29. Data Science – Machine Learning – K Nearest Neighbor

Contents

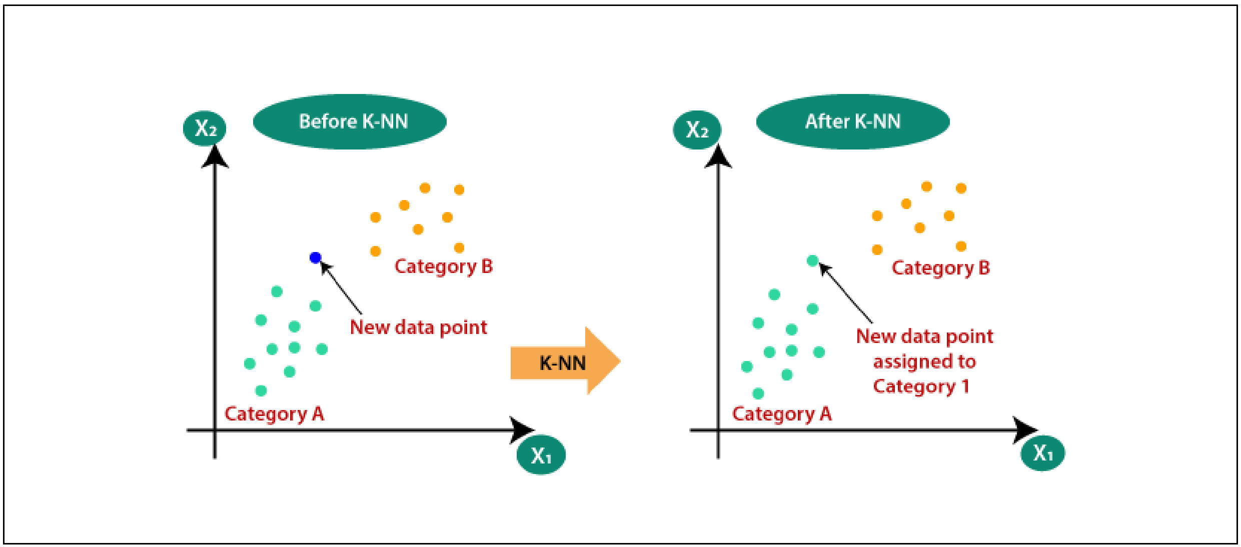
29. Data Science – Machine Learning – K Nearest Neighbor

1. K-Nearest Neighbor Algorithm

- ✓ K-Nearest Neighbor is a Supervised Learning technique.
- ✓ K-NN algorithm follow one basic rule that is, similar things are near to each other.
- ✓ It is also called a lazy learner algorithm because it does not learn from the training set immediately.
  - ○ At the time of training phase this algorithm just stores the dataset
  - ○ Whenever we get new data point then it classifies the category

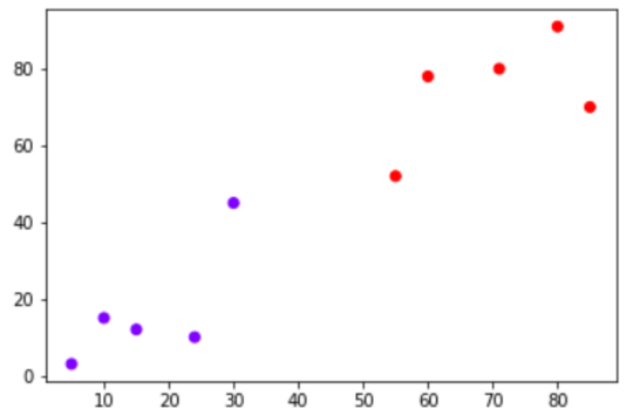
2. How it works?

- ✓ It simply calculates the distance of a new data point to all other training data points.
- ✓ The distance can be of any type e.g. Euclidean or Manhattan etc.
- ✓ It selects the K-nearest data points.
- ✓ Finally it assigns the data point to the class to which the majority of the K data points belong.

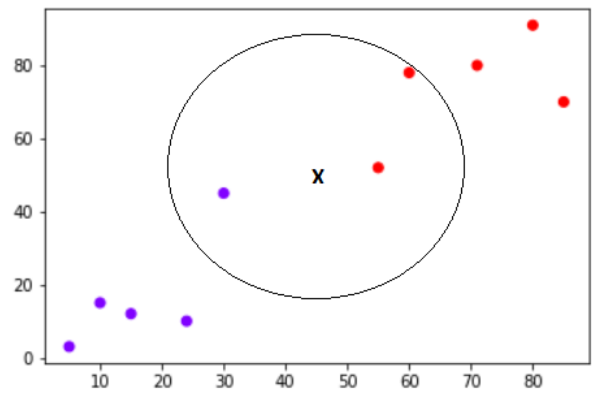


3. Scenario

- ✓ Suppose you have a dataset with two variables, which when plotted, looks like the one in the following figure.



- ✓ Our task is to classify a new data point with 'X' into "Blue" class or "Red" class.
- ✓ Suppose the value of K is 3.
- ✓ The KNN algorithm starts by calculating the distance of point X from all the points.
- ✓ It then finds the 3 nearest points with least distance to point X.
- ✓ This is shown in the figure below; the three nearest points have been encircled.

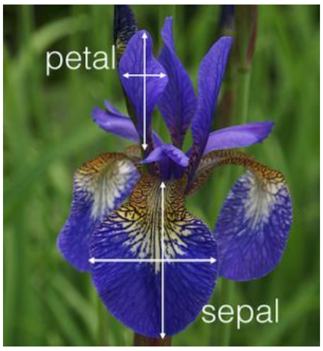


- ✓ The final step of the KNN algorithm is to assign new point to the class to which majority of the three nearest points belong.
- ✓ From the above image we can see that the two of the three nearest points belong to the class "Red" while one belongs to the class "Blue".
- ✓ Therefore the new data point will be classified as "Red".

4. Use case

- Assuming that Abhi had a hobby which is interested in distinguishing the species of some iris flowers that he has found
- He has collected some measurements associated with each iris, which are:
    - The length and width of the petals
    - The length and width of the sepals, all measured in centimetres.

- She also has the measurements of some irises that have been previously identified to the species
    - setosa,
    - versicolor
    - virginica
- The goal is to create a machine learning model that can learn from the measurements of these irises whose species are already known.
- So that we can predict the species for the new irises that she has found.



Iris Versicolor **Iris Setosa** **Iris Virginica**

Flower codes

✓ Setosa	-	0
✓ Versicolor	-	1
✓ Virginica	-	2

Program Name

Loading iris dataset
demo1.py

```
from sklearn.datasets import load_iris

iris = load_iris()

print(dir(iris))
```

Output

```
['DESCR', 'data', 'feature_names', 'filename', 'frame', 'target',
'target_names']
```

Program Name

Displaying feature names
demo2.py

```
from sklearn.datasets import load_iris

iris = load_iris()

print(iris.feature_names)
```

Output

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

Program Name

Displaying target names
demo3.py

```
from sklearn.datasets import load_iris

iris = load_iris()

print(iris.target_names)
```

Output

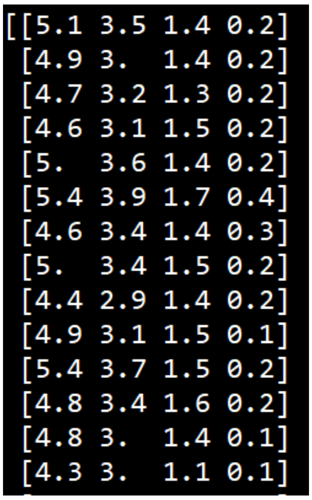
```
['setosa' 'versicolor' 'virginica']
```

Program Name

Displaying data
demo4.py

```
from sklearn.datasets import load_iris

iris = load_iris()

print(iris.data)
```

Output



```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]
 [5.4 3.7 1.5 0.2]
 [4.8 3.4 1.6 0.2]
 [4.8 3.  1.4 0.1]
 [4.3 3.  1.1 0.1]
```

Program Name

Length of the data
demo5.py

```python
from sklearn.datasets import load_iris

iris = load_iris()

print(len(iris.data))
```

Output

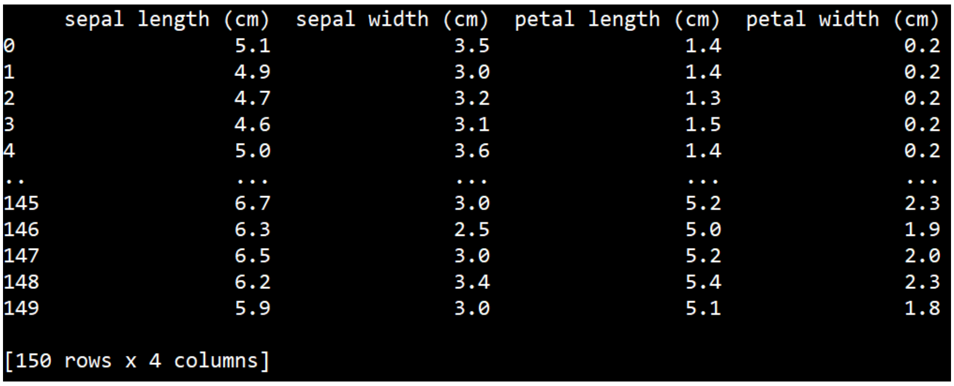
```
150
```

Program Name

Create a Dataframe by using data and features
demo6.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

print(df)
```

Output



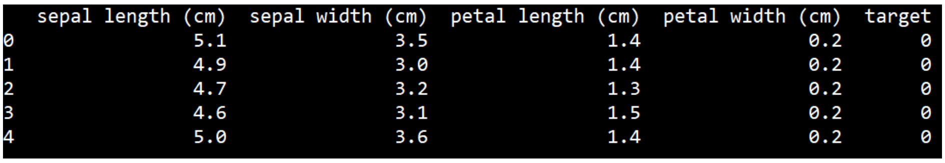
```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                  5.1               3.5                1.4               0.2
1                  4.9               3.0                1.4               0.2
2                  4.7               3.2                1.3               0.2
3                  4.6               3.1                1.5               0.2
4                  5.0               3.6                1.4               0.2
..                 ...               ...                ...               ...
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

[150 rows x 4 columns]
```

Program Name

Adding target column to the dataframe
demo7.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(df.head())
```

Output



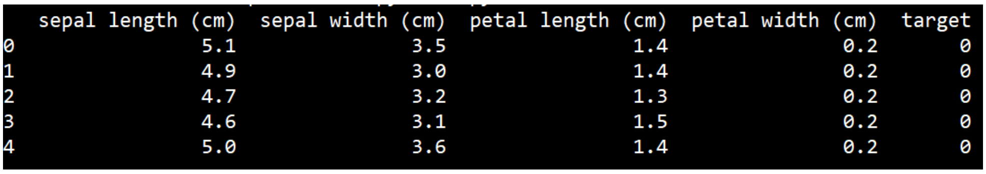
```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target
0                5.1               3.5                1.4               0.2       0
1                4.9               3.0                1.4               0.2       0
2                4.7               3.2                1.3               0.2       0
3                4.6               3.1                1.5               0.2       0
4                5.0               3.6                1.4               0.2       0
```

Program Name

Displaying target == 0 flowers
demo8.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(df[df.target==0].head())
```

Output



```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target
0                5.1               3.5                1.4               0.2       0
1                4.9               3.0                1.4               0.2       0
2                4.7               3.2                1.3               0.2       0
3                4.6               3.1                1.5               0.2       0
4                5.0               3.6                1.4               0.2       0
```

Program Name

Displaying length of the target == 0 flowers

demo9.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(len(df[df.target==0]))
```

Output

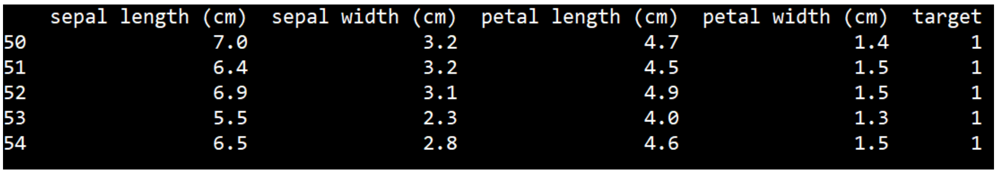
```
50
```

Program Name

Displaying target == 1 flowers
demo10.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(df[df.target==1].head())
```

Output



```
    sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target
50                7.0               3.2                4.7               1.4       1
51                6.4               3.2                4.5               1.5       1
52                6.9               3.1                4.9               1.5       1
53                5.5               2.3                4.0               1.3       1
54                6.5               2.8                4.6               1.5       1
```

Program Name

Displaying length of the target == 0 flowers
demo11.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(len(df[df.target==1]))
```

Output

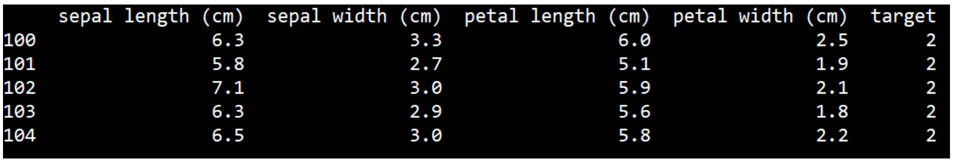
```
50
```

Program Name

Displaying target == 2 flowers
demo12.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(df[df.target==2].head())
```

Output



```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target
100                6.3               3.3                6.0               2.5       2
101                5.8               2.7                5.1               1.9       2
102                7.1               3.0                5.9               2.1       2
103                6.3               2.9                5.6               1.8       2
104                6.5               3.0                5.8               2.2       2
```

Program Name

Displaying length of the target == 2 flowers
demo13.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(len(df[df.target==2]))
```

Output

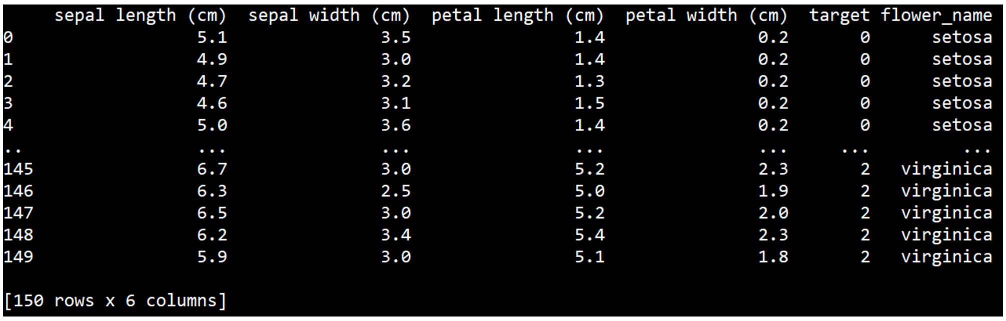
```
50
```

Program Name

Displaying the flower names
demo14.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

df['flower_name'] =df.target.apply(lambda x: iris.target_names[x])
print(df)
```

Output



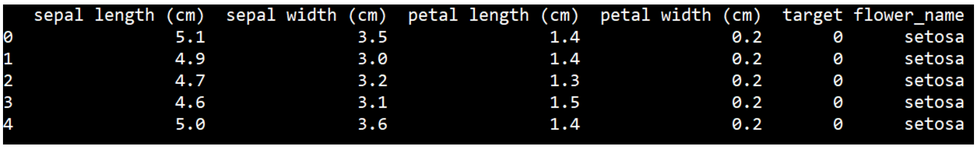
```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target flower_name
0                  5.1               3.5                1.4               0.2       0      setosa
1                  4.9               3.0                1.4               0.2       0      setosa
2                  4.7               3.2                1.3               0.2       0      setosa
3                  4.6               3.1                1.5               0.2       0      setosa
4                  5.0               3.6                1.4               0.2       0      setosa
..                 ...               ...                ...               ...     ...         ...
145                6.7               3.0                5.2               2.3       2   virginica
146                6.3               2.5                5.0               1.9       2   virginica
147                6.5               3.0                5.2               2.0       2   virginica
148                6.2               3.4                5.4               2.3       2   virginica
149                5.9               3.0                5.1               1.8       2   virginica

[150 rows x 6 columns]
```

Program Name

All setosa flowers
demo15.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

df['target'] = iris.target
df['flower_name'] = df.target.apply(lambda x:
iris.target_names[x])

setosa_50 = df[:50]
print(setosa_50.head())
```

Output

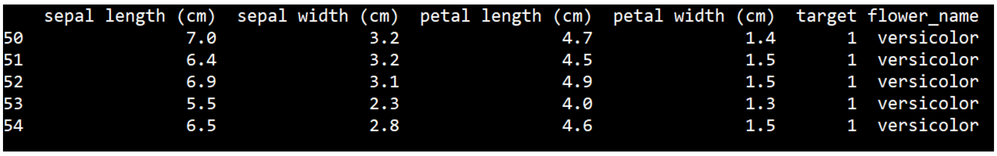


```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target flower_name
0                5.1               3.5                1.4               0.2       0      setosa
1                4.9               3.0                1.4               0.2       0      setosa
2                4.7               3.2                1.3               0.2       0      setosa
3                4.6               3.1                1.5               0.2       0      setosa
4                5.0               3.6                1.4               0.2       0      setosa
```

Program Name   All versicolor flowers
demo16.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

df['target'] = iris.target
df['flower_name'] = df.target.apply(lambda x:
iris.target_names[x])

versicolor_50 = df[50:100]
print(versicolor_50.head())
```

Output



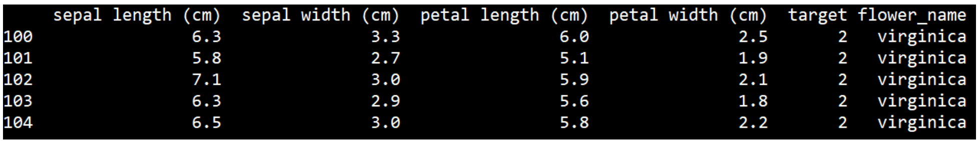
```
    sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target flower_name
50                7.0               3.2                4.7               1.4       1  versicolor
51                6.4               3.2                4.5               1.5       1  versicolor
52                6.9               3.1                4.9               1.5       1  versicolor
53                5.5               2.3                4.0               1.3       1  versicolor
54                6.5               2.8                4.6               1.5       1  versicolor
```

Program Name

All virginica flowers
demo17.py

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

df['target'] = iris.target
df['flower_name'] = df.target.apply(lambda x:
iris.target_names[x])

verginica_50 = df[100:]
print(verginica_50.head())
```

Output



```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target flower_name
100                6.3               3.3                6.0               2.5       2   virginica
101                5.8               2.7                5.1               1.9       2   virginica
102                7.1               3.0                5.9               2.1       2   virginica
103                6.3               2.9                5.6               1.8       2   virginica
104                6.5               3.0                5.8               2.2       2   virginica
```

Program Name

Splitting the data
demo18.py

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

df['target'] = iris.target
df['flower_name'] = df.target.apply(lambda x:
iris.target_names[x])

X = df.drop(['target', 'flower_name'], axis='columns')
y = df.target

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

print("Splitting the data")
```

Output

```
Splitting the data
```

Program Name

Model training
demo19.py

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

df['target'] = iris.target
df['flower_name'] = df.target.apply(lambda x:
iris.target_names[x])

X = df.drop(['target', 'flower_name'], axis='columns')
y = df.target

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

# Train Using K Neighbor classifier

classifier = KNeighborsClassifier(n_neighbors = 5)
classifier.fit(X_train, y_train)

print('Model got trained')
```

Output

```
Model got trained
```

Program Name

Model score
demo20.py

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

df['target'] = iris.target
df['flower_name'] = df.target.apply(lambda x:
iris.target_names[x])

X = df.drop(['target', 'flower_name'], axis='columns')
y = df.target

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train, y_train)

print(classifier.score(X_test, y_test))
```

Output

```
0.9666666666666667
```

Program Name

Model prediction
demo21.py

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

df['target'] = iris.target
df['flower_name'] = df.target.apply(lambda x:
iris.target_names[x])

X = df.drop(['target', 'flower_name'], axis='columns')
y = df.target

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train, y_train)

print(classifier.predict([[4.8, 3.0, 1.5, 0.3]]))
```

Output

```
[0]
```

Program Name

Model prediction
demo22.py

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

df['target'] = iris.target
df['flower_name'] = df.target.apply(lambda x:
iris.target_names[x])

X = df.drop(['target', 'flower_name'], axis='columns')
y = df.target

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

print(y_pred)
```

Output

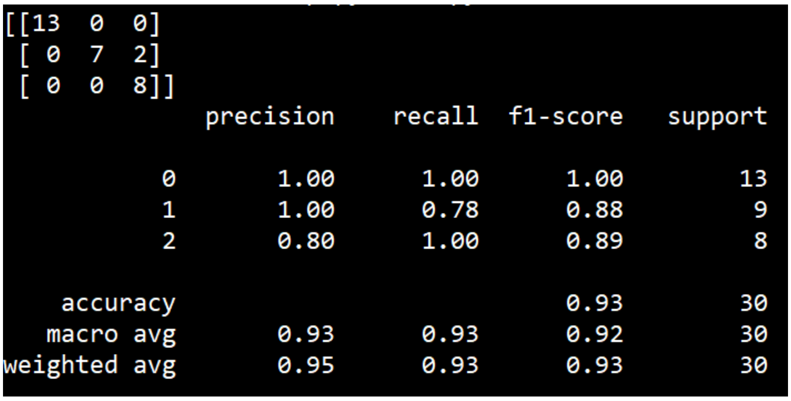
```
[0 1 0 1 2 1 0 1 2 0 0 2 1 0 1 0 0 0 0 1 1 0 2 2 0 2 0 0 1 0]
```

Program Name

Model evaluation
demo22.py

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report,
confusion_matrix

iris = load_iris()

df = pd.DataFrame(iris.data, columns=iris.feature_names)

df['target'] = iris.target
df['flower_name'] = df.target.apply(lambda x:
iris.target_names[x])

X = df.drop(['target', 'flower_name'], axis='columns')
y = df.target

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Output



```
[[13  0  0]
 [ 0  7  2]
 [ 0  0  8]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        13
           1       1.00      0.78      0.88         9
           2       0.80      1.00      0.89         8

    accuracy                           0.93        30
   macro avg       0.93      0.93      0.92        30
weighted avg       0.95      0.93      0.93        30
```