Marathwada Mitra Mandal's

COLLEGE OF ENGINEERING, PUNE

Accredited with 'A++' Grade by NAAC

Department of Computer Engineering

Lab Manual

410246 : LP-III: Machine Learning

Prepared by,

Prof. Mayuri Shelke

Prof. Kalyani Ghuge

BE COMP (2019 Pattern)

Academic Year 2023-24 Sem I

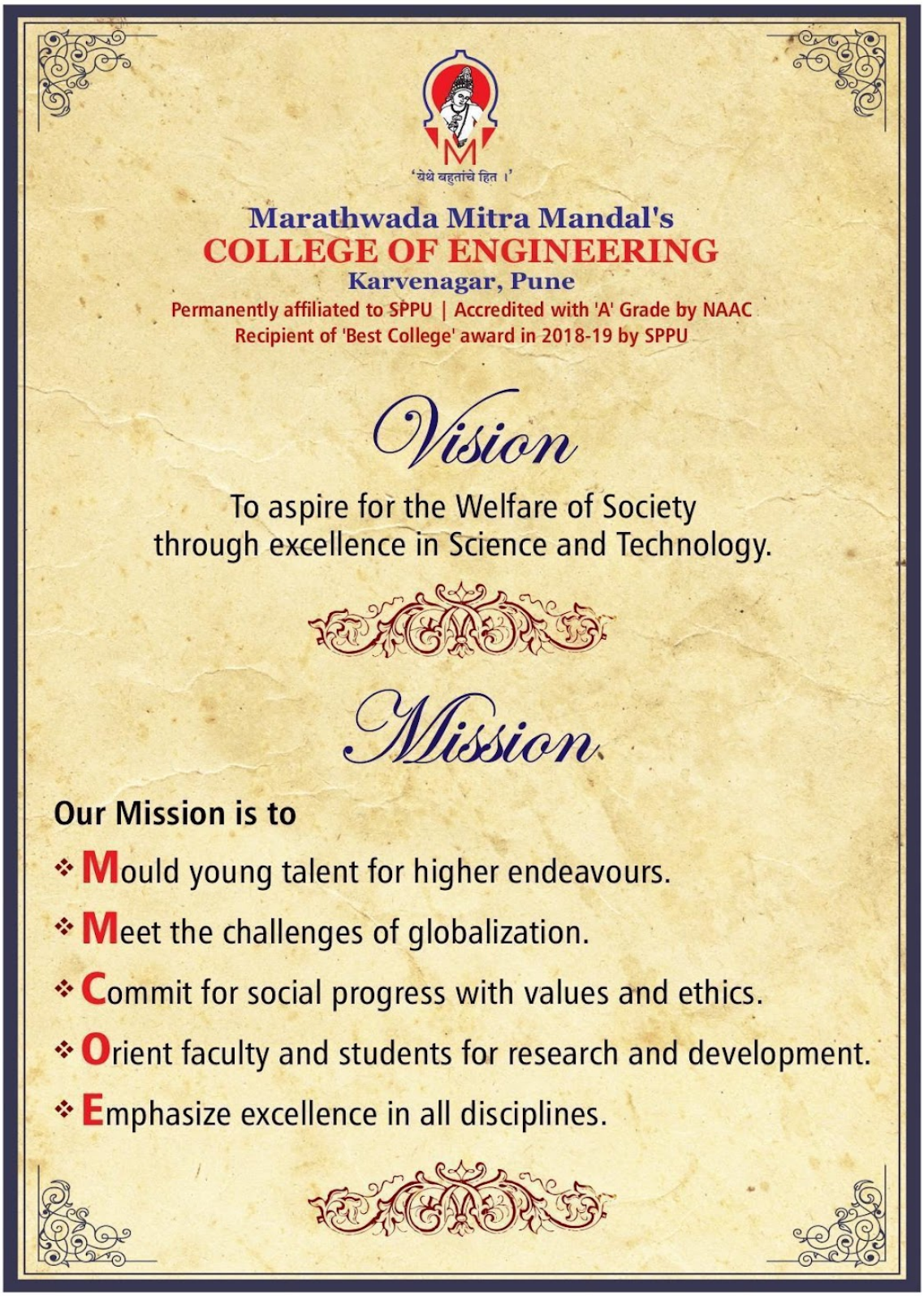
Preface

Machine learning is a subset of artificial intelligence in the field of computer science that often uses statistical techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome

Machine learning can be described as a form of statistical analysis, often even utilizing well-known and familiar techniques, that has a bit of a different focus than traditional analytical practice in applied disciplines. We started by developing automatic functions to perform classic machine learning tasks such as classification, regression, or dimensionality reduction. Then, we developed a user-friendly neural network framework. Along the way, we used these tools to develop applications such as image identification, topic identification, or text entity recognition.

Here are the characteristics of Machine Learning

- The ability to perform automated data visualization
- Understanding of the dataset and performing per-processing on the datasets
- Apply different supervised and unsupervised machine learning algorithms on different use cases
- Analyze the performances of algorithms based on the model generated
- Machine learning (ML) is coming into its own, with a growing recognition that ML can play a key role in a wide range of critical applications


'येथे बहुतांचे हित ।'
Marathwada Mitra Mandal's
COLLEGE OF ENGINEERING
Karvenagar, Pune
Permanently affiliated to SPPU | Accredited with 'A' Grade by NAAC
Recipient of 'Best College' award in 2018-19 by SPPU
Vision
To aspire for the Welfare of Society
through excellence in Science and Technology.
Mission
Our Mission is to
Mould young talent for higher endeavours.
Meet the challenges of globalization.
Commit for social progress with values and ethics.
Orient faculty and students for research and development.
Emphasize excellence in all disciplines.

'येथे बहुतांचे हित ।'

Marathwada Mitramandal's
COLLEGE OF ENGINEERING

Karvenagar, Pune - 411052

Department of Computer Engineering

Vision

To contribute to welfare of society by empowering students with latest skills, tools and technologies in the field of Computer Engineering through excellence in education and research

Mission

- To provide excellent academic environment for continuous improvement in the domain knowledge of Computer Engineering to solve real world problems
- To impart value-based education to students, with innovative and research skills to make them responsible engineering professionals for societal upliftment
- To strengthen links with industries through partnerships and collaborative developmental works

Program Educational Objectives (PEOs)

- To develop globally competent graduates with strong fundamental knowledge and analytical capability in latest technological trends
- To prepare the graduates as ethical and committed professionals with a sense of societal and environmental responsibilities
- To inculcate research attitude in multidisciplinary domains with experiential learning and developing entrepreneurship skills
- To groom graduates by incorporating investigative approach among them to effectively deal with global challenges

Program Specific Outcomes (PSOs)

A graduate of the Computer Engineering Program will be able to

- Analyze the problems and design solutions in the areas of Artificial Intelligence & High Performance Computing
- Develop advanced digital solutions using standard software engineering practices

Program Outcomes (POs)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2. Problem Analysis: Identify, formulates, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design / development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research – based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7.Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Rubrics for Lab Assessment (CAS)

Dimension	Scale				
	1	2	3	4	5
Regularity and punctuality	Did not Perform, submitted in time	Performed and submitted later than scheduled date	Performed on schedule; submitted two weeks late	Performed on schedule; submitted one week late	Performed and submitted as per schedule
Understanding and preparation for Objective	Neither shows any understanding of the objective nor can relate it to theory.	States the objective very vaguely	Can only state the objective but shows poor understanding	Understands objective but cannot place it in context of a theory	Understands objective and can relate it to an appropriate theory topic
Participation in performance and conduction of experiment	Does not participate in experiment	Performs the experiment only with the help from supervisor /others and is confused and untidy.	Performs the experiment with some supervisory help; but forgets some crucial reading and is confused and untidy.	Performs experiment on own without supervisory help; records all readings properly but untidy.	Performs experiment on his/her own without supervisory help; records all readings properly. Keeps the setup clean and tidy.

Post experiment skills	Cannot follow the procedure and do any work	Follows procedure half-heartedly	Follows right procedure; but cannot analyze data and interpret it	Follows right procedure and can analyze data and interpret it	Follows right procedure; can analyze data and interpret it with justification

Syllabus

Savitribai Phule Pune University Fourth Year of Computer Engineering (2019 Course) 410246: Laboratory Practice III		
Teaching Scheme: **Practical: 04** **Hours/Week**	**Credit** **02**	**Examination Scheme:** **Term work: 50 Marks** **Practical: 50 Marks**

Companion Course: Design and Analysis of Algorithms (410241), Machine Learning(410242), Blockchain Technology(410243)

Course Objectives:

- Learn effect of data preprocessing on the performance of machine learning algorithms
- Develop in depth understanding for implementation of the regression models.
- Implement and evaluate supervised and unsupervised machine learning algorithms.
- Analyze performance of an algorithm.
- Learn how to implement algorithms that follow algorithm design strategies namely divide and conquer, greedy, dynamic programming, backtracking, branch and bound.
- Understand and explore the working of Blockchain technology and its applications.

Course Outcomes:

After completion of the course, students will be able to
CO1: Apply preprocessing techniques on datasets.
CO2: Implement and evaluate linear regression and random forest regression models.
CO3: Apply and evaluate classification and clustering techniques.
CO4: Analyze performance of an algorithm.
CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound.
CO6: Interpret the basic concepts in Blockchain technology and its applications

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as a softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to a journal must be avoided. Use of DVD containing student programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

7.	**Mini Project** Write a program to implement matrix multiplication. Also implement multithreaded matrix multiplication with either one thread per row or one thread per cell. Analyze and compare their performance. OR Implement merge sort and multithreaded merge sort. Compare time required by both the algorithms. Also analyze the performance of each algorithm for the best case and the worst case. OR Implement the Naive string matching algorithm and Rabin-Karp algorithm for string matching. Observe difference in working of both the algorithms for the same input.
Group B: Machine Learning	
Any 4 assignments and 1 Mini project are mandatory.	
1.	Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks: 1. Pre-process the dataset. 2. Identify outliers. 3. Check the correlation. 4. Implement linear regression and random forest regression models. 5. Evaluate the models and compare their respective scores like R2, RMSE, etc. Dataset link: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset
2.	Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance. Dataset link: The emails.csv dataset on the Kaggle https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv
3.	Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months. Dataset Description: The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc. Link to the Kaggle project: https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling Perform following steps: 1. Read the dataset. 2. Distinguish the feature and target set and divide the data set into training and test sets. 3. Normalize the train and test data. 4. Initialize and build the model. Identify the points of improvement and implement the same. 5. Print the accuracy score and confusion matrix (5 points).
4.	Implement Gradient Descent Algorithm to find the local minima of a function. For example, find the local minima of the function $y=(x+3)^2$ starting from the point x=2.
5.	Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset. Dataset link : https://www.kaggle.com/datasets/abdallamahgoub/diabetes

7.	**Mini Project** Write a program to implement matrix multiplication. Also implement multithreaded matrix multiplication with either one thread per row or one thread per cell. Analyze and compare their performance. OR Implement merge sort and multithreaded merge sort. Compare time required by both the algorithms. Also analyze the performance of each algorithm for the best case and the worst case. OR Implement the Naive string matching algorithm and Rabin-Karp algorithm for string matching. Observe difference in working of both the algorithms for the same input.
Group B: Machine Learning	
Any 4 assignments and 1 Mini project are mandatory.	
1.	Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks: 1. Pre-process the dataset. 2. Identify outliers. 3. Check the correlation. 4. Implement linear regression and random forest regression models. 5. Evaluate the models and compare their respective scores like R2, RMSE, etc. Dataset link: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset
2.	Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance. Dataset link: The emails.csv dataset on the Kaggle https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv
3.	Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months. Dataset Description: The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc. Link to the Kaggle project: https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling Perform following steps: 1. Read the dataset. 2. Distinguish the feature and target set and divide the data set into training and test sets. 3. Normalize the train and test data. 4. Initialize and build the model. Identify the points of improvement and implement the same. 5. Print the accuracy score and confusion matrix (5 points).
4.	Implement Gradient Descent Algorithm to find the local minima of a function. For example, find the local minima of the function $y=(x+3)^2$ starting from the point x=2.
5.	Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset. Dataset link : https://www.kaggle.com/datasets/abdallamahgoub/diabetes

6.	Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset. Determine the number of clusters using the elbow method. Dataset link : https://www.kaggle.com/datasets/kyanyoga/sample-sales-data
7.	**Mini Project** Use the following dataset to analyze ups and downs in the market and predict future stock price returns based on Indian Market data from 2000 to 2020. Dataset Link: https://www.kaggle.com/datasets/sagara9595/stock-data OR Build a machine learning model that predicts the type of people who survived the Titanic shipwreck using passenger data (i.e. name, age, gender, socio-economic class, etc.). Dataset Link: https://www.kaggle.com/competitions/titanic/data
Group C: Blockchain Technology	
Any 4 assignments and a Mini project are mandatory.	
1.	Installation of Metamask and study spending Ether per transaction.
2.	Create your own wallet using Metamask for crypto transactions.
3.	Write a smart contract on a test network, for Bank account of a customer for following operations: • Deposit money • Withdraw Money • Show balance
4.	Write a program in solidity to create Student data. Use the following constructs: • Structures • Arrays • Fallback Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas values.
5.	Write a survey report on types of Blockchains and its real time use cases.
6.	**Mini Project:** Create a dApp (de-centralized app) for e-voting system.

CO PO and PSO Mapping

A. Course Outcome

Course Outcome	Statement
	At the end of the course, student will be able to
410246.1	Apply preprocessing techniques on datasets.
410246.2	Implement and evaluate linear regression and random forest regression models.
410246.3	Apply and evaluate classification and clustering techniques.

Course Outcome	Program outcomes												PSO	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
410246.1	2	2	1	2	-	2	-	2	-	1	1	1	2	1
410246.2	2	2	1	2	1	2	-	2	2	1	2	2	2	2
410246.3	2	2	1	2	1	2	-	2	2	1	2	2	2	2

INDEX

Sr. No.	Group	Title of Assignment	CO	PO
1	B	Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks: 1. Pre-process the dataset. 2. Identify outliers. 3. Check the correlation. 4. Implement linear regression and random forest regression models. 5. Evaluate the models and compare their respective scores like R2, RMSE, etc. Dataset link: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset	1,2	1, 2, 3, 4, 6, 8, 10, 11,12
2		Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance. Dataset link: The emails.csv dataset on the Kaggle https://www.kaggle.com/datasets/balaka18/email-spam-classificatio n-dataset-csv	1, 3	1, 2, 3, 4, 5, 6, 8, 9,10, 11,12
3		Implement Gradient Descent Algorithm to find the local minima of a function. For example, find the local minima of the function y=(x+3)2 starting from the point x=2.	1, 3	1, 2, 3, 4, 5, 6, 8, 9, 11, 12

4		Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset. Dataset link :https://www.kaggle.com/datasets/abdallamahgoub/diabetes	**1, 3**	1, 2, 3, 4, 5, 6, 8, 9, 11, 12
5		Implement K-Means clustering/ hierarchical clustering on sales_data_sampe.csv dataset. Determine the number of clusters using the elbow method. Dataset link : https://www.kaggle.com/datasets/kyanyoga/sample-sales-data	**1, 3**	1, 2, 3, 4, 5, 6, 8, 9, 10, 12
6		**Mini Project** Use the following dataset to analyze ups and downs in the market and predict future stock price returns based on Indian Market data from 2000 to 2020. Dataset Link: https://www.kaggle.com/datasets/sagara9595/stock-data **OR** Build a machine learning model that predicts the type of people who survived the Titanic shipwreck using passenger data (i.e. name, age, gender, socio-economic class, etc.). Dataset Link: https://www.kaggle.com/competitions/titanic/data **OR** Develop a application for signature identification by creating your own dataset of your college student	**1, 2,3**	1,2, 3,7, 10,11

7		**Content Beyond Syllabus** Assignment on Decision Tree Classifier: A dataset collected in a cosmetics shop showing details of customers and whether or not they responded to a special offer to buy a new lip-stick is shown in table below. Use this dataset to build a decision tree, with Buys as the target variable, to help in buying lip-sticks in the future.Find the root node of the decision tree. According to the decision tree you have made from the previous training data set, what is the decision for the test data: [Age < 21, Income = Low,Gender = Female, Marital Status = Married]?	**1,3**	1, 2, 3, 4, 5, 6, 8, 9,12

Use software testing concepts to implement the problem statements. Testing done using Selenium tools/Junit/pyunit.

Software Required:

1. 64 bit open source operating system
2. Programming tools recommended: - Python, Anaconda Navigator,VS code, Google Colab

Write-ups must include:

- **Assignment No.**
- **Title**
- **Problem Statement**
- **Prerequisites**
- **Course Objectives**
- **Course Outcomes**
- **Theory(in brief)**
- **Conclusion**
- **FAQs:**
- **Output: Printout of program with output.**

ASSIGNMENT NO: 1

TITLE: Linear regression and Random Forest regression.

PROBLEM STATEMENT: Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks:

1. Pre-process the dataset.
2. Identify outliers.
3. Check the correlation.
4. Implement linear regression and random forest regression models.
5. Evaluate the models and compare their respective scores like R2, RMSE, etc.

Dataset link: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset

PREREQUISITES:

Basic of Python, Data Mining Algorithm

COURSE OBJECTIVE:

To Implement linear regression and random forest regression models.

COURSE OUTCOME:

Implement linear regression and random forest regression models.

THEORY:

Linear Regression

Regression analysis is used in stats to find trends in data. For example, you might guess that there's a connection between how much you eat and how much you weigh; regression analysis can help you quantify that.

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.

Prerequisites for Regression

Simple linear regression is appropriate when the following conditions are satisfied.

- The dependent variable *Y* has a linear relationship to the independent variable X. To check this, make sure that the XY scatter plot is linear and that the residual plot shows a random pattern. For each value of X, the probability distribution of Y has the same standard deviation σ.
- When this condition is satisfied, the variability of the residuals will be relatively constant across all values of X, which is easily checked in a residual plot.
- For any given value of X,
    - The Y values are independent, as indicated by a random pattern on the residual plot.
    - The Y values are roughly normally distributed (i.e., symmetric and unimodal). A little skewness is ok if the sample size is large. A histogram or a dotplot will show the shape of the distribution.

Random Forest Regressor :

Random Forest Regressor is an ensemble learning algorithm used for regression tasks. It combines multiple decision trees to make accurate predictions. By aggregating the outputs of individual trees, it minimizes overfitting and enhances prediction reliability. Random Forest Regressor is versatile, handles both numerical and categorical data, and is widely employed in various applications, such as finance, healthcare, and marketing, where precise predictions are essential. Its feature importance analysis aids in understanding the impact of different variables on the model's predictions, making it a valuable tool for data-driven decision-making.

The steps followed in the Random Forest algorithm are :
1. Pick at random k data points from the training set.
2. Build a decision tree associated with these k data points.
3. Choose the number N of trees you want to build and repeat steps 1 and 2. 4. For a new data point, make each one of your N-tree trees predict the value of y for the data point in question and assign the new data point to the average across all of the predicted y values.

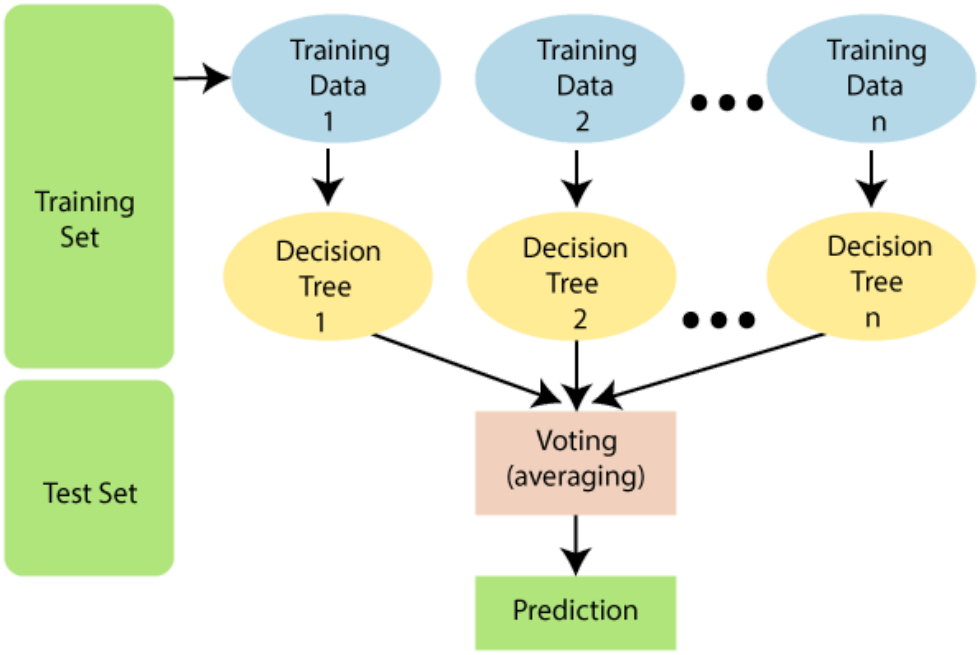


Fig. Working of the Random Forest algorithm

Algorithm Steps:

1. Analyze and visualize the data to understand its distribution and relationships between variables.
2. **Data Preprocessing:**

   a. Handle Missing Data: Deal with missing values in the dataset, either by imputation or removal.

   b. Outlier Treatment: Address outliers, which can significantly impact the linear regression model.

   c. Data Scaling: Standardize or normalize the data if the variables are on different scales.

3. **Model Selection :**

   Choose Linear Regression model and random forest regressor model

4. **Split Data:**

   a. Divide the dataset into a training set and a testing set. The training set is used to build the model, and the testing set is used to evaluate its performance.

5. **Model Training:**

   Linear Regression

   a. Fit the linear regression model to the training data by finding the best-fit line that minimizes the sum of squared errors (least squares method).

   b. Calculate the coefficients (slope and intercept) of the linear equation.

   Random forest Regressor

   a. Choose the Random Forest Regressor as your predictive model.

6. **Model Evaluation:**

   a. Assess the model's performance using various metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2).

   b. Visualization: Plot the regression line and residuals to check for model adequacy.

7. **Prediction:**

   a. Use the trained model to make predictions on the testing dataset or new data.

CONCLUSION: We Studied, Linear Regression is a simple effective method for modeling relationships between variables. Random Forest Regressor is a powerful ensemble technique that excels in handling intricate patterns and is less prone to overfitting, making it suitable for diverse predictive tasks.

FAQ'S:

Q. 1 What is Linear Regression?

Ans. Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

Q.2 Explain Random Forest Regression?

Ans. Random forest is a supervised learning algorithm that uses an ensemble learning method for classification and regression. Random forest is a bagging technique and not a boosting technique.

Q. 3 What is the necessity of splitting the dataset?

Ans The main idea of splitting the dataset into a validation set is to prevent our model from overfitting i.e. The model becomes really good at classifying the samples in the training set but cannot generalize and make accurate classifications on the data it has not seen before.

Q. 4 What is the significance of RMSE?

Ans. Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data. Formally it is defined as f:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors) Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are.

Q. 5 What is r2score ?

Ans. The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. It is pronounced as R squared and is also known as the coefficient of determination. It works by measuring the amount of variance in the predictions explained by the dataset.

OUTPUT

ASSIGNMENT NO: 2

TITLE: K-Nearest Neighbors and Support Vector Machine

PROBLEM STATEMENT: Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance. Dataset link: The emails.csv dataset on the Kaggle https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv

PREREQUISITES:

Understand distance metrics like Euclidean distance for measuring similarity between data points.

COURSE OBJECTIVE:

To implement the concept of K-Nearest Neighbors and Support Vector Machine

COURSE OUTCOME:

Implementing the concept of K-Nearest Neighbors and Support Vector Machine

THEORY:

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K-NN algorithm. The K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. The KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

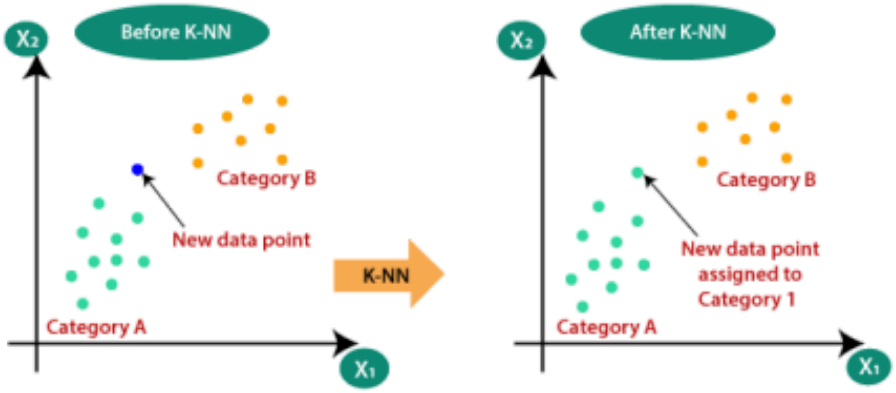


Fig. Class of a particular dataset

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

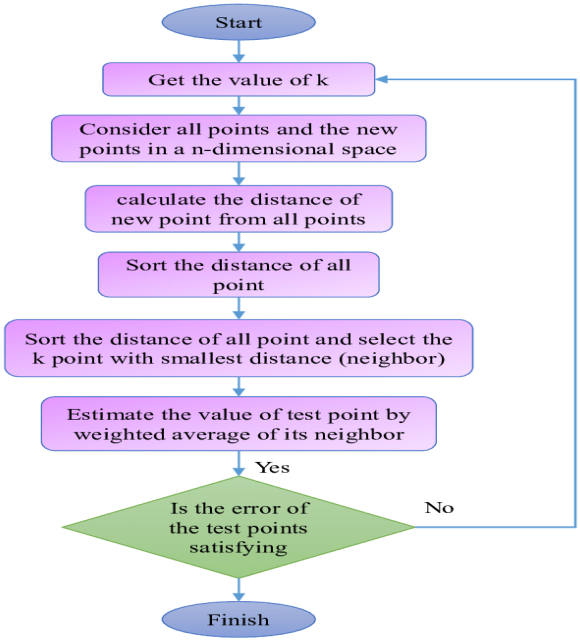


Fig. Flowchart for the k-nearest neighbor modeling

Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which

is used for Classification as well as Regression problems. 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 support vectors, and hence the 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:

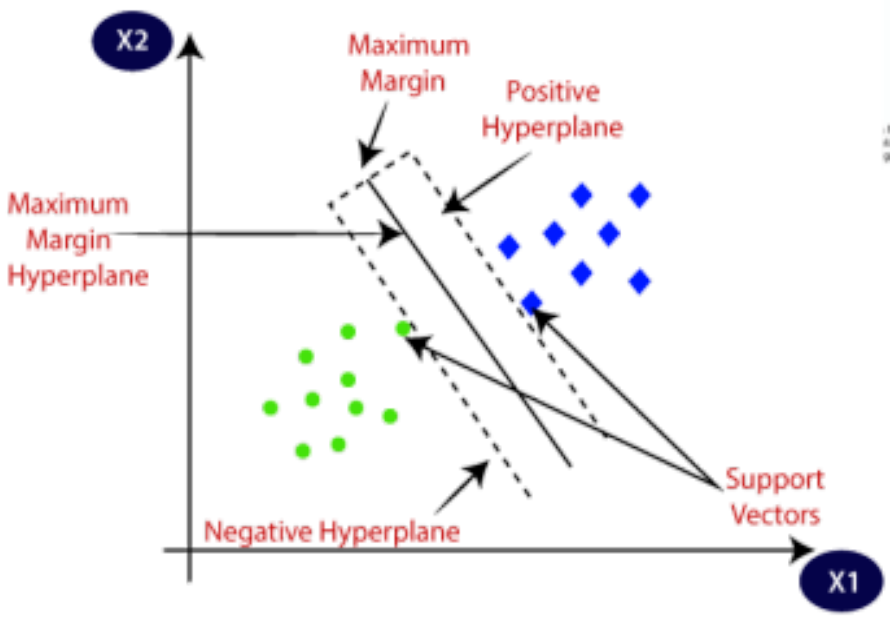


Fig. Different categories are classified using a decision boundary or hyperplane

Dataset Description :

This is a csv file containing related information of 5172 randomly picked email files and their respective labels for spam or not-spam classification. The csv file contains 5172 rows, each row for each email. There are 3002 columns. The first column indicates Email name. The name has been set with numbers and not recipients' names to protect privacy. The last column has the labels for prediction : 1 for spam, 0 for not spam. The remaining 3000 columns are the 3000 most common words in all the emails, after excluding the non-alphabetical characters/words. For each row, the count of each word(column) in that email(row) is stored in the respective cells. Thus, information regarding all 5172 emails are stored in a compact dataframe rather than as separate text files.

ALGORITHM:

1. Importing libraries,
2. Importing datasets
3. Finding Missing Data
4. Encoding Categorical Data
5. Splitting dataset into training and test set
6. Applying classifiers K-Nearest Neighbors and Support Vector Machine for classification of spam and not-spam
7. Evaluating performance of classifiers in terms of metrics such as RMSE, r2 score etc.

CONCLUSION: We have understand how to apply KNN and SVM algorithm on the dataset

FAQ'S:

1. What are K-Nearest Neighbors?

Ans: K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data

2. What is a Support Vector Machine ?

Ans . Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well, it's best suited for classification. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

Q3 . State applications of Support Vector Machine

Ans: ● Inverse Geosounding Problem.
● Seismic Liquefaction Potential.
● Protein Fold and Remote Homology Detection.
● Data Classification using SVM.
● Facial Expression Classification.
● Texture Classification using SVM.
● Text Classification.
● Speech Recognition.

Q4. State applications of KNN.

Ans. ● Text mining.
● Agriculture.
● Finance.
● Medical.
● Facial recognition.
● Recommendation systems (Amazon, Hulu, Netflix, etc)

Q5. Compare SVM and KNN

Ans. SVM and kNN exemplify several important trade-offs in machine learning (ML). SVM is less computationally demanding than kNN and is easier to interpret but can identify only a limited set of patterns. On the other hand, kNN can find very complex patterns but its output is more challenging to interpret.

OUTPUT

ASSIGNMENT NO: 3

TITLE: Gradient Descent Algorithm

PROBLEM STATEMENT: Implement Gradient Descent Algorithm to find the local minima of a function. For example, find the local minima of the function y=(x+3)2 starting from the point x=2.

PREREQUISITES:

Understanding the basic concepts of calculus and understanding concepts

like matrix multiplication, vector operations, and eigenvalues/eigenvectors.

COURSE OBJECTIVE:

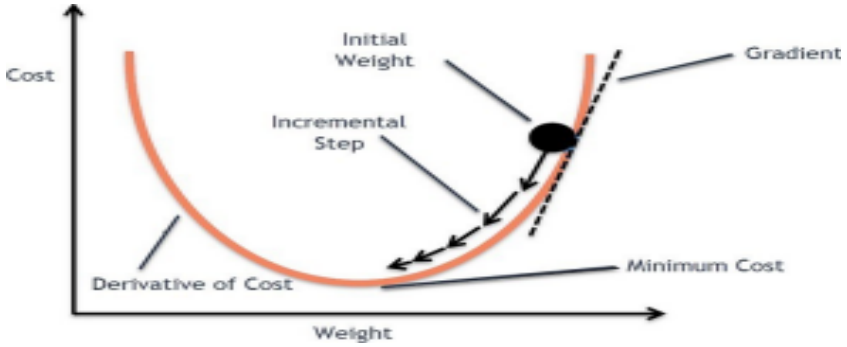
To Implement Gradient Descent Algorithm

COURSE OUTCOME:

Understanding of Implementation of Gradient Descent Algorithm

THEORY:

Gradient descent is an iterative optimization algorithm for finding the local minimum of a function. To find the local minimum of a function using gradient descent, we must take steps proportional to the negative of the gradient (move away from the gradient) of the function at the current point. Gradient descent is an iterative optimization algorithm for finding the local minimum of a function. To find the local minimum of a function using gradient descent, we must take steps proportional to the negative of the gradient (move away from the gradient) of the function at the current point. If we take steps proportional to the positive of the gradient (moving towards the gradient), we will approach a local maximum of the function, and the procedure is called Gradient Ascent.



The goal of the gradient descent algorithm is to minimize the given function (say cost function).To achieve this goal, it performs two steps iteratively:

1. Compute the gradient (slope), the first order derivative of the function at that point

2. Make a step (move) in the direction opposite to the gradient, opposite direction of slope increase from the current point by alpha times the gradient at that point

The Gradient Descent algorithm

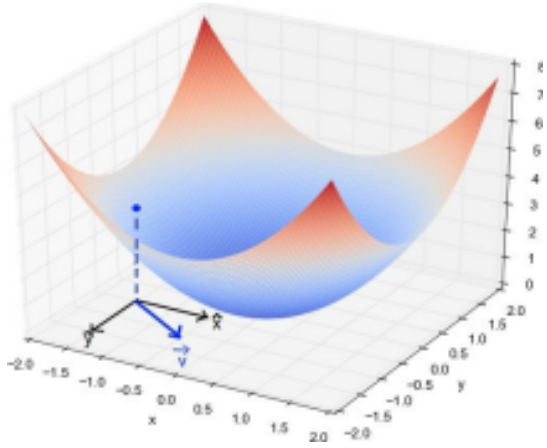
- **Initialization:** Start with an initial set of parameters or weights (often chosen randomly) for your model.
- **Define the Cost Function:** This function quantifies how well your model is performing. In the context of machine learning, it's typically a function that measures the difference between the predicted values and the actual values (i.e., the error). The goal is to minimize this function.
- **Compute the Gradient:** Calculate the gradient of the cost function with respect to the model's parameters. The gradient points in the direction of the steepest increase in the cost function.
- **Update Parameters:** Adjust the model's parameters in the opposite direction of the gradient to minimize the cost function.
  This is done using the formula
  new_parameters = old_parameters - learning_rate * gradient
  Here, the learning rate is a hyperparameter that determines the step size in each iteration. It's crucial to choose an appropriate learning rate, as too large a learning rate can lead to

overshooting the minimum, and too small a learning rate can slow down convergence.

- **Repeat:** Steps 3 and 4 are repeated for a specified number of iterations (epochs) or until a convergence criterion is met. Convergence criteria can be based on the change in the cost function or the gradient magnitude.

Plotting the Gradient Descent Algorithm

When we have a single parameter (theta), we can plot the dependent variable cost on the y-axis and theta on the x-axis. If there are two parameters, we can go with a 3-D plot, with cost on one axis and the two parameters (thetas) along the other two axes.



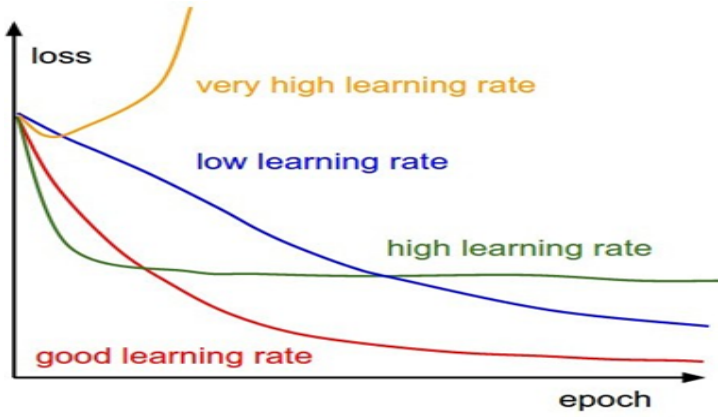
It can also be visualized by using Contours. This shows a 3-D plot in two dimensions with parameters along both axes and the response as a contour. The value of the response increases away from the center and has the same value along with the rings. The response is directly proportional to the distance of a point from the center (along a direction).

Alpha – The Learning Rate

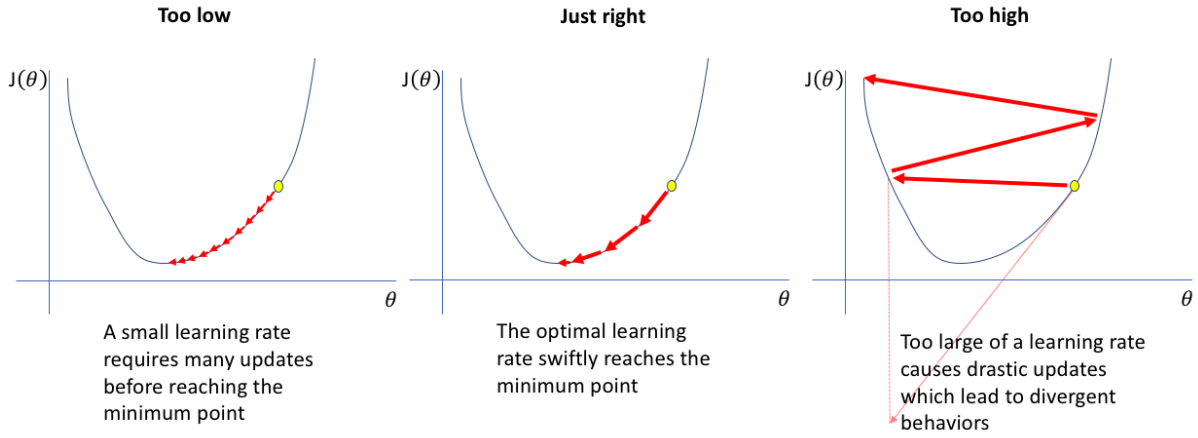
Direction we want to move in, now we must decide the size of the step we must take. *It must be chosen carefully to end up with local minima.

- If the learning rate is too high, we might OVERSHOOT the minima and keep bouncing, without reaching the minima
- If the learning rate is too small, the training might turn out to be too long
- Learning rate is too small, it takes more time but converges to the minimum
- Learning rate is optimal, model converges to the minimum
- Learning rate is higher than the optimal value, it overshoots but converges ($1/C < \eta < 2/C$)
- Learning rate is very large, it overshoots and diverges, moves away from the minima,

performance decreases on learning



There are many scenarios to expect and to take into consideration while using gradient descent :



Local Minimum:

A local minimum is a point in the optimization landscape where the function has a lower value compared to its neighboring points but not necessarily the absolute lowest value in the entire domain.

In the context of Gradient Descent, if the algorithm starts at a random point and iteratively updates its position based on the gradient, it may converge to a local minimum. This local minimum may be a good solution, but it's not guaranteed to be the best possible solution in the entire parameter space.

Local minima can be problematic in optimization because the algorithm can get trapped in them, preventing it from reaching the global minimum**.**

Global Minimum:

The global minimum is the lowest possible value of the function over the entire domain.

In the context of optimization problems, finding the global minimum is often the ultimate goal

because it represents the best possible solution.

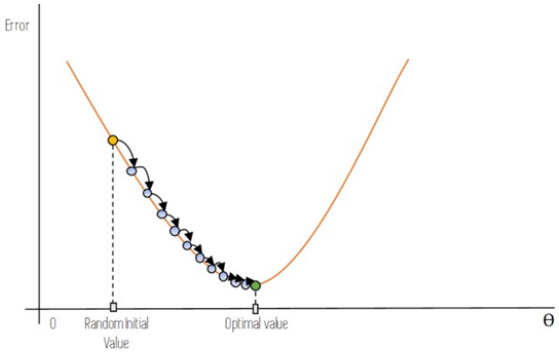
However, in complex optimization landscapes, it can be challenging to guarantee that you've found the global minimum. Many optimization algorithms, including Gradient Descent, may find a local minimum, but they may not necessarily find the global minimum unless certain conditions are met.

CONCLUSION: We are able to understand the use gradient descent to find the values of a function's parameters to minimize function costs, programmers use gradient descent as an optimization algorithm when training machine learning models. Gradient descent iteratively adjusts some of its parameters to minimize a particular function based on convex functions.

FAQ'S:

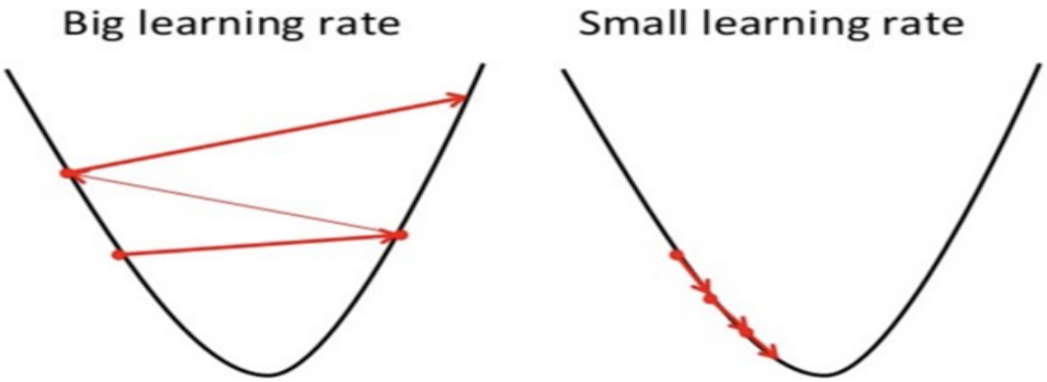
Q1. Explain the intuition behind Gradient Descent algorithm

Ans:Gradient descent is an optimization algorithm that's used when training a machine learning model and is based on a convex function and tweaks its parameters iteratively to minimize a given function to its local minimum (that is, slope = 0). For a start, we have to select a random bias and weights, and then iterate over the slope function to get a slope of 0.



The way we change the value of the bias and weights is through a variable called the learning rate.
We have to be wise on the learning rate because choosing:

- A small leaning rate may lead to the model to take some time to learn
- A large learning rate will make the model converge as our pointer will shoot and we'll not be able to get to minima.



Q2. What is the difference between Cost Function vs Gradient Descent?

Ans: A Cost Function is something we want to minimize. For example, our cost function might be the sum of squared errors over the training set. Gradient Descent is a method for finding the minimum of a function of multiple variables.

Q3. What is the idea behind the Gradient Descent?

Ans:

- A Gradient Descent is a type of optimization algorithm used to find the local minimum of a differentiable function.
- The main idea behind the gradient descent is to take steps in the negative direction of the gradient. This will lead to the steepest descent and eventually it will lead to the minimum point.
- It is shown as an equation by:

an+1=an−γ∇F(an)a_{n+1} = a_n - \gamma \nabla F(a_n)an+1=an−γ∇F(an) Where:

- a is the point.
- γ\gammaγ is the step size.
- F(x) is the multivariable function.

Q4. Compare Batch Gradient Descent and Stochastic Gradient Descent

Ans: The applicability of batch or stochastic gradient descent depends on the error manifold expected.

- Batch gradient descent computes the gradient using the whole dataset. This is great for convex,

or relatively smooth error manifolds. In this case, we move somewhat directly towards an optimum solution, either local or global. Additionally, batch gradient descent, given an annealed learning rate, will eventually find the minimum located in its basin of attraction.

- Stochastic gradient descent (SGD) computes the gradient using a single sample. Most applications of SGD use a minibatch of several samples. SGD works better than batch gradient descent for error manifolds that have lots of local maxima/minima. In this case, the somewhat noisier gradient calculated using the reduced number of samples tends to jerk the model out of local minima into a region that hopefully is more optimal. Single samples are noisy, while mini-batches tend to average a little of the noise out. Thus, the amount of jerk is reduced when using mini-batches. A good balance is struck when the minibatch size is small enough to avoid some of the poor local minima but large enough that it doesn't avoid the global minima or better-performing local minima.

Q5. Explain how does the Gradient descent work in Linear Regression

Ans: The Gradient Descent works by starting with random values for each coefficient in the linear regression model.

- After this, the sum of the squared errors is calculated for each pair of input and output values (loss function), using a learning rate as a scale factor.
- For each iteration, the coefficients are updated in the direction towards minimizing the error,
- Then we keep repeating the iteration process until a minimum sum squared error is achieved or no further improvement is possible.

OUTPUT

ASSIGNMENT NO: 4

TITLE: K-Nearest Neighbors algorithm

PROBLEM STATEMENT: Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset.
Dataset link : https://www.kaggle.com/datasets/abdallamahgoub/diabetes

PREREQUISITES:

- Understanding the fundamental concepts of supervised learning.
- Understanding of distance metrics to measure the similarity between data points. Distance metrics like Euclidean distance and Manhattan distance, which are commonly used in k-NN.
- Concept of confusion matrix

COURSE OBJECTIVE:

To understand the concept of K-Nearest Neighbors algorithm

COURSE OUTCOME:

Understanding of the concept of K-Nearest Neighbors algorithm

THEORY:

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. Algorithms can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. At the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

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

- Step-1: Select the number K of the neighbors
- Step-2: Calculate the Euclidean distance of K number of neighbors
- Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step-4: Among these k neighbors, count the number of the data points in each category.
- Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
- Step-6: Our model is ready.

Confusion Matrix :

A confusion matrix is a performance measurement tool used in machine learning to evaluate the quality of a classification model. It provides a summary of the model's predictions and the actual outcomes in a tabular format, allowing you to assess how well the model performed for different classes.

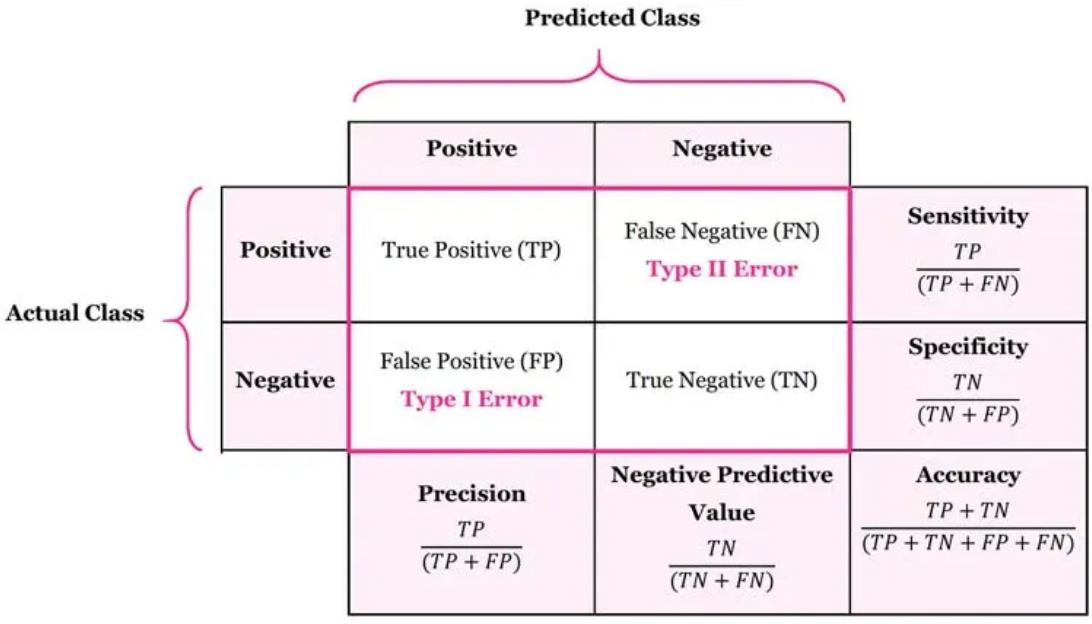


Fig. Confusion matrix

In a binary classification scenario (two classes, often labeled as "positive" and "negative"), the confusion matrix has the following elements:

True Positives (TP): The model correctly predicted instances of the positive class.

True Negatives (TN): The model correctly predicted instances of the negative class.

False Positives (FP): The model incorrectly predicted instances as the positive class when they were actually the negative class (Type I error).
False Negatives (FN): The model incorrectly predicted instances as the negative class when they were actually the positive class (Type II error).

Dataset Description : This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- BloodPressure: Diastolic blood pressure (mm Hg)
- SkinThickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg/(height in m)^2)
- DiabetesPedigreeFunction: Diabetes pedigree function
- Age: Age (years)
- Outcome: Class variable (0 or 1)

CONCLUSION: Thus, we have learned k-NN is often used for simple classification problems and serves as a baseline model. A confusion matrix provides a more detailed view of a classification model's performance compared to a single accuracy score.

FAQ'S:

Q1. What is the K value in KNN?

Ans. K value indicates the count of the nearest neighbors. We have to compute distances between test points and trained labels points. Updating distance metrics with every iteration is computationally expensive, and that's why KNN is a lazy learning algorithm.

Q2. What is the difference between k means and KNN?

Ans. k-Means Clustering is an unsupervised learning algorithm that is used for clustering whereas KNN is a supervised learning algorithm used for classification. KNN is a classification algorithm which falls under the greedy techniques however k-means is a clustering algorithm (unsupervised machine learning technique).

Q3. Why KNN is non-parametric?

Ans. KNN is a lazy learning, non-parametric algorithm. It uses data with several classes to predict the classification of the new sample point. KNN is non-parametric since it doesn't make any assumptions on the data being studied, i.e. the model is distributed from the data.

Q4. State advantages and disadvantages of KNN

Ans. Advantages of KNN Algorithm:
- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:
- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

Q5.What is the confusion matrix in KNN?

Ans. The confusion matrix is a table that is used to show the number of correct and incorrect predictions on a classification problem when the real values of the Test Set are known.

OUTPUT

ASSIGNMENT NO: 5

TITLE: K-Means clustering algorithm

PROBLEM STATEMENT:
Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset. Determine the number of clusters using the elbow method.
Dataset link : https://www.kaggle.com/datasets/kyanyoga/sample-sales-data

PREREQUISITES: -

Understanding of basic of K-Means Clustering and Hierarchical Clustering

COURSE OBJECTIVE:

To understand the concept of K-Means Clustering and Hierarchical Clustering

COURSE OUTCOME:

Understanding of the concept of K-Means Clustering and Hierarchical Clustering

THEORY:

K-means clustering is a widely used unsupervised machine learning algorithm that divides a dataset into K distinct, non-overlapping clusters. It aims to group similar data points together and separate dissimilar data points.

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what K-means clustering algorithm is, how the algorithm works, along with the Python implementation of k-means clustering. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.
The k-means clustering algorithm mainly performs two tasks:
● Determines the best value for K center points or centroids by an iterative process.
● Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.
Hence each cluster has data points with some commonalities, and it is away from other clusters.

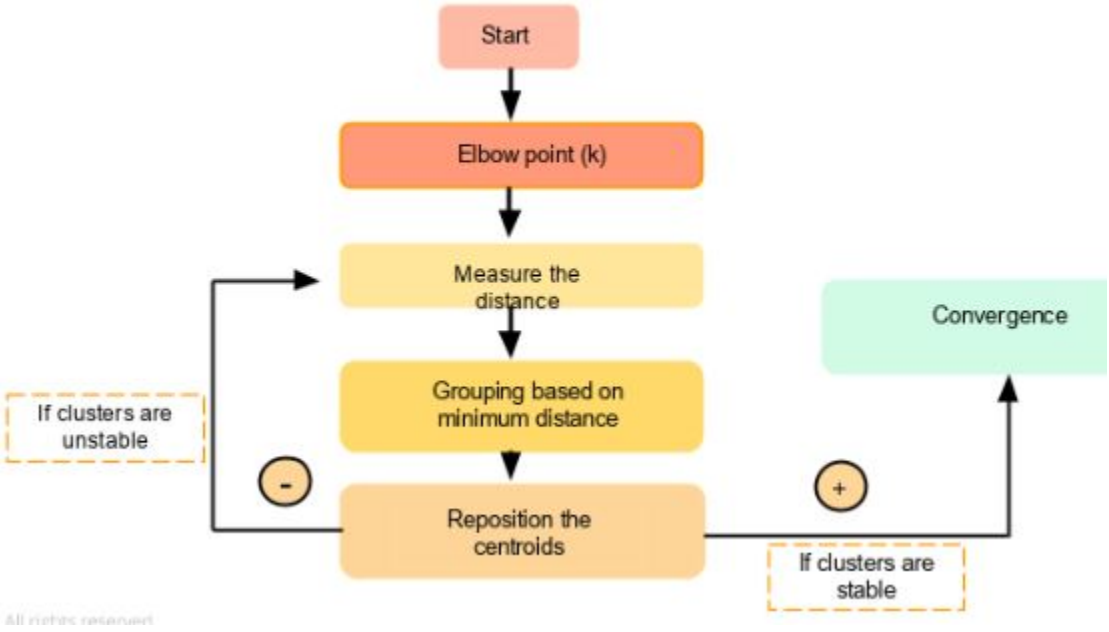


Fig. K-means Clustering working

Algorithm of K-means clustering:

The working of the K-Means algorithm is explained in the below steps:
Step-1: Select the number K to decide the number of clusters.

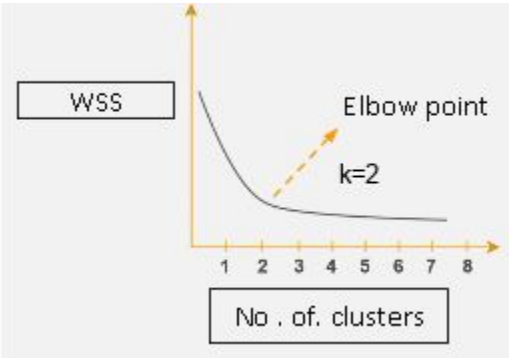
$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$

Where x_i = data point and c_i = closest point to centroid

The Elbow method is the best way to find the number of clusters. The elbow method constitutes running K-Means clustering on the dataset.
Next, we use within-sum-of-squares as a measure to find the optimum number of clusters that can be formed for a given data set. Within the sum of squares (WSS) is defined as the sum of the squared

distance between each memberThe WSS is measured for each value of K. The value of K, which has the least amount of WSS, is taken as the optimum value.

Now,draw a curve between WSS and the number of clusters.



Here, WSS is on the y-axis and the number of clusters on the x-axis.We can see that there is a very gradual change in the value of WSS as the K value increases from 2. So, you can take the elbow point value as the optimal value of K. It should be either two, three, or at most four. But, beyond that, increasing the number of clusters does not dramatically change the value in WSS, it stabilizes. B of the cluster and its centroid.

Step-2: Select random K points or centroids. (It can be different from the input dataset).
Step-3: Assign each data point to their closest centroid, which will form the predefi ned K clusters.
Step-4: Calculate the variance and place a new centroid of each cluster.
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
Step-7: The model is ready.

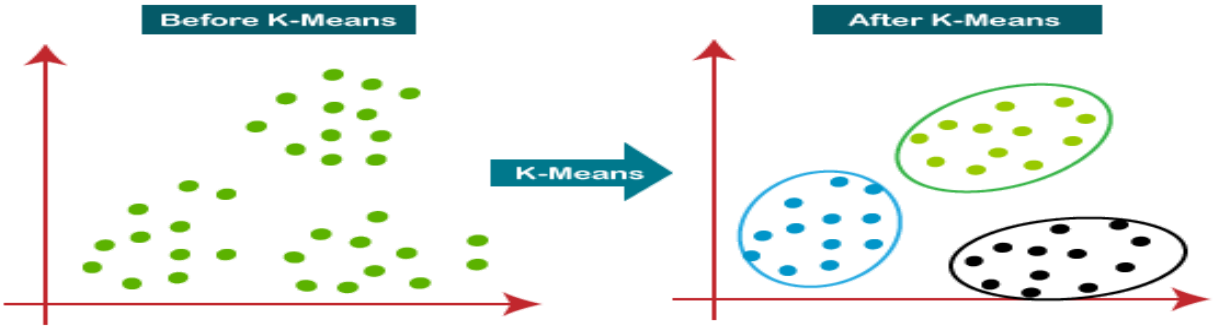


Fig.K-means Cluster formation

Hierarchical clustering: Hierarchical clustering is a method of cluster analysis that builds a hierarchy of clusters. It is an unsupervised machine learning technique that does not require specifying the number of clusters in advance. Hierarchical clustering can be divided into two main types: Agglomerative (bottom-up) and Divisive (top-down).

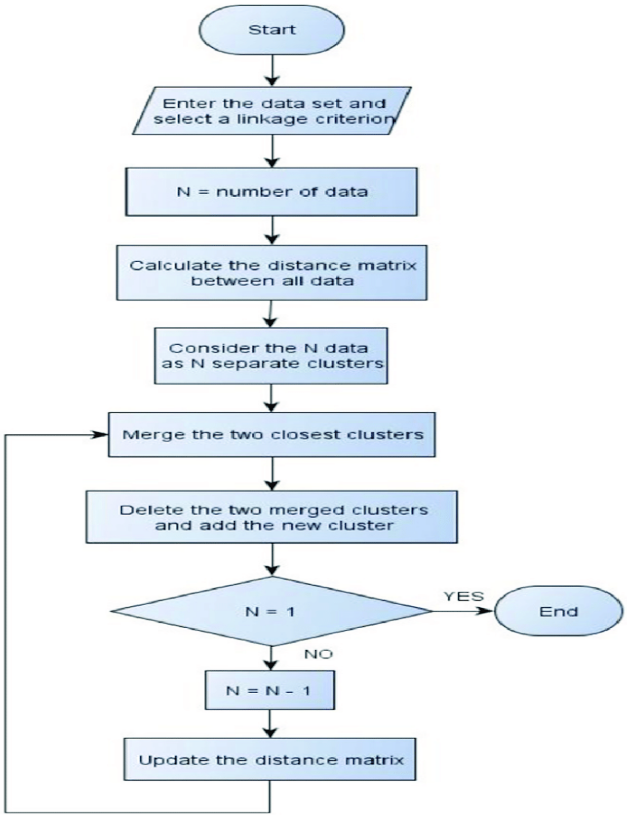


Fig. Hierarchical Clustering working

The hierarchical clustering technique has two approaches:

Agglomerative Hierarchical clustering:

The agglomerative hierarchical clustering algorithm is a popular example of HCA. To group the datasets into clusters, it follows the bottom-up approach. It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together. It does this until all the clusters are merged into a single cluster that contains all the datasets. This hierarchy of clusters is represented in the form of the dendrogram.

Steps of Agglomerative Hierarchical Clustering:

- Start with each data point as an individual cluster.
- Find the two nearest clusters and merge them into a single cluster.
- Recalculate the distances between the new cluster and the remaining clusters.
- Repeat the merging and distance updating until only one cluster containing all data points remains.
- Create a dendrogram to visualize the hierarchy.

Divisive Hierarchical Clustering (Top-Down):

Divisive clustering takes the opposite approach by starting with all data points in a single cluster and then recursively divides the clusters into smaller clusters.It is a "top-down" approach where the entire dataset is partitioned into smaller clusters step by step.Divisive clustering can be computationally intensive and less commonly used.

Steps of Divisive Hierarchical Clustering:

- Start with all data points in a single cluster.
- Identify the cluster that is least similar (e.g., highest inter-cluster distance) and divide it into two smaller clusters.
- Recursively divide the clusters into smaller clusters based on some criterion until individual data points form separate clusters.
- Create a dendrogram to visualize the hierarchy.

CONCLUSION:We have understand K-means clustering is a powerful unsupervised machine learning technique that groups data points into distinct clusters based on similarity, enabling insights and pattern discovery in a wide range of applications

FAQ's:

Q1. What does K mean clustering work?

Ans: K-means clustering is a simple unsupervised learning algorithm that is used to solve clustering problems. It follows a simple procedure of classifying a given data set into a number of clusters, defined by the letter "k,"

Q2. Is k-means clustering supervised or unsupervised?

Ans: K-Means clustering is an unsupervised learning algorithm. There is no labeled data for this clustering, unlike in supervised learning.

Q3. What are centroids in k-means?

Ans. Centroid is the center of a cluster but initially, the exact center of data points will be unknown so, we select random data points and define them as centroids for each cluster.

Q4. What is the objective of the k-means algorithm?

Ans. The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.

Q5. State Advantages of k-means clustering.

Ans: ● It is very easy to understand and implement.

● If we have a large number of variables then, K-means would be faster than Hierarchical clustering.

● On re-computation of centroids, an instance can change the cluster.

OUTPUT

Mini-Projects

TITLE: Mini-project

PROBLEM STATEMENT:
Mini Project - Use the following dataset to analyze ups and downs in the market and predict future stock price returns based on Indian Market data from 2000 to 2020. Dataset Link: https://www.kaggle.com/datasets/sagara9595/stock-data

Build a machine learning model that predicts the type of people who survived the Titanic shipwreck using passenger data (i.e. name, age, gender, socio-economic class, etc.). Dataset Link: https://www.kaggle.com/competitions/titanic/data

Develop an application for signature identification by creating your own dataset of your college student.

PREREQUISITES:

Understanding the fundamental concepts of machine learning.

COURSE OBJECTIVE:

To understand and implement the algorithm related to machine learning

COURSE OUTCOME:

Understanding and implementing the algorithm related to machine learning

GROUP: B

Content Beyond Syllabus

ASSIGNMENT NO: 6

TITLE: Implementation of Decision Tree Classifier

PROBLEM STATEMENT: A dataset collected in a cosmetics shop showing details of customers and whether or not they responded to a special offer to buy a new lip-stick is shown in table below. Use this dataset to build a decision tree, with Buys as the target variable, to help in buying lip-sticks in the future.Find the root node of the decision tree. According to the decision tree you have made from the previous training data set, what is the decision for the test data: [Age < 21, Income = Low,Gender = Female, Marital Status = Married]?

PREREQUISITES:

Understanding of the decision tree classifier.

COURSE OBJECTIVE:

To implement decision tree classification

COURSE OUTCOME:

Implementation of decision tree classifier.

THEORY:

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. A Decision Tree consists of:

- Nodes
- Leaves
- Terminals

Decision tree algorithms fall 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.

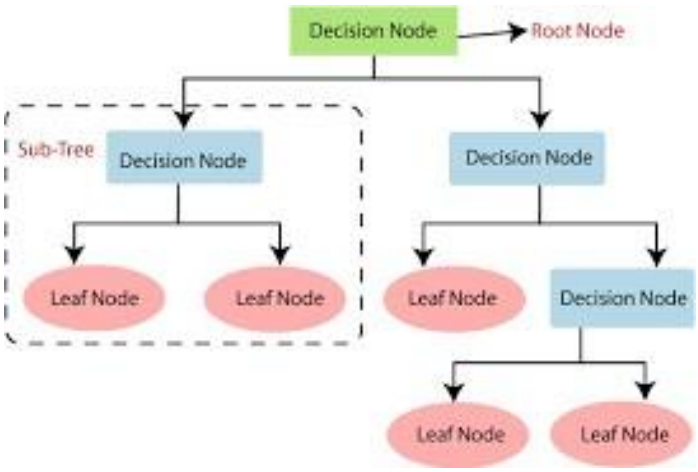
Below are some assumptions that we made while using decision tree:
• At the beginning, we consider the whole training set as the root.
• Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.

• On the basis of attribute values records are distributed recursively.

• We use statistical methods for ordering attributes as root or the internal node.

ID3 (Iterative Dichotomiser 3)

In Decision Tree Learning, ID3 is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset. It is typically used in the machine learning and natural language processing domains.



Working of ID3:

The ID3 algorithm begins with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set S and calculates the entropy H(S) (or information gain IG(S)) of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set S is then split or partitioned by the selected attribute to produce subsets of the data. (For example, a node can be split into child nodes based upon the subsets of the population whose ages are less than 50, between 50 and 100, and greater than 100.) The algorithm continues to recur on each subset, considering only attributes never selected before.

Recursion on a subset may stop in one of these cases:

● every element in the subset belongs to the same class; in which case the node is turned into a leaf node and labeled with the class of the examples.

● there are no more attributes to be selected, but the examples still do not belong to the same class. In this case, the node is made a leaf node and labeled with the most common class of the examples in the subset.

● there are no examples in the subset, which happens when no example in the parent set was found to match a specific value of the selected attribute. An example could be the absence of a person among the population with age over 100 years. Then a leaf node is

created and labeled with the most common class of the examples in the parent node's set. Throughout the algorithm, the decision tree is constructed with each non-terminal node (internal node) representing the selected attribute on which the data was split, and terminal nodes (leaf nodes) representing the class label of the final subset of this branch.

• Attribute selection

• Entropy, Information, Information Gain

• Gain Ratio

In Decision Tree the major challenge is to identify 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. Entropy

Algorithm:

Decision Tree Algorithm Pseudocode

1. Start at the root. Place the best attribute of the dataset at the root of the tree.

2. Perform the test.

3. Follow the edge corresponding to the outcome.

4. Goto 2 unless leaf.

5. Predict that outcome associated with the leaf.

ID3 Algorithm

Function ID3

• Input: Example set S

• Output: Decision Tree DT
If all examples in S belong to the same class c

return a new leaf and label it with c

Else

i. Select an attribute A according to some heuristic function

ii. Generate a new node DT with A as test

iii. For each Value v i of A

(a) Let S i = all examples in S with A = v i

(b) Use ID3 to construct a decision tree DT i for example set S i

(c) Generate an edge that connects DT and DT i

Conclusion: We are able to understand the use of decision tree classifiers.

FAQs:

Q1. What is a Decision Tree?

Ans: A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes).

Q2. What are the advantages of Decision Trees?

Ans: Compared to other algorithms, decision trees require less effort for data preparation during pre-processing. A decision tree does not require normalization of data, A decision tree does not require scaling of data as well. Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.

Q3. Why is the decision tree used?

Ans: Decision trees help you to evaluate your options. Decision Trees are excellent tools for helping you to choose between several courses of action. They provide a highly effective structure within which you can lay out options and investigate the possible outcomes of choosing those options.

Q4. Which algorithms are used for the creation of decision trees?

Ans: 1. ID3 (Iterative Dichotomiser): Uses Information Gain as attribute selection measure. 2. C4.5 (Successor of ID3): Uses Gain Ratio as attribute selection measure. 3. CART (Classification and Regression Trees) – Uses Gini Index as attribute selection measure.

Q5. What is CART?

Ans:The CART stands for Classification and Regression Trees is a greedy algorithm that greedily searches for an optimum split at the top level, then repeats the same process at each of the subsequent levels.

OUTPUT