EXPERIMENT:8 KNN

Aim:

To classify data points using the K-Nearest Neighbors (KNN) algorithm and evaluate its performance on a sample dataset.

Procedure :

1. Import necessary libraries: pandas, numpy, sklearn (for KNN, train-test split, metrics), and matplotlib for visualization.
2. Load a dataset (e.g., Iris dataset) and perform basic data preprocessing like handling missing values and encoding categorical features.
3. Split the dataset into training and testing sets using train_test_split.
4. Create a KNN classifier using KNeighborsClassifier, fit it on the training data, and make predictions on the test set.
5. Evaluate the model using metrics like accuracy, confusion matrix, and visualize results if needed.

In [3]:

```
import numpy as np
import pandas as pd
df=pd.read_csv("C:\\Users\\kaviy\\Downloads\\Iris (1).csv")
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal.length  150 non-null    float64
 1   sepal.width   150 non-null    float64
 2   petal.length  150 non-null    float64
 3   petal.width   150 non-null    float64
 4   variety       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [4]:

```
df.variety.value_counts()
```

Out[4]:

```
Setosa        50
Versicolor    50
Virginica     50
Name: variety, dtype: int64
```

In [5]:

```
df.head()
```

Out[5]:

EXPERIMENT:8 KNN

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa

In [9]:

```
features=df.iloc[:,:-1].values
label=df.iloc[:,4].values
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
xtrain,xtest,ytrain,ytest=train_test_split(features,label,test_size=.2,rand
om_state=42)
model_KNN=KNeighborsClassifier(n_neighbors=5)
model_KNN.fit(xtrain,ytrain)
```

Out[9]:

```
KNeighborsClassifier
KNeighborsClassifier()
```

In [10]:

```
print(model_KNN.score(xtrain,ytrain))
print(model_KNN.score(xtest,ytest))
0.9666666666666667
1.0
```

In [11]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(label,model_KNN.predict(features))
```

Out[11]:

```
array([[50,  0,  0],
       [ 0, 47,  3],
       [ 0,  1, 49]], dtype=int64)
```

In [12]:

```
from sklearn.metrics import classification_report
print(classification_report(label,model_KNN.predict(features)))
              precision    recall  f1-score   support

      Setosa       1.00      1.00      1.00        50
  Versicolor       0.98      0.94      0.96        50
   Virginica       0.94      0.98      0.96        50

    accuracy                           0.97       150
```

```
   macro avg       0.97      0.97      0.97       150
weighted avg       0.97      0.97      0.97       150
```

Result :

- The KNN model successfully classified the data points based on their nearest neighbors.
- Accuracy on the test set depends on the chosen value of **k** and distance metric.
- Confusion matrix and classification report show the performance and areas of misclassification.