PREDICTIVE ANALYSIS ANSWERS

1.Comparison of Biological Neuron vs Artificial Neural Network (ANN) Neuron

Biological Neuron	Artificial Neuron (ANN)
Has dendrites (inputs), cell body (processing), and axon (output).	Has inputs, weights, summation function, and activation function.
Inputs are electrical signals received through synapses.	Inputs are data features provided to the model.
Processes signals, and if the total exceeds a threshold, it fires.	Multiplies inputs by weights, sums them, and applies an activation function.
Output is a nerve impulse sent to other neurons.	Output is a numerical value passed to the next layer.
Learns through synaptic strength adjustment (biological learning).	Learns through weight adjustment using training algorithms (e.g., backpropagation).

2.Components of a Neural Network

1.Layers

- A neural network is made up of different layers of neurons.
- Input Layer: Accepts raw data/features from the dataset. Example: in image recognition, pixels act as inputs.

- Hidden Layers: Intermediate layers where most of the computations take place. They transform inputs into meaningful representations.
- Output Layer: Produces the final prediction or classification. Example: identifying whether an image is of a cat or dog.

---

2. Neurons
    - A neuron is the fundamental processing unit of a neural network.
    - Each neuron receives multiple inputs, applies weights, sums them, and passes the result through an activation function.
    - Neurons work together in layers to capture complex patterns in the data.

---

3. Weights
    - Weights are parameters that define the importance of each input connection.
    - During training, weights are adjusted to minimize the error between predicted and actual output.
    - Larger weights mean stronger influence of the input feature on the output.

---

4. Bias

- Bias is an additional parameter provided to neurons along with inputs.
- It helps in shifting the activation function curve and allows the model to fit the data more effectively.
- Without bias, the network's ability to learn patterns would be limited.

---

5. Activation Functions
    - Activation functions introduce non-linearity into the model, enabling the network to learn complex relationships.
    - Common activation functions:
        - Sigmoid: Outputs values between 0 and 1, often used for binary classification.
        - ReLU (Rectified Linear Unit): Outputs zero for negative inputs and linear for positive ones, widely used due to efficiency.
        - Tanh: Outputs values between –1 and 1, useful for centered data.
    - Without activation functions, the network would behave like a simple linear model and fail to capture real-world complexities

3.Describe the working of KNN.

K-Nearest Neighbors (K-NN) is a well-known instance-based learning Algorithm.

In summary, instance-based learning makes decisions by comparing new instances to previously stored ones, without creating a model in advance.

• K-Nearest Neighbor is one of the **simplest Machine Learning** algorithms based on Supervised Learning technique.

• K-NN algorithm **assumes the similarity between the new case/data and available cases and put the new case into the category** that is most similar to the available categories.

• K-NN **algorithm stores all the available data and classifies a new** data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN

algorithm.

• K-NN algorithm can be **used for Regression as well as for Classification** but mostly it is used for the Classification problems

• K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.

• It is also called a **lazy learner algorithm** because it does not learn from the training set immediately **instead it stores the dataset** and at the time of classification, it performs an action on the dataset.

• KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

• Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

How does K-NN work ?

• Step-1: Select the number K of the neighbors

• Step-2: Calculate the Euclidean distance of K number of neighbors

• Step-3: Take the K nearest neighbors as per the calculated Euclidean

distance.

• Step-4: Among these k neighbors, count the number of the data points

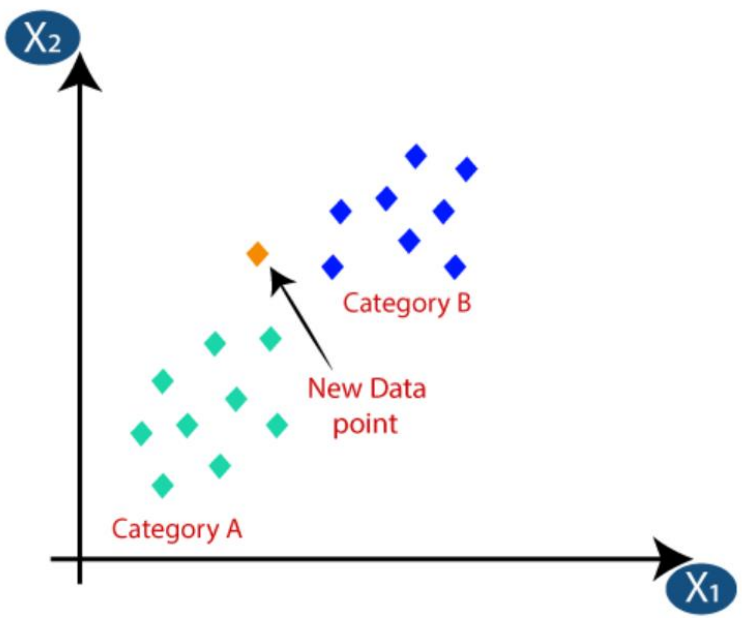
in each category.

• Step-5: Assign the new data points to that category for which the

number of the neighbor is

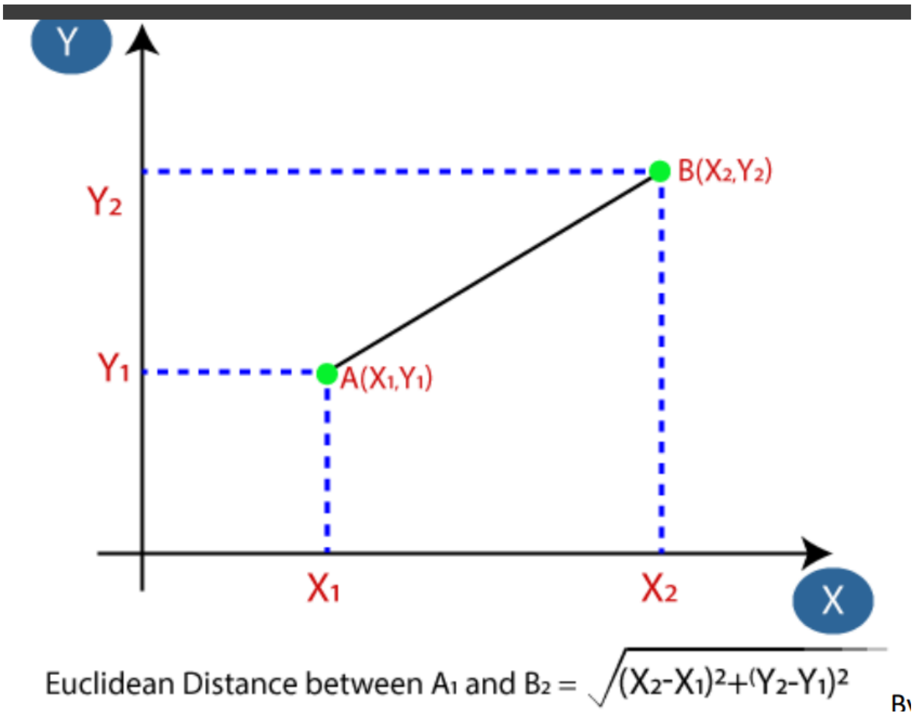
maximum

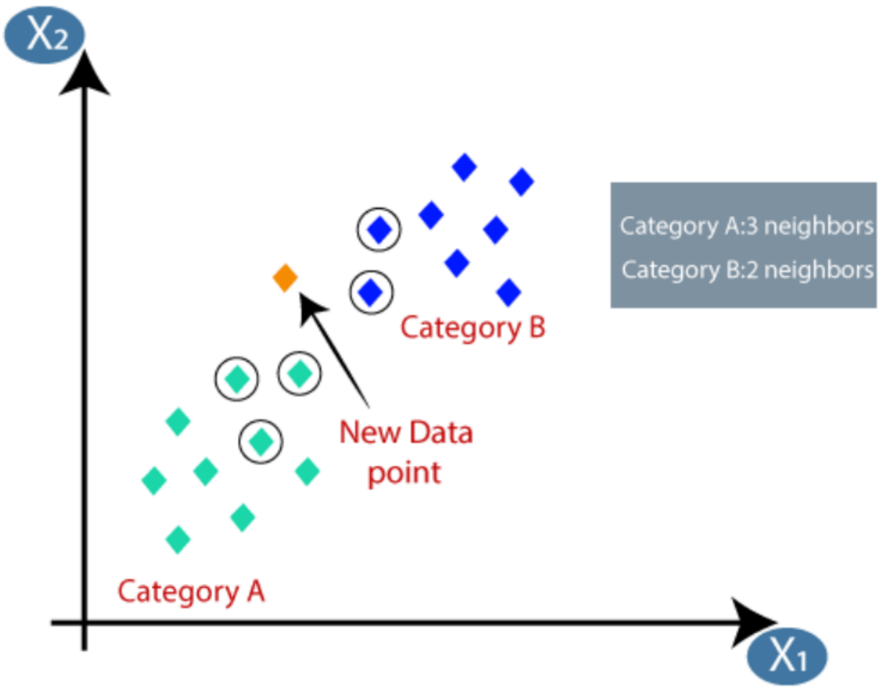
• Step-6: Our model is ready



Firstly, we will choose the number of neighbors, so we will choose the k=5.

• Next, we will calculate the Euclidean distance between the data points.





Advantages of KNN Algorithm:

• It is simple to implement.

• It is robust to the noisy training data

• It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

• Always needs to determine the value of K which may be complex

some time.

• The computation cost is high because of calculating the distance

• between the data points for all the training

• **Applications of KNN**

1. Recommendation Systems
2. Fraud Detection
4. Medical Diagnosis
5. Spam Detection

4.Explain working of SVM and what are hard margin and soft margin.

Support Vector Machine (SVM):

Support Vector Machine (SVM) is a supervised learning algorithm primarily used for classification tasks, though it can also be used for regression.

It finds the optimal decision boundary (or hyperplane) that separates classes in the feature space with the maximum margin.

Key Concepts

1. Hyperplane

A hyperplane is a decision boundary that separates data points into different classes.

In an n- dimensional feature space:

A hyperplane is an (n−1) dimensional subspace (e.g., a line in 2D or a plane in 3D).

2. Margin

The margin is the distance between the hyperplane and the closest data points from each class.

SVM aims to maximize this margin to ensure the classifier is robust and generalizes well to unseen data.

3. Support Vectors

Support vectors are the data points that lie closest to the hyperplane.

These points are critical as they determine the position and orientation of the hyperplane.

Objective of SVM

- SVM optimizes the following function:

$$\text{Maximize: } \frac{2}{||w||}$$

  - w: Weight vector that defines the hyperplane.
  - $||w||$: Magnitude of the weight vector.
- Subject to the constraint:

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

  - y_i: Class label ($+1$ or -1).
  - x_i: Feature vector.
  - b: Bias term.

Steps for Implementing SVM

1. Prepare Data: Preprocess and scale the features.

2. Choose Kernel: Select an appropriate kernel function (e.g., linear, RBF).

3. Train the Model: Use an SVM library to train the model.

4. Hyperparameter Tuning: Adjust , kernel parameters (,) for optimal performance.

5. Evaluate Performance: Use metrics like accuracy, precision, recall, or F1-score.

Kernel Functions

1. Linear Kernel

2. Polynomial Kernel

3. Radial Basis Function (RBF)

4. Sigmoid Kernel

Advantages

1. Effective for high-dimensional data.

2. Works well with a clear margin of separation.

Disadvantages

1. Computationally expensive for large datasets.

2. Choice of kernel and hyperparameters (e.g., ,) can be challenging.

3. Sensitive to noise and outliers.

Applications

1. Text classification (e.g., spam detection).

2. Image recognition.

3. Bioinformatics (e.g., protein classification).

4. Financial analysis

Hard Margin SVM:

- Finds a hyperplane that separates classes **perfectly** with **no misclassification**.
- Works only if data is **linearly separable**.
- Sensitive to noise and outliers.

Soft Margin SVM:

- Allows **some misclassification** to handle noisy or overlapping data.
- Uses **slack variables** and regularization parameter **C** to control the trade-off between margin width and errors.
- More flexible and practical for real-world datasets.