```
import numpy as np
import pandas as pd
```

```
data=pd.read_csv("iris.csv")
data.tail()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

```
data.shape
```

```
(150, 6)
```

```
# Check Category count

data['Species'].value_counts()
```

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

```
# check information of table

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             150 non-null    int64
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object
dtypes: float64(4), int64(1), object(1)
```

```
memory usage: 7.2+ KB
```

```
data.describe()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
# X Values

x=data.iloc[:,1:5]
x.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
y=data.iloc[:,-1]
y.head()
```

```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

```
##                  Feature Scaling - StandardScaler
```

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
```

```
x=scaler.fit_transform(x)
x[0:5]
```

```
array([[-0.90068117,  1.03205722, -1.3412724 , -1.31297673],
       [-1.14301691, -0.1249576 , -1.3412724 , -1.31297673],
       [-1.38535265,  0.33784833, -1.39813811, -1.31297673],
       [-1.50652052,  0.10644536, -1.2844067 , -1.31297673],
       [-1.02184904,  1.26346019, -1.3412724 , -1.31297673]])
```

```
##               Devide into Test and Train data


from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.2)
```

```
x_train.shape
```

```
(120, 4)
```

```
x_test.shape
```

```
(30, 4)
```

```
##              Used K-Nearest-Neighbors-Algorithem

from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=1)  # assume best k value is 1
model.fit(x_train,y_train)
```

```
KNeighborsClassifier(n_neighbors=1)
```

```
pred=model.predict(x_test)
pred[0:5]
```

```
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
array(['Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor'], dtype=object)
```

```
y_test[0:5]
```

```
145     Iris-virginica
24         Iris-setosa
34         Iris-setosa
77     Iris-versicolor
82     Iris-versicolor
Name: Species, dtype: object
```

```
                    ## Check Accuracy

from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_test,pred)
accuracy
```

```
1.0
```

```
                  ## Confusion Metrics

from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,pred)
cm
```

```
array([[13,  0,  0],
       [ 0,  7,  0],
       [ 0,  0, 10]], dtype=int64)
```

```
                  ## check  predicted value and  actual value

result=pd.DataFrame(data=[y_test.values,pred],index=["y_test","pred"])
#horizonantaly display
result.transpose()
# result --> verticle display values
```

	y_test	pred
0	Iris-virginica	Iris-virginica
1	Iris-setosa	Iris-setosa
2	Iris-setosa	Iris-setosa
3	Iris-versicolor	Iris-versicolor
4	Iris-versicolor	Iris-versicolor
5	Iris-setosa	Iris-setosa
6	Iris-setosa	Iris-setosa
7	Iris-setosa	Iris-setosa
8	Iris-virginica	Iris-virginica
9	Iris-setosa	Iris-setosa
10	Iris-virginica	Iris-virginica
11	Iris-versicolor	Iris-versicolor
12	Iris-virginica	Iris-virginica
13	Iris-virginica	Iris-virginica
14	Iris-virginica	Iris-virginica
15	Iris-setosa	Iris-setosa
16	Iris-versicolor	Iris-versicolor
17	Iris-virginica	Iris-virginica
18	Iris-setosa	Iris-setosa
19	Iris-setosa	Iris-setosa
20	Iris-setosa	Iris-setosa
21	Iris-setosa	Iris-setosa
22	Iris-versicolor	Iris-versicolor
23	Iris-setosa	Iris-setosa
24	Iris-virginica	Iris-virginica
25	Iris-setosa	Iris-setosa
26	Iris-versicolor	Iris-versicolor
27	Iris-virginica	Iris-virginica
28	Iris-versicolor	Iris-versicolor
29	Iris-virginica	Iris-virginica

```
##           To find correct K values

correct_sum=[]

for i in range(1,20):
    model=KNeighborsClassifier(n_neighbors=i)
    model.fit(x_train,y_train)
    pred=model.predict(x_test)
    correct=np.sum(pred==y_test)
    correct_sum.append(correct)
```

```
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
```

```
        mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

```
                # test data values -30
                # correct_sum - this returns number of correct prediction according to the y_test values
correct_sum
```

```
[30, 29, 30, 30, 30, 30, 30, 29, 30, 29, 29, 29, 29, 29, 28, 28, 27, 27, 27]
```

```
result=pd.DataFrame(data=correct_sum)
result.index=result.index+1
result.T
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	30	29	30	30	30	30	30	29	30	29	29	29	29	29	28	28	27	27	27

```
# low k values-----> underfiting
# high k alues-----> overfitting
# best middle values ---->   k=9


model=KNeighborsClassifier(n_neighbors=9)
model.fit(x_train,y_train)
pred=model.predict(x_test)
```

```
C:\Users\Isuru sandaruwan\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

```
accuracy=accuracy_score(y_test,pred)
accuracy
```

```
1.0
```

```
cm=confusion_matrix(y_test,pred)
cm
```

```
array([[13,  0,  0],
       [ 0,  7,  0],
       [ 0,  0, 10]], dtype=int64)
```

```
Start coding or generate with AI.
```