Date 4/1/25
Page

—: LAB-05 :—

✡ KNN Algo :-

K-Nearest Neighbors (KNN) is a simple, non-parametric, and lazy machine learning algorithm used for classification and regression tasks.

It majorly works on finding the 'K' closest data points to a given point and making predictions based on these neighbours.

⇒ Steps :-

01 Choose the number 'K' : determining the neighbors to consider when to classify a new data point.

02 Calculate the distance b/w the new data point

03 Sort the distance and select the 'K' nearest neighbors.

04 At last, the output.

05 Return the predicted label or value.

$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ {Distance Metric}.

Date / /
Page 21

Code Using sklearn

```
iris = load_iris()
x = iris.data
y = iris.target

x_train, x_test, y_train, y_test = train_test_split
(x, y, test_size = 0.2, random_state = 42)

scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)

knn = KNeighborsClassifier(n_neighbors = 3)

knn.fit(x_train_scaled, y_train)

y_pred = knn.predict(x_test_scaled)

accuracy = accuracy_score(y_test, y_pred)
```

=> Output :-

```
Accuracy of KNN classifier : 1.00
Predictions: [1 0 2 1 1 0 1 2 1 1 2 0 0 0 0
              1 2 1 1 2 0 2 0 2 0 2 2 2 2 2 0]
True labels: [1 0 2 1 1 0 1 2 1 1 2 0 0 0 0
              1 2 1 1 2 0 2 0 2 2 2 2 2 0 0]
```

Tuning 'K':

If K is too small, the model may be noisy and overfit the data (High Variance)

Date / /
Page

- If K is too large, the model may be too simple and underfit the data.

- A common practice is to use cross-validation to find the optimal value for K

SVM (Support Vector Machine):-

a powerful, supervised machine learning algorithm commonly used for classification and regression tasks.

→ Algorithm:-

01 Input: A dataset with labeled examples.

02 Output: A hyperplane that best separates the classes in the feature space.

03 Training:

- Find the optimal hyperplane by maximizing the margin b/w classes.

- For non-linearly separable data, apply kernal functions to transform the data into higher dimensions, where a hyperplane can be found.

⇒ classification:

1 → Yes
0 → No

Date __/__/__
Page _____

- Input: A set of labeled data points.
- Output: A class label based on the optimal hyperplane.
- Goal: Maximize the margin between classes while minimizing misclassification.

Code:-

For Example:

- Age (in years)
- Income (in thousands of dollars)
- Product Usage Frequency (scale from 1 to 10)

```
np.random.seed(42)

n_samples = 1000

age = np.random.randint(18, 70, n_samples)
income = np.random.randint(30, 150, n_samples)
usage_freq = np.random.randint(1, 11,
n_samples)
```

Output:-

```
Accuracy of SVM Classification on
Customer Purchase
Prediction: 0.99
Prediction: [0 1 1 1 1 0 0 1 0]
True labels: 521
```