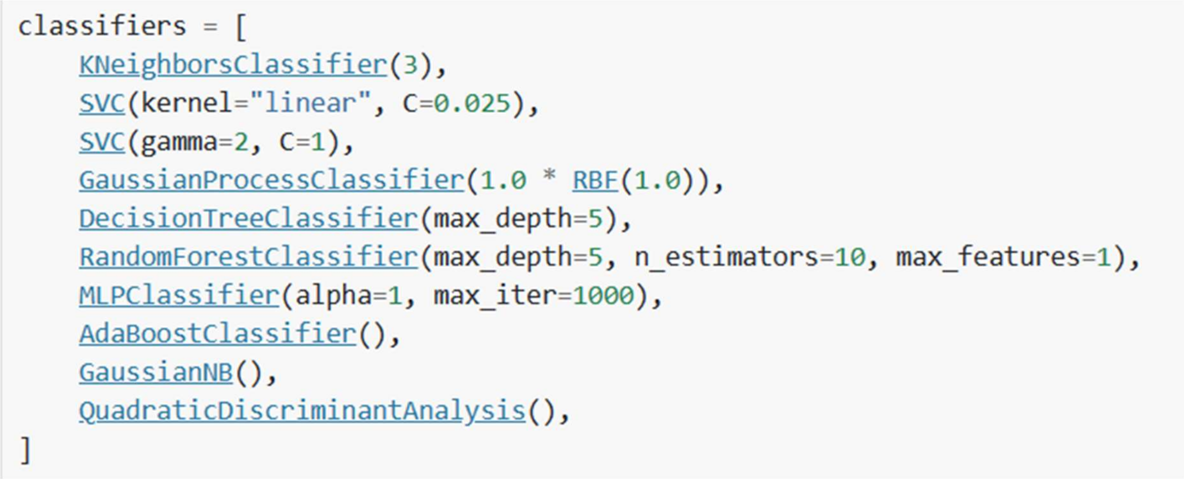
IRIS CLASSIFIER PROJECT

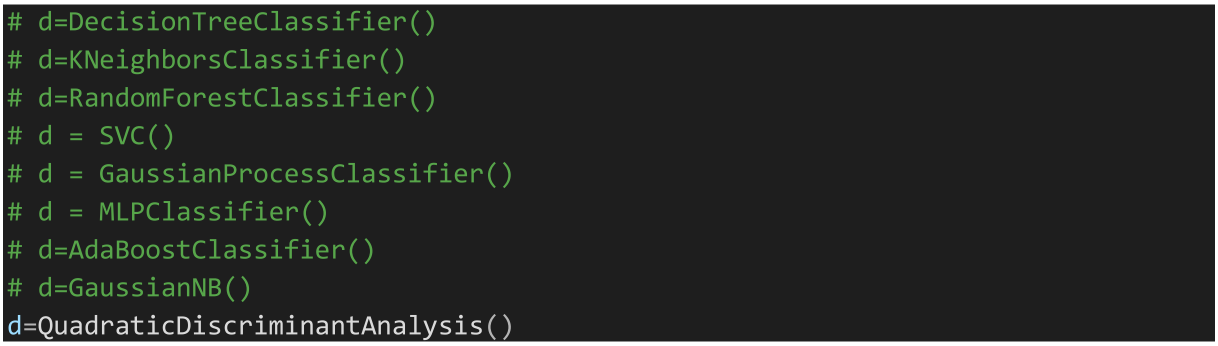
CLASSIFIERS IN SKLEARN :



```
classifiers = [
    KNeighborsClassifier(3),
    SVC(kernel="linear", C=0.025),
    SVC(gamma=2, C=1),
    GaussianProcessClassifier(1.0 * RBF(1.0)),
    DecisionTreeClassifier(max_depth=5),
    RandomForestClassifier(max_depth=5, n_estimators=10, max_features=1),
    MLPClassifier(alpha=1, max_iter=1000),
    AdaBoostClassifier(),
    GaussianNB(),
    QuadraticDiscriminantAnalysis(),
]
```

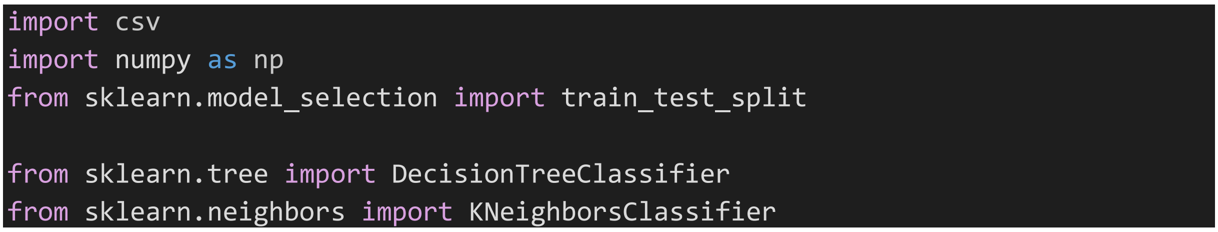
I have checked the classification with each of these classifiers:

In the following code to run for each classifier remove the “#” and run it for the corresponding classifier.



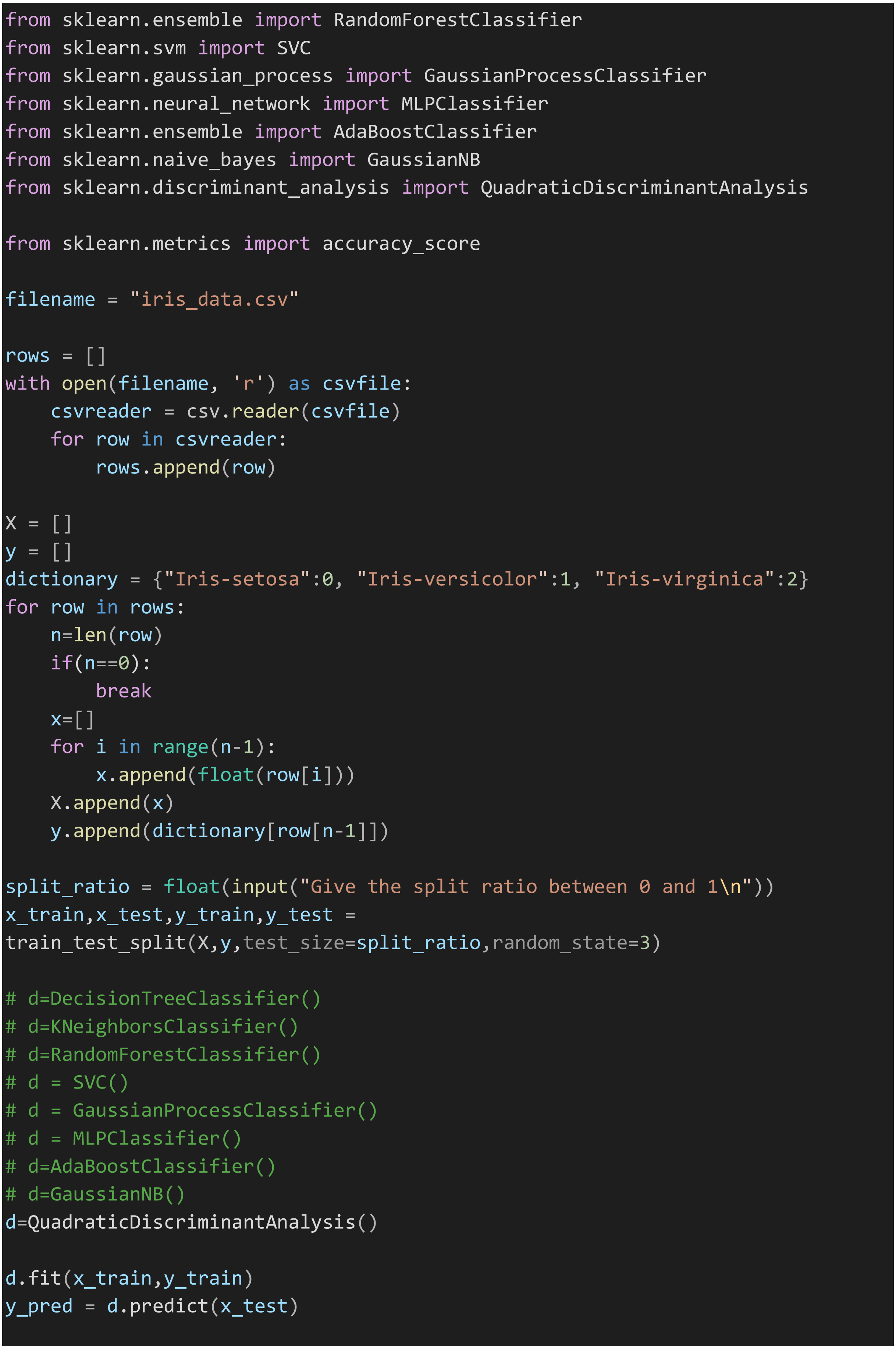
```
# d=DecisionTreeClassifier()
# d=KNeighborsClassifier()
# d=RandomForestClassifier()
# d = SVC()
# d = GaussianProcessClassifier()
# d = MLPClassifier()
# d=AdaBoostClassifier()
# d=GaussianNB()
d=QuadraticDiscriminantAnalysis()
```

CODE :



```
import csv
import numpy as np
from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
```



```
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis

from sklearn.metrics import accuracy_score

filename = "iris_data.csv"

rows = []
with open(filename, 'r') as csvfile:
    csvreader = csv.reader(csvfile)
    for row in csvreader:
        rows.append(row)

X = []
y = []
dictionary = {"Iris-setosa":0, "Iris-versicolor":1, "Iris-virginica":2}
for row in rows:
    n=len(row)
    if(n==0):
        break
    x=[]
    for i in range(n-1):
        x.append(float(row[i]))
    X.append(x)
    y.append(dictionary[row[n-1]])

split_ratio = float(input("Give the split ratio between 0 and 1\n"))
x_train,x_test,y_train,y_test =
train_test_split(X,y,test_size=split_ratio,random_state=3)

# d=DecisionTreeClassifier()
# d=KNeighborsClassifier()
# d=RandomForestClassifier()
# d = SVC()
# d = GaussianProcessClassifier()
# d = MLPClassifier()
# d=AdaBoostClassifier()
# d=GaussianNB()
d=QuadraticDiscriminantAnalysis()

d.fit(x_train,y_train)
y_pred = d.predict(x_test)
```

```
print("The accuracy is: ",accuracy_score(y_pred,y_test)*100)
```

After running for so many split ratios from 0 to 1 , I observed that irrespective of the split ratios value running more than once is giving good accuracy most of the times . This is because the data gets splitted randomly.

But when the split ratio is <=0.5 that is when train data has more data compared to test data the results are better this is because when there is more train data the model will have more information and it could classify the new data more accurately.