R.M.K

GROUP OF ENGINEERING INSTITUTIONS

R.M.K

GROU’POF INSTITUTIONS

Please read this disclaimer before proceeding:

This document is confidential and intended solely for the educational purpose of RMK Group of Educational Institutions. If you have received this document through email in error, please notify the system manager. This document contains proprietary information and is intended only to the respective group / learning community as intended. If you are not the addressee you should not disseminate, distribute or copy through e-mail. Please notify the sender immediately by e-mail if you have received this document by mistake and delete this document from your system. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited.

22AI401 MACHINE LEARNING (LAB INTEGRATED)

Department :AI&DS

Batch/Year : 2022-2026 / II

Created by : Dr. G. Sangeetha & Ms. G. Mageshwari

Date : 17.02.2024

1. TABLEOF CONTENTS

Table of Contents

Course Objectives

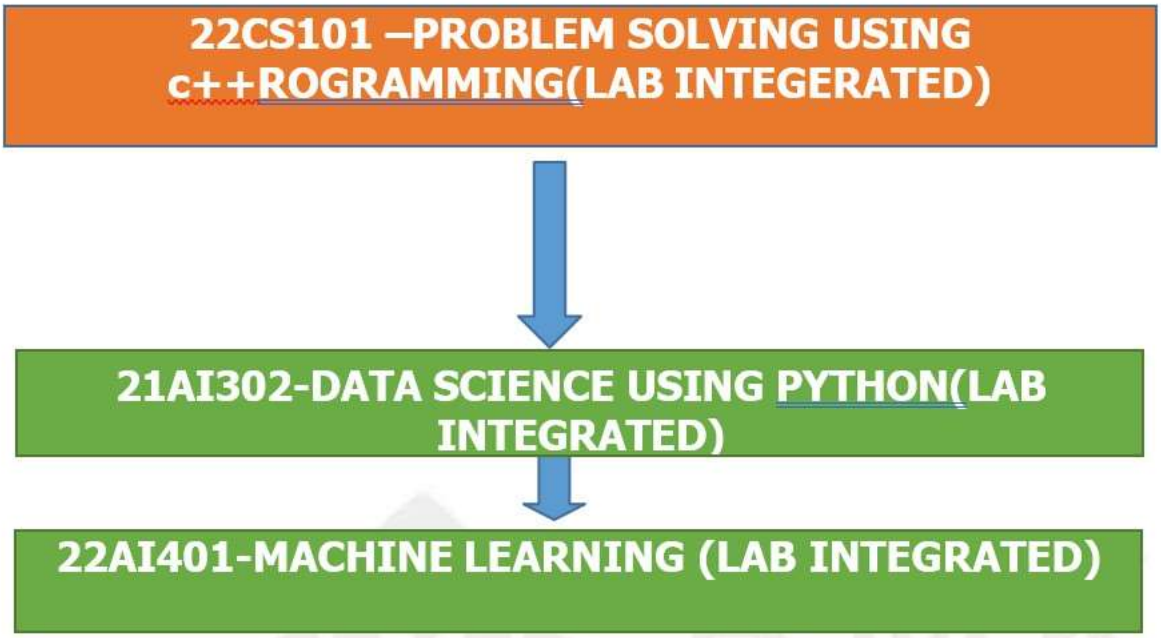
2. COURSE OBJECTIVES

- To discuss the basics of Machine Learning and Supervised Algorithms.
- To understand the various classification algorithms.
- To study dimensionality reduction techniques.
- To elaborate on unsupervised learning techniques.
- To discuss various Graphical models and understand thebasics of reinforcement learning.

PRE REQUISITES

3. PRE REQUISITES

- PRE-REQUISITE CHART



22CS101 –PROBLEM SOLVING USING c++ROGRAMMING(LAB INTEGERATED)

21AI302-DATA SCIENCE USING PYTHON(LAB INTEGRATED)

22AI401-MACHINE LEARNING (LAB INTEGRATED)

4. SYLLABUS

22AI401 - MACHINE LEARNING (LAB INTEGRATED) L P T C

3 0 2 4

OBJECTIVE:

- To discuss the basics of Machine Learning and model evaluation.
- To study dimensionality reduction techniques.
- To understand the various classification algorithms.
- To elaborate on unsupervised learning techniques.
- To discuss the basics of neural networks and various types of learning.

UNIT I INTRODUCTION 9+6

Machine Learning – Types – Applications – Preparing to Model – Activities – Data – Exploring-structure ofData – Data Quality and Remediation – Data Pre-processing – Modelling and Evaluation: Selecting a Model -Training a Model – Model representation and Interpretability – Evaluating Performance of a Model –Improving Performance.

Lab Programs:

1. Implementation of Candidate Elimination algorithm
2. Implementation of ML model evaluation techniques (R-Squared/Adjusted R-Squared/Mean AbsoluteError/Mean Squared Error)

Implementation of ML model evaluation techniques (Confusion Matrix/F1 Score/AUC-ROC Curve)

UNIT II FEATURE ENGINEERING AND DIMENSIONALITY REDUCTION 9+6

Feature Engineering – Feature Transformation – Feature Subset Selection - Principle Component Analysis– Feature Embedding – Factor Analysis – Singular value decomposition and Matrix Factorization – Multidimensional scaling – Linear Discriminant Analysis – Canonical Correlation Analysis – Isomap – Locally linear Embedding – Laplacian Eigenmaps.

Lab Programs:

1. Write python code to identify feature co-relations (PCA)
2. Interpret Canonical Covariates with Heatmap
3. Feature Engineering is the way of extracting features from data and transforming them into formats that are suitable for Machine Learning algorithms. Implement python code for Feature Selection/ Feature Transformation/ Feature Extraction.
4. Mini Project – Feature Subset Selection

UNIT III SUPERVISED LEARNING 9+6

Linear Regression -Relation between two variables – Steps – Evaluation – Logistic Regression –Decision Tree – Algorithms – Construction – Classification using Decision Tree – Issues – Rule- based Classification – Pruning the Rule Set – Support Vector Machines – Linear SVM – Optimal Hyperplane – Radial Basis Functions – Naïve Bayes Classifier – Bayesian Belief Networks.

Lab Programs:

1. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points.Select the appropriate data set for your experiment and draw graphs.

2. Implement and demonstrate the working of the decision tree-based ID3 algorithmBuild a Simple Support Vector Machines using a data set

UNIT IV UNSUPERVISED LEARNING 9+6

Clustering – Types – Applications - Partitioning Methods – K-means Algorithm – K-Medoids –

Hierarchicalmethods – Density based methods DBSCAN – Finding patterns using Association Rules – Hidden Markov Model.

Lab Programs:

1. Implement a k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrongpredictions
2. Implement market basket analysis using association rules
3. Mini Project using Clustering analysis

UNIT V NEURAL NETWORKS AND TYPES OF LEARNING 9+6

Biological Neuron – Artificial Neuron – Types of Activation function – Implementations of ANN – Architectures of Neural Networks – Learning Process in ANN – Back propagation – Deep Learning – Representation Learning – Active Learning – Instance based Learning – Association Rule Learning – Ensemble Learning Algorithm – Regularization Algorithm- Reinforcement Learning – Elements- Model-based- Temporal Difference Learning.

Lab Programs:

1. Build an ANN by implementing the Single-layer Perceptron. Test it using appropriate data sets.
2. Implement Multi-layer Perceptron and test the same using appropriate data sets.
3. Build a RBF Network to calculate the fitness function with five neurons.
4. Mini Project – Face recognition,

TOTAL: 45+30 = 75 PERIODS

OUTCOMES:

At the end of this course, the students will be able to:

CO1: Explain the basics of Machine Learning and model evaluation.

CO2: Study dimensionality reduction techniques.

CO3: Understand and implement various classification algorithms.

CO4: Understand and implement various unsupervised learning techniques.CO5: Build Neural Networks and understand the different types of learning.

TEXT BOOKS:

1. Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, "Machine Learning", Pearson,2019.(Unit 1 – chap 1,2,3/ Unit 2 – Chap 4 / Unit 4 – 9 / Unit 5 – Chap 10, 11)

Ethem Alpaydin, "Introduction to Machine Learning, Adaptive Computation and Machine LearningSeries", Third Edition, MIT Press, 2014. (Unit 2 – Chap 6 / Unit 4 – chap 8.2.3/ Unit 5 – Chap 18)

REFERENCES:

1. Anuradha Srinivasaraghavan,Vincy Joseph, "Machine Learning", First Edition, Wiley, 2019.(Unit3 – Chap 7,8,9,10,11 / Unit 4 – 13, 11.4, 11.5,12)
2. Peter Harrington, "Machine Learning in Action", Manning Publications, 2012.
3. Stephen Marsland, "Machine Learning – An Algorithmic Perspective", Second Edition,
4. Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.
5. Tom M Mitchell, "Machine Learning", First Edition, McGraw Hill Education, 2013.
6. Christoph Molnar, "Interpretable Machine Learning - A Guide for Making Black Box Models
Explainable", Creative Commons License, 2020.
7. NPTEL Courses:

Introduction to Machine Learning - https://onlinecourses.nptel.ac.in/noc23_cs18/preview

Course Outcomes

5. COURSE OUTCOME

Course Code	Course Outcome Statement	Cognitive / Affective Level of the Course Outcome	Course Outcome
Course Outcome Statements in Cognitive Domain			
22AI401	Explain the basics of Machine Learning and Supervised Algorithms	Apply K3	CO1
22AI401	Understand the various classification algorithms.	Apply K3	CO2
22AI401	Study dimensionality reduction techniques	Apply K3	CO3
22AI401	Elaborate on unsupervised learning techniques	Apply K4	CO4
22AI401	Understand various Graphical models and understand the basics of reinforcement learning	Apply K4	CO5

CO / PO Mapping

6. CO-PO/PSO MAPPING

Correlation Matrix of the Course Outcomes to Programme Outcomes and Programme Specific Outcomes.

Course Outcomes (Cos)		Programme Outcomes (POs), Programme Specific Outcomes (PSOs)														
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
22AI401.1	K2	3	3	1	-	-	-	-	-	-	-	-	-	2	2	2
22AI401.2	K3	3	2	1	-	-	-	-	-	-	-	-	-	2	2	2
22AI401.3	K3	3	2	1	-	-	-	-	-	-	-	-	-	2	2	2
22AI401.4	K3	3	3	2	-	-	-	-	-	-	-	-	-	2	2	2
22AI401.5	K2	3	2	2	-	-	-	-	-	-	-	-	-	2	2	2

UNIT III

SUPERVISED LEARNING

LECTURE PLAN – UNIT III

Sl. No	UNIT III TOPIC	NO OF PERI O DS	PROPOSE D LECTUR E	ACTUA L LECTUR E	PERTAI N I NG CO(s)	TAXONO M Y LEVEL	MODE OF DELIVER Y
			PEROID	PERIOD			
1	Linear Regression - Relation between two variables – Steps	1	22.2.24		CO3	K3	MD1 , MD5
2	Evaluation – Logistic Regression	1	22.2.24		CO3	K3	MD1 , MD 5
3	Implementation of regression Algorithm in order to fit data points	1	23.2.24		CO3	K3	MD1 , MD5
4	Select the appropriate data set for your experiment and draw graphs	1	24.2.24		CO3	K3	MD1 , MD 5
5	Decision Tree - Algorithms	1	24.2.24		CO3	K3	MD1 , MD5
6	Construction – Classification using Decision Tree	1	26.2.24		CO3	K3	MD1 , MD5
7	Implentation of Decision Tree algorithm based on ID3 algorithm	1	28.2.24		CO3	K3	MD1,MD5
8	Issues – Rule-based Classification	1	29.2.24		CO3	K3	MD1 , MD5
9	Pruning the Rule Set	1	29.2.24		CO3	K3	MD1 , MD5

10	Support Vector Machines	1	1.03.24		CO3	K3	MD1 , MD5
11	Linear Support Vector Machine – Optimal Hyperplane	1	1.03.24		CO3	K3	MD1 , MD5
12	Build Support Vector Machines using a dataset	1	2.03.24		CO3	K3	MD1 , MD5
13	Bayesian Belief Networks	1	2.03.24		CO3	K3	MD1 , MD5
14	Radial Basis Functions	1	7.03.24		CO3	K3	MD1 , MD5
15	Naive Bayes Classifier	1	8.03.24		CO3	K3	MD1 , MD5

LECTURE PLAN – UNIT II

ASSESSMENT COMPONENTS

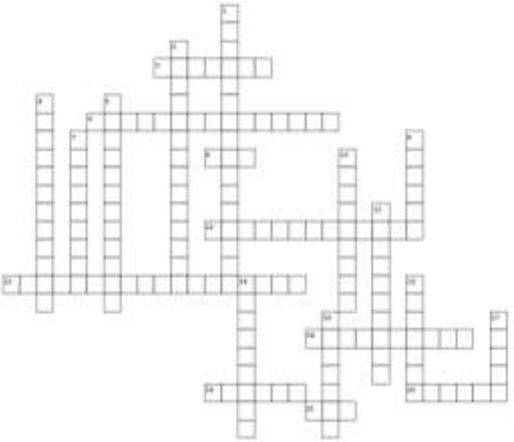
- AC 1. Unit Test
- AC 2. Assignment
- AC 3. Course Seminar
- AC 4. Course Quiz
- AC 5. Case Study
- AC 6. Record Work
- AC 7. Lab / Mini Project
- AC 8. Lab Model Exam
- AC 9. Project Review

MODE OF DELEIVERY

MD 1. Oral presentation
MD 2. Tutorial
MD 3. Seminar
MD 4 Hands On
MD 5. Videos
MD 6. Field Visit

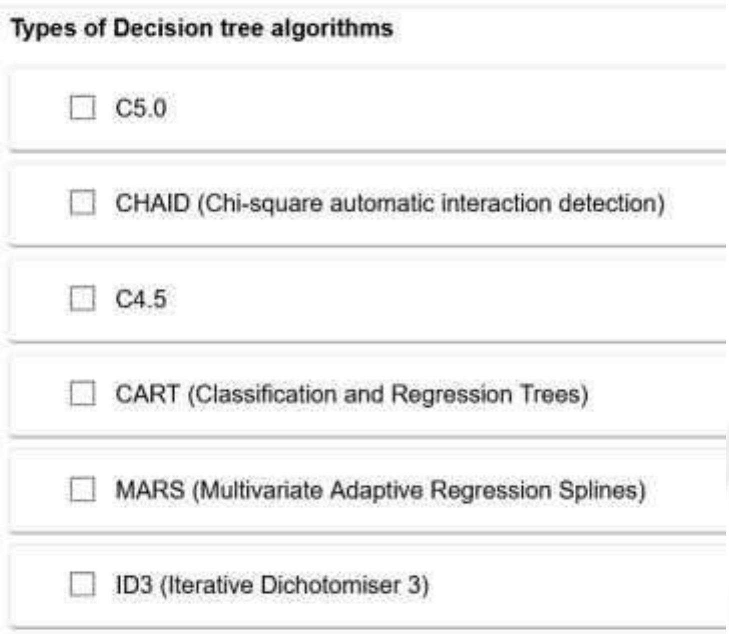
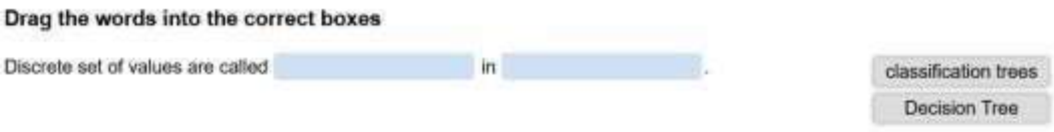
8. ACTIVITY BASED LEARNING : UNIT – III

ACTIVITY 1:

S NO	TOPICS
1	**Cross word Puzzle**  **Across** **3. In machine learning, a mechanism for** bucketing categorical data **6. The primary algorithm for performing** gradient descent on neural networks. **9. Abbreviation for independently and** identically distributed **12. The more common label in a class-**imbalanced dataset. **13. Applying a constraint to an algorithm to** ensure one or more definitions of fairness oarre satisfied. **18.A process used, as part of training, to** evaluate the quality of a machine learning model using the validation set. **19. A coefficient for a feature in a linear** model, or an edge in a deep network. **20. A column-oriented data analysis API.** **21.Abbreviation for generative adversarial** network **Down** **1.A post-prediction adjustment, typically to** account for prediction bias. **2.A TensorFlow programming environment in** which operations run immediately. **4.Obtaining an understanding of data by** considering samples, measurement, and visualization. **5. An ensemble approach to finding the** decision tree that best fits the training data **7. state-action value function** **8.Loss function based on the absolute value of** the difference between the values that a model is predicting and the actual values of the labels **10. A metric that your algorithm is trying to** optimize. **11. The recommended format for saving and** recovering TensorFlow models. **14. A statistical way of comparing two (or** more) techniques, typically an incumbent against a new rival. **15. When one number in your model becomes** a NaN during training, which causes many all other numbers in your model to eventually become a NaN. **16. Q-learning In reinforcement learning,** implementing Q-learning by using a table to store the Q-functions **17. A popular Python machine learning API**

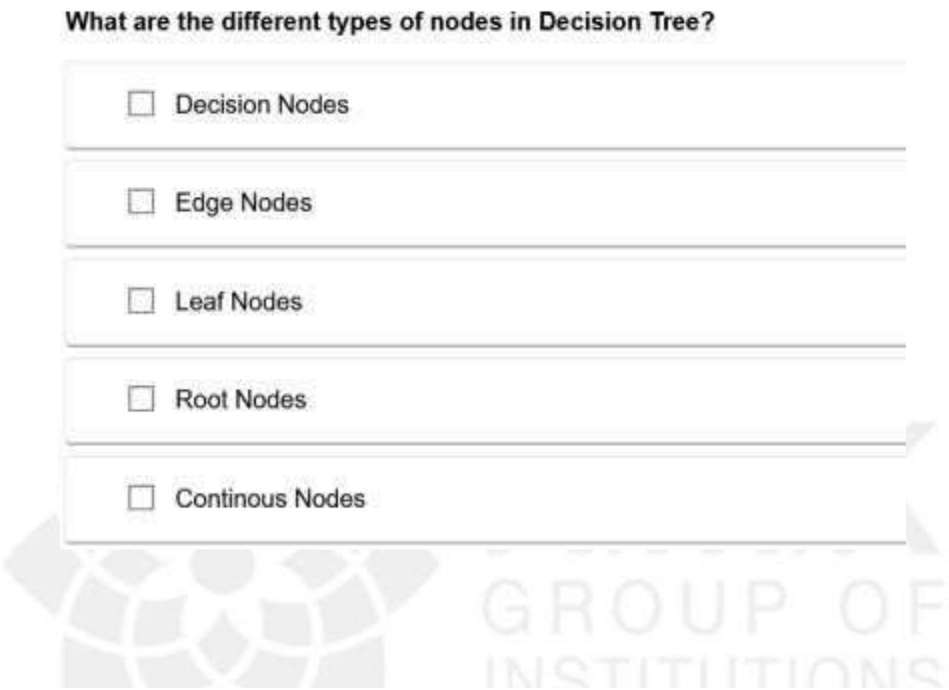
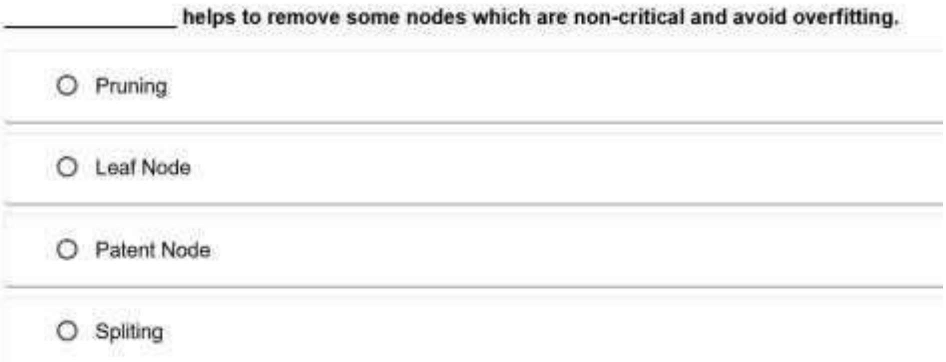
ACTIVITY BASED LEARNING

(MODEL BUILDING/PROTOTYPE)

S NO	TOPICS
	Work Sheet
1	 Types of Decision tree algorithms □ C5.0 □ CHAID (Chi-square automatic interaction detection) □ C4.5 □ CART (Classification and Regression Trees) □ MARS (Multivariate Adaptive Regression Splines) □ ID3 (Iterative Dichotomiser 3)
2	 The Decision Tree algorithm belongs to the supervised learning algorithm. ○ True ○ False
3	 Drag the words into the correct boxes Discrete set of values are called ________ in ________. classification trees Decision Tree

ACTIVITY BASED LEARNING

(MODEL BUILDING/PROTOTYPE)

S NO	TOPICS
4	**Work Sheet**  What are the different types of nodes in Decision Tree? □ Decision Nodes □ Edge Nodes □ Leaf Nodes □ Root Nodes □ Continous Nodes
5	 The ______ cannot be split into further node called ______
6	 ______________ helps to remove some nodes which are non-critical and avoid overfitting. ○ Pruning ○ Leaf Node ○ Patent Node ○ Spliting

8. ACTIVITY BASED LEARNING : UNIT – III

ACTIVITY 1: Build a machine learning model on Binary Classification

Binary classification refers to those classification tasks that have two class labels.

Examples include:

- Email spam detection (spam or not).
- Churn prediction (churn or not).
- Conversion prediction (buy or not).

Typically, binary classification tasks involve one class that is the normal state and another class that is the abnormal state. For example "not spam" is the normal state and "spam" is the abnormal state. Another example is "cancer not detected" is the normal state of a task that involves a medical test and "cancer detected" is the abnormal state.

The class for the normal state is assigned the class label 0 and the class with the abnormal state is assigned the class label 1. It is common to model a binary classification task with a model that predicts a Bernoulli probability distribution for each example.

The Bernoulli distribution is a discrete probability distribution that covers a case where an event will have a binary outcome as either a 0 or 1. For classification, this means that the model predicts a probability of an example belonging to class 1, or the abnormal state.

Some algorithms are specifically designed for binary classification and do not natively support more than two classes; examples include Logistic Regression and Support Vector Machines.

Next, let's take a closer look at a dataset to develop an intuition for binary classification problems.

We can use the make_blobs() function to generate a synthetic binary classification dataset.

9. LECTURE NOTES : UNIT – III

SUPERVISED LEARNING

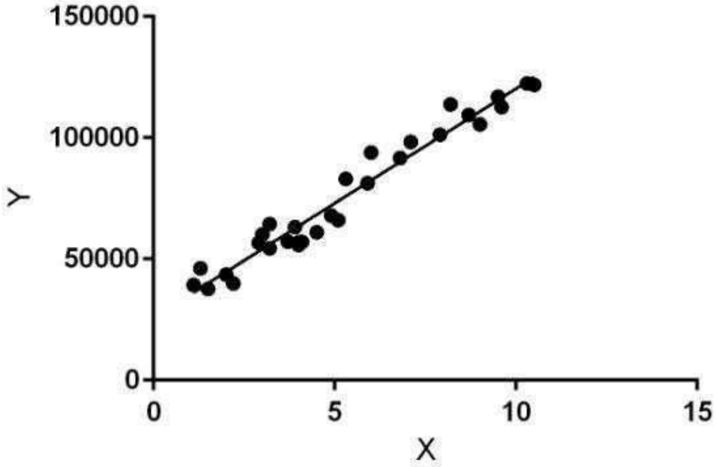
Syllabus:

Linear Regression -Relation between two variables –Steps – Evaluation – Logistic Regression – Decision Tree – Algorithms – Construction – Classification using Decision Tree – Issues – Rule-based Classification –Pruning the Rule Set – Support Vector Machines – Linear SVM – Optimal Hyperplane – Radial Basis Functions – Naïve Bayes Classifier – Bayesian Belief Networks.

UNIT-3
LECTURE NOTES-SUPERVISED LEARNING

3.1 Linear Regression

Linear Regression is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.



Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person.The regression line is the best fit line for our model.

Hypothesis function for Linear Regression :

$$y = \theta_1 + \theta_2.x$$

While training the model we are given :
x: input training data (univariate – one input variable(parameter))
y: labels to data (supervised learning)
When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ_1 and θ_2 values.
θ_1: intercept
θ_2: coefficient of x

Once we find the best θ_1 and θ_2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

How to update θ_1 and θ_2 values to get the best fit line ?

Cost Function (J):

By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ_1 and θ_2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).

$$minimize \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

Cost function(J) of Linear Regression is the **Root Mean Squared Error (RMSE)** between predicted y value (pred) and true y value (y).

Gradient Descent:

To update θ_1 and θ_2 values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random θ_1 and θ_2 values and then iteratively updating the values, reaching minimum cost.

3.2 LOGISTIC REGRESSION

This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas:

$Logit(pi) = 1/(1+ exp(-pi))$

$ln(pi/(1-pi)) = Beta\_0 + Beta\_1*X\_1 + … + B\_k*K\_k$

In this logistic regression equation, logit(pi) is the dependent or response variable and x is the independent variable. The beta parameter, or coefficient, in this model is commonly estimated via maximum likelihood estimation (MLE). This method tests different values of beta through multiple iterations to optimize for the best fit of log odds. All of these iterations produce the log likelihood function, and logistic regression seeks to maximize this function to find the best parameter estimate. Once the optimal coefficient (or coefficients if there is more than one independent variable) is found, the conditional probabilities for each observation can be calculated, logged, and summed together to yield a predicted probability. For binary classification, a probability less than .5 will predict 0 while a probability greater than 0 will predict 1. After the model has been computed, it's best practice to evaluate the how well the model predicts the dependent variable, which is called goodness of fit. The Hosmer–Lemeshow test is a popular method to assess model fit.

1. Interpreting logistic regression

Log odds can be difficult to make sense of within a logistic regression data analysis. As a result, exponentiating the beta estimates is common to transform the results into an odds ratio (OR), easing the interpretation of results. The OR represents the odds that an outcome will occur given a particular event, compared to the odds of the outcome occurring in the absence of that event. If the OR is greater than 1, then the event is associated with a higher odd of generating a specific outcome. Conversely, if the OR is less than 1, then the event is associated with a lower odd of that outcome occurring. Based on the equation from above, the interpretation of an odds ratio can be denoted as the following: the odds of a success changes by exp(cB_1) times for every c-unit increase in x. To use an example, let's say that we were to estimate the odds of survival on the Titanic given that the person was male, and the odds ratio for males was .0810. We'd interpret the odds ratio as the odds of survival of males decreased by a factor of .0810 when compared to females, holding all other variables constant.

2. Types of logistic regression

There are three types of logistic regression models, which are defined based on categorical response.

- **Binary logistic regression:** In this approach, the response or dependent variable is dichotomous in nature—i.e. it has only two possible outcomes (e.g. 0 or 1). Some popular examples of its use include predicting if an e-mail is spam or not spam or if a tumor is malignant or not malignant. Within logistic regression, this is the most commonly used approach, and more generally, it is one of the most common classifiers for binary classification.

- **Multinomial logistic regression:** In this type of logistic regression model, the dependent variable has three or more possible outcomes; however, these values have no specified order. For example, movie studios want to predict what genre of film a moviegoer is likely to see to market films more effectively. A multinomial logistic regression model can help the studio to determine the strength of influence a person's age, gender, and dating status may have on the type of film that they prefer. The studio can then orient an advertising campaign of a specific movie toward a group of people likely to go see it.
- **Ordinal logistic regression:** This type of logistic regression model is leveraged when the response variable has three or more possible outcome, but in this case, these values do have a defined order. Examples of ordinal responses include grading scales from A to F or rating scales from 1 to 5.

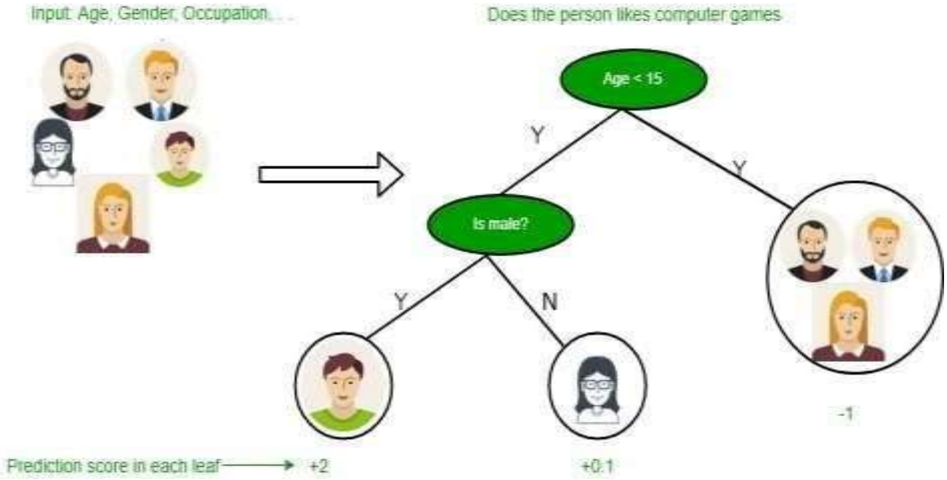
3. Use cases of logistic regression

Logistic regression is commonly used for prediction and classification problems. Some of these use cases include:

- **Fraud detection:** Logistic regression models can help teams identify data anomalies, which are predictive of fraud. Certain behaviors or characteristics may have a higher association with fraudulent activities, which is particularly helpful to banking and other financial institutions in protecting their clients. SaaS-based companies have also started to adopt these practices to eliminate fake user accounts from their datasets when conducting data analysis around business performance.
- **Disease prediction:** In medicine, this analytics approach can be used to predict the likelihood of disease or illness for a given population. Healthcare organizations can set up preventative care for individuals that show higher propensity for specific illnesses.
- **Churn prediction**: Specific behaviors may be indicative of churn in different functions of an organization. For example, human resources and management teams may want to know if there are high performers within the company who are at risk of leaving the organization; this type of insight can prompt conversations to understand problem areas within the company, such as culture or compensation. Alternatively, the sales organization may want to learn which of their clients are at risk of taking their business elsewhere. This can prompt teams to set up a retention strategy to avoid lost revenue.

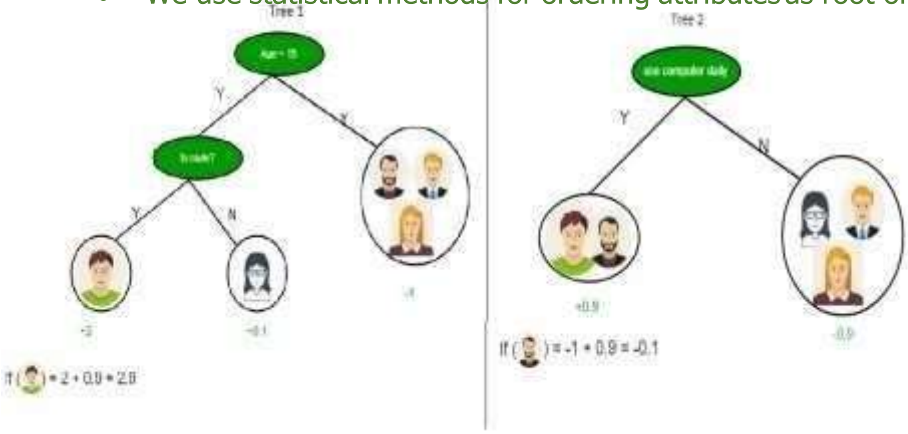
3.3 DECISION TREE

- Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.
- Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.
- We can represent any boolean function on discrete attributes using the decision tree.


Input: Age, Gender, Occupation, . . .
Does the person likes computer games
Age < 15
Y
Y
Is male?
Y
N
Prediction score in each leaf
+2
+0.1
-1

Below are some assumptions that we made while using decision tree:

- At the beginning, we consider the whole training set as the root.
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- On the basis of attribute values records are distributed recursively.
- We use statistical methods for ordering attributes as root or the internal node.



As you can see from the above image that Decision Tree works on the Sum of Product form which is also known as Disjunctive Normal Form. In the above image, we are predicting the use of computer in the daily life of the people. In Decision Tree the major challenge is to identification of the attribute for the root node in each level.

This process is known as attribute selection. We have two popular attribute selection measures:

1. Information Gain
2. Gini Index

1. Information Gain

When we use a node in a decision tree to partition the training instances into smaller subsets the entropy changes. Information gain is a measure of this change in entropy. Definition: Suppose S is a set of instances, A is an attribute, Sv is the subset of S with A = v, and Values (A) is the set of all possible values of A, then

Entropy

Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy more the information content.

Definition: Suppose S is a set of instances, A is an attribute, Sv is the subset of S with A = v, and Values (A) is the set of all possible values of A, then

Example:

For the set X = {a,a,a,b,b,b,b,b}

Total instances: 8

Instances of b: 5

Instances of a: 3

= -[0.375 * (-1.415) + 0.625 * (-0.678)]

=-(-0.53-0.424)

= 0.954

Building Decision Tree using Information Gain

The essentials:

- Start with all training instances associated with the root node
- Use info gain to choose which attribute to label each node with
- Note: No root-to-leaf path should contain the same discrete attribute twice
- Recursively construct each subtree on the subset of training instances that would be classified down that path in the tree.

The border cases:

- If all positive or all negative training instances remain, label that node “yes” or “no” accordingly

- If no attributes remain, label with a majority vote of training instances left at that node
- If no instances remain, label with a majority vote of the parent’s training instances

Example:

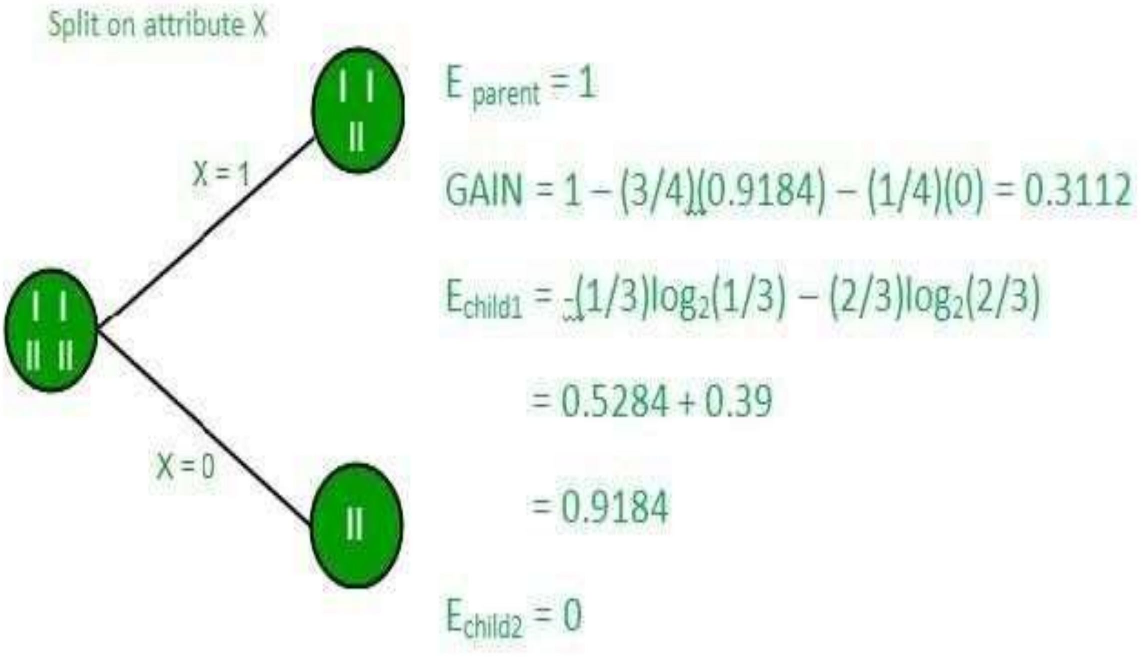
Now, lets draw a Decision Tree for the following data using Information gain.

Training set: 3 features and 2 classes

X	Y	Z	C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

Here, we have 3 features and 2 output classes.

To build a decision tree using Information gain. We will take each of the feature and calculate the information for each feature.



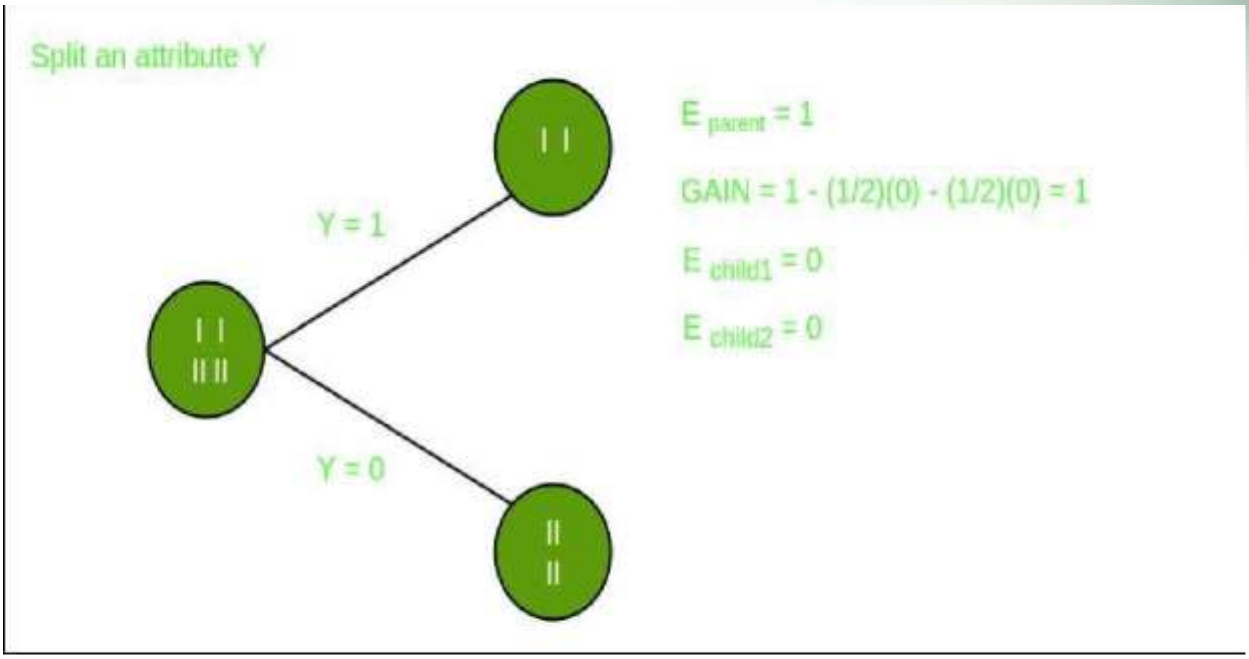
$E_{parent} = 1$

$GAIN = 1 - (3/4)(0.9184) - (1/4)(0) = 0.3112$

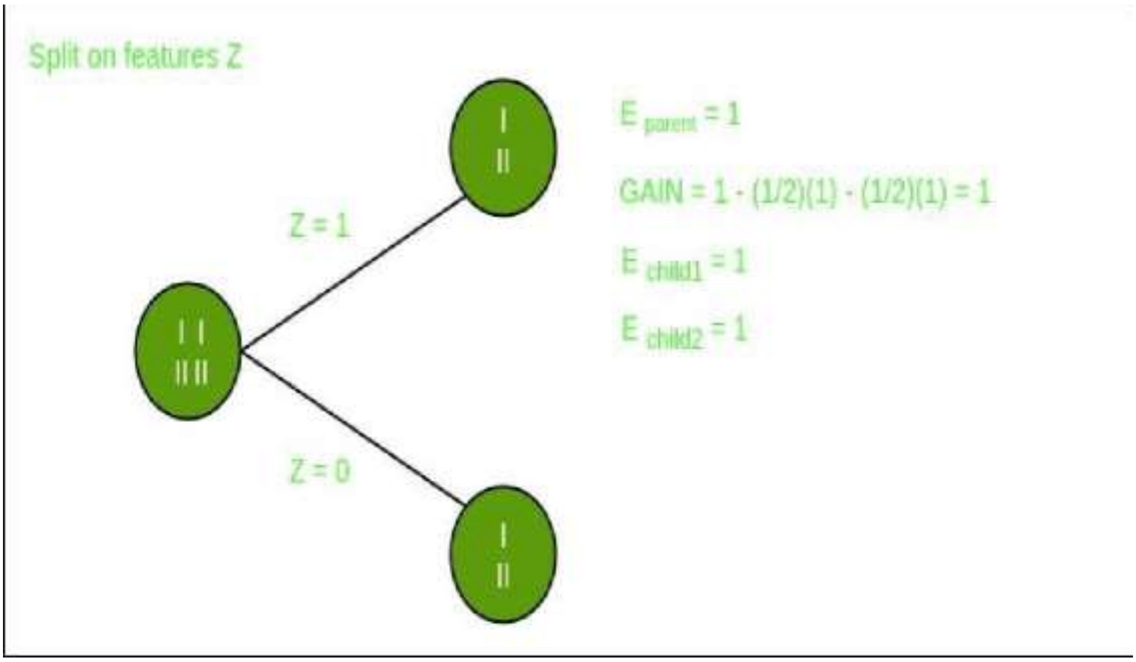
$$E_{child1} = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3)$$
$$= 0.5284 + 0.39$$
$$= 0.9184$$

$E_{child2} = 0$

Split on feature X



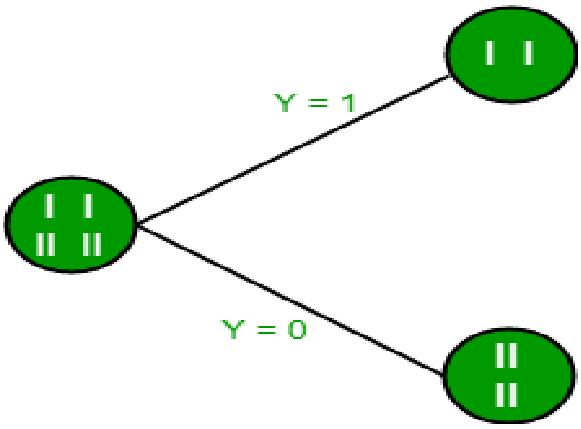
Split on feature Y



Split on feature Z

From the above images we can see that the information gain is maximum when we make a split on feature Y. So, for the root node best suited feature is feature Y. Now we can see that while splitting the dataset by feature Y, the child contains pure subset of the target variable. So we don't need to further split the dataset.

The final tree for the above dataset would be look like this:



2. Gini Index

- Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified.
- It means an attribute with lower Gini index should be preferred.
- Sklearn supports "Gini" criteria for Gini Index and by default, it takes "gini" value.

The Formula for the calculation of the of the Gini Index is given below,
Example:
Lets consider the dataset in the image below and draw a decision tree using gini index.

Index	A	B	C	D	E
1	4.8	3.4	1.9	0.2	Positive
2	5	3	1.6	1.2	Positive
3	5	3.4	1.6	0.2	Positive
4	5.2	3.5	1.5	0.2	positive
5	5.2	3.4	1.4	0.2	positive
6	4.7	3.2	1.6	0.2	positive

Index	A	B	C	D	E
7	4.8	3.1	1.6	0.2	positive
8	5.4	3.4	1.5	0.4	positive
9	7	3.2	4.7	1.4	negative
10	6.4	3.2	4.7	1.5	negative
11	6.9	3.1	4.9	1.5	negative
12	5.5	2.3	4	1.3	negative
13	6.5	2.8	4.6	1.5	negative
14	5.7	2.8	4.5	1.3	negative
15	6.3	3.3	4.7	1.6	negative
16	4.9	2.4	3.3	1	negative

In the dataset above there are 5 attributes from which attribute E is the predicting feature which contains 2(Positive & Negative) classes. We have an equal proportion for both the classes.In Gini Index, we have to choose some random values to categorize each attribute. These values for this dataset are:

A	B	C	D
>= 5	>= 3.0	>= 4.2	>= 1.4
< 5	< 3.0	< 4.2	< 1.4

Calculating Gini Index for Var A:
Value >= 5: 12

Attribute A >= 5 & class = positive:

Attribute A >= 5 & class = negative:

Gini(5, 7) = 1 –
Value < 5: 4

Attribute A < 5 & class = positive:
Attribute A < 5 & class = negative:
Gini(3, 1) = 1 –
By adding weight and sum each of the gini indices:

Calculating Gini Index for Var B:
Value >= 3: 12

Attribute B >= 3 & class = positive:

Attribute B >= 5 & class = negative:

Gini(5, 7) = 1 –
Value < 3: 4

By adding weight and sum each of the gini indices:

Using the same approach we can calculate the Gini index for C and D attributes.

	Positive	Negative
For A|>= 5.0	5	7
|<5	3	1

Ginin Index of A = 0.45825

	Positive	Negative
For B|>= 3.0	8	4
|< 3.0	0	4

Gini Index of B= 0.3345

	Positive	Negative
For C|>= 4.2	0	6
|< 4.2	8	2

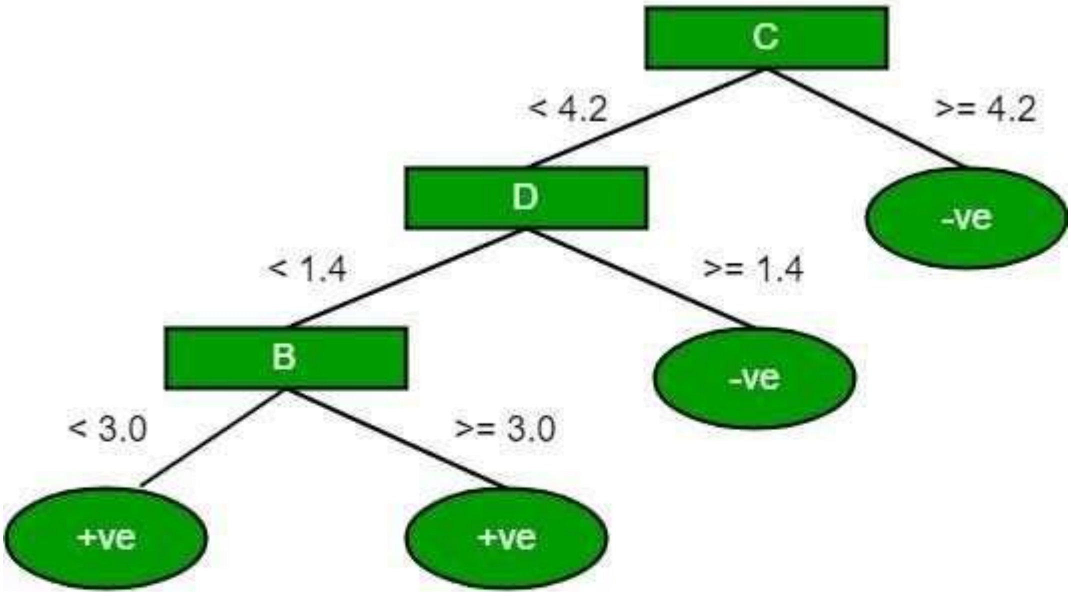
Gini Index of C= 0.2

	Positive	Negative
For D|>= 1.4	0	5

|< 1.4   8   3

Gini Index of D= 0.273

Decision tree for above dataset



The most notable types of decision tree algorithms are:-

1. Iterative Dichotomiser 3 (ID3): This algorithm uses Information Gain to decide which attribute is to be used classify the current subset of the data. For each level of the tree, information gain is calculated for the remaining data recursively.

2.C4.5: This algorithm is the successor of the ID3 algorithm. This algorithm uses either Information gain or Gain ratio to decide upon the classifying attribute. It is a direct improvement from the ID3 algorithm as it can handle both continuous and missing attribute values.

3.Classification and Regression Tree(CART): It is a dynamic learning algorithm which can produce a regression tree as well as a classification tree depending upon the dependent variable.

3.4 RULE BASED CLASSIFICATION:

Rule-based classifiers are just another type of classifier which makes the class decision depending by using various “if..else” rules. These rules are easily interpretable and thus these classifiers are generally used to generate descriptive models. The condition used with “if” is called the antecedent and the predicted class of each rule is called the consequent.

Properties of rule-based classifiers:

- Coverage: The percentage of records which satisfy the antecedent conditions of a particular rule.
- The rules generated by the rule-based classifiers are generally not mutually exclusive, i.e. many rules can cover the same record.
- The rules generated by the rule-based classifiers may not be exhaustive, i.e. there may be some records which are not covered by any of the rules.
- The decision boundaries created by them is linear, but these can be much more complex than the decision tree because the many rules are triggered for the same record.

An obvious question, which comes into the mind after knowing that the rules are not mutually exclusive is that how would the class be decided in case different rules with different consequent cover the record.

There are two solutions to the above problem:

- Either rules can be ordered, i.e. the class corresponding to the highest priority rule triggered is taken as the final class.
- Otherwise, we can assign votes for each class depending on some their weights, i.e. the rules remain unordered.

Example:
Below is the dataset to classify mushrooms as edible or poisonous:

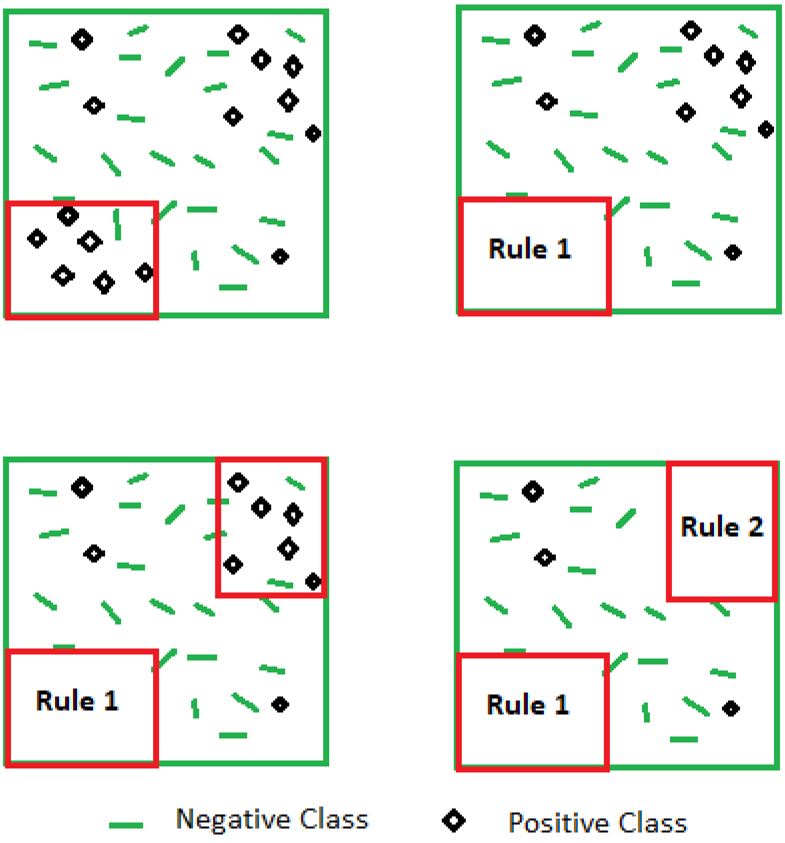
Class	Cap Shape	Cap Surface	Bruises	Odour	Stalk Shape	Population	Habitat
Edible	flat	Scaly	yes	anise	tapering	scattered	grasses
Poisonous	convex	Scaly	yes	pungent	enlarging	several	grasses
Edible	convex	Smooth	yes	almond	enlarging	numerous	grasses
Edible	convex	Scaly	yes	almond	tapering	scattered	meadows
Edible	flat	fibrous	yes	anise	enlarging	several	woods

Class	Cap Shape	Cap Surface	Bruises	Odour	Stalk Shape	Population	Habitat
Edible	flat	fibrous	no	none	enlargening	several	urban
Poisonous	conical	Scaly	yes	pungent	enlargening	scattered	urban
Edible	flat	Smooth	yes	anise	enlargening	numerous	meadows
Poisonous	convex	Smooth	yes	pungent	enlargening	several	urban

3.4.1 Rules:

Odour = pungent and habitat = urban -> Class = poisonous
Bruises = yes -> Class = edible : This rules covers both negative and positive records.

The given rules are not mutually exclusive.
How to generate a rule:



Sequential Rule Generation.

Rules can be generated either using general-to-specific approach or specific-to-general approach. In the general-to-specific approach, start with a rule with no antecedent and keep on adding conditions to it till we see major improvements in our evaluation metrics. While for the other we keep on removing the conditions from a rule covering a very specific case. The evaluation metric can be accuracy, information gain, likelihood ratio etc.

3.4.2 Algorithm for generating the model incrementally:

The algorithm given below generates a model with unordered rules and ordered classes, i.e. we can decide which class to give priority while generating the rules.

```
A <-Set of attributes
T <-Set of training records
Y <-Set of classes
Y’ <-Ordered Y according to relevance
R <-Set of rules generated, initially to an empty list
for each class y in Y’
while the majority of class y records are not covered
```

```
generate a new rule for class y, using methods given above
Add this rule to R
Remove the records covered by this rule from T
end while
end for
Add rule {}->y' where y' is the default class
```

3.4.3 Classifying a record:

The classification algorithm described below assumes that the rules are unordered and the classes are weighted.

```
R <-Set of rules generated using training Set
T <-Test Record
W <-class name to Weight mapping, predefined, given as input
F <-class name to Vote mapping, generated for each test record, to be calculated
for each rule r in R
check if r covers T
if so then add W of predicted_class to F of predicted_class end for
Output the class with the highest calculated vote in F
```

Note: The rule set can be also created indirectly by pruning(simplifying) other already generated models like a decision tree.

5. PRUNING THE RULE SET

1. Rule Generation

Once a decision tree has been constructed, it is a simple matter to convert it into an equivalent set of rules.

Converting a decision tree to rules before pruning has three main advantages:

1. Converting to rules allows distinguishing among the different contexts in which a decision node is used.
   - Since each distinct path through the decision tree node produces a distinct rule, the pruning decision regarding that attribute test can be made differently for each path.
   - In contrast, if the tree itself were pruned, the only two choices would be:

        1. Remove the decision node completely, or
        2. Retain it in its original form.
3. Converting to rules removes the distinction between attribute tests that occur near the root of the tree and those that occur near the leaves.
    - We thus avoid messy bookkeeping issues such as how to reorganize the tree if the root node is pruned while retaining part of the subtree below this test.
4. Converting to rules improves readability.
    - Rules are often easier for people to understand.

To generate rules, trace each path in the decision tree, from root node to leaf node, recording the test outcomes as antecedents and the leaf-node classification as the consequent.

3.5.2 Rule Simplification Overview

Once a rule set has been devised:

1. Eliminate unecessary rule antecedents to simplify the rules.
    - Construct contingency tables for each rule consisting of more than one antecedent.
        - Rules with only one antecedent cannot be further simplified, so we only consider those with two or more.
    - To simplify a rule, eliminate antecedents that have no effect on the conclusion reached by the rule.
    - A conclusion's independence from an antecendent is verified using a test for independence, which is
        - a chi-square test if the expected cell frequencies are greater than 10.
        - Yates' Correction for Continuity when the expected frequencies are between 5 and 10.
        - Fisher's Exact Test for expected frequencies less than 5.
1. Eliminate unecessary rules to simplify the rule set.
    - Once individual rules have been simplified by eliminating redundant antecedents, simplify the entire set by eliminating unnecessary rules.
    - Attempt to replace those rules that share the most common consequent by a default rule that is triggered when no other rule is triggered.
    - In the event of a tie, use some heuristic tie breaker to choose a default rule.

3.5.3 Contingency Tables

The following is a contingency table, a tabular representation of a rule.

	C1	C2	Marginal Sums
R1	x11	x12	R1T = x11 + x12
R2	x21	x22	R2T = x21 + x22
Marginal Sums	CT1 = x11 + x21	CT2 = x12 + x22	T = x11 + x12 + x21 + x22

- R1 and R2 represent the Boolean states of an antecedent for the conclusions C1 and C2(C2 is the negation of C1).

- x11, x12, x21 and x22 represent the frequencies of each antecedent-consequent pair.
- R1T, R2T, CT1, CT2 are the marginal sums of the rows and columns, respectively.

The marginal sums and T, the total frequency of the table, are used to calculate expected cell values in step 3 of the test for independence.

4. Test for Independence

Given a contingency table of dimensions r by c (rows x columns):

1. Calculate and fix the sizes of the marginal sums.

2. Calculate the total frequency, T, using the marginal sums.

3. Calculate the expected frequencies for each cell.

The general formula for obtaining the expected frequency of any cell xij, $1 \leq i \leq r$, $1 \leq j \leq c$ in a contingency table is given by:

$$e_{ij} = \frac{R_{iT} \cdot C_{Tj}}{T}$$

where RiT and CTj are the row total for ith row and the column total for jth column.

4. Select the test to be used to calculate χ^2 based on the highest expected frequency, m:

if

$m > 10$	Chi-Square Test
$5 \leq m \leq 10$	Yates' Correction for Continuity
$m < 5$	Fisher's Exact Test

5. Calculate χ^2 using the chosen test.

6. Calculate the degrees of freedom. df

= (r - 1)(c - 1)

7. Use a chi-square table with χ^2 and df to determine if the conclusions are independent from the antecedent at the selected level of significance, α.

   - Assume α = 0.05 unless otherwise stated.
   - If $\chi^2 > \chi^2_\alpha$

        - Reject the null hypothesis of independence and accept the alternate hypothesis of dependence.
            - We keep the antecedents because the conclusions are dependent upon them.
- If $\chi^2 \leq \chi^2_{\alpha}$
        - Accept the null hypothesis of independence.
            - We discard the antecedents because the conclusions are independent from them.

3.5.5 Chi-Square Formulae

- **Chi-Square Test**

$$x^2 = \sum_i \sum_j \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

- **Yates' Correction for Continuity**

$$x^2(corrected) = \sum_i \sum_j \frac{(|o_{ij} - e_{ij}| - 0.5)^2}{e_{ij}}$$

- **Fisher's Exact Test**

Decision Lists

A decision list is a set of if-then statements.

It is searched sequentially for an appropriate if-then statement to be used as a rule.

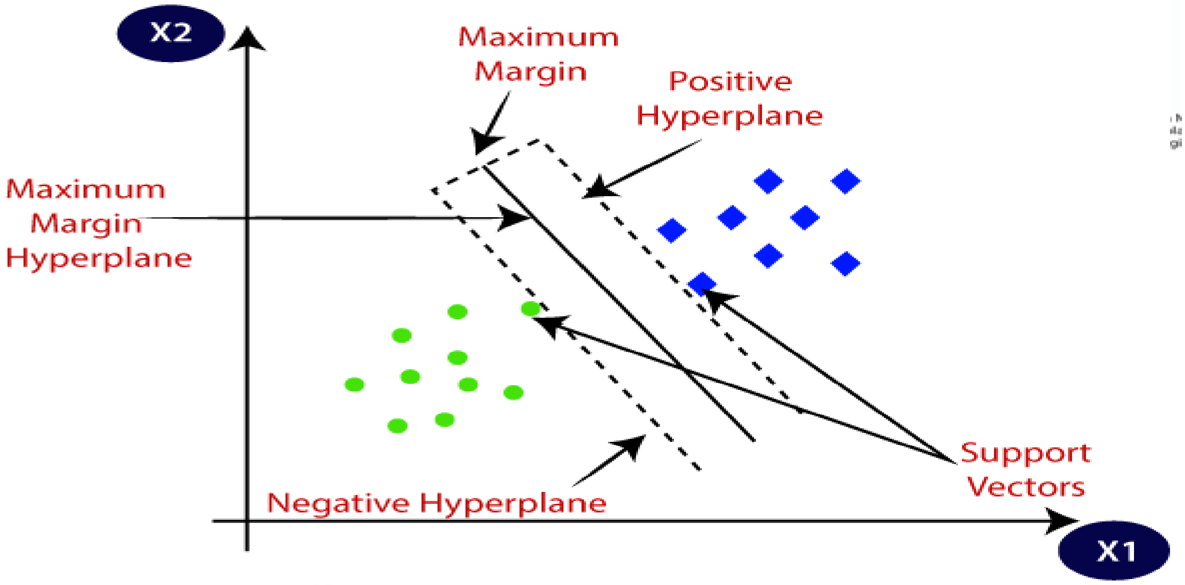
3.6 SUPPORT VECTOR MACHINES

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

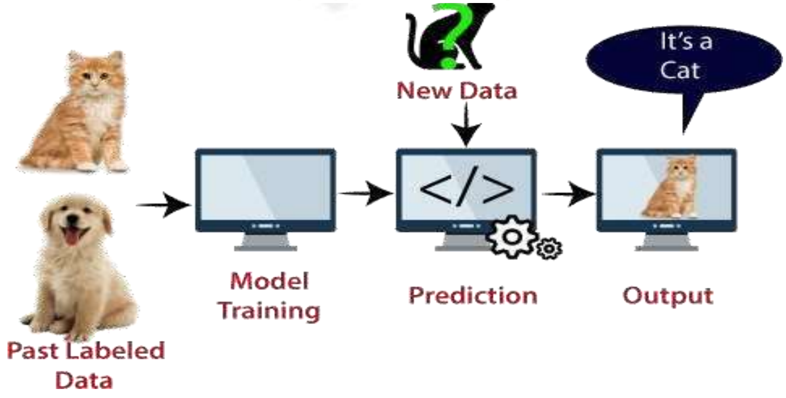
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

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.

Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Example: SVM can be understood with the example that we have used in the KNN classifier.
Suppose we see a strange cat that also has some features of dogs, so if we want a model that
can accurately identify whether it is a cat or dog, so such a model can be created by using the
SVM algorithm. We will first train our model with lots of images of cats and dogs sot hat it
learn about different features of cats and dogs, and then we test it with this strang can
So as support vector creates a decision boundary between these two data (cat an ecreature
.
basis of the support vectors, it will classify it as a cat. Consider the below diagram: ddog)
and
choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the



SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

1. Types of SVM

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

2. Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors:

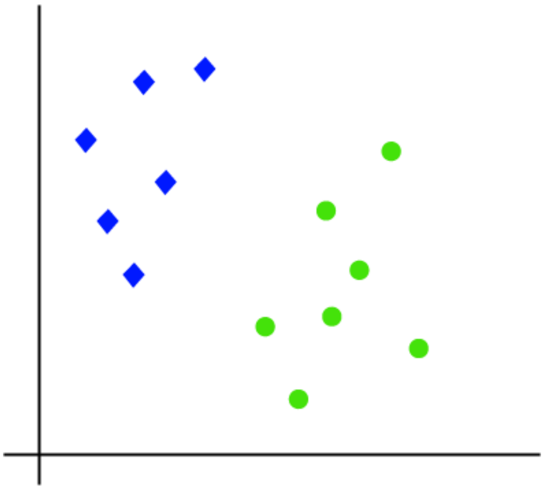
The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

How does SVM works?

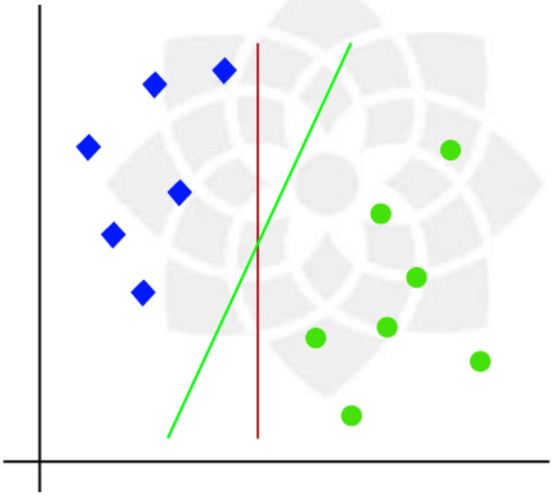
Linear SVM:

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue.

Consider the below image:

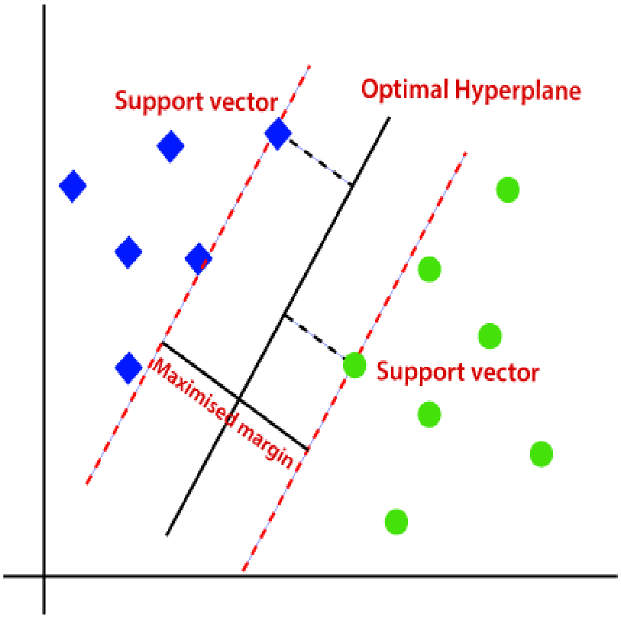


So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



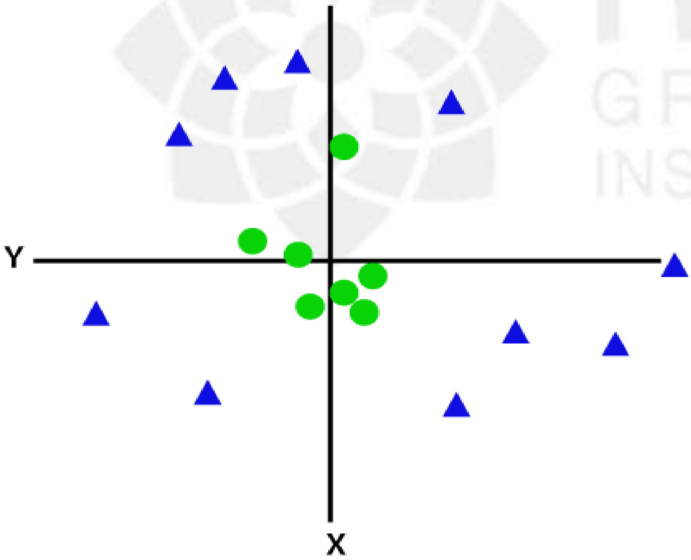
Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors.

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



Non-Linear SVM:

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:

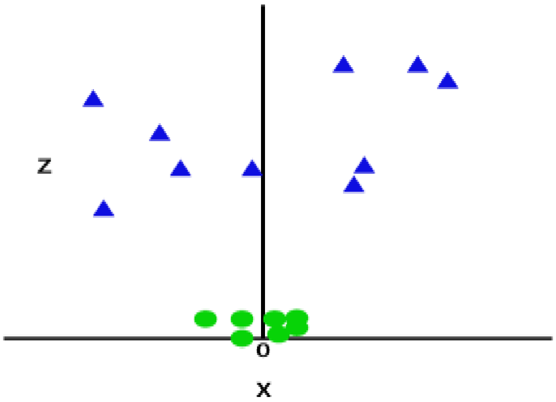


So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z.

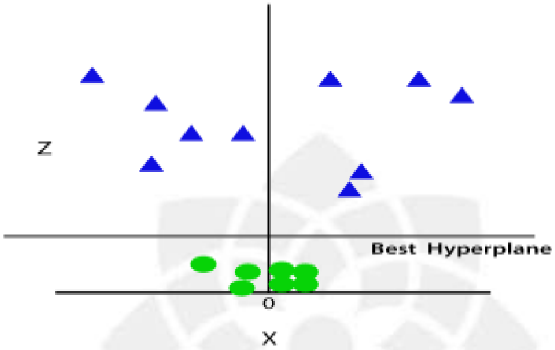
It can be calculated as:

z=x2 +y2

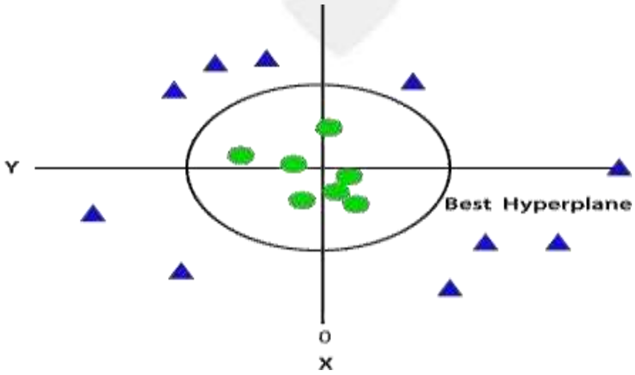
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below image:



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:

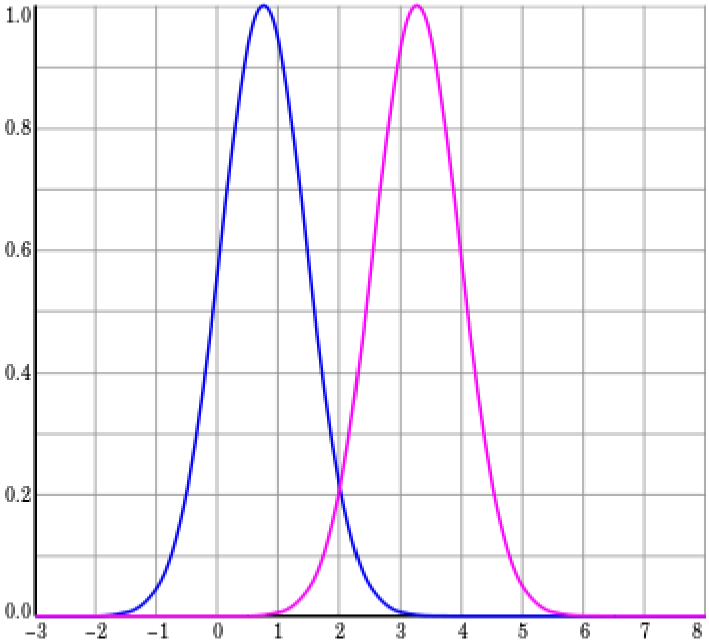


Hence we get a circumference of radius 1 in case of non-linear data

7. RADIAL BASIS FUNCTIONS:

1. What is a Radial Basis Function?

A Radial basis function is a function whose value depends only on the distance from the origin. In effect, the function must contain only real values. Alternative forms of radial basis functions are defined as the distance from another point denoted C, called a center.



Source

3.7.2 How does a Radial Basis Function work?

A Radial basis function works by defining itself by the distance from its origin or center. This is done by incorporating the absolute value of the function. Absolute values are defined as the value without its associated sign (positive or negative). For example, the absolute value of -4, is 4. Accordingly, the radial basis function is a function in which its values are defined as:

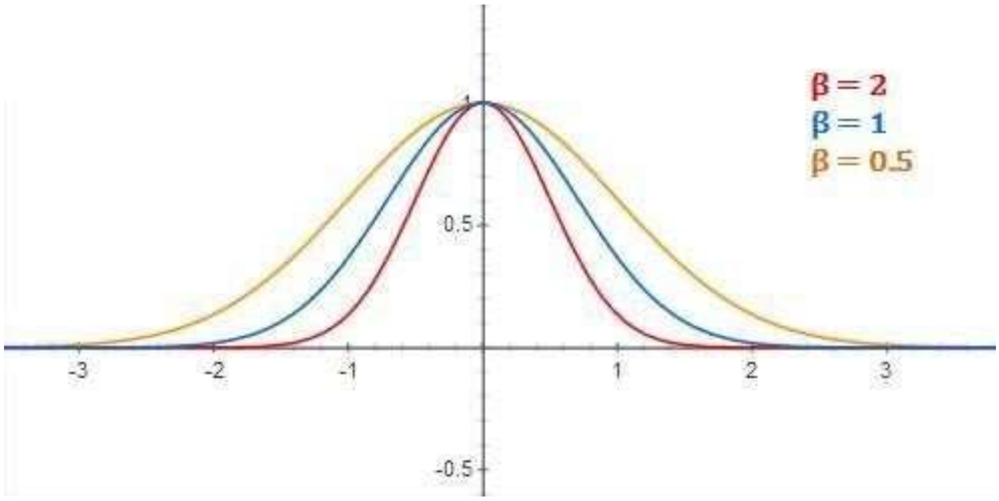
$$\varphi(\mathbf{x}) = \varphi(||\mathbf{x}||)$$

The Gaussian variation of the Radial Basis Function, often applied in Radial Basis Function Networks, is a popular alternative. The formula for a Gaussian with a one-dimensional input

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

is:

The Gaussian function can be plotted out with various values for Beta:



Source

3.7.3 Applications of the Radial Basis Function

Radial basis functions make up the core of the Radial Basis Function Network, or RBFN. This particular type of neural network is useful in cases where data may need to be classified in a non-linear way. RBFNs work by incorporating the Radial basis function as a neuron and using it as a way of comparing input data to training data. An input vector is processed by multiple Radial basis function neurons, with varying weights, and the sum total of the neurons produce a value of similarity. If input vectors match the training data, they will have a high similarity value. Alternatively, if they do not match the training data, they will not be assigned a high similarity value. Comparing similarity values with different classifications of data allows for non-linear classification.

3.8 NAIVE BAYES CLASSIFIER

What is a classifier?

A classifier is a machine learning model that is used to discriminate different objects based on certain features.

3.8.1 Principle of Naive Bayes Classifier:

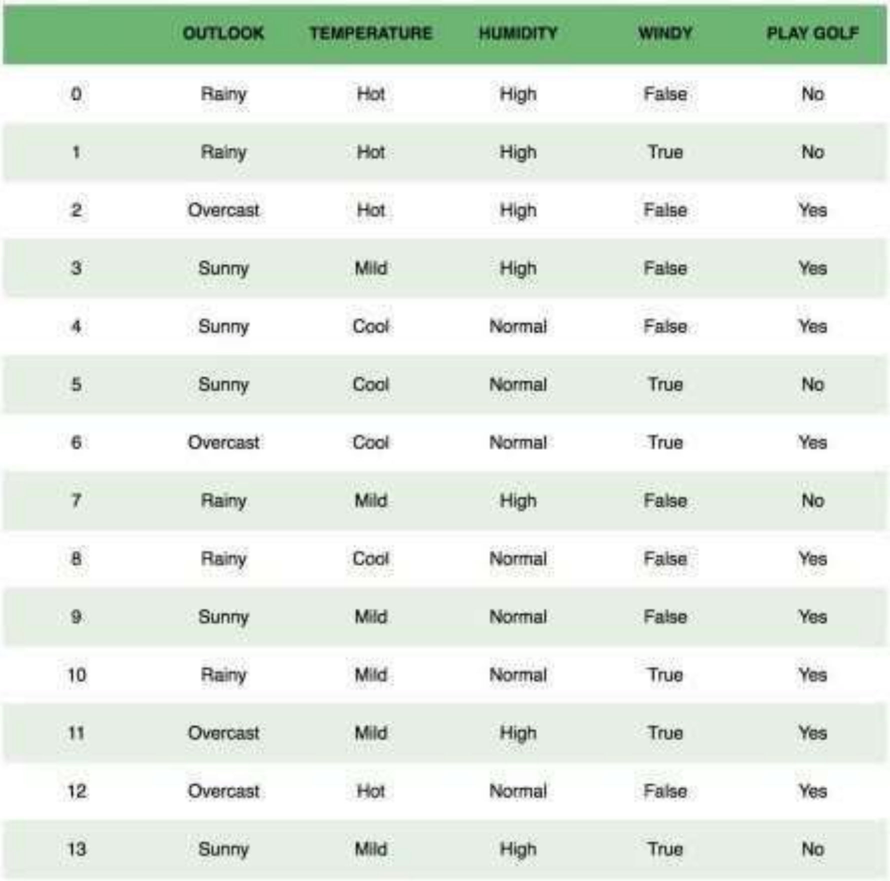
A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive. **Example:** Let us take an example to get some better ntuition. Consider the problem of playing golf.

The dataset is represented as below.



	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

We classify whether the day is suitable for playing golf, given the features of the day. The columns represent these features and the rows represent individual entries. If we take the first row of the dataset, we can observe that is not suitable for playing golf if the outlook is rainy, temperature is hot, humidity is high and it is not windy. We make two assumptions here, one as stated above we consider that these predictors are independent. That is, if the temperature is hot, it does not necessarily mean that the humidity is high. Another assumption made here is that all the predictors have an equal effect on the outcome. That is, the day being windy does not have more importance in deciding to play golf or not.

According to this example, Bayes theorem can be rewritten as:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

The variable **y** is the class variable(play golf), which represents if it is suitable to play golf or not given the conditions. Variable **X** represent the parameters/features.

X is given as,

$$X = (x_1, x_2, x_3, ..., x_n)$$

Here x_1,x_2….x_n represent the features, i.e they can be mapped to outlook, temperature, humidity and windy. By substituting for **X** and expanding using the chain rule we get,

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Now, you can obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change, it remain static.

Therefore, the denominator can be removed and a proportionality can be introduced.

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

In our case, the class variable(**y**) has only two outcomes, yes or no. There could be cases where the classification could be multivariate. Therefore, we need to find the class **y** with maximum probability.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Using the above function, we can obtain the class, given the predictors.

3.8.2 Types of Naive Bayes Classifier:

Multinomial Naive Bayes:

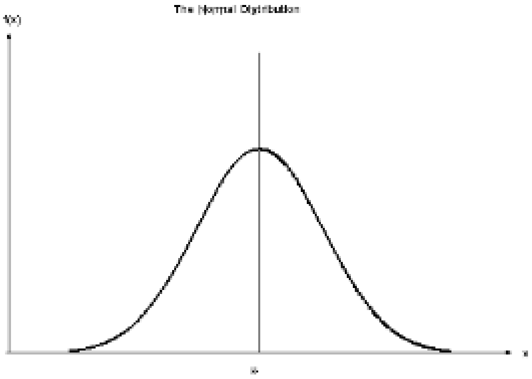
This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

Bernoulli Naive Bayes:

This is similar to the multinomial naive bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

Gaussian Naive Bayes:

When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.



Gaussian Distribution(Normal Distribution)

Since the way the values are present in the dataset changes, the formula for conditional probability changes to,

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

3.8.3 Conclusion: Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier.

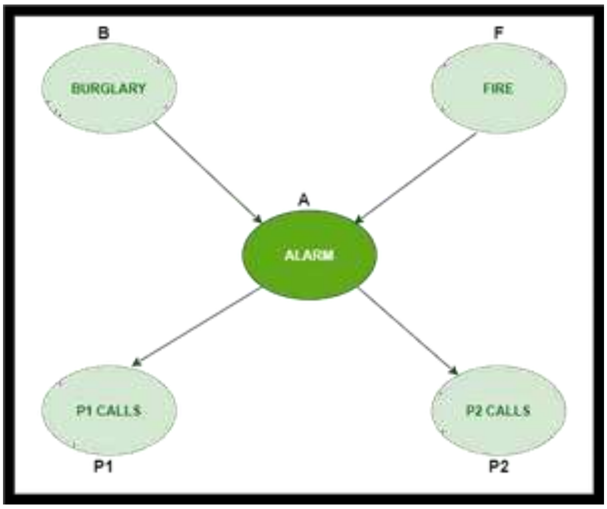
3.9 BAYESIAN BELIEF NETWORK:

Bayesian Belief Network is a graphical representation of different probabilistic relationships among random variables in a particular set. It is a classifier with no dependency on attributes i.e it is condition independent.

Due to its feature of joint probability, the probability in Bayesian Belief Network is derived, based on a condition — P(attribute/parent) i.e probability of an attribute, true over parent attribute.

(Note: A classifier assigns data in a collection to desired categories.)

- Consider this example:



- In the above figure, we have an alarm 'A' – a node, say installed in a house of a person 'gfg', which rings upon two probabilities i.e burglary 'B' and fire 'F', which are – parent nodes of the alarm node. The alarm is the parent node of two probabilities P1 calls 'P1' & P2 calls 'P2' person nodes.

- Upon the instance of burglary and fire, 'P1' and 'P2' call person 'gfg', respectively. But, there are few drawbacks in this case, as sometimes 'P1' may forget to call the person 'gfg', even after hearing the alarm, as he has a tendency to forget things, quick. Similarly, 'P2', sometimes fails to call the person 'gfg', as he is only able to hear the alarm, from a certain distance.

Example Problem:
Q)Find the probability that 'P1' is true (P1 has called 'gfg'), 'P2' is true (P2 has called 'gfg') when the alarm 'A' rang, but no burglary 'B' and fire 'F' has occurred.

=> P (P1, P2, A, ~B, ~F) [where- P1, P2 & A are 'true' events and '~B' & '~F' are 'false' events]

[Note: The values mentioned below are neither calculated nor computed. They have observed values]

Burglary 'B' –

- P (B=T) = 0.001 ('B' is true i.e burglary has occurred)
- P (B=F) = 0.999 ('B' is false i.e burglary has not occurred) Fire

'F' –

- P (F=T) = 0.002 ('F' is true i.e fire has occurred)
- P (F=F) = 0.998 ('F' is false i.e fire has not occurred)

Alarm 'A' –

B	F	P (A=T)	P (A=F)
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

- The alarm ‘A’ node can be ‘true’ or ‘false’ (i.e may have rung or may not have rung). It has two parent nodes burglary ‘B’ and fire ‘F’ which can be ‘true’ or ‘false’ (i.e may have occurred or may not have occurred) depending upon different conditions.

Person ‘P1’ –

A	P (P1=T)	P (P1=F)
T	0.95	0.05
F	0.05	0.95

- The person ‘P1’ node can be ‘true’ or ‘false’ (i.e may have called the person ‘gfg’ or not) . It has a parent node, the alarm ‘A’, which can be ‘true’ or ‘false’ (i.e may have rung or may not have rung ,upon burglary ‘B’ or fire ‘F’).

Person ‘P2’ –

A	P (P2=T)	P (P2=F)
T	0.80	0.20
F	0.01	0.99

- The person ‘P2’ node can be ‘true’ or false’ (i.e may have called the person ‘gfg’ or not). It has a parent node, the alarm ‘A’, which can be ‘true’ or ‘false’ (i.e may have rung or may not have rung, upon burglary ‘B’ or fire ‘F’).

Solution: Considering the observed probabilistic scan –

With respect to the question — P (P1, P2, A, ~B, ~F) , we need to get the probability of ‘P1’. We find it with regard to its parent node – alarm ‘A’. To get the probability of ‘P2’, we find it with regard to its parent node — alarm ‘A’.
We find the probability of alarm ‘A’ node with regard to ‘~B’ & ‘~F’ since burglary ‘B’ and fire ‘F’ are parent nodes of alarm ‘A’.

From the observed probabilistic scan, we can deduce –

P (P1, P2, A, ~B, ~F)
= P (P1/A) * P (P2/A) * P (A/~B~F) * P (~B) * P (~F)
= 0.95 * 0.80 * 0.001 * 0.999 * 0.998
= 0.00075

10. ASSIGNMENT 1- UNIT 3

Category I

Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select the appropriate data set for your experiment and draw graphs. (K4,CO3)

Category II

The rules are easily interpretable and thus these classifiers are generally used to generate descriptive models. The condition used with "if" is called the antecedent and the predicted class of each rule is called the consequent. Properties of rule-based classifiers:Apply Rule based classification to the dataset to classify Weather prediction. (K4,CO3)

The rules are easily interpretable and thus these classifiers are generally used to generate descriptive models. The condition used with "if" is called the antecedent and the predicted class of each rule is called the consequent. Properties of rule-based classifiers:Apply Rule based classification to the dataset to classify Health care. (K4,CO3)

Category III

Construct a Decision tree using attributes selection measure Information Gain . (K3,CO3)

Construct a Decision tree using attributes selection measure Gini Ratio. (K3,CO3)

Category IV

Elaborate in detail about Linear Regression and hypothesis function (K2,CO3)

Explain the working of Bayesian belief network (K2,CO3)

Category V

Ellucidate in detail about Radial basis function? (K2,CO3)

Elaborate in detail about the naive bayes classifier. (K2,CO3)

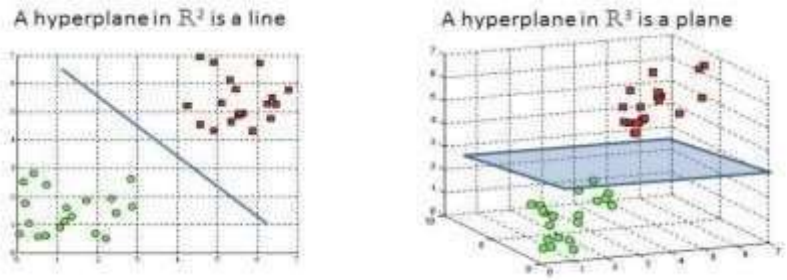
11. PART A Q & A (WITH K LEVEL AND CO) UNIT 3

1. What are the two types of problems solved by Supervised Learning?

- Classification: It uses algorithms to assign the test data into specific categories. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest.
- Regression: It is used to understand the relationship between dependent and independent variables. Linear regression, logistical regression, and polynomial regression are popular regression algorithms.

2. What is Support Vector Machine?

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.



Support vector machines focus only on the points that are the most difficult to tell apart, whereas other classifiers pay attention to all of the points.

The intuition behind the support vector machine approach is that if a classifier is good at the most challenging comparisons (the points in B and A that are closest to each other), then the classifier will be even better at the easy comparisons (comparing points in B and A that are far away from each other).

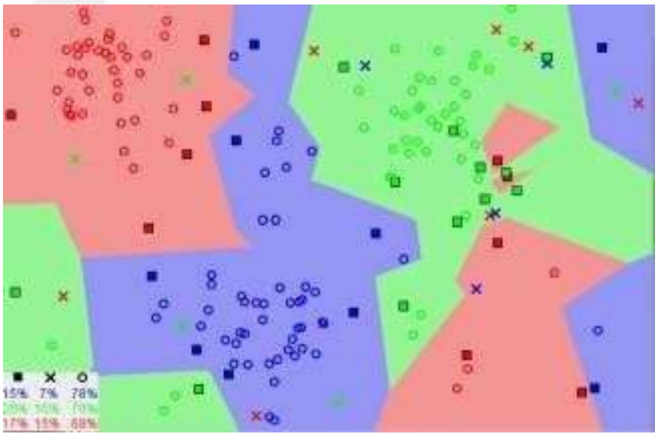
3. Give a real life example of Supervised Learning and Unsupervised Learning

- Supervised learning examples:
    - You get a bunch of photos with information about what is on them and then you train a model to recognize new photos.

  - You have a bunch of molecules and information about which are drugs and you train a model to answer whether a new molecule is also a drug.
  - Based on past information about spams, filtering out a new incoming email into Inbox (normal) or Junk folder (Spam)
  - Cortana or any speech automated system in your mobile phone trains your voice and then starts working based on this training.
  - Train your handwriting to OCR system and once trained, it will be able to convert your hand-writing images into text (till some accuracy obviously)

4. What is k-Nearest Neighbors algorithm?

- k-Nearest Neighbors is a supervised machine learning algorithm that can be used to solve both classification and regression problems.
- It assumes that similar things are closer to each other in certain feature spaces, in other words, similar things are in close proximity.



- The image above shows how similar points are closer to each other. KNN hinges on this assumption being true enough for the algorithm to be useful.
- There are many different ways of calculating the distance between the points, however, the straight line distance (Euclidean distance) is a popular and familiar choice.

5. What is Bias in Machine Learning?

In supervised machine learning an algorithm learns a model from training data.

The goal of any supervised machine learning algorithm is to best estimate the mapping function (f) for the output variable (Y) given the input data (X). The mapping function is often called the target function because it is the function that a given supervised machine learning algorithm aims to approximate.

Bias are the simplifying assumptions made by a model to make the target function easier to learn. Generally, linear algorithms have a high bias making them fast to learn and easier to understand but generally less flexible.

- Examples of low-bias machine learning algorithms include: Decision Trees, k-Nearest Neighbors and Support Vector Machines.

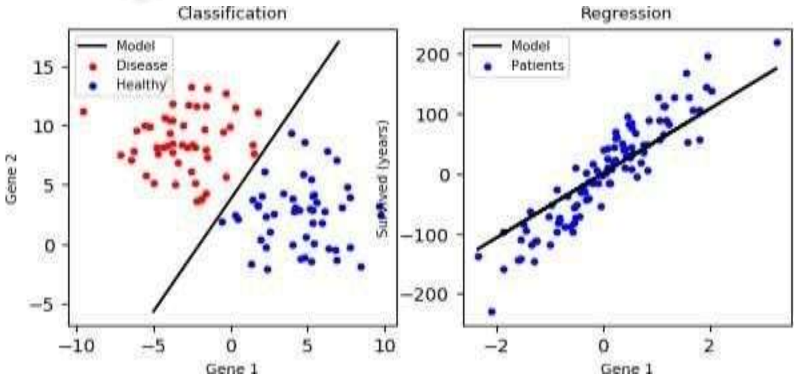
- Examples of high-bias machine learning algorithms include: Linear Regression, Linear Discriminant Analysis and Logistic Regression.

6. What is the difference between a Regression problem and a Classification problem?

Answer

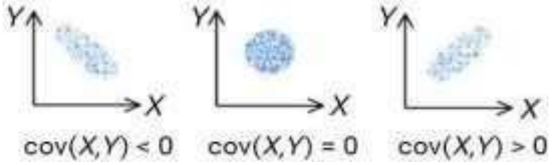
- Classification is the problem of identifying which set of categories an observation belongs to.
- Regression is a set of statistical processes for estimating the relationships between a dependent variable and an independent variable.

- Classification is used to predict the values of a categorical variable, so the output is generally in the form of integers, or binary (0 or 1).
- Regression is used to predict a continuous variable, so the output is also a floating-point number (0.1, 0.74, 0.69, etc.).



7. What's the difference between Covariance and Correlation?

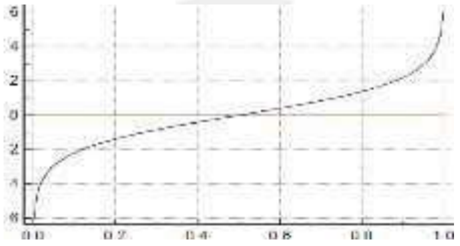
- Covariance measures whether a variation in one variable results in a variation in another variable, and deals with the linear relationship of only 2 variables in the dataset. Its value can take range from -∞ to +∞.
- Simply speaking Covariance indicates the direction of the linear relationship between variables.



- Correlation measures how strongly two or more variables are related to each other. Its values are between -1 to 1. Correlation measures both the strength and direction of the linear relationship between two variables. Correlation is a function of the covariance.

8. How do you use a supervised Logistic Regression for Classification?

- Logistic regression is a statistical model that utilizes logit function to model classification problems. It is a regression analysis to conduct when the dependent variable is binary. The logit function is shown below:



Looking at the logit function, the next question that comes to mind is how to fit that graph/equation. The fitting of the logistic regression is done using the maximum likelihood function.

- In a supervised logistic regression, features are mapped onto the output. The output is usually a categorical value (which means that it is mapped with one-hot vectors or binary numbers).
- Since the logit function always outputs a value between 0 and 1, it gives the probability of the outcome.

9. What are some challenges faced when using a Supervised Regression Model?
Some challenges faced when using a supervised regression model are:

- Nonlinearities: Real-world data points are more complex and do not follow a linear relationship. Sometimes a non-linear model is better at fitting the dataset. So, it is a challenge to find the perfect equation for the dataset.
- Multicollinearity: Multicollinearity is a phenomenon where one predictor variable in a multiple regression model can be linearly predicted from the otherswith a substantial degree of accuracy. If there is a problem of multicollinearitythen even the slightest change in the independent variable causes the output to change erratically. Thus, the accuracy of the model is affected and it undermines the quality of the whole model.
- Outliers: Outliers can change and make huge impact on the machine learning model. This happens because the regression model tries to fit the outliers into the model as well.

12.PART B Q s (WITH K LEVEL AND CO) UNIT 3

1. Elaborate in detail about Linear Regression and hypothesis function?

2. Explain in detail about Logistic regression and its types with use-cases?

3. Elucidate the Decision tree algorithm and explain the construction of Decision tree?

4. Explain in detail about the two attributes and types of decision tree algorithm?

5. Explain about construction of decision tree and pruning made to the branches?

6. Explain the conditions how the pruning rules sets are considered?

7. Explain the working of Support vector machine?

8. Elaborate the function of SVM and its types?

9. What is the major condition to be followed for selecting optimal hyper plane? Explain the crux of optimal hyper-plane?

10. Ellucidate in detail about Radial basis function?

11. Elaborate in detail about the naïve bayes classifier?

12. Explain the working of Bayesian belief network?

.

13. Supportive online Certification courses

1. NPTEL COURSE LINK- https://onlinecourses.nptel.ac.in/noc22_cs97/preview
2. COURSERA- https://www.coursera.org/learn/machine-learning

14. REAL TIME APPLICATIONS IN DAY-TO-DAY LIFE AND TO INDUSTRY

- Customer churn decrease
- Customer Lifetime Worth Assessment
- Personalizing reviews for goods
- Human resource distribution
- Sales Forecasting
- Supply and demand analysis
- To identify fraud
- Predicting repair of equipment

15. ASSESSMENT SCHEDULE

Tentative schedule for the Assessment During 2023-2024 EVEN semester

S.NO	Name of the Assessment	Start Date	Portions
1	IAT 1	10-02-2024	UNIT 1 & 2
2	IAT 2	01-04-2024	UNIT 3 & 4
3	Model	20-04-2024	ALL 5 UNITS

16. PRESCRIBED TEXT BOOKS & REFERENCE BOOKS

TEXT BOOKS:

1. Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, Machine Learning, Pearson, 2019. (Unit 1 – chap 1,2,3/ Unit 2 – Chap 4 / Unit 4 – 9 / Unit 5 – Chap 10, 11)
2. Ethem Alpaydin, Introduction to Machine Learning, Adaptive Computation and Machine Learning Series, Third Edition, MIT Press, 2014. (Unit 2 – Chap 6 / Unit 4 – chap 8.2.3 / Unit 5 – Chap 18)

REFERENCES:

1. Anuradha Srinivasaraghavan, Vincy Joseph, Machine Learning, First Edition, Wiley, 2019. (Unit 3 – Chap 7,8,9,10,11 / Unit 4 – 13, 11.4, 11.5,12)
2. Peter Harrington, "Machine Learning in Action", Manning Publications, 2012.
3. Stephen Marsland, "Machine Learning – An Algorithmic Perspective", Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.
4. Tom M Mitchell, Machine Learning, First Edition, McGraw Hill Education, 2013.
5. Christoph Molnar, "Interpretable Machine Learning - A Guide for Making Black Box Models Explainable", Creative Commons License, 2020.

17. MINI PROJECT SUGGESTION

1. Movie Recommendations with Movielens Dataset

Almost everyone today uses technology to stream movies and television shows. While figuring out what to stream next can be daunting, recommendations are often made based on a viewer's history and preferences. This is done through machine learning and can be a fun and easy project for beginners to take on. New programmers can practice by coding in either Python or R languages and with data from the Movielens Dataset. Generated by more than 6,000 users, Movielens currently includes more than 1 million movie ratings of 3,900 films.

2. Develop A Sentiment Analyzer:

This is one of the interesting machine learning project ideas. Although most of us use social media platforms to convey our personal feelings and opinions for the world to see, one of the biggest challenges lies in understanding the 'sentiments' behind social media posts.

Social media is thriving with tons of user-generated content. By creating an ML system that could analyze the sentiment behind texts, or a post, it would become so much easier for organizations to understand consumer behaviour. This, in turn, would allow them to improve their customer service, thereby providing the scope for optimal consumer satisfaction.

You can try to mine the data from Twitter or Reddit to get started off with your sentiment analyzing machine learning project. This might be one of those rare cases of deep learning projects which can help you in other aspects as well.

3. Turning Handwritten Documents into Digitized Versions.

The problem with this project is to classify handwritten digits. The goal is to take an image of a handwritten digit and determine what that digit is. The digits range from one (1) through nine (9).

Apply Nearest Neighbor (NN) techniques to solve the problem.

4. Spam email classification using SupportVector Machine

You will use a SVM to classify emails into spam or non-spam categories. And report the classification accuracy for various SVM parameters and kernel functions.

Data Set Description: An email is represented by various features like frequency of occurrences of certain keywords, length of capitalized words etc.

A data set containing about 4601 instances are available in this link (data folder): https://archive.ics.uci.edu/ml/datasets/Spambase.

You have to randomly pick 70% of the data set as training data and the remaining as test data.

Assignment Tasks: In this assignment you can use any SVM package to classify the above data set.

You should use one of the following languages: C/C++/Java/Python. You have to study performance of the SVM algorithms.

You have to submit a report in pdf format.

The report should contain the following sections:

1. Methodology: Details of the SVM package used.
2. Experimental Results:

i. You have to use each of the following three kernel functions (a) Linear, (b) Quadratic, (c) RBF.

ii. For each of the kernels, you have to report training and test set classification accuracy for the best value of generalization constant C.

The best C value is the one which provides the best test set accuracy that you have found out by trial of different values of C. Report accuracies in the form of a comparison table, along with the values of C

Thank you

Disclaimer:

This document is confidential and intended solely for the educational purpose of RMK Group of Educational Institutions. If you have received this document through email in error, please notify the system manager. This document contains proprietary information and is intended only to the respective group / learning community as intended. If you are not the addressee you should not disseminate, distribute or copy through e-mail. Please notify the sender immediately by e-mail if you have received this document by mistake and delete this document from your system. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited.