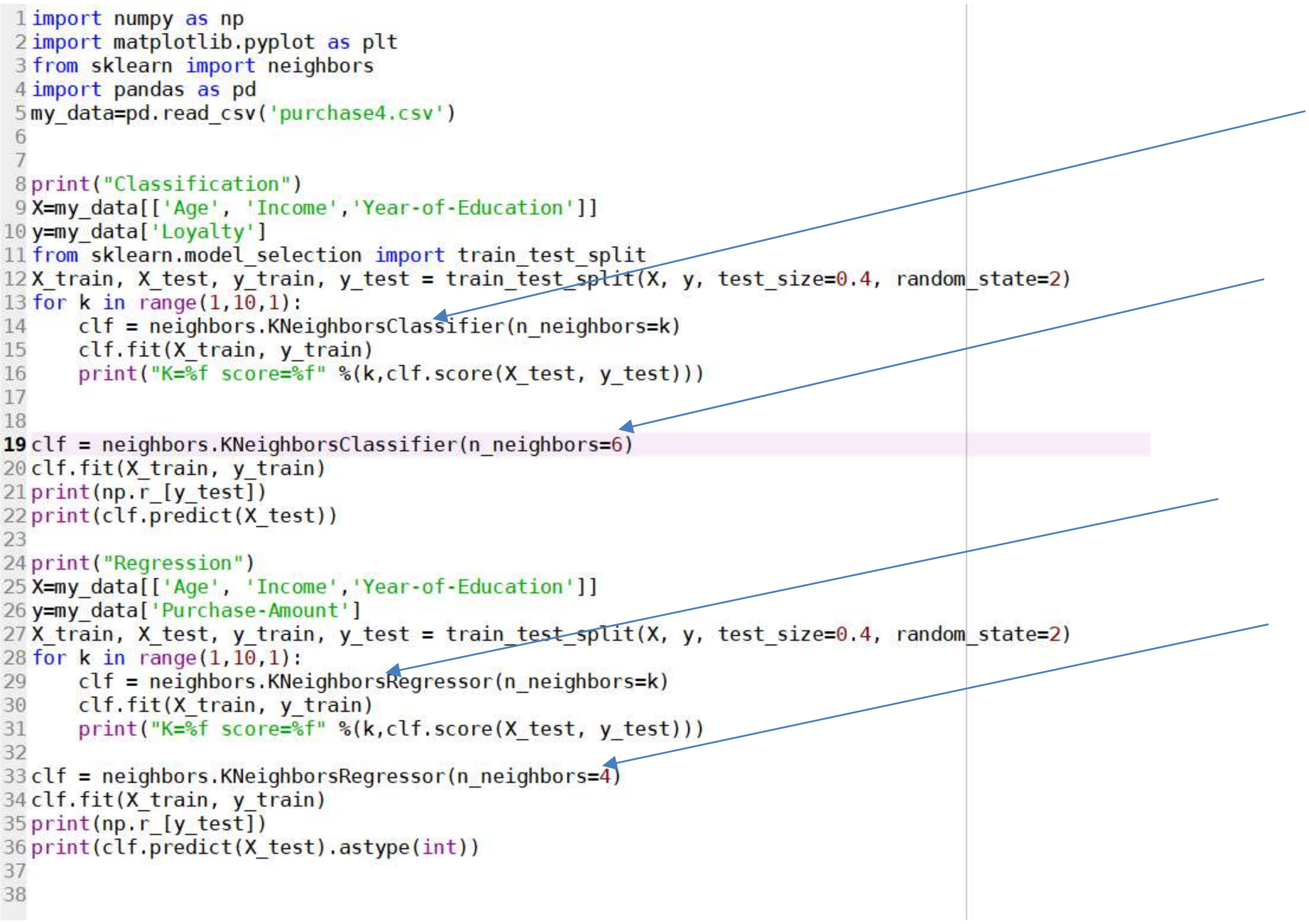
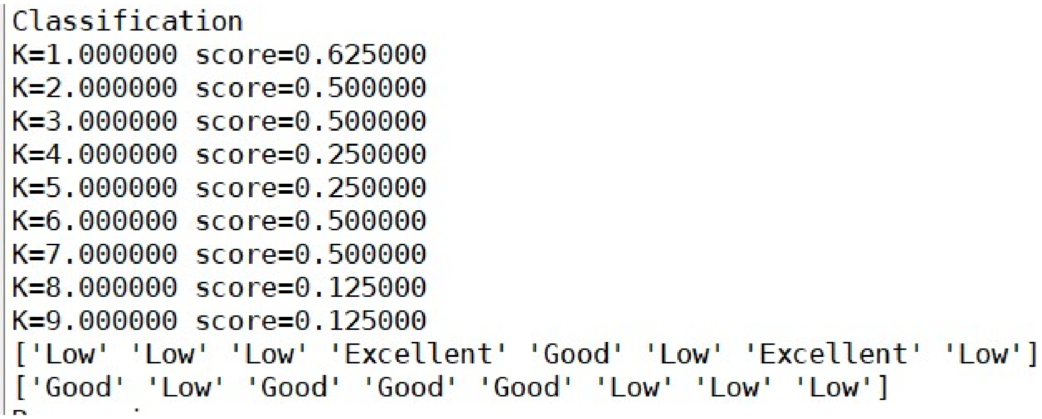
The python code run K-nn algorithm and select the best K



```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn import neighbors
import pandas as pd
my_data=pd.read_csv('purchase4.csv')


print("Classification")
X=my_data[['Age', 'Income','Year-of-Education']]
y=my_data['Loyalty']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=2)
for k in range(1,10,1):
    clf = neighbors.KNeighborsClassifier(n_neighbors=k)
    clf.fit(X_train, y_train)
    print("K=%f score=%f" %(k,clf.score(X_test, y_test)))


clf = neighbors.KNeighborsClassifier(n_neighbors=6)
clf.fit(X_train, y_train)
print(np.r_[y_test])
print(clf.predict(X_test))

print("Regression")
X=my_data[['Age', 'Income','Year-of-Education']]
y=my_data['Purchase-Amount']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=2)
for k in range(1,10,1):
    clf = neighbors.KNeighborsRegressor(n_neighbors=k)
    clf.fit(X_train, y_train)
    print("K=%f score=%f" %(k,clf.score(X_test, y_test)))

clf = neighbors.KNeighborsRegressor(n_neighbors=4)
clf.fit(X_train, y_train)
print(np.r_[y_test])
print(clf.predict(X_test).astype(int))
```

The first part predicts Categorical Variable ‘Loyalty’ using KNeighborsClassifier

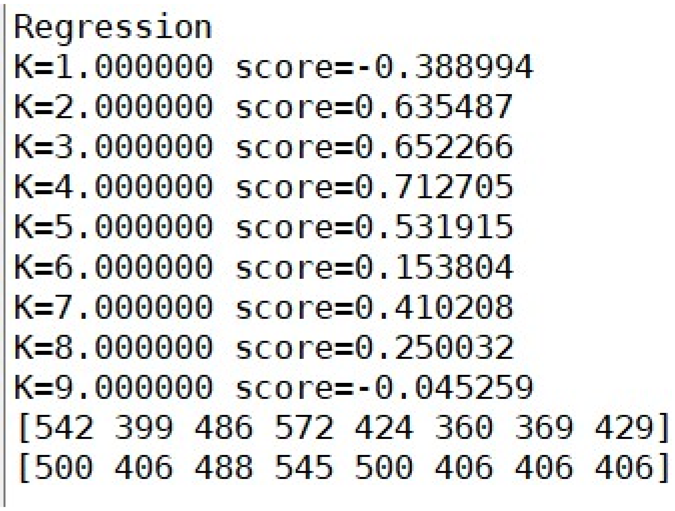


```
Classification
K=1.000000 score=0.625000
K=2.000000 score=0.500000
K=3.000000 score=0.500000
K=4.000000 score=0.250000
K=5.000000 score=0.250000
K=6.000000 score=0.500000
K=7.000000 score=0.500000
K=8.000000 score=0.125000
K=9.000000 score=0.125000
['Low' 'Low' 'Low' 'Excellent' 'Good' 'Low' 'Excellent' 'Low']
['Good' 'Low' 'Good' 'Good' 'Good' 'Low' 'Low' 'Low']
```

We select K=6 as the best K. We print the true values and predicted values.

Here we see this prediction is not good.

The second part predicts numerical Variable ‘Purchase-Amount’ using KNeighborsRegressor



```
Regression
K=1.000000 score=-0.388994
K=2.000000 score=0.635487
K=3.000000 score=0.652266
K=4.000000 score=0.712705
K=5.000000 score=0.531915
K=6.000000 score=0.153804
K=7.000000 score=0.410208
K=8.000000 score=0.250032
K=9.000000 score=-0.045259
[542 399 486 572 424 360 369 429]
[500 406 488 545 500 406 406 406]
```

It is easy to see K=4 is the best K. We print the true values and predicted values.