**# naive\_bayes.py**

df = pd.read\_csv('./Final\_Refined\_Encoded\_Normalysed.csv')

df0 = df[['SIM', 'CPU', 'GPU', 'memory\_card', 'weight\_g', 'screen\_to\_body\_ratio', \

'primary\_camera', 'internal\_memory', 'Thickness',\

'display\_size', 'OS', 'radio', 'RAM', 'EDGE'\

]]

features = df0.values[:,:]

df1 = df[['Price']]

target = df1.values[:,0]

features\_train, features\_test, target\_train, target\_test = train\_test\_split(features,

target, test\_size = 0.3, random\_state = 10)

clf = BernoulliNB()

clf.fit(features\_train, target\_train)

target\_pred = clf.predict(features\_test)

accuracy\_rate = accuracy\_score(target\_test, target\_pred, normalize = True)

print("BernoulliNB Accuracy Rate:", accuracy\_rate)

clf = MultinomialNB()

clf.fit(features\_train, target\_train)

target\_pred = clf.predict(features\_test)

accuracy\_rate = accuracy\_score(target\_test, target\_pred, normalize = True)

print("MultinomialNB Accuracy Rate:", accuracy\_rate)

clf = GaussianNB()

clf.fit(features\_train, target\_train)

target\_pred = clf.predict(features\_test)

accuracy\_rate = accuracy\_score(target\_test, target\_pred, normalize = True)

print("GaussianNB Accuracy Rate:", accuracy\_rate)

fo = open("./nb\_result.txt", "w")

fo.write(str(accuracy\_rate))

fo.close()

**Output**

➜ python3 naive\_bayes.py

BernoulliNB Accuracy Rate: 0.44503664223850764

MultinomialNB Accuracy Rate: 0.24783477681545638

GaussianNB Accuracy Rate: 0.47168554297135246