DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

II YEAR / IV SEM

CS3491 – ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

UNIT III SUPERVISED LEARNING

SYLLABUS:

Introduction to machine learning – Linear Regression Models: Least squares, single & multiple variables, Bayesian linear regression, gradient descent, Linear Classification Models: Discriminant function – Probabilistic discriminative model - Logistic regression, Probabilistic generative model – Naive Bayes, Maximum margin classifier – Support vector machine, Decision Tree, Random forests

PART A

1. Define Machine Learning.

- Arthur Samuel, an early American leader in the field of computer gaming and artificial intelligence, coined the term "Machine Learning " in 1959 while at IBM.
- He defined machine learning as "the field of study that gives computers the ability to learn without being explicitly programmed ".
- Machine learning is programming computers to optimize a performance criterion using example data or past experience. The model may be predictive to make predictions in the future, or descriptive to gain knowledge from data.

2 Mention the various classification of Machine Learning

- Machine learning implementations are classified into four major categories, depending on the nature of the learning "signal" or "response" available to a learning system which are as follows:
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
  - Semi-supervised learning

3. Define Supervised learning

- Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.
- The given data is labeled.

- Both classification and regression problems are supervised learning problems.

5. Define Unsupervised learning

- Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses.
- In unsupervised learning algorithms, classification or categorization is not included in the observations.
- In unsupervised learning the agent learns patterns in the input without any explicit feedback.
- The most common unsupervised learning task is clustering: detecting potentially useful clusters of input examples.

6. What is Reinforcement learning?

- In reinforcement learning the agent learns from a series of reinforcements: rewards and punishments.
- Reinforcement learning is the problem of getting an agent to act in the world so as to maximize its rewards.
- A learner is not told what actions to take as in most forms of machine learning but instead must discover which actions yield the most reward by trying them.

7. What is Semi-supervised learning?

- Semi-Supervised learning is a type of Machine Learning algorithm that represents the intermediate ground between Supervised and Unsupervised learning algorithms.
- It uses the combination of labeled and unlabeled datasets during the training period, where an incomplete training signal is given: a training set with some of the target outputs missing.

8. How to Categorize algorithm based on required Output?

- Classification
- Regression
- Clustering

9. Define Classification.

- The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data.
- In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups.

- Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc. Classes can be called as targets/labels or categories.

10. Define Regression.

- Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on the one or more predictor variables.
- It is mainly used for prediction, forecasting, time series modeling, and determining the causal-effect relationship between variables.

11. Define Clustering.

- Clustering or cluster analysis is a machine learning technique, which groups the unlabeled dataset.
- It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points.
- The objects with the possible similarities remain in a group that has less or no similarities with another group."

12. What is Linear Regression?

- ***In statistics, linear regression*** is a linear approach to modeling the relationship between a dependent variable and one or more independent variables.
- Let **X** be the independent variable and **Y** be the dependent variable.
- A linear relationship between these two variables as follows:

$$Y = mX + c$$

Where,

m: Slope

c: y-intercept

13. What is Least Squares Regression Line?

- Least squares are a commonly used method in regression analysis for estimating the unknown parameters by creating a model which will minimize the sum of squared errors between the observed data and the predicted data.

14. Narrate Least Squares Regression Equation

- The equation that minimizes the total of all squared prediction errors for known Y scores in the original correlation analysis.

LEAST SQUARES REGRESSION EQUATION

$$Y' = bX + a$$

where

Y′ represents the predicted value;

X represents the known value;

b and a represent numbers calculated from the original correlation analysis

15. List and define the types of Linear Regression.

It is of two types: **Simple and Multiple**.

- **Simple Linear Regression** is where only one independent variable is present and the model has to find the linear relationship of it with the dependent variable
- **Equation of Simple Linear Regression**, where b_o is the intercept, b_1 is coefficient or slope, x is the independent variable and y is the dependent variable.

$$y = b_o + b_1 x$$

- In **Multiple Linear Regression** there are more than one independent variables for the model to find the relationship.

Equation of Multiple Linear Regression, where bo is the intercept, $b_1, b_2, b_3, b_4 \ldots, b_n$ are coefficients or slopes of the independent variables $x_1, x_2, x_3, x_4 \ldots, x_n$ and y is the dependent variable.

$$y = b_o + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots . + b_n x_n$$

16. Define Linear Regression Model.

- A Linear Regression model's main aim is to find the best fit linear line and the optimal values of intercept and coefficients such that the error is minimized.

17. What is error or residual?

- Error is the difference between the actual value and Predicted value and the goal is to reduce this difference.
- The vertical distance between the data point and the regression line is known as error or residual.
- Each data point has one residual and the sum of all the differences is known as **the Sum of Residuals/Errors.**

PART B

1. Define Machine Learning. Give an introduction to Machine Learning.

INTRODUCTION TO MACHINE LEARNING

1.1 Machine Learning

- 1.1.1 **Definition of Machine Learning**
- 1.1.2 **Definition of learning**
- 1.1.3 **Examples**
  - 1.1.3.1 **Handwriting recognition learning problem**
  - 1.1.3.2 **A robot driving learning problem**

1.2 Classification of Machine Learning

- 1.2.1 **Supervised learning**
- 1.2.2 **Unsupervised learning**
- 1.2.3 **Reinforcement learning**
- 1.2.4 **Semi-supervised learning**

1.3 Categorizing based on required Output

- 1.3.1 **Classification**
- 1.3.2 **Regression**
- 1.3.3 **Clustering**

1.1 Machine Learning:

1.1.1 Definition of Machine Learning:

- Arthur Samuel, an early American leader in the field of computer gaming and artificial intelligence, coined the term "Machine Learning " in 1959 while at IBM.
- He defined machine learning as "the field of study that gives computers the ability to learn without being explicitly programmed ".
- Machine learning is programming computers to optimize a performance criterion using example data or past experience. The model may be predictive to make predictions in the future, or descriptive to gain knowledge from data.

1.1.2 Definition of learning:

- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks T, as measured by P , improves with experience E.

1.1.3 Examples

1.1.3.1 Handwriting recognition learning problem

- Task T : Recognizing and classifying handwritten words within images
- Performance P : Percent of words correctly classified
- Training experience E : A dataset of handwritten words with given classifications

1.1.3.2 A robot driving learning problem

- Task T : Driving on highways using vision sensors
- Performance P : Average distance traveled before an error
- Training experience E : A sequence of images and steering commands recorded while observing a human driver

1.2 Classification of Machine Learning

- Machine learning implementations are classified into four major categories, depending on the nature of the learning "signal" or "response" available to a learning system which are as follows:

1.2.1 Supervised learning:

- Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.
- The given data is labeled.
- Both classification and regression problems are supervised learning problems.
- **For example**, the inputs could be camera images, each one accompanied by an output saying "bus" or "pedestrian," etc.
- An output like this is calleda label.
- The agent learns a function that, when given a new image, predicts theappropriate label.

1.2.2 Unsupervised learning:

- Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses.
- In unsupervised learning algorithms, classification or categorization is not included in the observations.
- In unsupervised learning the agent learns patterns in the input without any explicit feedback.
- The most common unsupervised learning task is clustering: detecting potentially useful clusters of input examples.

- **For example**, when shown millions of images taken from the Internet, a computer vision system can identify a large cluster of similar images which an English speaker would call "cats."

1.2.3 Reinforcement learning:

- In reinforcement learning the agent learns from a series of reinforcements: rewards and punishments.
- Reinforcement learning is the problem of getting an agent to act in the world so as to maximize its rewards.
- A learner is not told what actions to take as in most forms of machine learning but instead must discover which actions yield the most reward by trying them.
- **For example** — Consider teaching a dog a new trick: we cannot tell him what to do, what not to do, but we can reward/punish it if it does the right/wrong thing.

1.2.4 Semi-supervised learning:

- Semi-Supervised learning is a type of Machine Learning algorithm that represents the intermediate ground between Supervised and Unsupervised learning algorithms.
- It uses the combination of labeled and unlabeled datasets during the training period, where an incomplete training signal is given: a training set with some of the target outputs missing.

1.3 Categorizing based on required Output

1.3.1 Classification:

- The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data.
- In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups.
- Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc. Classes can be called as targets/labels or categories.

1.3.1 Regression:

- Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on the one or more predictor variables.

- It is mainly used for prediction, forecasting, time series modeling, and determining the causal-effect relationship between variables.

1.3.2 Clustering:

- Clustering or cluster analysis is a machine learning technique, which groups the unlabeled dataset.
- It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points.
- The objects with the possible similarities remain in a group that has less or no similarities with another group."

2. Explain in detail about Linear Regression Models. Or Explain Linear Regression Models: Least squares, single & multiple variables, Bayesian linear regression, gradient descent.

LINEAR REGRESSION MODELS

2.1 Linear Regression
2.2 Least Squares Regression Line
2.2.1 Least Squares Regression Equation
2.2.2 Least Squares Regression in Python
2.3 Types of Linear Regression
2.4 Linear Regression Model
2.5 Bayesian Regression
2.5.1 Implementation Of Bayesian Regression Using Python
2.6 Gradient descent
2.6.1 Cost Function
2.6.2 Gradient Descent Algorithm.

2.1 Linear Regression

- In statistics, linear regression is a linear approach to modeling the relationship between a dependent variable and one or more independent variables.
- Let **X** be the independent variable and **Y** be the dependent variable.
- A linear relationship between these two variables as follows:

$$Y = mX + c$$

Where,

m: Slope

c: y-intercept

- Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression.
- Linear regression finds how the value of the dependent variable is changing according to the value of the independent variable.
- The linear regression model provides a sloped straight line representing the relationship between the variables.
- Consider the below Figure 3.1, which represents the relationship between independent and dependent variables

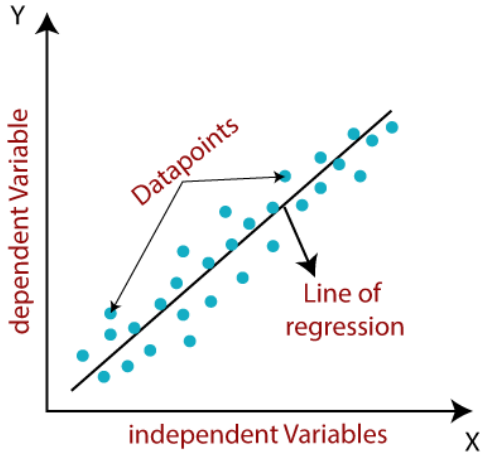


Figure 3.1 - Relationship between independent and dependent variables

2.2 Least Squares Regression Line

- Least squares are a commonly used method in regression analysis for estimating the unknown parameters by creating a model which will minimize the sum of squared errors between the observed data and the predicted data.

2.2.1 Least Squares Regression Equation

- The equation that minimizes the total of all squared prediction errors for known Y scores in the original correlation analysis.

LEAST SQUARES REGRESSION EQUATION

$$Y' = bX + a$$

where

Y′ represents the predicted value;

X represents the known value;

b and a represent numbers calculated from the original correlation analysis

2.2.2 Least Squares Regression in Python

Scenario: *A rocket motor is manufactured by combining an igniter propellant and a sustainer propellant inside a strong metal housing. It was noticed that the shear strength of the bond between two propellers is strongly dependent on the age of the sustainer propellant.*

Problem Statement: *Implement a simple linear regression algorithm using Python to build a machine learning model that studies the relationship between the shear strength of the bond between two propellers and the age of the sustainer propellant.*

Step 1: *Import the required Python libraries.*

```
# Importing Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Step 2: *Next step is to read and load the dataset.*

```
# Loading dataset
data = pd.read_csv('PropallantAge.csv')
data.head()
data.info()
```

Step 3: *Create a scatter plot just to check the relationship between these two variables.*

```
# Plotting the data
plt.scatter(data['Age of Propellant'],data['Shear Strength'])
```

Step 4: *Next step is to assign X and Y as independent and dependent variables respectively.*

```
# Computing X and Y
X = data['Age of Propellant'].values
Y = data['Shear Strength'].values
```

Step 5: *Compute the mean of variables X and Y to determine the values of slope (m) and y-intercept.*

Also, let n be the total number of data points.

```
# Mean of variables X and Y
mean_x = np.mean(X)
mean_y = np.mean(Y)
# Total number of data values
n = len(X)
```

Step 6: *Calculate the slope and the y-intercept using the formulas*

```
# Calculating 'm' and 'c'
        num = 0
        denom = 0
        for i in range(n):
        num += (X[i] - mean_x) * (Y[i] - mean_y)
        denom += (X[i] - mean_x) ** 2
        m = num / denom
        c = mean_y - (m * mean_x)

# Printing coefficients
        print("Coefficients")
        print(m, c)
```

The above step has given the values of m and c. Substituting them ,

Shear Strength =

2627.822359001296 + (-37.15359094490524)

*** Age of Propellant**

Step 7: *The above equation represents the linear regression model. Let's plot this graphically. Refer fig 3.2*

```
# Plotting Values and Regression Line
        maxx_x = np.max(X) + 10
      minn_x = np.min(X) - 10

# line values for x and y
        x = np.linspace(minn_x, maxx_x, 1000)
        y = c + m * x
# Plotting Regression Line
plt.plot(x, y, color='#58b970', label='Regression Line')
# Plotting Scatter Points
        plt.scatter(X, Y, c='#ef5423', label='Scatter Plot')
        plt.xlabel('Age of Propellant (in years)')
        plt.ylabel('Shear Strength')
        plt.legend()
        plt.show()
```

Output:

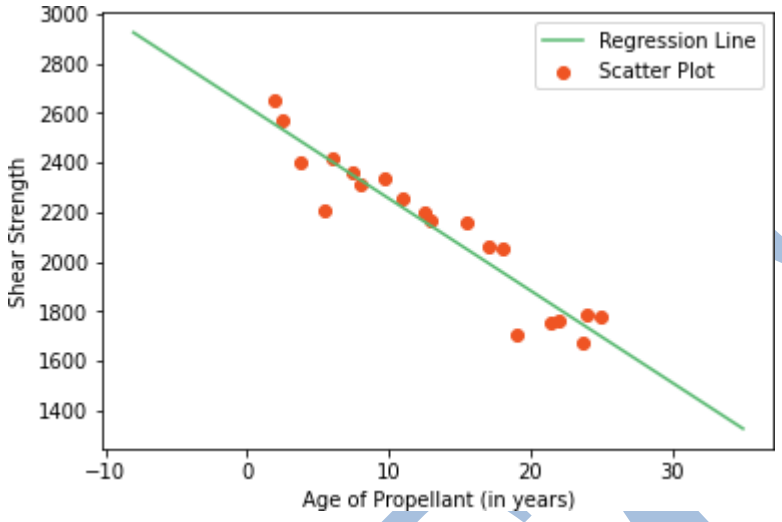


Figure 3.2 - Example for Regression Line

2.3 Types of Linear Regression

It is of two types: **Simple and Multiple**.

- **Simple Linear Regression** is where only one independent variable is present and the model has to find the linear relationship of it with the dependent variable

  Equation of Simple Linear Regression, where b_o is the intercept, b_1 is coefficient or slope, x is the independent variable and y is the dependent variable.

$$y = b_o + b_1 x$$

- In **Multiple Linear Regression** there are more than one independent variables for the model to find the relationship.

  Equation of Multiple Linear Regression, where bo is the intercept, $b_1, b_2, b_3, b_4 \ldots, b_n$ are coefficients or slopes of the independent variables $x_1, x_2, x_3, x_4 \ldots, x_n$ and y is the dependent variable.

$$y = b_o + b_1 x_1 + b_2 x_2 + b_3 x_3 \ldots . + b_n x_n$$

2.4 Linear Regression Model

- A Linear Regression model's main aim is to find the best fit linear line and the optimal values of intercept and coefficients such that the error is minimized.
- Error is the difference between the actual value and Predicted value and the goal is to reduce this difference.

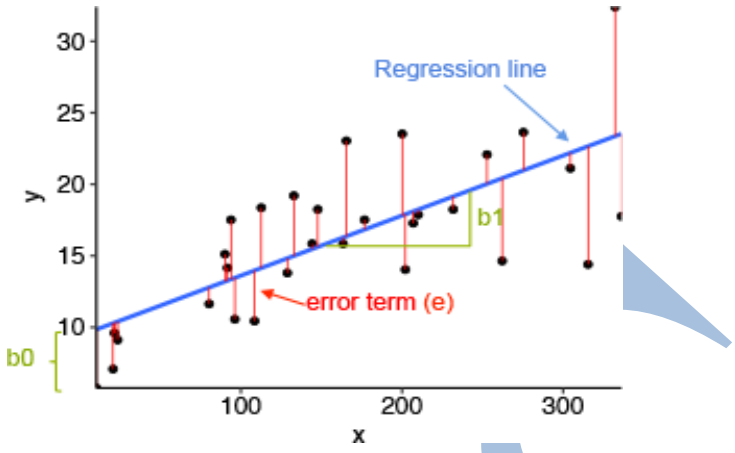


Figure 3.3 - Example for Linear Regression Model

- In the above figure 3.3,
  - x is our dependent variable which is plotted on the x-axis and y is the dependent variable which is plotted on the y-axis.
  - Black dots are the data points i.e the actual values.
  - b_o is the intercept which is 10 and b_1 is the slope of the x variable.
  - The blue line is the best fit line predicted by the model i.e the predicted values lie on the blue line.
- **The vertical distance between the data point and the regression line is known as error or residual.**
- Each data point has one residual and the sum of all the differences is known as **the Sum of Residuals/Errors.**

Mathematical Approach:

Residual/Error = Actual values – Predicted Values

Sum of Residuals/Errors = Sum(Actual- Predicted Values)

Square of Sum of Residuals/Errors = (Sum(Actual- Predicted Values))^2

$$\sum e_i^2 = \sum (Y_i - \hat{Y}_i)^2$$

2.5 Bayesian Regression

- Bayesian Regression is used when the data is insufficient in the dataset or the data is poorly distributed.
- The output of a Bayesian Regression model is obtained from a probability distribution.

- The aim of Bayesian Linear Regression is to find the '**posterior**' distribution for the model parameters.
- The expression for Posterior is :

$$Posterior = \frac{(Likelihood*Prior)}{Normalization}$$

where

- **Posterior:** It is the probability of an event to occur; say, H, given that another event; say, E has already occurred. i.e., P(H | E).
- Prior: It is the probability of an event H has occurred prior to another event. i.e., P(H)
- Likelihood: It is a likelihood function in which some parameter variable is marginalized.

- The Bayesian Ridge Regression formula is as follows:

p(y|λ) = N(w|0, λ^-1Ip)

where

- 'y' is the expected value,
- lambda is the distribution's shape parameter before the lambda parameter
- the vector "w" is made up of the elements w0, w1,....

2.5.1 Implementation Of Bayesian Regression Using Python

- Boston Housing dataset, which includes details on the average price of homes in various Boston neighborhoods.
- The r2 score will be used for evaluation.
- The crucial components of a Bayesian Ridge Regression model:

Program

```
fromsklearn.datasets import load_boston
fromsklearn.model_selection import train_test_split
fromsklearn.metrics import r2_score
fromsklearn.linear_model import BayesianRidge

# Loading the dataset
        dataset = load_boston()
        X, y = dataset.data, dataset.target

# Splitting the dataset into testing and training sets
        X_train, X_test, y_train, y_test = train_test_split
                (X, y, test_size = 0.15, random_state = 42)
```

```
# Creating to train the model
model = BayesianRidge()
model.fit(X_train, y_train)

# Model predicting the test data
prediction = model.predict(X_test)

# Evaluation of r2 score of the model against the test dataset
print(f"Test Set r2 score : {r2_score(y_test, prediction)}")
```

Output

Test Set r2 score : 0.7943355984883815

Advantages of Bayesian Regression:

- Very effective when the size of the dataset is small.
- Particularly well-suited for on-line based learning (data is received in real-time), as compared to batch based learning, where we have the entire dataset on our hands before we start training the model. This is because Bayesian Regression doesn't need to store data.
- The Bayesian approach is a tried and tested approach and is very robust, mathematically. So, one can use this without having any extra prior knowledge about the dataset.

Disadvantages of Bayesian Regression:

- The inference of the model can be time-consuming.
- If there is a large amount of data available for our dataset, the Bayesian approach is not worth it.

2.6 Gradient descent

2.6.1 Cost Function

- The cost is the error in our predicted value.
- It is calculated using the Mean Squared Error function as shown in figure 3.4.

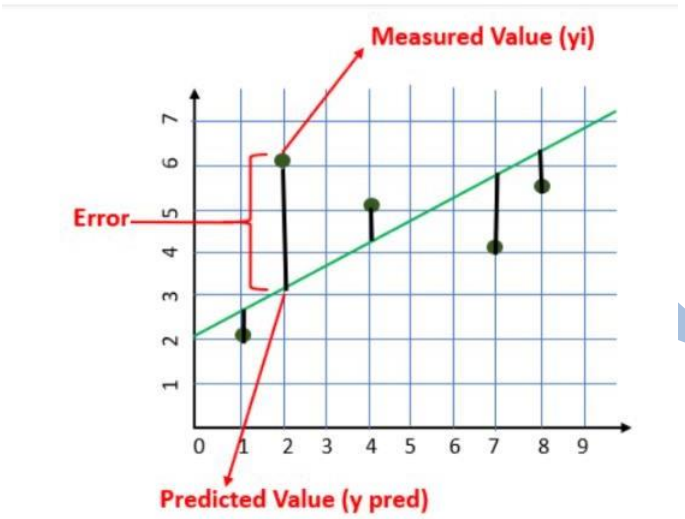


Figure 3.4 - Example for Cost function

$$Cost\ Function(MSE) = \frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2$$

Replace $y_{i\,pred}$ with $mx_i + c$

$$Cost\ Function(MSE) = \frac{1}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))^2$$

- The goal is to minimize the cost as much as possible in order to find the best fit line.

2.6.2 Gradient Descent Algorithm.

- Gradient descent is an optimization algorithm that finds the best-fit line for a given training dataset in a smaller number of iterations.
- If m and c are plotted against MSE, it will acquire a bowl shape as shown in figure 3.4a and figure 3.4b.

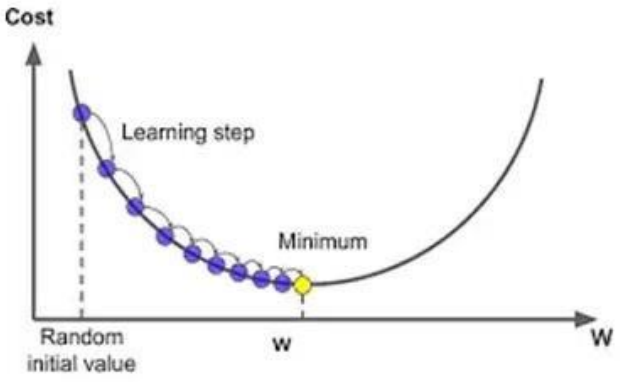


Figure 3.4a - Process of gradient descent algorithm

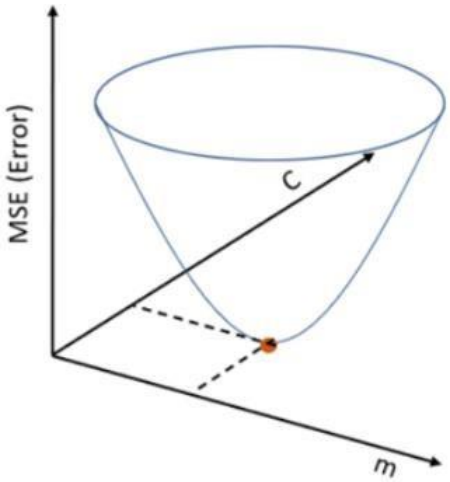


Figure 3.4b - Gradient Descent Shape

Learning Rate

- A learning rate is used for each pair of input and output values. It is a scalar factor and coefficients are updated in direction towards minimizing error.
- The process is repeated until a minimum sum squared error is achieved or no further improvement is possible.

Step by Step Algorithm:

1. Initially, let m = 0, c = 0
   Where L = learning rate — controlling how much the value of "m" changes with each step.
   The smaller the L, greater the accuracy. L = 0.001 for a good accuracy.
2. Calculating the partial derivative of loss function "m" to get the derivative D.

$$D_m = \frac{\partial(Cost\ Function)}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - (mx_i + c))$$

$$= \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - y_{i\,pred})$$

- Similarly, find the partial derivative with respect to c, Dc..

$$D_c = \frac{\partial(Cost\ Function)}{\partial c} = \frac{\partial}{\partial c}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))$$

$$\frac{-2}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})$$

- Update the current values of m and c using the following equation:

$$m = m - LD_m$$

$$c = c - LDc$$

- Repeat this process until our Cost function is very small (ideally 0).

3. Explain in detail about Linear Classification Models – Discriminant function.

LINEAR CLASSIFICATION MODELS – DISCRIMINANT FUNCTION.

1.1 Linear Classification Models

1.2 Types of ML Classification Algorithms

1.3 Discriminant function

3.1 Linear Classification Models

- The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data.
- In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, **Yes or No, 0 or 1, Spam or Not Spam, cat or dog,** etc.

- Classes can be called as targets/labels or categories.
- The output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc.
- Since the Classification algorithm is a supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output.
- In classification algorithm, a discrete output function(y) is mapped to input variable(x).

  y=f(x), where y = categorical output

- The best example of an ML classification algorithm is **Email Spam Detector**.
- The goal of the classification algorithm is
  - Take a D-dimensional input vector x
  - Assign it to one of K discrete classes Ck , k = 1, . . . , K
- In the most common scenario, the classes are taken to be disjoint and each input is assigned to one and only one class
- The input space is divided into decision regions
- The boundaries of the decision regions
  - decision boundaries
  - decision surfaces
- With linear models for classification, the decision surfaces are linear functions, Classes that can be separated well by linear surfaces are linearly separable.
- In the figure 3.5, there are two classes, class A and Class B.
- These classes have features that are similar to each other and dissimilar to other classes.

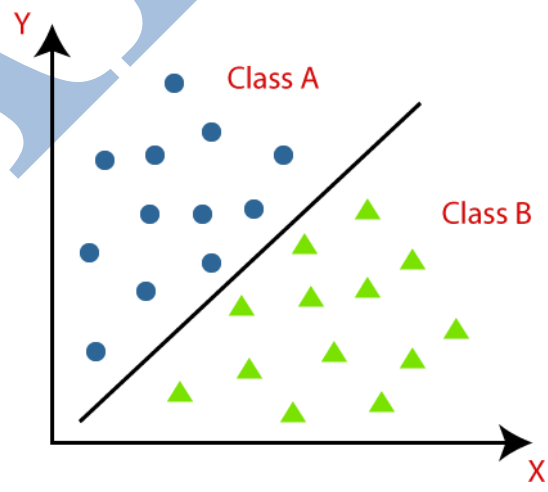


Figure 3.5 - Example of Classification

- The algorithm which implements the classification on a dataset is known as a classifier.
- There are two types of Classifications:
  - **Two-class problems :**
    - **Binary representation or Binary Classifier:**
      - If the classification problem has only two possible outcomes, then it is called as Binary Classifier.
      - There is a single target variable $t \in \{0, 1\}$
        - t = 1 represents class C1
        - t = 0 represents class C2
      - **Examples:** YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.
  - **Multi-class Problems:**
    - 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.
    - **1-of-K coding scheme**
      - There is a K-long target vector t, such that
        If the class is Cj, all elements t_k of t are zero for $k \neq j$ and one for $k = j$ t_k is the probability that the class is C_k, K = 6 and $C_k = 4$, then $t = (0, 0, 0, 1, 0, 0)^T$

- The simplest approach to classification problems is through construction of a discriminant function that directly assigns each vector x to a specific class

3.2 Types of ML Classification Algorithms:

- Logistic Regression
- K-Nearest Neighbors
- Support Vector Machines
- Kernel SVM
- Naïve Bayes
- Decision Tree Classification
- Random Forest Classification

1.4 Discriminant function

- A function of a set of variables that is evaluated for samples of events or objects and used as an aid in discriminating between or classifying them.

- A discriminant function (DF) maps independent (discriminating) variables into a latent variable D.
- DF is usually postulated to be a linear function:

$$D = a_0 + a_1 x_1 + a_2 x_2 \ldots a_N x_N$$

- The goal of discriminant analysis is to find such values of the coefficients $\{a_i, i=0,\ldots,N\}$ that the distance between the mean values of DF is maximal for the two groups.
- Whenever there is a requirement to separate two or more classes having multiple features efficiently, the Linear Discriminant Analysis model is considered the most common technique to solve such classification problems.
- For example, if there are classes with multiple features and need to separate them efficiently. Classify them using a single feature, then it may show overlapping as shown in figure 3.6.

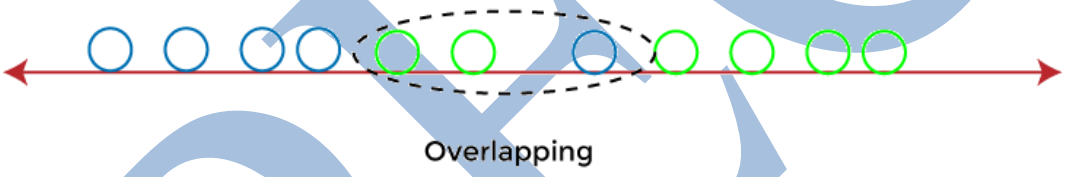


Figure 3.6 - Example for Classification using single feature

- To overcome the overlapping issue in the classification process, must increase the number of features regularly.

4. Explain in detail about Linear Discriminant Functions and its types. Also elaborate about logistic regression in detail.

LINEAR DISCRIMINANT FUNCTIONS AND LOGISTIC REGRESSION

4.1 Linear Discriminant Functions
4.2 The Two-Category Case
4.3 The Multi-category Case
4.4 Generalized Linear Discriminant Functions
4.5 Probabilistic discriminative models
4.6 Logistics Regression
4.6.1 Logistic Function (Sigmoid Function)
4.6.2 Assumptions for Logistic Regression
4.6.3 Logistic Regression Equation
4.6.4 Type of Logistic Regression
4.6.5 Steps in Logistic Regression
4.6.6 Advantages of Logistic Regression Algorithm

4.1 Linear Discriminant Functions

A discriminant function that is a linear combination of the components of x can be written as

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 \quad (3.1)$$

where **w** is the *weight vector* and w_0 the *bias* or *threshold weight*.

4.2 The Two-Category Case

- For a discriminant function of the form of eq.3.1, a two-category classifier implements the following decision rule:
- Decide w_1 if $g(\mathbf{x})>0$ and w_2 if $g(\mathbf{x})<0$.
- Thus, **x** is assigned to w_1 if the inner product $\mathbf{w}^T\mathbf{x}$ exceeds the threshold – w_0 and to w_2 otherwise.
- If $g(\mathbf{x})=0$, **x** can ordinarily be assigned to either class, or can be left undefined.
- The equation $g(\mathbf{x})=0$ defines the decision surface that separates points assigned to w_1 from points assigned to w_2.
- When g(x) is linear, this decision surface is a hyperplane.
- If $\mathbf{x}_1$ and $\mathbf{x}_2$ are both on the decision surface, then

$$\mathbf{w}^T\mathbf{x}_1 + w_0 = \mathbf{w}^T\mathbf{x}_2 + w_0 \quad (3.2)$$

or

$$\mathbf{w}^T(\mathbf{x}_1 - \mathbf{x}_2) = 0 \quad (3.3)$$

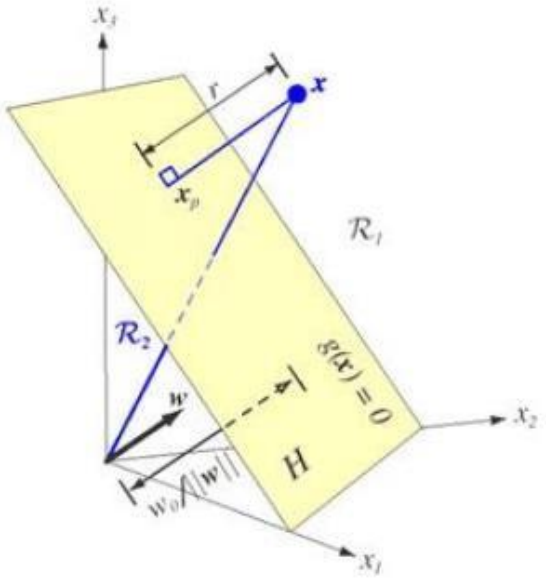


Figure 3.7: The linear decision boundary H separates the feature space into two half-spaces.

- In figure 3.7, the hyperplane H divides the feature space into two half-spaces:
  - Decisionregion R_1 for w_1
  - region R_2 for w_2.
- The discriminant function $g(\mathbf{x})$ gives an algebraic measure of the distance from $\mathbf{x}$ to the hyperplane.
- The way to express $\mathbf{x}$ as

$$\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|} \tag{3.4}$$

where $\mathbf{x}_p$ is the normal projection of $\mathbf{x}$ onto H, and r is the desired algebraic distance which is positive if $\mathbf{x}$ is on the positive side and negative if $\mathbf{x}$ is on the negative side. Then, because $g(\mathbf{x}_p)=0$,

$$\mathbf{x}_p = \mathbf{x} - r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

$$g(\mathbf{x}_p) = \mathbf{w}^T\mathbf{x}_p + w_0 = 0$$

$$g(\mathbf{x}_p) = \mathbf{w}^T\left(\mathbf{x} - r\frac{\mathbf{w}}{\|\mathbf{w}\|}\right) + w_0 = 0$$

$$\mathbf{w}^T\mathbf{x} - r\frac{\mathbf{w}^T\mathbf{w}}{\|\mathbf{w}\|}) + w_0 = 0$$

Since $\mathbf{w}^T\mathbf{w} = \|\mathbf{w}\|^2$ then

$$\mathbf{w}^T\mathbf{x} + w_0 - r\|\mathbf{w}\| = 0$$

$$\mathbf{w}^T\mathbf{x} + w_0 = r\|\mathbf{w}\|$$

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = r\|\mathbf{w}\| \tag{3.5}$$

or

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|} \tag{3.6}$$

- The distance from the origin to H is given by $\frac{w_0}{\|\mathbf{w}\|}$.
- If $w_0>0$, the origin is on the positive side of H, and if $w_0<0$, it is on the negative side.
- If $w_0=0$, then $g(\mathbf{x})$ has the homogeneous form $\mathbf{w}^T\mathbf{x}$, and the hyperplane passes through the origin

4.3 The Multi-category Case

- To devise multi category classifiers employing linear discriminant functions reduce the problem to c two-class problems.
- Defining c linear discriminant functions

$$g_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + w_{i0} \qquad i = 1,\ldots,c \tag{3.7}$$

and assigning $\mathbf{x}$ to w_i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all j^1 i; in case of ties, the classification is left undefined.

- The resulting classifier is called a *linear machine.*

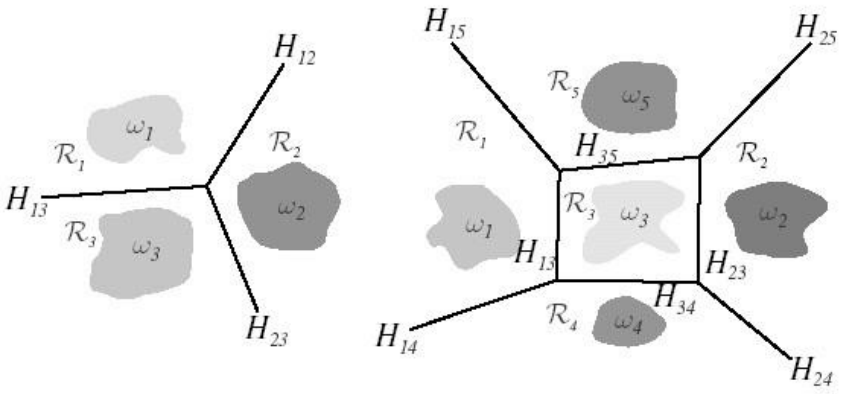


Figure 3.8: Decision boundaries defined by linear machines

- A linear machine divides the feature space into c decision regions as shown in figure 3.8, with $g_j(\mathbf{x})$ being the largest discriminant if $\mathbf{x}$ is in region R_i.
- If R_i and R_j are contiguous, the boundary between them is a portion of the hyperplane H_{ij} defined by

$$g_i(\mathbf{x}) = g_j(\mathbf{x}) \quad \text{or} \quad (\mathbf{w}_i - \mathbf{w}_j)^T \mathbf{x} + (w_{i0} - w_{j0}) = 0$$

$\mathbf{w}_i - \mathbf{w}_j$ is normal to H_{ij} and the signed distance from $\mathbf{x}$ to H_{ij} is given by

$$\frac{(g_i(\mathbf{x}) - g_j(\mathbf{x}))}{\|\mathbf{w}_i - \mathbf{w}_j\|}$$

4.4 Generalized Linear Discriminant Functions

- The linear discriminant function $g(\mathbf{x})$ can be written as

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i \tag{3.8}$$

where the coefficients w_i are the components of the weight vector $\mathbf{w}$.

Quadratic Discriminant Function

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} x_i x_j \quad (3.9)$$

4.5 Probabilistic discriminative models

- ***Discriminative models*** are a class of supervised machine learning models which make predictions by estimating conditional probability *P(y|x).*
- For the two-class classification problem, the posterior probability of classC1 can be written as a logistic sigmoid acting on a linear function of x

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma\left(\ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}\right) = \sigma(\mathbf{w}^T\mathbf{x} + w_0)$$

- For the multi-class case, the posterior probability of class Ckis given by a softmax transformation of a linear function of x

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_{j=1}^{K} p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp\left(\mathbf{w}_k^T\mathbf{x} + w_{k0}\right)}{\sum_{j=1}^{K}\exp\left(\mathbf{w}_j^T\mathbf{x} + w_{j0}\right)}$$

4.6 Logistics Regression

- Logistic regression is the Machine Learning algorithms, under the classification algorithm of Supervised Learning technique.
- Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables.
- The independent variables can be nominal, ordinal, or of interval type.
- Logistic regression predicts the output of a categorical dependent variable.
- Therefore the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. it gives the probabilistic values which lie between 0 and 1.
- Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- The figure 3.9 predicts the logistic function

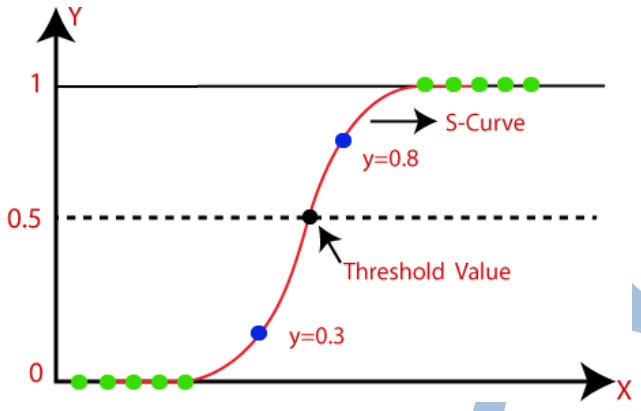


Figure 3.9 - Logistic Function or Sigmoid Function

4.6.1 Logistic Function (Sigmoid Function):

- The logistic function is also known as the sigmoid function.
- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- The value of the logistic regression must be between 0 and 1, so it forms a curve like the "S" form.
- The S-form curve is called the Sigmoid function or the logistic function.

4.6.2 Assumptions for Logistic Regression:

- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.

4.6.3 Logistic Regression Equation:

- The Logistic regression equation can be obtained from the Linear Regression equation.
- The **mathematical steps** are given below:
  - The equation of the straight line can be written as:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

  - In Logistic Regression y can be between 0 and 1 only, let's divide the above equation by (1-y):

$$\frac{y}{1-y}\text{; 0 for y= 0, and infinity for y=1}$$

  - For the range between -[infinity] to +[infinity], take logarithm of the equation:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

4.6.4 Type of Logistic Regression:

- Logistic Regression can be classified into three types:
  - **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
  - **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
  - **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

4.6.5 Steps in Logistic Regression:

- To implement the Logistic Regression using Python, the steps are given below:
  - Data Pre-processing step
  - Fitting Logistic Regression to the Training set
  - Predicting the test result
  - Test accuracy of the result
  - Visualizing the test set result.

4.6.6 Advantages of Logistic Regression Algorithm

- Logistic regression performs better when the data is linearly separable
- It does not require too many computational resources
- There is no problem scaling the input features
- It is easy to implement and train a model using logistic regression

5. Elaborate in detail about Probabilistic Generative model and Naïve Bayes.

PROBABILISTIC GENERATIVE MODEL AND NAÏVE BAYES

5.1 Probabilistic Generative model
5.2 Simple example
5.3 Generative models
5.4 Discriminative models

5.1 Probabilistic Generative model

- Given a model of one conditional probability, and estimated probability distributions for the variables *X* and *Y*, denoted P(X) and P(Y), can estimate the conditional probability using Bayes' rule:

$$P(X \mid Y)P(Y) = P(Y \mid X)P(X).$$

- A **generative model** isa statistical model of the joint probability distribution on given observable variable *X* and target variable *Y*.
  Given a generative model for P(X|Y), can estimate:

$$P(Y \mid X) = P(X \mid Y)P(Y)/P(X).$$

- A **discriminative model** is a model of the conditional probability of the target *Y*, given an observation *x*given a discriminative model forP(Y|X), can estimate:

$$P(X \mid Y) = P(Y \mid X)P(X)/P(Y).$$

- Classifier based on a generative model is a **generative classifier**, while a classifier based on a discriminative model is a **discriminative classifier**

5.2 Simple example

Suppose the input data is $x \in \{1, 2\}$, the set of labels for x is $y \in \{0, 1\}$, and there are the following 4 data points:
$(x, y) = \{(1, 0), (1, 1), (2, 0), (2, 0)\}$

For the above data, estimating the joint probability distribution $p(x, y)$ from the empirical measure will be the following:

	$y = 0$	$y = 1$
$x = 1$	$1/4$	$1/4$
$x = 2$	$2/4$	0

while $p(y|x)$ will be following:

	$y = 0$	$y = 1$
$x = 1$	$1/2$	$1/2$
$x = 2$	1	0

5.3 Generative models

Types of generative models are:

- Naive Bayes classifier or Bayesian network
- Linear discriminant analysis

5.4 Discriminative models

- Logistic regression
- Support Vector Machines
- Decision Tree Learning
- Random Forest

6. Elaborate in detail about Support Vector Machine (SVM).

SUPPORT VECTOR MACHINE

6.1 Support Vector Machine (SVM)
6.2 Cost Function and Gradient Updates
6.2.1 Hinge loss function
6.3 SVM Kernel
6.4 Types of SVMs
6.5 Advantages of SVM
6.6 Disadvantages
6.7 Applications

6.1 Support Vector Machine (SVM)

- Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression.
- The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.
- Hyperplanes are decision boundaries that help classify the data points.
- The dimension of the hyperplane depends upon the number of features.
- If the number of input features is 2, then the hyperplane is just a line.
- If the number of input features is 3, then the hyperplane becomes a two-dimensional plane.
- It becomes difficult to imagine when the number of features exceeds 3.
- The objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes.
- Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane.
- Using these support vectors, can maximize the margin of the classifier.
- Deleting the support vectors will change the position of the hyperplane.
- Example Refer Figure 3.10

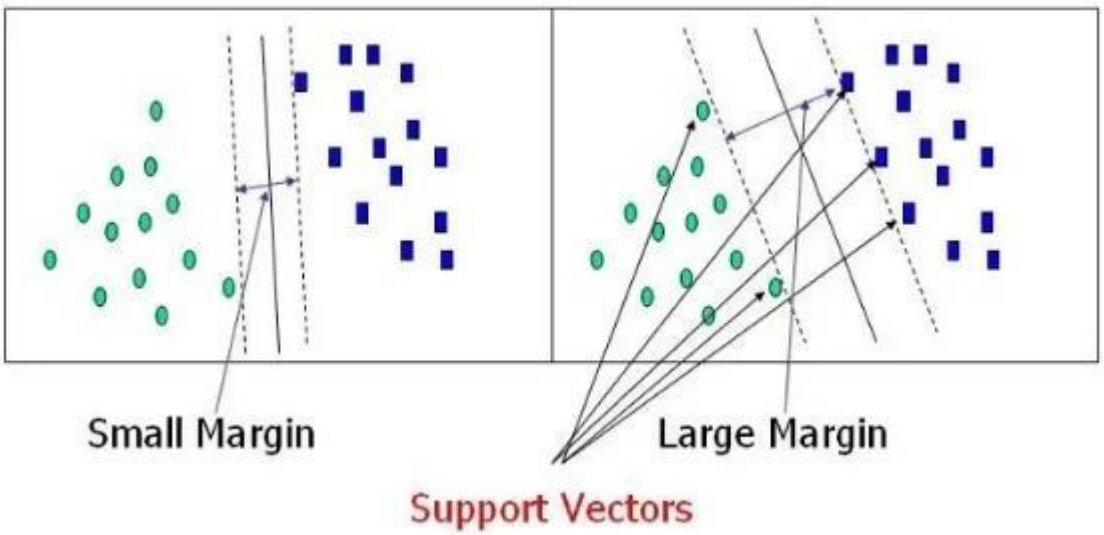


Figure 3.10 - Example for Support Vectors

- Let's consider two independent variables x1, x2 and one dependent variable which is either a blue circle or a red box as shown in figure 3.11.

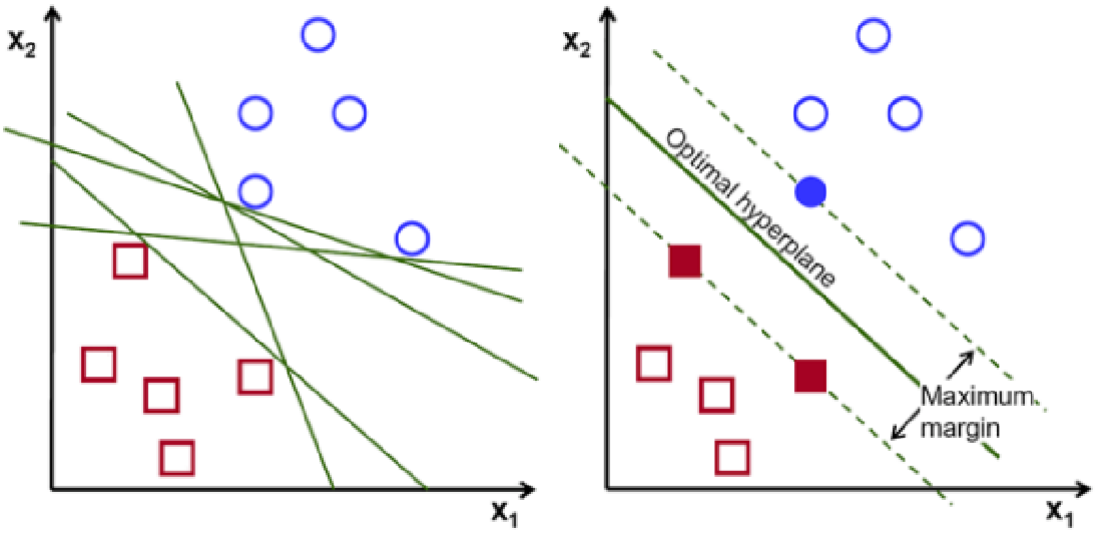


Figure 3.11 - Linearly Separable Data points