener un dataset final más útil y claro.

0201 df_inversores

Limpiemos un poco el dataframe df_inversores, que contiene la actividad inversora de diversos fondos.

020101 Columna Date - 10 mins

Crea la columna "date" a partir de "year" y "quarter", asumiendo como fecha el día 15 del mes intermedio del trimestre. Es decir, para el año 2007 y trimestre Q1, la fecha será será 15-02-2007. Para el año 2007 y trimestre Q2, la fecha será 15-05-2007. Haz que esta nueva columna sea la primera del dataframe.

In [ ]:
```

```

020102 Limpia Stock - 5mins

La columna stock tiene ciertos espacios y un guión en (parece) todas sus entradas. Comprueba primero que es así (que todas las filas empiezan por un guión), y si es así, elimina dichos espacios y guiones sobrantes.

In [ ]:
```

```

020103 Limpia Share Change - 5 mins

La columna share change parece contener números, pero presentados como cadenas de caracteres. Comprueba primero, que efectivamente todos los valores son enteros, y luego conviértelos a un formato numérico apropiado.

In [ ]:
```

```

020104 Transforma Activity - 15 mins

Comprueba la columna "Activity". El formato no es muy aprovechable tal cual se nos da. Modifícala de manera que:

- una venta total sea un valor de 0
- una venta del 17% sea 0.83
- una compra total sea un valor de 1
- una compra de un 20% más de acciones sea un valor de 1.2

In [ ]:

In [ ]:

020105 Comodín df_inversores

Importa df_inversores ya limpio para asegurarnos seguir trabajando con los mismos datos de aquí en adelante.

In [ ]:
```
df_inversores = pd.read_csv("https://raw.githubusercontent.com/JotaBlanco/cnmv_public/main/data/CLEAN_df_inversores.csv")
df_inversores
```

03 Análisis - 30 mins

0301 Distribución del precio - 5 mins

Estudia la variable price de df_coches: extrae los estadísticos básicos (media, cuantiles, mediana, etc.) y comenta la distribución brevemente.

In [ ]:

Tus comentarios (una o dos frases es suficiente):

--

--

0302 Diagrama de dispersión - 10 mins

Muestra la asociación entre los siguientes pares de variables:

- potencia y precio
- meses de antigüedad y precio
- kms y precio

Para cada caso, hazlo:

- de manera visual con un diagrama de dispersión
- de manera cuantitativa calculando el coeficiente de correlación de pearson

Comenta los resultados.

In [ ]:

Comenta la asociación entre variables brevemente (una o dos frases es suficiente):

—

—

0303 Responde a las siguientes preguntas - 15 mins

Volvemos al dataframe sobre actividad inversora: df_inversores.

030301 ¿Qué año ha habido más operaciones (sea de compra o venta)?

In [ ]:

030302 ¿Qué inversor hizo más operaciones en el año 2015?

In [ ]:

030303 Genera una lista mostrando para cada año el inversor que vendió más acciones ("Shares") ese año.

In [ ]:

030304 Inversores con caracter comprandor/vendedor

Muestra los 3 inversores con un caracter más marcadamente comprador y los 3 con un carácter más vendedor en la década del 2010. Calculamos el carácter comprador/vendedor como el ratio de acciones compradas y vendidas.

In [ ]:

4 Modelo predictivo (40 mins)

Crea un modelo predictivo simple que prediga el precio (variable "price") de un vehículo listado dadas sus características. Emplea el dataframe df_coches, usando las variables, algoritmos y métodos que prefieras. Ten en cuenta que el dataset no está limpio aún (continue nulos, variables tipo texto, etc.). Si hay pasos que no te da tiempo a ejecutar, déjalos descritos.

In [ ]:
```
df_coches.head()
```

In [ ]: