# we will talk about Apple stock data

import pandas as pd

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

from sklearn.metrics import confusion\_matrix

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

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

from sklearn import preprocessing

import matplotlib.pyplot as plt

import seaborn as sns

apple\_data = pd.read\_csv('AAPL.csv')

print('--------------------------data\_test---------------------')

print(apple\_data)

print(apple\_data.shape)

print('---------data.coulmns----------')

print(apple\_data.head(0))

print('---------data.describ----------')

print(apple\_data.describe())

print('---------data.duplicated----------')

print(apple\_data.duplicated().sum())

print('---------empty cell----------')

print(apple\_data.isnull().sum())

print('--------------------------data\_train---------------------')

# to draw the data

sns.heatmap(apple\_data.corr(), annot=True)

plt.show()

plt.xlabel('Open')

plt.scatter(apple\_data['Open'], apple\_data['Close'])

plt.ylabel('Close')

plt.show()

x = apple\_data.iloc[:, :-1].values

y = apple\_data.iloc[:, -1].values

scaler = preprocessing.MinMaxScaler()

scaled\_data\_train = scaler.fit\_transform(x)

scaled\_data\_train = pd.DataFrame(scaled\_data\_train, columns=apple\_data.columns[:-1])

print(f'the data \_scaled= \n {scaled\_data\_train}')

# we imported the train test split method to train the module to the data

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(x, y, test\_size=0.2)

print('------------------data before val--------------------')

print(f'X\_train =\n\n{X\_train}')

print(f'X\_test=\n\n {X\_test}')

print(f'Y\_train=\n \n{Y\_train}')

print(f'Y\_test= \n\n{Y\_test}')

# we used the logistic regression

lg = LogisticRegression()

lg.fit(X\_train, Y\_train)

Y\_pred = lg.predict(X\_test)

co = confusion\_matrix(Y\_test, Y\_pred)

print('confusion\_matrix to LogisticRegression=\n', co)

sns.heatmap(co, annot=True)

plt.show()

re = recall\_score(Y\_test, Y\_pred, average='micro')

print('recall\_score to LogisticRegression', re)

f1 = f1\_score(Y\_test, Y\_pred, average='micro')

print('f1\_score to LogisticRegression=', f1)

clf = SVC(kernel='linear')

clf.fit(X\_train, Y\_train)

Y\_pred2 = clf.predict(X\_test)

co\_2 = confusion\_matrix(Y\_test, Y\_pred)

print('confusion\_matrix to SVC=\n', co)

sns.heatmap(co\_2, annot=True)

plt.show()

re2 = recall\_score(Y\_test, Y\_pred, average='micro')

print('recall\_score to svc', re2)

f1\_2 = f1\_score(Y\_test, Y\_pred, average='micro')

print('f1\_score to svc=', f1\_2)