S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 2

Aim: Implement the concept of K-Nearest Neighbors Algorithm in Machine Learning.

Name of Student : Shrutika Pradeep Bagdi

Roll No. : CS22130

Semester/Year : 6th / 3rd

Academic Session : 2024-2025

Date of Performance :

Date of Submission :

AIM: Implement the concept of K-Nearest Neighbors Algorithm in Machine Learning.

OBJECTIVE/EXPECTED LEARNING OUTCOME:

The objectives and expected learning outcome of this practical are:

- Develop a deeper understanding of the K-Nearest Neighbors Algorithm and its limitations;
- Know how to diagnose and apply corrections to some problems with the generalized K-Nearest Neighbors Algorithm.
- Use and understand generalizations of the K-Nearest Neighbors Algorithm;
- To predict the classification of a new sample point based on data points that are separated into several individual classes.

HARDWARE AND SOFTWARE REQUIREMENTS:

Hardware Requirement:

Software Requirement:

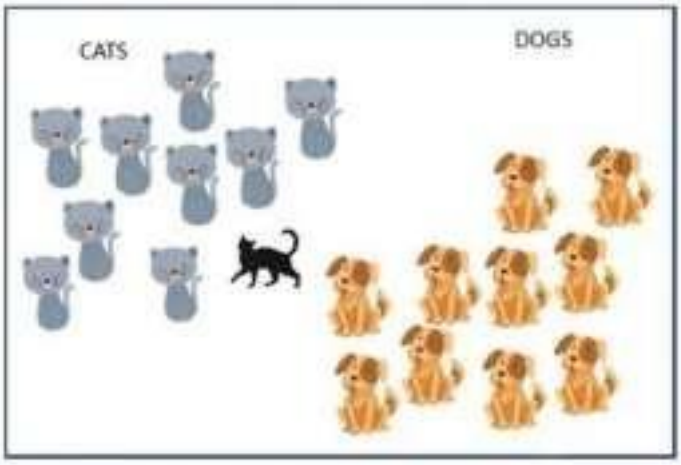
THEORY:

KNN:

The KNN algorithm is useful when you are performing a pattern recognition task for classifying objects based on different features.

Suppose there is a dataset that contains information regarding cats and dogs. There is a new data point and you need to check if that sample data point is a cat or dog. To do this, you need to list the different features of cats and dogs.

Now, let us consider two features: claw sharpness and ear length. Plot these features on a 2D plane and check where the data points fit in.



How to Choose the Factor 'K'?

A KNN algorithm is based on feature similarity. Selecting the right K value is a process called parameter tuning, which is important to achieve higher accuracy.

PROGRAM CODE:

OUTPUT (SCREENSHOT):

Commands + Code + Text

```
#Shrutika Pradeep Bagdi (CS22130)
import seaborn as sns
import pandas as pd
import numpy as np
```

```
#Shrutika Pradeep Bagdi (CS22130)
df= sns.load_dataset('iris')
df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
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

Commands + Code + Text

```
#Shrutika Pradeep Bagdi (CS22130)
df
```

	sepal_length	sepal_width	petal_length	petal_width	species
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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
#Shrutika Pradeep Bagdi (CS22130)
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   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
#Shrutika Pradeep Bagdi (CS22130)
df.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df.shape
```

```
(150, 5)
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df.tail()
```

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df['species'].unique()
```

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df.isnull().sum()
```

	0
sepal_length	0
sepal_width	0
petal_length	0
petal_width	0
species	0

dtype: int64

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df=df[df['species']!='setosa']
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df.shape
```

```
(100, 5)
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df['species']=df['species'].map({'versicolor':0, 'virginica':1})
```

```
<ipython-input-12-cb6bebc20fa7>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  df['species']=df['species'].map({'versicolor':0, 'virginica':1})
```

```
#Shrutika Pradeep Bagdi (CS22130)
df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
50	7.0	3.2	4.7	1.4	0
51	6.4	3.2	4.5	1.5	0
52	6.9	3.1	4.9	1.5	0
53	5.5	2.3	4.0	1.3	0
54	6.5	2.8	4.6	1.5	0

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    df.tail()
```

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	1
146	6.3	2.5	5.0	1.9	1
147	6.5	3.0	5.2	2.0	1
148	6.2	3.4	5.4	2.3	1
149	5.9	3.0	5.1	1.8	1

```
#Shrutika Pradeep Bagdi (CS22130)
X=df.iloc[:,:-1]
print(X)
```

```
     sepal_length  sepal_width  petal_length  petal_width
50             7.0          3.2           4.7          1.4
51             6.4          3.2           4.5          1.5
52             6.9          3.1           4.9          1.5
53             5.5          2.3           4.0          1.3
54             6.5          2.8           4.6          1.5
..             ...          ...           ...          ...
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

[100 rows x 4 columns]
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    y=df.iloc[:,-1]
    print(y)
```

```
50     0
51     0
52     0
53     0
54     0
      ..
145    1
146    1
147    1
148    1
149    1
Name: species, Length: 100, dtype: int64
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, test_size = 0.25, random_state = 42)
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    from sklearn.neighbors import KNeighborsClassifier
    knn= KNeighborsClassifier(n_neighbors=5)
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    knn.fit(X_train, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier()
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    y_pred = knn.predict(X_test)
    print(y_pred)
```

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

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    print(confusion_matrix(y_pred, y_test))
```

```
[[13  3]
 [ 1  8]]
```

```
#Shrutika Pradeep Bagdi (CS22130)
score = accuracy_score(y_pred, y_test)
print(score)
```

```
0.84
```

```
[ ] #Shrutika Pradeep Bagdi (CS22130)
    print(classification_report(y_pred, y_test))
```

```
              precision    recall  f1-score   support

           0       0.93      0.81      0.87        16
           1       0.73      0.89      0.80         9

    accuracy                           0.84        25
   macro avg       0.83      0.85      0.83        25
weighted avg       0.86      0.84      0.84        25
```

CONCLUSION:

DISCUSSION AND VIVA VOCE:

- What is the KNN Algorithm?
- Why is the odd value of "K" preferred over even values in the KNN Algorithm?
- What is the space and time complexity of the KNN Algorithm?
- Why is the KNN Algorithm known as Lazy Learner?

REFERENCE:·

❖ https://www.analyticsvidhya.com/blog/2021/05/20-questions-to-test-your-skills-on-k nearest-neighbour/

❖ https://colab.research.google.com/github/nholmber/google-colab cs231n/blob/master/assignment1/knn.ipynb