Porque me pide cambiar estos datos si pongo close y dates y no me aparece como etiquetas reales:

# Importamos las librerías que vamos a necesitar

import numpy as np

import pandas as pd

import os

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, classification\_report

data\_url = r"C:\Users\Ferran\OneDrive - es.logicalis.com\Escritorio\hkgn\hkgn pf\Binance\_BTCUSDT\_d (1).csv"

if os.path.exists(data\_url):

raw\_df = pd.read\_csv(data\_url, sep=";")

# Resto del código...

else:

print("El archivo no existe en la ubicación especificada.")

print(raw\_df.head(10))

print(raw\_df.tail(10))

import os

import pandas as pd

data\_url = r"C:\Users\Ferran\OneDrive - es.logicalis.com\Escritorio\hkgn\hkgn pf\Binance\_BTCUSDT\_d (1).csv"

if os.path.exists(data\_url):

raw\_df = pd.read\_csv(data\_url, sep=";")

# Resto del código...

# Ahora puedes llamar al método info() en la variable 'raw\_df'

raw\_df.info()

else:

print("El archivo no existe en la ubicación especificada.")

raw\_df = pd.read\_csv(data\_url, sep=";")

num\_filas, num\_columnas = raw\_df.shape

print("Número de filas:", num\_filas)

print("Número de columnas:", num\_columnas)

data = pd.read\_csv(data\_url, sep=";")

resumen\_estadistico = data.describe()

print(resumen\_estadistico)

from sklearn import metrics

y\_pred = reg.predict(X\_train)

print('R^2:',metrics.r2\_score(y\_train, y\_pred))

print('Adjusted R^2:',1 - (1-metrics.r2\_score(y\_train, y\_pred))\*(len(y\_train)-1)/(len(y\_train)-X\_train.shape[1]-1))

print('MAE:',metrics.mean\_absolute\_error(y\_train, y\_pred))

print('MSE:',metrics.mean\_squared\_error(y\_train, y\_pred))

print('RMSE:',np.sqrt(metrics.mean\_squared\_error(y\_train, y\_pred)))

print('R^2:', r2)

print('Adjusted R^2:', adj\_r2)

print('MAE:', mae)

print('MSE:', mse)

print('RMSE:', rmse)

plt.hist(y\_train, bins=20, range=(0, 2000))

# Agregar etiquetas de eje y título

plt.xlabel('Date')

plt.ylabel('Close')

# Mostrar el gráfico

plt.show()

print(data['Close'].unique())