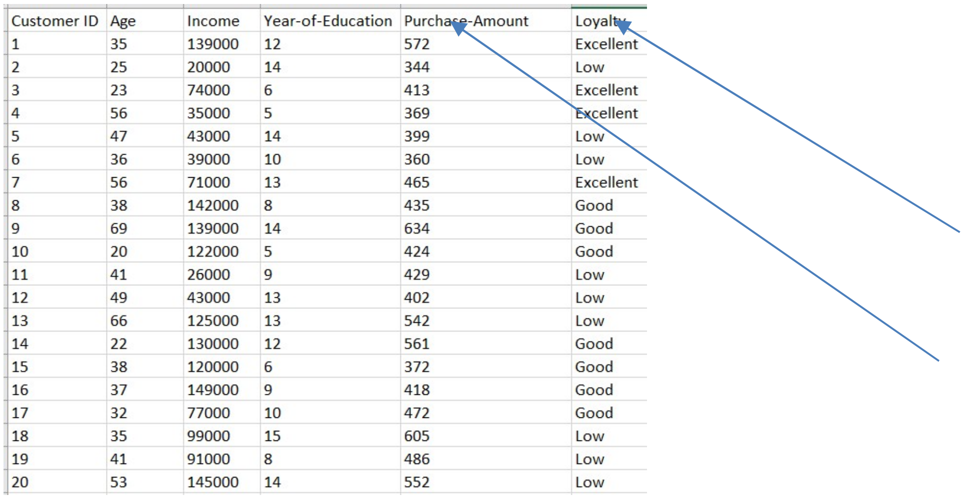
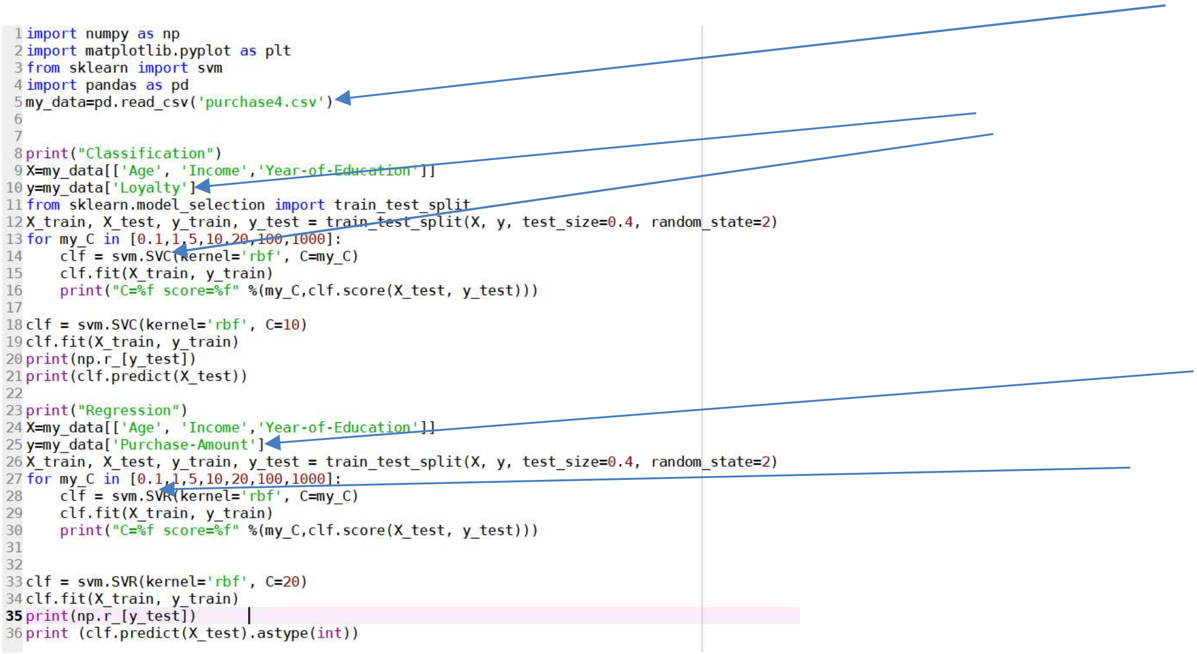
If your dependent variable is categorical, you run svm classification and you choose “SVC”

This is the file “purchase4.csv”



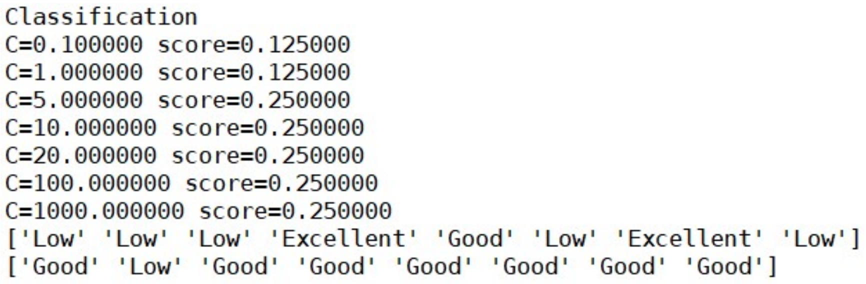
Customer ID	Age	Income	Year-of-Education	Purchase-Amount	Loyalty
1	35	139000	12	572	Excellent
2	25	20000	14	344	Low
3	23	74000	6	413	Excellent
4	56	35000	5	369	Excellent
5	47	43000	14	399	Low
6	36	39000	10	360	Low
7	56	71000	13	465	Excellent
8	38	142000	8	435	Good
9	69	139000	14	634	Good
10	20	122000	5	424	Good
11	41	26000	9	429	Low
12	49	43000	13	402	Low
13	66	125000	13	542	Low
14	22	130000	12	561	Good
15	38	120000	6	372	Good
16	37	149000	9	418	Good
17	32	77000	10	472	Good
18	35	99000	15	605	Low
19	41	91000	8	486	Low
20	53	145000	14	552	Low

If your dependent variable is numerical, you run svm regression and you choose “SVR”



```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
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 my_C in [0.1,1,5,10,20,100,1000]:
    clf = svm.SVC(kernel='rbf', C=my_C)
    clf.fit(X_train, y_train)
    print("C=%f score=%f" %(my_C,clf.score(X_test, y_test)))

clf = svm.SVC(kernel='rbf', C=10)
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 my_C in [0.1,1,5,10,20,100,1000]:
    clf = svm.SVR(kernel='rbf', C=my_C)
    clf.fit(X_train, y_train)
    print("C=%f score=%f" %(my_C,clf.score(X_test, y_test)))


clf = svm.SVR(kernel='rbf', C=20)
clf.fit(X_train, y_train)
print(np.r_[y_test])
print (clf.predict(X_test).astype(int))
```

Here the first block of the code is to predict “Loyalty” which is categorical, so we use SVC.

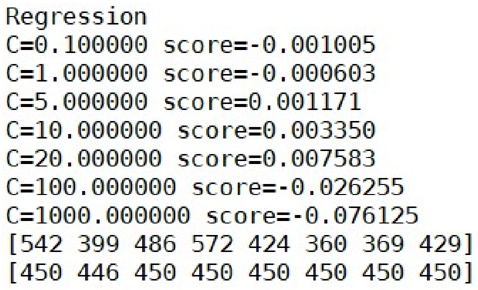


```
Classification
C=0.100000 score=0.125000
C=1.000000 score=0.125000
C=5.000000 score=0.250000
C=10.000000 score=0.250000
C=20.000000 score=0.250000
C=100.000000 score=0.250000
C=1000.000000 score=0.250000
['Low' 'Low' 'Low' 'Excellent' 'Good' 'Low' 'Excellent' 'Low']
['Good' 'Low' 'Good' 'Good' 'Good' 'Good' 'Good' 'Good']
```

According to the comparisons of all Cs, it seems C=10 is a good choice.

So we run C=10 and print out the true values and predicted values.

The second block of the code is to predict “Purchase-Amount” which is numerical, so we use SVR.



```
Regression
C=0.100000 score=-0.001005
C=1.000000 score=-0.000603
C=5.000000 score=0.001171
C=10.000000 score=0.003350
C=20.000000 score=0.007583
C=100.000000 score=-0.026255
C=1000.000000 score=-0.076125
[542 399 486 572 424 360 369 429]
[450 446 450 450 450 450 450 450]
```

According to the comparisons of all Cs, it seems C=20 is a good choice.

So we run C=20 and print out the true values and predicted values.