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
```