Lab Activity 2: KNN Classification using Scikit-learn

Lab report

Submitted to:
Dr Muhammad Abul Hasan
Assistant professor
School of Science & Engineering

Submitted by:
Apurba Kumar
182014045

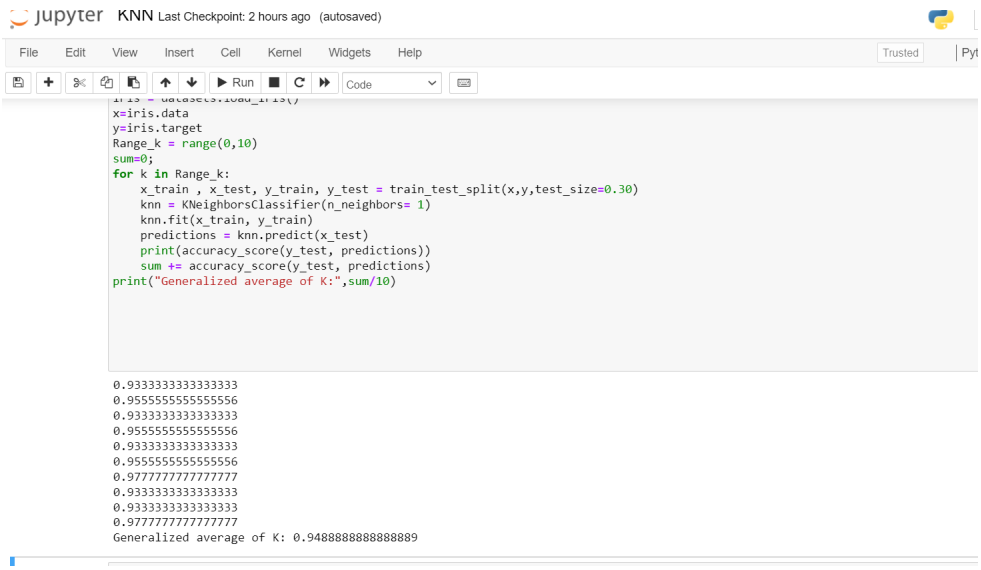
Submitted On:
17-11-2021

Finding the best K based which takes 10 accuracy for each K and calculate their average performance for finding the best K:

This code calculates the generalized average of K:

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
iris = datasets.load_iris()
x=iris.data
y=iris.target
Range_k = range(0,10)
sum=0;
for k in Range_k:
    x_train , x_test, y_train, y_test = train_test_split(x,y,test_size=0.30)
    knn = KNeighborsClassifier(n_neighbors= 1)
    knn.fit(x_train, y_train)
    predictions = knn.predict(x_test)
    print(accuracy_score(y_test, predictions))
    sum += accuracy_score(y_test, predictions)
print("Generalized average of K:",sum/10)
```

In this program, we can easily find the generalized average of k, here our test ratio is fixed and when we run this code , for each loop datasets were shuffled which



generates different accuracy for K .

For,

K =1 and Test size= 0.30

Generalized average accuracy =0.94

K =3 and Test size= 0.30

Generalized average accuracy =0.95

K =5 and Test size= 0.30

Generalized average accuracy =0.94

K =7 and Test size= 0.30

Generalized average accuracy =0.96

K =9 and Test size= 0.30

Generalized average accuracy =0.95

K =11 and Test size= 0.30

Generalized average accuracy =0.97

K =13 and Test size= 0.30

Generalized average accuracy =0.96

K =15 and Test size= 0.30

Generalized average accuracy =0.962

K =17 and Test size= 0.30

Generalized average accuracy =0.97

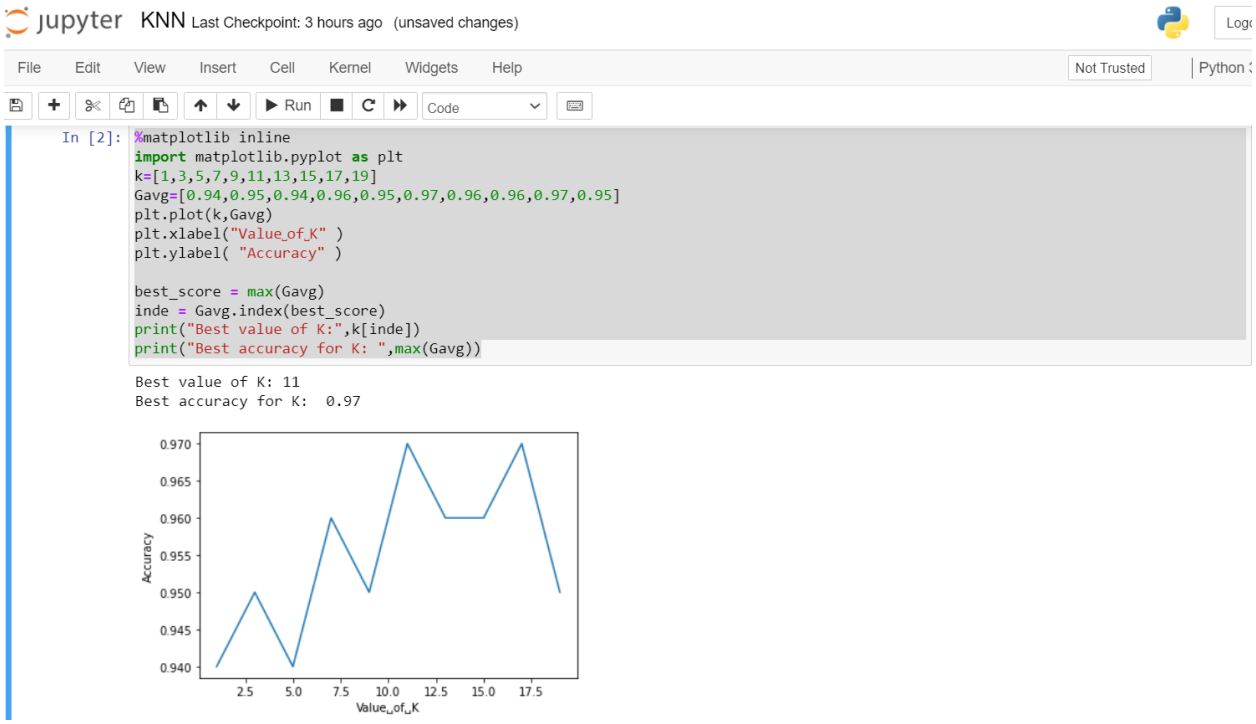
K =19 and Test size= 0.30
Generalized average accuracy =0.95

In this following program we can easily find the best value of K which known as hyperparameter:

```
%matplotlib inline
import matplotlib.pyplot as plt
k=[1,3,5,7,9,11,13,15,17,19]
Gavg=[0.94,0.95,0.94,0.96,0.95,0.97,0.96,0.96,0.97,0.95]
plt.plot(k,Gavg)
plt.xlabel("Value␣of␣K" )
plt.ylabel( "Accuracy" )

best_score = max(Gavg)
inde = Gavg.index(best_score)
print("Best value of K:",k[inde])
print("Best accuracy for K: ",max(Gavg))
```

Output:



The best training and test set ratio in classifying Irish flowers:

This code calculates the generalized average of ratio of datasets split:

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
iris = datasets.load_iris()
x=iris.data
y=iris.target
Range_k = range(0,10)
sum=0;
for k in Range_k:
    x_train , x_test, y_train, y_test = train_test_split(x,y,test_size=0.15)
    knn = KNeighborsClassifier(n_neighbors= 11)
    knn.fit(x_train, y_train)
```

```
    predictions = knn.predict(x_test)
    print(accuracy_score(y_test, predictions))
    sum += accuracy_score(y_test, predictions)
print("Generalized average of Ratio test size  of split dataset:",sum/10)
```

In [94]:
```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
iris = datasets.load_iris()
x=iris.data
y=iris.target
Range_k = range(0,10)
sum=0;
for k in Range_k:
    x_train , x_test, y_train, y_test = train_test_split(x,y,test_size=0.15)
    knn = KNeighborsClassifier(n_neighbors= 11)
    knn.fit(x_train, y_train)
    predictions = knn.predict(x_test)
    print(accuracy_score(y_test, predictions))
    sum += accuracy_score(y_test, predictions)
print("Generalized average of Ratio test size of split dataset:",sum/10)
```

```
0.9130434782608695
0.9130434782608695
1.0
1.0
0.9565217391304348
1.0
0.9130434782608695
1.0
1.0
1.0
Generalized average of Ratio test size of split dataset: 0.9695652173913043
```

In this program, we can easily find the generalized average of the ratio test size of split datasets , here our best value of k =11 is fixed and when we run this code , for each loop datasets were shuffled which generates different accuracy for test size .
For,
K =11 and Test size= 0.10
Generalized average accuracy =0.98
K =11 and Test size= 0.15
Generalized average accuracy =0.96
K =11 and Test size= 0.20
Generalized average accuracy =0.97
K =11 and Test size= 0.25
Generalized average accuracy =0.95
K =11 and Test size= 0.30
Generalized average accuracy =0.97
K =11 and Test size= 0.35
Generalized average accuracy =0.95
K =11 and Test size= 0.40

```
Generalized average accuracy =0.97
 K =11 and Test size= 0.45
Generalized average accuracy =0.95
 K =11 and Test size= 0.50
Generalized average accuracy =0.96
 K =11 and Test size= 0.55
Generalized average accuracy =0.94
```

In this following program we can easily find the best ratio of test data which known as hyperparameter ratio of dataset split:

```
%matplotlib inline
import matplotlib.pyplot as plt
Ratio_test_size=[0.10,0.15,0.20,0.25,0.30,0.35,0.40,0.45,0.50,0.55]
Gavg=[0.98,0.96,0.97,0.95,0.97,0.95,0.97,0.95,0.96,0.94]
plt.plot(Ratio_test_size,Gavg)
plt.xlabel("Ratio of test dataset" )
plt.ylabel( "Accuracy" )

best_score = max(Gavg)
inde = Gavg.index(best_score)
print("Best Ratio of dataset split:",Ratio_test_size[inde])
print("Best accuracy for K: ",max(Gavg))
```

jupyter KNN Last Checkpoint: 3 hours ago (unsaved changes)

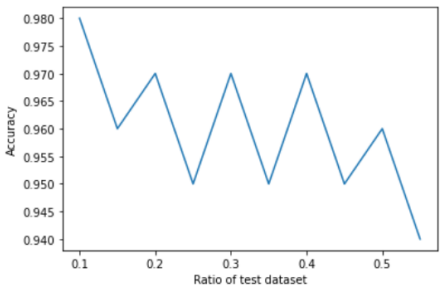
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code

In [5]:
```
%matplotlib inline
import matplotlib.pyplot as plt
Ratio_test_size=[0.10,0.15,0.20,0.25,0.30,0.35,0.40,0.45,0.50,0.55]
Gavg=[0.98,0.96,0.97,0.95,0.97,0.95,0.97,0.95,0.96,0.94]
plt.plot(Ratio_test_size,Gavg)
plt.xlabel("Ratio of test dataset" )
plt.ylabel( "Accuracy" )

best_score = max(Gavg)
inde = Gavg.index(best_score)
print("Best Ratio of dataset split:",Ratio_test_size[inde])
print("Best accuracy for K: ",max(Gavg))
```

```
Best Ratio of dataset split: 0.1
Best accuracy for K:  0.98
```



Finally we find the best value of k is 11 and the best ratio of test size is 0.10 for the iris dataset.