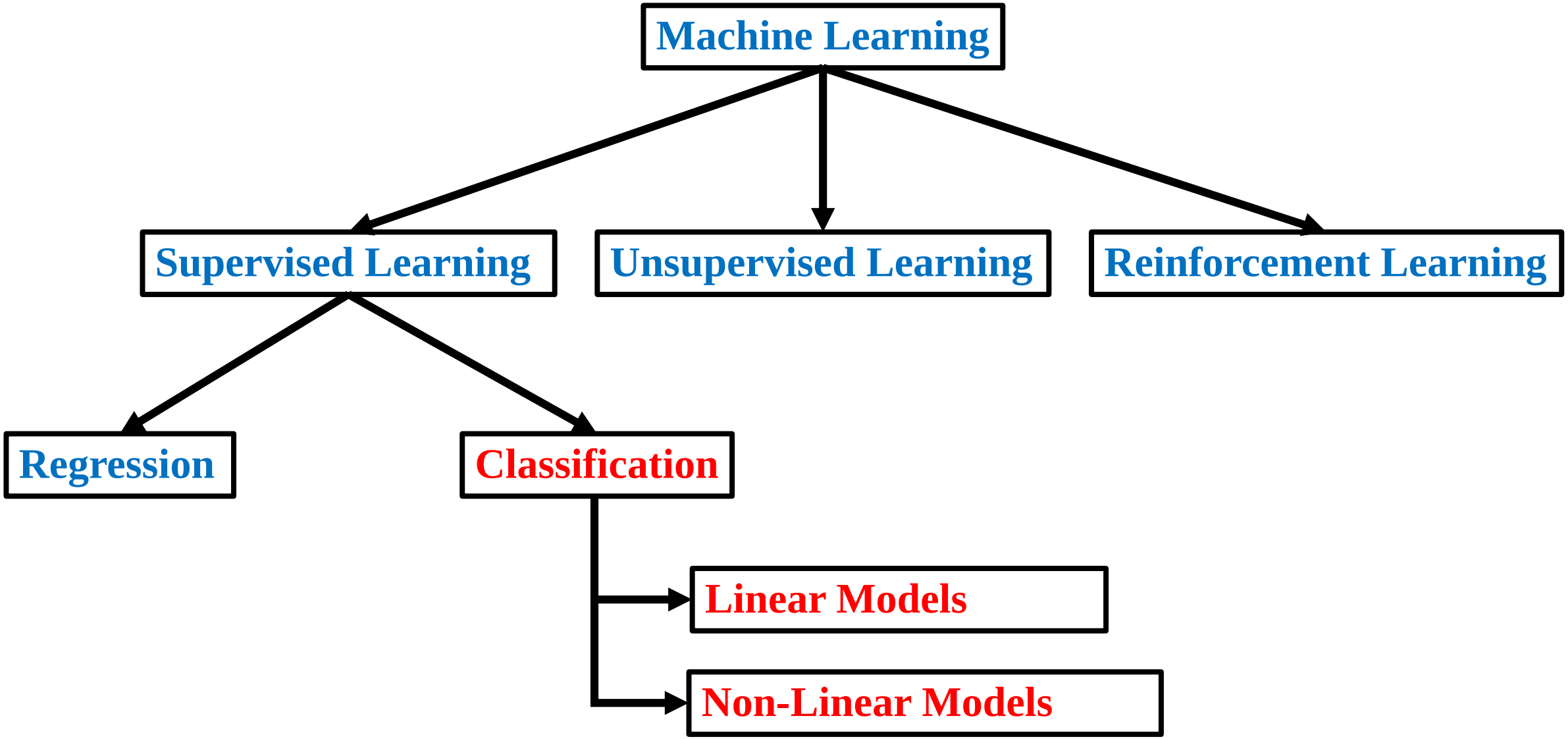
Unit IV Syllabus

- **Classification:** K-nearest Neighbour, Support Vector Machine.
- **Ensemble Learning:** Bagging, Boosting, Random Forest, Adaboost.
- **Binary-vs-Multiclass Classification, Balanced and Imbalanced Multiclass Classification Problems, Variants of Multiclass Classification: One-vs-One and One-vs-All**
- **Evaluation Metrics and Score:** Accuracy, Precision, Recall, Fscore, Cross-validation, Micro-Average Precision and Recall, Micro-Average F-score, Macro-Average Precision and Recall, Macro-Average F-score.

Types in Machine Learning



Types of Problems in Machine Learning

- **Regression**

  Output is a continuous quantity. To predict the weight of a person using height. e.g. Linear Regression.

- **Classification**

  Output is a categorical value. To classify emails into two classes, spam and non-spam. e.g. classification algorithms such as Support Vector Machines, Naive Bayes, Logistic Regression, K Nearest Neighbor.

- **Clustering**

  Problem involves assigning the input into two or more clusters based on feature similarity. Similar groups based on their interests, age, geography, etc can be done by using Unsupervised Learning algorithms like K-Means Clustering.

Classification

- The algorithm which implements the classification on a dataset is known as a **classifier**.

There are two types of Classifications:

Binary Classifier: If the classification problem has only two possible outcomes, then it is called as Binary Classifier.

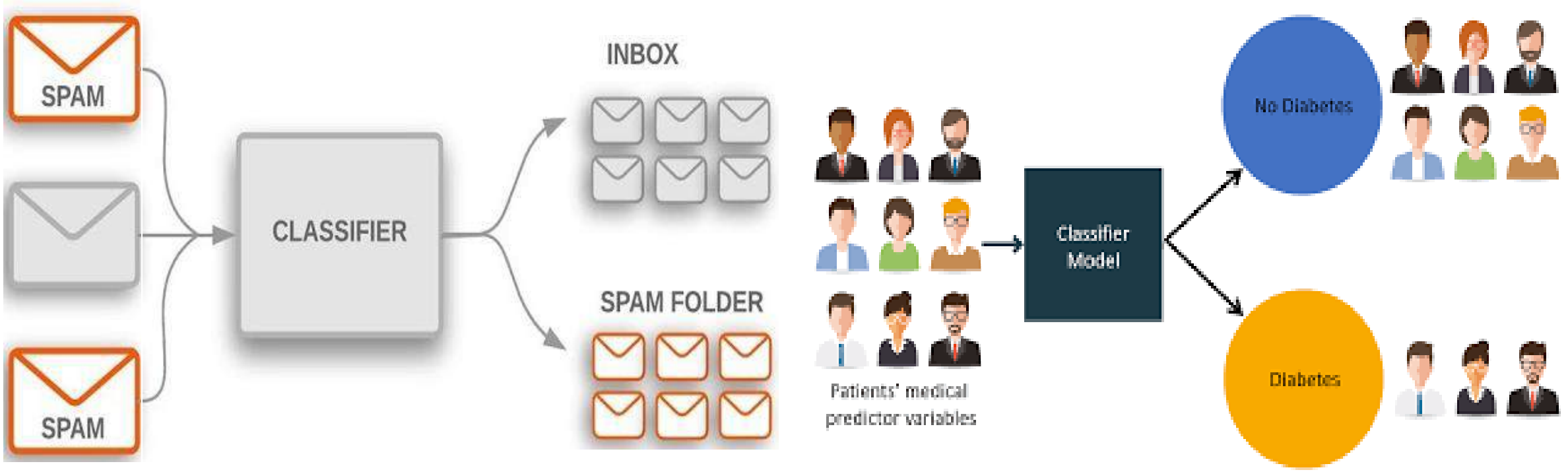
Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.

Multi-class Classifier: If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.

Example: Classifications of types of crops, Classification of types of music.

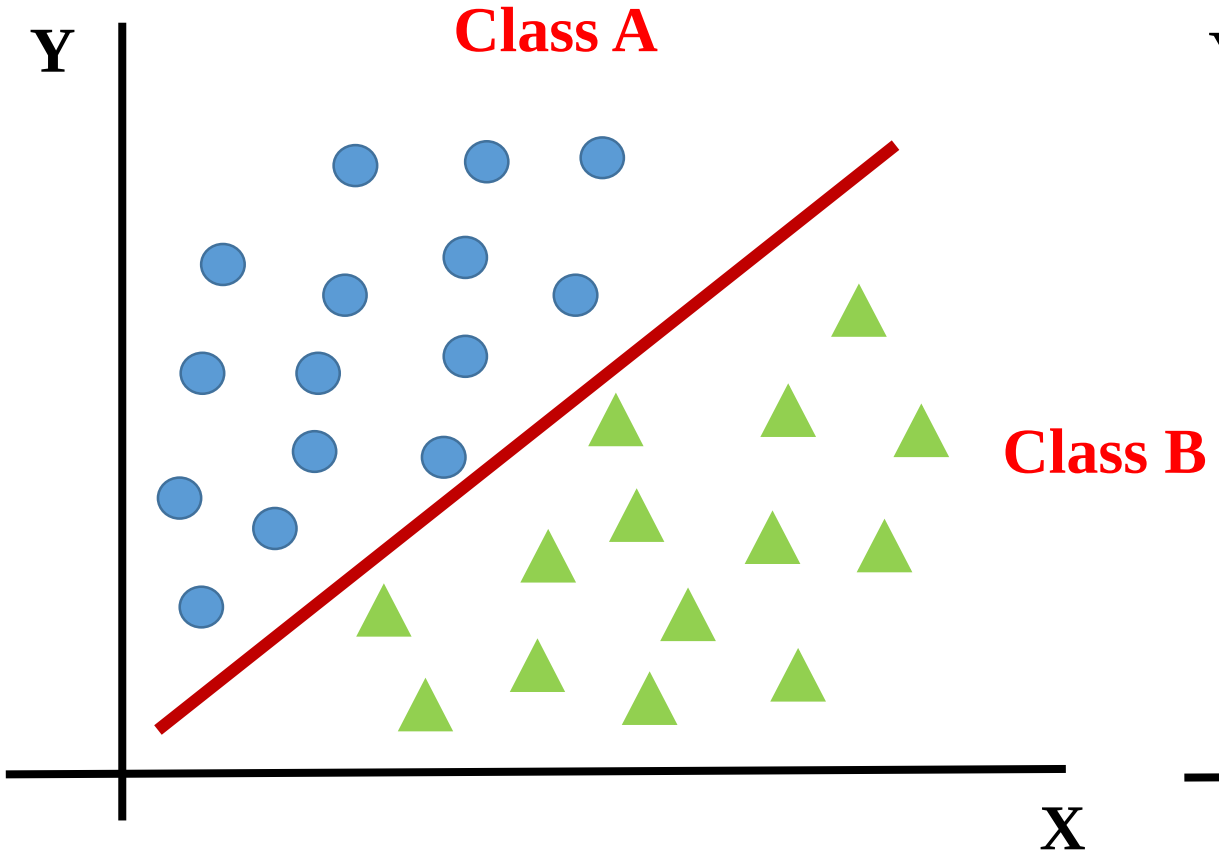
Classification

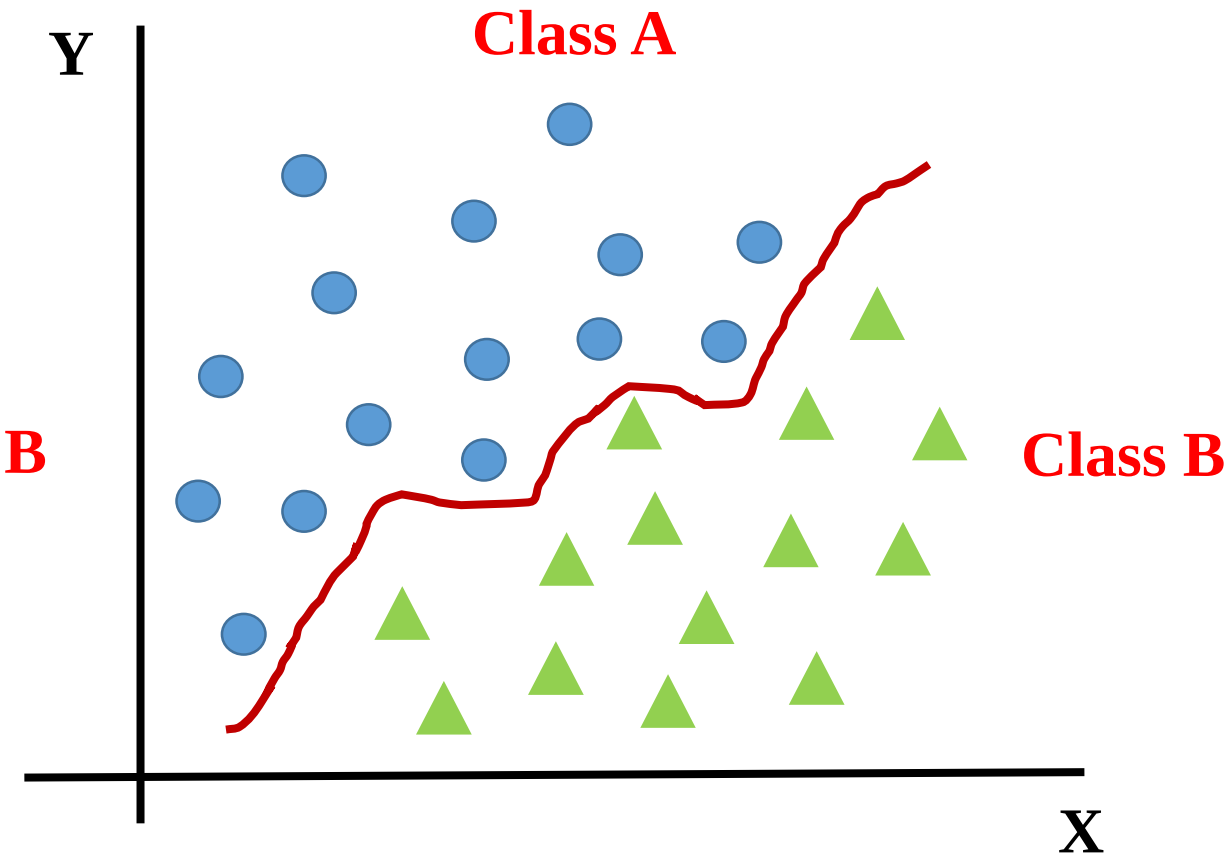
- Output is a categorical value. To classify emails into two classes, spam and non-spam. e.g. classification algorithms such as Support Vector Machines, K-nearest Neighbor.



Classification

- Output is a Numerical value. To classify emails into two classes, spam and non-spam. e.g. classification algorithms such as Support Vector Machines, K-nearest Neighbor.





Classification

In the classification problems, there are two types of learners:

Lazy Learners:

- Lazy Learner firstly stores the training dataset and wait until it receives the test dataset.
- In Lazy learner case, classification is done on the basis of the most related data stored in the training dataset.
- It takes less time in training but more time for predictions.

Example: K-NN Algorithm

Eager Learners:

- Eager Learners develop a classification model based on a training dataset before receiving a test dataset.
- Opposite to Lazy learners, Eager Learner takes more time in learning, and less time in prediction.

Example: Decision Trees, Naïve Bayes, ANN.

Classification

Classification Algorithms can be further divided into the mainly two category:

Linear Models

- Logistic Regression
- Support Vector Machines

Non-linear Models

- K-Nearest Neighbours
- Kernel SVM
- Naïve Bayes
- Decision Tree Classification
- Random Forest Classification

K-nearest Neighbour

- 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 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.

K-nearest Neighbour

Training Dataset

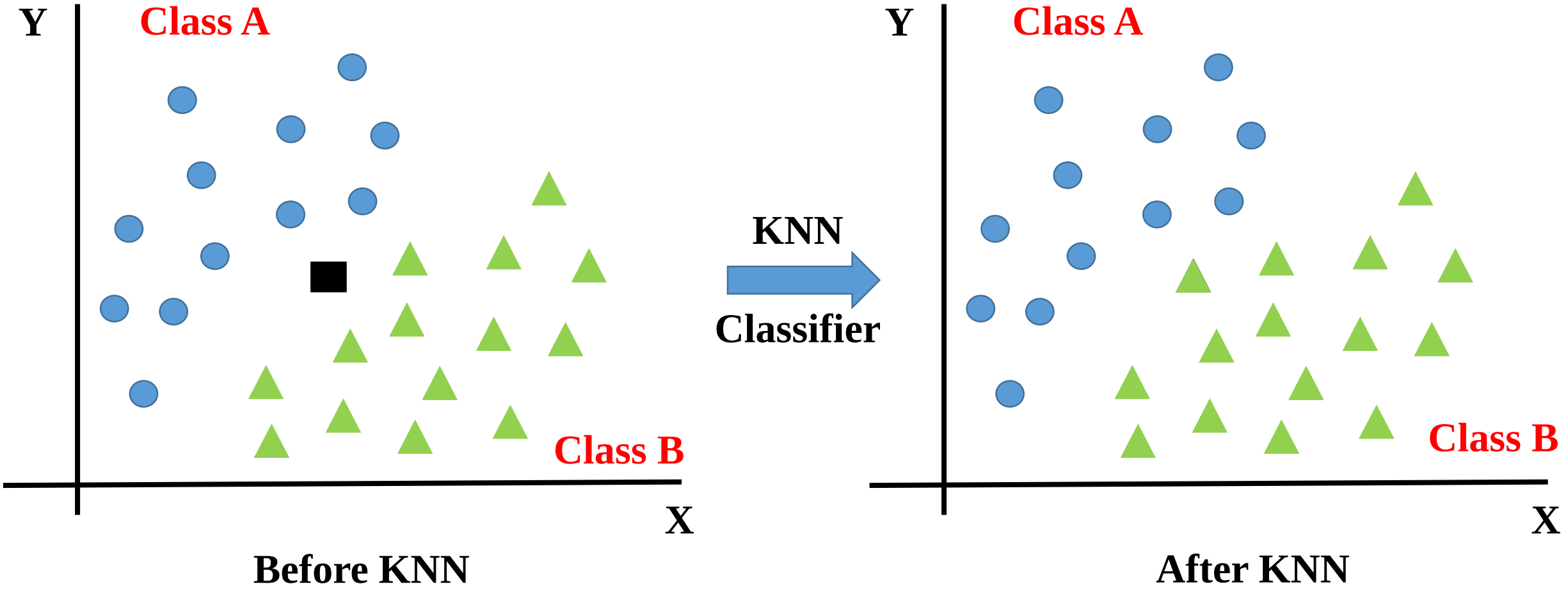
Test Data

KNN Classifier

Predicted Class

Cat Class

Why K-nearest Neighbour is Needed?



How K-nearest Neighbour Works?

The K-NN working can be explained on the basis of the below algorithm:

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.

K-nearest Neighbour

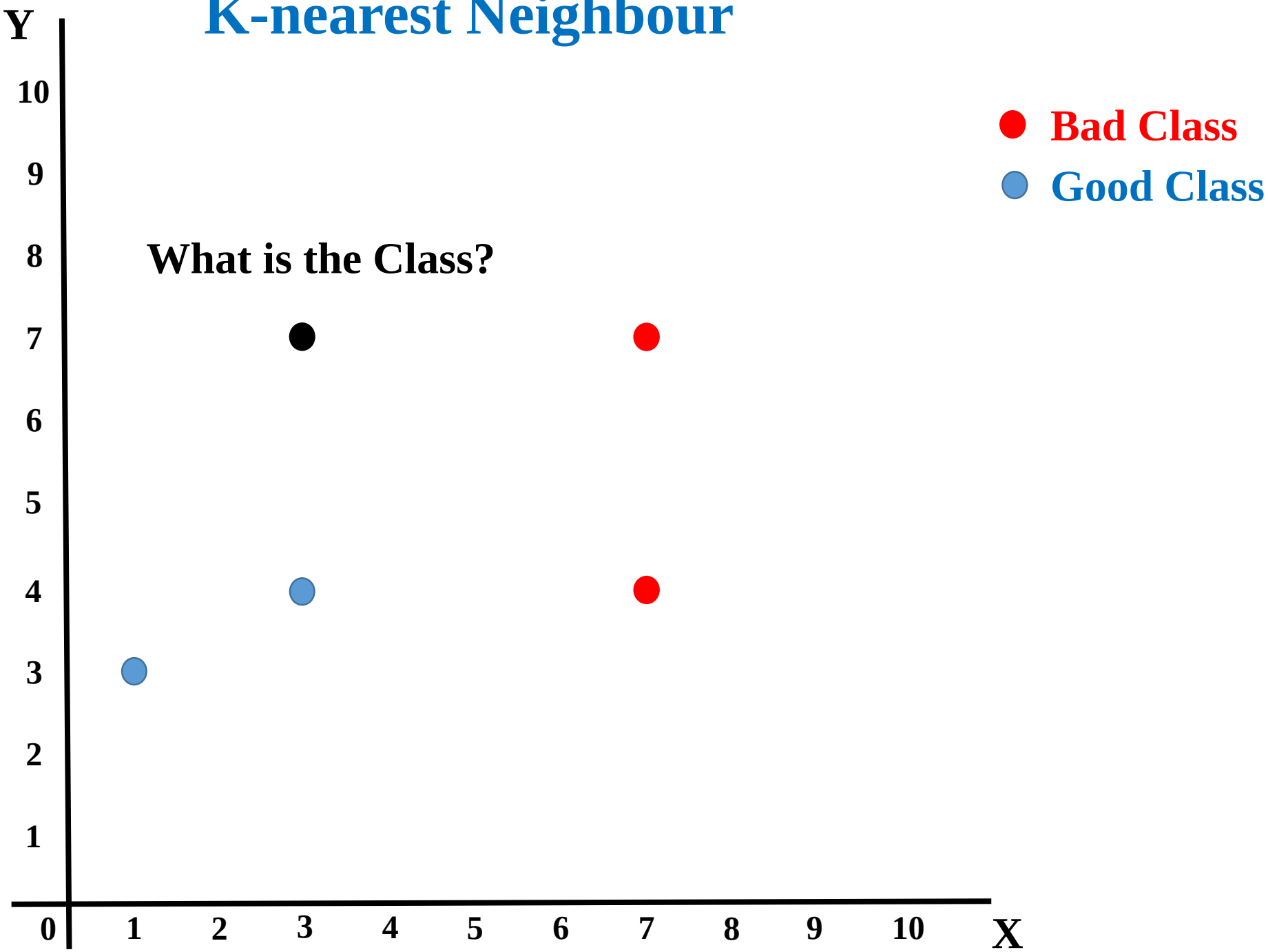
Step-1: Select the number K of the neighbors

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. **The most preferred value for K is 5**.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

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

$$distance(A,B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K-nearest Neighbour



K-nearest Neighbour

Given below data points, Find the class for X = 3 and Y = 7 using KNN.

X1	X2	Predicted Values (Y)
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good
3	7	?

K-nearest Neighbour

Given below data points, Find the class for X = 3 and Y = 7 using KNN.

Let K = 3, and calculate Euclidean distance.

X1	X2	Predicted Values (Y)	Euclidean Distance
7	7	Bad	16
7	4	Bad	25
3	4	Good	9
1	4	Good	13
3	7	?	

K-nearest Neighbour

Given below data points, Find the class for X = 3 and Y = 7 using KNN.

Now Rank the minimum Euclidean distance.

X1	X2	Predicted Values (Y)	Euclidean Distance	Rank
7	7	Bad	16	3
7	4	Bad	25	4
3	4	Good	9	1
1	4	Good	13	2
3	7	?		

K-nearest Neighbour

Given below data points, Find the class for X = 3 and Y = 7 using KNN.

Is it included in 3 - nearest neighbors.

X1	X2	Predicted Values (Y)	Euclidean Distance	Rank	3 NN
7	7	Bad	16	3	Yes
7	4	Bad	25	4	No
3	4	Good	9	1	Yes
1	4	Good	13	2	Yes
3	7	?			

K-nearest Neighbour

Given below data points, Find the class for X = 3 and Y = 7 using KNN.

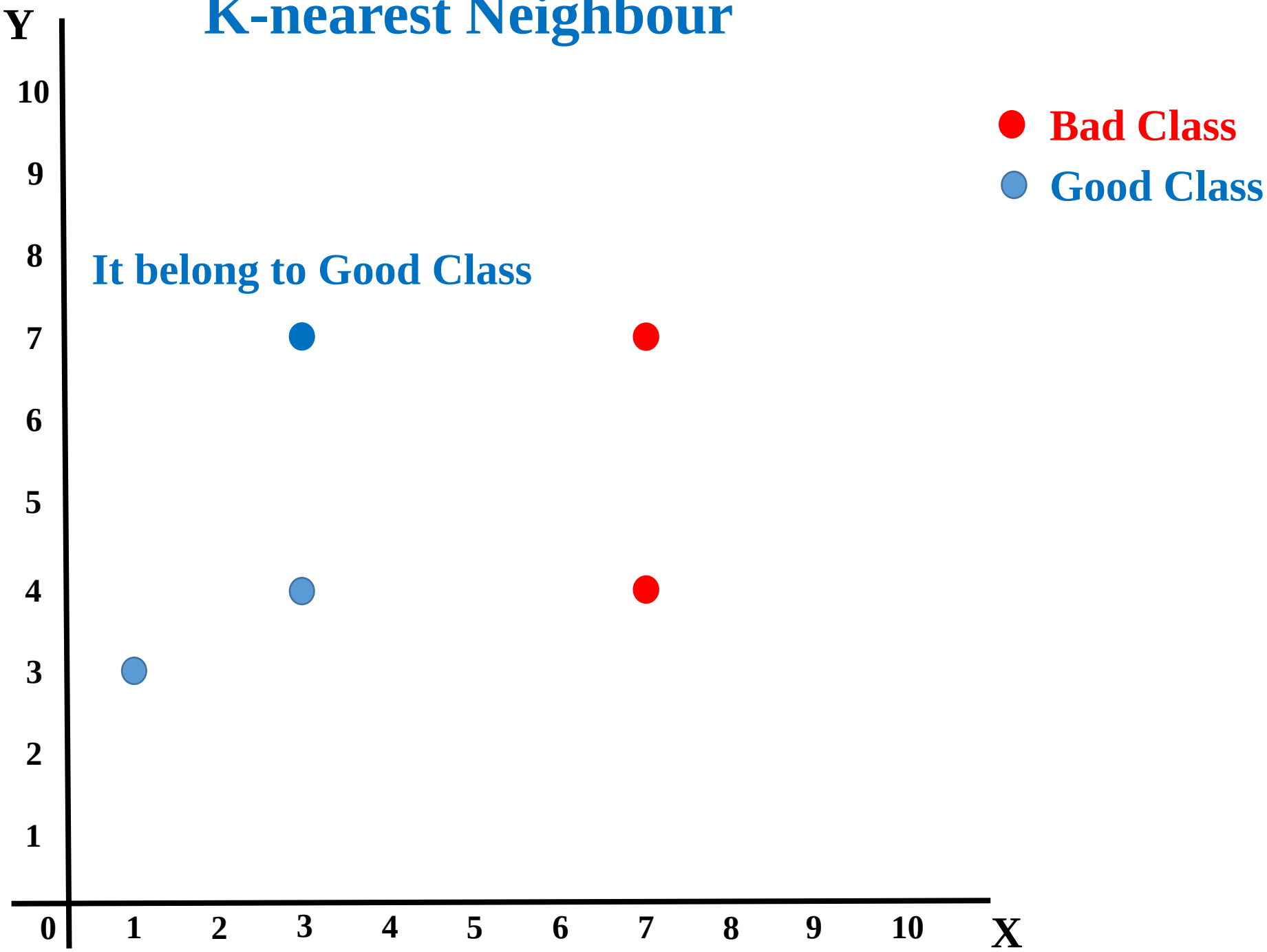
Mention class of selected 3 – nearest neighbors.

X1	X2	Predicted Values (Y)	Euclidean Distance	Rank	3 NN	Class
7	7	Bad	16	3	Yes	Bad
7	4	Bad	25	4	No	-
3	4	Good	9	1	Yes	Good
1	4	Good	13	2	Yes	Good
3	7	?				

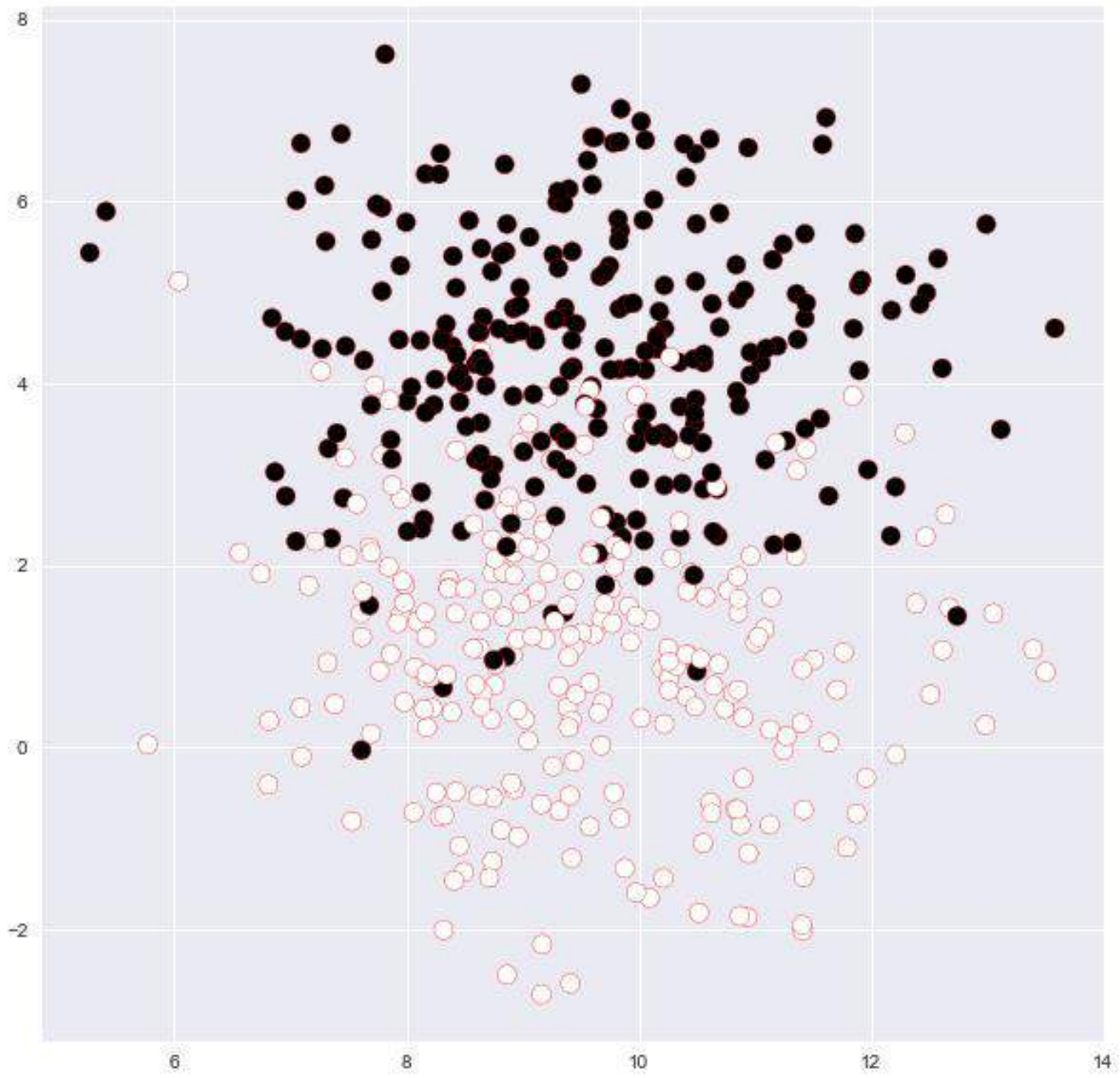
Out of 3 – nearest neighbors, 2 belong to Good class and 1 belong to bad class.

Here 2>1, So new test data points, 3 and 7 belongs to Good class.

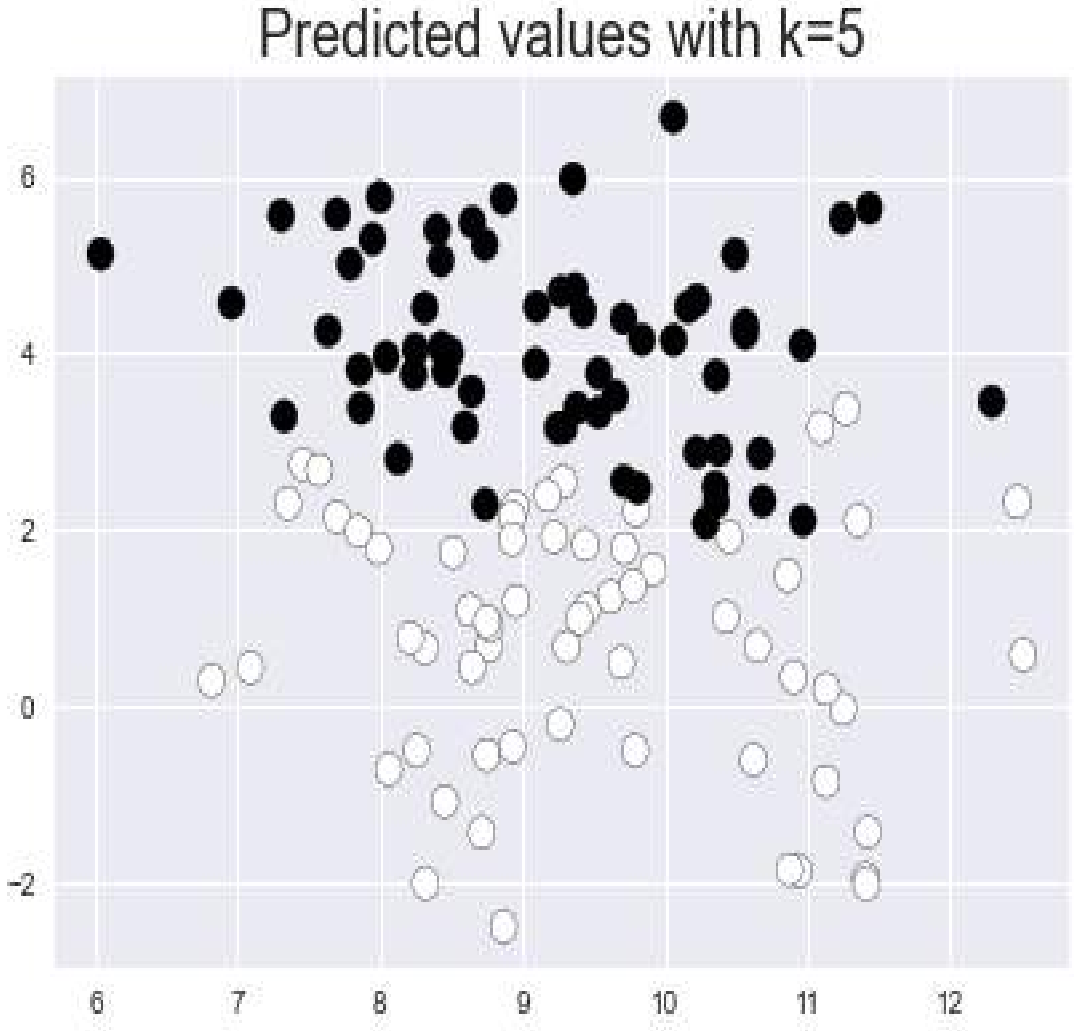
K-nearest Neighbour



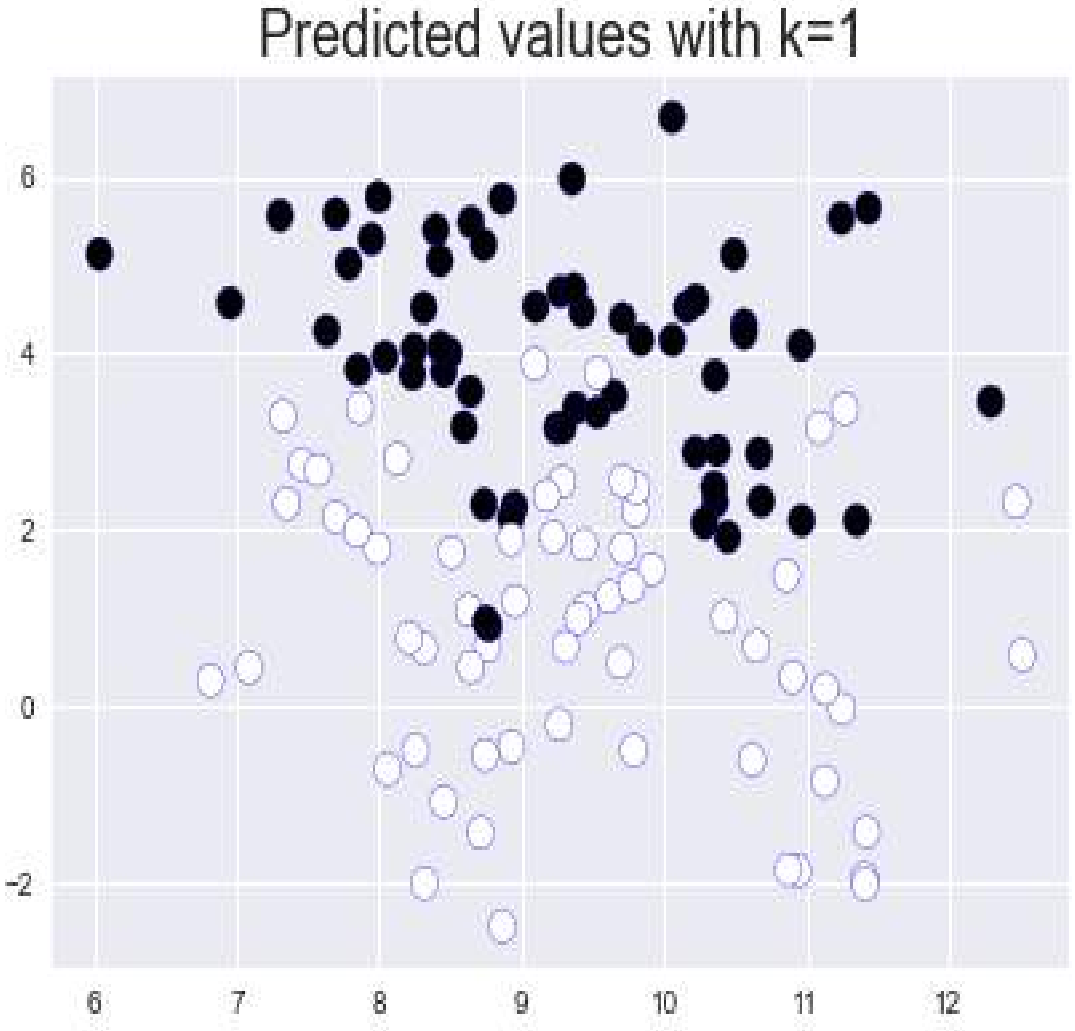
K-nearest Neighbour



K-nearest Neighbour



Accuracy with k=5 : 85.6



Accuracy with k=1: 76.8

K-nearest Neighbour

Advantages of KNN

- 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

- 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 samples.

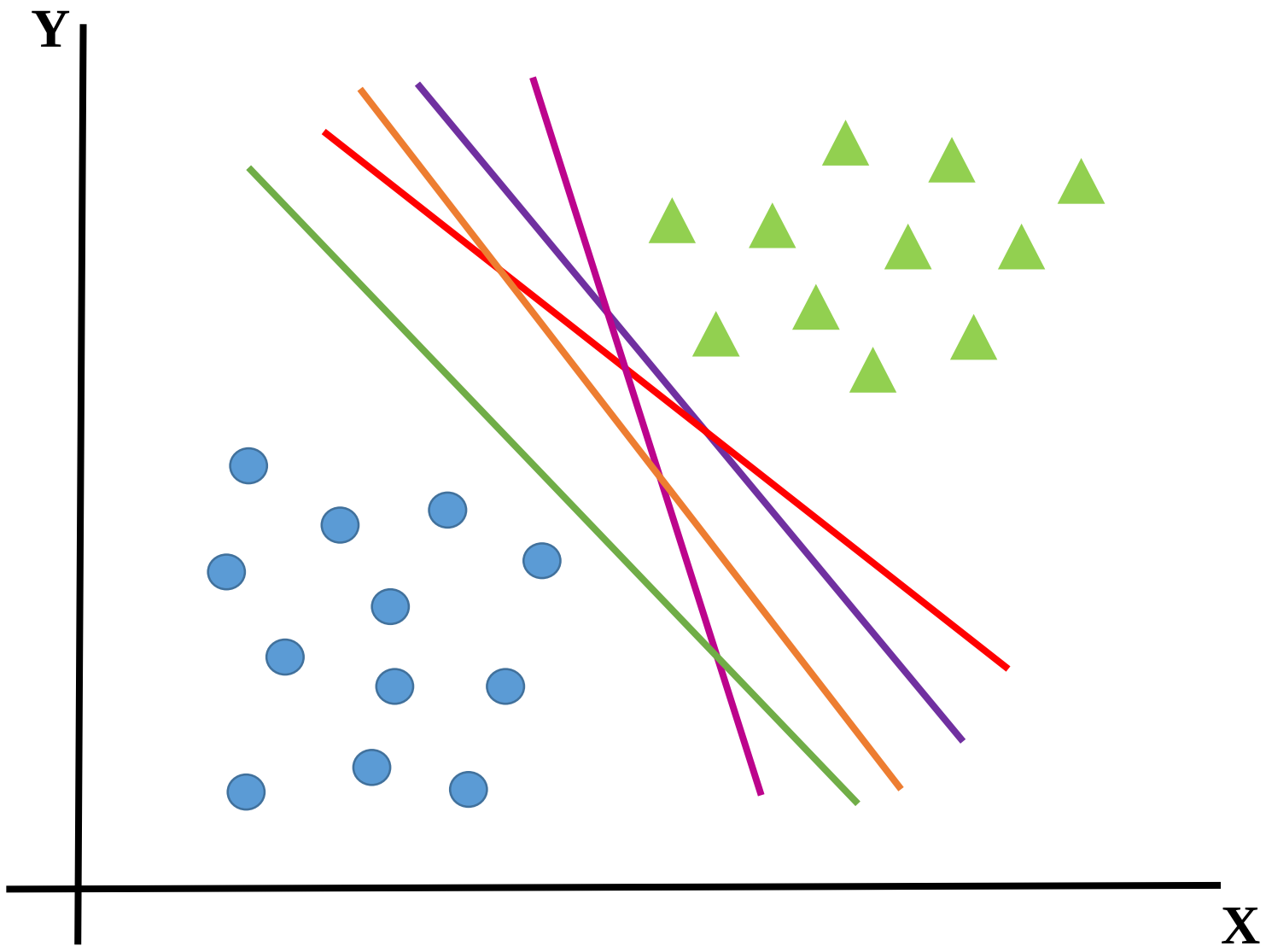
Support Vector Machine

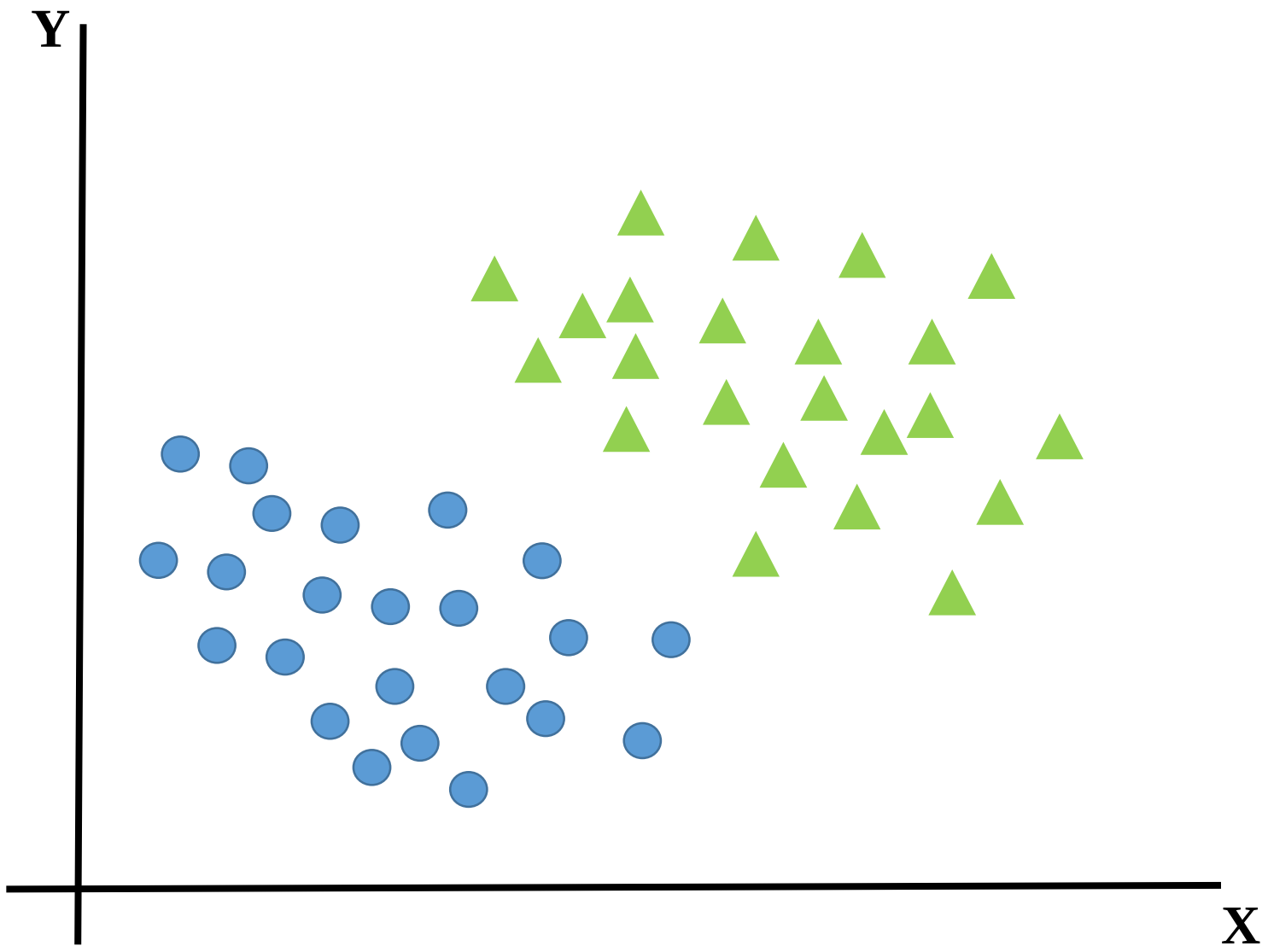
- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.

- The goal of the SVM algorithm is to create the **best line** or **decision boundary** that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a **hyperplane**.

- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as **support vectors**, and hence algorithm is termed as **Support Vector Machine**.

Support Vector Machine



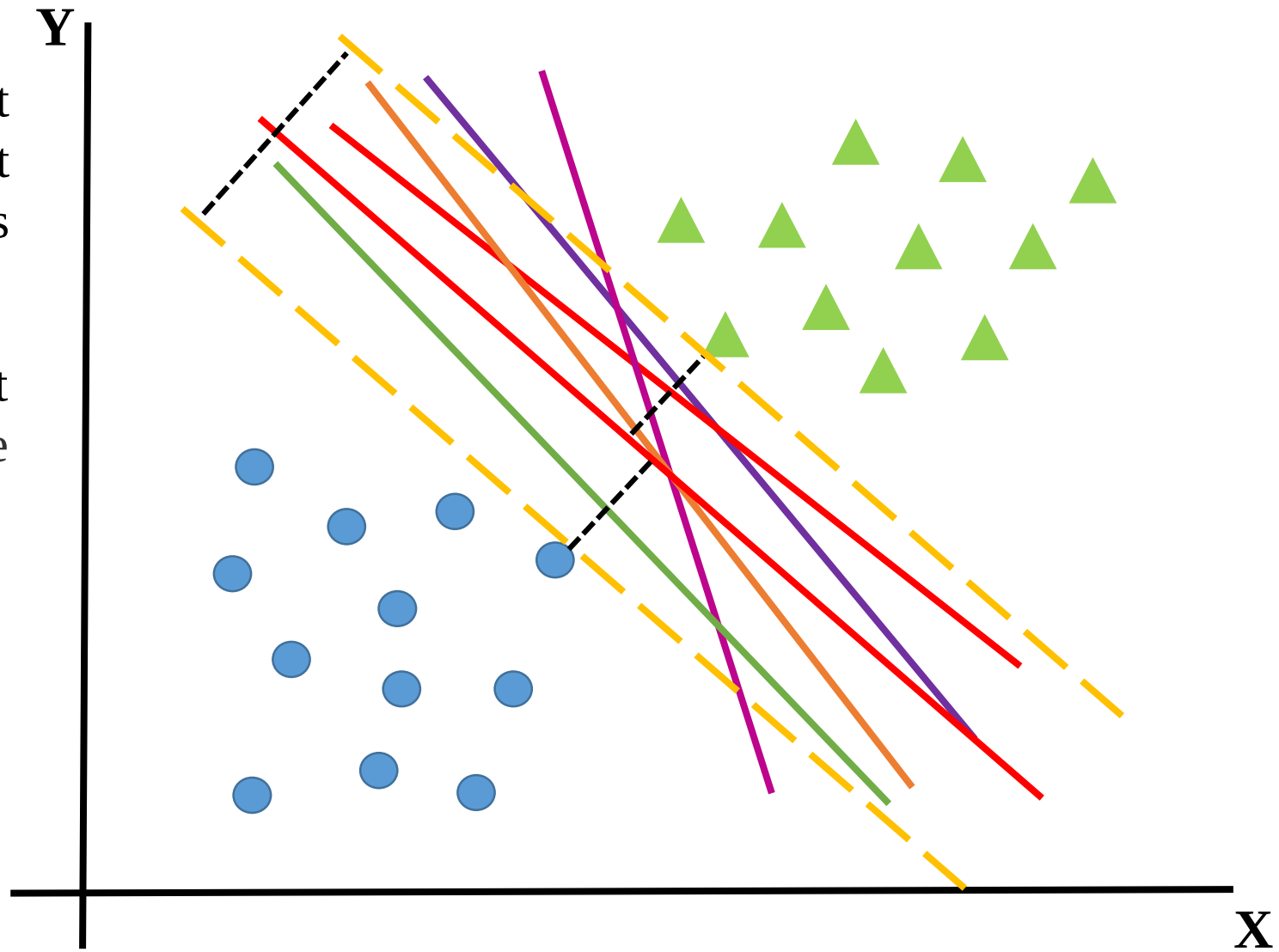

Y
X

Support Vector Machine

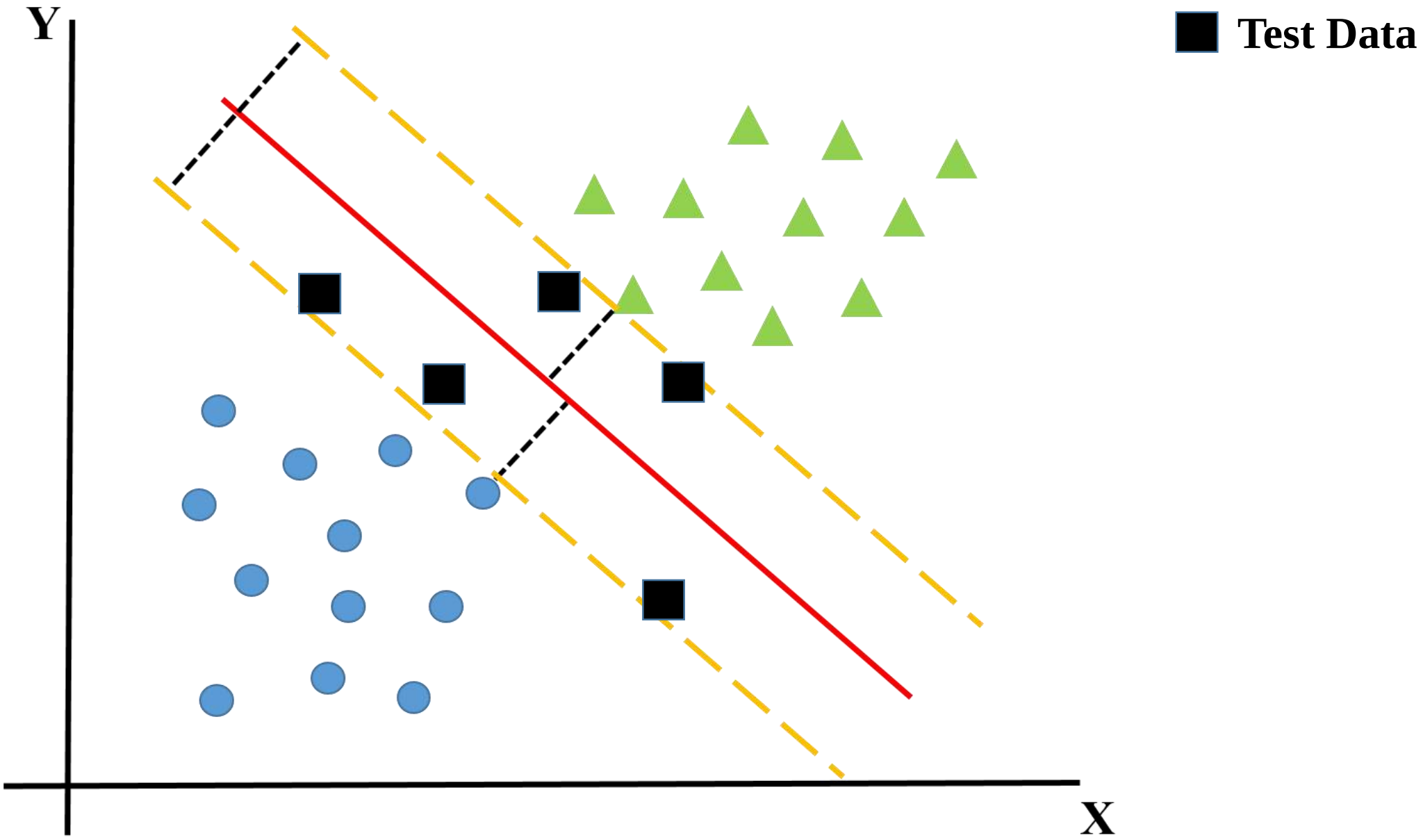
SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**.

SVM algorithm finds the closest point of the lines from both the classes. These points are called **support vectors**.

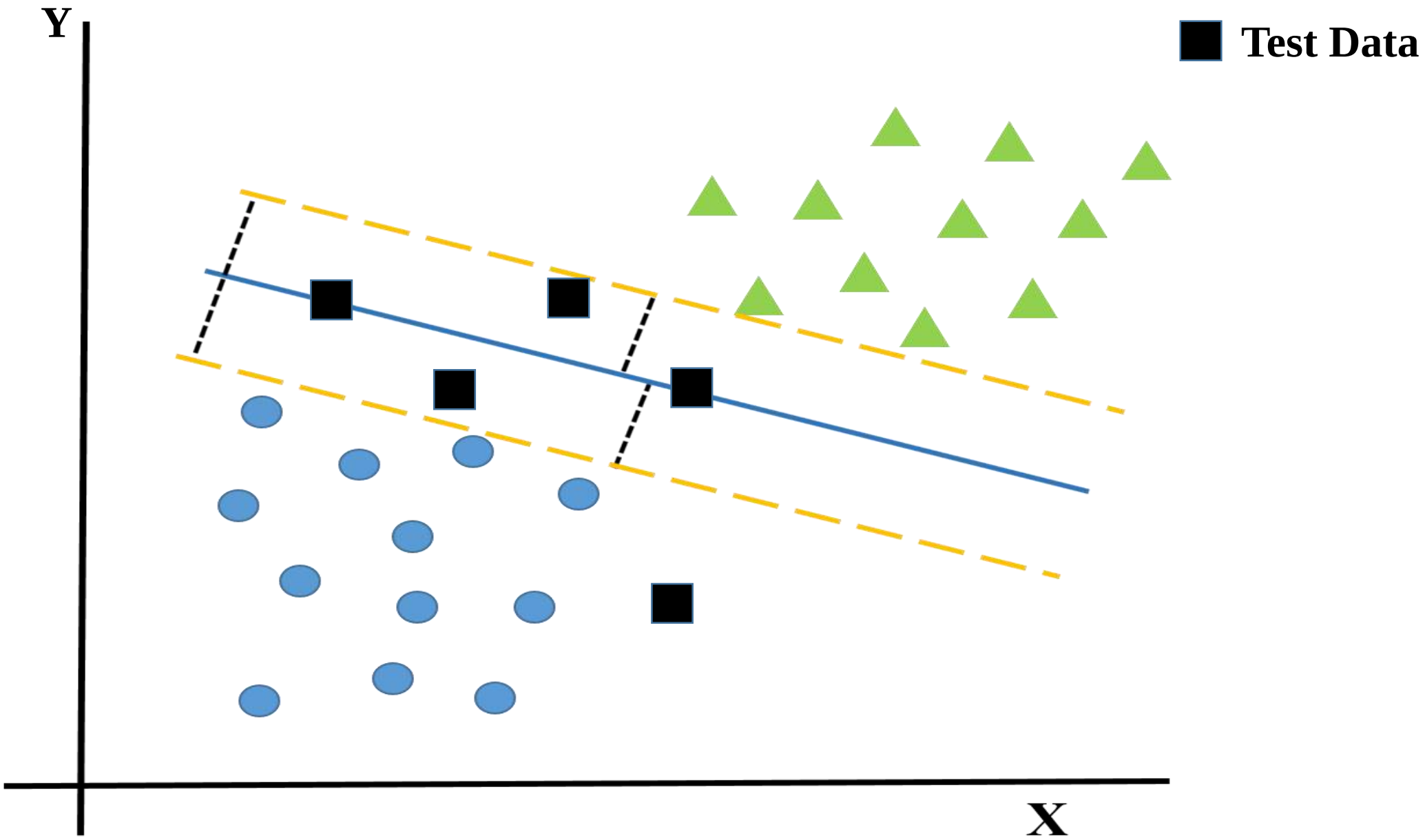
The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to **maximize this margin**.



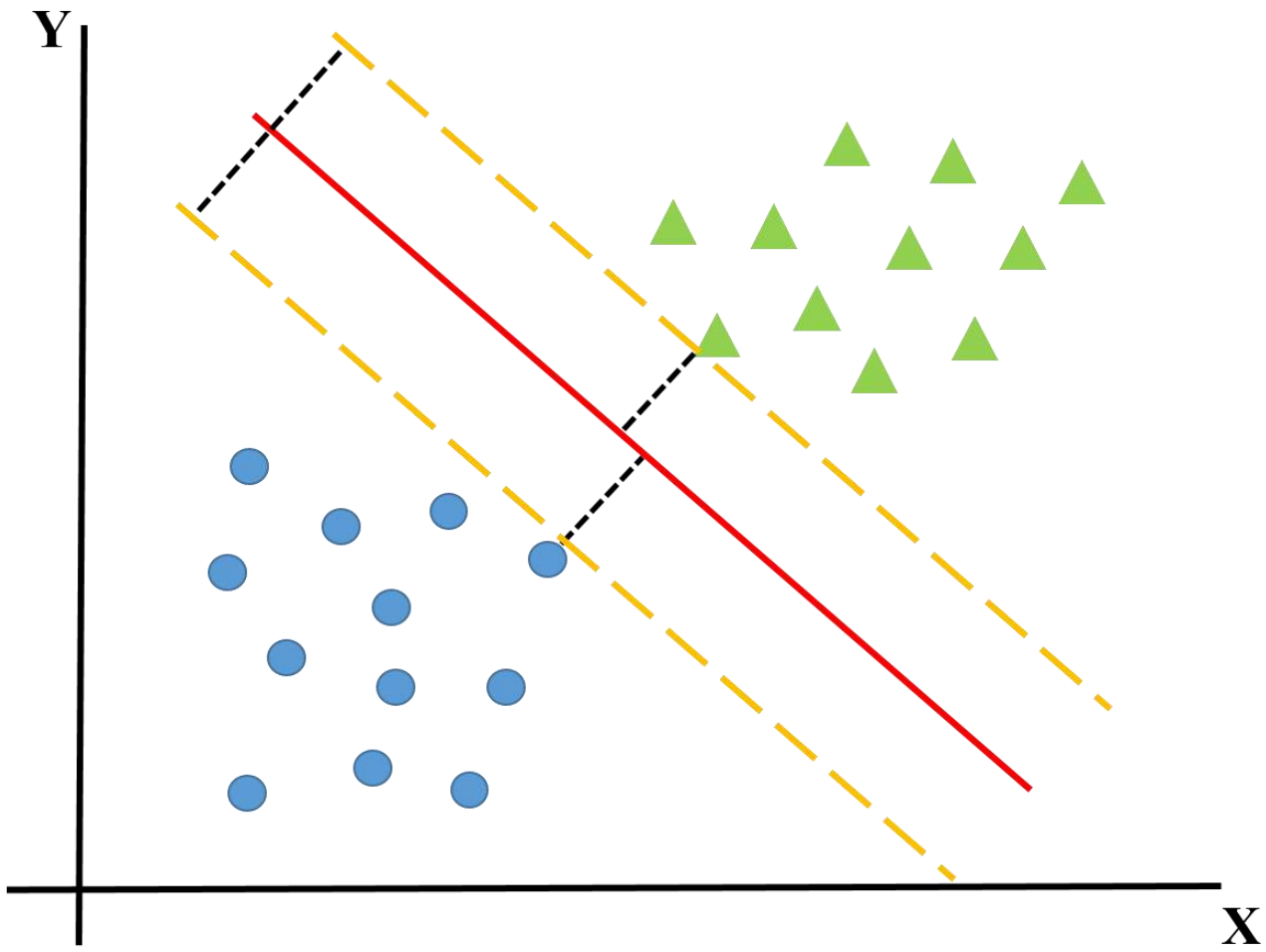
Support Vector Machine

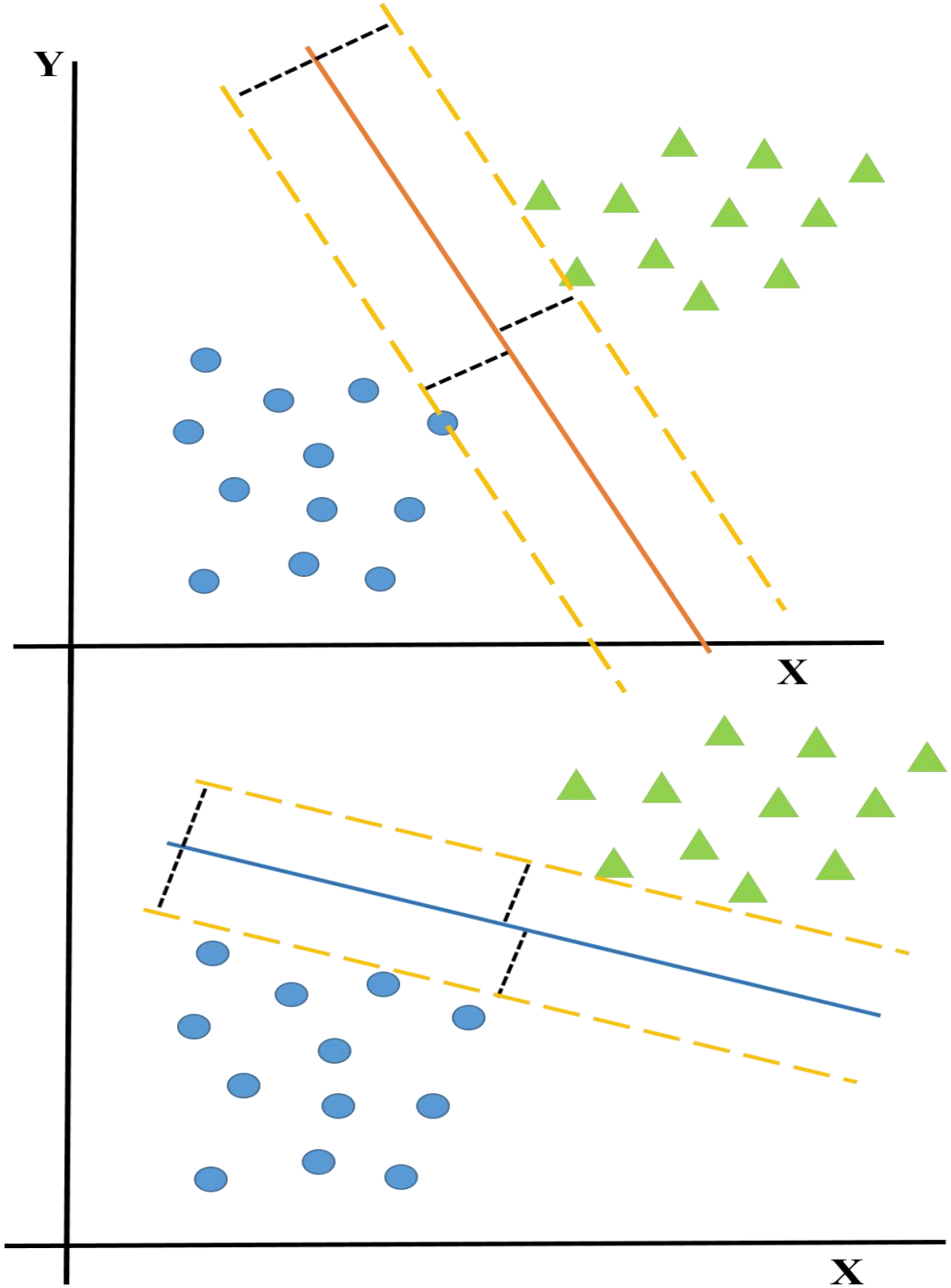


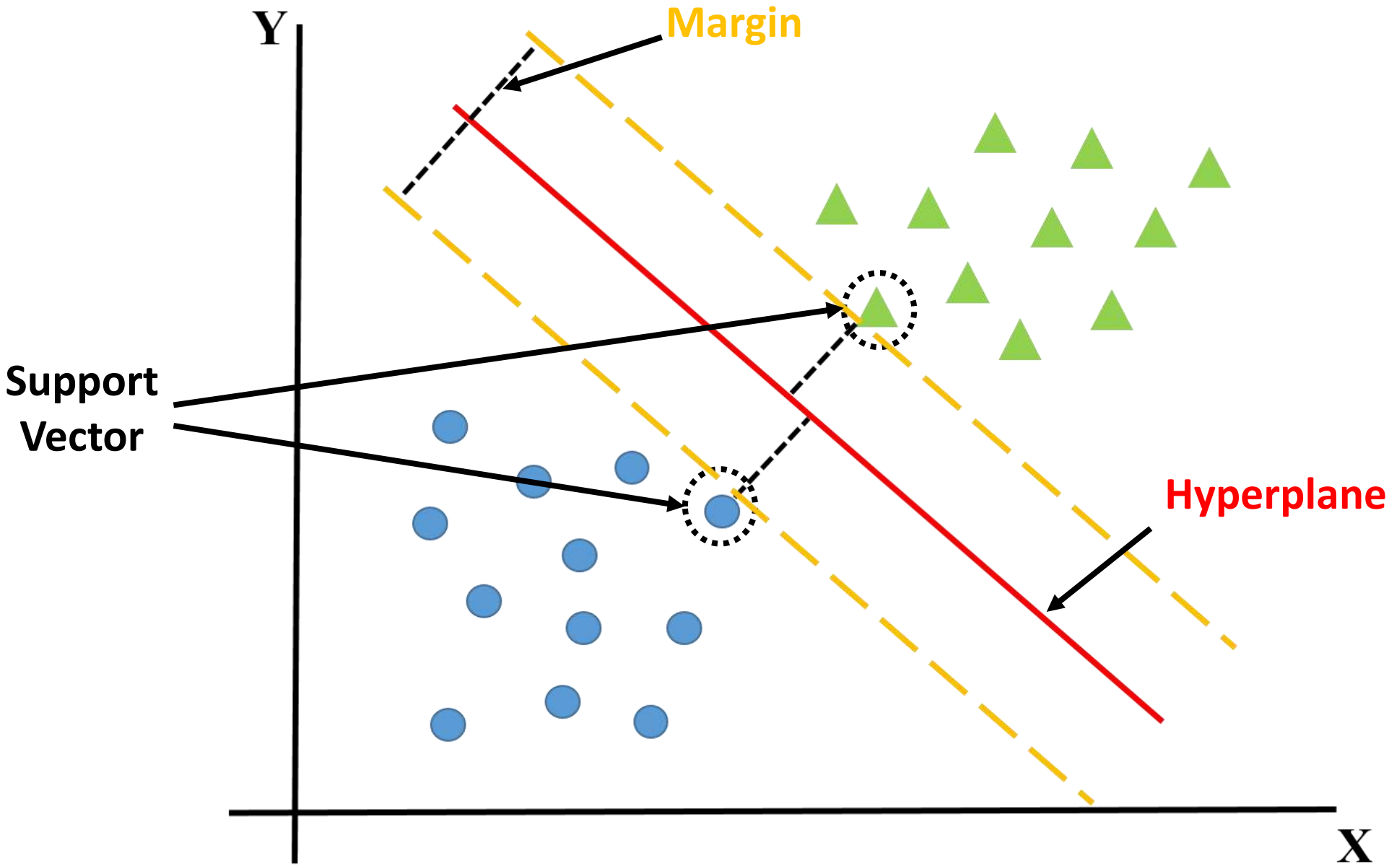
Support Vector Machine



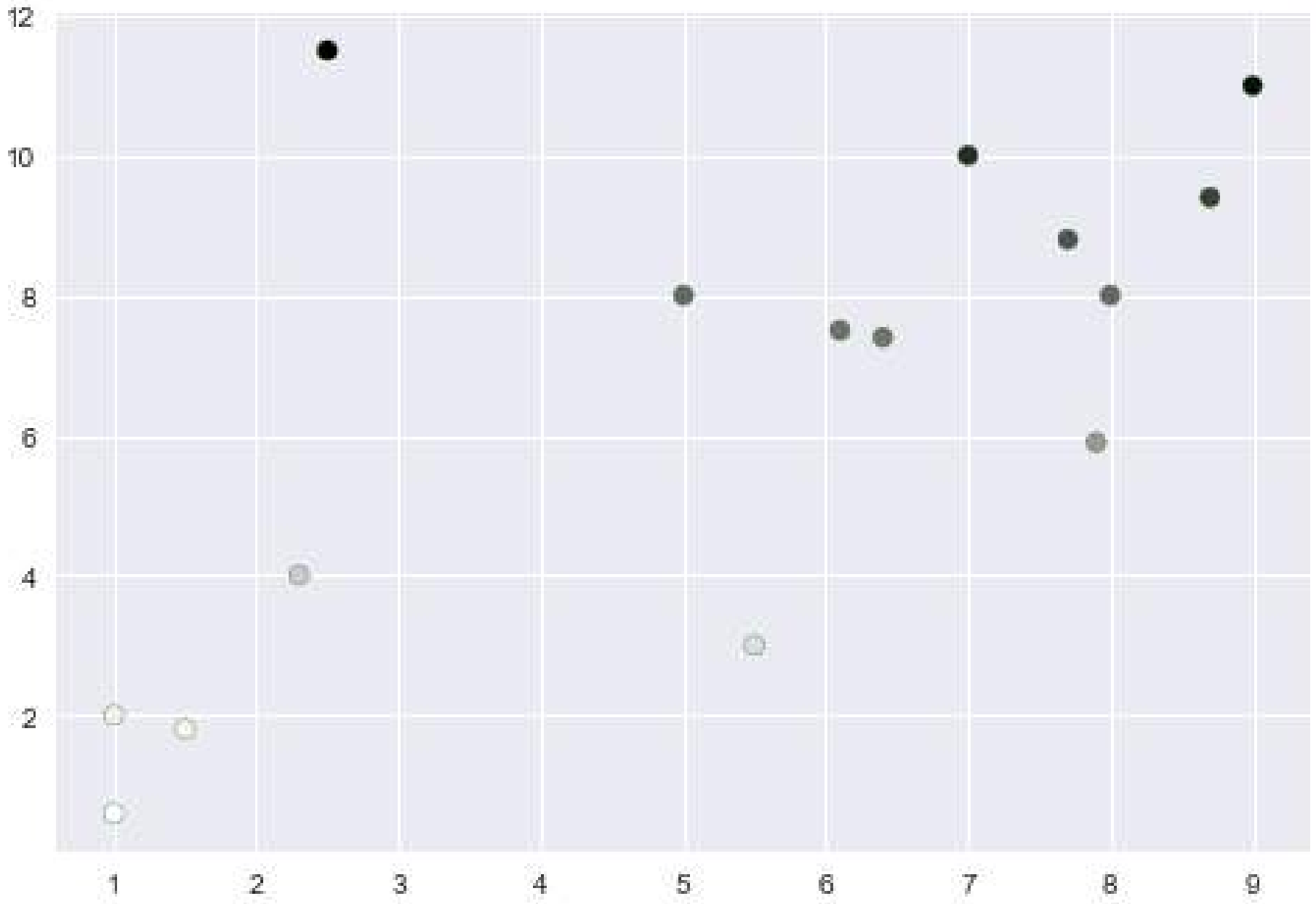
Support Vector Machine



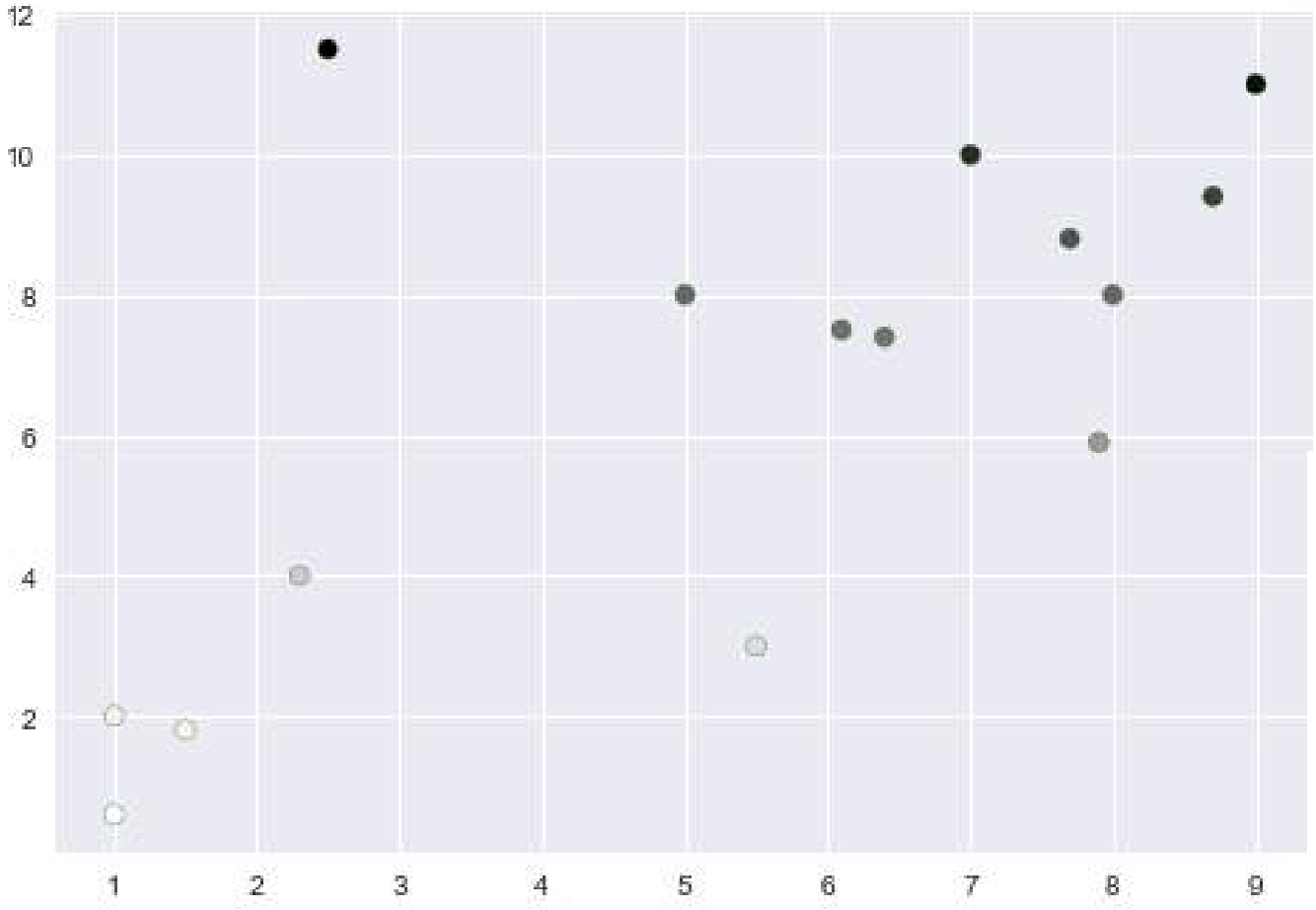


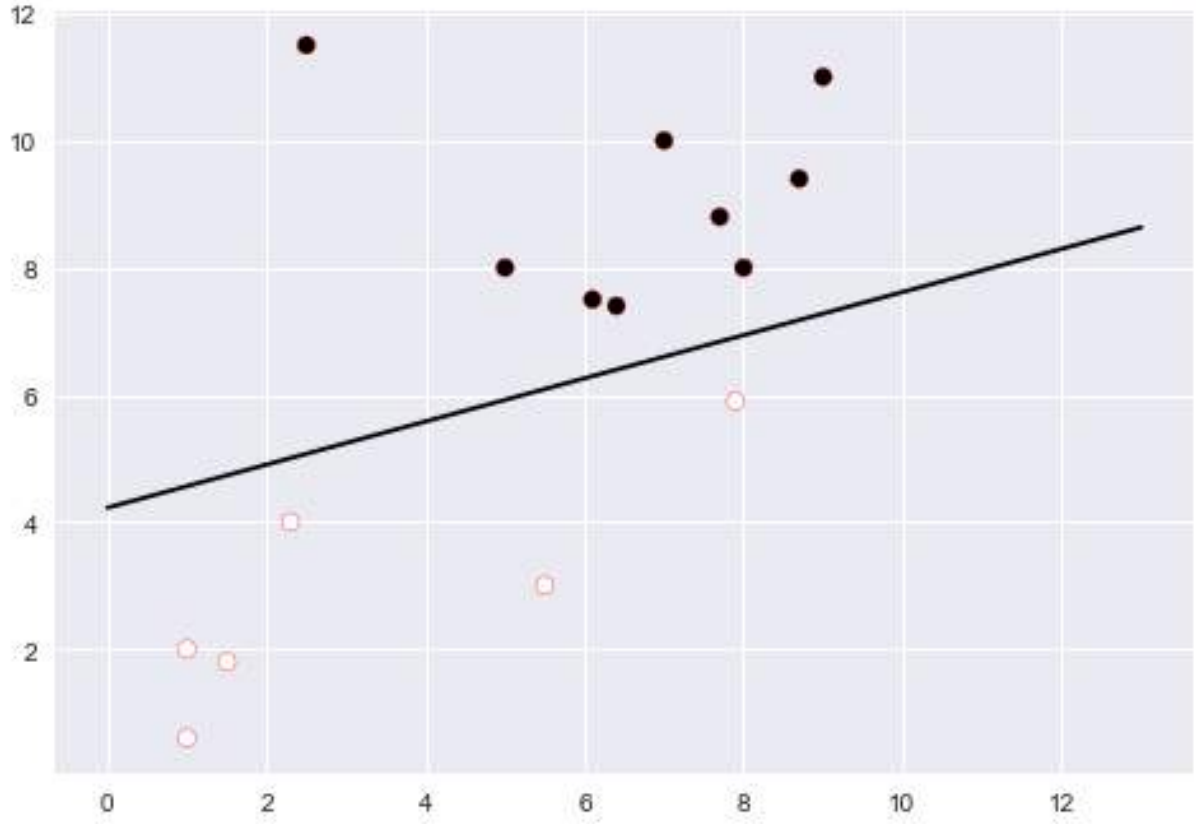

Y
Margin
Support
Vector
Hyperplane
X

Support Vector Machine

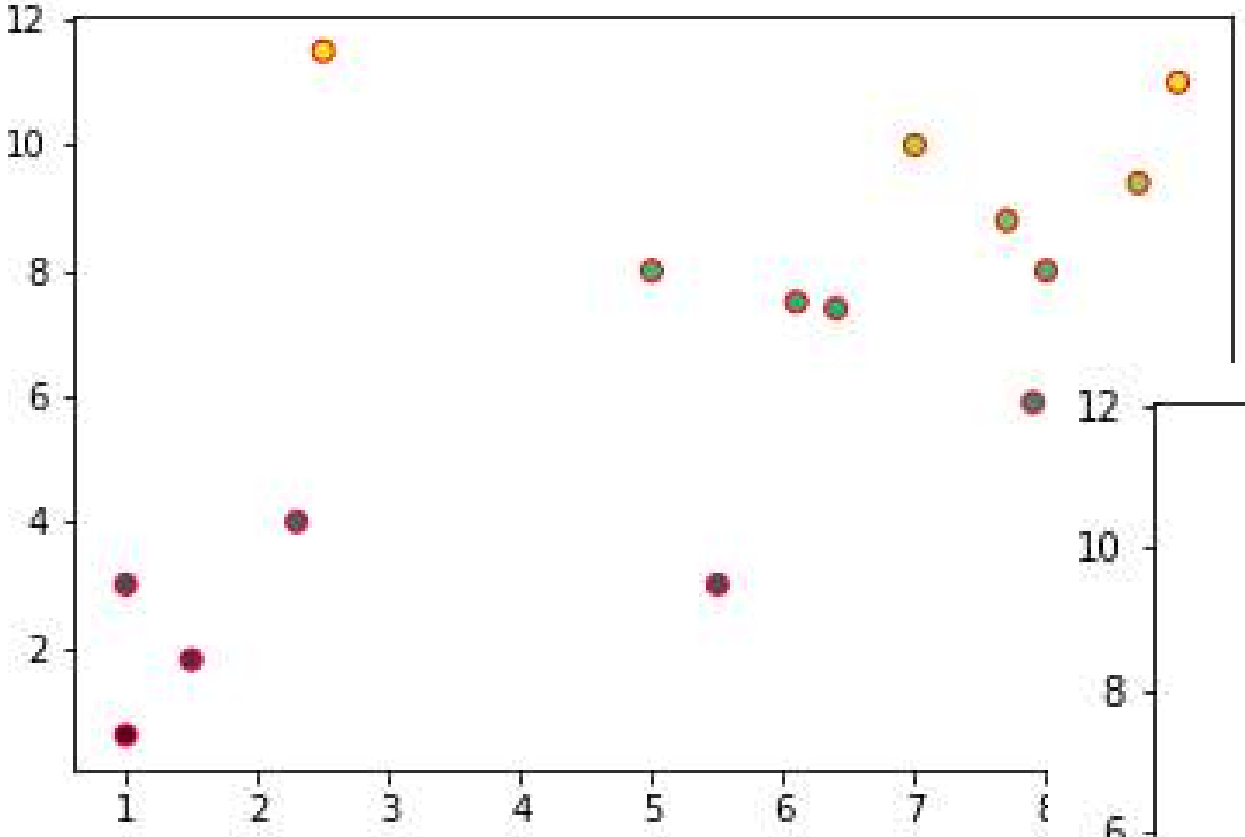


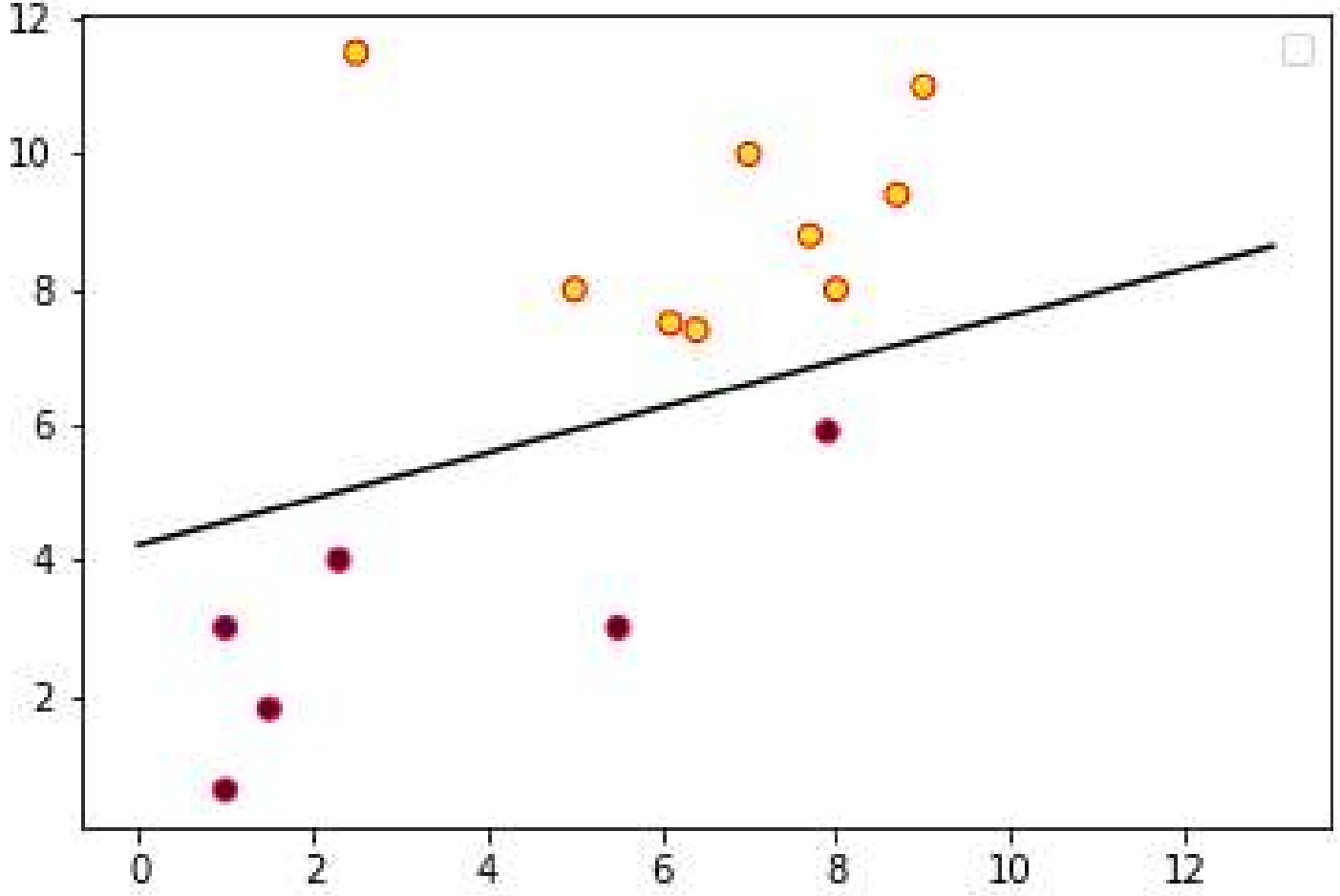
Support Vector Machine



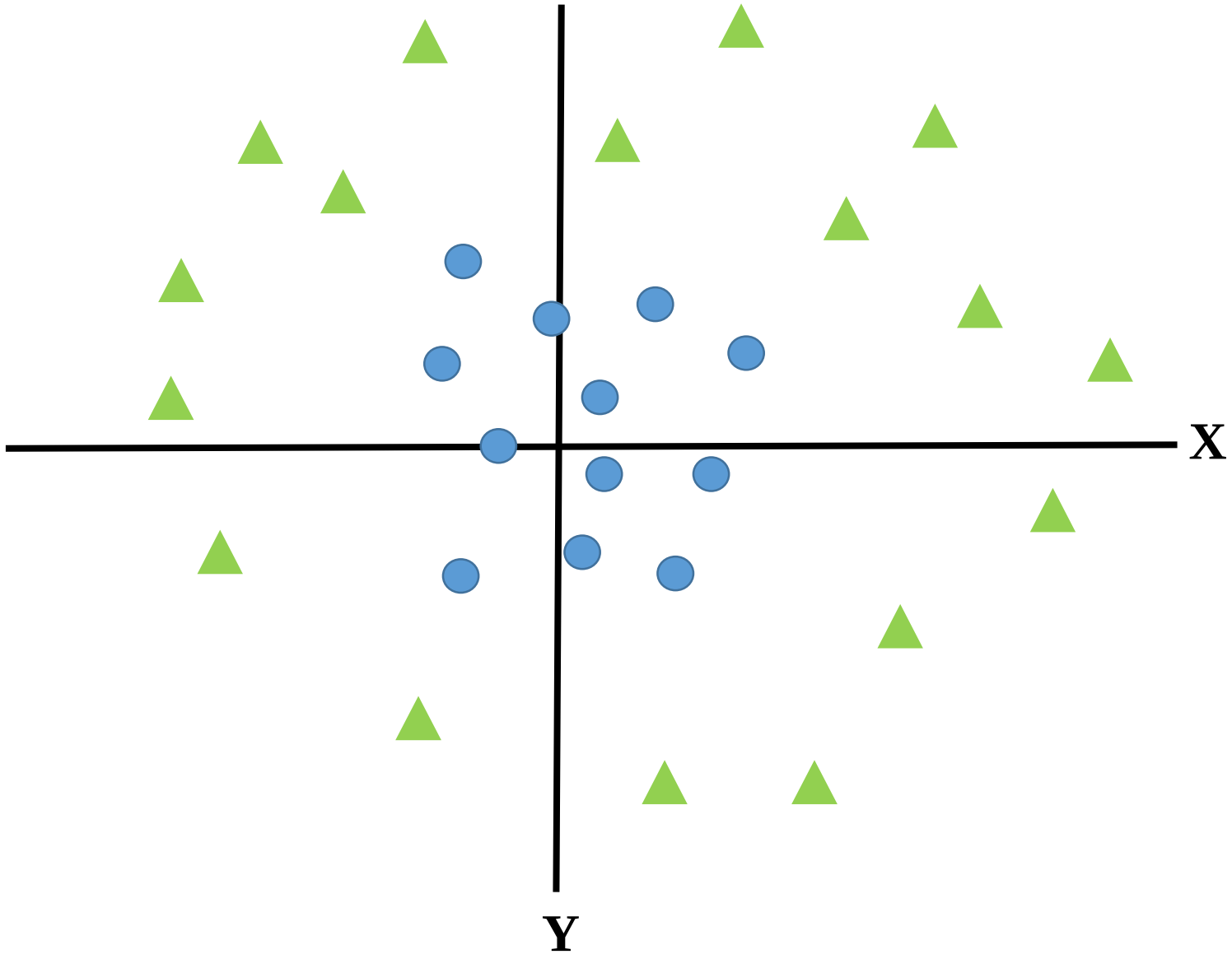


Support Vector Machine



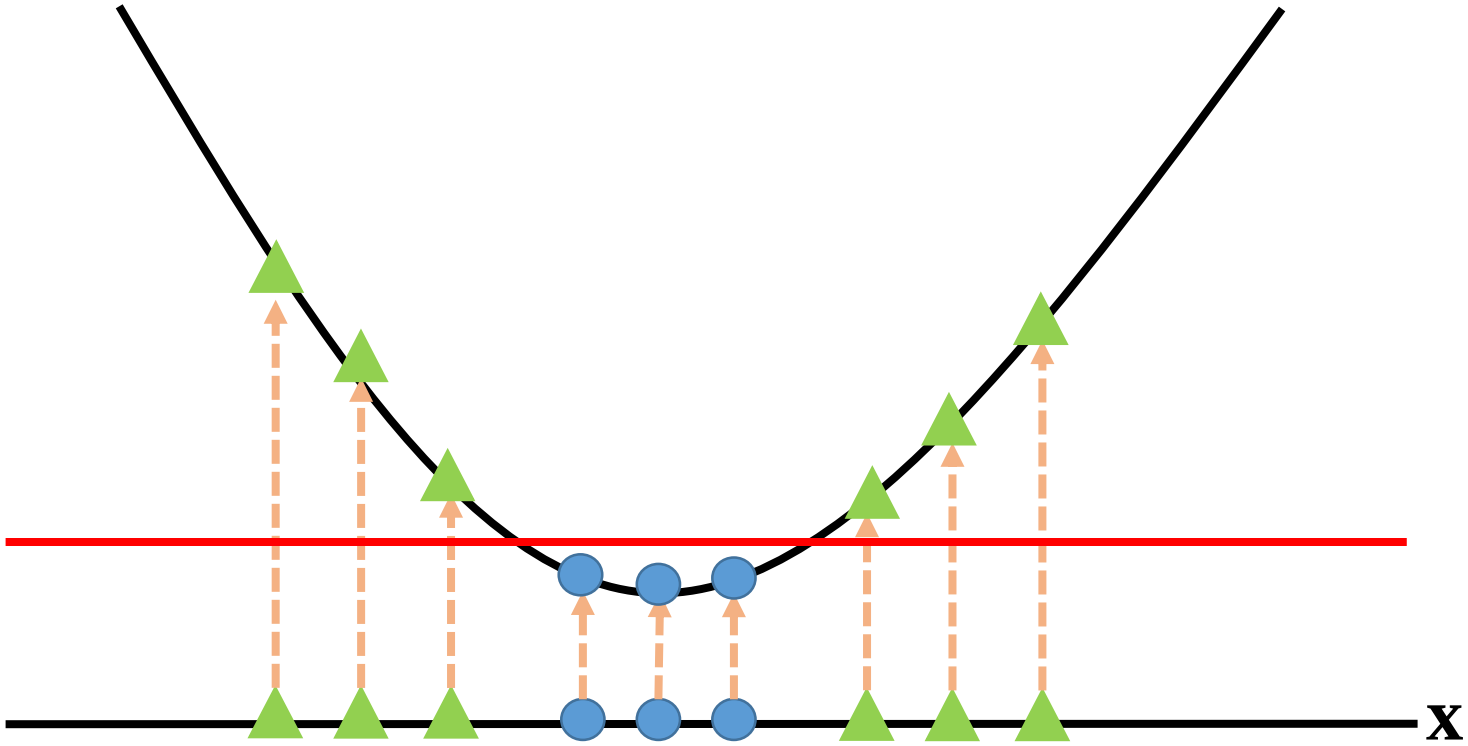


Support Vector Machine

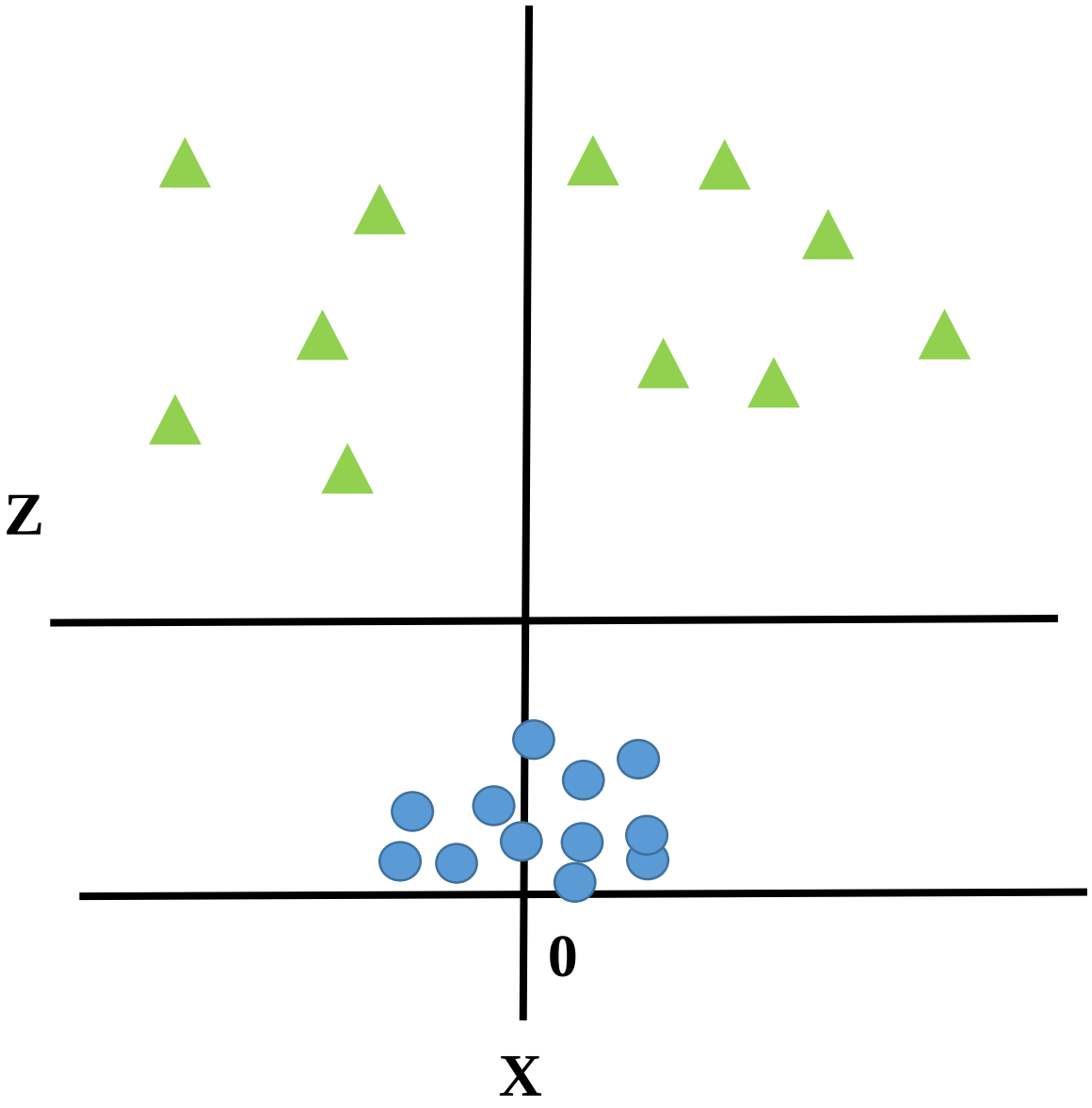


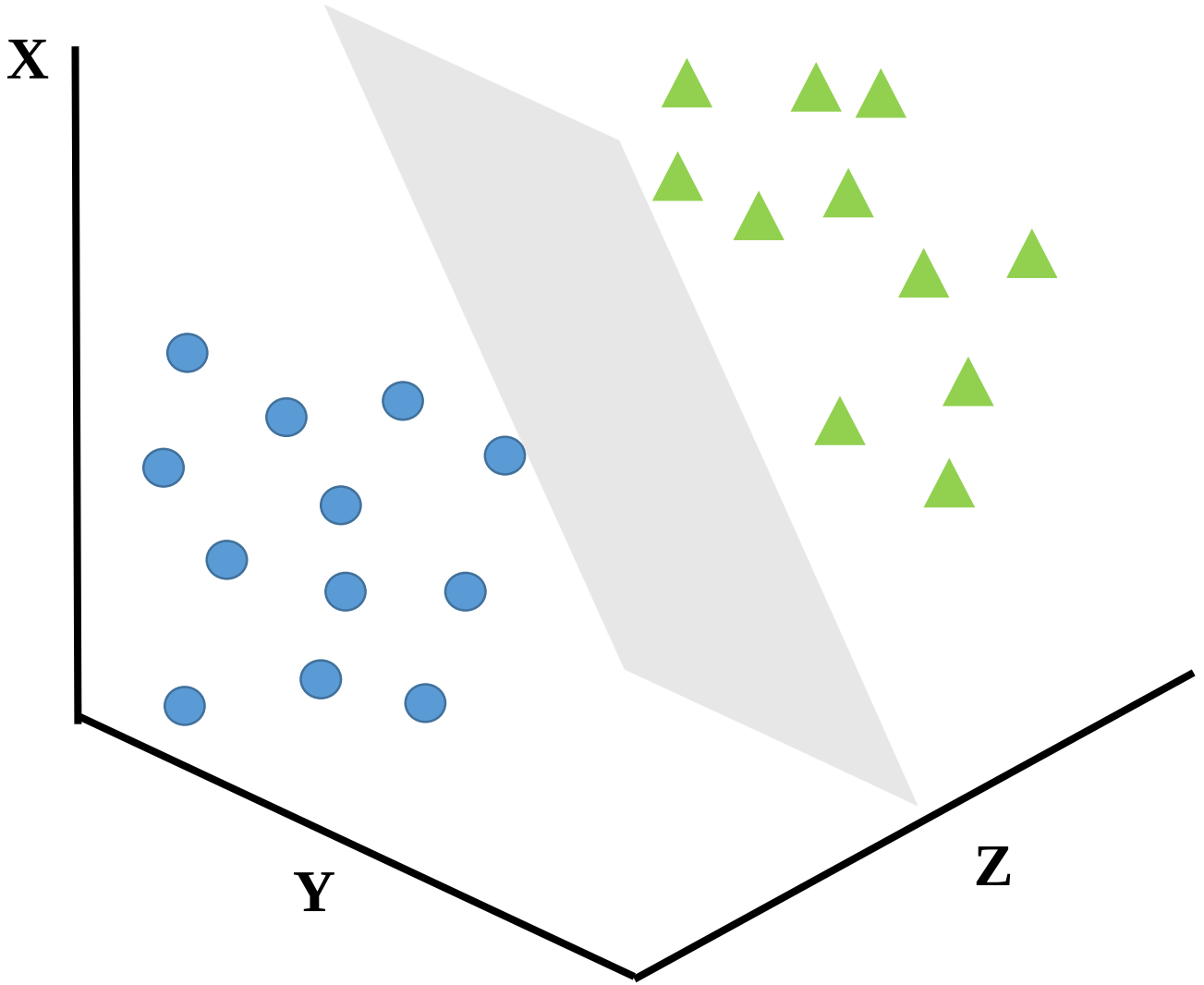
Support Vector Machine Kernel

Converting Lower Dimension Data to Higher Dimension Data

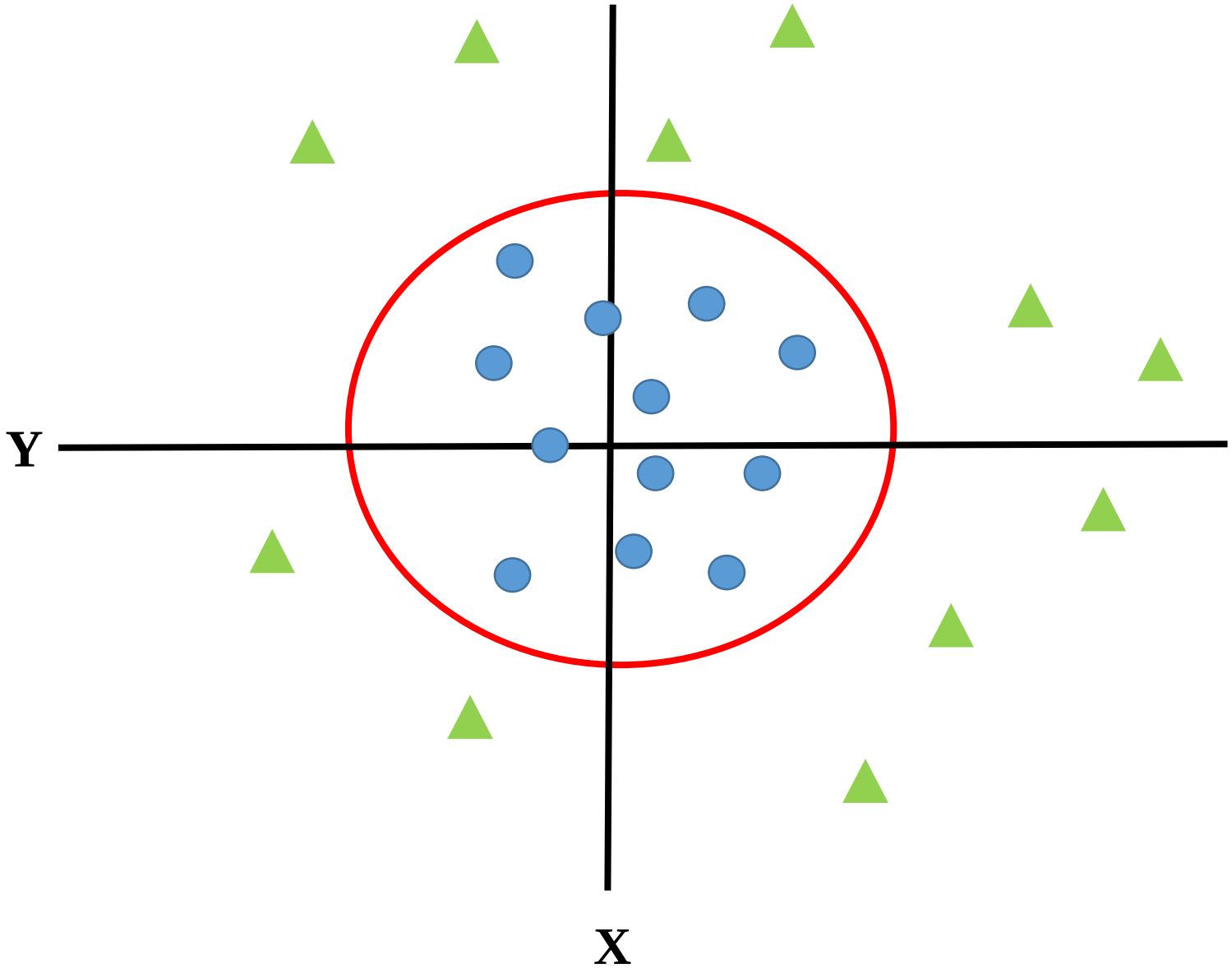


Support Vector Machine

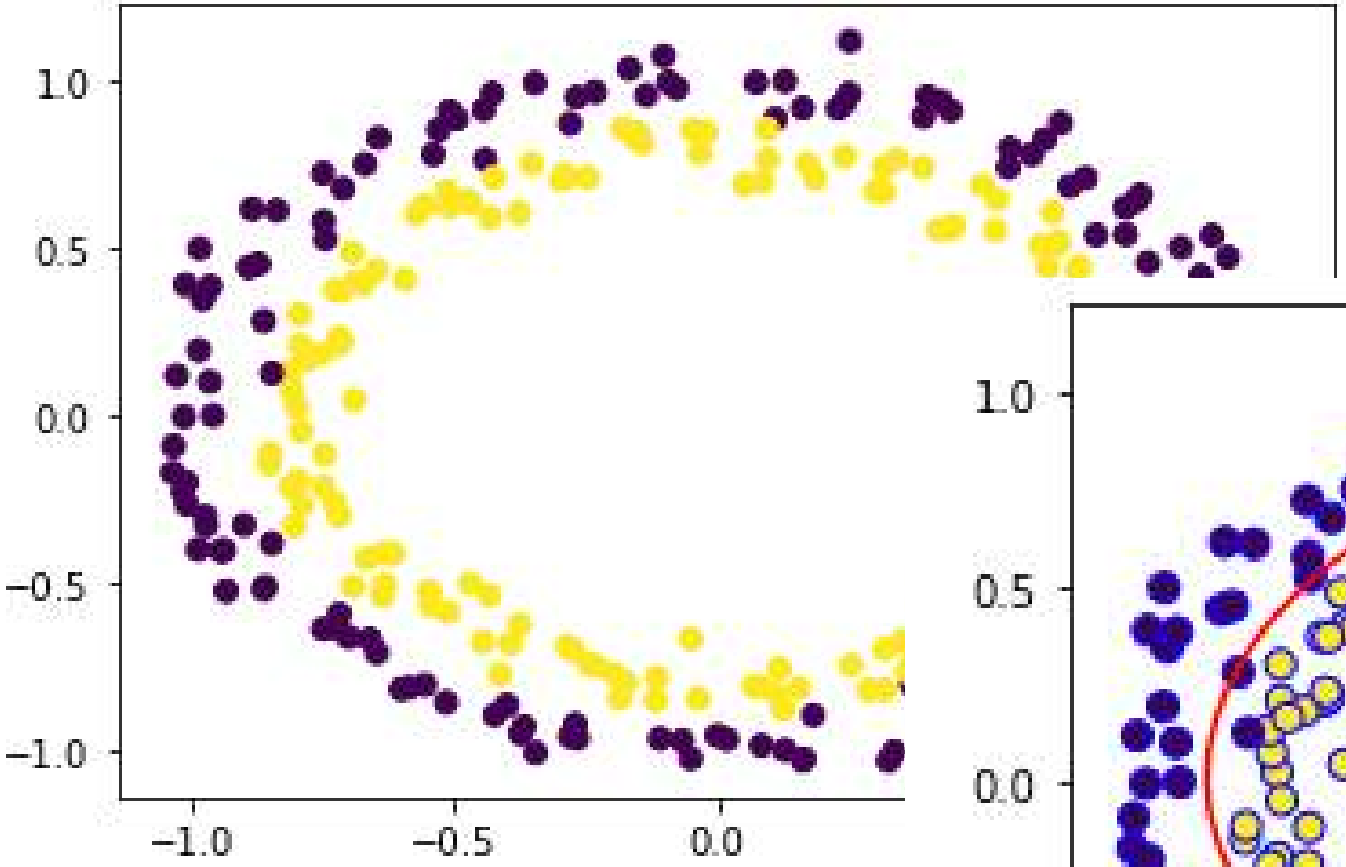


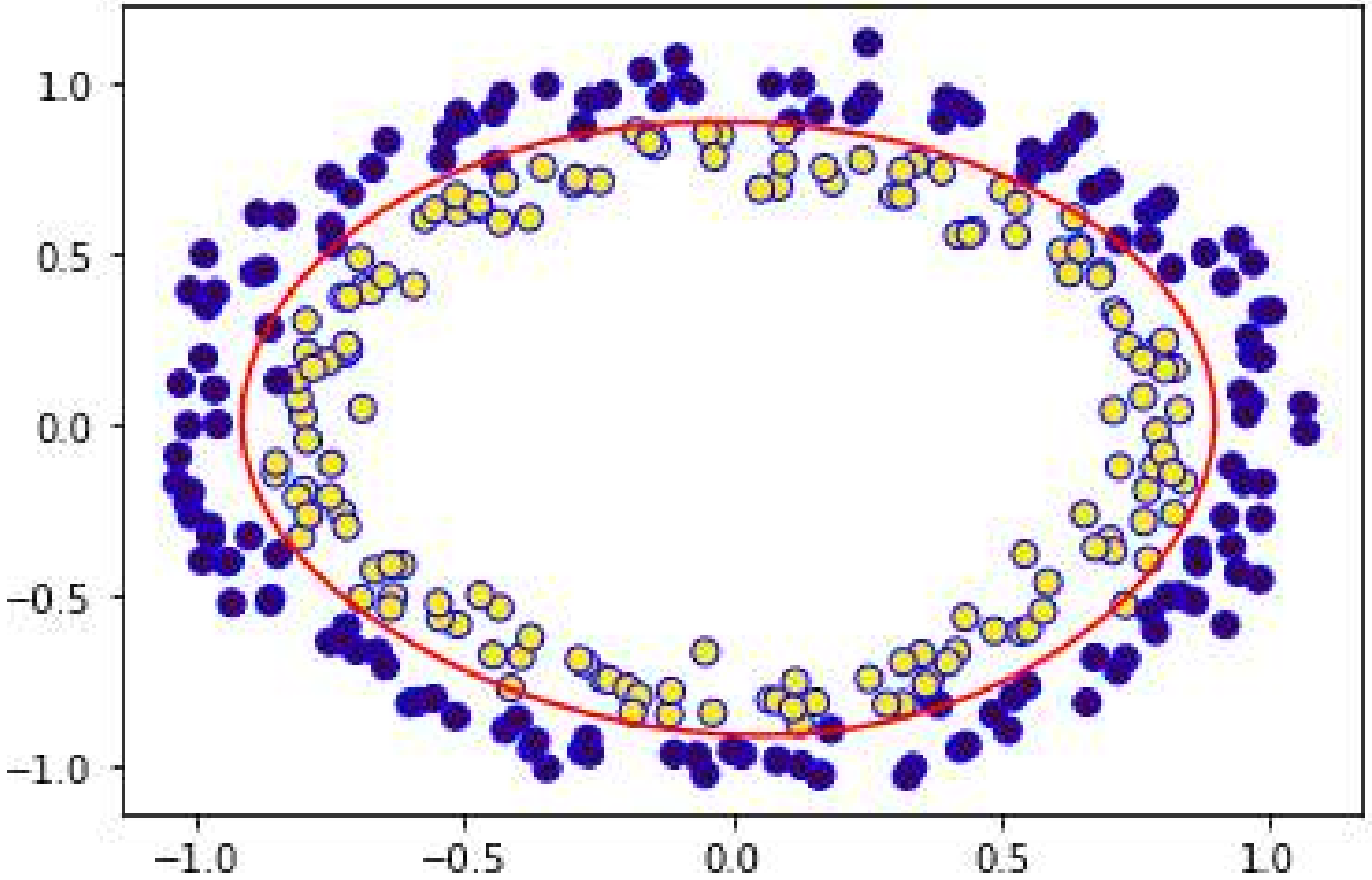


Support Vector Machine

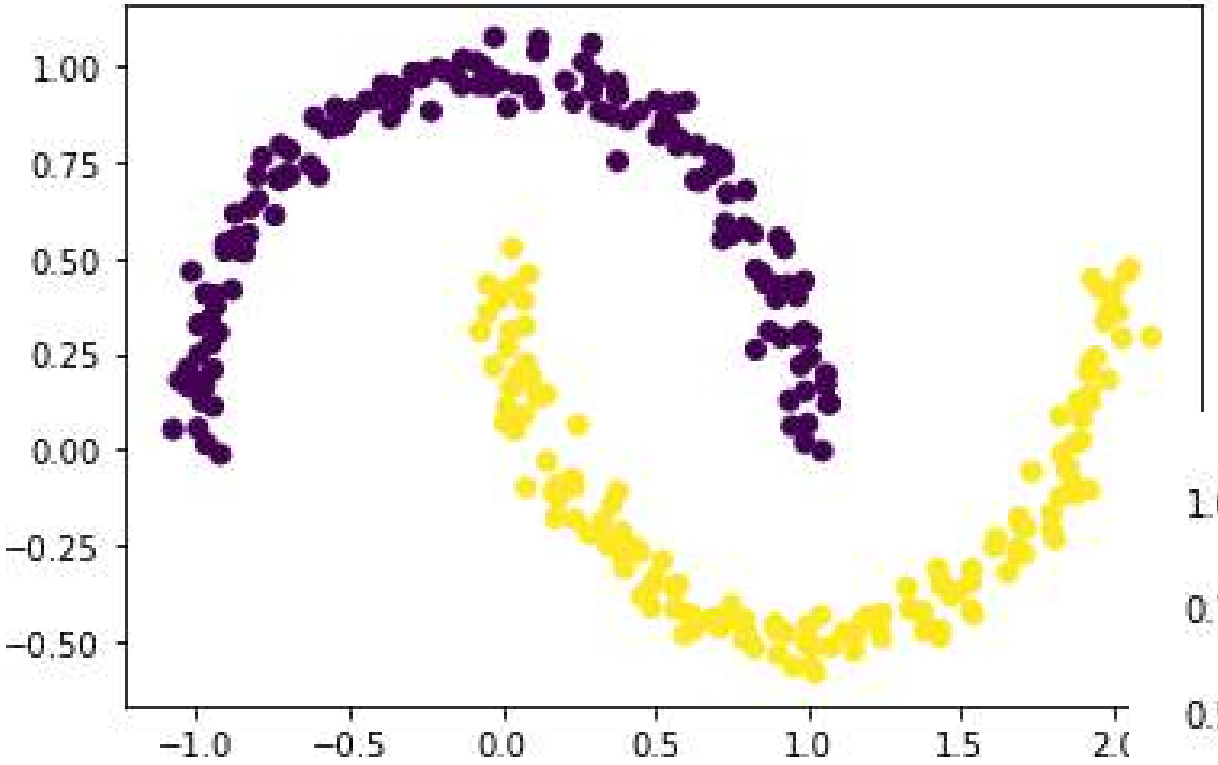


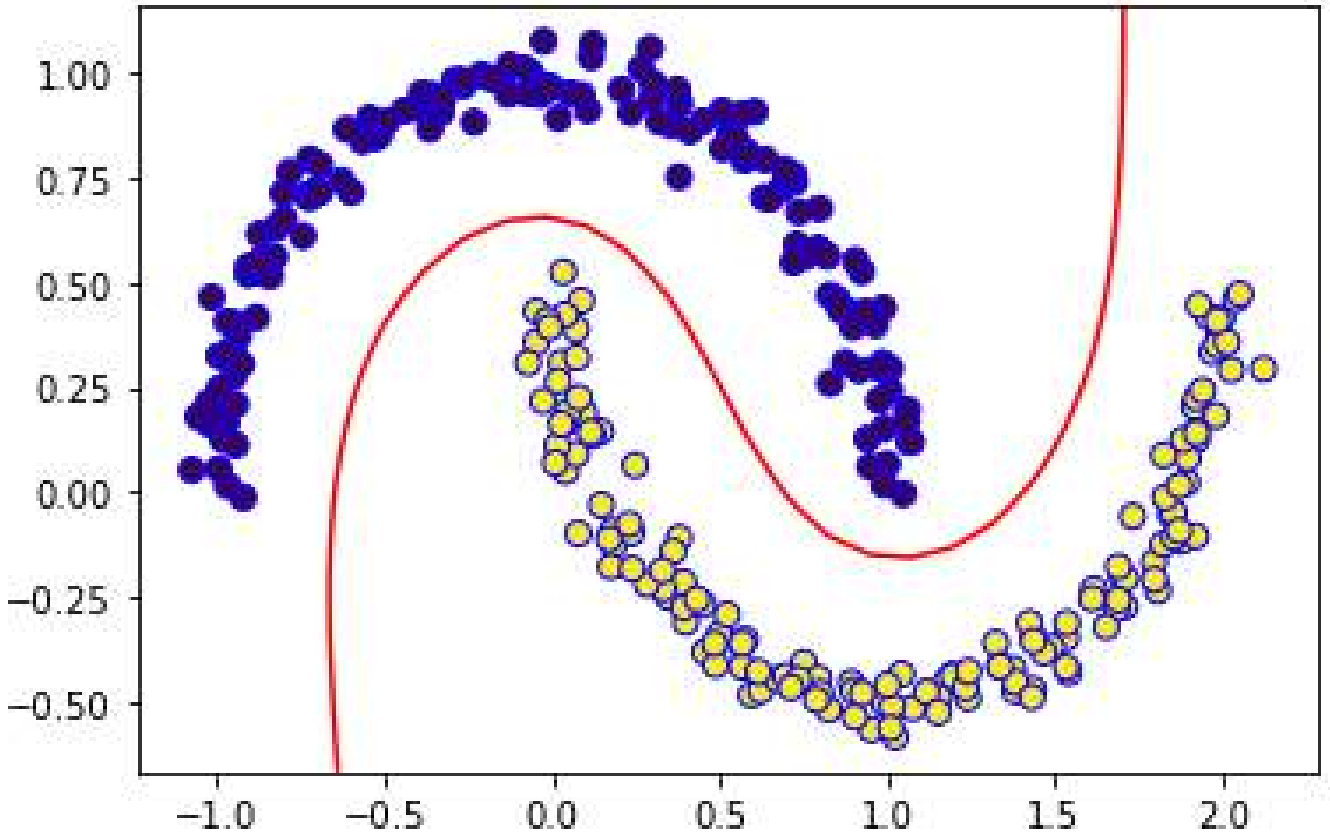
Support Vector Machine





Support Vector Machine





Support Vector Machine

Advantages of SVM

- Effective on datasets with multiple features, like financial or medical data.
- Effective in cases where number of features is greater than the number of data points.
- Uses a subset of training points in the decision function called support vectors which makes it memory efficient.
- Different kernel functions can be specified for the decision function. You can use common kernels, but it's also possible to specify custom kernels.

Disadvantages of SVM

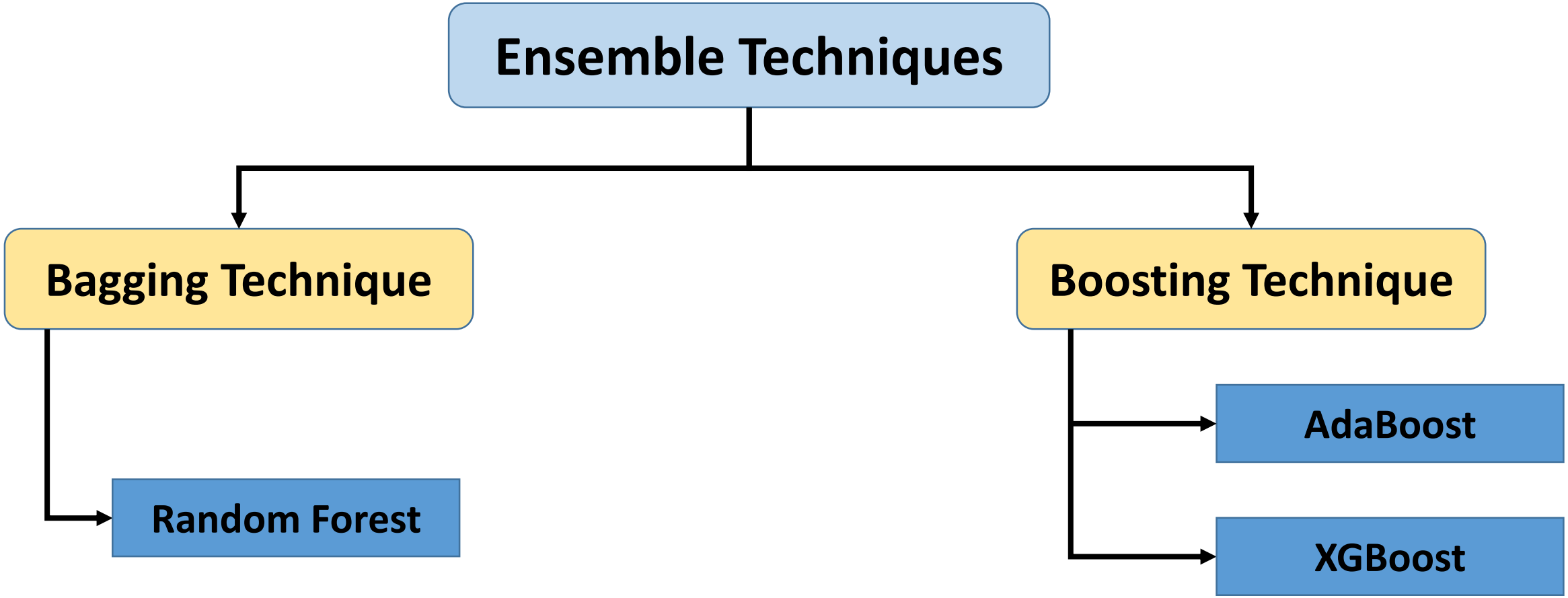
- If the number of features is a lot bigger than the number of data points, avoiding over-fitting when choosing kernel functions and regularization term is crucial.
- Works best on small sample sets because of its high training time.

Ensemble Learning

Ensemble Learning

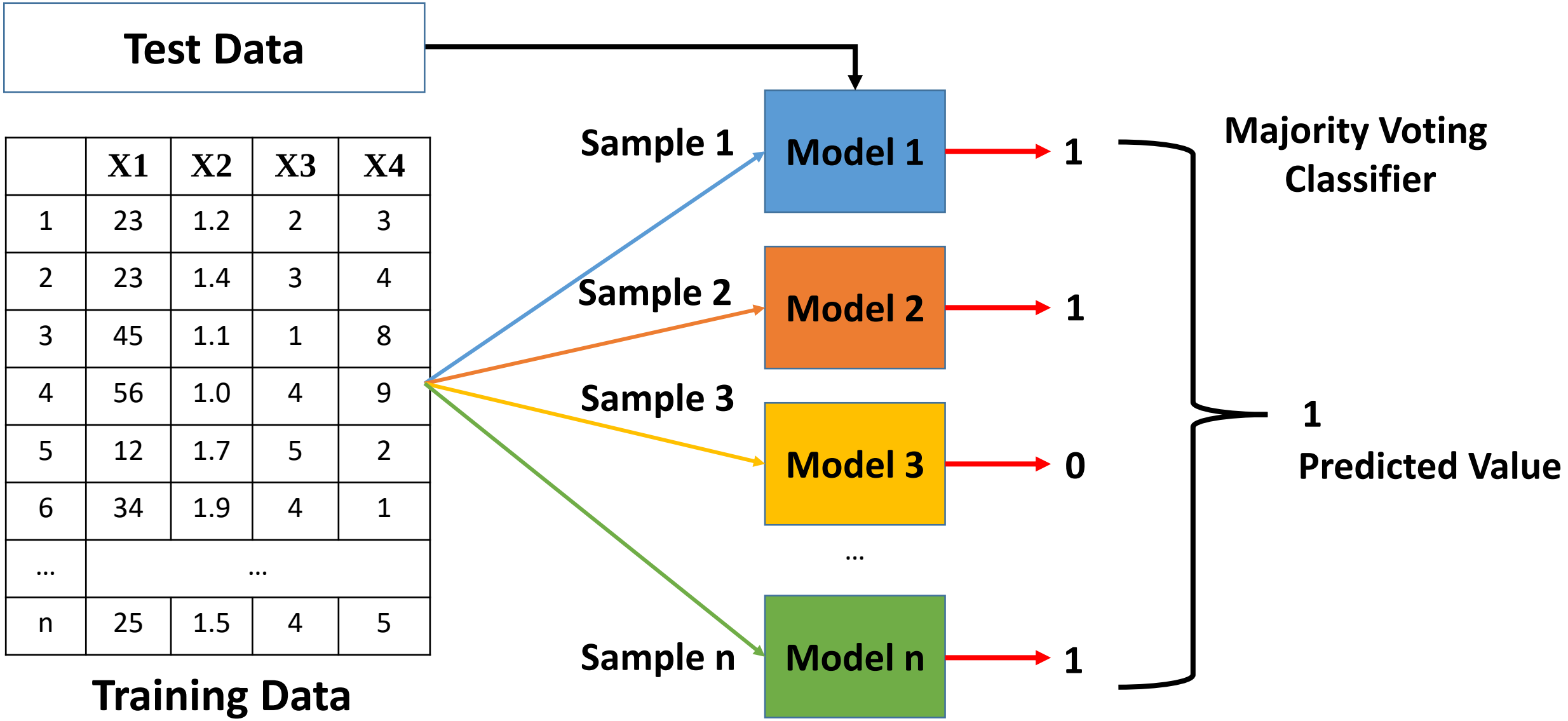
Ensemble Learning

Ensemble methods is a machine learning technique that **combines several base models** in order to produce **one optimal predictive model.**



Ensemble Learning

Bagging Ensemble Technique (Bootstrap Aggregation)



	X1	X2	X3	X4
1	23	1.2	2	3
2	23	1.4	3	4
3	45	1.1	1	8
4	56	1.0	4	9
5	12	1.7	5	2
6	34	1.9	4	1
…	…			
n	25	1.5	4	5

Training Data

Random Forest

Random Forest works in 2 phases:

1. First is to create the random forest by combining N number of decision trees
2. Second is to make predictions for each tree created in the first phase.

Algorithm

Step-1: Select random K data points from the training set using Row Selection with Replacement and Feature Selection with Replacement.

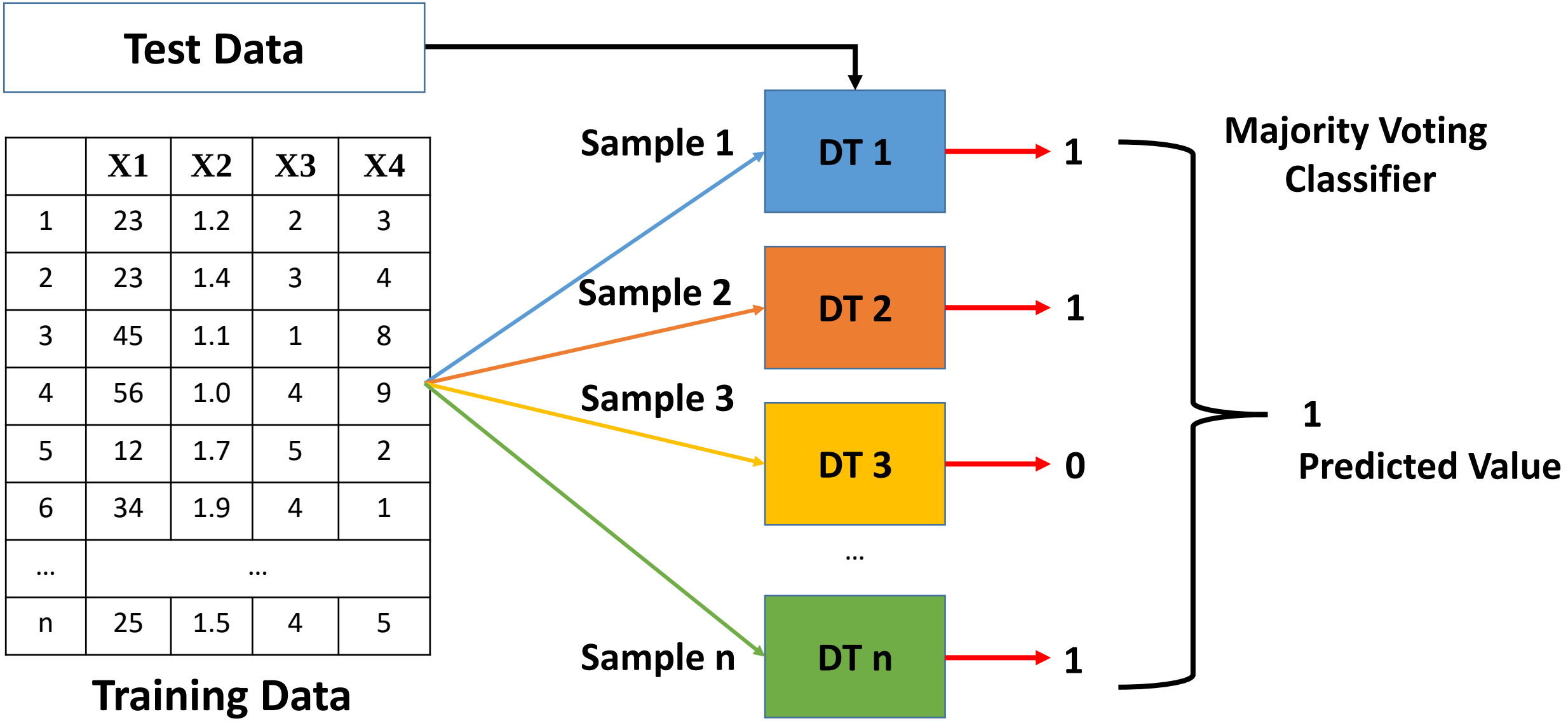
Step-2: Build the decision trees associated with the selected data points (Bootstrap Samples).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Bagging Ensemble

Random Forest (Classification Problem)

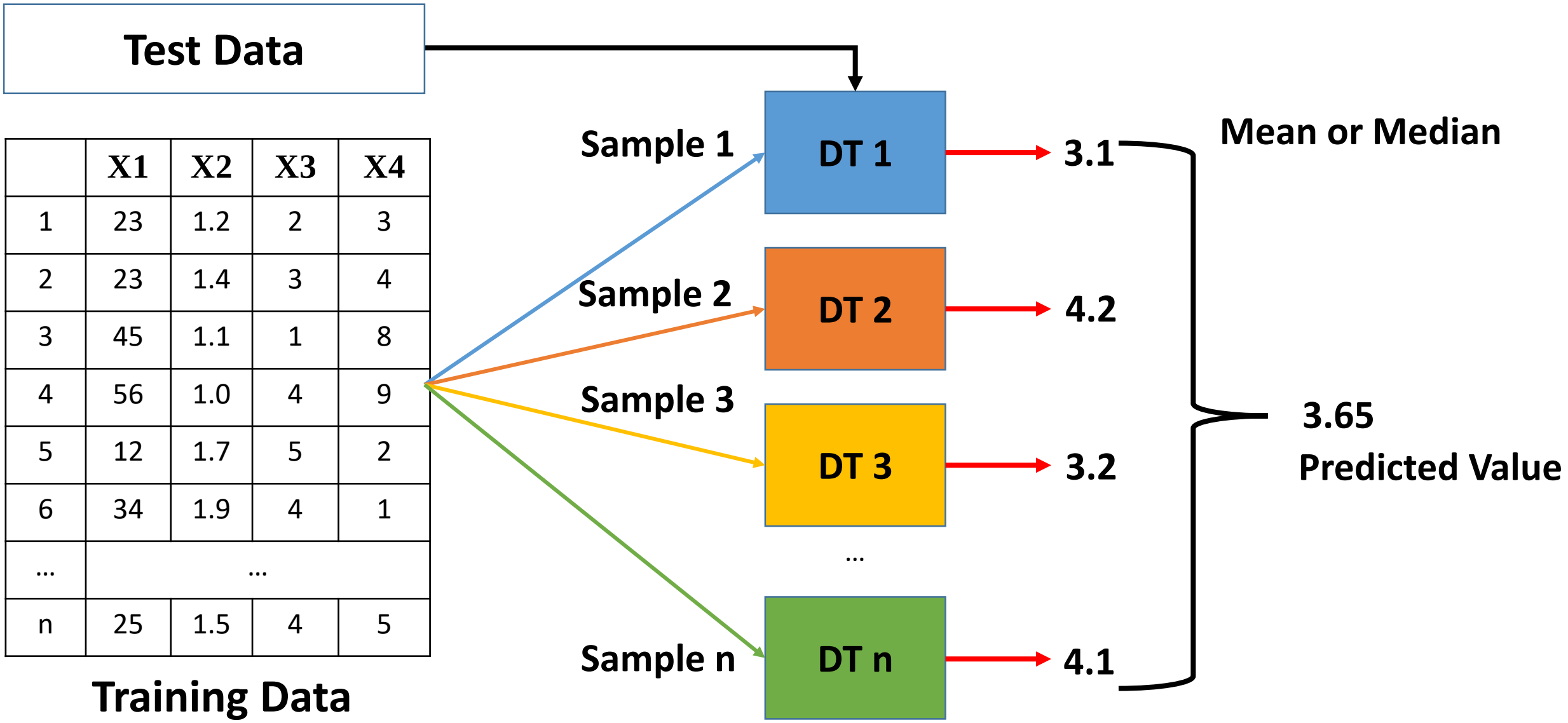


	X1	X2	X3	X4
1	23	1.2	2	3
2	23	1.4	3	4
3	45	1.1	1	8
4	56	1.0	4	9
5	12	1.7	5	2
6	34	1.9	4	1
...	...			
n	25	1.5	4	5

Training Data

Bagging Ensemble

Random Forest (Regression Problem)

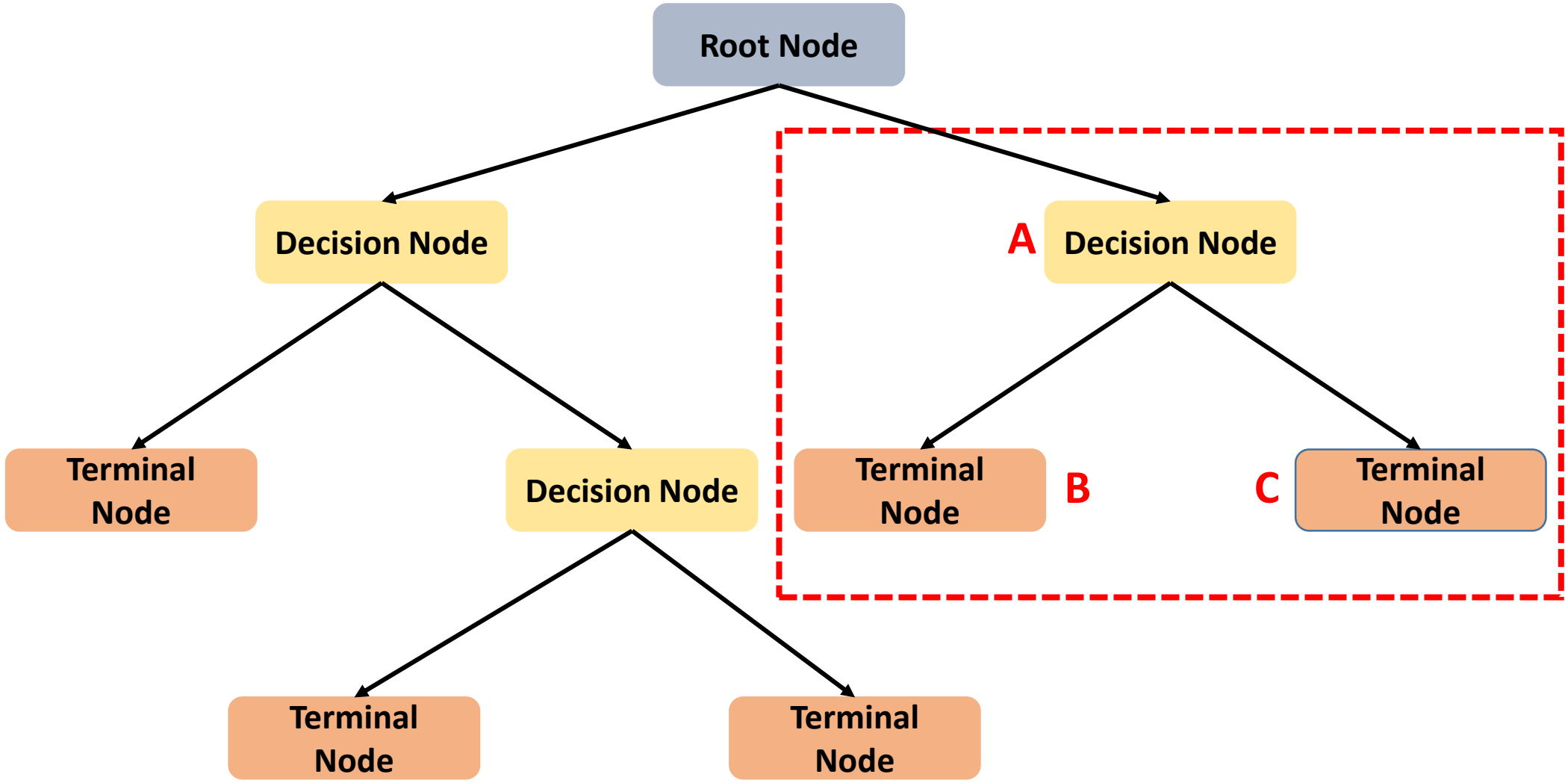


	X1	X2	X3	X4
1	23	1.2	2	3
2	23	1.4	3	4
3	45	1.1	1	8
4	56	1.0	4	9
5	12	1.7	5	2
6	34	1.9	4	1
...	...			
n	25	1.5	4	5

Training Data

Bagging Ensemble

Decision Tree

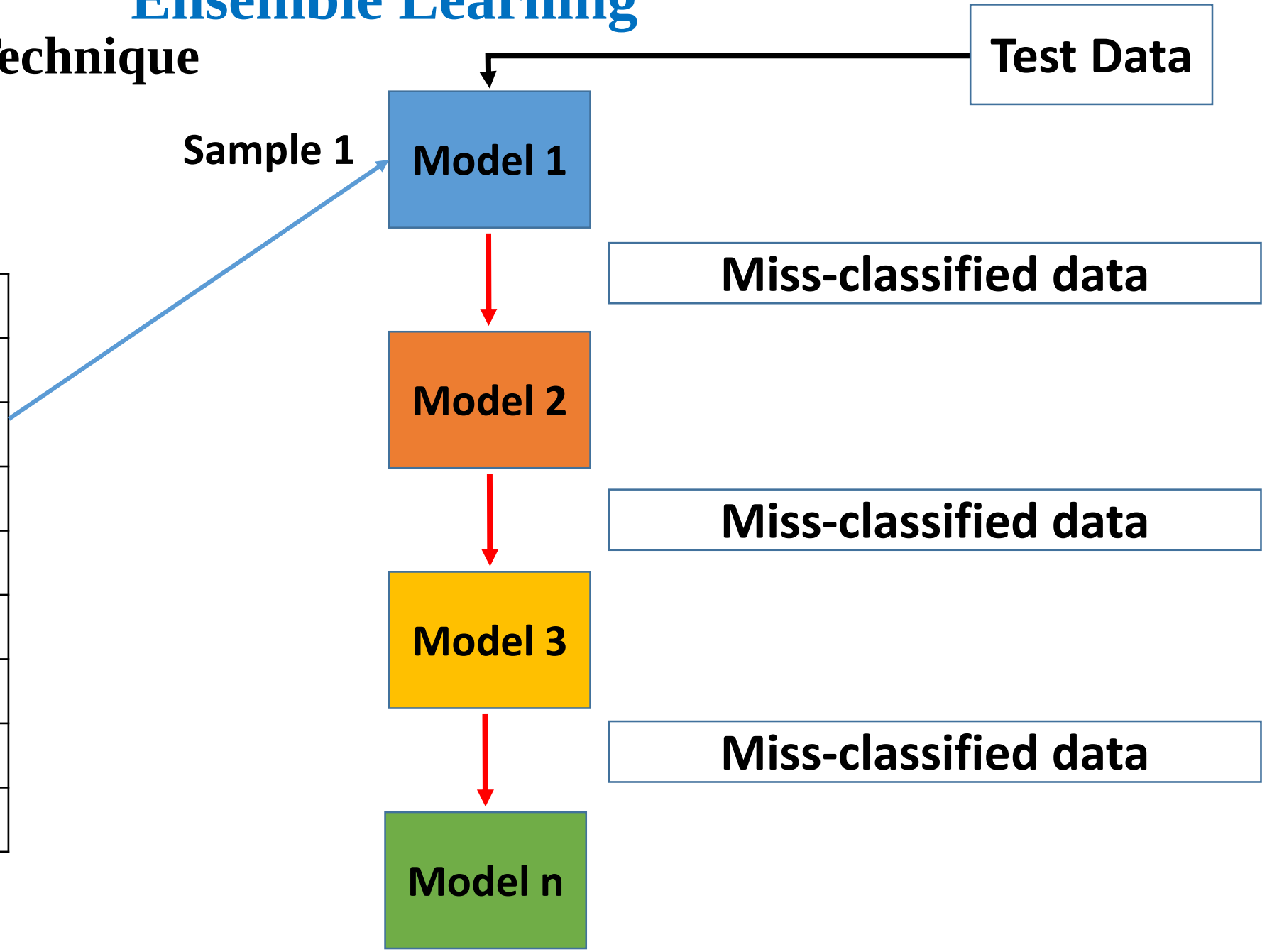


Ensemble Learning

Boosting Ensemble Technique

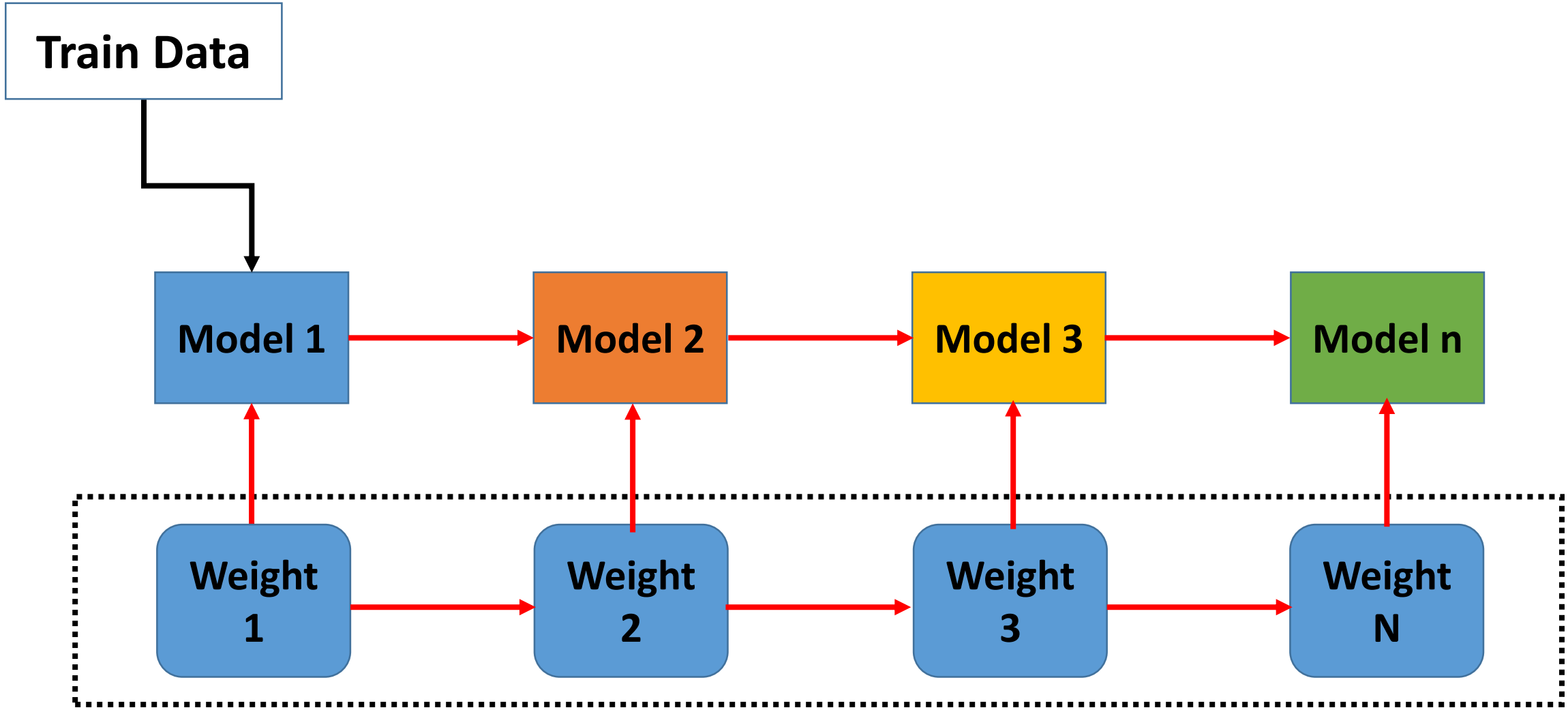
	X1	X2	X3	X4
1	23	1.2	2	3
2	23	1.4	3	4
3	45	1.1	1	8
4	56	1.0	4	9
5	12	1.7	5	2
6	34	1.9	4	1
…	…			
n	25	1.5	4	5

Training Data



Ensemble Learning

AdaBoosting Ensemble Technique



Ensemble Learning

Algorithm

Step-1: Assign Sample weights to all records in the dataset.

$$\text{sample weight} = \frac{1}{n}\text{ , n = total number of records in the dataset}$$

Step-2:

i. Create stumps for each feature in the dataset.

ii. Calculate the Gini Index for the current stump.

iii. Select stump with minimum Gini Index.

Step-3: For selected stump, calculate the total error,

$$\text{Total Error (TE)} = \sum Sum\ of\ Weights\ with\ Incorrectlty\ Classified\ Record$$

Step-4: Calculate the Performance of the Stump

$$\text{Performance of Stump} = \frac{1}{2}\log\left(\frac{1-TE}{TE}\right)$$

Step-5: Update the old weight by new weight:

$$\text{Weight}_{new} = \text{Weight}_{old} * e^{Performance\ of\ Stump}$$

$$\text{Weight}_{new} = \text{Weight}_{old} * e^{-Performance\ of\ Stump}$$

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

City	Gender	Income	Illness
Dallas	Male	40367	No
Dallas	Female	41524	Yes
Dallas	Male	46373	Yes
New York City	Male	98096	No
New York City	Female	102089	No
New York City	Female	100662	No
New York City	Male	117263	Yes
Dallas	Male	56645	No

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

Step-1: Assign Sample weights to all records in the dataset.

City	Gender	Income	Illness	Sample Weight
Dallas	Male	40367	No	1/8
Dallas	Female	41524	Yes	1/8
Dallas	Male	46373	Yes	1/8
New York City	Male	98096	No	1/8
New York City	Female	102089	No	1/8
New York City	Female	100662	No	1/8
New York City	Male	117263	Yes	1/8
Dallas	Male	56645	No	1/8

Ensemble Learning

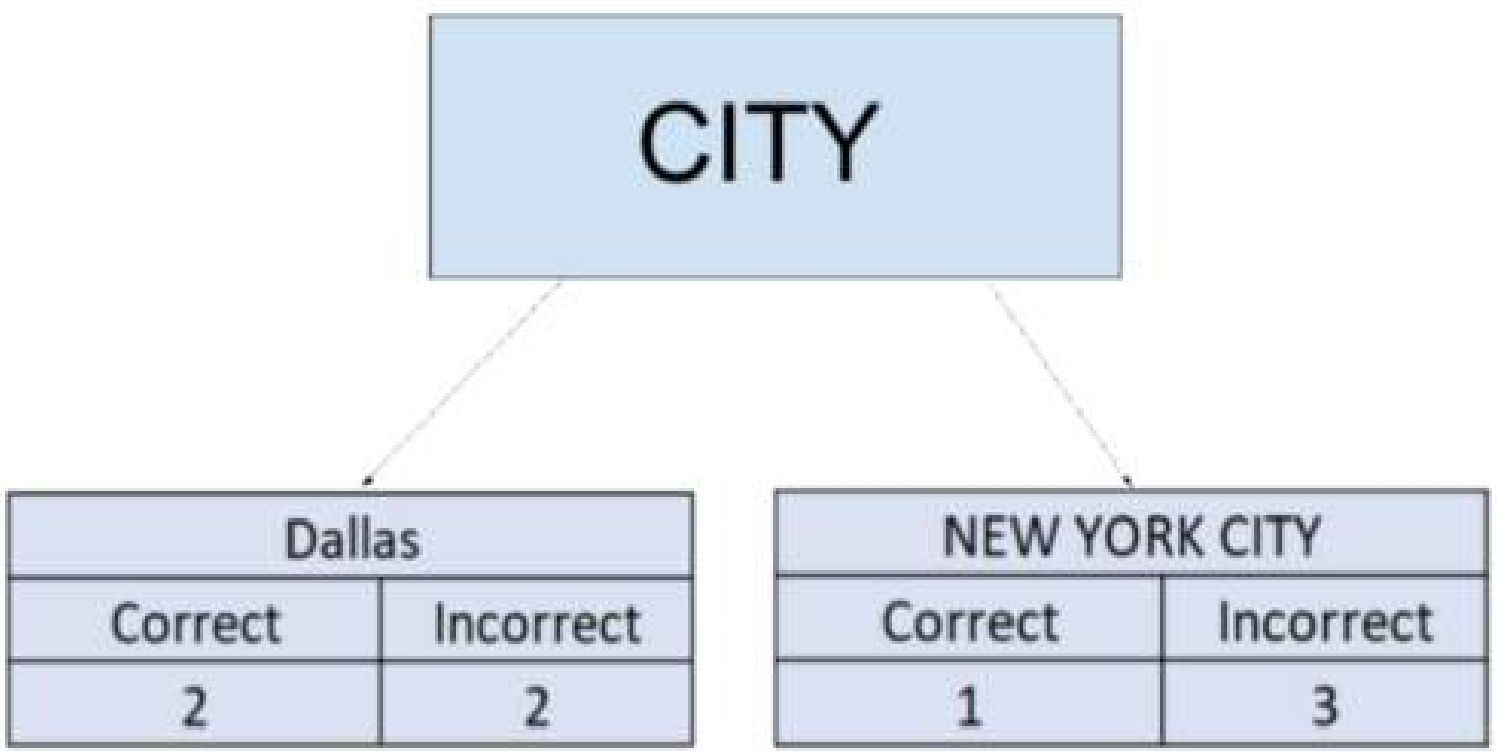
Adaboosting (Adaptive Boosting) Technique

Step-2:

i. **Create stumps for each feature in the dataset.**

ii. **Calculate the Gini Index for the current stump.**

iii. **Select stump with minimum Gini Index.**



City	Illness
Dallas	No
Dallas	Yes
Dallas	Yes
New York City	No
New York City	No
New York City	No
New York City	Yes
Dallas	No

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

Step-2:

i. **Create stumps for each feature in the dataset.**

ii. **Calculate the Gini Index for the current stump.**

iii. **Select stump with minimum Gini Index.**

$$\text{Gini} = 1 - \sum_{i=1}^{n} (p_i)^2$$

Gini Coefficient of Dallas City = 1 -(2/(2+2))² -(2/(2+2))² = 0.5

Gini Coefficient of New York City = 1 -(1/(1+3))² -(3/(1+3))² = 0.375

Gini Coefficient for City Column = 0.5 * ((2+2) / (2+2+1+3))+ 0.375 *((1+3) / (2+2+1+3)) = 0.437

Ensemble Learning

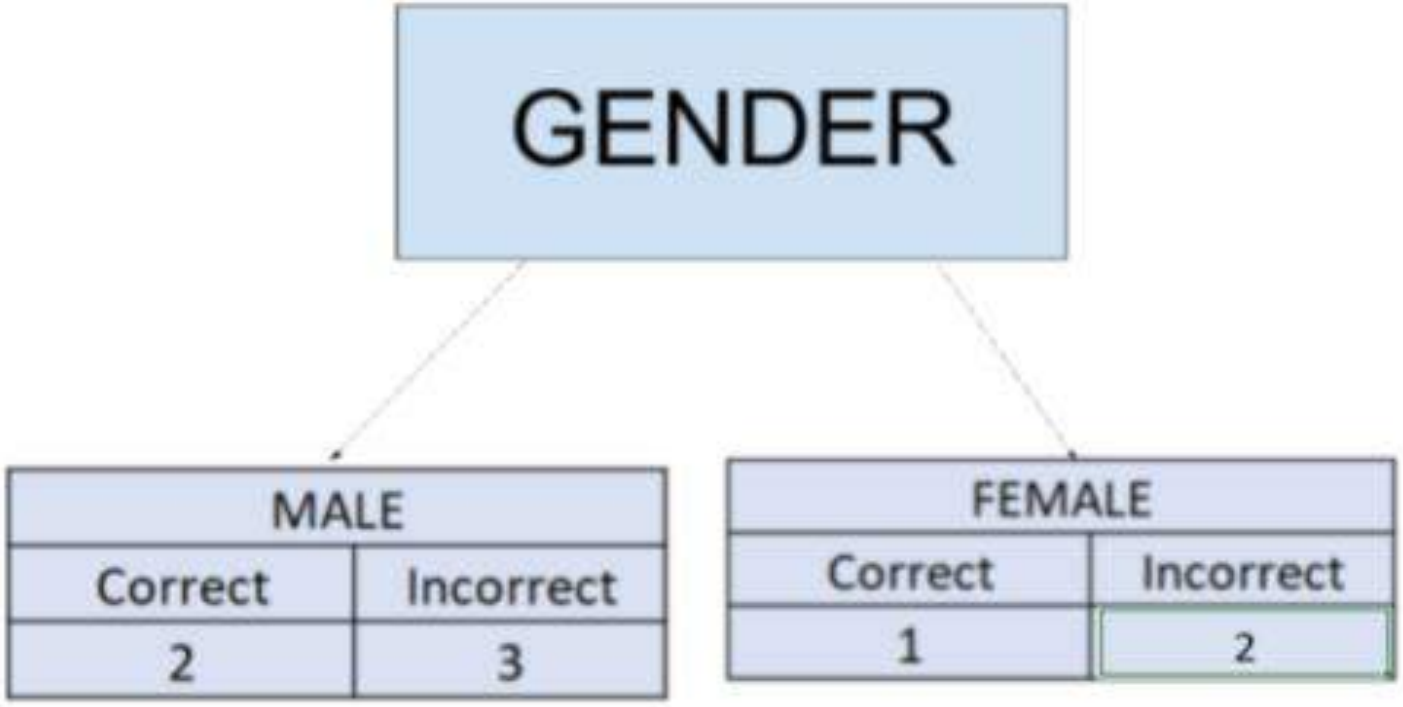
Adaboosting (Adaptive Boosting) Technique

Step-2:

i. **Create stumps for each feature in the dataset.**

ii. **Calculate the Gini Index for the current stump.**

iii. **Select stump with minimum Gini Index.**



Gender	Illness
Male	No
Female	Yes
Male	Yes
Male	No
Female	No
Female	No
Male	Yes
Male	No

Gender column = 0.48 * ((2+3) / (2+3+1+2))+ 0.375 *((1+2) / (2+3+1+2)) = 0.44

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

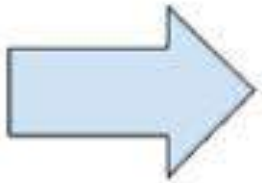
Step-2:

i. **Create stumps for each feature in the dataset.**

ii. **Calculate the Gini Index for the current stump.**

iii. **Select stump with minimum Gini Index.**

Income	Illness
40367	No
41524	Yes
46373	Yes
98096	No
102089	No
100662	No
117263	Yes
56645	No



Income	Illness
40367	No
41524	Yes
46373	Yes
56645	No
98096	No
100662	No
102089	No
117263	Yes

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

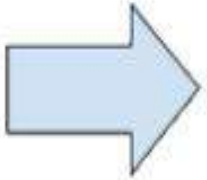
Step-2:

i. **Create stumps for each feature in the dataset.**

ii. **Calculate the Gini Index for the current stump.**

iii. **Select stump with minimum Gini Index.**

Income	Illness
40367	No
41524	Yes
46373	Yes
98096	No
102089	No
100662	No
117263	Yes
56645	No



Income	Illness
40367	No
41524	Yes
46373	Yes
56645	No
98096	No
100662	No
102089	No
117263	Yes

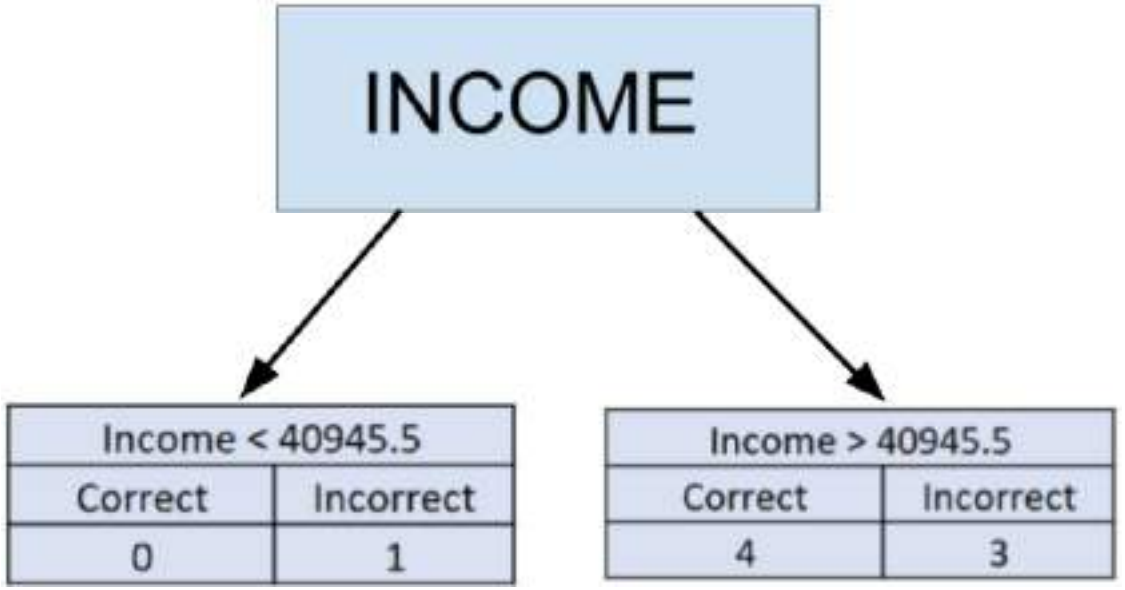
Income	Average
40367	40945.5
41524	
41524	43948.5
46373	
46373	51509
56645	
56645	77370.5
98096	
98096	99379
100662	
100662	101375.5
102089	
102089	109676
117263	

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

Income	Illness
40367	No
41524	Yes
46373	Yes
98096	No
102089	No
100662	No
117263	Yes
56645	No

Income	Illness
40367	No
41524	Yes
46373	Yes
56645	No
98096	No
100662	No
102089	No
117263	Yes

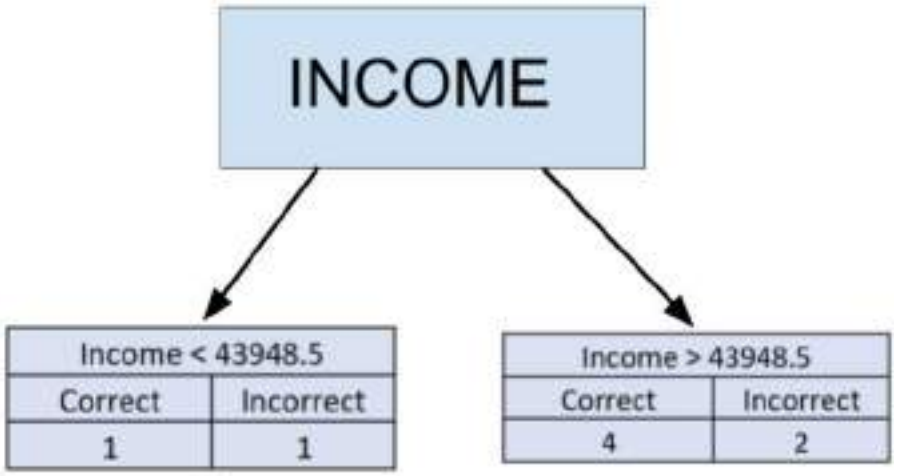


Gini Index for 40945.5 value=

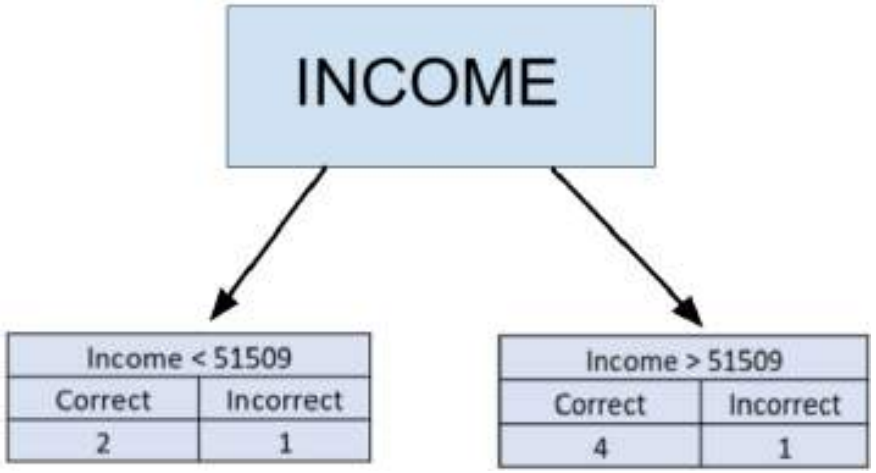
0.48 * ((4+3) / (0+1+4+3))= 0.42

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique



Gini Index for **43948.5** value= 0.5 * ((1+1) / (1+1+2+4))+ 0.44 *((2+4) / (1+1+2+4)) = **0.455**



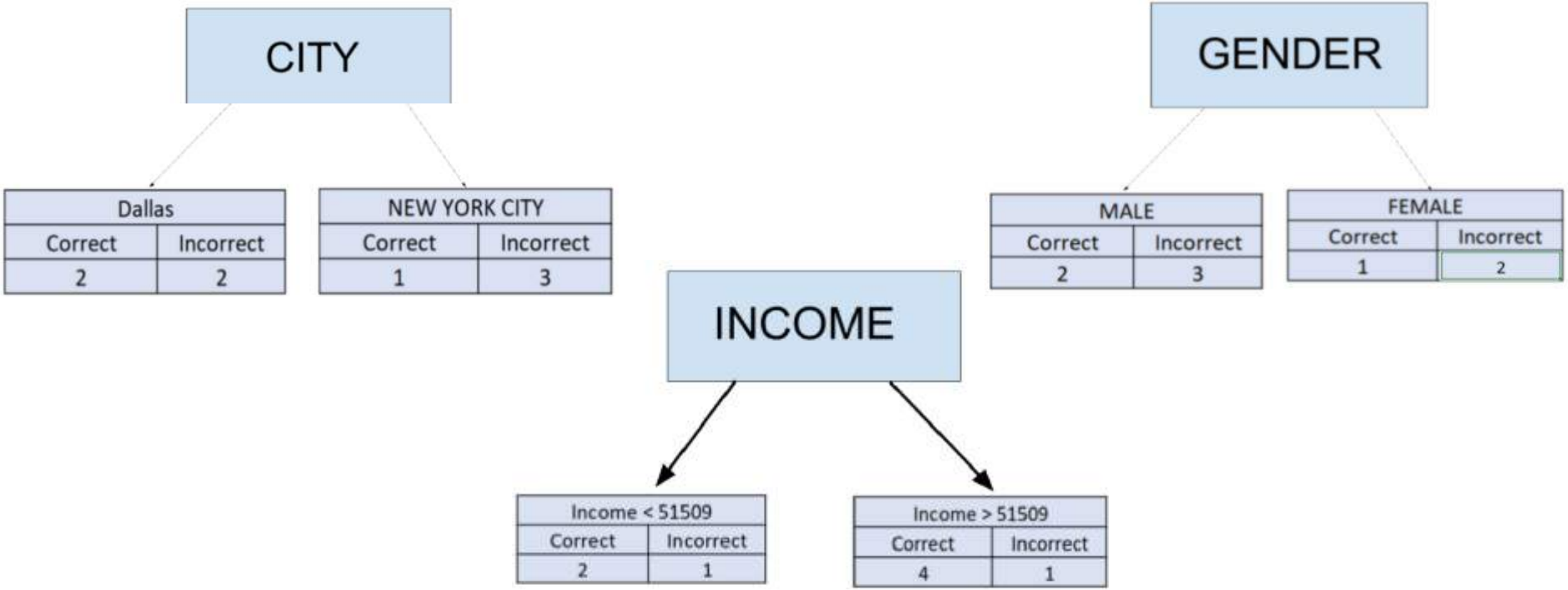
Gini Index for **51509** value= 0.11 * ((2+1) / (2+1+1+4))+ 0.24125 *((1+4) / (2+1+1+4)) = **0.19**

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

Step-3: For selected stump, calculate the total error,

$$\text{Total Error (TE)} = \sum Sum\ of\ Weights\ with\ Incorrectlty\ Classified\ Record$$



Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

Step-4: Calculate the Performance of the Stump

$$\text{Amount of Say} = \frac{1}{2}\log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$

Amount of Say for City Column is : (1/2)*log((1- (5/8))/(5/8)) = -0.11

Amount of Say for Gender Column is : (1/2)*log((1- (5/8))/(5/8)) = -0.11

Amount of Say for Income Column is : (1/2)*log((1- (2/8))/(2/8))= 0.23

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

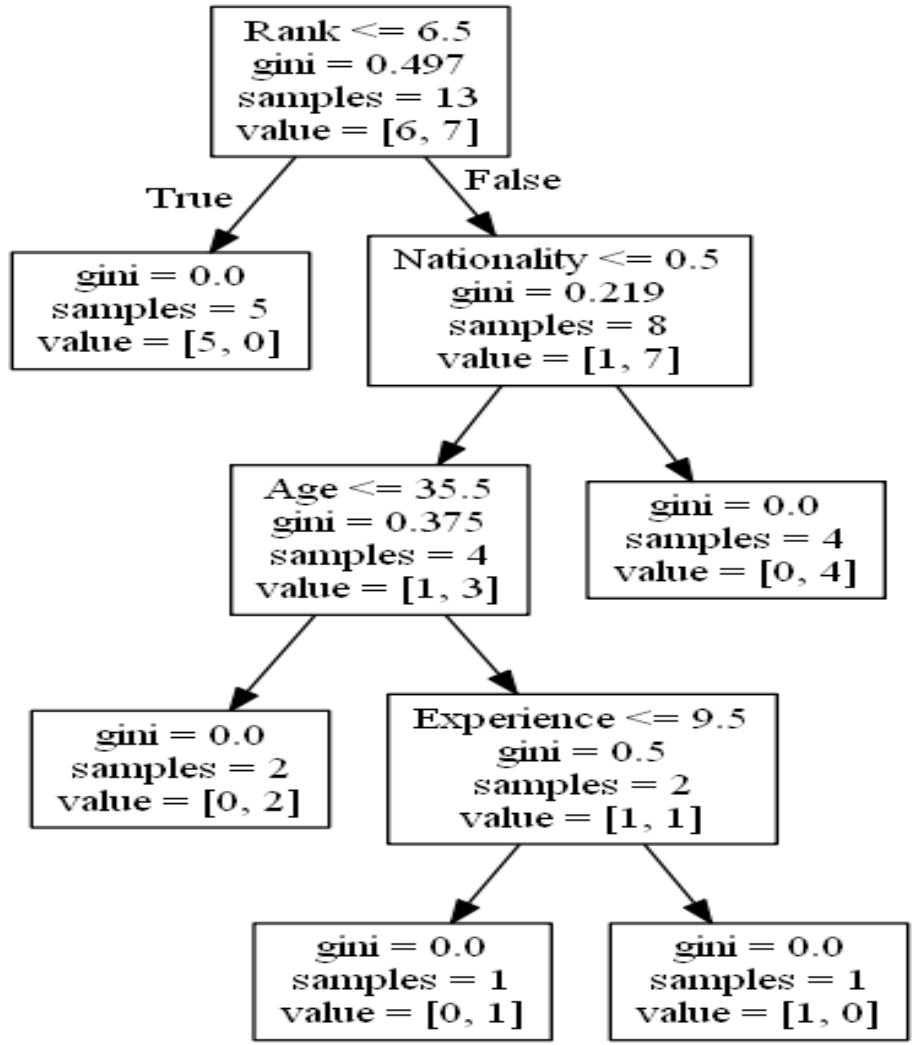
Step-5: Update the old weight by new weight:

$$\text{Weight}_{new} = \text{Weight}_{old} * e^{Performance\ of\ Stump}$$

Update weights of the data points which are *incorrectly* classified by the column stump having lowest Gini Index.

$$\text{Weight}_{new} = \text{Weight}_{old} * e^{-Performance\ of\ Stump}$$

Update weights of the data points in that column that are *correctly* classified.


Rank <= 6.5
gini = 0.497
samples = 13
value = [6, 7]
True
False
gini = 0.0
samples = 5
value = [5, 0]
Nationality <= 0.5
gini = 0.219
samples = 8
value = [1, 7]
Age <= 35.5
gini = 0.375
samples = 4
value = [1, 3]
gini = 0.0
samples = 4
value = [0, 4]
gini = 0.0
samples = 2
value = [0, 2]
Experience <= 9.5
gini = 0.5
samples = 2
value = [1, 1]
gini = 0.0
samples = 1
value = [0, 1]
gini = 0.0
samples = 1
value = [1, 0]

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

City	Gender	Income	Illness	Sample Weight	New Weight
Dallas	Male	40367	No	1/8	0.629
Dallas	Female	41524	Yes	1/8	0.397
Dallas	Male	46373	Yes	1/8	0.397
New York City	Male	98096	No	1/8	0.397
New York City	Female	102089	No	1/8	0.397
New York City	Female	100662	No	1/8	0.397
New York City	Male	117263	Yes	1/8	0.397
Dallas	Male	56645	No	1/8	0.629

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

City	Gender	Income	Illness	New Weight	Normalized Weight
Dallas	Male	40367	No	0.629	0.17
Dallas	Female	41524	Yes	0.397	0.1
Dallas	Male	46373	Yes	0.397	0.1
New York City	Male	98096	No	0.397	0.1
New York City	Female	102089	No	0.397	0.1
New York City	Female	100662	No	0.397	0.1
New York City	Male	117263	Yes	0.397	0.1
Dallas	Male	56645	No	0.629	0.17

(0.629+0.397+0.397+ 0.397+0.397+0.397+0.397+0.629)= **3.64**

0.629/3.64 = **0.17** 0.397/3.64 = **0.10**

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

City	Gender	Income	Illness	New Weight	Normalized Weight
Dallas	Male	40367	No	0.629	0.17
Dallas	Female	41524	Yes	0.397	0.1
Dallas	Male	46373	Yes	0.397	0.1
New York City	Male	98096	No	0.397	0.1
New York City	Female	102089	No	0.397	0.1
New York City	Female	100662	No	0.397	0.1
New York City	Male	117263	Yes	0.397	0.1
Dallas	Male	56645	No	0.629	0.17

City	Gender	Income	Illness	New Weight

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

City	Gender	Income	Illness	Weights
Dallas	Male	40367	No	0.17
Dallas	Female	41524	Yes	0.1
Dallas	Male	40367	No	0.17
Dallas	Male	40367	No	0.17
New York City	Female	102089	No	0.1
New York City	Female	100662	No	0.1
Dallas	Male	40367	No	0.17
Dallas	Male	40367	No	0.17

Ensemble Learning

Algorithm

Step-1: Assign Sample weights to all records in the dataset.

$$\text{sample weight} = \frac{1}{n}\text{ , n = total number of records in the dataset}$$

Step-2:

i. Create stumps for each feature in the dataset.

ii. Calculate the Gini Index for the current stump.

iii. Select stump with minimum Gini Index.

Step-3: For selected stump, calculate the total error,

$$\text{Total Error (TE)} = \sum Sum\ of\ Weights\ with\ Incorrectlty\ Classified\ Record$$

Step-4: Calculate the Performance of the Stump

$$\text{Performance of Stump} = \frac{1}{2}\log\left(\frac{1-TE}{TE}\right)$$

Step-5: Update the old weight by new weight:

$$\text{Weight}_{new} = \text{Weight}_{old} * e^{Performance\ of\ Stump}$$

$$\text{Weight}_{new} = \text{Weight}_{old} * e^{-Performance\ of\ Stump}$$

Ensemble Learning

Adaboosting (Adaptive Boosting) Technique

Chest Pain (X1)	Blocked Arteries (X2)	Patient Weight (X3)	Heart Disease
Yes	Yes	205	Yes
No	Yes	180	Yes
Yes	No	210	Yes
Yes	Yes	167	Yes
No	Yes	156	No
No	Yes	125	No
Yes	No	168	No
Yes	Yes	172	No

Assignment IV

Q1. Explain SVM in Detail?

Q2. Explain the Term: Bagging and Boosting?

Q3. Explain various steps in Adaboosting Algorithm?

Q4. Explain Random Forest Algorithm with advantages and disadvantages?

Check your assignment on or before 24/10/2023.

References

Test Books

1. Bishop, Christopher M., and Nasser M. Nasrabadi, “Pattern recognition and machine learning”,Vol. 4. No. 4. New York: springer, 2006.
2. Ethem Alpaydin, “ Introduction to Machine Learning”, PHI 2nd Edition-2013

Reference Books

1. Tom Mitchell, “Machine learning”, McGraw-Hill series in Computer Science.
2. Shalev-Shwartz, Shai, and Shai Ben-David, “Understanding machine learning: From theory to algorithms”, Cambridge university press, 2014.
3. Jiawei Han, Micheline Kamber, and Jian Pie, “Data Mining: Concepts and Techniques”, Elsevier Publishers 3 Edition.
4. Hastie, Trevor, et al., “The elements of statistical learning: data mining, inference, and prediction”, Vol. 2. New York: springer, 2009.
5. McKinney, “Python for Data Analysis “,O' Reilly media.
6. Trent hauk, “Scikit-learn”, Cookbook , Packt Publishing, ISBN: 9781787286382
7. Goodfellow I.,Bengio Y. and Courville, “ A Deep Learning”, MIT Press, 2016