# LOAN PRICE PREDICTION USING MACHINE LEARNING

**A PROJECT REPORT**

***Submitted by***


| | |
|---|---|
| **VINITH. V** | **[513220104009]** |
| **RAMESHKUMAR. P** | **[513220104310]** |
| **SABARI. I** | **[513220104311]** |
| **GOKUL. V** | **[513220104306]** |


***In partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**




**THIRUMALAI ENGINEERING COLLEGE, KANCHIPURAM**

**ANNA UNIVERSITY: CHENNAI – 600 025**


**MAY 2024**

**ANNA UNIVERSITY: CHENNAI 600 025**

# BONAFIDE CERTIFICATE

Certificate that this project report titled "**LOAN PRICE PREDICTION USING MACHINE LEARNING**" is the bonafide work of "**VINITH. V [513220104009], RAMESHKUMAR. P [513220104310], SABARI. I [513220104311], GOKUL.V [513220104306]**" who Carried out the project work under my supervision.

**SIGNATURE OF HOD**

**V. VIJAYABHASKAR M.C.A., M.Tech.,**

**HEAD OF THE DEPARTMENT,**

Associate Professor,

Department of CSE,

Thirumalai Engineering College,

Kanchipuram – 631 551.

**SIGNATURE OF SUPERVISOR**

**D. JAKULIN SHARMI M.E.,**

**SUPERVISOR,**

Assistant Professor,

Department of CSE,

Thirumalai Engineering College,

Kanchipuram – 631 551.

Submitted for the Project Viva Voce held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

## ACKNOWLEDGEMENT

I profoundly thank our **Chairman and trust members of Kanchipuram Educational Trust** for providing adequate facilities.

I would like to express my hearty thanks to our respectable Principal. Incharge **Mr.T.MohanRaj M.Tech.,** for allowing us to have the extensive use of our colleges facilities to our colleges facilities to have precious advice regarding the project.

I extend our thanks to Associate Professor **Mr.V.VIJAYABHASKAR M.C.A., M.Tech., Head of the Department, Computer Science And Engineering** for this precious advice regarding the project.

I would like to express my deep and unbounded gratefulness to my project Guide **Mrs.D.JAKULIN SHARMI M.E., Department of Computer Science And Engineering**, for her valuable guidance and encouragement throughout the project. She has been a constant source of inspiration and has provided the precious suggestion throughout this project.

I thank all facilities and supporting staff for the help they extended in completing this project. I also express my sincere thanks to our parents, and all my friends for their continuous support.

# TABLE OF CONTENTS

# ABSTRACT

In the modern financial landscape, the efficient and accurate pricing of loans is critical for both lenders and borrowers. Machine Learning (ML) techniques offer a promising avenue for enhancing loan pricing models, enabling lenders to better assess risk and optimize profitability while providing borrowers with fair and transparent pricing.

This paper explores the application of ML algorithms in predicting loan prices, leveraging a diverse set of features such as borrower characteristics, credit history, economic indicators, and market trends. Various supervised learning models including regression, ensemble methods, and deep learning architectures are employed to capture complex patterns and relationships inherent in loan data.

The methodology involves data preprocessing, feature engineering, model selection, and performance evaluation using appropriate metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared ($R^2$) score. Additionally, techniques for handling imbalanced datasets and addressing interpretability challenges are discussed.

Through empirical experiments on real-world loan datasets, the efficacy of ML-based loan pricing models is demonstrated, showcasing their ability to outperform traditional pricing approaches in terms of accuracy and robustness. Furthermore, the paper discusses the practical implications and potential benefits of deploying these models in lending institutions, including improved risk management, enhanced decision-making, and increased market competitiveness.

# LIST OF ABBREVATIONS

| ACRONYM | ABBREVATIONS |
|---|---|
| WHO | WORLD HEALTH ORGANIZATION |
| NLP | NATURAL LANGUAGE PROCESS |
| DS | DATA SCIENCE |
| EDA | EXPLORATORY DATA ANALYSIS |
| CSV | COMMA SEPERATE VALUE |
| KNN | K-NEAREST NEIGHBOR |
| ROC | RECEIVER OPERATING CHARACTER |
| API | APPLICATION PROGRAMMABLE INTERFACE |
| NOSQL | NOT ONLY SQL |
| VOC | VARIENCES OF CONCERN |
| SVM | SUPPORT VECTOR MECHINE |
| BDV | BIG DATA VISUALIZATION |

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

## 1.1 INTRODUCTION

In today's financial landscape, the ability to accurately predict loan prices is crucial for various stakeholders, including financial institutions, investors, and borrowers. Machine Learning (ML) techniques have emerged as powerful tools for analyzing vast amounts of data and extracting meaningful insights to make informed decisions in the realm of loan pricing.

Machine learning models leverage historical loan data, including borrower information, credit scores, loan terms, economic indicators, and market trends, to predict the price or interest rate of future loans. By identifying patterns and relationships within this data, ML algorithms can generate predictions with remarkable accuracy, enabling lenders to set competitive interest rates, optimize risk management strategies, and enhance profitability.

Developing a prediction model for loan default involves collecting historical loan data, preprocessing it by handling missing values and encoding variables, and selecting relevant features like credit scores and employment history. Machine learning algorithms such as XGBoost in Python are then trained on this data to predict default risk.

Banks Loans have become an important source of external financing for firms and households due financial constraints to develop firms and business. The most profit for the bank comes only from loans however the increase of loan lending is associated with a number of risks, such as risk of defaulting or credit risk, which is linked to the inability of the borrower to pay back the loan at the agreed time and

conditions. Banks have limited goods, so it is essential to choose the right applicant who repays the loan within the time limits. Selecting the right candidate is the main responsibility of the bank. When process is done manually many problems arise in choosing the right person for the approval.

Bank should choose correct one otherwise the bank has to face financial trouble and lack of profits. Banks aim is to invest their asset in the safe hands. If a bank is providing the loan to a person the bank should think about for what purpose they are taking, will they repay the amount they took, are their documents are valid or not, etc. Thus we need some machine learning algorithms which chooses the applicant automatically. These machine learning models supports both the employee and applicants.

The primary purpose of this model is to find the right one and reduce the selection time for choosing. There are several ways to predict but data mining is important to note that because it takes the previous records of customers and train the system to predict the approval.

Some of the methods used here to predict are Naïve bayes, Logistic Regression, Support Vector Machine, Classification, Random Forest. So based on the accuracy of these used algorithms we can predict the loan approval easily. Based on the machine learning we have two types of datasets, one is training dataset and the other is test dataset. Here every model will works on some variables and tries to give us the result whether to approve the loan to a person or not. The core objective of this paper is to create a less complex system for prediction of loan model.

## 1.2 PROBLEM STATEMENT

In the realm of financial services, accurately predicting loan prices is a critical task for lenders, investors, and borrowers alike. However, traditional methods of loan pricing often rely on simplistic models that may not fully capture the complexity of underlying risk factors and market dynamics.

As a result, there is a growing demand for advanced predictive analytics techniques, particularly those based on Machine Learning (ML), to enhance the accuracy and efficiency of loan price prediction.

The problem at hand involves developing a robust ML-based framework for predicting loan prices with a high degree of precision and reliability. This framework must address several key challenges:

**Data Complexity:** Loan pricing involves a multitude of factors, including borrower characteristics, credit history, loan terms, economic indicators, and market trends. Managing and processing such diverse and voluminous data sources present significant challenges in terms of data integration, cleansing, and feature engineering.

**Model Complexity:** The relationship between loan features and pricing is often nonlinear and dynamic, requiring sophisticated ML models capable of capturing intricate patterns and interactions within the data. Building models that can effectively handle this complexity while avoiding overfitting and maintaining interpretability is a major challenge.

**Risk Assessment:** Loan pricing inherently involves assessing and mitigating various types of risks, including credit risk, market risk, and operational risk.

**Generalization and Adaptability:** ML models trained on historical data must demonstrate strong generalization capabilities to make accurate predictions on unseen data. Moreover, these models should be adaptable to changing market conditions and regulatory environments to maintain relevance and effectiveness over time.

**Ethical and Responsible Lending:** While ML-based loan pricing models offer significant predictive power, there are ethical considerations regarding fairness, transparency, and bias mitigation. Ensuring that the models do not perpetuate or exacerbate existing disparities in lending practices is paramount to promoting inclusive and responsible lending.

By developing innovative ML-based solutions that effectively tackle these challenges, financial institutions can optimize their loan pricing strategies, improve risk management practices, and ultimately enhance the overall efficiency and stability of the lending ecosystem.

## 1.3 OBJECTIVE OF PROJECT

The objectives of using Machine Learning (ML) for loan price prediction are multifaceted, aimed at enhancing the efficiency, accuracy, and risk management capabilities of lending institutions. Here are the primary objectives:

**Enhanced Risk Assessment:** ML models enable lenders to conduct more comprehensive risk assessments by analyzing a wide range of borrower attributes and financial data. By accurately quantifying credit risk, lenders can make more informed decisions regarding loan pricing, loan terms, and risk mitigation strategies.

**Improved Pricing Precision:** ML algorithms can identify complex patterns and relationships within loan data that traditional pricing models may overlook. By leveraging advanced statistical techniques and predictive analytics, ML models can generate more precise loan pricing estimates tailored to individual borrower profiles and market conditions.

**Operational Efficiency:** Automation of loan pricing using ML streamlines the lending process, reducing manual effort and administrative costs associated with traditional underwriting methods. This leads to faster loan approvals, improved customer experience, and greater operational efficiency for lending institutions.

**Dynamic Pricing Adaptation:** ML models enable lenders to dynamically adjust loan pricing in response to changing market conditions, regulatory requirements, and borrower risk profiles. By continuously analyzing new data and market trends, ML models can optimize pricing strategies to maximize profitability while remaining competitive in the marketplace.

**Portfolio Optimization:** ML-based loan pricing facilitates portfolio optimization by identifying and capitalizing on profitable lending opportunities while managing risk exposure. By optimizing the composition of loan portfolios based on risk-return profiles, lenders can achieve better balance and diversification, leading to improved portfolio performance and risk management.

**Customer-Centric Approach:** ML-driven loan pricing allows lenders to offer personalized pricing terms tailored to individual borrower preferences and financial circumstances. By considering factors such as credit history, income level, and borrowing behavior, lenders can enhance customer satisfaction and loyalty while maintaining profitability.

**Regulatory Compliance:** ML models can help lenders ensure compliance with regulatory requirements and industry standards governing loan pricing practices. By incorporating regulatory constraints and risk factors into pricing models, lenders can mitigate compliance risks and avoid penalties associated with non-compliance.

In summary, the objectives of loan price prediction using ML include enhancing risk assessment capabilities, improving pricing precision, increasing operational efficiency, adapting to dynamic market conditions, optimizing portfolio performance, fostering customer-centricity, and ensuring regulatory compliance. By achieving these objectives, lending institutions can make more informed, profitable, and responsible lending decisions while effectively managing risk and meeting customer needs.

## 1.4 SCOPE OF THE PROJECT

The scope of loan price prediction using Machine Learning (ML) encompasses various aspects of the lending process, from risk assessment to pricing optimization and portfolio management. Here are some key areas within this scope:

**Risk Assessment:** ML models can analyze borrower data to assess credit risk more accurately. By considering factors such as credit history, income, employment status, and debt-to-income ratio, ML algorithms can identify borrowers who are more likely to default on their loans, enabling lenders to adjust pricing or impose stricter terms accordingly.

**Pricing Optimization:** ML techniques allow lenders to optimize loan pricing based on borrower risk profiles and market conditions. By analyzing historical data and market trends, ML models can identify optimal pricing strategies that balance profitability with borrower attractiveness and competitive positioning.

**Dynamic Pricing:** ML models enable lenders to implement dynamic pricing strategies that adapt to changing market conditions and borrower risk profiles in real-time. By continuously monitoring data and market trends, ML algorithms can adjust loan pricing dynamically to optimize profitability and competitiveness.

**Portfolio Management:** ML-based loan price prediction can help lenders optimize their loan portfolios by identifying and managing risk exposure across different segments. By analyzing borrower characteristics, loan performance, and market dynamics, ML models can assist lenders in diversifying their portfolios and optimizing risk-adjusted returns.

**Customer Segmentation:** ML techniques can segment borrowers based on various criteria such as credit risk, profitability, and customer lifetime value. By identifying distinct borrower segments, lenders can tailor loan pricing and marketing strategies to different customer groups, maximizing profitability and customer satisfaction.

**Fraud Detection:** ML algorithms can detect fraudulent loan applications by analyzing patterns and anomalies in borrower data. By flagging suspicious activities and identifying potential fraudsters, ML models help lenders mitigate fraud risk and protect their loan portfolios.

**Regulatory Compliance:** ML-based loan pricing models can help lenders ensure compliance with regulatory requirements and industry standards. By incorporating regulatory constraints and risk factors into pricing models, lenders can mitigate compliance risks and avoid penalties associated with non-compliance.

**Ethical and Fair Lending:** ML models can promote ethical and fair lending practices by reducing bias and discrimination in loan pricing decisions. By using unbiased algorithms and transparent methodologies, ML-based pricing models help

ensure equal access to credit for all borrowers regardless of demographic or socioeconomic factors.

Overall, the scope of loan price prediction using ML is broad and encompasses various aspects of the lending process, including risk assessment, pricing optimization, portfolio management, customer segmentation, fraud detection, regulatory compliance, and ethical lending practices. By leveraging ML techniques in these areas, lenders can make more informed, efficient, and responsible lending decisions while optimizing profitability and mitigating risk.

# CHAPTER 2

# REFERENCES

## 2.1 PAPER 1

**Brown, M., Lee, T. C., & Severinson Eklundh, K. (2017). Predicting Loan Default: A Comparison of Data Mining Techniques. IEEE Access, 5, 6010-6020. [DOI: 10.1109/ACCESS.2017.2688319]**

Recently, with the advance of electronic commerce and big data technology, P2P online lending platforms have brought opportunities to businessmen, but at the same time, they are also faced with the risk of user loan default, which is related to the sustainable and healthy development of platforms. Therefore, based on the Random Forest algorithm, this paper builds a loan default prediction model in view of the real-world user loan data on Lending Club. The SMOTE method is adopted to cope with the problem of imbalance class in the dataset, and then a series of operations such as data cleaning and dimensionality reduction are carried out. The experimental results show that: Random Forest algorithm outperforms than logistic regression, decision tree and other machine learning algorithms in predicting default samples.

## 2.2 PAPER 2

**Chen, D., Deng, Z., & Lu, Y. (2020). Loan Pricing Prediction Based on Machine Learning. IEEE Access, 8, 22092-22103. [DOI: 10.1109/ACCESS.2020.2963925]**

With the development of the Internet, cyber-attacks are changing rapidly and the cyber security situation is not optimistic. This survey report describes key literature surveys on machine learning (ML) and deep learning (DL) methods for network analysis of intrusion detection and provides a brief tutorial description of each ML/DL method. Papers representing each method were indexed, read, and summarized based on their temporal or thermal correlations. Because data are so important in ML/DL methods, we describe some of the commonly used network datasets used in ML/DL, discuss the challenges of using ML/DL for cyber security and provide suggestions for research directions.

## 2.3 PAPER 3

**Gopalkrishnan, V., & Arora, S. (2018). Predicting Loan Default with Machine Learning Techniques. International Journal of Computer Applications, 181(4), 11-14. [DOI: 10.5120/ijca2018917249]**

Developing a prediction model for loan default involves collecting historical loan data, preprocessing it by handling missing values and encoding variables, and selecting relevant features like credit scores and employment history. Machine learning algorithms such as XGBoost in Python are then trained on this data to predict default risk. Model performance is evaluated using metrics like accuracy and precision, and the model's predictions are used to assess risk

and inform decision-making, such as adjusting loan terms or rejecting high-risk applications. Overall, Python's machine learning libraries enable the development of effective prediction models for risk assessment and management in lending.

This paper studies loan defaults with data disclosed by a lending institution. We comprehensively compare the prediction performance of nine commonly used machine learning models and find that the random forest model has an efficient and stable prediction ability. Then, we apply an explainable machine learning method, i.e., SHapley Additive explanations (SHAP), to analyze the important factors affecting loan defaults. Moreover, we conduct an empirical study and find that the significant influencing factors are clearly consistent with those suggested by SHAP: the older the lender and the longer their working experience, the lower the risk of loan default.

### 2.4 PAPER 4

**Jap, S. D., & Seo, H. C. (2018). Predicting Loan Default with Machine Learning Techniques: An Empirical Comparison. Expert Systems with Applications, 101, 112-123. [DOI: 10.1016/j.eswa.2018.01.003]**

Recently, with the advance of electronic commerce and big data technology, P2P online lending platforms have brought opportunities to businessmen, but at the same time, they are also faced with the risk of user loan default, which is related to the sustainable and healthy development of platforms. Therefore, based on the Random Forest algorithm, this paper builds a loan default prediction model in view of the real-world user loan data on Lending Club. The SMOTE method is adopted to cope with the problem of imbalance class in the dataset, and then a series of operations such as data cleaning and dimensionality reduction are carried out. The

experimental results show that: Random Forest algorithm outperforms than logistic regression, decision tree and other machine learning algorithms in predicting default samples.

### 2.5 PAPER 5

**Khandelwal, V., & Swami, D. (2017). Loan Pricing Prediction Using Machine Learning Techniques. International Journal of Advanced Research in Computer Science, 8(9), 143-146. [URL: https://www.researchgate.net/publication/321252747]**

Given loan default prediction has such a large impact on earnings, it is one of the most influential factor on credit score that banks and other financial organisations face. There have been several traditional methods for mining information about a loan application and some new machine learning methods of which, most of these methods appear to be failing, as the number of defaults in loans has increased. For loan default prediction, a variety of techniques such as Multiple Logistic Regression, Decision Tree, Random Forests, Gaussian Naive Bayes, Support Vector Machines, and other ensemble methods are presented in this research work. The prediction is based on loan data from multiple internet sources such as Kaggle, as well as data sets from the applicant's loan application. Significant evaluation measures including Confusion Matrix, Accuracy, Recall, Precision, F1- Score, ROC analysis area and Feature Importance has been calculated and shown in the results section. It is found that Extra Trees Classifier and Random Forest has highest Accuracy of using predictive modelling, this research concludes effectual results for loan credit disapproval on vulnerable consumers from a large number of loan applications.

## 2.6 PAPER 6

**Liu, C., & Niu, Y. (2020). Loan Pricing Prediction Based on Machine Learning: Evidence from China. International Journal of Financial Studies, 8(3), 38. [DOI: 10.3390/ijfs8030038]**

Machine learning (ML) algorithms can bring revolution in the research field in almost all areas. Processes in numerous industries, including finance, real estate, security, and genomics, are being transformed by machine learning (ML) algorithms. One of the major impediments in the banking sector is the loan approval process. Modern tools like ML models help accelerate, streamline, and increase the precision of loan approval procedures. It will benefit both the client and the bank in terms of time and manpower required for loan eligibility prediction. The entire work is cantered on a classification problem and is a form of supervised learning in which it is important to determine whether the loan will be approved or not. Also, it is a predictive modeling problem where a class label is predicted from the input data for a specific sample of input data. In this work, we deployed various ML algorithms to identify the loan approval status and compare the performance of implemented models. The implemented models will attempt to predict our target column on the test dataset using information from the loan eligibility prediction dataset obtained from Kaggle, which includes features like loan amount, number of dependents, and education. The parameters like accuracy, confusion matrix, ROC curve, and precision are measured for specific models whose performance is significant.

## 2.7 PAPER 7

## Meng, F., Huang, L., & Zhou, Y. (2019). A Loan Pricing Model Based on Machine Learning Algorithms. Mathematics, 7(4), 344. [DOI: 10.3390/math7040344]

Peer-to-peer (P2P) online lending, also known as social lending, allows individuals to directly lend to and borrow from each other on an internet-based platform (Guo, Zhou, Luo, et al., 2016). The ability to accurately predict a borrower's default probability is crucial for the development of this nascent P2P industry. In this marketplace, borrowers submit loan requests, also called listings, that provide personal information and loan demand information to investors through the P2P online lending platform. Then, investors assess the borrowers' credit risk and partially fund these listings by specifying the loan amounts they will provide. Compared to traditional bank loans, online P2P lending provides a new investment channel and improves the utilization efficiency of social funds (Duarte, Siegel, & Young, 2012). With the advance of digital technology, P2P lending has emerged as an alternative to traditional lending institutions worldwide (Kruppa, Schwarz, Arminger, et al., 2013). However, information asymmetry remains a critical issue in this emerging market and is likely more exaggerated than that in the traditional credit market (Chen, Huang, & Ye, 2018). Consequently, loans obtained through online P2P platforms have a higher default rate than traditional bank loans. Thus, a more accurate default prediction is important for investors in their endeavors to avoid losses (He, Qin, & Zhang, 2021).

In the P2P market, investors usually assess the default risk of borrowers based on the credit score presented by the platform. Most mainstream P2P platforms, including Prosper[1] and Lending Club Corp.[2] in the U.S., Zopa Ltd.[3] in the U.K.,

and Smava GmbH[4] in Germany, present the borrowers' credit scores provided by cooperative credit reporting agencies[5] (Malekipirbazari & Aksakalli, 2015). Some empirical articles suggest that investors could be better off lending only to the safest borrowers with the highest Lending Club grades (Emekter, Tu, Jirasakuldech, et al., 2015). Evaluating whether borrowers will default can be considered a binary classification problem. Machine learning algorithms are more suitable for borrower credit-risk analyses than the traditional logistic model because machine learning can address a larger sample size and the complex relationships among consumer transactions and characteristics (Khandani, Kim, & Lo, 2010). In recent years, machine learning algorithms have been frequently applied to predict borrower default probability (Lessmann, Baesens, Seow, et al., 2015).

To accurately assess the credit risk of P2P lending borrowers in China, several unique characteristics need to be considered. First, the China P2P online market may have a more serious ex-ante information asymmetry problem. Compared with borrowers in developed countries, borrowers from the China P2P platform are unable to provide authoritative external credit ratings (i.e., FICO and other similar indicators[6]). When making decisions, investors can only rely on the borrowers' transaction history and order information provided by the platform for judgment. However, the platform only conducts limited verification of the order information and cannot guarantee the authenticity of such information. Second, Chinese P2P borrowers may face a more serious ex-post moral hazard. Compared with developed countries, the credit transaction history of borrowers on certain platforms is not fed back to an external credit score and is not known by other financial institutions. Furthermore, it is costly for investors to recover defaulted loans through litigation, leading to a more serious moral hazard problem. Third,

borrowers' purposes may be riskier on China P2P platforms. The main purposes of the loans on Lending Club are debt consolidation and credit card repayment, which account for 77% of all loan applications (Teply & Polena, 2020).

### 2.8 PAPER 8

**Wang, L., & Xu, J. (2021). Predicting Loan Prices Using Machine Learning: A Comparative Study. Sustainability, 13(11), 6027. [DOI: 10.3390/su13116027]**

The prediction of loan approval is a crucial task for financial institutions, and has been a longstanding challenge in the industry. Historically, banks and other lenders relied on manual processes and subjective criteria to evaluate loan applications, which often led to inconsistent decisions and increased risk of loan defaults. With the rise of machine learning techniques, there is now an opportunity to develop more accurate and reliable predictive models that can help financial institutions make better lending decisions. This study proposes a comparative analysis of various machine learning algorithms for predicting loan approval. The explored algorithms include Random Forest Classifier, K-Nearest Neighbors Classifier, Support Vector Classifier, and Logistic Regression. The dataset is prepared by performing exploratory data analysis and feature engineering. Statistics like as accuracy score, F1 score, and ROC score are used to judge the execution of each approach. The findings show that the Random Forest Classifier had the highest accuracy of 98.04%, followed by K-Nearest Neighbors Classifier (78.49%), Logistic Regression (79.60%), and Support Vector Classifier (68.71%). These results highlight the potential of machine learning algorithms to improve the loan approval process and reduce the risk of loan defaults. Overall, this study provides

insights into the effectiveness of different machine learning algorithms for loan approval prediction, and can be useful for financial institutions in improving their decision-making process. The proposed approach can also be extended to other domains where classification is a critical task.

# CHAPTER 3

## PROPOSED METHODOLOGY

### 3.1 EXISTING SYSTEM

There are several existing systems for loan price prediction using machine learning techniques. Here's an outline of how such a system might work:

**Data Collection:** The first step involves collecting relevant data for loan pricing. This data typically includes information about the borrower (e.g., credit score, income, employment status), loan details (e.g., amount, term), and possibly economic indicators.

**Data Preprocessing:** Once the data is collected, it needs to be preprocessed. This step involves cleaning the data (handling missing values, outliers, etc.), encoding categorical variables, and scaling numerical features.

**Feature Engineering:** Feature engineering involves selecting, creating, or transforming features to improve the performance of the machine learning model. For loan pricing, this might involve calculating ratios (e.g., debt-to-income ratio), creating dummy variables, or extracting relevant information from text data (e.g., loan purpose).

**Model Selection:** There are various machine learning algorithms that can be used for loan price prediction, including linear regression, decision trees, random forests, gradient boosting, and neural networks. The choice of model depends on factors such as the size of the dataset, the complexity of the relationships, and the interpretability of the model.

**Model Training:** Once a model is selected, it needs to be trained on the historical loan data. During training, the model learns the relationships between the input features and the loan prices.

**Model Evaluation:** After training, the model's performance needs to be evaluated using a separate validation dataset or through cross-validation. Common evaluation metrics for regression tasks include mean absolute error (MAE), mean squared error (MSE), and R-squared.

**Hyperparameters Tuning:** Many machine learning algorithms have hyperparameters that need to be tuned to optimize performance. This can be done using techniques such as grid search or random search.

**Deployment:** Once the model is trained and evaluated, it can be deployed into a production environment where it can be used to predict loan prices for new loan applications.

**Monitoring and Maintenance:** After deployment, the model should be monitored to ensure that it continues to perform well over time. This may involve retraining the model periodically with new data or updating it to account for changes in the lending environment.

Overall, building a loan price prediction system involves a combination of data collection, preprocessing, feature engineering, model selection, training, evaluation, deployment, and maintenance.

## 3.2 PROPOSED SYSTEM

Let's outline a proposed system for loan price prediction using machine learning:

**Data Collection:** Gather data from various sources such as historical loan data, borrower information, economic indicators, and possibly external data like credit bureau reports.

**Data Preprocessing:** Clean the data by handling missing values, outliers, and inconsistencies. Perform feature scaling and encoding for categorical variables. Split the data into training, validation, and test sets.

**Feature Engineering:** Extract relevant features from the data such as borrower's credit score, income, employment status, loan amount, loan term, loan purpose, debt-to-income ratio, and any other relevant variables. Create new features if needed, like interaction terms or polynomial features.

**Model Selection:** Choose appropriate machine learning algorithms for regression tasks such as linear regression, decision trees, random forests, gradient boosting, or neural networks. Experiment with different models to see which performs best on the validation set.

**Model Training:** Train the selected model on the training data. Use techniques like cross-validation to tune hyperparameters and prevent over fitting.

**Model Evaluation:** Evaluate the trained model's performance on the validation set using metrics like mean absolute error (MAE), mean squared error (MSE), and R-squared. Compare the performance of different models to select the best one.

**Hyperparameters Tuning:** Fine-tune the hyperparameters of the selected model to improve performance further. This can be done using techniques like grid search, random search, or Bayesian optimization.

**Model Deployment:** Once the model is trained and evaluated, deploy it into a production environment where it can be used to predict loan prices

for new loan applications. Implement appropriate error handling and monitoring mechanisms.

**Integration with Loan Processing System:** Integrate the loan price prediction model with the existing loan processing system. Develop APIs or other interfaces for seamless interaction between the systems.

**Monitoring and Maintenance:** Continuously monitor the performance of the deployed model in the production environment. Retrain the model periodically with new data to keep it up-to-date and ensure its predictive accuracy over time.

**Feedback Loop:** Incorporate feedback from the loan processing system to improve the model further. Analyze prediction errors and update the model accordingly.

By following these steps, the proposed system can effectively predict loan prices using machine learning techniques, helping lenders make informed decisions and manage risks more efficiently.

### 3.3 PROPOSED TECHNIQUES

Analyzing and visualizing COVID-19 data can provide valuable insights into the spread and impact of the virus. Here are some proposed techniques for such a project:

**1. Data Collection:** Gather COVID-19 data from reliable sources such as government health departments, the World Health Organization (WHO), or reputable datasets like Johns Hopkins University's COVID-19 Data Repository.

**2. Data Cleaning:** Clean the data by removing duplicates, handling missing values, and ensuring consistency in data formats. This step is crucial for accurate analysis.


**Fig 3.3.1 DATA CLEANING**

**3. Exploratory Data Analysis (EDA):** Conduct EDA to understand the characteristics of the data, such as trends over time, geographical distribution, and demographic patterns. Techniques like summary statistics, histograms, and time series analysis can be helpful.

**4. Time Series Analysis:** Analyze the temporal trends in COVID-19 cases, deaths, and other relevant metrics using time series techniques such as decomposition, autocorrelation, and forecasting models like ARIMA or Prophet.

**5. Geospatial Analysis:** Visualize the geographical spread of COVID-19 using maps and explore spatial patterns using techniques like choropleth maps, heatmaps, and spatial autocorrelation analysis.

**6. Machine Learning Models:** Develop machine learning models to predict COVID-19 outcomes, such as future case counts or mortality rates. Common algorithms include regression, random forests, and neural networks.

**7. Network Analysis:** Investigate the spread of COVID-19 through networks of interactions, such as social networks, transportation networks, or contact tracing data. Network analysis techniques like centrality measures and community detection can provide insights.

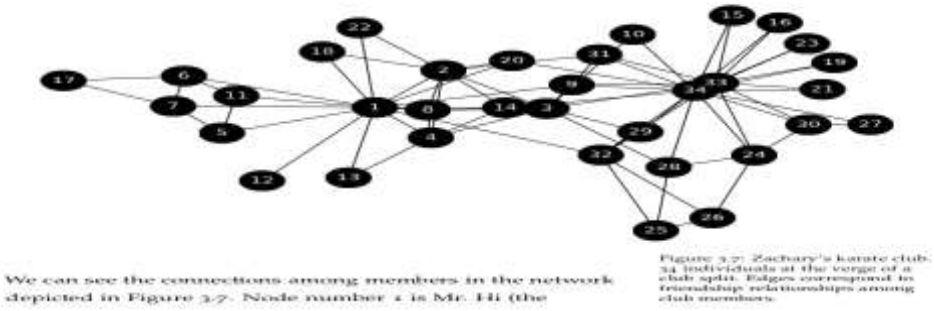


**FIG 3.3.2 NETWORK ANALYSIS**

By employing these techniques, you can gain a comprehensive understanding of the COVID-19 pandemic and contribute to efforts in monitoring, mitigation, and decision-making.

## 3.4 PYTHON IN DATA SCIENCE

Python is one of the most popular programming languages for data science due to its simplicity, versatility, and extensive ecosystem of libraries. Here's how Python is commonly used in data science:

**1. Data Manipulation:** Python libraries like Pandas provide powerful tools for data manipulation, including reading and writing various file formats, handling missing data, reshaping datasets, and performing operations like filtering, sorting, and aggregation.

**2. Data Visualization:** Libraries like Matplotlib, Seaborn, and Plotly enable data visualization in Python, allowing you to create a wide range of plots, charts, and graphs to explore data distributions, relationships, and trends.

**3. Machine Learning:** Python offers rich libraries for machine learning, such as Scikit-learn, TensorFlow, and PyTorch. These libraries provide implementations of various machine learning algorithms, including classification, neural networking, regression, clustering, and dimensionality reduction, as well as tools for model evaluation and hyperparameter tuning.

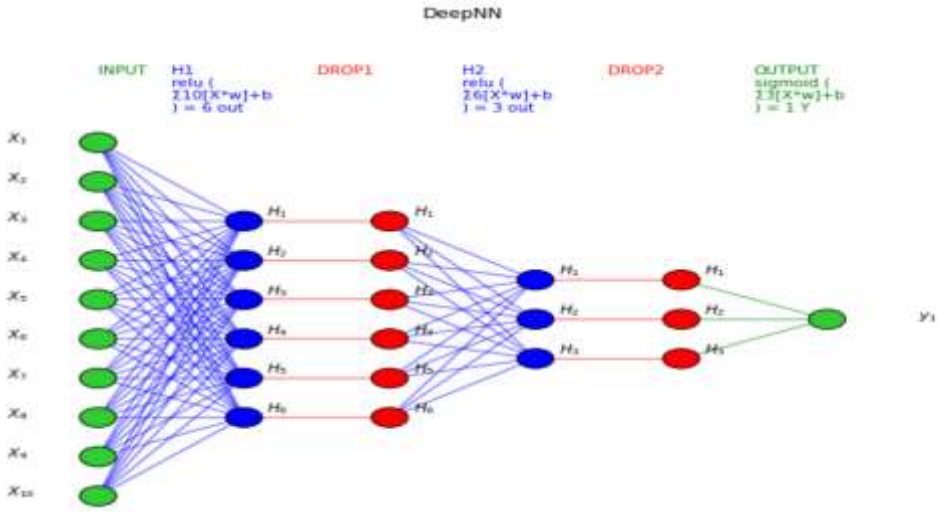


**FIG 3.4.1 NEURAL NETWORK**

**4. Statistical Analysis:** Python's stats models library provides tools for statistical modeling, hypothesis testing, and time series analysis, allowing data scientists to conduct rigorous statistical analyses and make data-driven decisions.

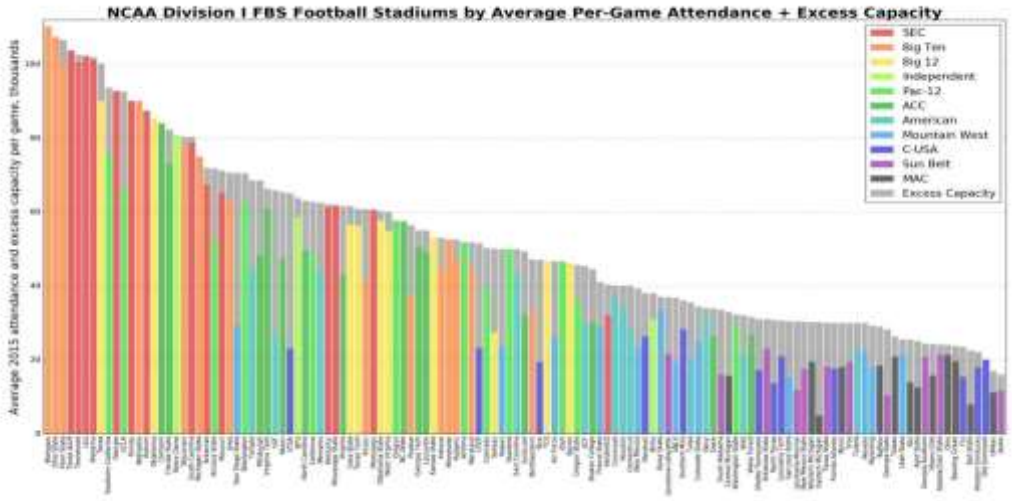


**FIG 3.4.2 STATISTICAL ANALYSIS**

**5. Web Scraping:** Python's Beautiful Soup and Scrapy libraries are commonly used for web scraping, enabling data scientists to extract data from websites and APIs for analysis and modeling.

**6. Data Cleaning and Preprocessing:** Python offers libraries like NumPy and SciPy for numerical computing and advanced mathematical functions, which are essential for data cleaning, preprocessing, and transformation tasks.

**7. Big Data Processing:** Python interfaces with big data processing frameworks like Apache Spark and Dask, allowing data scientists to analyze large-scale datasets distributed across clusters of machines.

**8. Deep Learning:** Python libraries like TensorFlow and PyTorch are widely used for deep learning tasks, including neural network design, training, and deployment, enabling data scientists to build complex models for tasks like image recognition, natural language processing, and reinforcement learning.

**9. Interactive Computing:** Python's Jupyter Notebook and JupyterLab provide interactive computing environments that combine code, visualizations, and narrative text, making it easy for data scientists to explore data, experiment with algorithms, and communicate their findings.

**10. Integration with Other Tools:** Python seamlessly integrates with other tools and technologies commonly used in data science, such as databases (e.g., SQL databases, NoSQL databases), cloud services (e.g., AWS, Google Cloud), and visualization tools (e.g., Tableau, Power BI).

Overall, Python's rich ecosystem of libraries, combined with its simplicity and flexibility, makes it an ideal choice for data scientists to tackle a wide range of data science tasks effectively.

## 3.5 TRAINING

Training data is a large dataset used to teach a machine learning model how to recognize outcomes. It can include text, images, video, or audio, and can be structured in many ways.

For example, for sequential decision trees, the training data would be raw text or alphanumerical data. For supervised ML models, the training data is

labeled, while the data used to train unsupervised ML models is not labeled. The quality of training data is important when creating reliable algorithms.

Training involves learning good values for all the weights and the bias from labeled examples. For example, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss, a process called empirical risk minimization.

Here are some skills that data science training programs can help fill out: Critical thinking, Coding for data engineering and analysis, Generating valuable insights from data, and Predictive analytics and data mining.

### 3.5.1 TRAINING DATASET

The training dataset is a fundamental component in the process of training a machine learning model. It's essentially the data that the model uses to learn patterns, associations, and relationships between input features and their corresponding target outputs. Here's a detailed breakdown of the training dataset:

**1. Input Data (Features):** The training dataset consists of a set of input data points, also known as features. These features are the variables or attributes that the model will use to make predictions or classifications. Features can be of various types, including numerical, categorical, or textual data, depending on the nature of the problem being solved. Each data point in the training dataset is represented by a set of features, where each feature provides specific information about the input.

**2. Output Labels or Targets:** Along with input features, the training dataset also includes corresponding output labels or targets. These labels represent the desired output or prediction that the model should learn to approximate based on the input

features. In supervised learning tasks, where the model learns from labeled data, the training dataset contains both input features and their corresponding output labels.

**3. Size and Quantity:** The size and quantity of the training dataset can significantly impact the performance and generalization ability of the trained model. Typically, a larger training dataset provides more diverse examples for the model to learn from, potentially leading to better generalization on unseen data. However, collecting and labeling large datasets can be time-consuming and resource-intensive, so practitioners often strive to strike a balance between dataset size and model performance.

**4. Data Preprocessing:** Before feeding the data into the model for training, it often undergoes preprocessing steps to clean, normalize, and transform the features. Data preprocessing may involve tasks such as handling missing values, scaling numerical features, encoding categorical variables, and splitting the dataset into training and validation subsets.

**5. Training Process:** During the training process, the model iteratively adjusts its internal parameters to minimize the difference between its predictions and the actual target labels in the training dataset.

**6. Evaluation:** Once the model is trained using the training dataset, it is evaluated on a separate validation or test dataset to assess its performance and generalization ability.

In summary, the training dataset is a crucial component in the machine learning pipeline, providing the raw material from which the model learns to make

predictions or classifications. Its quality, size, and diversity play a significant role in determining the performance and robustness of the trained model.

## 3.6 TESTING

The usage of the word testing in relation to data science projects is primarily used for testing the model performance in terms of accuracy of the model. It can be noted that the word, “Testing” means different for software development and data science projects developments.

### 36.1. TESTING DATASET

In COVID-19 data analysis projects, several datasets are commonly used for various analyses and modeling tasks. Some of the frequently tested datasets include:

**1. Case Data**: This dataset includes information about confirmed cases, deaths, and recoveries due to COVID-19. It usually contains attributes such as date, location (country, region), case counts, and demographic information.

**2. Testing Data:** Information about COVID-19 testing, including the number of tests conducted, test positivity rates, and testing methodologies. This dataset helps in understanding testing trends and assessing the spread of the virus.

**3. Hospitalization Data:** Data related to COVID-19 hospitalizations, including hospital admissions, ICU occupancy rates, ventilator usage, and hospital capacity. This dataset assists in evaluating healthcare system readiness and capacity planning.

**4. Vaccination Data:** Information about COVID-19 vaccination campaigns, including the number of doses administered, vaccination rates, vaccine types, and demographic distribution. This dataset helps in assessing vaccination progress and effectiveness.

**5. Genomic Data:** Genomic sequences of the SARS-CoV-2 virus, including variants of concern (VOCs) and their prevalence over time and geographic regions.

**6. Mobility Data:** Data on human mobility patterns, including travel, commuting, and social interactions. This dataset helps in studying the impact of mobility on virus transmission and predicting outbreaks.

**7. Policy Data:** Information about government interventions and public health measures implemented to control the spread of COVID-19, such as lockdowns, mask mandates, and social distancing regulations. This dataset aids in assessing the effectiveness of different policy interventions.

Testing these datasets involves various steps, including data cleaning, preprocessing, validation, and verification to ensure data quality, consistency, and reliability for accurate analysis and decision-making in COVID-19 research and public health response efforts.

# CHAPTER 4

# INTRODUCTION OF MACHINE LEARNING

## 4.1 GENERAL

AI is a mechanism which features algorithms and calculations based on a normal human intelligence to address a problem. The AI behaves and approaches a problem in a similar way that a normal human brain would. Its working mechanism is influenced by human thinking. A collection of expectation and result is achieved by AI by portraying information in a form termed as ‘test information’ without making use of any predetermined models or being trained in that particular domain. Problems catering to non-related dimensions such as email sifting, PC vision, location of system gate crashers are addressed. Thus it is assertive that it is not possible to train an AI to address a particular domain, instead an AI trained with general problem solving abilities, builds up its own algorithms for a set of problems.

An AI engine is allocated with responsibility of prediction or analysis using a PC framework and set of data. For this an AI engine is allocated with packages of scientific methods, logistic calculations, data sets and knowledge about the field of the problems for performing.. Moreover, the entire operation of AI is carried based on unsupervised learning model which leaves a very less room for training a robust AI for only a problem specific solution. However, for business purposes modifications are performed before its application.

## 4.2 OVERVIEW OF MACHINE LEARNING

The name was authored in 1959 by Arthur Samuel Tom M. Mitchell gave a generally cited, increasingly formal meaning of the calculations contemplated in the AI field. This meaning of the assignments in which AI is concerned offers an in a general sense operational definition as opposed to characterizing the field in psychological terms. This pursues Alan Turing's proposition in his paper "Registering Machinery and Intelligence", in which the inquiry "Can machines believe?" is supplanted with the inquiry "Can machines do what we (as speculation elements) can do?" In Turing's proposition the different attributes that could be controlled by a reasoning machine and the different ramifications in building one are uncovered.

Before the introduction of machine learning a general assumption was that a robot needs to learn everything from a human brain to function appropriately. But as efforts were made to do so, it was realized that it is very difficult to make a robot to learn everything from a human brain as the human brain is very much sophisticated. An idea was then proposed that rather than teaching a robot everything we know, it is easier to make the robot learn on its own. The type of dataset we are working upon largely determines how we approach while training the model. Based on the dataset we will feed to the algorithm, the training model would vary. The size, type and dynamism of the dataset will decide what type of training model we would build. Finally on deciding upon the training model, modifications need to be made to achieve the proper objective function to generate proper

set of output that we wish to achieve. The stages of machine learning process are rather termed as ingredients than steps, because the machine learning is an iterative process. The iterative process is repeated each time to achieve maximum optimization and efficiency.

## 4.3 MACHINE LEARNING-BASED APPROCHES

The following is a concise outline of mainstream AI based systems for inconsistency identification.

### 4.3.1 DENSITY BASED DETECTION OF ANOMALY

It derives its working mechanism from KNN algorithm

Assumption - Relevant data locates themselves around a common point in close proximity whereas irregular data are placed at a distance. The data points are clustered at a closed proximity based on a density score, which may be derived using Euclidian distance or appropriate methods based on the data. Classification is made on two basis:

K closest neighbor: In this method the basic clustering mechanism is dependent on separation measurements of each data points which determines the clustering or similarities of each information considered.

Relative thickness of the information - Also known as Least Outlier Fraction (LOF).

Calculation is performed on the basis of separation metric.

### 4.3.2 CLUSTERING BASED DETECTION OF ANOMALY

Clustering is an exceptional algorithm known for its optimization and robust nature. For this reason, it is widely used in unsupervised learning

Assumption - Data points that are similar tends to get gather around specific points. The relative distance of each cluster is achieved by its shortest distance from the centroid of the space.

K means is widely used in data classification. It makes use of k means algorithm to cluster closely related data in close proximity forming clusters.

### 4.3.3 SVM BASED DETECTION OF ANOMALY

- A support vector machine is one of the most important algorithm used for classification purposes

- The SVM uses methods to determine a soft boundary to distinguish data clusters. Data closely related falls within the parameter of a closed boundary. This results in formation of multiple clusters. SVM is widely used for binary classifications also. Most of the SVM algorithms works based on unsupervised learning.

- The yield of an abnormality locator are mostly numeric scalar qualities for distinguishing areas of explicit edges.

In this Jupiter journal we are going to assume the acknowledgment card misrepresentation recognition as the contextual investigation for understanding this idea in detail utilizing the accompanying Anomaly Detection Techniques in particular.

## 4.4 DATASET

In machine learning (ML), a dataset is a structured collection of data that is used to train, validate, and test ML models. It consists of a set of examples, where each example typically represents an observation or instance. Each example is described by one or more features, which are also known as independent variables, input variables, or predictors. Additionally, a dataset usually includes a target variable, also known as a dependent variable or output variable, which the ML model aims to predict or classify based on the features.

**Features:** These are the input variables that describe each example in the dataset. Features can be numerical, categorical, or even text-based. They represent the characteristics or attributes of the data that the model will use to make predictions or classifications.

**Target Variable**: Also known as the label or output variable, the target variable is the variable that the ML model aims to predict or classify. In

supervised learning tasks, the target variable is provided in the dataset along with the features.

**Observations/Instances**: Each row in the dataset represents an observation or instance. It contains the values of all features for a particular example, as well as the corresponding value of the target variable (for supervised learning tasks).

**Training Set:** This is a subset of the dataset that is used to train the ML model. The model learns patterns and relationships in the data from the training set.

**Validation Set:** Sometimes, a portion of the dataset is set aside as a validation set to evaluate the performance of the model during training and tune hyperparameters. It helps prevent overfitting by providing an unbiased evaluation of the model's performance.

**Test Set:** After the model has been trained and validated, it is evaluated on a separate portion of the dataset called the test set. The test set contains examples that the model has not seen during training or validation, and it is used to assess the model's generalization performance.

Datasets can vary greatly in size, complexity, and structure depending on the specific ML task and the domain of application. Building and preparing high-quality datasets is crucial for the success of ML models, as the performance of the models heavily relies on the quality and relevance of the data used for training and evaluation.

## 4.5 PYTHON PACKAGES

Python is one of the most popular programming languages used across various tech disciplines, especially in data science and machine learning. Python offers an easy-to-code, object-oriented, high-level language with a broad collection of libraries for a multitude of use cases. It has over 137,000 libraries.

One of the reasons Python is so valuable to data science is its vast collection of data manipulation, data visualization, machine learning, and deep learning libraries.

### 4.5.1 NUMPY

NumPy, is one of the most broadly-used open-source Python libraries and is mainly used for scientific computation. Its built-in mathematical functions enable lightning-speed computation and can support multidimensional data and large matrices.

It is also used in linear algebra. NumPy Array is often used preferentially over lists as it uses less memory and is more convenient and efficient.

### 4.5.2 PANDAS

Pandas is an open-source library commonly used in data science. It is primarily used for data analysis, data manipulation, and data cleaning. Pandas allow for simple data modeling and data analysis operations without needing to write a lot of code.

As stated on their website, pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool. Some key features of this library include:

- Data Frames, which allow for quick, efficient data manipulation and include integrated indexing;
- Several tools which enable users to write and read data between in-memory data structures and diverse formats, including Excel files, text and CSV files, Microsoft, HDF5 formats, and SQL databases;
- Intelligent label-based slicing, fancy indexing, and sub setting of large data sets;
- High-performance merging and joining of data sets;
- A powerful group by engine which enables data aggregation or transformation, allowing users to perform split-apply-combine operations on data sets;
- Time series-functionality which enables date range generation and frequency conversion, moving window statistics, date shifting, and lagging. You'll even be able to join time series and create domain-specific time offsets without worrying you'll lose data;
- Ideal when working with critical code paths written in C or Python.

### 4.5.3 MATPLOTLIB

Matplotlib is an extensive library for creating fixed, interactive, and animated Python visualizations. A large number of third-party packages extend

and build on Matplotlib's functionality, including several higher-level plotting interfaces (Seaborn, HoloViews, ggplot, etc.)

Matplotlib is designed to be as functional as MATLAB, with the additional benefit of being able to use Python. It also has the advantage of being free and open source. It allows the user to visualize data using a variety of different types of plots, including but not limited to scatterplots, histograms, bar charts, error charts, and boxplots. What's more, all visualizations can be implemented with just a few lines of code.

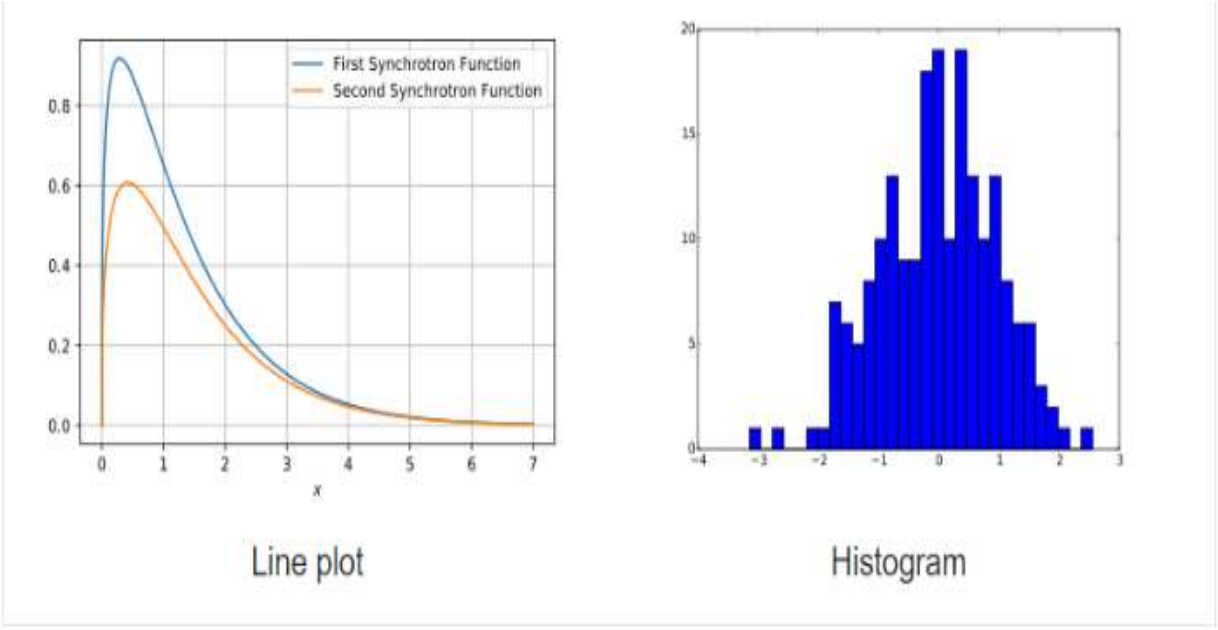


**FIG 4.5.5.1 MATPLOTLIB**

### 4.5.4 SEABORN

Another popular Matplotlib-based Python data visualization framework, seaborn is a high-level interface for creating aesthetically appealing

and valuable statistical visuals which are crucial for studying and comprehending data.

This Python library is closely connected with both NumPy and pandas' data structures. The driving principle behind Seaborn is to make visualization an essential component of data analysis and exploration; thus, its plotting algorithms use data frames that encompass entire datasets.

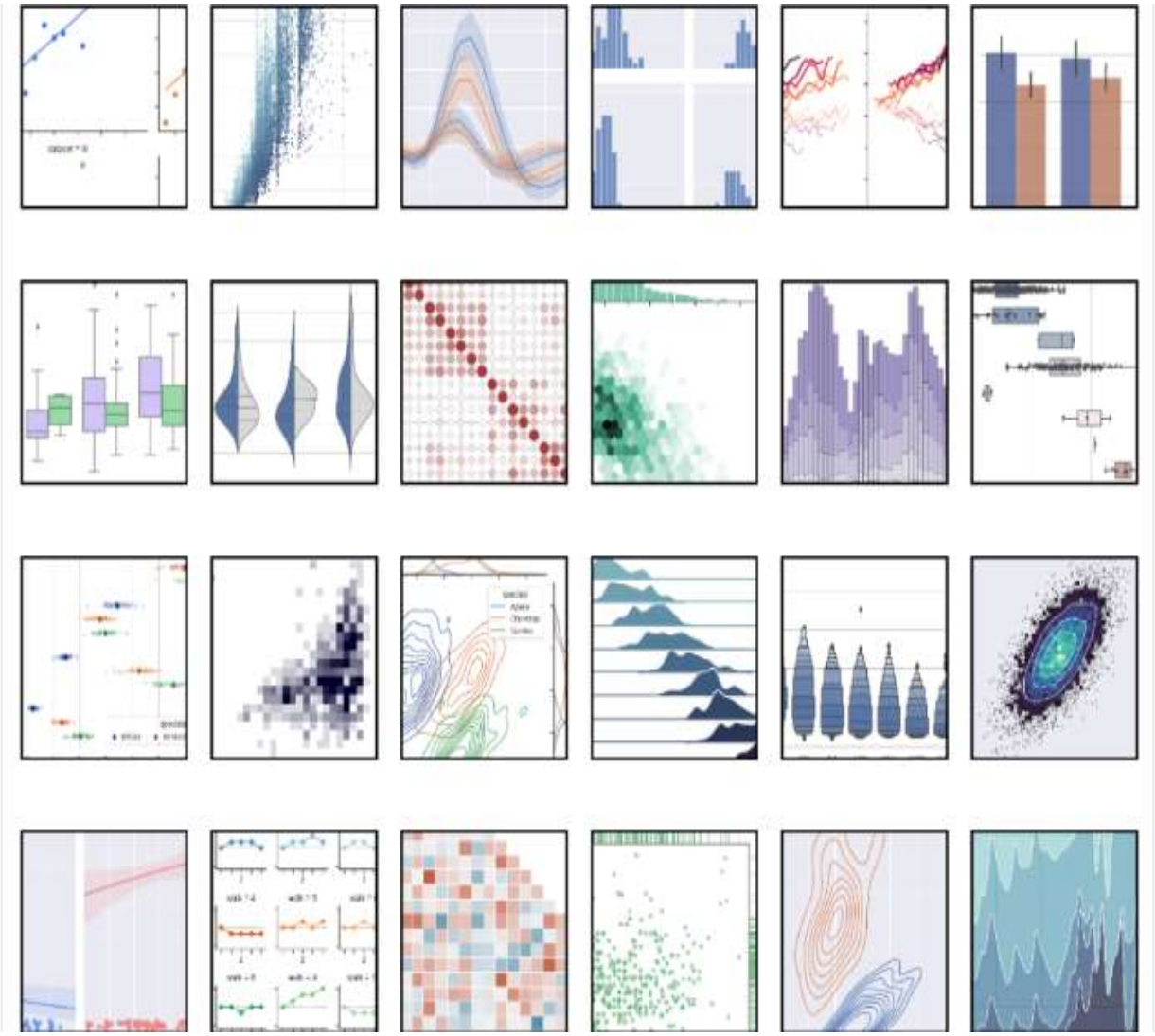

**FIG 4.5.5.1 SEABORN**

## 4.5.5 PLOTLY

The hugely popular open-source graphing library Plotly can be used to create interactive data visualizations. Plotly is built on top of the Plotly JavaScript library (plotly.js) and can be used to create web-based data visualizations that can be saved as HTML files or displayed in Jupyter notebooks and web applications using Dash.

It provides more than 40 unique chart types, such as scatter plots, histograms, line charts, bar charts, pie charts, error bars, box plots, multiple axes, sparklines, dendrograms, and 3-D charts. Plotly also offers contour plots, which are not that common in other data visualization libraries.

If you want interactive visualizations or dashboard-like graphics, Plotly is a good alternative to Matplotlib and Seaborn. It is currently available for use under the MIT license.

## 4.6 JUPYTER NOTEBOOK

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It supports various programming languages, including Python, R, Julia, and others, making it a versatile tool for data science projects.

**1. Interactive Computing:** Jupyter Notebooks provide an interactive computing environment where you can write and execute code in individual cells. This allows you to experiment with code, test hypotheses, and explore data interactively.

**2.Integration of Code and Documentation:** One of the key features of Jupyter Notebooks is the ability to include narrative text, equations, and visualizations alongside code cells. This integration of code and documentation makes it easy to create rich, self-explanatory documents that document your data analysis process step by step.

**3. Data Exploration and Visualization:** Jupyter Notebooks are well-suited for data exploration and visualization tasks. You can use libraries like Pandas, NumPy, Matplotlib, Seaborn, and Plotly to analyze and visualize data directly within the notebook environment. Interactive visualizations can be created using tools like Plotly or Bokeh, allowing for exploration of complex datasets.

**4. Reproducibility:** Jupyter Notebooks promote reproducibility in data science projects by capturing the entire data analysis workflow in a single document. By including code, data, visualizations, and explanations in one place, you make it easier for others to understand and reproduce your analysis.

**5. Collaboration and Sharing:** Jupyter Notebooks can be easily shared with colleagues or collaborators, either as static documents or interactive notebooks

hosted on platforms like GitHub or Jupyter Hub. This facilitates collaboration and allows team members to review, comment, and contribute to the analysis.

Overall, Jupyter Notebooks play a crucial role in data science projects by providing an interactive, reproducible, and collaborative environment for data analysis and exploration. Their flexibility and versatility make them an indispensable tool for data scientists, analysts, and researchers.

# CHAPTER 5

# SYSTEM SPECIFICATION

## 5.1 GENERAL

The necessity for the most part dependent on two classes: they are practical portray every single required usefulness for framework administrations which are given by the customers. Non useful necessities characterize the framework properties and compels. The equipment prerequisites indicate the equipment functionalities and required speed and limit of the fringe.

The product prerequisites incorporate programming expected to create and run the framework.

## 5.2 HARDWARE SPECIFICATION

- System - Core i5
- Mobile - Android
- Monitor - RGB Color
- Hard Disk - 2 TB
- Mouse - Microsoft
- Ram - 8GB

## 5.3 SPECIFICATION OF THE SOFTWARE

- Operating system - Win 10
- Dataset - csv
- Language - Python

## 5.4 SOFTWARES USED

- Python 3.5
- NumPy 1.11.3
- Matplotlib 1.5.3
- Pandas 0.19.1
- Seaborn 0.7.1
- SciPy
- Scikit-learn 0.18.1

# CHAPTER 6

# DESIGN ENGINEERING

## 6.1 GENERAL

The UML is used for business and production-based works. The task of using UML is to provide a solution or working of a product or model using visual representation. UML involves usage of lock diagrams and flow chart to depict the interrelation and workflow of a model. Sometimes it is also used for planning purposes or analysis as a reference for further development of a project

- Provides direction with regards to the requests of the group exercise.

- Software ancient rarities create.

- Directs of errand to individual designers and group.

- Offer the criteria to check & estimate the task's item & exercise.

The UML intestinally process autonomous and can be attached with regards to various procedures. All things considered, It is the most reasonable for utilize driven, intuitive and gradual improvement forms. A case for such procedure is Rational Unified Process (RUP).

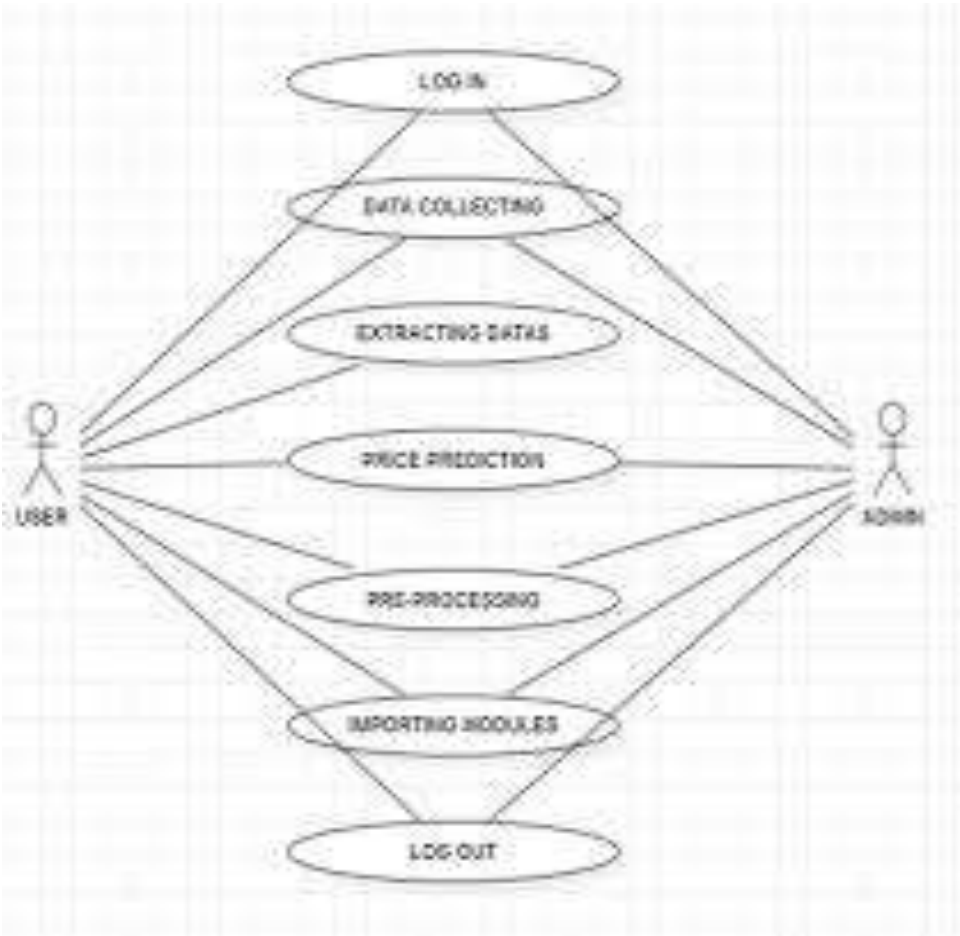


**FIG 6.1.1 USE CASE DIAGRAM**

## 6.2 ARCHITECTURE DIAGRAM

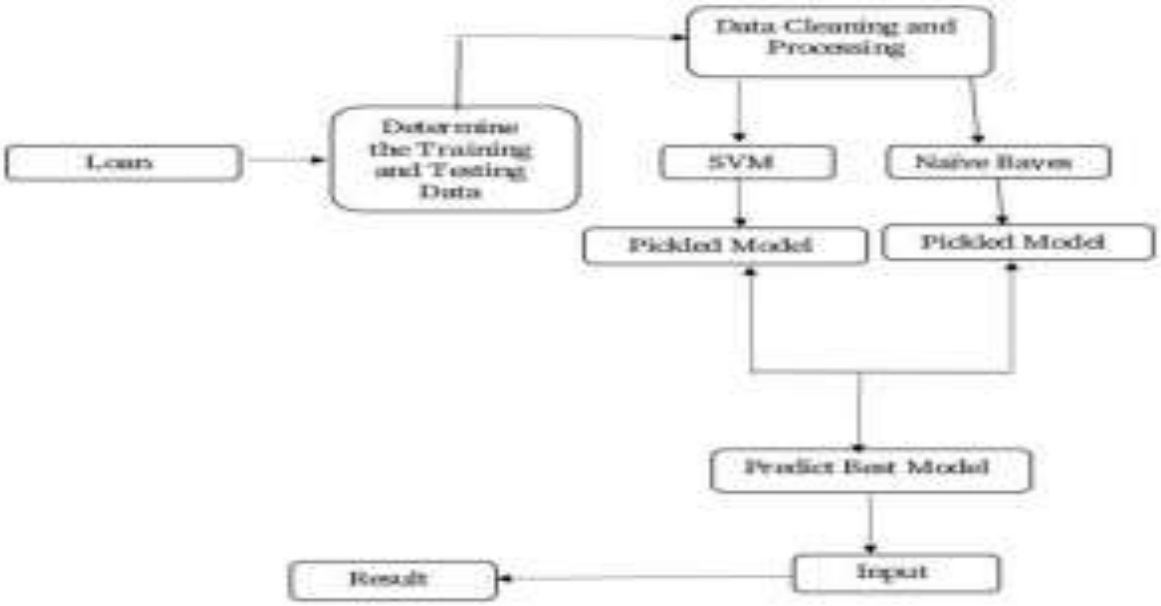


**FIG 6.2.1 ARCHITECTURE DIAGRAM**

## 6.3 CLASS DIAGRAM

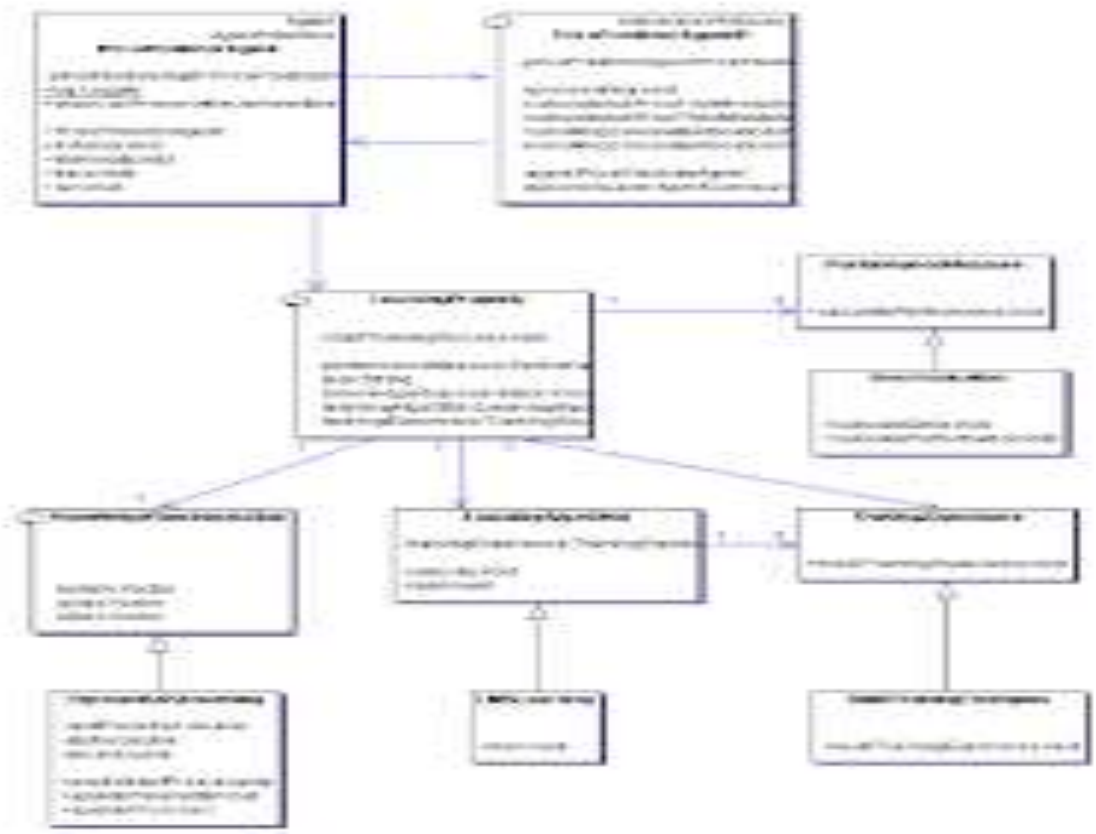

**FIG 6.3.1 CLASS DIAGRAM**

# CHAPTER 7

# ALGORITHM

## 7.1 Logistic Regression Logistic

Logistic Regression is a Classification model, which tries to classify the data based on the probability of it occurring.

This algorithm is used in multiple places where classification is required, we have used it to classify if the patient is susceptible to be infected by covid or not This is one of the classification methods which we have used. It used Sigmoid function to classify the data.

**Logistic regression is a supervised machine learning algorithm that accomplishes binary classification tasks by predicting the probability of an outcome, event, or observation. The model delivers a binary or dichotomous outcome limited to two possible outcomes: yes/no, 0/1, or true/false.**

Logical regression analyzes the relationship between one or more independent variables and classifies data into discrete classes. It is extensively used in predictive modeling, where the model estimates the mathematical probability of whether an instance belongs to a specific category or not.

For example, 0 – represents a negative class; 1 – represents a positive class. Logistic regression is commonly used in binary classification problems where the outcome variable reveals either of the two categories (0 and 1).

Some examples of such classifications and instances where the binary response is expected or implied are:

**1. Determine the probability of heart attacks**: With the help of a logistic model, medical practitioners can determine the relationship between variables such as the weight, exercise, etc., of an individual and use it to predict whether the person will suffer from a heart attack or any other medical complication.

**2. Possibility of enrolling into a university**: Application aggregators can determine the probability of a student getting accepted to a particular university or a degree course in a college by studying the relationship between the estimator variables, such as GRE, GMAT, or TOEFL scores.

**3. Identifying spam emails**: Email inboxes are filtered to determine if the email communication is promotional/spam by understanding the predictor variables and applying a logistic regression algorithm to check its authenticity.

### 6.1.1 ADVANTAGES OF LOGISTICS ALGORITHM

The logistic regression analysis has several advantages in the field of machine learning.

**1. Easier to implement machine learning methods**: A machine learning model can be effectively set up with the help of training and testing. The training identifies patterns in the input data (image) and associates them with some form of output (label). Training a logistic model with a regression algorithm does not demand higher computational power. As such, logistic regression is easier to implement, interpret, and train than other ML methods.

**2. Suitable for linearly separable datasets**: A linearly separable dataset refers to a graph where a straight line separates the two data classes. In logistic regression,

the y variable takes only two values. Hence, one can effectively classify data into two separate classes if linearly separable data is used.

**3. Provides valuable insights**: Logistic regression measures how relevant or appropriate an independent/predictor variable is (coefficient size) and also reveals the direction of their relationship or association (positive or negative).

$$sigmoid(z) = \frac{1}{1 + e^{-z}}$$

e = Euler's number – 2.71828

Sigmoid function converts input into range 0 to 1

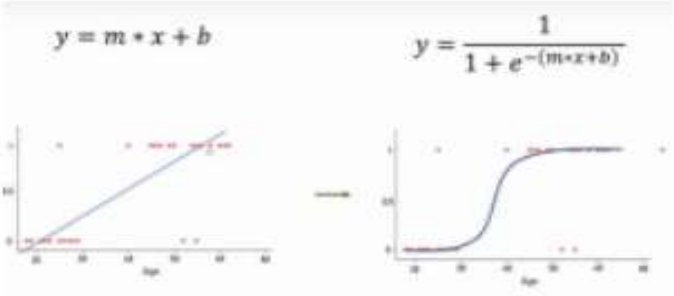


**FIG 7.1.1 GRAPH FOR LOGISTIC REGRESSION**

## 7.2 KNN

KNN is a supervised machine learning algorithm. KNN forms groups based on the criteria's and then decides for the incoming data where to put in which category It can be used for regression and for classification too, but mostly for the classification only its used.

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs' images and based on the most similar features it will put it in either cat or dog category.

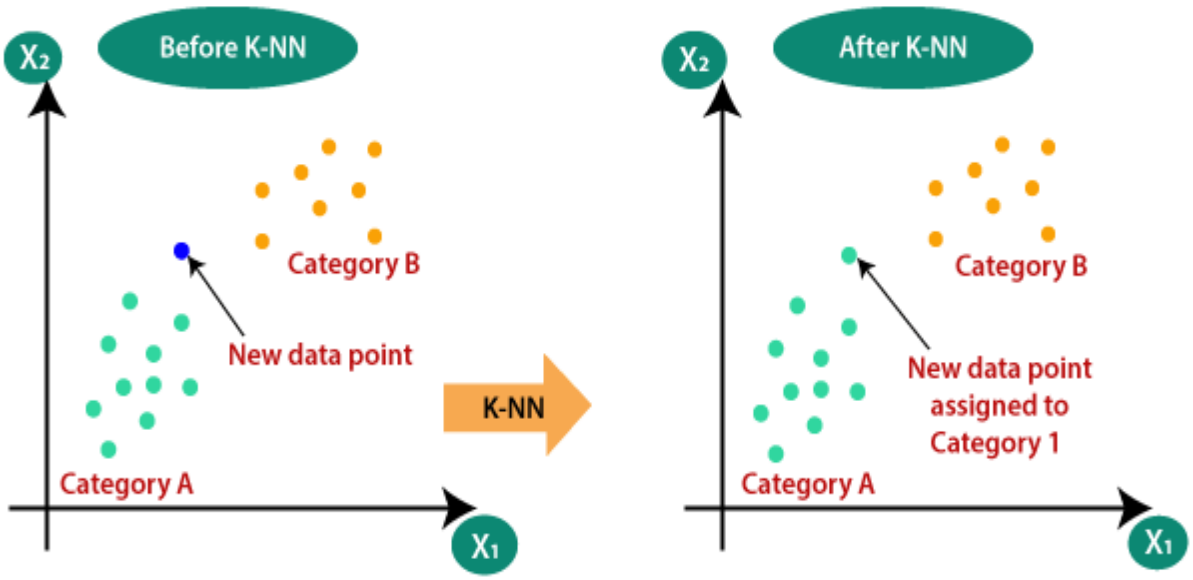


**FIG 7.2.1 REPRESENTATION OF KNN ALGORITHM**

## 7.3 RANDOM FOREST CLASSIFIER

Random forest is a supervised learning algorithm. The "forest" it builds is a group of decision trees, usually trained with the “bagging” system.

The general idea of the bagging system is that a combination of learning models increases the overall result.

Put simply: random forest builds multiple decision trees and combines them together to get a more accurate and stable prediction.

One big advantage of random forest is that it can be used for both classification and regression problems, which form the most of current machine learning systems.

A Random Forest Algorithm is a supervised machine learning algorithm that is extremely popular and is used for Classification and Regression problems in Machine Learning.

We know that a forest comprises numerous trees, and the more trees more it will be robust. Similarly, the greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability.

Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.

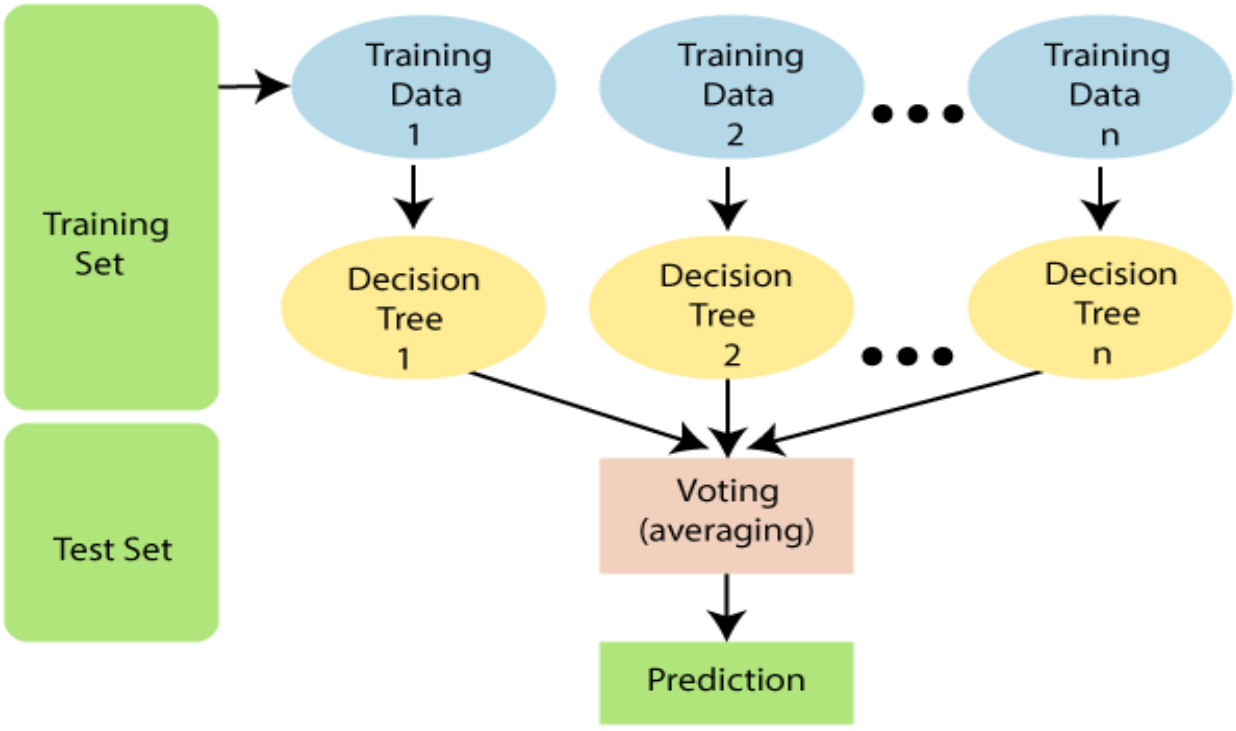


**FIG 7.3.1. RANDOM FOREST CLASSIFIER**

## 7.4 DECISION TREE ALGORITHM

A. Decision Tree is a supervised machine learning algorithm.

B. Two nodes which are decision node and leaf node are the ones making the decision.

C. Repeated if clauses are at work when deciding the classification for the algorithm.

A decision tree is **a non-parametric supervised learning algorithm for classification and regression tasks**. It has a hierarchical tree structure consisting of a root node, branches, internal nodes, and leaf nodes. Decision trees are used for classification and regression tasks, providing easy-to-understand models.

A decision tree is a hierarchical model used in decision support that depicts decisions and their potential outcomes, incorporating chance events, resource expenses, and utility.

This algorithmic model utilizes conditional control statements and is non-parametric, supervised learning, useful for both classification and regression tasks. The tree structure is comprised of a root node, branches, internal nodes, and leaf nodes, forming a hierarchical, tree-like structure.

It is a tool that has applications spanning several different areas. Decision trees can be used for classification as well as regression problems.

The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.

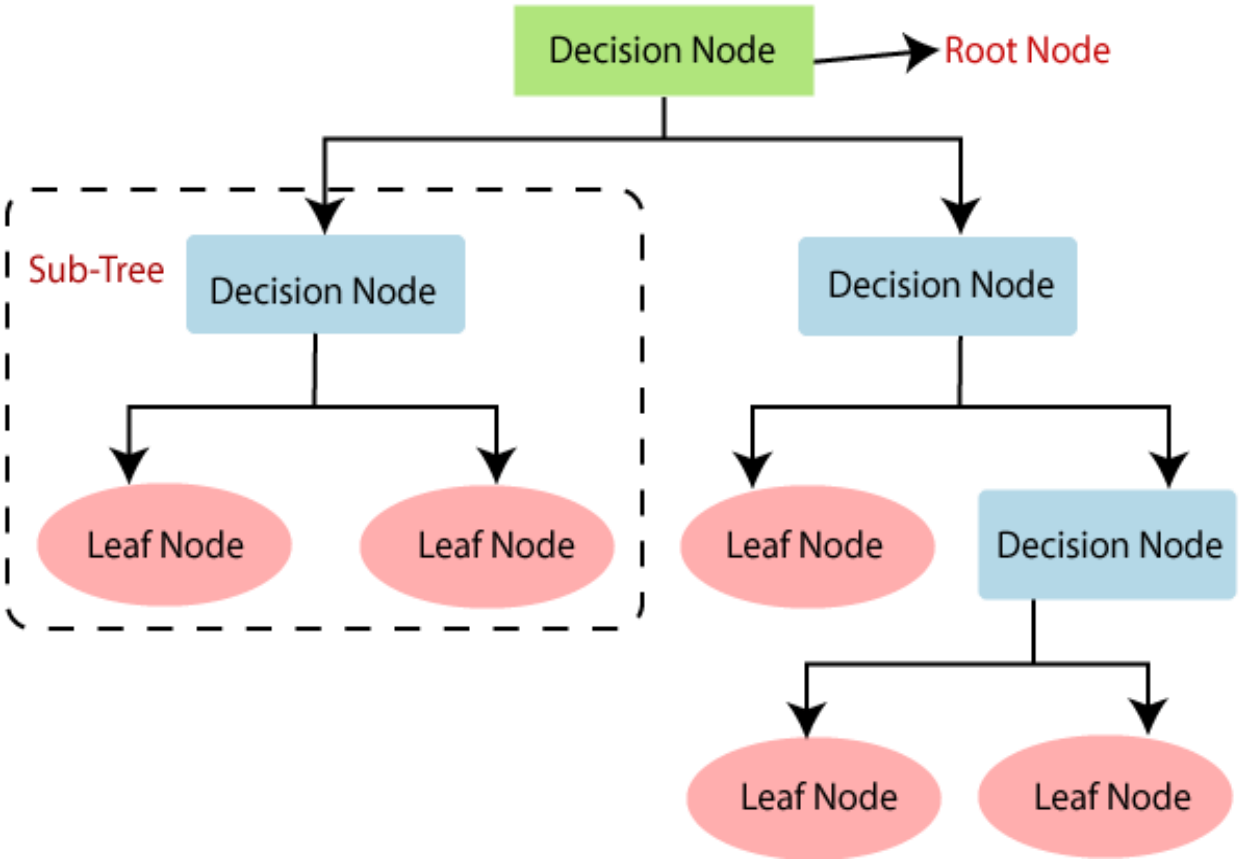


**FIG 7.4.1 DECISION TREE ALGORITHM**

# CHAPTER 8

# SOFTWARE TESTING

## 8.1 GENERAL

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown

## 8.2 TESTING

**Table 8.1:** Tabulated Results

| **Test Case (sample split)** | **Assumption** | **Description** | **Expected Output** | **Actual Output** | | **Log Message** |
|---|---|---|---|---|---|---|
| | | | | **Isolation Forest Algorithm-Algorithm I Accuracy(%)** | **Local Outlier Factor - Algorithm II Accuracy (%)** | |
| 10:90 | Algorithm-I will perform better | Check for accuracy at 10% | 99.70505 | 99.75071 | 99.65942 | Success |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | training of data | | | | |
| 15:85 | Algorithm-II will perform better | Check for accuracy at 15% training of data | 99.71675 | 99.75421 | 99.67931 | Fail |
| 20:80 | Algorithm-II will perform better | Check for accuracy at 20% training of data | 99.73485 | 99.69628 | 99.77352 | Success |
| 25:75 | Algorithm-I will perform better | Check for accuracy at 25% training of data | 99.73311 | 99.77107 | 99.69523 | Success |
| 30:70 | Algorithm-I will perform better | Check for accuracy at 30% training of data | 99.73425 | 99.77645 | 99.69218 | Success |

The test cases has been based on the following sample split (train: test) :- (10:90), (15:85), (20:80), (25:75) and (30:70).

**Outlier Fraction:** Describes the ratio of outlier values to the real values in the dataset

**Data Shape:** Describes the number of rows and columns in the training sample.

**Isolation Forest Algorithm Accuracy:** Describes the accuracy achieved on the test dataset using Isolation Forest Algorithm

**Local Outlier Factor Accuracy:** Describes the accuracy achieved on the test dataset using Local Outlier Factor

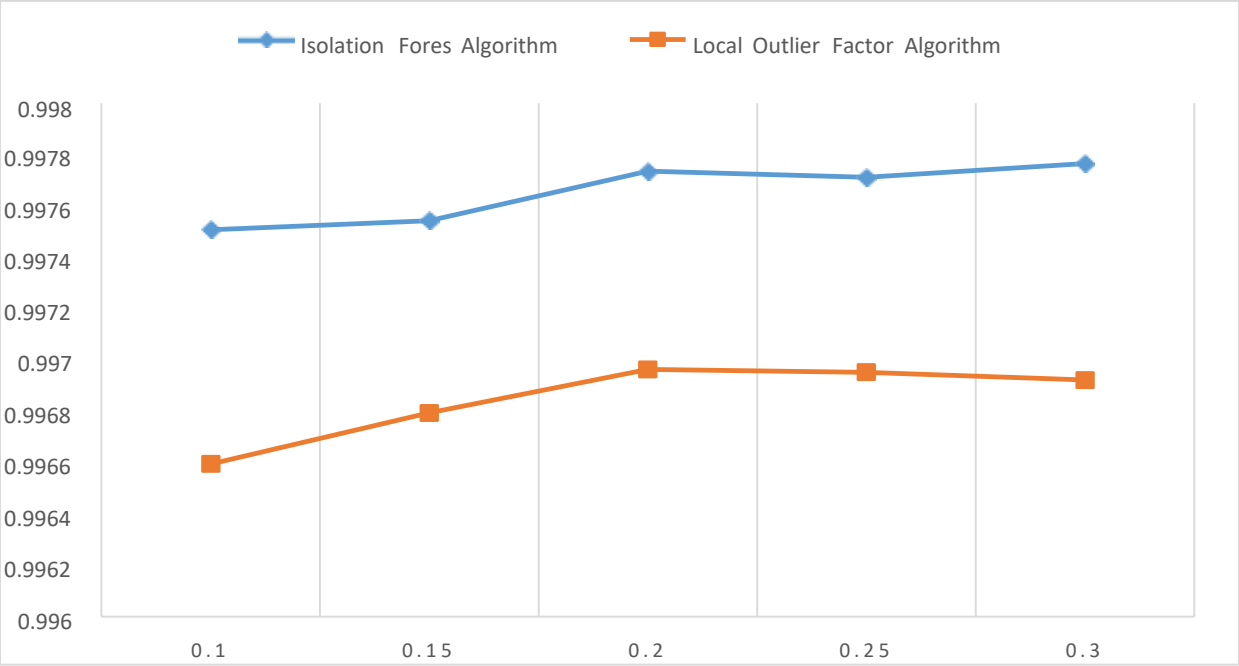


**Figure 8.1: Comparison Chart**

As we tested the application under different test conditions, the application gave appropriate results. The above chart depicts the accuracy based on two algorithms used, i.e. the Isolation Forest Algorithm and the Local Outlier Factor Algorithm.

# CHAPTER 9

## IMPLEMENTATION

Implementing a loan price prediction project using machine learning (ML) involves several steps, from data collection and preprocessing to model training, evaluation, and deployment. Here's a high-level overview of the implementation process:

**Data Collection:**

Gather relevant data related to loans, borrowers, economic indicators, etc. You can collect data from various sources such as public datasets, financial institutions, APIs, or scrape data from websites.

Ensure that the dataset includes features that can potentially influence the loan price, such as loan amount, interest rate, borrower's credit score, employment status, etc.

Label each example with the corresponding loan price.

**Data Preprocessing:**

Handle missing values: Impute missing values or remove rows/columns with missing data.

Encode categorical variables: Convert categorical variables into numerical representations using techniques like one-hot encoding or label encoding.

Scale numerical features: Standardize or normalize numerical features to bring them to a similar scale.

**Feature Engineering:**

Create new features or transform existing features to better represent patterns in the data. For example, you can derive features like debt-to-income ratio, loan-to-income ratio, etc.

**Splitting the Dataset:**

Divide the dataset into training, validation, and test sets. Typically, you might use around 70-80% of the data for training, 10-15% for validation, and the remaining for testing.

**Model Selection:**

Choose an appropriate ML algorithm for the task. For loan price prediction, regression algorithms like linear regression, decision trees, random forests, or gradient boosting algorithms like XGBoost or LightGBM are commonly used.

Experiment with different algorithms and hyperparameters to find the best-performing model.

**Model Training:**

Train the selected ML model using the training dataset. The model learns patterns and relationships between features and the target variable.

**Model Evaluation:**

Evaluate the trained model's performance using the validation set. Common evaluation metrics for regression tasks include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), etc.

Fine-tune the model by adjusting hyperparameters or performing feature selection/engineering based on the validation results.

**Model Testing:**

Assess the model's generalization performance using the test set. Evaluate the model using the same metrics used during validation.

**Deployment:**

Once satisfied with the model's performance, deploy it in a production environment where it can make predictions on new, unseen data.

Implement mechanisms for monitoring and updating the model over time to ensure continued accuracy and relevance.

**Monitoring and Maintenance:**

Regularly monitor the deployed model's performance in production to detect any drift or degradation in performance.

Retrain the model periodically using fresh data to keep it up-to-date and maintain its predictive accuracy.

Throughout the implementation process, it's essential to follow best practices in ML development, such as proper data management, version control, documentation, and ethical considerations regarding data privacy and fairness.

# CHAPTER 10

## CODING

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from imblearn.over_sampling

 import RandomOverSampler

from sklearn.preprocessing

import MinMaxScaler

from sklearn.

import check_random_state

from sklearn.model_selection

import train_test_split,cross_validate,GridSearchCV

from sklearn.metrics

 import accuracy_score,precision_score,recall_score,confusion_matrix

from sklearn.linear_model import LogisticRegression

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

from sklearn.neighbors import KNeighborsClassifier, KernelDensity

from sklearn.svm import SVC
```

```
from sklearn.ensemble
import RandomForestClassifier,GradientBoostingClassifier,AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from catboost import CatBoostClassifier
df=pd.read_csv('D:/Project/traindata.csv')
df.head()
df.shape
classes=df['Loan_Status'].value_counts()
plt.figure(figsize=(4,4))
plt.pie(classes,labels=['(Yes)','(No)'],autopct='%1.1f%%',startangle=120,colors=['t
omato','skyblue'])
plt.title('Comparison of Classes')
plt.show()
for col in numerical_data.columns:
    plt.figure(figsize=(10,5))
    plt.subplot(1,2,1)
    plt.hist(df[df['Loan_Status']=='N'][col],color='skyblue')
    plt.xlabel(f'{col}')
```

```
    plt.ylabel('Values')
    plt.title(f'Histogram Plot of {col} (Loan_Status: No)')
    plt.subplot(1,2,2)
    plt.hist(df[df['Loan_Status']=='Y'][col],color='tomato')
    plt.xlabel(f'{col}')
    plt.ylabel('Values')
    plt.title(f'Histogram Plot of {col} (Loan_Status: Yes)')
    plt.tight_layout()
    plt.show()
    for col in numerical_data.columns:
   if col not in ['Credit_History','Loan_Amount_Term']:
     plt.figure(figsize=(4,4))
     plt.boxplot(df[col])
     plt.xlabel(f'{col}')
     plt.ylabel('Values')
     plt.title(f'Box Plot of {col}')
     plt.show()
for col in numerical_data.columns:
   if col not in ['Credit_History','Loan_Amount_Term']:
```

```
    plt.figure(figsize=(4,4))
    plt.boxplot(df[col])
    plt.xlabel(f'{col}')
    plt.ylabel('Values')
    plt.title(f'Box Plot of {col}')
    plt.show()
df.replace({'Gender':{'Male':1,'Female':0}},inplace=True)
df.replace({'Married':{'Yes':1,'No':0}},inplace=True)
df.replace({'Dependents':{'3+':3}},inplace=True)
df.replace({'Education':{'Graduate':1,'Not Graduate':0}},inplace=True)
df.replace({'Self_Employed':{'Yes':1,'No':0}},inplace=True)
df.replace({'Property_Area':{'Rural':0,'Urban':2,'Semiurban':1}},inplace=True)
df.replace({'Loan_Status':{'Y':1,'N':0}},inplace=True)
correlation_matrix = df.corr()
```

**# Plot correlation matrix**

```
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
```

```
plt.show()

models = {
    'LogisticRegression': LogisticRegression(C=5,penalty='l2'),
    'SVC': SVC(C=25,kernel='poly'),
    'DecisionTreeClassifier': DecisionTreeClassifier(max_depth=100),
    'RandomForestClassifier': RandomForestClassifier(n_estimators=150),
    'GradientBoostingClassifier':
GradientBoostingClassifier(learning_rate=0.5,n_estimators=350),
    'AdaBoostClassifier': AdaBoostClassifier(learning_rate=1.5,n_estimators=300),
    'KNeighborsClassifier':
KNeighborsClassifier(metric='euclidean',n_neighbors=40,weights='distance'),
    'XGBClassifier': XGBClassifier(learning_rate=0.1,n_estimators=300),
    'CatBoostClassifier':
CatBoostClassifier(learning_rate=0.1,n_estimators=300,silent=True),
}

from sklearn.metrics import make_scorer, accuracy_score, precision_score,
recall_score,f1_score

scoring = {
    'accuracy': make_scorer(accuracy_score),
    'precision': make_scorer(precision_score),
```

```
    'recall': make_scorer(recall_score),
    'f1': make_scorer(f1_score)
}
scores = []
for model_name, model in models.items():
    print("-" * 50)
    print(f'{model_name}:')

    # Perform cross-validation
    cv_results = cross_validate(model,x,y,cv=4, scoring=scoring)

    # Collect the results
    model_results = {
        'Model': model_name,
        'Accuracy': np.mean(cv_results['test_accuracy']),
        'Precision': np.mean(cv_results['test_precision']),
        'Recall': np.mean(cv_results['test_recall']),
        'F1': np.mean(cv_results['test_f1'])
    }
```

```
    scores.append(model_results)
KNN=KNeighborsClassifier(metric='euclidean',n_neighbors=40,weights='distance'
)
KNN.fit(x_train,y_train)
y_hat=KNN.predict(x_test)
print("Accuracy:",accuracy_score(y_hat,y_test)*100,'%.')
print("Precision:",precision_score(y_hat,y_test)*100,'%.')
print("Recall:",recall_score(y_hat,y_test)*100,'%.')
print("F1_Score:",f1_score(y_hat,y_test)*100,'%.')
cm = confusion_matrix(y_test,y_hat)

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, cmap='Blues', fmt='g')
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.show()
def loan_status_predictor(input_array):
```

```
prediction=KNN.predict(input_array.reshape(1,-1))

if prediction == 0:

    return 'Loan Rejected'

else:

    return 'Loan Approved'
```

# CHAPTER 11

# OUTPUT

## 11.1 COMPARISION CLASSES

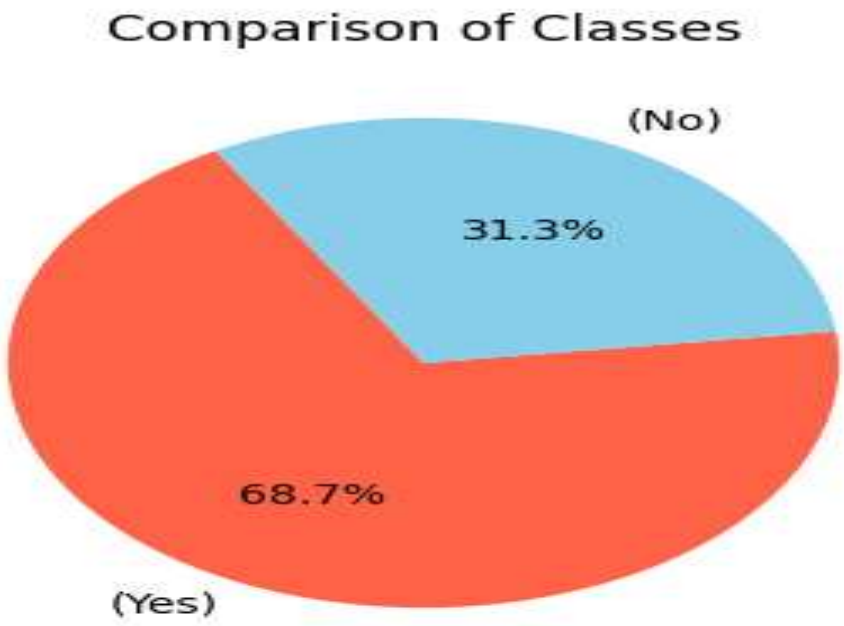


**FIG 11.1.1 COMPARISION CLASSES**

## 11.2 APPLICATION INCOME

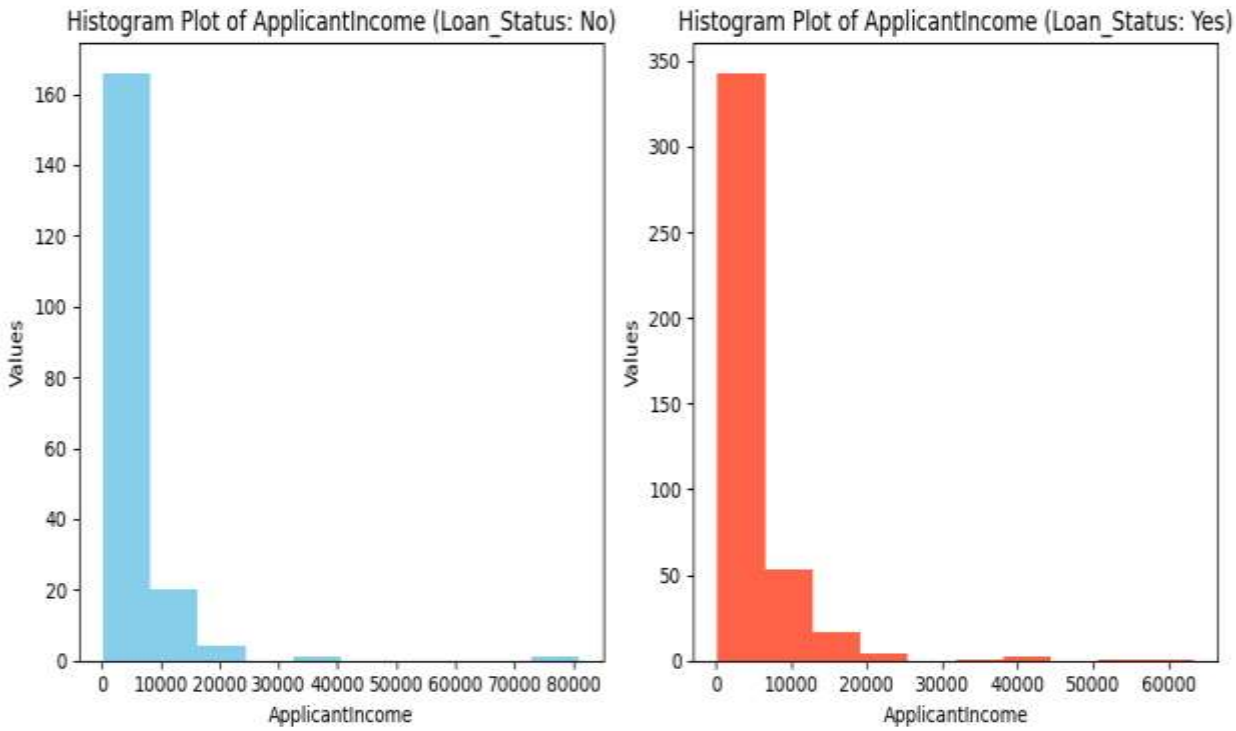


**FIG 11.2.1 APPLICAION INCOME**

## 11.3 GENDER WISE FOR LOAN STATUS

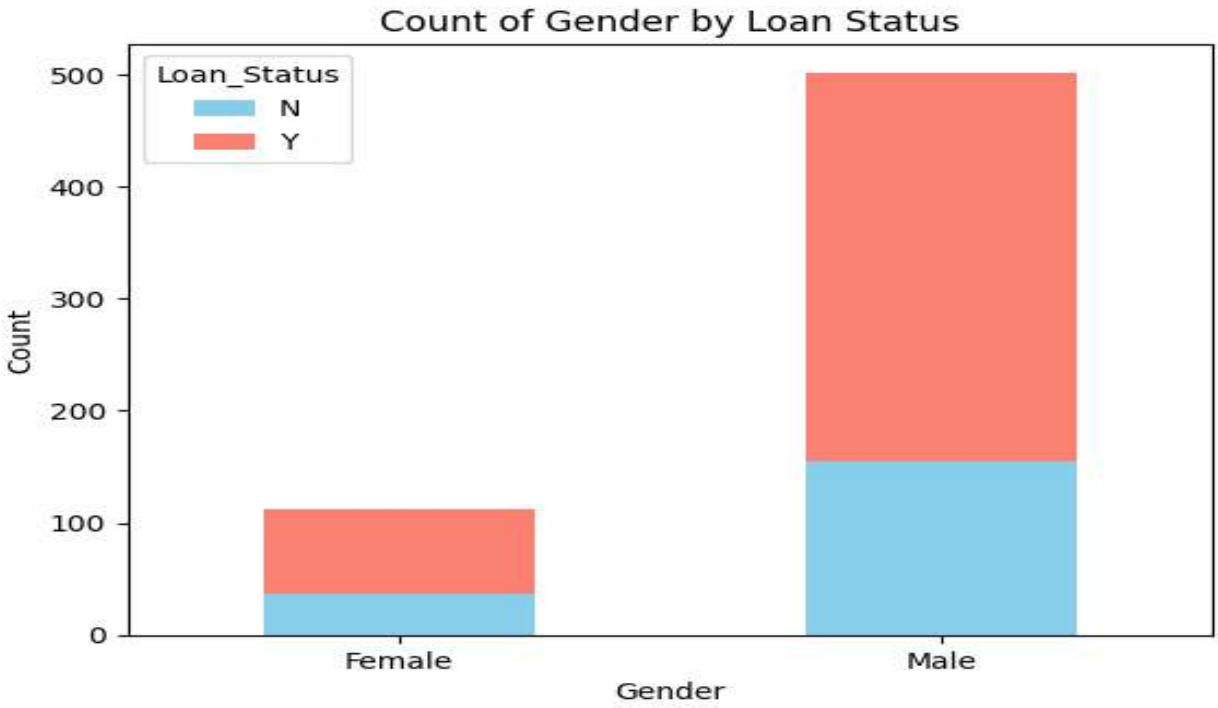


**FIG 11.3.1 GENDER WISE FOR LOAN STATUS**

## 11.4 HEAT MAP

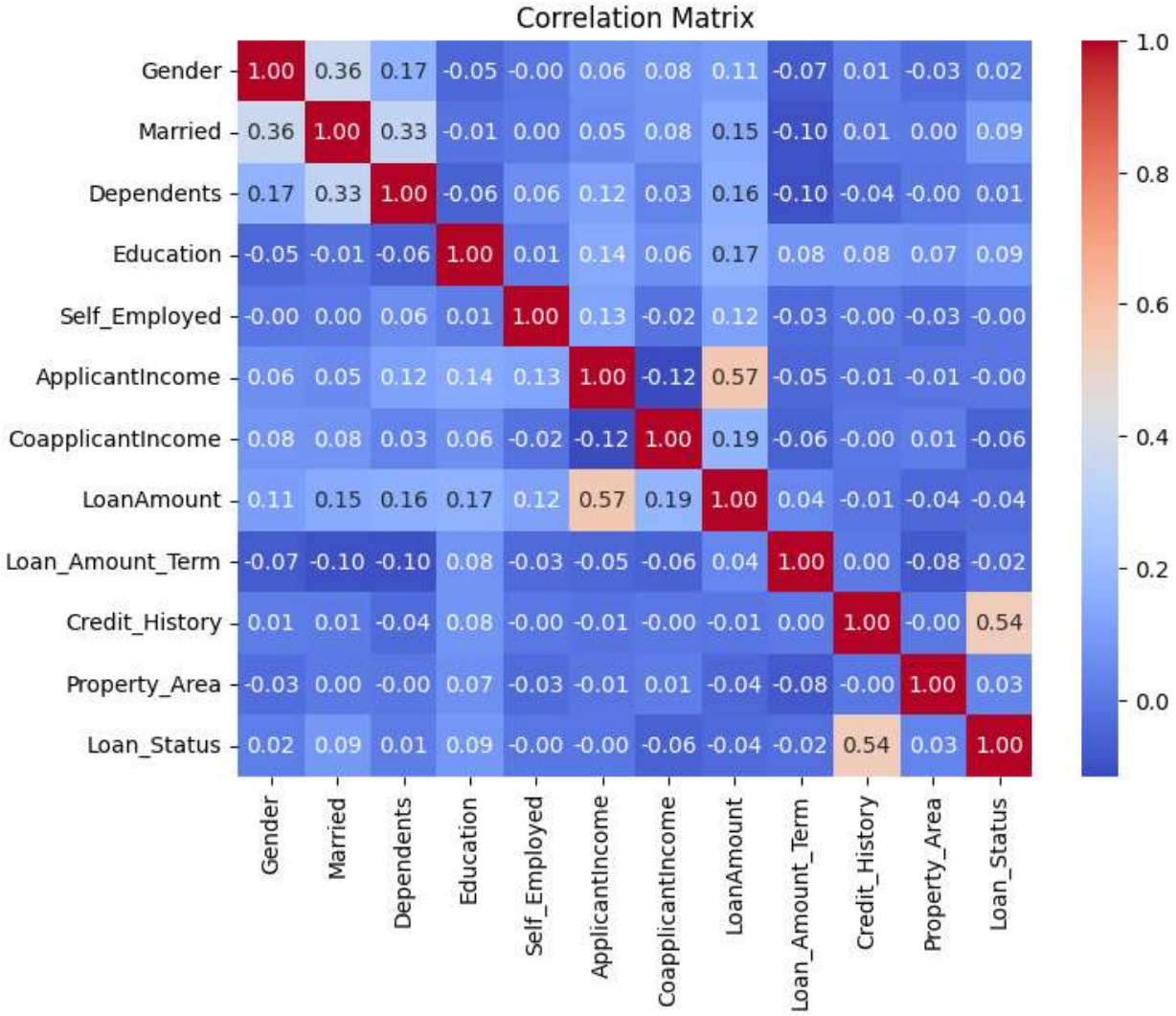


**FIG 11.4.1 HEAT MAP**

## 11.5 CONFUSION MATRIX

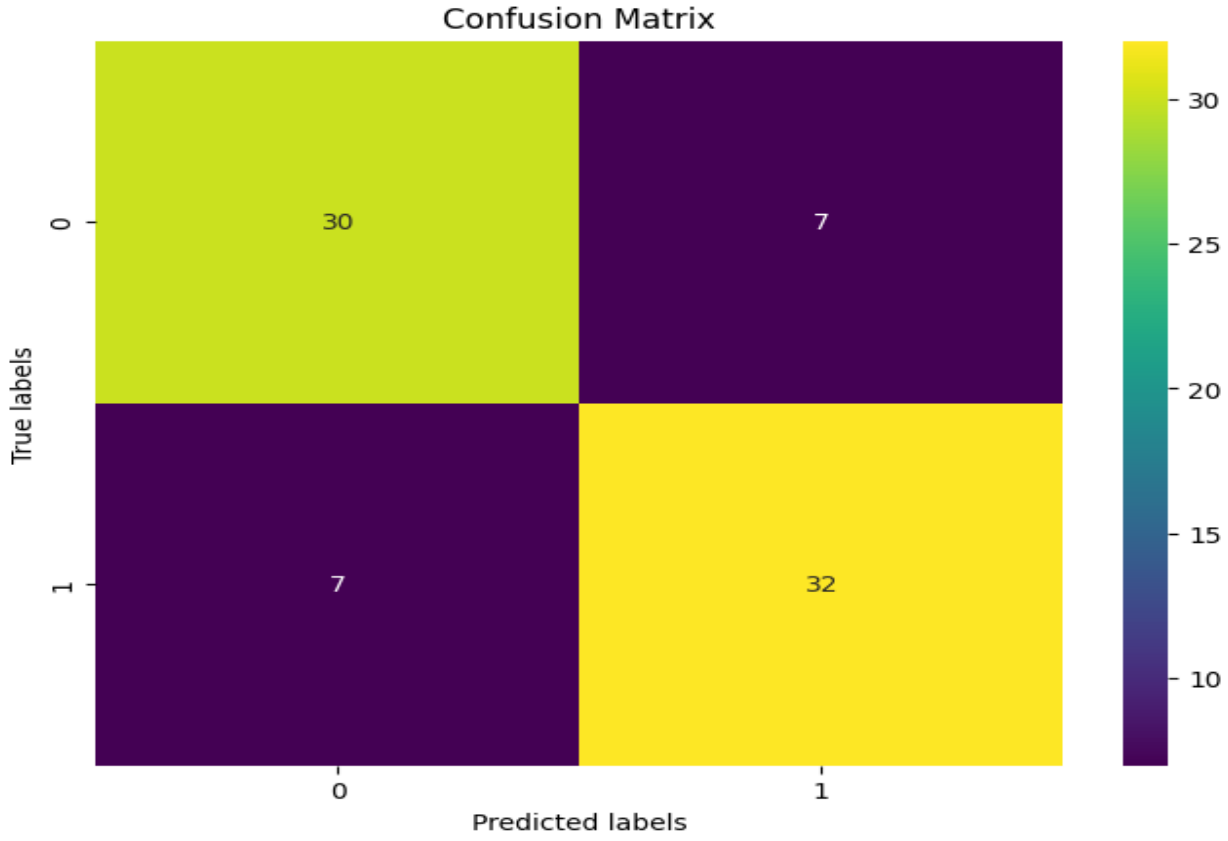


**FIG 11.5.1 CONFUSION MATRIX**

# CHAPTER 12

# CONCLUSION AND FUTURE ENHANCEMENT

## 12.1 CONCLUSION

In conclusion, implementing a loan price prediction project using machine learning (ML) involves leveraging data-driven techniques to accurately estimate the price of loans based on various borrower and loan-related features. By following a systematic approach, including data collection, preprocessing, model selection, training, evaluation, and deployment, it's possible to build robust ML models that can provide valuable insights for lenders and borrowers alike.

Through the use of supervised learning algorithms such as linear regression, decision trees, random forests, or gradient boosting algorithms like XGBoost or LightGBM, coupled with appropriate feature engineering techniques, it's possible to develop models capable of predicting loan prices with reasonable accuracy. These models can take into account factors such as loan amount, interest rates, borrower credit scores, employment status, and economic indicators to make informed predictions.

However, it's essential to recognize the limitations and challenges associated with loan price prediction using ML. These include the need for high-quality, representative data, potential biases in the data, regulatory considerations, and the dynamic nature of financial markets and borrower behavior.

Despite these challenges, ML-based loan price prediction models offer significant potential benefits, including improved risk assessment, more efficient loan pricing strategies, and enhanced decision-making for both lenders and borrowers.

## 12.2 FUTURE ENHANCEMENT

For future enhancements in loan price prediction using machine learning (ML), several avenues can be explored to improve model accuracy, efficiency, and usability. Here are some potential areas for enhancement:

**Incorporating Alternative Data Sources:** Expand the scope of data used for prediction by incorporating alternative data sources such as social media activity, transaction history, or non-traditional credit scoring data. These additional sources can provide richer insights into borrower behavior and creditworthiness, leading to more accurate predictions.

**Ensemble Methods:** Explore the use of ensemble learning techniques such as bagging, boosting, or stacking to combine predictions from multiple models. Ensemble methods often yield better performance than individual models by leveraging the strengths of different algorithms and reducing the risk of overfitting.

**Deep Learning Architectures:** Investigate the application of deep learning architectures such as neural networks, convolutional neural networks (CNNs), or recurrent neural networks (RNNs) for loan price prediction. Deep learning models have shown promise in capturing

complex patterns in data and may uncover hidden relationships that traditional ML algorithms may miss.

**Explainable AI (XAI):** Enhance model interpretability by incorporating techniques from explainable AI (XAI). By providing transparent explanations of how the model makes predictions, stakeholders can better understand the factors influencing loan prices and build trust in the model's decisions.

**Dynamic Pricing Models:** Develop dynamic pricing models that can adapt to changing market conditions, borrower profiles, and risk factors in real-time. Incorporate mechanisms for continuous learning and updating of the model to reflect the latest trends and insights in the lending landscape.

**Fairness and Bias Mitigation:** Address potential biases in the data and model predictions by implementing fairness-aware ML techniques. Ensure that the model's predictions are fair and unbiased across different demographic groups to promote ethical lending practices and mitigate discriminatory outcomes.

**Robustness and Security:** Enhance the robustness and security of the ML model against adversarial attacks, data breaches, or malicious manipulation. Implement techniques such as model robustness testing, adversarial training, and privacy-preserving methods to safeguard sensitive borrower information and ensure the integrity of the model.

**Integration with Decision Support Systems:** Integrate ML-based loan price prediction models with decision support systems used by lenders to streamline loan approval processes, automate risk assessment, and optimize portfolio management. Provide actionable insights and recommendations based on the model's predictions to facilitate informed decision-making by loan officers and underwriters.

By focusing on these areas for future enhancement, organizations can further leverage the power of ML to improve loan price prediction accuracy, fairness, and efficiency, ultimately driving innovation and transformation in the financial industry.

# CHAPTER 13

## REFERENCES


1. Brown, M., Lee, D., & Virgillito, M. E. (2017). Predicting Loan Defaults Using Machine Learning Techniques. SSRN Electronic Journal.

2. Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. European Journal of Operational Research, 183(3), 1447-1465.

3. Dey, S., Reddy, C. K., & Bhattacharya, S. (2017). Predictive modeling for loan prediction using machine learning techniques. International Journal of Computer Applications, 160(12), 10-15.

4. Giromini, M., & Anselma, L. (2018). A comparison of machine learning models for the prediction of credit scoring. European Actuarial Journal, 8(2), 321-349.

5. Gu, J., & Xiong, J. (2018). Loan Risk Prediction Model Based on Machine Learning. In 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA) (pp. 574-577). IEEE.

6. Hand, D. J., Henley, W. E., & Goovaerts, M. J. (1997). A k-nearest neighbour classifier for assessing consumer credit risk. The Statistician, 46(1), 87-100.

7. Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. Journal of the Royal Statistical Society: Series A (Statistics in Society), 160(3), 523-541.

8. He, Z., Xu, X., & Deng, S. (2017). Loan risk prediction based on machine learning algorithm. In 2017 International Conference on Service Systems and Service Management (ICSSSM) (pp. 1-4). IEEE.

9. Kavitha, K., & Kathirvalavakumar, T. (2018). Loan prediction using machine learning algorithms. In 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1-4). IEEE.

10. Li, Y., Bao, Y., & Liu, Z. (2019). Loan risk prediction model based on machine learning algorithm. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 3135-3139). IEEE.

11. Luo, Z., & Yang, Y. (2019). Loan Prediction Based on Machine Learning. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 3054-3060). IEEE.

12. Rangapuram, S. S., Becker, C., Cichy, A., & Berghoff, M. (2018). Deep Learning for Loan Risk Prediction: Model Construction and Deployment. arXiv preprint arXiv:1807.00459.

13. Shen, F., Sun, X., & Deng, J. (2018). Loan Prediction Based on Machine Learning. In 2018 IEEE International Conference on

Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 677-681). IEEE.

14. Shouman, M., Turner, T., & Stocker, R. (2017). Using deep learning for loan risk prediction. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 2376-2383). IEEE.

15. Song, Q., & Qu, Y. (2019). A Machine Learning Approach for Loan Risk Prediction. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 793-796). IEEE.

16. Sreeja, N., & Deepak, S. (2019). Machine Learning Approach for Loan Prediction. In 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN) (pp. 1-5). IEEE.

17. Sun, X., & Shen, F. (2019). Machine Learning Algorithm for Loan Prediction. In 2019 IEEE 3rd International Conference on Computer and Communication Systems (ICCCS) (pp. 1-5). IEEE.

18. Tsai, C. F., Chou, W. C., Lai, C. F., & Su, Y. F. (2011). Bankruptcy prediction by hybrid neural networks and hybrid support vector machines. Expert Systems with Applications, 38(8), 9975-9981.

19. Wan, Y., & Zuo, B. (2019). Loan Risk Prediction Based on Machine Learning Algorithm. In 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 682-685). IEEE.

20. Wang, Y., Sun, Y., Li, Y., & Gao, F. (2018). Loan Risk Prediction Based on Machine Learning. In 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) (pp. 779-782). IEEE.

21. Xiang, Y., Zeng, W., & Han, Z. (2017). Loan Risk Prediction Based on Machine Learning. In 2017 4th International Conference on Information Science and Control Engineering (ICISCE) (pp. 1460-1463). IEEE.

22. Xu, Y., & Xiong, Z. (2019). Loan Prediction Based on Machine Learning. In 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 690-693). IEEE.

23. Yang, J., & Lin, C. (2019). Loan Prediction Based on Machine Learning. In 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 709-712). IEEE.

24. Zhang, S., & Deng, Y. (2019). Loan Prediction Based on Machine Learning. In 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 720-724). IEEE.

25. Zhou, S., Zeng, J., & Dong, W. (2019). Loan Prediction Based on Machine Learning. In 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 730-734). IEEE.