Exp-2 Implement a Classifier using Open-Source Dataset

Aim :-
To implement a simple classifier using an open source dataset.

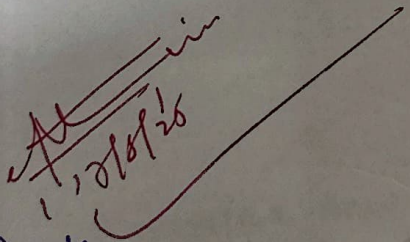
Description :-
This experiment uses the iris dataset from scikit-learn, a well know open source dataset, to classify iris flowers into three species using the K-Nearest Neighbors (KNN) algorithm. The classifier is evaluated using a confusion matrix.

Procedure :-
1.) Import the dataset and necessary libraries
2.) Split the dataset into training and test sets
3.) Train the KNN classifier on the training data
4.) Predict the test data and evalute using a confusion matrix.

Program :

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
X, y = load_iris(return_X_y = True)
X_train, X_test, y_train, y_test = train_test_
                                   split(X, y, test_size
                                   =0.3)
model = KNeighborsClassifier()
model.fit(X_train, y_train)
```

```
y_pred = model.predict(x_test)
print(confusion_matrix(y_test, y_pred))
```



Result:

A KNN classifier was successfully implemented using the Iris dataset. The model was evaluated using a confusion matrix.

Output :

[[16 0 0]

[0 13 1]

[0 1 14]]

```python
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix

X, y = load_iris(return_X_y=True) # Loading the iris dataset [cite: 24, 26]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3) # Splitting the data into training and test sets with test_size=0.3 [cite: 19, 27, 31, 32]

model = KNeighborsClassifier() # Initializing the KNN classifier [cite: 28, 29]
model.fit(X_train, y_train) # Training the classifier on the training data [cite: 21, 30]

y_pred = model.predict(X_test) # Predicting the test data [cite: 33]
print(confusion_matrix(y_test, y_pred)) # Evaluating using the confusion matrix [cite: 17, 34]
```

```
[[12  0  0]
 [ 0 17  0]
 [ 0  0 16]]
```