**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 xq to be classified,
* Let x1 . . .xk denote the k instances from training examples that are nearest to xq
* Return



* Where, f(xi) 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.