

ANTICIPATING THE FUTURISTIC LOANS CONFIRMATION USING MACHINE LEARNING

A PROJECT REPORT

**Submitted in partial fulfillment of the requirements
for the award of**

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

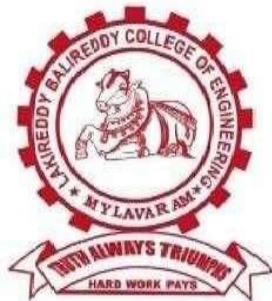
BY

M. TEJASAI	(19761A0496)
P. NITEESH REDDY	(19761A04A5)
K. THARUN	(19761A0487)

Under the Guidance of

Dr. V. RAVI SEKHARA REDDY

Sr.Asst Professor



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)

L.B. REDDY Nagar, Mylavaram-521230,

Affiliated to JNTUK, Kakinada & Approved by AICTE, New Delhi

Accredited by NAAC & NBA

Certified by ISO 21001:2018,

(2022-2023)

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)

L.B.Reddy Nagar, Mylavaram – 521 230.

Affiliated to JNTUK, Kakinada & Approved by AICTE, New Delhi

Accredited by NBA and NAAC, Certified by ISO 21001:2018

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING



CERTIFICATE

This is to certify that project work entitled “**ANTICIPATING THE FUTURISTIC LOANS CONFIRMATION USING MACHINE LEARNING**” is a bonafide work done and submitted by M. Teja Sai (19761A0496), P. Niteesh Reddy (19761A04A5) and K. Tharun (19761A0487) in partial fulfillment of requirement for the award of Bachelor of Technology in Electronics and Communication Engineering in Lakireddy Bali Reddy College Of Engineering, Mylavaram during the academic year 2022-2023.

PROJECT GUIDE

Dr. V. Ravi Sekhara Reddy
Sr. Asst Professor
Project Guide

HEAD OF THE DEPARTMENT

Dr. Y.Amar Babu
Professor &HOD

EXTERNAL EXAMINER

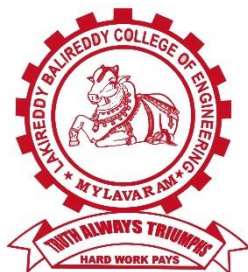
LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)

L.B.Reddy Nagar, Mylavaram – 521 230.

Affiliated to JNTUK, Kakinada & Approved by AICTE, New Delhi

Accredited by NBA and NAAC, Certified by ISO 21001:2018

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING



DECLARATION

We hereby declare that the project entitled “**ANTICIPATING THE FUTURISTIC LOANS CONFIRMATION USING MACHINE LEARNING**” submitted for the award of B.Tech. in Electronics and Communication Engineering is our original work and the project has not submitted to any other institution or University for the award of any degree.

M. Teja Sai	(19761A0496)
P. Niteesh Reddy	(19761A04A5)
K. Tharun	(19761A0487)

Place:

Date:

ACKNOWLEDGEMENTS

The Satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless co-operation made it possible, whose constant guidance and encouragement crown all efforts with success.

We humbly express our thanks to our Principal **Dr. K. Appa Rao** for extending his support and for providing us with an environment to complete our project successfully.

We would also like to thank our Vice Principal, **Dr. K. Harinadha Reddy** for encouraging us which certainly helped to complete the project in time.

We would like to express our deep gratitude to Dean R & D, **Dr. E. V. Krishna Rao** sir for his valuable suggestions.

We deeply indebted to our Head of Department **Dr. Y. Amar Babu**, who modeled us both technically and morally for achieving greater success in life.

We extremely thankful to our guide **Dr.V.Ravi Sekhara Reddy**, Sr.Asst Professor, Department of Electronics and Communication Engineering, for his/her excellent guidance, timely and valuable suggestions and encouragement that enabled the success of the project.

We would like to express our heart full thanks to our parents for their unflinching support and constant encouragement throughout the period of our project work for making it a successful one.

We would like to thank all the teaching and non-teaching staff members of Electronics and Communication Engineering, who have extended their full co-operation during the course of our project.

We thank all our friends who helped us sharing knowledge and by providing material to complete the project in time.

M.TEJASAI

(19761A0496)

P.NITEESH REDDY

(19761A04A5)

K.THARUN

(19761A0487)

ABSTRACT

In our banking system, banks are capable of selling a range of items, but bank credit lines are their main source of income. Because of the interest on the loans they have credited, they may benefit from it. The loans that a bank provides, namely whether or not the borrowers are making timely loan payments, determine the majority of the bank's profitability or loss. By predicting loan defaulters, the bank can minimise its non-performing assets. This demonstrates how important it is to look into this topic. Previous research from this era suggest that there are numerous strategies to investigate the problem of preventing loan default. Using the logistic regression model, a key method in predictive analytics, the subject of anticipating future loan defaulters is examined. For analysis and forecasting, data from Kaggle is gathered. Using logistic regression models that have been run, several performance indicators have been created. To compare the models, performance measurements like sensitivity and specificity are employed. The model gives a range of results, as seen by the end results. The model is somewhat more accurate since it considers other variables in addition to checking account information, which reflects the customer's wealth (such as the customer's personal attributes, such as age, purpose, credit history, credit amount, and credit term). Consequently, using a logistic regression approach, it is straightforward to pinpoint the best customers to pursue in the loan approval area by analysing a customer's likelihood of defaulting on a loan.

KEYWORDS: Machine learning, Bank telemarketing, Data Mining, Training, Testing, Logistic Regression, Random Forest

INDEX

CHAPTER	TOPIC	PAGE NO
1	INTRODUCTION	1-17
	1.1. Overview	1
	1.2. Machine learning	2-4
	1.3. Types of machine learning	4
	1.3.1. Supervised learning	4
	1.3.1.1. k-Nearest Neighbor	5-6
	1.3.1.2. Decision tree	6-7
	1.3.1.3. Naïve bayes	7-8
	1.3.1.4. Logistic regression	8-9
	1.3.1.5. Support vector machine	9-10
	1.3.2. Unsupervised learning	10-11
	1.3.2.1. k-means clustering	11
	1.3.2.2. K-Nearest neighbors	11
	1.3.2.3. Hierarchical clustering	12-13
	1.3.2.4. Principal Component Analysis	14
	1.3.2.5. Neural Networks	14
	1.3.3. Reinforcement learning	15-16
	1.3.3.1. Q-learning	15
	1.3.3.2. S.A.R.S.A	15

	1.3.3.3. Deep Q-Networks	15
	1.3.3.4. Policy gradient methods	15
	1.3.3.5. Actor-critic Methods	16
	1.3.3.6. Monte carlo tree search	16
	1.3.3.7. Proximal policy optimization	16
	1.4. Applications of machine learning	17
	1.4.1. Image and speech recognition	17
	1.4.2. Natural language processing	17
	1.4.3. Recommendation systems	17
	1.4.4. Fraud detection	17
	1.4.5. Healthcare	17
	1.4.6. Financial modeling	17
	1.4.7. Autonomous vehicles	17
	1.4.8. Social media analysis	17
	1.4.9. Energy management	17
	1.4.10. Robotics	17
2	LITERATURE SURVEY	18-23
	2.1. Existing system	18
	2.2. Proposed system	19
	2.3. Feasibility study	20-23
3	SYSTEM MODEL	24-28
	3.1. Introduction	24
	3.2. System model	24
	3.2.1. Data collection	25

3.2.1.1. Define the problem	25
3.2.1.2. Determine the data sources	25
3.2.1.3. Select the appropriate data type	25
3.2.1.4. Collect sufficient data	25
3.2.1.5. Ensure data quality	25
3.2.1.6. Ensure data privacy and security	25
3.2.1.7. Label the data	25
3.2.2. Data exploration	26
3.2.2.1. Data visualization	26
3.2.2.2. Descriptive statistics	26
3.2.2.3. Correlation analysis	26
3.2.2.4. Outlier detection	26
3.2.2.5. Dimensionality reduction	26
3.2.3. Data cleaning	26
3.2.3.1. Handling missing values	26
3.2.3.2. Handling outliers	26
3.2.3.3. Data normalization	27
3.2.3.4. Removing duplicate data	27
3.2.3.5. Handling categorical variables	27
3.2.3.6. Data type conversion	27
3.2.4. Data pre-processing	27

	3.2.4.1. Data cleaning	27
	3.2.4.2. Data normalization	27
	3.2.4.3. Data encoding	27
	3.2.4.4. Data splitting	27
	3.2.4.5. Data augmentation	27
	3.2.5. Training the dataset	28
	3.2.6. Predicting output	28
	3.2.7. Error/accuracy checking	28
4	METHODOLOGY	29-43
	4.1. Introduction	29
	4.2. Tools required	29-31
	4.3. Design steps	31-43
5	RESULTS	44-46
	5.1. Training accuracy	45
	5.2. Testing accuracy	45-46
6	ADVANTAGES	47
7	APPLICATIONS	48
8	CONCLUSION	49
9	FUTURE SCOPE	50
	REFERENCES	51-52

List of Figures

Fig No.	Name of the Figure	Page No.
1.2	Machine Learning	02
1.3.1.1	The Dataset	05
1.3.1.2	The 1NN Classification Map	05
1.3.1.3	The CNN Reduced Dataset	06
1.3.1.4	Decision Tree	07
1.3.1.5	Logistic Regression	09
1.3.1.6	Support Vector Machines	10
1.3.2.1	K- Means Clustering	11
1.3.2.2	K-Nearest Neighbors	12
1.3.2.3	Agglomerative Hierarchical Clustering	13
1.3.2.4	Divisive Hierarchical Clustering	13
2.1	Existing System	18
2.2	Proposed System	19
3.2	System Model	24
4.3.1	Target Variable	35
4.3.2	Age Graph	35
4.3.3	Job Graph	36
4.3.4	Default Graph	36

Fig No.	Name of the figure	Page No.
4.3.5	Subscribed Graph Between Default and Percentage	37
4.3.6	Subscribed Graph Between Default and Percentage	38
4.3.7	Correlation Heat Map	39
4.3.8	Accuracy	40
5.1	Training Accuracy	45
5.2	Testing Accuracy	46

LIST OF TABLES

Table No.	Name of the Table	Page No.
1.2.1	Differences between Artificial Intelligence, Machine Learning and Deep Learning.	03
1.2.2	Advantages and Disadvantages of Machine Learning.	04
1.3.3	Difference between supervised and unsupervised and reinforcement learning.	16

CHAPTER 1

INTRODUCTION

1.1 Overview

The primary role of every bank is the distribution of mortgages. The distribution of mortgages is almost all banks' primary task. Most of a bank's assets are directly associated with the money it brought in via the mortgages it granted out. Putting assets in places where they are in secure ownership is the essential goal for financial systems. Although many banks and financial institutions nowadays offer mortgages afterwards an prolonged process of validation and verification, there is no belief that the application approved is the one most worth applicant out of all applicants. With the addition of this software, we can assess if an particular application could be safe or not, and machine learning is utilized to entirely automated the feature confirmation method. The model's flaw is that it gives every unit various amounts of weight. The flaw in this model is that it emphasizing different weights for each component but, in daily life, loans might be sanctioned solely based on one significant factor, which makes this model inconceivable.

The Mortgage anticipating delivers substantial advantages to both customers and executives at banks. The main objective of the article is providing an instantly effortless, and right away way of choosing suitable candidates. It might give the bank some advantages. Every attribute linked to the approval process could be granted a weight by the Mortgage Forecasting System, and on fresh test data, the same attributes are processed corresponding to their linked weight.

It would be feasible to choose a date by which the candidate can find out whether or not their loan will be acknowledged. By proceeding on to a distinct execution, Loans Forecasting Software has the ability to analyze it in alphabetical order. The sole client for this paperwork is the governing body of the financial institution or finance corporation, and no other parties will be able to tamper with it simply because the whole anticipation technique takes place in secret. Other financial offices might receive results for a particular Loan Id, permitting them to adapt to queries in the most effective manner feasible. Thus it makes it less difficult for the other departments to carry out further formalities.

1.2 Machine Learning

In machine learning, software receive instruction to come up with anticipates or judgments based on data inputs. Machine learning is an instance of artificial intelligence (AI). It is a way of teaching systems how to analyze data, spot trends, and draw conclusions or predictions without needing these features specifically encoded in it.

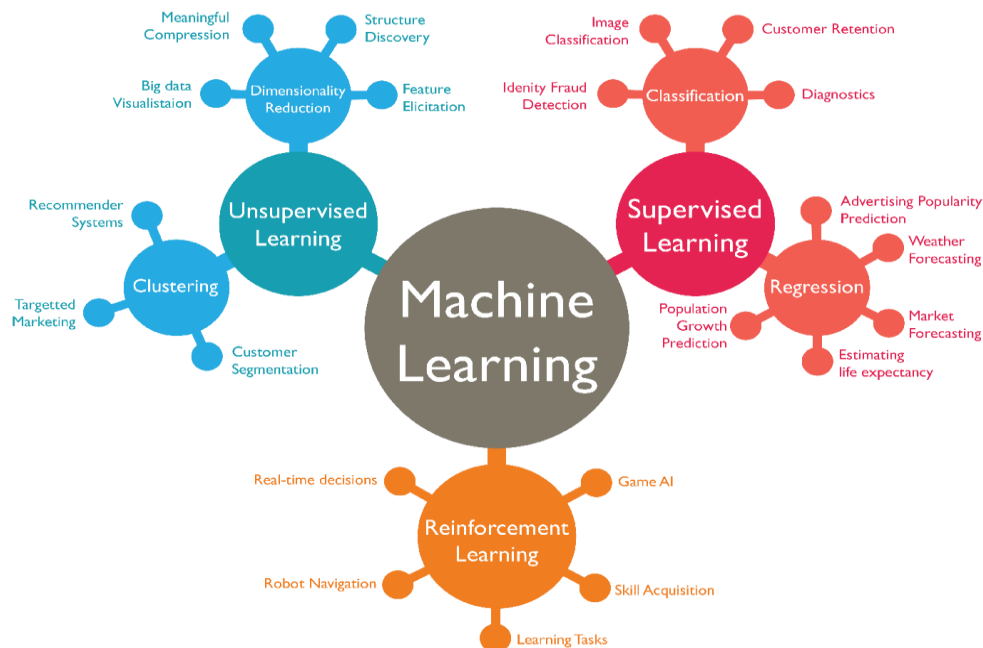


Fig. 1.2 Machine Learning

Source: <https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications>

Before predicting or drawing conclusions about new data, machine learning algorithms are trained on big datasets to look for patterns and correlations in the data. Machine learning has various applications, including fraud detection, natural language processing, personalized recommendations, picture identification, and many more.

Supervised learning, unsupervised learning, and reinforcement learning are the three primary subcategories of machine learning. An algorithm is trained on labelled data using supervised learning, where the right responses are given. When trained on unlabeled data, an algorithm must independently discover patterns and relationships. This process is known as unsupervised learning.

Table.1.2.1 Differences between Artificial Intelligence, Machine Learning and Deep Learning:

Artificial Intelligence	Machine Learning	Deep Learning
AI stands for Artificial Intelligence and is basically the study/process which enables machines to mimic human behavior through particular algorithm.	ML stands for Machine Learning and is the study that uses statistical methods enabling machines to improve with experience.	DL stands for Deep Learning, and is the study that makes use of Neural Networks (similar to neurons present in human brain) to imitate functionality just like a human brain.
AI is the broader family consisting of ML and DL as its components.	ML is the subset of AI.	DL is the subset of ML.
The aim is to basically increase chances of success and not accuracy.	The aim is to increase accuracy not caring much about the success ratio.	It attains the highest rank in terms of accuracy when it is trained with large amount of data.
The efficiency Of AI is basically the efficiency provided by ML and DL respectively.	Less efficient than DL as it can't work for longer dimensions or higher amount of data.	More powerful than ML as it can easily work for larger sets of data.
Examples of AI applications include Google's AI-Powered Predictions, Ridesharing Apps Like Uber and Lyft, Commercial Flights Use an AI Autopilot, etc.	Examples of ML applications include Virtual Personal Assistants: Siri, Alexa, Google, etc., Email Spam and Malware Filtering.	Examples of DL applications include Sentiment based news aggregation, Image analysis and caption generation, etc.

Table.1.2.2 Advantages and Disadvantages of Machine Learning

Advantages	Disadvantages
Automation of Everything	Possibility of High Error
Efficient Handling of Data	Algorithm Selection
Scope of Improvement	Data Acquisition
Best for Education and Online Shopping	Time and Space

1.3 Types of Machine Learning

1.3.1 Supervised learning

In supervised learning, a type of machine learning, the algorithm is trained using labelled data, or information that has already been annotated with the intended result. The model is trained using the tagged data so that it can correctly predict the results of unexpected and distinctive inputs.

Throughout the supervised learning process, the algorithm receives a dataset with input variables (also known as features) and associated output variables. (also known as labels or targets). By incrementally changing the model's parameters, the algorithm learns how to translate input variables to output variables.

The algorithm receives feedback on how well it is performing throughout the training phase by comparing its anticipated outputs with the actual outputs in the labelled data. The algorithm then modifies its parameters in order to reduce the discrepancy between the expected and actual results. This process is repeated until the training set accuracy of the algorithm is enough. Once trained, the model may be used to generate predictions based on fresh, unknown data. To do this, fresh data is input into the model, which then predicts the outcome for the fresh data using the parameters it learnt during training.

1. k-Nearest Neighbors

The k-Nearest Neighbors algorithm is a well-liked non-parametric classification and regression technique in machine learning. (k-NN). Predictions are made by the model in this type of instance-based learning based on how much the incoming data points match the training data points.

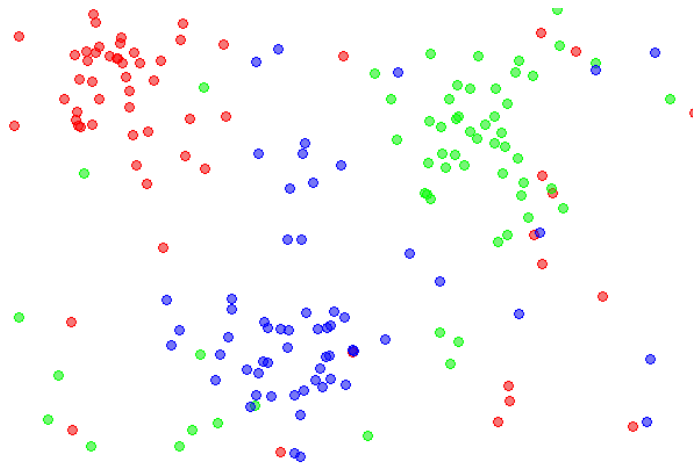


Fig. 1.3.1.1 The dataset

Source: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

The primary principle behind k-NN, which is used to forecast the class or value of the new data point, is to find the k data points in the training set that are most comparable to the new data point. Depending on the situation, different hyperparameter K values will be suitable.

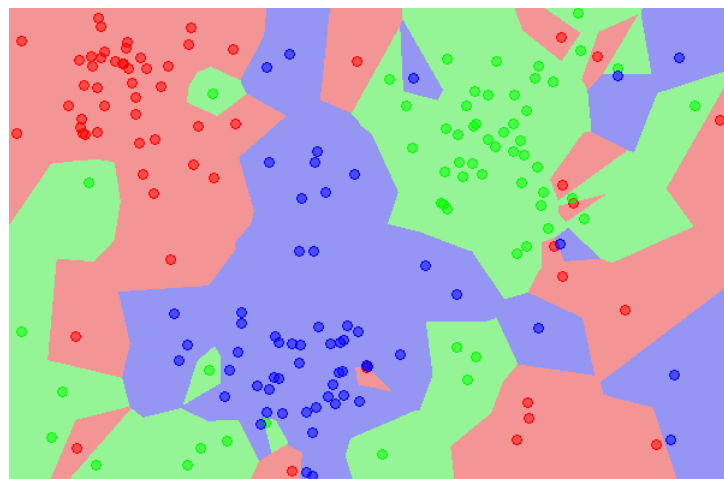


Fig. 1.3.1.2 The 1NN classification map

Source: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

In the k-NN approach for classification issues, the new data point is allocated to the majority of the class of the k nearest neighbours. The k-NN approach applies the mean or median value of the k nearest neighbours to the new data point in regression problems. The ability to handle complex decision constraints and the lack of assumptions regarding the input's underlying distribution are two benefits of k-NN. K-NN might be computationally costly, especially for big datasets, which is a drawback. The choice of unit of measurement and the value of k may also have an impact.

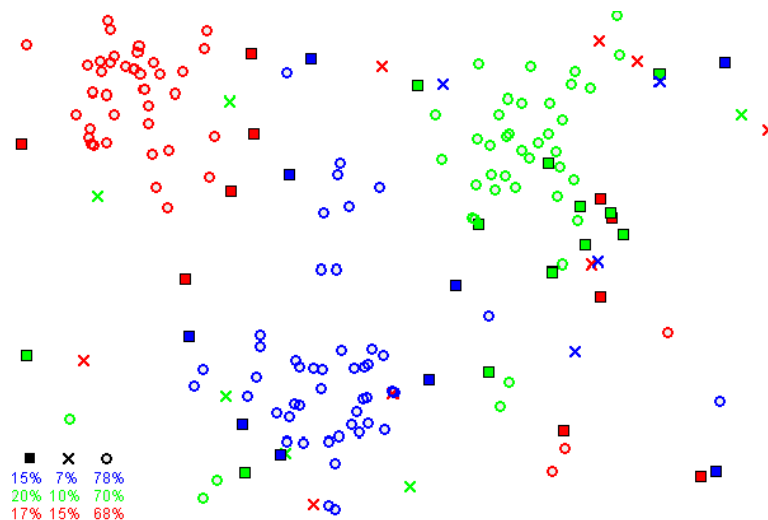


Fig. 1.3.1.3The CNN reduced dataset.

Source: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

2. Decision Trees

Classification and regression issues are dealt with using the machine learning method known as decision trees. These supervised learning techniques create models in the shape of trees, where each leaf node represents the outcome or class label, each internal node represents a feature or characteristic, and each branch represents a decision rule. The decision tree approach iteratively separates the data into subgroups based on the values of the input attributes to reduce entropy or information gain.

Up until all leaf nodes are pure or have only one class label, the procedure continues the operation for each level of the tree after that. The technique chooses the characteristic as the root node of the tree that provides the most information gain. When there are few qualities and ordered data, decision trees are a popular and efficient solution for a variety of issues.

They are frequently recruited in industries like banking, marketing, healthcare, and customer service where the capacity for rapid, correct decision-making is essential. Decision trees are used by businesses and groups because they are common and simple to comprehend and interpret. When generalising to new data, decision trees may perform badly if the tree is very complicated, which can result in overfitting. Pruning, regularization, and ensemble methods like gradient boosting and random forests have all been created as solutions to this issue.

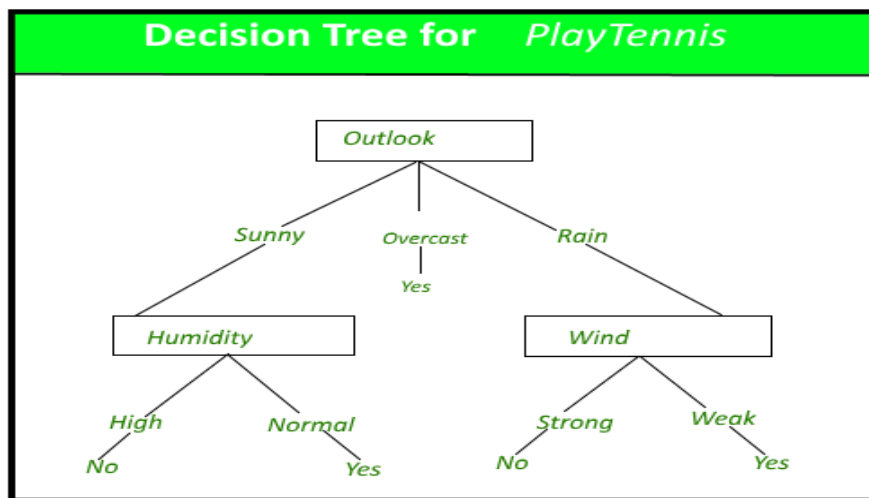


Fig.1.3.1.4 Decision Tree

Source: <https://www.geeksforgeeks.org/decision-tree/>

3. Naive Bayes

A probabilistic machine learning technique called Naive Bayes is employed for categorization problems. The Bayes theorem, which asserts that the likelihood of a hypothesis, such as a classification label, given the data (i.e., the input characteristics), is inversely proportional to the likelihood of the data given the hypothesis, multiplied by the prior probability of the hypothesis, provides the theoretical underpinning of this concept.

It is easier to calculate the likelihood that the data would support the hypothesis since the word "naive" in Naive Bayes relates to the idea that qualities are unconnected to one another. Despite this simplification, Naive Bayes has been shown to be efficient in a variety of real-world applications, especially when there are several characteristics and minimal training data.

The most popular Naive Bayes variants include:

- a) Gaussian Naive Bayes, which uses the Gaussian distribution to determine the likelihood that the data would match the hypothesis while assuming that the features are normally distributed.
- b) Multinomial Naive Bayes: This method makes use of a multinomial distribution to determine the likelihood that the data would support the hypothesis if the attributes were assumed to be counts. (such as word frequencies).
- c) Bernoulli Naive Bayes: This strategy is comparable to Multinomial Naive Bayes but assumes that the characteristics are binary (whether a word is there or not, for example), and it makes use of the Bernoulli distribution to determine the likelihood that the data will support the hypothesis.

Classification issues involving several classes and binary data may be quickly and easily handled with Naive Bayes. When the independence assumption is broken or when there are significant linkages between characteristics, it could not function as intended.

4. Logistic Regression

Logistic regression forecasts the likelihood of a binary result (such as true or false, yes or no) in a binary classification based on one or more input factors. (also known as predictors or independent variables). Logistic regression, in other words, predicts the likelihood of the result given the input variables. A probability value between 0 and 1 that shows the chance that the binary output would be accurate given the input variables is the outcome of a logistic regression model. A threshold is then used with the probability value to create a binary forecast.

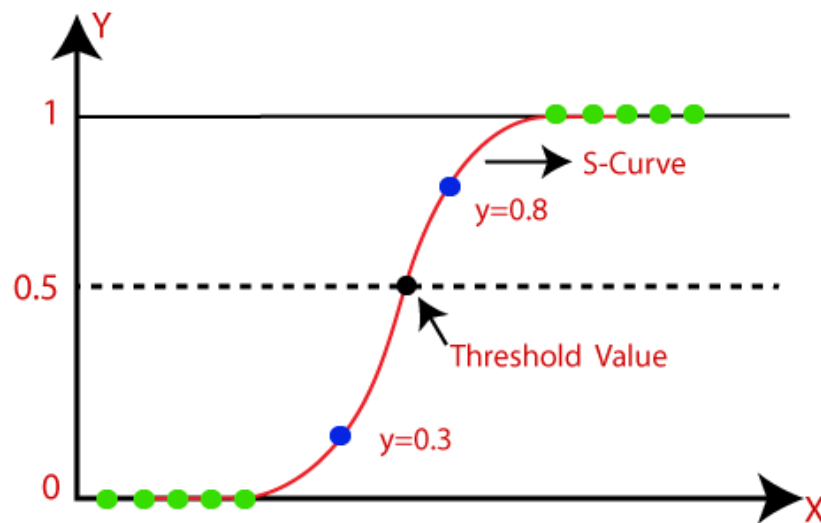


Fig.1.3.1.5 Logistic Regression

Source: <https://www.javatpoint.com/logistic-regression-in-machinelearning>

The logistic regression approach uses the logistic function, sometimes called the sigmoid function, to express the connection between the input variables and the likelihood of the outcome. Each input value is transformed by the logistic function into a number between 0 and 1, which is crucial for modelling probability.

Logistic regression is a well-liked machine learning technique for binary classification problems in numerous industries, such as banking, healthcare, and marketing. It may even be expanded to meet issues with multi-class categorization and is simple to use and understand. Although this may not always be the case in practice, logistic regression assumes a linear connection between the input factors and the output probability.

5. Support Vector Machines

Categorization, regression, and outlier detection are studied using Support Vector Machines, a type of machine learning technique. The goal of supervised learning is to develop a mapping function that can accurately classify new input data into one of several possible categories. SVMs are commonly used in this procedure. The fundamental aim of SVM is to find a hyperplane that best separates the different classes of data points in the feature space. The hyperplane, which is chosen to optimise that distance, is determined using the support vectors—the closest data points—from each class. This distance, called the margin, is measured in order to find the hyperplane with the largest margin.

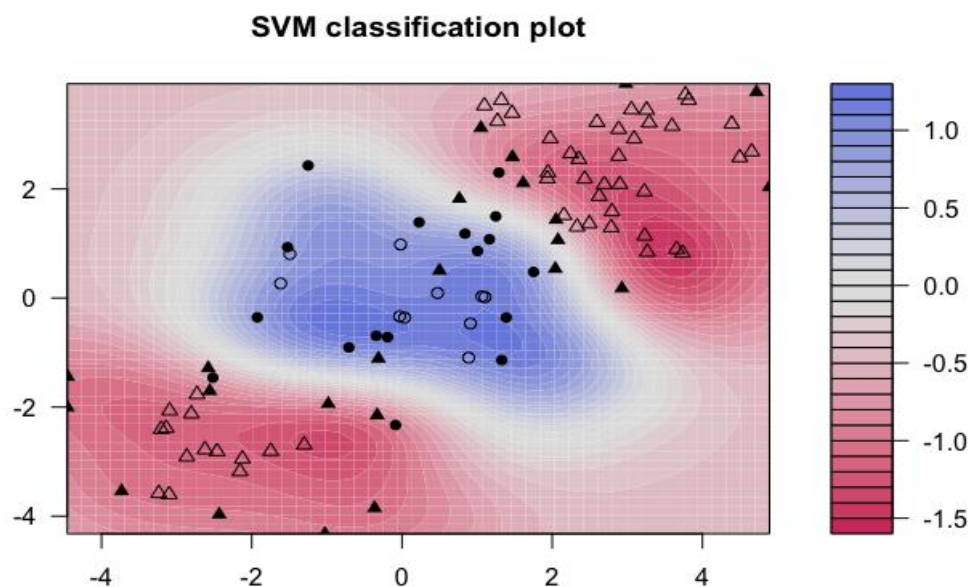


Fig.1.3.1.6 Support Vector Machines

Source: <http://uc-r.github.io/svm>

By employing kernel functions to shift the input data into a higher-dimensional space where a linear separation is achievable, SVMs are able to handle both linearly separable and non-linearly separable data. The polynomial, radial basis function (RBF), and sigmoid functions are a few typical kernel functions used with SVMs that have been demonstrated to be efficient in a variety of applications, including image classification, text classification, and bioinformatics. They could, however, be computationally costly, particularly when working with huge datasets, and they might also be sensitive to the selection of hyperparameters and kernel functions.

1.3.2 Unsupervised learning

Machine learning techniques such as unsupervised learning allow the computer to identify links and patterns in data without being explicitly taught what they are. Unsupervised learning recognises patterns in data that don't already have labels or categories, as opposed to supervised learning, when the algorithm is given labelled samples to learn from. Unsupervised learning methods can be used in applications including dimensionality reduction, anomaly detection, and grouping. Anomaly detection looks for data points that stand out from the rest of the data while clustering groups similar data points into clusters. The technique of lowering the number of features in a dataset while keeping as much information as is practicable is known as "dimensionality reduction."

Some common unsupervised learning algorithms as shown below:

- k-means clustering
- K-Nearest Neighbors (KNN)
- hierarchical clustering
- principal component analysis (PCA)
- Neural Networks

1. k-means clustering

Unsupervised learning is used in data mining and machine learning, with one example being K-means clustering. It is a straightforward and well-liked approach to similarity-based data classification. The method divides the collection of data points into k clusters based on how far off they are from a set of starting cluster centroids. Each point is given to the cluster whose centroid is closest to it after measuring the distance between it and each cluster's centroid.

The technique iteratively updates the centroids and redistributes points to clusters until the clusters stop changing. The initial centroids are often selected at random. When the within-cluster sum of squares is minimized, the algorithm reaches its convergence. When the clusters are spherical and well-isolated from one another, the K-means method is effective. However, it may not be effective for clusters with asymmetric or overlapping structures, and it may depend on the initial centroids selected. Numerous data analysis activities, including text mining, customer segmentation, and picture

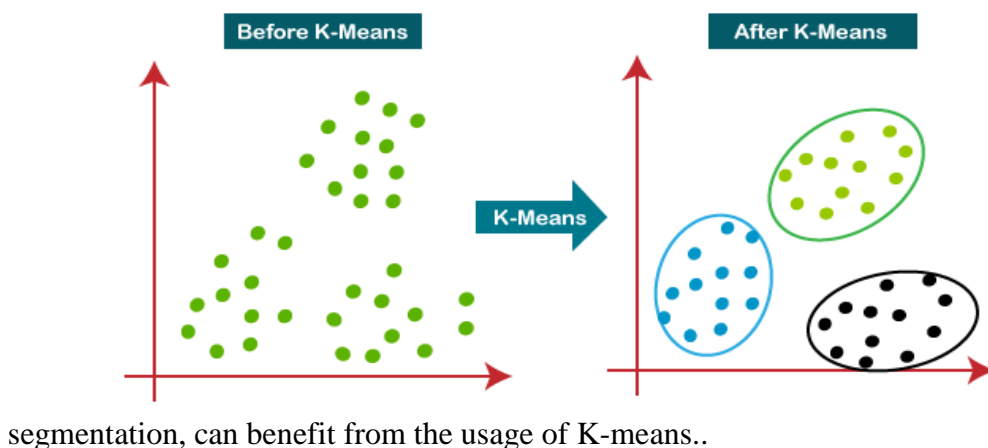


Fig.1.3.2.1 k-means clustering

Source: <https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning>

2. K-Nearest Neighbors

K-Nearest Neighbors is a machine learning method for classification and regression problems. The KNN method determines the K number of training data points that are most similar to a particular test data point, and then predicts the label or value of the test data point using the labels or values of the test data point's K nearest neighbours as a guide. K's value is a hyperparameter that may be altered to improve the algorithm's efficiency. A decision boundary may be more difficult for a lower value of K, which might lead to overfitting, whereas a decision boundary may be simpler for a higher value of K, which could lead to underfitting.

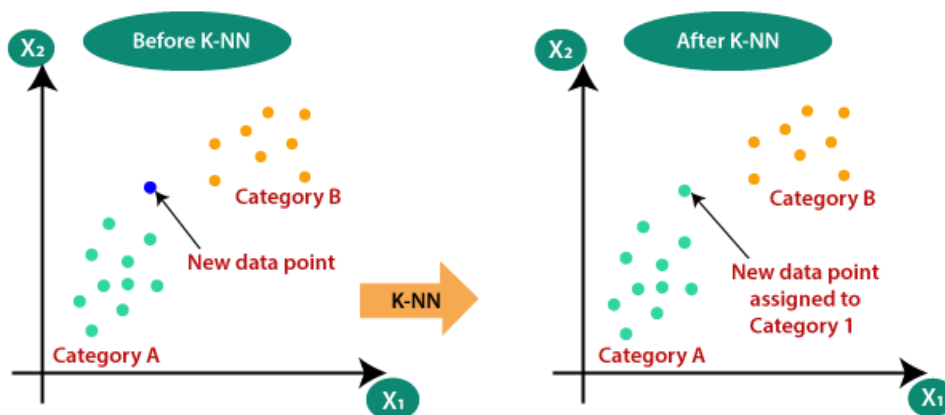


Fig.1.3.2.2 K-Nearest Neighbors

Source: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>

Because KNN is a non-parametric technique, it does not presuppose a certain data distribution and can handle both linear and nonlinear decision limits. Given that it must calculate the distances between each training and test data point, KNN could be computationally costly for huge datasets.

3. Hierarchical clustering

A form of unsupervised machine learning method called hierarchical clustering seeks to group together related data points based on their similarity or closeness. A hierarchy of clusters is created by periodically merging or dividing clusters using this clustering technique up until a halting condition is satisfied. Agglomerative and divisive clustering are the two forms of hierarchical clustering. Each data point begins aggregative clustering as its own cluster. The nearest two clusters are then repeatedly combined to combine all the points into a single cluster, depending on a similarity score. Divitive clustering, in contrast, divides the data points into incrementally smaller groups until each

data point is in its own cluster. All of the data points are first grouped together.

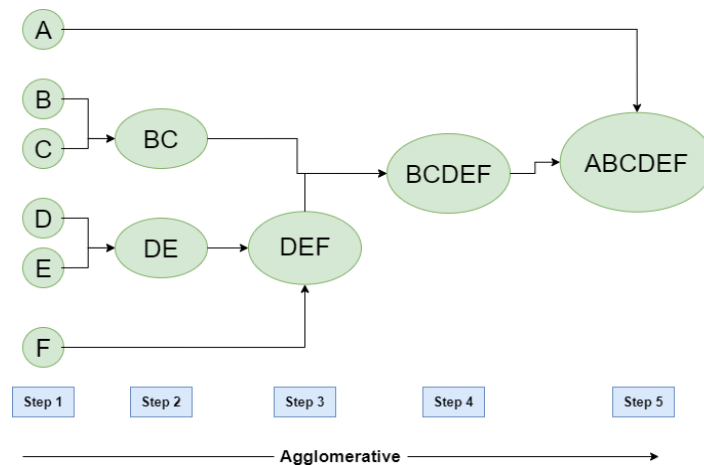


Fig.1.3.2.3 Agglomerative Hierarchical clustering

Source: <https://www.geeksforgeeks.org/hierarchical-clustering-indatamining/>

A dendrogram, which resembles a tree and depicts the hierarchy of the groups, is the result of hierarchical clustering. The dendrogram can be used to determine how many clusters are best to utilise for a certain inquiry. The height at which the dendrogram is sliced, which may be assessed using a number of techniques like the elbow approach or silhouette analysis, determines the ideal number of clusters. Numerous applications, including customer segmentation, text clustering, and picture segmentation, can benefit from hierarchical clustering. It is a powerful exploratory data analysis technique that may be used to find patterns and correlations in vast, complicated data sets.

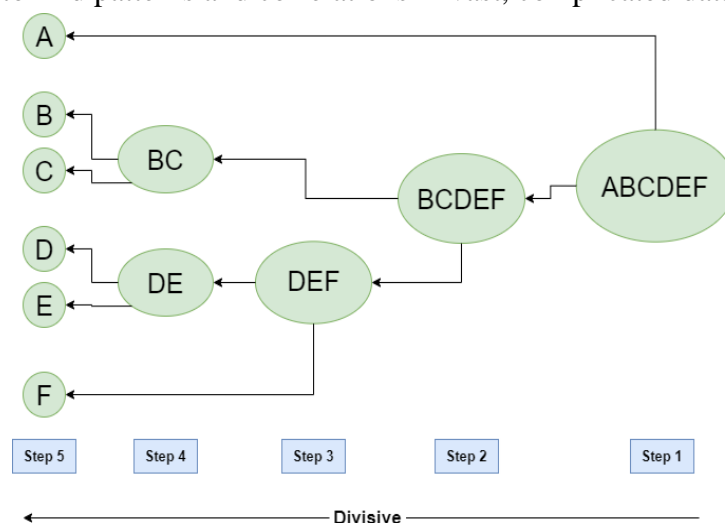


Fig.1.3.2.4 Divisive Hierarchical clustering

Source: <https://www.geeksforgeeks.org/hierarchical-clustering-indatamining/>

4. Principal Component Analysis

An analytical method called principal component analysis (PCA) is used to save as much of the original data as is practicable while reducing the dimensionality of big data sets. The first axis, also known as the first principle component, captures the bulk of variance in PCA, which transforms the data into a new coordinate system. The data's second-highest level of variation is captured by the second axis, also referred to as the second main component, and so on. Each primary component is a linear combination of the initial variables and is orthogonal (i.e., perpendicular) to the others.

PCA operates by figuring out the eigenvectors and eigenvalues of the original data's covariance matrix. The largest axes of variation in the data are reflected in the eigenvectors, and the eigenvalues indicate how much variation is contained by each eigenvector. The data is then translated into the new coordinate system, where each axis denotes a significant component, using the eigenvectors. Data visualization, data compression, and noise reduction are all possible uses for PCA. The data's reduced dimensionality makes it simpler to analyse and comprehend complicated data sets. PCA may be used to compress data by keeping just the most crucial core components, requiring less processing power and storage space.

5. Neural Networks

The structure and operation of the human brain served as an inspiration for the machine learning method known as neural networks. They consist of interconnected nodes (or neurons) that function as information processing and communication channels. Each neuron in the network gets information from one or more other neurons and then generates an output that is sent to further neurons. The network's performance is improved during training by adjusting and updating the connections between its neurons. Predictive modeling, audio and picture identification, and natural language processing are just a few of the many uses for neural networks. They thrive at jobs involving intricate data patterns or connections, which more conventional machine learning algorithms would find challenging.

There are many different kinds of neural networks, including deep neural networks, convolutional neural networks, feedforward neural networks, and recurrent neural networks. Each kind has a distinct structure and may be used to solve a variety of issues. In the context of machine learning, neural networks have emerged as a popular and efficient technology that has significantly improved a number of research and application areas.

1.3.3 Reinforcement learning

Reinforcement learning is a type of machine learning that teaches an agent to make decisions in a given environment by getting feedback in the form of rewards or penalties. Benefit-based learning aims to teach the agent how to act to optimise its total cumulative benefit. The agent engages in environmental interaction, acts, notices the state and reward that results, and then repeats the cycle. The agent changes its policy, which functions as a table of states and actions, using the new knowledge. The agent behaves in each state in line with the policy depending on the reward it anticipates receiving.

To find the best policy, reinforcement learning algorithms frequently employ trial-and-error techniques. The agent experiments with various strategies and monitors the incentives it gets, utilising this data to progressively alter its strategy. The reinforcement learning algorithms Q-learning, Deep Q-Networks, and SARSA are a few of the more popular ones. (DQNs). Numerous fields, including robotics, video games, recommendation engines, and autonomous cars, employ reinforcement learning. It is an exciting field of study with the potential to transform many industries.

Here are some common reinforcements learning algorithms:

1. **Q-learning:** Q-learning is a model-free algorithm that learns an optimal action-value function from experience. It updates the value of a state-action pair based on the reward received and the maximum Q-value of the next state.
2. **SARSA:** SARSA (State-Action-Reward-State-Action) is another model-free algorithm that learns a policy by estimating the Q-value of the current state-action pair and the next state-action pair.
3. **Deep Q-Networks (DQN):** DQN is a deep learning-based algorithm that uses a neural network to approximate the Q-value function. It has shown to be effective in complex environments and has achieved human-level performance in some video games.
4. **Policy Gradient Methods:** Policy Gradient Methods directly optimize the policy function by updating the weights of a neural network or by computing the gradient of a parameterized policy function.

5. **Actor-Critic Methods:** Actor-Critic Methods combine the policy-based and value-based approaches by having both an actor network that generates actions and a critic network that estimates the value function.
6. **Monte Carlo Tree Search (MCTS):** MCTS is a tree-based search algorithm that performs a guided exploration of the search space by repeatedly simulating the environment and selecting the most promising action based on the results of the simulations.
7. **Proximal Policy Optimization (PPO):** PPO is a policy gradient method that updates the policy in a way that is guaranteed to improve performance without moving too far away from the current policy. It has been shown to be effective in continuous control tasks.

Table 1.3.3 Differences Between Supervised vs Unsupervised vs Reinforcement Learning

Supervised ML	Unsupervised ML	Reinforcement ML
Learns by using labelled data.	Trained using unlabeled data without any guidance.	Works on interacting with the environment.
Regression and classification.	Association and Clustering.	Exploitation or Exploration.
Labelled data	Unlabeled data	No – predefined data
Linear Regression, Logistic Regression, SVM, KNN etc.	k-means clustering, K-Nearest Neighbors, Neural Networks	Q-learning, SARSA, Deep Q Network
Risk Evaluation, Forecast Sales.	Recommendation System, Anomaly Detection.	Self-Driving Cars, Gaming, Healthcare.

1.4 Applications of Machine learning

Machine learning has become an essential tool in many fields, and its applications are diverse. Here are some of the most common applications of machine learning:

1. **Image and speech recognition:** Machine learning is used to develop computer vision systems that can recognize images and videos. Speech recognition is another area where machine learning is used to improve the accuracy of automated speech recognition systems.
2. **Natural language processing:** Machine learning is used to build algorithms that can analyze, understand, and generate human language.
3. **Recommendation systems:** Machine learning algorithms are used to build recommendation systems that suggest products, movies, and other items based on a user's preferences and behaviors.
4. **Fraud detection:** Machine learning is used to detect fraudulent activities, including credit card fraud, insurance fraud, and identity theft.
5. **Healthcare:** Machine learning is used in the healthcare industry to analyze patient data and make predictions about patient outcomes.
6. **Financial modeling:** Machine learning is used to build financial models that can predict stock prices, market trends, and other financial indicators.
7. **Autonomous vehicles:** Machine learning is used to develop autonomous vehicles that can recognize traffic patterns, avoid obstacles, and navigate complex environments.
8. **Social media analysis:** Machine learning algorithms are used to analyze social media data to gain insights into consumer behavior, sentiment analysis, and other marketing-related tasks.
9. **Energy management:** Machine learning is used to optimize energy usage, predict energy consumption patterns, and improve the efficiency of energy management systems.
10. **Robotics:** Machine learning is used in robotics to teach robots how to perform tasks and make decisions based on data and sensory input.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM:

Creating an interactive user interface that accepts input from the user and presents results. The outcomes of these case studies provide insight into methods for precisely predicting loan defaulters. They test the efficacy of two machine learning algorithms, namely Random Forest, KNN, and XG Boost, to predict and categorise student performance using MI techniques. Based on client behavior, this dataset was used in the current research. If the project is successful, academics will receive a lot of support in their efforts to overhaul the financial system. The prediction accuracy for both datasets is pretty good, according to this study. For instance, if a customer is experiencing a crisis, the algorithm might not be able to forecast the right outcome.

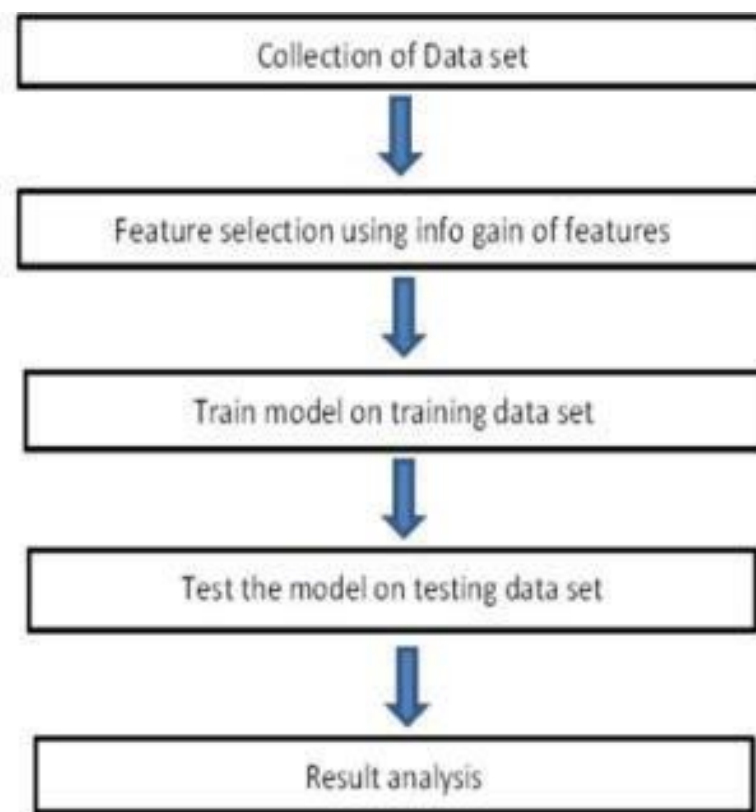


Fig.2.1 Existing System

2.2 PROPOSED SYSTEM:

Here, we'll talk about the benefits of loan prediction. Using this technique, we can determine if a loan applicant will be able to repay the loan or not. We forecast that the consumer would be given a loan if they are able to make the payments. We also forecast that the customer won't be qualified if the applicant is rejected. The benefit of this approach is that it allows us to determine from the facts if a customer fulfils the eligibility requirements by defining the algorithms and outlining specific scenarios. It is possible to create a system that can accept user input for data like salary, address, loan amount, and loan term, etc. You should state if you believe the bank will approve their application or not.

The advice of this model would evaluate a customer's prior history to describe their present conduct. These client records are used to generate a data collection. These data sets are utilised to forecast whether or not the customer's loan will be approved, together with a trained machine learning model. The suggested approach fosters the most important factors while more precisely forecasting the class of customers to determine whether or not a consumer will be able to repay the loan. The enormous volumes of client data gathered were crucial and advantageous for target marketing strategies like telemarketing. Telemarketing, which uses an interactive method to contact potential consumers via phone, mail, social media, etc. in order to make direct sales of goods or services, is an operationalized kind of direct marketing through a contact centre.

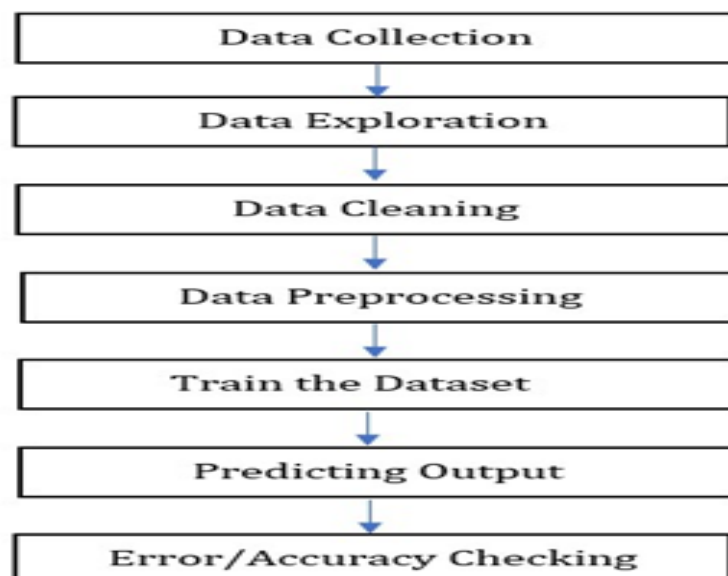


Fig.2.2 Proposed System

2.3 FEASIBILITY STUDY:

2.3.1 An Approach for Prediction of Loan Approval using Machine Learning Algorithm

Logistic regression model types. Data from Kaggle is gathered for predictions and analysis. Logistic regression models have been used to estimate the various performance measures. The models are compared based on performance criteria like sensitivity and specificity. The final results have shown that the model produces a range of results.

The well-liked and efficient machine learning method known as logistic regression is used to classification problems. The advantage of logistic regression is that it is a prediction analysis. It is used to describe data and make clear how one binary variable is related to one or more independent nominal, ordinal, and ration level variables. Credit risk calculation is a big problem. The loss of many data points from the majority class in order to balance the class is a drawback of adopting under sampling procedures.

2.3.2 Loan Default Forecasting using Data Mining

In order to assist the banks in making better judgements in the future, we applied data mining techniques to identify the likely defaulters using a dataset that contains data on house loan applications. The strength and stability of a country's fiscal system should be taken into account when considering whether to invest in its economy. It provides information on the health, safety, and living conditions of its residents. Numerous models are used in the procedure known as gradient boosting. The uncertainty of a variable's interpretation makes it uncommon to delete it. On the other hand, even if doing so reduces the overall model accuracy, it is customary practise to exclude variables while fitting logistic regression.

2.3.3 Credit Collectability Prediction of Debtor Candidate Using Dynamic K-Nearest Neighbor Algorithm and Distance and Attribute Weighted

The accuracy, precision, and recall of this procedure are superior than those of other methods. In compared to the domain expert's original order of relevance of the characteristics, the Correlation Attribute Evaluation-adjusted order of importance of the qualities yields a higher recall score of 54.35% for k=5.

The performance outcomes for scenarios 1 and 2 were compared, and it was found that scenario 1 had greater levels of recall, accuracy, and precision. One may argue that utilizing dynamic K-Nearest Neighbor, distance, and attribute weights for weighted attributes is preferable to not using them. When choosing the class label, especially in data with an imbalanced class, it is crucial to choose attribute weights close to other characteristics. This is demonstrated by a considerable reduction in performance outcomes.

One may argue that utilizing dynamic K-Nearest Neighbor, distance, and attribute weights for weighted attributes is preferable to not using them. When choosing the class label, especially in data with an imbalanced class, it is crucial to choose attribute weights close to other characteristics. This is demonstrated by a considerable reduction in performance outcomes.

2.3.4 Exploring the Machine Learning Algorithm for Prediction the Loan Sanctioning Process

The primary goal of the essay is to determine if it is secure to assign the loan to a certain individual. The portions of this essay are as follows: Data collecting, data purification, and performance evaluation make up the first three steps. Experimental studies demonstrate that the Nave Bayes model outperforms alternative approaches for loan forecasting.

It is clear that the Nave Bayes model is extremely successful and yields a superior result when compared to other models after thoroughly studying advantageous features and limitations. It works well, satisfies bankers' needs, and is interoperable with a variety of different systems. Computerized prediction systems include a number of flaws, including content inaccuracies, weight adjustment, and computer issues.

The conditional probability is calculated twice in the scenario with dependent input characteristics, yielding incorrect results. As a consequence, several input characteristics are chosen and processed to produce predictions that are superior than those of the NB model. The suggested loan prediction method is demonstrated in the following pseudocode.

2.3.5 Developing Prediction Model of Loan Risk in Bank Using Data Mining

The loan evaluation process is the series of actions taken to determine whether or not to give a loan to the customer. The "5 C's," which stand for Character (or Credit History), Cash Flow (or Capacity), Collateral, Capitalization, and Conditions, are to be taken into account when a consumer asks a loan. It serves as a useful framework for

analyzing loan requests and determining the credit risk associated with potential creditors.

Weka software has been used to apply the model. Our analysis of the Bayes Net, naive Bayes, and j48 algorithms used in data mining for classification reveals that the j48 method is the most efficient one for classifying loans. The outcome demonstrates that J48 is the greatest algorithm because of how accurate it is and how little its mean absolute error is.

When it comes to identifying which loans are excellent or bad loans, the J48 algorithm performs the best. The training and test sets were varied in size during the trials (80% training 20% test set, 60% training 40% test, and 70% training 30% test), and the trials were repeated many times. This approach aids decision-makers by determining if a transaction will put the bank at risk or not before recommending or rejecting loan applications.

2.3.6 Prediction of Loan Status in Commercial Bank using Machine Learning Classifier

We provide a technique for analyzing credit data based on machine learning classifiers. We integrate Min-Max normalization with the K Nearest Neighbor (K-NN) classifier. The R tool program suite is used to achieve the goal. The most precise important information is provided by the suggested model. The status of loans at commercial banks may be predicted using a machine learning classifier.

This model may be used by lenders to decide whether to approve a loan request. The comparative inquiry also included a number of iteration levels. Iteration level 30 of the k-NN model has much greater accuracy than earlier levels. Make use of this concept to halt the devastating loss of commercial banks.

Due to the fact that it affects several real-time data sets, this issue ought to be fixed for better outcomes. To balance this out, we employ a random sampling method. The default customer records in the training and testing datasets are now spread equally. 50% of the dataset will be utilized for training and 50% for testing, which is the dataset division that will be used in the percentage. An overview of the data divisions is presented in the table below.

2.3.7 Prediction of Loan Defaulter Using Machine Learning

Our primary area of interest is the use of data mining tools to identify and categorize loan defaulters. In this study, the three algorithms Random Forest, KNN, and XG-Boost are compared and contrasted. The same dataset will be utilized for all methods. Using a massive dataset and several machine learning algorithms, it is possible to predict data both inside and outside the training sample. (testing data).

The model is calculated to be used as a guide for the customer and his bank when determining whether to give loans in order to minimize risk and maximize profit. Each aspect that influences loan processing will be decided upon-the-fly in this system, and the same factors will be treated in accordance with their respective weights upon testing of freshly produced data.

Creating an interactive that responds to user input and generates results. The outcomes of these case studies provide insight into methods for precisely predicting loan defaulters. They test the efficacy of two machine learning algorithms, namely Random Forest, KNN, and XG Boost, to predict and categorise student performance using MI techniques. Based on client behavior, this dataset was used in the current research.

2.3.8 Prediction of Modernized Loan Approval System Based on Machine Learning Approach

Three machine learning algorithms are utilized in this study to evaluate how well the data set predicts.

- (a) XG Boost
- (b) Random Forest
- (c) Decision Tree

The benefit of this approach is that it allows us to determine from the facts if a customer fulfils the eligibility requirements by defining the algorithms and outlining specific scenarios. It is feasible to create a system that predicts if a user's loan application will be granted by the bank based on user inputs like salary, address, loan amount, and duration. The prediction accuracy for both datasets is pretty good, according to this study. For instance, when a customer is experiencing a crisis, the algorithm might be unable to foresee the acceptable outcome.

CHAPTER 3

SYSTEM MODEL

3.1 Introduction

Delivering loans is usually the most important bank's major focus. The provision of loans is virtually all banks' primary activity. Most of a firm's assets are directly connected to the revenue it obtained from the mortgages it provided. Placing assets in locations where they are in safe hands is the basic objective of a financial system. There is no assurance that the proposal chosen is the most valid candidate out of all applicants, although several banks and financial organizations today lend money after a drawn-out system of validation and verification. This technology allows us to ascertain the security of a particular application, and it automates the feature evaluation process in its entirety. Loan prediction offers significant advantages to both applicants and bank staff.

This work aims to provide a quick, simple, and practical approach to choosing the most qualified individuals. It might offer the bank certain privileges. Every characteristic involved in loan processing may be automatically assigned a weight by the Loan Prediction System, the same parameters are processed in accordance with their associated weights on fresh test data too. Users can designate a deadline by which the client must declare if their loan request will be acknowledged. By moving on to a specific application, the Loan Prediction System can evaluate it in priority order.

3.2 System Model

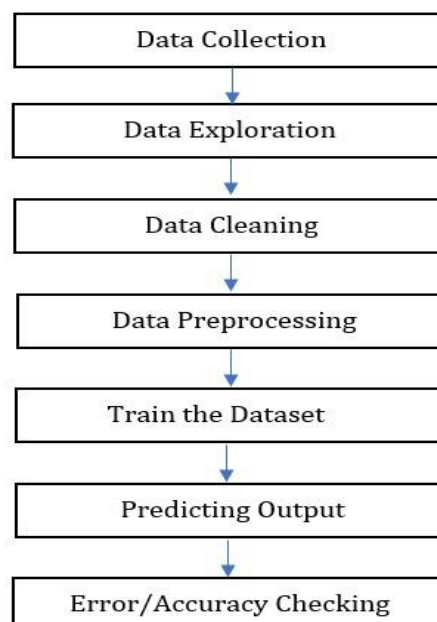


Fig.3.2 System Model

3.2.1 Data Collection

Data collecting is a crucial phase in machine learning (ML), since the standard of the data utilized has a direct impact on the precision and effectiveness of subsequent models. In ML, data collection involves gathering a representative dataset that reflects the real-world problem that the model aims to solve. The following are some considerations for data collection in ML:

- 1. Define the problem:** Before collecting data, it is essential to define the problem that the model aims to solve. This helps to identify the necessary data attributes, the type of data required, and the target variable that the model should predict.
- 2. Determine the data sources:** Data can be sourced from various sources, including existing datasets, web scraping, surveys, sensors, and IoT devices. It is crucial to identify reliable sources that provide relevant and accurate data.
- 3. Select the appropriate data type:** Structured, semi-structured, and unstructured data are all possible. Unstructured data has to be preprocessed before usage and is not well organized. Structured data is easy to search and is highly organized. The type of data that should be used depends on the issue at hand.
- 4. Collect sufficient data:** Depending on the difficulty of the problem being solved is, a model's training data requirements will vary. Collecting more data can improve the model's accuracy, but it also increases the training time and computational resources required.
- 5. Ensure data quality:** Important factors include the quality of the data utilized to train the models. The data must be free from errors, bias, and inconsistencies. Data cleaning and preprocessing techniques can be applied to ensure data quality.
- 6. Ensure data privacy and security:** Privacy and security of data are critical in data collection. It is essential to comply with regulations such as GDPR and CCPA and to protect sensitive data.
- 7. Label the data:** For supervised learning algorithms, labeled data is required for training the model. It is essential to ensure that the labeling is consistent and accurate.

3.2.2 Data exploration

In every machine learning project, data exploration is a crucial phase. Gaining knowledge of the data's structure, distribution, and trends requires analysing and comprehending the data. Data exploration assists in locating possible problems that might affect the quality and dependability of the machine learning model, such as missing values, outliers, and inconsistencies in the data.

In machine learning, the following are some typical methods for data exploration:

1. **Data visualization:** This technique involves generating graphic representations of the data, such as histograms, scatter plots, and heat maps, to aid in understanding the distribution and connections between variables.
2. **Descriptive statistics:** These summarized statistics, such mean, median, mode, and standard deviation, assist explain the data's central tendency and variability.
3. **Correlation analysis:** This process examines how different variables are related to one another. For instance, the coefficient of Pearson's correlation is used to assess the strength and direction of linear correlations between two continuous variables.
4. **Outlier detection:** Involves discovering values that are quite distinct from the rest of the data and may have an influence on the performance of the model.
5. **Dimensionality reduction:** This strategy includes lowering the amount of features in the data by choosing the most significant variables, therefore simplifying the data and enhancing the performance of the model.

3.2.3 Data cleaning

Data cleaning is an essential stage in machine learning (ML), as it helps to increase the precision and dependability of models by ensuring that the training data is precise, comprehensive, and consistent. A few typical methods for cleaning data in ML are the ones listed below:

1. **Handling missing values:** It is a common issue in machine learning. This issue may be resolved using methods like imputation, in which the missing values are filled in based on other values in the dataset.
2. **Handling outliers:** Extreme values that considerably deviate from the dataset's average value are known as outliers. These may either be eliminated or changed into more sensible values to be dealt with.

3. **Data normalization:** To make data easier for algorithms to handle, data is scaled to a standard range by normalization. This may be accomplished using methods like z-score normalization or min-max scaling.
4. **Removing duplicate data:** Since duplicate data might skew the findings of the model, they must be eliminated.
5. **Handling categorical variables:** Algorithms require categorical variables to be transformed into a numerical format, such as gender or colour. This can be accomplished using methods like label encoding or one-hot encoding.
6. **Data type conversion:** In order for the variables in the dataset to be compatible with the algorithms, it is occasionally necessary to modify the data types of the variables.

3.2.4 Data preprocessing

Preparing raw data in a way that ML models can readily understand and analyze is known as data preprocessing, and it is a crucial stage in the Machine Learning (ML) process. Data preprocessing aims to enhance the data's quality and dependability and prepare it for usage by machine learning algorithms.

Data preparation methods often employed include the following.

1. **Data cleaning:** Data cleaning entails locating and fixing data mistakes, inconsistencies, and missing values.
2. **Data normalization:** Normalization of the data entails scaling the data to a standard range so that various characteristics have comparable magnitudes. This enhances the effectiveness of the ML algorithms and prevents some characteristics from outperforming others.
3. **Data encoding:** Data encoding is the process of transforming category data into a numerical format that is simple for machine learning algorithms to interpret. One-hot, label, and binary encoding are a few examples of widely used encoding methods.
4. **Data splitting:** Data splitting is the process of partitioning a dataset into training, validation, and testing datasets. As a result, overfitting is avoided and it is easier to assess how well the ML model performs on new data.
5. **Data augmentation:** Data augmentation entails applying different changes to the current data samples in order to produce new data samples. This method can assist increase the accuracy of the ML model and is especially helpful when the amount of the dataset is minimal.

3.2.5 Training the Dataset

A set of data called a training dataset is used to instruct or train a machine learning algorithm. It is an aspect of the larger dataset that has been gathered or acquired specifically for this activity. A model that can predict outcomes on fresh, unexplored data is created using the training dataset. Typically, target labels and input characteristics are both present in the training dataset. The variables or properties of the input

Target labels are the values that the model is trained to predict, whereas descriptive characteristics are those that characterize the data. In order to accurately predict future data, the machine learning algorithm uses the training dataset to understand the correlations between the input characteristics and target labels. For the machine learning algorithm to learn from, it's critical to make sure the training dataset is representative of the entire dataset and has a sufficient number of different samples. Through this, overfitting may be avoided, a situation in which a model gets overly customized to its training dataset and performs poorly on fresh data.

3.2.6 Predicting Output

Predicting output often refers to a machine learning model's prediction for a certain input. To put it another way, when a machine learning model has been trained using a set of input data and labels, it may be used to predict the label or output for brand-new, unforeseen input data. The prediction made by the model for those new input data is the predicting output.

For example, in a spam email detection model, the predicting output would be whether a given email is spam or not based on the model's prediction for that email. Similarly, in an image recognition model, the predicting output would be the label or class predicted by the model for a given input image.

3.2.7 Error/Accuracy Checking

Error/accuracy checking is a process of verifying the accuracy and correctness of data, information, or calculations. It involves comparing the actual data or results against expected or known values to determine if there are any discrepancies or errors.

In computer science and programming, error checking is an important process that helps to ensure the integrity and reliability of software applications. This includes checking for syntax errors, logic errors, and runtime errors that can cause the software to crash or behave unexpectedly. Accuracy checking is also important in various fields where precision and correctness are critical, such as scientific research, financial analysis, and data analysis.

CHAPTER 4

METHODOLOGY

4.1 Introduction

Bank term deposits are a popular investment option that provide a fixed rate of return over a set period. Accurately predicting whether a customer will be interested in opening a term deposit account can help banks optimize their marketing efforts and improve their customer acquisition rates. One approach to predicting bank term deposit subscriptions is to use machine learning algorithms. By analyzing historical customer data such as age, job, education level, and previous banking activity, machine learning models can identify patterns and make predictions on whether a particular customer is likely to subscribe to a term deposit account.

The benefits of using machine learning for predicting bank term deposits include improved accuracy, reduced marketing costs, and better customer targeting. By understanding which customers are most likely to subscribe to a term deposit account, banks can tailor their marketing campaigns to focus on those individuals and increase the likelihood of a successful subscription. Overall, the use of machine learning for predicting bank term deposits is a powerful tool that can help banks optimize their marketing efforts and improve their overall business performance.

4.2 Tools Required

1. Anaconda: Python and R are two widely used open-source programming languages for scientific computing and data analysis. Anaconda is one such distribution. It includes a package management system and a collection of pre-installed packages and tools for data science, machine learning, and artificial intelligence. Anaconda provides an easy-to-use graphical installer that simplifies the process of installing and managing Python and R packages, making it a popular choice for data scientists and researchers who need to work with complex software stacks. It also includes the Anaconda Navigator, a GUI tool for managing packages and creating environments, and Jupyter Notebook, an interactive web-based environment for working with data and code. Anaconda can be used on Windows, macOS, and Linux operating systems, and it is free to download and use. The Anaconda distribution is maintained and developed by Anaconda, Inc., a company that provides commercial support and services for Anaconda users.

2. Jupyter Notebook: An open-source web tool called Jupyter Notebook enables you to create and share documents with real-time code, equations, visuals, and text. It is compatible with several programming languages, including Python, R, Julia, and others. Cells in the notebook's structure can either hold markdown text or code. By reaching Shift Enter or by selecting the "Run" button from the toolbar, you may execute the code that is now in a cell. The cell behind it will display the code's output. Because it makes it simple for users to share their work and work together, Jupyter Notebook has grown to be a well-liked tool for data analysis, scientific computing, and machine learning. It includes other helpful features as well, such automated coding.

3. Python: Released for the first time in 1991, Python is a high-level, interpreted programming language. It has been well-liked for its simplicity, adaptability, and enormous standard library. It is designed to be simple to understand and write. Python is utilized in a broad range of fields, such as web development, scientific computing, data analysis, automation, and artificial intelligence. Programming styles such as imperative, functional, and object-oriented are all supported. The simplicity of Python makes it a wonderful language for novices to learn and one of its main advantages. Its syntax is lucid and straightforward, making it simple to understand and create code. Additionally, Python has a sizable and helpful community that offers developers a multitude of tools and information. As an open-source language, Python is available for use, modification, and distribution at no cost. Numerous operating systems, including Windows, macOS, Linux, and Unix, are supported by it.

4. Pandas: Pandas is a well-known open-source toolkit for Python that facilitates data manipulation and analysis. It contains data structures like Series (1-dimensional labelled array) and Data Frame (2-dimensional labelled data structure), which are developed on top of the NumPy library to make dealing with data in Python more logical and effective.

Among the essential characteristics of pandas are:

- Data manipulation: Made feasible through a variety of operations offered by Pandas, such as merging, grouping, filtering, and sorting.
- Data cleaning: Pandas provides a number of routines to manage duplication, missing data, and other typical data cleaning chores.
- Data visualization: Pandas has built-in utilities for creating plots and charts that may be used to visualize data.

- Data input/output: Pandas supports a number of file formats, including CSV, Excel, SQL databases, and others, for data input and output.
- The data science community frequently uses Pandas for activities including data preparation and cleaning, exploratory data analysis, and data visualization.

5. Seaborn: On top of the Matplotlib library, Seaborn is a well-known Python data visualization framework. For producing eye-catching statistical visuals, it offers a high-level interface. There are many other visualization options available from Seaborn, including as scatter plots, line plots, bar plots, box plots, violin plots, heatmaps, and more. In order to personalize the look of the visualizations, Seaborn also provides a number of pre-built colour palettes. Seaborn's capacity to construct complex visualizations with just a few lines of code is one of its key features. Users may fine-tune the appearance of their visualizations using a variety of customization tools provided by Seaborn. Additionally, Pandas, a widely-liked toolkit for data processing, works nicely with Seaborn, making it simple to produce visualizations using Panda's structure of data. Overall, the data science community makes extensive use of Seaborn, a potent library for data visualization.

6. NumPy: Working with arrays, matrices, and mathematical operations requires the usage of the Python package NumPy. Numerical Python is what it stands for. High-performance multidimensional array objects and tools are available in NumPy for use with these arrays. Additionally, it contains capabilities for random number generation, Fourier transformation, and linear algebra. The core Python package for scientific computing is called NumPy, and it's frequently used with other libraries like SciPy, Matplotlib, and Pandas. (for data analysis). In scientific and technical applications, its array object is often utilized and offers effective huge dataset storage and management.

7. Matplotlib: With the help of the Python graphing module Matplotlib, users may use Python to build a broad range of static, animated, and interactive visualizations. Line plots, scatter plots, bar plots, histograms, 3D plots, and other visualization-building tools are available. Popular library Matplotlib is used in many different industries, including data science, engineering, and scientific research. It may be included into web apps, scripts, and Jupyter notebooks. With Matplotlib, users may completely customize their visualizations by choosing the colors, labels, axes, legends, and other elements. For studying and using Matplotlib, there is a thorough documentation as well as a wealth of online resources, such as user manuals, examples, and tutorials. It is also accessible with other Python libraries, like Pandas, SciPy, and NumPy, making it a flexible tool for data analysis and visualization.

4.3 Design Steps

Step 1: To load packages in most programming languages, you need to first install them on your system.

Here are some examples of how to load them in different languages:

```
import pandas as pd
import numPy as np          # For mathematical calculations.
import seaborn as sns       # For data visualization.
import matplotlib.pyplot as plt
import seaborn as sn        # For plotting graphs.
get_ipython().run_line_magic('matplotlib', 'inline')
import warnings             # To ignore any warnings.
warnings.filterwarnings("ignore")
from sklearn import metrics
```

Step 2:

Data

- Three CSV files—train, test, and sample submission—have been provided for this issue.
- The model will learn from the train file, which will be used to train the model. Both the target variable and all the independent variables are present.
- All of the independent variables are present in the test file, but not the target variable. The model will be used to forecast the target variable for the test set of data.
- The format in which we must submit our forecasts is shown in the sample submission file.

Reading Data

```
test = pd.read_csv(r"D:\project\test.csv")
train = pd.read_csv(r"D:\project\train.csv")
```

Step 3:

Understanding the Data

The train and test datasets will be examined in this section. Prior to looking at their data kinds, we will first evaluate the features that are present in our data.

```
train.columns
Index(['ID', 'age', 'job', 'marital', 'education', 'default', 'balance', 'housing', 'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays', 'previous', 'poutcome', 'subscribed'],
      dtype='object')

test.columns
Index(['ID', 'age', 'job', 'marital', 'education', 'default', 'balance', 'housing', 'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays', 'previous', 'poutcome'], dtype='object')
```

Step 4:

Print Data types

As we can see, data types come in two formats:

- object: The object format indicates categorical variables. The following categorical variables are included in our dataset: Loan_ID, Gender, Married, Dependents, Education, Self_Employed, Property_Area, and Loan_Status
- int64: It represents the integer variables. This format applies to applicant income.

```
train.dtypes
ID          int64
age         int64
job         object
marital     object
education   object
default     object
balance     int64
housing     object
loan        object
contact     object
day         int64
month       object
duration    int64
campaign    int64
pdays      int64
previous    int64
poutcome    object
subscribed  object
dtype: object
```

Let's see how the dataset is structured.

'train.shape' and 'test.shape' ((31647, 18), (13564, 17))

18 rows and 31647 columns make up the training dataset, whereas 13564 rows and 17 columns make up the test dataset.

Step 5:

To obtain the first n rows, use the head() method. Based on location, this method returns the

object's first n rows. It is helpful for rapidly determining whether the proper type of data is present in

your object.

```
train.head()
```

Step 6:

Target Variable

We'll start by taking a look at the target variable, which is Subscribed. Let's look at the frequency table, percentage distribution, and bar plot for this variable as it is a categorical one. We can find the number of each category in a variable by looking at its frequency table.

```
train['subscribed'].value_counts()
no 27932
yes 3715
```

Identifier: subscribed; data type: int64

```
train['subscribed'].value_counts(normalize=True)
```

```
no 0.882611
yes 0.117389
```

Identifier: subscribed, using the float64 data type

```
train['subscribed'].value_counts().plot.bar()
```

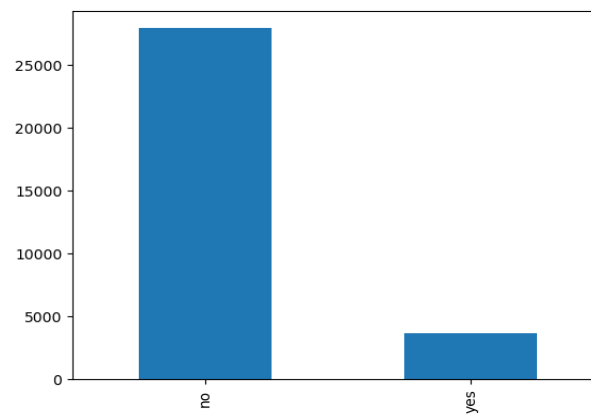


Fig.4.3.1 Target Variable

Step 7:

Independent Variable (Categorical)

```
sn.distplot(train["age"])
```

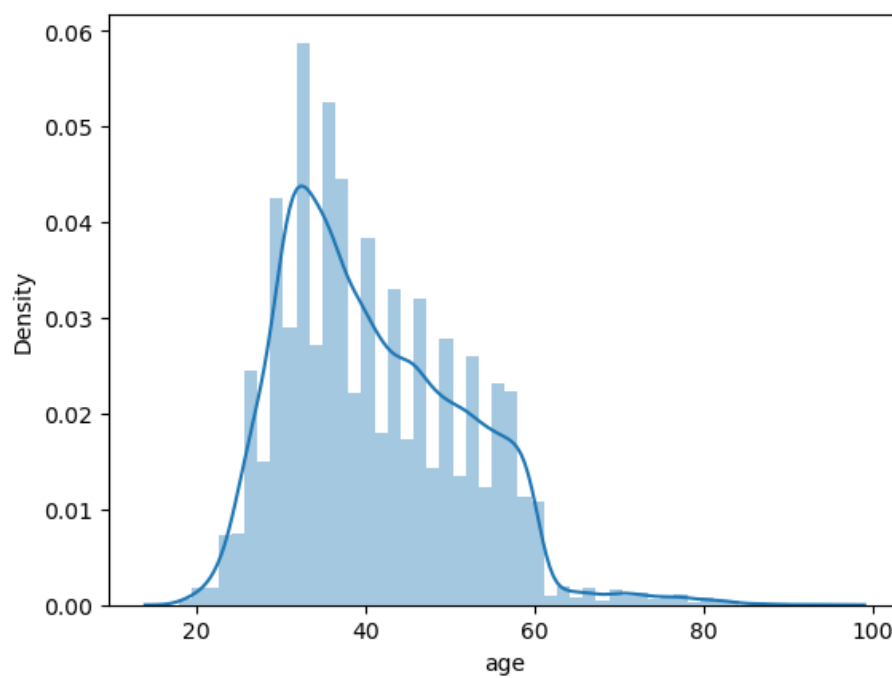


Fig. 4.3.2 Age Graph

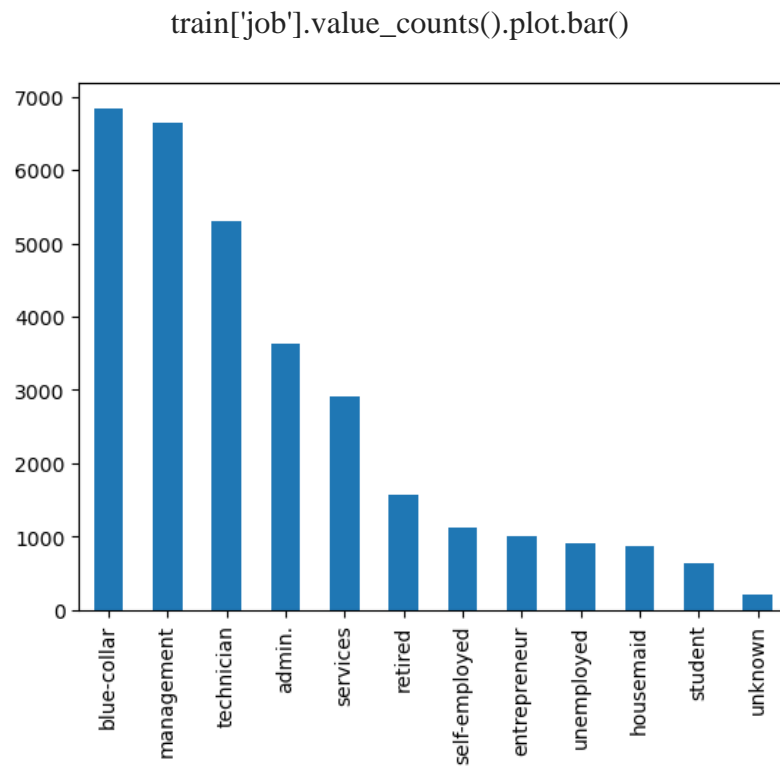


Fig. 4.3.3 Job Graph

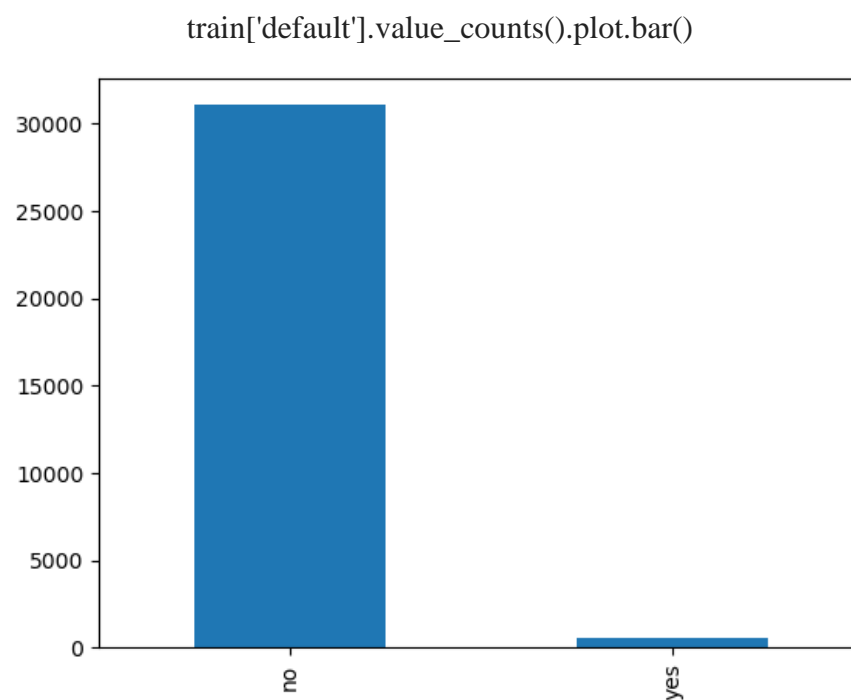


Fig. 4.3.4 Default Graph

Step 8:

Independent Variable (Ordinal)

```
print(pd.crosstab(train['job'],train['subscribed']))  
job=pd.crosstab(train['job'],train['subscribed'])  
job.div(job.sum(1).astype(float), axis=0).plot(kind="bar", stacked=True, figsize=(8,8))  
plt.xlabel('Job')  
plt.ylabel('Percentage')
```

	subscribed	no	yes
job			
admin.	3179	452	
blue-collar	6353	489	
entrepreneur	923	85	
housemaid	795	79	
management	5716	923	
retired	1212	362	
self-employed	983	140	
services	2649	254	
student	453	182	
technician	4713	594	
unemployed	776	129	
unknown	180	26	

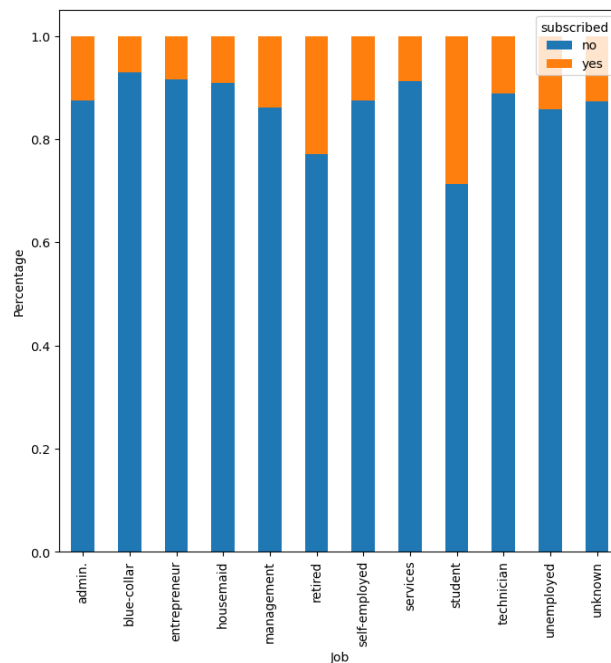


Fig. 4.3.5 Subscribed Graph Between Job and Percentage

```
print(pd.crosstab(train['default'],train['subscribed']))
default=pd. crosstab(train['default'],train['subscribed'])
default.div(default.sum(1).astype(float), axis=0).plot(kind="bar", stacked=True, figsize=(
8,8))
plt.xlabel('default')
plt.ylabel('Percentage')
```

subscribed	no	yes
default		
no	27388	3674
yes	544	41

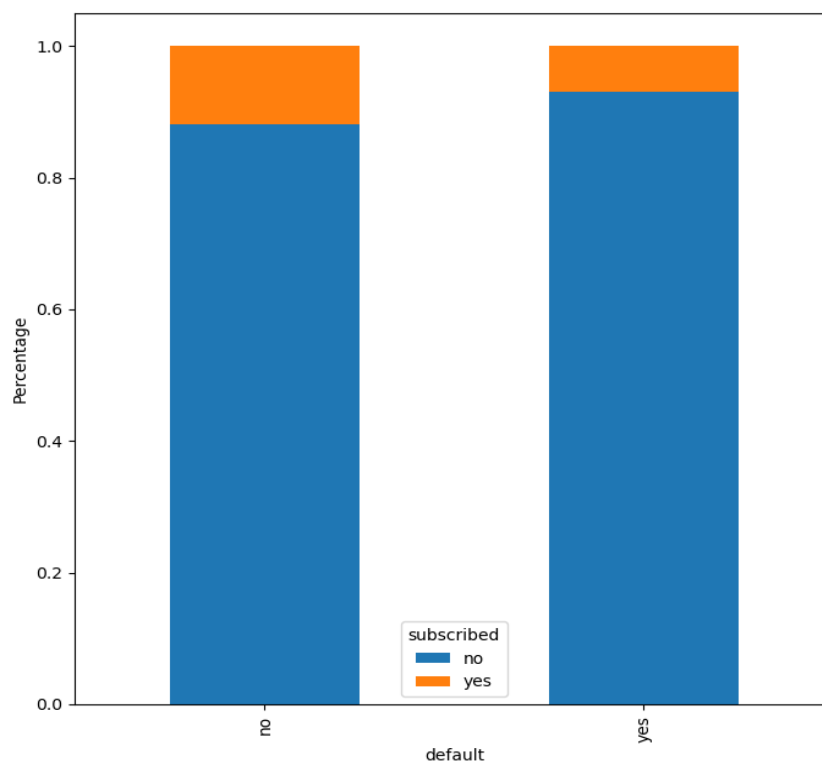


Fig.4.3.6 Subscribed Graph Between Default and Percentage

Step 9:

Let's now examine the relationship between all of the numerical variables. The correlation will be shown using the heat map. Data is visualised using heatmaps by changing the colour. The factors whose colours are darker indicate more correlation.

```
train['subscribed'].replace("no," 0, "inplace="True")
train['subscribed'].replace("yes," "1, inplace="True"
```

```
corr = train.corr()
mask = np.array(corr)
mask[np.tril_indices_from(mask)] = False
fig,ax=plt.subplots()
fig.set_size_inches(20,10)
sn.heatmap(corr, mask=mask,vmax=.9, square=True,annot=True, cmap="YlG
nBu")
```

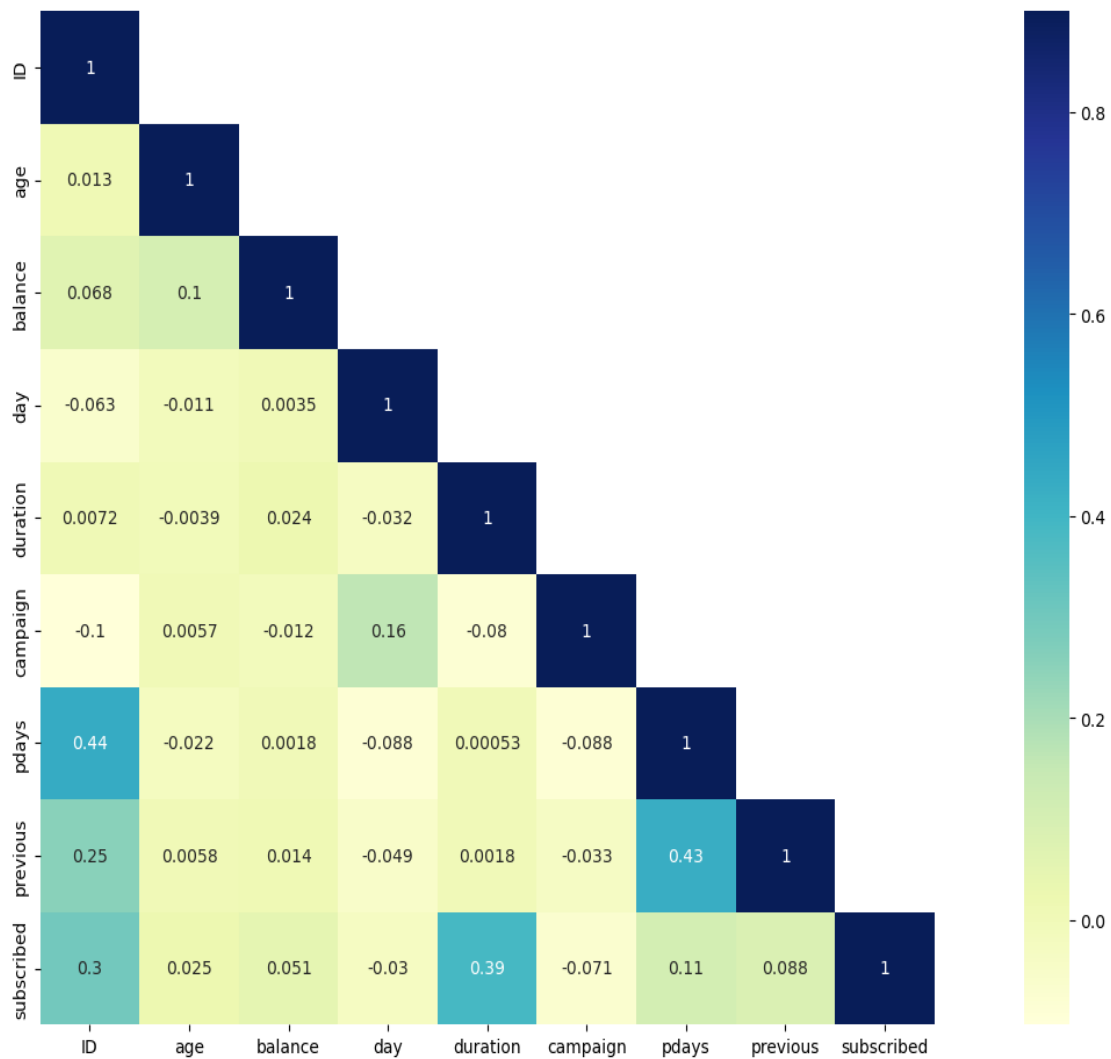


Fig.4.3.7 Correlation Heat Map

Step 10:

Missing Value and Outlier Treatment

We may now impute the missing values and address the outliers after thoroughly examining all the variables in our data because missing data and outliers might negatively impact model performance

```
train.isnull().sum()
```

```
ID          0
age         0
job         0
marital     0
education   0
default     0
balance     0
housing     0
loan        0
contact     0
day         0
month       0
duration    0
campaign    0
pdays     0
previous    0
poutcome    0
subscribed  0
dtype: int64
```

Step 11:

Evaluation Metrics for Classification

A classification model's performance may be assessed using a variety of assessment indicators. Several frequently employed categorization assessment metrics are as follows:

- 1. Accuracy:** The proportion of properly identified samples to all samples is measured using this assessment metric, which is the simplest. The formula for calculating it is $(TP+TN)/(TP+TN+FP+FN)$, where TP stands for the number of true positives, TN for the number of true negatives, FP for the number of false positives, and FN for the number of false negatives.

		Predicted	
		Good	Bad
Actual	Good	True Positive [d]	False Negative [c]
	Bad	False Positive [b]	True Negative [a]

Fig.4.3.8 Accuracy

- True Positive – Targets that have been projected as true but are in fact true are referred to be True Positives.
- True Negative – Targets that we have forecasted as false but which are really true.
- False Positive – Targets that we forecasted as positive but are really untrue.
- False Negative – Targets that we forecasted as false but are really true.

2. Precision: Precision is the ratio of real positive results to all expected positive results. It has the following definition:

$$TP/(TP+FP)$$

3. Recall: Recall calculates the ratio of genuine positives to all other positive results. It has the following definition:

$$TP/(TP+FN)$$

4. F1 Score: The harmonic mean of recall and accuracy is the F1 score. It is a balanced measurement that considers both recall and accuracy.

It is characterised as:

$$2*(Precision*Recall)/(Precision+Recall).$$

Step 12:

Model Building

Logistic Regression:

Logistic Regression: Let's use this method to predict the desired variable using our initial model. To forecast binary outcomes, we will start with logistic regression.

- An algorithm for classifying data is logistic regression. A set of independent variables are used to predict a binary result (1/0, Yes/No, True/False).
- The Logit function is estimated using logistic regression. The logit function is nothing more than a log of the event's chances.

```
target = train['subscribed']
```

```
train = train.drop('subscribed',1)
```

For the categorical variables, we will now create dummy variables. Dummy variables make categorical variables more simpler to measure and compare by converting them to a sequence of 0 and 1. Let's first examine the dummies' process:

mrngThink about the "Gender" parameter. Male and female courses are available.

- The "Gender" variable will become two variables (Gender_Male and Gender_Female), one for each class, i.e., Male and Female, if we apply dummies to it.
- If the gender is Female, Gender_Male will have a value of 0, and if the gender is Male, it will have a value of 1.

```
pd.get_dummies = train(train)
```

We will now use the training dataset to train the model and provide predictions for the test dataset. Can we verify these forecasts, though? Our train dataset may be split into the train and validation components as one method of accomplishing this. We may train the model on this training portion and use that information to create predictions for the validation portion. By having the accurate predictions for the validation phase, we may validate our forecasts in this manner. (which we do not have for the test dataset). Our train dataset will be divided using the

```
train_test_split algorithm from sklearn. In order to get started, let's import train_test_split
from sklearn.model_selection. import train_test_split (train, target, test_size = 0.2,
random_state=12) X_train, X_val, Y_train, Y_val
```

There are training and validation sections of the dataset. Let's import the accuracy_score and the logistic regression model from Sklearn and fit it.

```
import from sklearn.linear_model LogisticRegression
```

```
LogisticRegression is defined as lreg = lreg.fit(X_train,y_train).
```

```
From the sklearn.metrics function LogisticRegression() prediction =
lreg.predict(X_val) the accuracy_score import
```

Let's determine the precision of our forecasts using. So first, let us import train_test_split.

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_val, y_train, y_val = train_test_split(train, target, test_size = 0.2,
random_state=12)
```

The dataset has been divided into training and validation parts. Let us import Logistic Regression and accuracy_score from sklearn and fit the logistic regression model.

```
from sklearn.linear_model import LogisticRegression
```

```
lreg = LogisticRegression()
```

```
lreg.fit(X_train,y_train)
```

```
LogisticRegression()
```

```
prediction = lreg.predict(X_val)
```

```
from sklearn.metrics import accuracy_score
```

Let us calculate how accurate our predictions are by calculating the accuracy.

```
accuracy_score(y_val, prediction)
```

```
0.8873617693522907
```

Decision tree:

A decision tree is a particular kind of supervised learning algorithm (with a predefined goal variable) that is frequently employed in classification tasks. As part of this method, we divide the population or sample into two or more homogenous groups (or sub-populations) depending on the most important a splitter or differentiator for input variables. Multiple techniques are used by decision trees to determine whether to divide a node into two or more sub-nodes. The homogeneity of the resulting sub-nodes is increased by the development of sub-nodes. With regard to the target variable, we may say that the purity of the node rises.

```
from sklearn.tree import DecisionTreeClassifier

DC = DecisionTreeClassifier(max_depth=4, random_state=0)
DC.fit(X_train,y_train)
DecisionTreeClassifier(max_depth=4, random_state=0)
predict = DC.predict(X_val)
```

Let us calculate how accurate our predictions are by calculating the accuracy.

```
accuracy_score(y_val, predict)

0.9042654028436019
```

Random Forest:

Random Forest is a tree-based bootstrapping approach that combines a certain number of weak learners (decision trees) to create an effective prediction model. A decision tree model is created for each individual learner using a random sample of rows and a few randomly selected factors. All of the guesses offered by the many students may operate as the final forecast. The mean of every forecast may serve as the final prediction in a regression issue.

```
from sklearn.ensemble import RandomForestClassifier

RFC = RandomForestClassifier(n_estimators=100, n_jobs=1)
RFC.fit(X_train, y_train)
RandomForestClassifier(n_jobs=1)
pred_rfc = RFC.predict(X_val)
RFC.score(X_val, predict)

0.9423380726698263
```

Let us calculate how accurate our predictions are by calculating the accuracy.

```
accuracy_score( y_val, pred_rfc)

0.9078988941548183
```

CHAPTER 5

RESULTS

We shall decide whether or not to approve the loan application based on the information provided by the applicant. These factors, which the applicant is required to supply, can be used by the model to determine whether or not the loan will be authorised. Every train-test split for supervised learning rules already exists. To create a model and assess its efficacy, the full dataset is split into two datasets known as Train and Check and Independent Options and Dependent Options. (only in supervised). A machine learning model is trained on a dataset before being utilised to perform learning tasks. Test Dataset to measure the effectiveness of a machine learning model.

1. **Accuracy:** The proportion of correctly identified samples to the total number of samples is the most fundamental evaluation statistic. Its formula is $(TP+TN)/(TP+TN+FP+FN)$, where TP stands for "true positives," TN for "true negatives," FP for "false positives," and FN for "false negatives."
2. **Precision:** Accuracy is defined as the ratio of genuine positives to all anticipated positives. Its formula is $TP/(TP+FP)$.
3. **Recall:** Recall calculates the ratio of genuine positives to all other positive results. Its formula is $TP/(TP+FN)$.
4. **F1 Score:** The harmonic mean of recall and accuracy is the F1 score. It is a balanced measurement that considers both recall and accuracy. The formula for this is $2*(Precision*Recall)/(Precision+Recall)$

$Accuracy = (True\ Positives + True\ Negatives) / Total\ Sample$

Our accuracy was 0.91.

$Precision\ exactness = (Number\ of\ True\ Positive) / (True\ Positive + False\ Positive)$

Our exact score was 0.93.

$Recall = (True\ Positives) / (True\ Positive + False\ Negative)$

Our Recall score was 0.97.

$F1\ Score = 2 / ((1/Precision) + (1/Recall))$

Our F1 Score was 0.95.

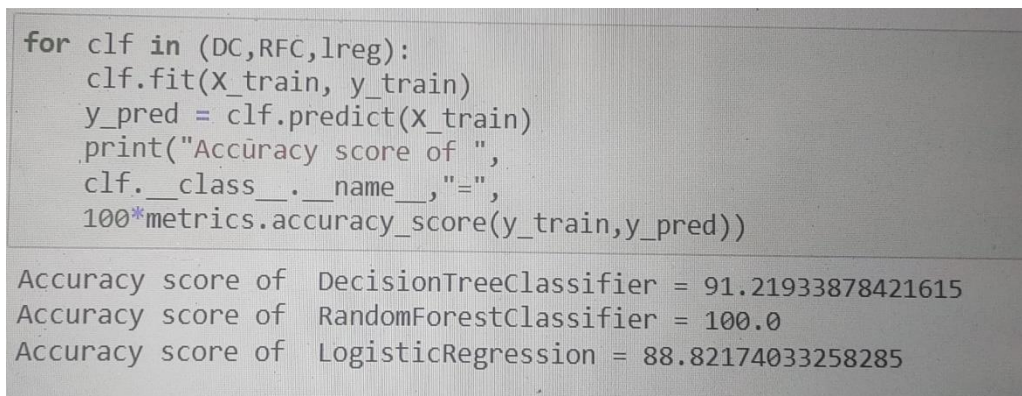
Accuracy score of Decision Tree Classifier = 91.21933878421615

Accuracy score of Random Forest Classifier = 100.0

Accuracy score of Logistic Regression = 88.82174033258285

Training Accuracy:

```
for clf in (DC,RFC,lreg):
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_train)
    print("Accuracy score of ",
          clf.__class__.__name__,"=",
          100*metrics.accuracy_score(y_train,y_pred))
```



```
for clf in (DC,RFC,lreg):
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_train)
    print("Accuracy score of ",
          clf.__class__.__name__,"=",
          100*metrics.accuracy_score(y_train,y_pred))
```

Accuracy score of DecisionTreeClassifier = 91.21933878421615
Accuracy score of RandomForestClassifier = 100.0
Accuracy score of LogisticRegression = 88.82174033258285

Fig.5.1 Training Accuracy

Accuracy score of Decision Tree Classifier = 91.21933878421615

Accuracy score of Random Forest Classifier = 100.0

Accuracy score of Logistic Regression = 88.82174033258285

Testing Accuracy:

```
for clf in (DC,RFC,lreg):
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_val)
    print("Accuracy score of ",
          clf.__class__.__name__,"=",
          100*metrics.accuracy_score(y_val,y_pred))
```

```
for clf in (DC,RFC,lreg):
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_val)
    print("Accuracy score of ",
          clf.__class__.__name__,"=",
          100*metrics.accuracy_score(y_val,y_pred))

Accuracy score of DecisionTreeClassifier = 90.42654028436019
Accuracy score of RandomForestClassifier = 90.44233807266983
Accuracy score of LogisticRegression = 88.73617693522907
```

Fig.5.2 Testing Accuracy

Accuracy score of Decision Tree Classifier = 90.42654028436019

Accuracy score of Random Forest Classifier = 90.44233807266983

Accuracy score of Logistic Regression = 88.73617693522907

CHAPTER 6

ADVANTAGES

Improved Accuracy: Machine learning algorithms are created to learn from prior data and generate predictions based on patterns and trends. As a result, there is a lower chance of fraudulent or default loans since the loan confirmation procedure may be more precise.

Speed and Efficiency: Machine learning algorithms are capable of processing enormous volumes of data fast and effectively. This implies that loan applications may be verified and processed more quickly, enabling clients to get their loans in less time.

Cost Savings: By employing machine learning to automate the loan confirmation process, financial institutions may cut costs like manpower and training that are related to manual loan underwriting.

Better Customer Experience: Customers might have a better experience when applying for loans if loan confirmation times are quicker and loan choices are made with higher accuracy. This might benefit financial companies' reputation-building efforts and client retention rates.

Predictive Analytics: Machine learning algorithms are capable of analysing data from a variety of sources to find trends and patterns that might assist financial organisations in making better loan choices. This can result in better-informed lending decisions and a decreased probability of default on loans.

Overall, predicting the confirmation of prospective loans by machine learning may enhance accuracy, speed, efficiency, cost savings, and the client experience while also enabling predictive analytics and better lending decisions.

CHAPTER 7

APPLICATIONS

Credit Scoring: To assess a borrower's creditworthiness and the risk that they would default, machine learning algorithms may be used to examine their credit history, income, and other pertinent data. This may enable lenders to decide whether to approve or reject loan applications with greater knowledge.

Fraud Detection: Algorithms trained in machine learning can spot the patterns of fraudulent loan applications, such as forged documents or identity theft. As a result, there is a lower chance of financial loss and financial institutions won't approve fake loans.

Loan Portfolio Management: Machine learning algorithms are capable of analysing the loan portfolio of a financial institution to spot trends and patterns that can point to a higher risk of default. This might help lenders in managing their loan portfolios pro-actively and lowering the danger of suffering losses.

Customer Segmentation: Machine learning algorithms may examine customer data to pinpoint several borrower categories with certain traits and borrowing tendencies. Financial organisations may use this to better cater their lending policies to particular clientele groups and boost client satisfaction.

Automated Underwriting: Machine learning algorithms may be used to automate the loan underwriting process, eliminating the need for manual inspection and enhancing the speed and effectiveness of loan approval. Financial organisations may do this to improve client satisfaction and save time and money.

Overall, machine learning's ability to predict the confirmation of prospective loans offers a wide range of real-world uses for lenders, including automated underwriting, automated credit rating, fraud detection, loan portfolio management, and customer segmentation. These tools can assist loan choices made by financial firms, lower the risk of financial losses, and improve the customer experience.

CHAPTER 8

CONCLUSION

3.3 Conclusion:

These models are used by both the banking sector and anybody wishing to apply for a loan. The management of banks will benefit greatly from it. Fraud is significantly reduced throughout the loan approval process, according to data analysis. Based on user inputs like address and salary, this system may be built to predict if a user's loan application would be approved by the bank. Actually, lending money is a bank's main business. The majority of a bank's primary activity is loan providing. The money a firm made from the loan it took out determines the majority of its assets. The primary objective of a financial system is to place assets in secure locations.

Despite the fact that many financial institutions and organizations currently grant loans following a protracted process of confirmation and validation, there is no assurance that the application chosen is the most deserving applicant out of all applicants. We can evaluate the security of a specific application, the length of the loan, and the loan amount thanks to the automation provided by our technology.

CHAPTER 9

FUTURE SCOPE

Creating New Accounts: The program allows users to open either a current account or a savings account, two different kinds of accounts. It enables the customer to open and utilize these accounts without physically visiting the bank.

Depositing Money: As the globe moves away from the common usage of paper money, this application will make it easy to deposit money or transfer it from one bank to another by just clicking a few buttons.

Withdrawing Money: Using the program, requests for money transfers are also possible. The Account Holder List function is accessible to administrators. An account holder list is available to the administrator.

Account Holder List: This is a feature for the admin. The admin can view the list of all the account holders.

Balance Enquiry: The customer can check their balance via this application.

Changing Passwords/PIN: The customer can easily change the passwords and pin numbers using the application.

Closing: The customer can close their accounts too using this application.

Check Your Balance: The user may use this application to check their balance. Customers may easily alter their passwords and PINs with the aid of the program. Customers can also close their accounts using this application.

REFERENCES

- [1] Mohammad Ahmad Sheikh, Amit Kumar Goel, Tapas Kumar- “Approach for Prediction of Loan Approval using Machine Learning Algorithm”, (ICESC),IEEE-2020.
- [2] Bhoomi Patel, Harshal Patil, Jovita Hembram- “Loan Default Forecasting using Data Mining”, (INCET)-2020 .
- [3] Punitara Ruangthong and Saichon Jaiyen-“Bank Direct Marketing Analysis of Asymmetric Information Based on Machine Learning”,(JCSSE)- 2015.
- [4] Justice Frempong, Manoj Jayabalan-“Predicting Customer Response to Bank Direct Telemarketing Campaign”,(ICETT)-2017.
- [5] Aboobyda Jafar Hamid and Tarig Mohammed Ahmed- “DEVELOPING PREDICTION MODEL OF LOAN RISK IN BANKS USING DATA MINING”,(MLAIJ)-2016.
- [6] Tiara Fajrin,Ragil Saputra, Indra Waspada-“Credit Collectibility Prediction of Debtor Candidate Using Dynamic K-Nearest Neighbor Algorithm and Distance and Attribute Weighted”,(ICICoS)-2018 IEEE.
- [7] G. Arutjothi,Dr.C.Senthamarai- “Prediction of Loan Status in Commercial Bank using Machine Learning Classifier”, (ICISS)-2017.
- [8] E. Chandra Blessie, R. Rekha-“Exploring the Machine Learning Algorithm for Prediction the Loan Sanctioning Process”,(IJITEE)-2019.
- [9] Yash Divate, Prashant Rana, Pratik Chavan-“Loan Approval Prediction Using Machine Learning”,(IRJET)-2021.
- [10] Vishal Singh, Ayushman Yadav, Rajat Awasthi- “Prediction of Modernized Loan Approval System Based on Machine Learning Approach”,(CONIT)-2021.

[11] Ce'dric Ste'phane Te'kouabou Koume'tio, Walid Cherif, Silkan Hassan- "Optimizing the prediction of telemarketing target calls by a classification technique", (IEEE)-2018.

[12] Kadam Apurva, Pawar Harshada, Phapale Shweta- "Prediction of Loan Defaulter Using Machine Learning", (IJCRT)-2021.

[13] S.Vimala, K.C.Sharmili - "Prediction of Loan Risk using Naive Bayes and Support Vector Machine", (ICACT)- 2018.

ANTICIPATING THE FUTURISTIC LOANS CONFIRMATION USING MACHINE LEARNING

ORIGINALITY REPORT

24%

SIMILARITY INDEX

15%

INTERNET SOURCES

8%

PUBLICATIONS

18%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Visvesvaraya Technological University

Student Paper

2%

2

www.analyticsvidhya.com

Internet Source

2%

3

Submitted to Edith Cowan University

Student Paper

1%

4

www.geeksforgeeks.org

Internet Source

1%

5

Submitted to Liverpool John Moores University

Student Paper

1%

6

Submitted to SASTRA University

Student Paper

1%

7

Submitted to Harrisburg University of Science and Technology

Student Paper

1%

8

Puneeth B. R, Ashwitha K, Arhath Kumar, Balachandra Rao, Preethi Salian K, Supravi A

1%