DATE:

EXPERIMENT-08:

Aim:

To implement k-Nearest Neighbor algorithm to classify the iris dataset.

Requirements:

Libraries used:
numpy ,pandas ,sklearn.neighbors ,sklearn.model
jupyter notebook, python environment

Procedure:

K-Nearest Neighbor Algorithm

Training algorithm:

- For each training example (x, f (x)), add the example to the list training examples
  Classification algorithm:
- Given a query instance x_q to be classified,
- Let $x_1 . . . x_k$ denote the k instances from training examples that are nearest to x_q
- Return

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

- Where, $f(x_i)$ function to calculate the mean value of the k nearest training examples.

Code:

```
import sklearn

import pandas as pd

from sklearn.datasets import load_iris

iris=load_iris()

iris.keys()

df=pd.DataFrame(iris['data'])

print(df)
print(iris['target_names'])

iris['feature_names']
X=df

y=iris['target']
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train,y_train)
import numpy as np

x_new=np.array([[5,2.9,1,0.2]])
prediction=knn.predict(x_new)
iris['target_names'][prediction]

from sklearn.metrics import confusion_matrix

from sklearn.metrics import accuracy_score

from sklearn.metrics import classification_report

y_pred=knn.predict(X_test)

cm=confusion_matrix(y_test,y_pred)

print(cm)

print(" correct predicition",accuracy_score(y_test,y_pred))

print(" worng predicition",(1-accuracy_score(y_test,y_pred)))
```

Output:

```
  0    1    2    3
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]
['setosa' 'versicolor' 'virginica']
[[19  0  0]
 [ 0 15  0]
 [ 0  1 15]]
 correct predicition 0.98
 worng predicition 0.020000000000000018
```

Result:

The above KNN algorithm successfully executed and computed the accuracy of the classifier.