import sklearn

from sklearn.naive\_bayes import GaussianNB

import pandas as pd

df = pd.read\_csv('btc.csv')

Primero verifico si datos sin aleatorios con matriz de covarianza

Luego elimino

df = df.drop(['DaySpread','Apertura','Close-Open'], axis=1)

Normalizo

df['Subio-Bajo'] = df['Subio-Bajo']/100

df['Dias7'] = df['Dias7']/100

df['Dias14'] = df['Dias14']/100

df['Tomorrow'] = df['Tomorrow']/100

Si todo numérico

df[df['Dias14'].apply(lambda x: str(x).isdigit())]

Si muchos datos un sample

df2 = df.sample(n=1000)

Separe datos

import numpy as np

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

X1 = df2.drop('Tomorrow',axis='columns')

Y1 = testLabels = df2['Tomorrow'].astype(np.float32)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X1, Y1, test\_size=0.2, random\_state=0)

Naive Bayes

import sklearn

from sklearn.naive\_bayes import GaussianNB

Model\_gnb = GaussianNB()

model = Model\_gnb.fit(x\_train, y\_train)

ERROR:

if not \_unique\_labels: -->

raise ValueError("Unknown label type: %s" % repr(ys))

Razón

This data set has a continuous target variable.

GNB is a classification method, not a regression method.

You needs to be discrete classes, not a continuous variable.

Cambie a regresión

import pandas

from sklearn import linear\_model

regr = linear\_model.LinearRegression()

regr.fit(x\_train, y\_train)

pred = regr.predict(x\_test)

Comparar

regr.score(x\_test,y\_test)

test = pd.DataFrame([[

   -.00271408654,  -.04918215637,  -.05580614541

]])

regr.predict(test)