LetsGrowMore

Author-Arnab Pal

Task(1)-Iris Flowers Classification

Language-Python

Software-Jupyter Notebook(Colaboratory)

Importing Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import datasets
```

Inserting Datasets

In [2]:

```
hi=pd.read_csv("iris.csv")
```

Head of Data Set

In [20]:

```
hi.head()
```

Out[20]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Shape of dataset

In [21]:

```
hi.shape
```

Out[21]:

```
(150, 6)
```

Split the Datasets

In [4]:

```
from sklearn.model_selection import train_test_split
```

In [6]:

```
x=hi.drop("Species",axis=1)
```

In [7]:

```
x=x.drop("Id",axis=1)
```

In [8]:

```
y=hi['Species']
```

In [9]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)
```

Import library for KNN Algo

In [10]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [11]:

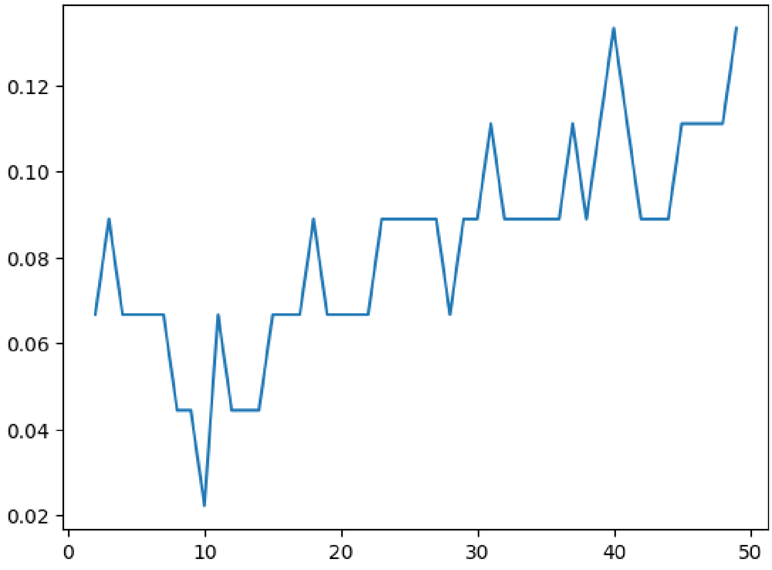
```
error_rate=[]
for i in range(2,50):
    knn=KNeighborsClassifier(i)
    knn.fit(x_train,y_train)
    pred=knn.predict(x_test)
    error_rate.append(np.mean(y_test!=pred))
```

In [12]:

```
plt.plot(range(2,50),error_rate)
```

Out[12]:

```
[<matplotlib.lines.Line2D at 0x242a7dcb970>]
```



Selection of hyperparameter k

In [13]:

```
knn=KNeighborsClassifier(6)
knn.fit(x_train,y_train)
pred=knn.predict(x_test)
```

Print prediction of x_test

In [22]:

```
pred
```

Out[22]:

```
array(['Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
       'Iris-setosa', 'Iris-setosa', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-versicolor',
       'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor'], dtype=object)
```

Print Confusion Matrix and Classification Report and Model Accuracy

In [18]:

```
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
```

In [15]:

```
print(confusion_matrix(y_test,pred))
```

```
[[12  0  0]
 [ 0 20  2]
 [ 0  1 10]]
```

In [17]:

```
print(classification_report(y_test,pred))
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        12
Iris-versicolor       0.95      0.91      0.93        22
 Iris-virginica       0.83      0.91      0.87        11

       accuracy                           0.93        45
      macro avg       0.93      0.94      0.93        45
   weighted avg       0.94      0.93      0.93        45
```

In [19]:

```
print(accuracy_score(y_test,pred))
```

```
0.9333333333333333
```

In [ ]: