KNN and SVM Machine learning

Introduction to k-Nearest Neighbors

▪

Definition: A simple, instance-based learning algorithm used for classification and regression tasks.

Important Points:

- Non-parametric (makes no assumptions about the data distribution)
- Lazy learning (no explicit training phase, relies on storing instances)
- Sensitive to the value of 'k' (number of neighbors considered)

How kNN Works

.

1. **Data Points**: Given a dataset of labeled instances.
2. **Choose k**: Select a value for k (e.g., k=3).
3. **Calculate Distance**: Use a distance metric (usually Euclidean distance) to find the nearest neighbors.
4. Vote/Mean:
   - **Classification**: Take a majority vote among the k-nearest neighbors.
   - **Regression**: Take the mean of the k-nearest neighbors' values.
5. **Assign Label/Value**: Assign the label or value based on the results

Important formula

▪

Distance Metrics:

- Euclidean Distance: d = $\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
- Manhattan Distance: d = $\sum_{i=1}^{n} |x_i - y_i|$ Choosing k:
- Lower values of k make the model sensitive to noise.
- Higher values of k can make the model smoother but less sensitive to fine-grained patterns.

Applications of kNN

.

Image Recognition: Used for categorizing images based on similar patterns.

Recommendation Systems: Suggests items based on user similarity.

Anomaly Detection: Identifying unusual patterns by observing the majority.

Drawbacks of kNN

▪

- **High Computational Cost**: Need to compute distance for all instances during prediction.
- **Memory Intensive**: Stores all training data for comparison.
- **Sensitive to Data Scaling**: Requires normalized or scaled data for effective performance.

Support Vector Machine(SVM)

Introduction

▪

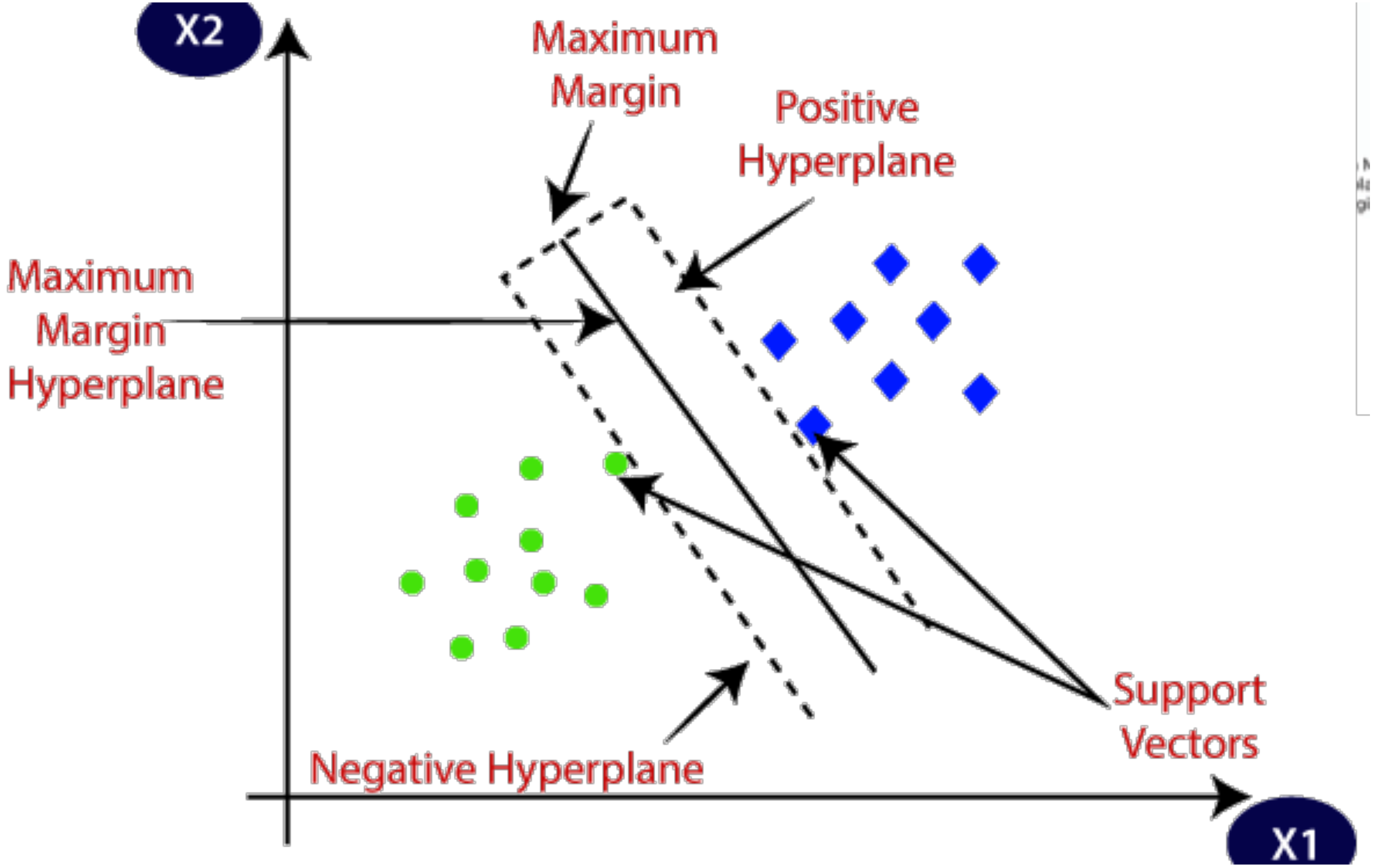
Definition: A supervised learning algorithm that finds the optimal hyperplane to separate data into classes.

- Important Points:
- Effective in high-dimensional spaces.
- Works well for classification and regression tasks.
- Maximizes the margin between classes, which increases robustness.

Diagram

▪

• .



X2
Maximum
Margin
Positive
Hyperplane
Maximum
Margin
Hyperplane
Support
Vectors
Negative Hyperplane
X1

SVM Works

▪

1. **Hyperplane Selection**: Identify a hyperplane that best divides the data into classes.

2. **Margin Maximization**: Choose the hyperplane with the maximum margin (distance) between nearest data points (support vectors).

3. **Support Vectors**: Data points closest to the hyperplane which influence its position.

4. **Kernel Trick**: Allows SVM to perform well on non-linear data by transforming data into a higher dimension.

Applications of SVM

▪

- **Text Classification**: Spam filtering, sentiment analysis.
- **Image Classification**: Recognizing objects in images.
- **Bioinformatics**: Classifying proteins and gene sequences.