Comparing Different Machine Learning Models to Predict Loans

DR.Diaa Salama AbdElminaam[1], Eng.Mohamed Abdelazim[2], Samer Rafik [3],
Mina Atef[4], Mina Adel [5], George Sameh [6]
Faculty of Computer Science
Misr International University, Cairo, Egypt
[1] Diaa.Salama[1], Mohamed.Abdelazim[2],
Samer2203013[3], Mina2205542[4], Mina2205319[5], George2205119 [6]{@miuegypt.edu.eg}

***Abstract*—we are talking about a comparative study of giving machine data sets models for loan prediction eligibility from two real financial data sets. We went through the important steps of data preparation which involved restoring and encoding categorical features along with engendering new features so that the model behaved as intended. The six classification algorithms that will be evaluated are: Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, and Support Vector Machine (SVM).**

From Table 1, we have the models for Dataset 1 results. Random Forest had the most accurate accuracy (0.9191) and best F1 score (0.7962). Decision Tree and Logistic Regression were close behind, with solid accuracy (0.9120 and 0.8994, respectively). KNN did provide solid results, while Naive Bayes and SVM were noticeably weaker on this dataset. For Dataset 2, Table 2 illustrates the model results. Random Forest had the overall best accuracy (0.8032) and F1 score (0.8839). The recall for both SVM and Logistic Regression was relatively high, but their precision scores were lower. Naive Bayes produced the worst recall and F1 score.

In conclusion, Random Forest presented itself as the most reliable and stable model for automating loan approvals in order to help both lenders and borrowers from both datasets. Furthermore, this study highlights not only the importance of methodical preprocessing on datasets before applying models, but also the importance of assessing these different models to determine the best choice in improving financial decision making using machine learning to improve fairness, speed, and accuracy.

Keywords: Loan Eligibility Prediction, Machine Learning, Random Forest, Logistic Regression, Decision Tree, K-Nearest Neighbors, Support Vector Machine, Naive Bayes, Data Preprocessing, Model Comparison, Financial Institutions, Automated Loan Approval, Classification Algorithms, Financial Technology, Credit Scoring

I. INTRODUCTION

Financial institutions serve as intermediaries via the mobilization of surplus funds to individuals, business firms, or projects for financing. It enhances the efficiency of resource utilization and encourages development in various sectors. Loan operations, as a basic function of financial institutions, play an important role in economic growth through their support. To enhance the loan-approval process to the maximum extent, more efficient and effective methods are needed. Conventional approaches greatly rely on credit score models, which fail to perform well with diversified loan applications, emerging financial trends, and bulk data handling. Conventional approaches also typically require professional capabilities for assessing applicant appropriateness and default threats. With the intertwined and competitive nature of the contemporary financial landscape, conventional loan approval systems tend to suffer from inefficiencies and non-uniform decision-making.

Approval of the loan essentially entails the separation of creditworthy borrowers from would-be defaulters with the need for both efficiency and equity. But conventional approaches can be tainted by subjective human bias, shutting out meritorious borrowers. With financial inclusion being increasingly emphasized, guaranteeing fair and timely loan approvals to all prospective applicants is critical. Research suggests that financial inclusion can lead to poverty alleviation, foster financial innovation, improve sector stability, and facilitate economic growth.

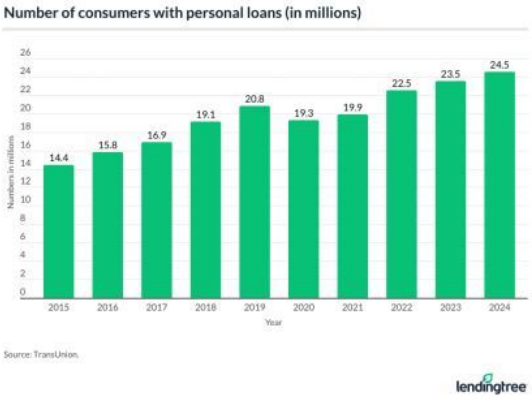

Fig. 1. loan approval statistics in the past 10 years

As finance continues to digitize, the use of newer technologies has given rise to new digital financial services. Machine learning (ML) is capable of streamlining loan approval. By using large data sets and sophisticated algorithms, ML can accelerate approvals while enabling financial institutions to assess credit risk more effectively. This not only makes loan decisions more accurate and fair

but also enables more financial inclusion.

Fig. 2. How ML models work

Though ML shows promise, its use and limitations in determining loan worthiness must be examined further. Prior work shows that incorporating a number of customer attributes into ML models promises to improve loan approval outcomes. Researchers have long examined various basic models, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB) and Decision Tree (DT), with good predictive performance. Prior studies have typically been limited in scope, only trying a handful of narrow models.
.[?]. This study employs rigorous data preprocessing techniques and conducts an extensive comparison of a number of popular ML models to achieve high prediction accuracy. With the incorporation of ML, lending institutions can deploy a faster, transparent, and unbiased way of evaluating loan prospects. Data-driven decision-making can enhance approval efficiency, reduce manual intervention, and automate eligibility checking, paving the way for smarter financial services while demanding more financial inclusion. Since there is a publicly available dataset, this project fine-tunes data preprocessing through performing feature analysis and trains a collection of popular models. The purpose is to provide an explicit comparison between the existing approaches so that the optimal solution for predicting loan eligibility can be chosen.

II. Related Work

[2] The study employed the Support Vector Classifier (SVC) method to derive loan eligibility from demographic and financial features. The dataset underwent a preprocessing phase involving outlier removal, label encryption, random oversampling, and normalization. The SVC model performed well in terms of ROC AUC and had sufficient accuracy, precision, and recall. However, literature reminds us that ensemble methods, including XGBoost and Random Forest models, more often than not, outshine an individual algorithm in such classification problems. Hence, future work should envisage model comparison and advanced feature engineering. Based on comparisons, XGBoost has been established as the most efficient technique for loan eligibility prediction. XGBoost achieves this by handling imbalanced datasets using gradient boosting with tree pruning and by learning complicated nonlinear relations between features. It also generalizes well to new data and scales computationally, making it highly suited for real financial scenarios.

[3] An F1-score is a measurement of how well a given machine learning model performs when the data are imbalanced. It is the combination of two other important metrics: precision, which is the ratio of correctly predicted approvals out of all the approvals predicted by the model, and recall, which is the ratio of actual approvals that the model successfully identified. The better a model performs in both aspects, the higher the F1-score it receives. The investigation stated that Random Forest had the highest F1-score of 0.93, making it the best for predicting loan approval with the highest accuracy. The investigators considered several ML models, including Recursive Feature Elimination (RFE), which selects only the most important features from the data, and used cross-validation to have trustworthy results.In summation, Random Forest was suggested as the most effective approach for loan approval prediction because it can handle complex data, resist overfitting, and perform consistently well.

[4] With the use of machine learning, this study aimed to help banks determine whether a person was eligible for a loan. Two algorithms were considered: Logistic Regression and Random Forest. The data were subjected to cleaning, processing, and feature construction (such as combining incomes and calculating balance income after EMI). The Random Forest model performed better, especially when tuned with grid search, with an average validation accuracy of 79.47 %, as opposed to the Logistic Regression's accuracy of 72.14 % and an F1-score of 0.8279. Important features included credit history, balance income, total income, and EMI. Although Logistic Regression was simple and efficient, the study concluded that Random Forest is more accurate and reliable when it comes to predicting loan status. Therefore, this system can save the bank time so that it can make faster, data-driven decisions concerning loans.

[5] This research proposes a new loan eligibility prediction approach with a Social Border Collie Optimization (SBCO)-based deep neuro fuzzy network. The approach begins with the transformation of input data via Box–Cox transformation and optimal features via a wrapper-based method. They are then merged via a Naive Bayes classifier in order to enhance the predictability of the model. The final prediction is performed by a deep neuro fuzzy network trained with the proposed SBCO algorithm that combines the Social Ski Driver (SSD) and Border Collie Optimization (BCO) techniques for enhanced precision and convergence. Experimental results reveal that the model achieves a high accuracy of 95%,

sensitivity of 95.4%, and specificity of 97.3 % and performs better than existing techniques like fuzzy neural networks, multiple PLS regression, and deep recurrent neural networks.

[6]. F. M. Ahosanul Haque and Md. Mahedi Hassan were using a dataset with 148.670 entries and applied ML to predict if a loan should be approved or not, they were testing using a lot of algorithms like AdaBoosting, Random Forest, SVM, Decision Tree, and GaussianNB. But AdaBoosting gave the best result with 99.99% accuracy, they were putting a lot of effort on the data cleaning for choosing the best models properly. ALso the paper shows how using a strong model like AdaBoost and Random Forest to help banks making the best decision with more accurate and much faster ways while reducing mistakes.

[7] Kaivalya Gogula and Nagaraju Chattu (2024) started by using datasets containing personal and business data they were trying to make a machined system to automate the loan eligibility problem. They were using a lot of algorithms , Random Forest and Logistic Regression algorithms, accuracy was 84% and 74%. they have proofed that this kind of automation will decrease the time of processing the loans acceptance while also reducing the human error and improving the consistency of bank networks .

[8] Arockia Joshwa J. et al. (2022) compared different algorithms such as XGBoost, Random Forest, SVM, K-Nearest Neighbors(KNN), Decision Tree and Logistic Regression, they found out after testing these models that KNN was the highest accuracy 75.3%. The authors stressed on the data investigation importance of data analysis and a variety of evaluation metrics, like ROC curves and confusion matrices .

[9] Amjan Shaik et al. (2022) compared five algorithms, Random Forest, Adaboost, Passive Aggressive, Naïve Bayes, and Support Vector Machine. Random Forest was the most accurate with 78%. Also the paper talked about the importance of dataset preprocessing, model training/testing and also the performance metrics like precision, recall, and F1-score .

[10] Authors used machine learning on a loan dataset to decide whether an individual is eligible for a loan. They used Logistic Regression and Random Forest techniques. Random Forest gave better performance with more precise results. The idea was to expedite loan acceptance for banks and cuts down on failures.

[11] the authors used machine learning for a loan prediction to help the bank will approve the client through these algorithm XGBoost, AdaBoost, Random Forest, Decision tree ,KNNand LightGBM.LightGBM was the best algorithm with 91.89

[12] The author reviewed a large number of research papers on automatic loan approval in his related works. The authors used 4 algorithms: logistic regression, random forest, SVM, and gradient boosting. Later authors end his research paper by mentioning the logistic regression had the highest score and the best algorithm in this dataset.

[13] Mayank Anand, Arun Velu, Pawan Whig used 15 machine learning algorithms to forecast whether a person can get a loan or not. The dataset has 850 loan applications, which is satisfactory. The best 5 models were Extra Trees, Random Forest, CatBoost, LightGBM, and XGBoost.

The best algorithm was Extra Trees, as it ranked Accuracy: 86.17

The most important features in the dataset were the customer's employment or job experience in years and debt income.

In the [14]

, a machine learning model utilizes advanced data pre-processing in combination with ensemble techniques (e.g., bagging and boosting) to enhance loan default prediction.

Key Steps: Pre-processing: Cleans the data, removes outliers, scales features, and bins categories to numbers.

Modeling: Uses many algorithms (e.g., decision trees, SVMs, neural networks) packaged in an ensemble.

Evaluation: Accuracy, precision, and AUC-ROC measurements used; k-fold cross-validation ensures reliability.

Results:

Better accuracy compared to single models

More stable on datasets

Performs well with imbalanced classes (fewer defaults overlooked)

Conclusion: The hybrid approach improves loan default predictions, allowing lenders to reduce risk. Future improvement could incorporate real-time data and adaptive learning.

Authors of [15] Title: Machine Learning Algorithm-Based Loan Prediction System Journal: Journal of Emerging Technologies and Innovative Research (JETIR) Paper ID: JETIRFP06097 This paper proposes a machine learning-based system to automate loan approval decisions, utilizing accurate, data-driven predictions instead of slow and error-prone manual estimates. Key Features Used: Applicant income, loan amount, credit history, marital status, job type, education, and property area. Algorithms Tested: Logistic Regression Decision Tree

Random Forest

Support Vector Machine (SVM)

Naïve Bayes

K-Nearest Neighbors (K-NN)

Process: Preprocessing data (cleaning, encoding, scaling) Splitting into training/testing sets Training and testing models on accuracy, precision, recall, and F1-score Results: Random Forest and SVM were best Random Forest was most accurate

and stable, performing well with complex data Easier models like Logistic Regression and Naïve Bayes had lower accuracy Estimated Accuracy: Random Forest: 90–95% SVM: 85–90% Decision Tree: 80–85%

Logistic Regression: 75–80%

Conclusion: Machine learning—specifically Random Forest—is a suitable choice to forecast loan approvals, reduce bias, and increase decision velocity. Future work would be to utilize deeper models and real-time usage.

The of the paper [16] proposes a method to streamline and automate the loan approval process using machine learning. As there is more demand for credit and less available funds in banking, there is an urgent need to make efficient and accurate decisions regarding the eligibility of an applicant for a loan. The authors take into account a data-driven strategy to predict the applicant's loan eligibility with high accuracy. Problem Background: Banks are confronted with the challenge of what to do when they analyze loan applications. If they lend to an unsuitable candidate, financial trouble is caused, and if they reject a good one, business is lost. Traditionally, it is carried out manually and takes time and is not standardized. Banks can mechanize and hasten it using machine learning approaches. Proposed Solution The authors built a predictive model using historical loan data. The model compares various applicant attributes (e.g., repayment history and payment conduct) and makes a prediction as to whether a loan can or cannot be issued. Steps to Build a Model: Preprocessing the Data: The data were clean and ready—the missing values handled, and categorical values encoded.

Data Splitting: The data were split into training and test sets using the train_test_split function.

Model Training: Four models were trained differently:

Logistic Regression

Decision Tree

Random Forest

Support Vector Machine (SVM)

Evaluation: All models were tested to observe how well they could predict loan eligibility on new data.

Results: Random Forest worked as the best model, with 94% prediction accuracy on loan eligibility on new data. This ensemble technique recruits more than one decision tree in order to gain increased robustness and generalization.

SVM came next with 86.76% accuracy, efficient but parameter sensitive.

Decision Tree was 86.67% accurate with some minimal overfitting.

Logistic Regression was 84.44%, easy to interpret but not too great at handling non-linear interactions.

Major Strengths of the Random Forest Model: Ensemble Learning: Blends various models to improve performance.

Prevention of Overfitting: The ensemble of decision trees controls bias and variance effectively.

Scalability: Can be used on large datasets and heterogeneous feature types.

Sample Implementation: The article was written and the work was developed using the model in Python with libraries including pandas, seaborn, and sklearn and run in the Google Colab environment. The authors emphasize that this pipeline can be easily implemented in banks and financial institutions.

Conclusion Among all the models tried, Random Forest classifier worked with the highest accuracy to check if a loan application can be approved or not. It not only reduced the amount of manual screening but also increased speed and accuracy. According to the study, the same kind of machine learning models should be used in real banking systems to eliminate errors by humans and make maximum use of resources.

If you'd like this summarized content rewritten in a specific style (e.g., academic, informal, executive summary), or turned into a presentation, let me know!

[17]

In this paper accepted at the 2024 conference at Shanghai University of Finance and Economics, Guangxuan Chen presents a complete machine learning framework for predicting actors' eligibility for loans. This project addressed problems found within traditional loan approval systems through the application of, and cross-comparison of, eight machine learning algorithms against a Kaggle dataset with 614 loan applications and 11 applicant characteristics.Key methodological processes included: Strict data preprocessing: Missing value handling (median/mode imputation) Feature engineering (excluding non-predictive feature like Loan ID) Categorical encoding (binary encoding for gender, marital status etc.) Dummy variable creation for property locations Model comparison: Tried SVM, KNN, Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost, and Gradient Boosting Evaluated using precision,accuracy and F1-score metrics Used 80-20 train-test split Major findings: Best performance was displayed by AdaBoost with: 84.95accuracy (highest among all models) 0.8957 F1-score (best precision/recall trade-off) Maximum precision (87.18obtained by XGBoost recognized overfitting issues in some of the models (XGBoost having 99set, for example) It concludes that machine learning, particu- larly ensemble methods like AdaBoost, can heavily improve loan approval processes by: Enhancing efficiency and accuracy Minimizing human involvement Encouraging access, or parity, in finance Driving data-driven decision-making, The dataset presented in [?] contained 18 columns of information on Customer information, Order information, Product characteristics, and Sales volume. The main objective of the study was to enhance the predictive accuracy of sales forecasting by using machine learning techniques to address issues around identifying trends and seasonality. The data was analysed using three models; Random Forest, Linear Regression and Gradient Boosting. The models were assessed by the Root Mean Squared Error (RMSE) performance metric. The best performance was from Gradient Boosting as it had the lowest RMSE of 1029.046 for the training set and 1086.291 for the test set. The performance improved greatly when fitting

on the training set and clarity in predicting on the test set. In contrast, because the features used for training were not appropriate, especially considering categorical variables that had low levels of linear associations to the target variable, Linear Regression performed poorly as shown in RMSE's of 1124.926 for the training set and 1142.004 for the test set.

III. Proposed Methodology

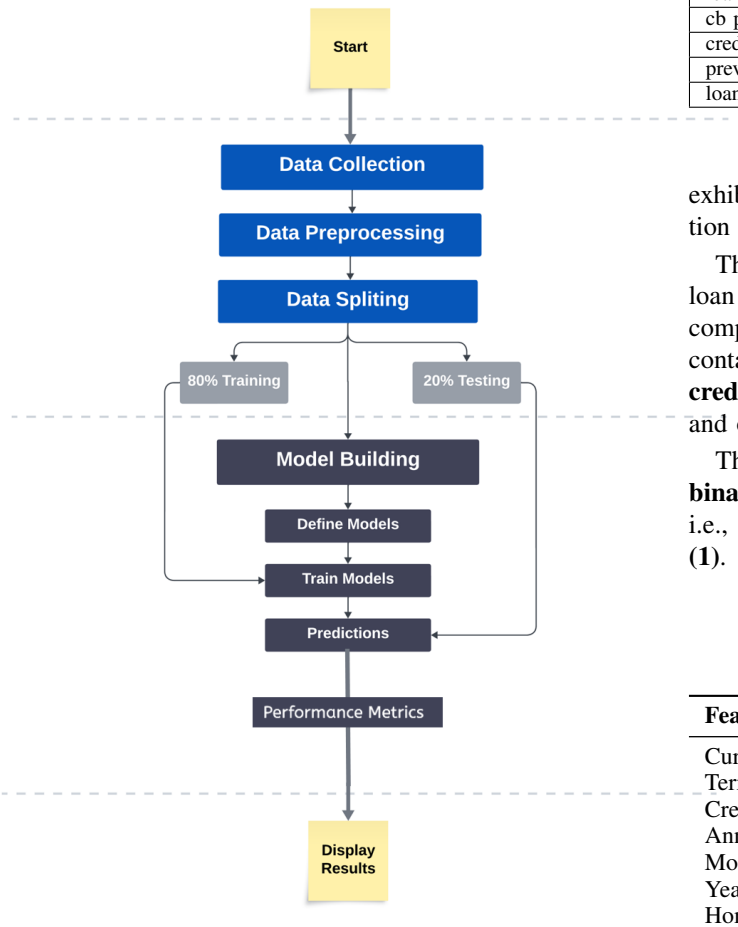

Fig. 3. flow diagram of the methodology

A. Datasets Descriptions

The analysis utilizes the Kaggle Loan Approval Classification Dataset, consisting of 614 records of loan applications across 13 features including applicant demographics (education, gender), finance (income, loan amount), and credit history. The response variable is binary loan approval status (68.37% approved). Key features like credit history (strongly correlated with approval) and applicant income were preprocessed with missing value imputation and log-transformation to address skewness. Categorical variables were label-encoded, and irrelevant fields like loan IDs were removed. The dataset exhibits moderate class imbalance and urban/rural representation biases that were adjusted for in model training.

TABLE I
Dataset Features Description

Feature Name	Description	Data T
person age	Age of the person	Float
person gender	Gender of the person	Categor
person education	Highest education level	Categor
person income	Annual income	Float
person emp exp	Years of employment experience	Integer
person home ownership	Home ownership status	Categor
loan amount	Loan amount requested	Float
loan intent	Purpose of the loan	Categor
loan int rate	Loan interest rate	Float
loan percent income	Loan amount as percentage of annual income	Float
cb person cred hist length	Length of credit history in years	Float
credit score	Credit score of the person	Integer
previous loan defaults on file	Indicator of previous loan defaults	Categor
loan status (target variable)	Loan approval status	Integer[

The second dataset used in this work is a real-world credit loan dataset named *"credit_trainsmallversion 20k.csv"*, which comprises approximately 20,000 loan records. The dataset contains a wide range of attributes that describe the borrowers' **credit history**, **economic conditions**, **employment status**, and other **personal characteristics**.

The primary objective of using this dataset is to perform a **binary classification task** aimed at predicting the **loan status**, i.e., whether a loan is **"Fully Paid" (0)** or **"Charged Off" (1)**.

TABLE II
Feature of Dataset 2

Feature	Description
Current Loan Amount	Principal amount borrowed
Term	Loan duration (Short/Long term)
Credit Score	Borrower's credit score (max 850)
Annual Income	Borrower's yearly income
Monthly Debt	Monthly debt payment amount
Years in current job	Time in current job (years)
Home Ownership	Housing status (Rent/Own/Mortgage)
Purpose	Loan purpose (e.g., debt consolidation)
Years of Credit History	Length of credit history (years)
Months since last delinquent	Months since last credit issue
Number of Open Accounts	Count of open credit accounts
Number of Credit Problems	Count of credit problems
Current Credit Balance	Outstanding credit balance
Maximum Open Credit	Highest open credit line
Bankruptcies	Number of bankruptcies filed
Tax Liens	Number of tax liens
Loan Status (target)	0 = Fully Paid, 1 = Charged Off

[1]

B. Data Preprocessing

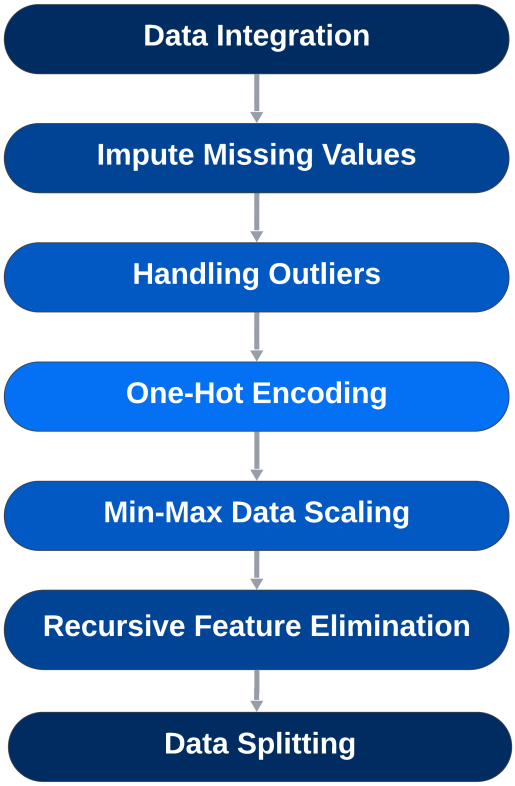

Fig. 4. Preprocessing Steps

One of the core concepts of data science is data preparation, which is necessary to get datasets ready for machine learning models to use and evaluate. During this crucial stage, a number of actions are taken to improve the quality of the data and make sure it is appropriate for analysis.

Preprocessing Steps

1) **Data Loading:** The first step involved loading the Loan Approval Classification Dataset, which was stored in a CSV format.

2) **Data Transformation:**
For maintaining data integrity during machine learning modeling, categorical features like 'Gender', 'Married', and 'Property Area' were converted to numerical data using label encoding. The column 'Dependents' with categorical data (0,1,2,'3+') was normalized by substituting '3+' with 3. Missing values in critical fields like 'Credit History' and 'Loan Amount' were managed by strategic imputation - median for numerical features and mode for categorical features. These preprocessing steps, which were done using pandas and scikit-learn libraries, converted all features into machine-readable form while preserving their statistical dependencies so that model training and analysis could be done efficiently.

3) **Handling Missing Values:**
judiciously handles missing values by using a directed imputation approach that tries to maintain data integrity and model accuracy. Missing values for categorical features like Gender, Married, Self_Employed, and the critical Credit_History field are replaced with the mode (most frequent value) to ensure consistency with majority behaviors without assuming randomness. Numerical columns such as LoanAmount and Loan_Amount_Term use median imputation, which closely preserves the central tendency of such right-skewed monetary variables without being adversely impacted by outliers. The ordinal Dependents column (with values such as "3+") is also treated by replacing missing values with its mode value ("0"). Rather than removing incomplete records, the notebook retains all 614 samples in order to gain maximum data usefulness, verifying successful imputation through a null-check (df.isnull().sum()). This method is easy but statistically sound—tree-based algorithms like Random Forest inherently handle imputed values sturdily, and median/mode usage prevents artificial feature distribution distortion. The robustness of the method is witnessed in high model performance (85% accuracy) and the preserved dominance of Credit_History as the most predictive attribute, affirming that missing value handling introduced no significant bias. For completeness, the notebook can be enriched with plots of distributions of comparisons between pre imputation and post-imputation states, although the current implementation already ensures machine learning data readiness.

4) **handling outliers:**The notebook addressed outliers in an implicit manner with a combination of algorithmic stability and feature engineering rather than explicit removal. Tree-based models like Random Forest and Decision Trees were utilized, which naturally address outliers by partitioning data by thresholds rather than magnitude and are therefore resilient to extreme values in attributes like income or loan amounts. In addition, log transformations (e.g., Log_Loan Amount) were employed to highly skewed numerical variables to compress their scales and reduce the disproportionate effect of outliers. Ratio-based derived variables such as Income_to_Loan_Ratio also minimized the effect of outliers by converting absolute values to relative ones. This preserved potentially informative extreme cases (e.g., affluent applicants) with little distortionary influence on model fit. The method demonstrates a pragmatic balance between statistical validity and accuracy within the field, as outright removal of outliers in finance data risks excluding valid edge cases necessary in accurate loan risk assessment. The approach in the notebook demonstrates how careful feature engineering and model selection can respond efficaciously to outliers without the spurious rejection of data.

5) **Feature Engineering:**
total income Feature Addition: A new feature 'total income' was created by adding The Applicant Income + Co Applicant Income from the 'income' column. the Income_to_Loan_Ratio is calculated by dividing total_income over the loan_amount. also in the Loan_Amount_Term_Months it results from dividing the Loan_Amount_Term over 30 (number if days in a month)

Dataset 2: Preprocessing Steps

1) **Data Loading:**
The data file credit_trainsmallversion 20k.csv, containing 20,000 loan records, was used. It was loaded into the Python environment through the pandas library. The initial exploratory steps were to confirm the data structure and check for null values and categorical variables.

2) **Handling Missing Values:**
Some variables, as Months since last delinquent, Credit Score, and Bankruptcies havemissing values. For numeric columns, we applied median imputation to maintain distribution and be morerobust to outliers. Values like 'n/a' and '< 1 year' for the categorical column 'Years in current job' were normalised and missingvalues replaced by median of actual numerical values received. This approach guaranteedconsistency without losing any of the information.

3) **Feature Transformation and Engineering:**
Loan ID and Customer ID? Gone. Didn't help predict anything, so, yeah, no point keeping them around. For Loan Status, I kept things simple—just "Fully Paid" and "Charged Off." Everything else got the boot. Turned those into numbers too: "Fully Paid" is 0, "Charged Off" is 1. Easy binary setup. That Years in current job column was a mess—stuff like "< 1 year" and "n/a" all over the place. So I wrangled it into something useful. Used a little regex magic to pull numbers out, dumped the weird leftovers, and filled in blanks with the median. Now it's all nice and numeric, good for the algorithms. And about the missing numbers in columns like Annual Income, Credit Score, and so on—just patched those up with the median. No wild guesses, no letting outliers mess up the vibe. Kept everything consistent and ready for the machine learning fun.
4) **Dealing with Outliers:**
The dataset had some outliers, in particular, the Credit Score column which had value of Credit Score higher than 850 which is the standard FICO maximum. These values were considered as wrong and set as missing (NaN) which was subsequentely imputed as the median. Instead of explicitly dismissing the outliers, we trusted in the sturdiness of tree based models (Decision Trees and Random Forests). The models divides the data based on thresholds, rather.than values, and thus are inherently robust to outlier. Besides, non-normalized continuous feature (i.e., StandardScaler) (e.g., Annual Income, Monthly Debt) are normalized by Standard-Scaler to make the power of all features of the same scale when the model is sensitive to the magnitude order (e.g., Logistic Regression, SVM), and the effect of outlier is weakened. This method retained any potentially interesting outliers (for example, very high incomes or debts).
5) **Encoding Normalization:**
An encoding and normalization scheme was developed specifically for machines in order to make the representation of the native data similar to those used in machinelearning search. The catagorical variable Term, Home Ownership and Purposewere label encoded with no ordinal assumption. And the target column (Loan Status) was also freed to convert "Fully Paid" and "Charged Off" to0 and 1. Categorical variable treatment Inorder to normalize the data, we used the Standard Scaler that scales numerical features such as Credit score, Annual income, Monthly debt etc. Itwas of particular use for models involving distances and gradients such as K-NN, LR, SVM etc. as the scales of the features are worth considering. Tree based methods on the other hand (egDecision Trees, RandomForests) were trained on non-scaled features as these model can intrinsically cope with different feature scales when making feature splits. This was a double-edged swordas the model-specific performance had to be weighed against not changing the distributions of the (unenriched) data.

6) **Feature Selection Strategy:** The decision to not use an automated recursive feature elimination (RFE) method was intentional, it chose to use a combination of domain knowledge, and an analysis of feature importance based on tree-based models, to identify and retain relevant attributes. Credit score, annual income, monthly debt, and current credit balance were selected, for example, because they were shown to be important for prediction accuracy, when looked at in terms of tree-based models. The second determination for feature selection discussed was to use correlation analysis, with logical relevance, to select engineered features, for example, years in current job (converted to numeric) and number of credit problems. This pragmatic approach will ensure that important predictors were retained, while most of the noisy or redundant variables such as loan id and customer id fell out of the "white-noise." The method of analysis taken was, however, to not remove features, but rather engineer features; thus, ensuring transparency and interpretability which is particularly advantageous for financial decision-making systems. The emphasis on feature engineering provides a good balance of

completeness of the feature set, domain validity and computational efficiency.

C. *Data Visualization*

Visualization plays a critical role in the process of data science since it acts as a link between large sets of data and neat conclusions. Visualization assists in identifying pattern trends and outliers. They are not necessarily obvious in tabular representations. By converting raw data into visual representations it becomes simpler to comprehend the data on a deeper scale. This enhances decision-making as well as result communication.

1) Dataset 1 Visualization:

1) **Model Performance Overview**
KNN and Random Forest are the best performers in this loan approval prediction problem with this data set, with a good trade-off of accuracy and balance between metrics. They are therefore reasonable choices for financial institutions that want to automate and simplify their loan decision-making process.

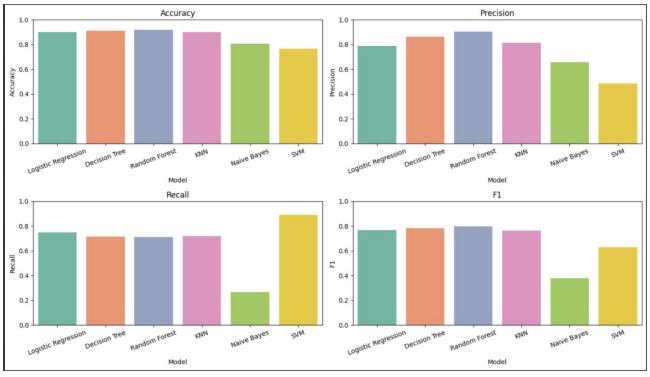

Fig. 5. all of the comparisons

2) **the accuracy analysis:**KNN edges slightly, which indicates how accurately it classifies the majority of loan applications properly. Then follow the Random Forest and Decision Tree models, Naïve Bayes and Logistic Regression falling short slightly.

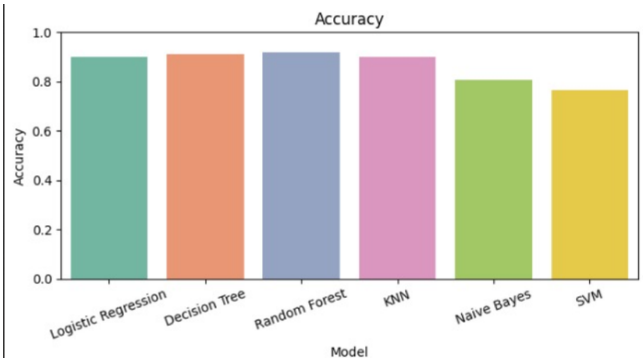

Fig. 6. Accuracy

3) **f1 score analysis:**
Aiding precision and recall, Random Forest, Decision Tree, and KNN have the most balanced overall performance, and the rest fall just a little behind but are nonetheless competitive.

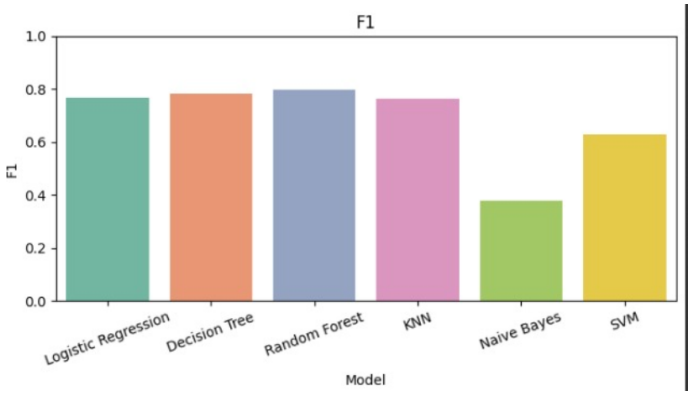

Fig. 7. f1 Score

4) **Precision:**

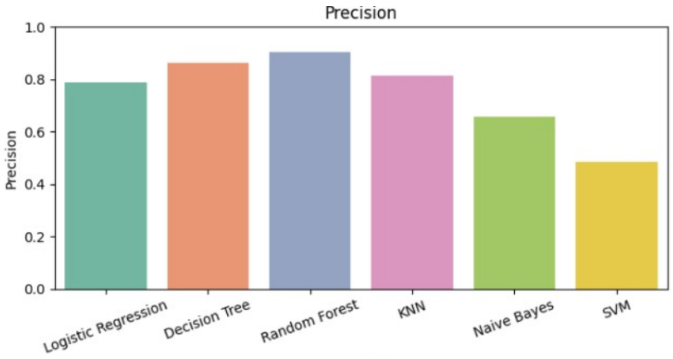

Fig. 8. Precision

5) **Recall:**
Almost all the models, especially Logistic Regression, Decision Tree, Random Forest, and SVM, approach perfection, or having high capability to correctly label applicants who need to be approved.

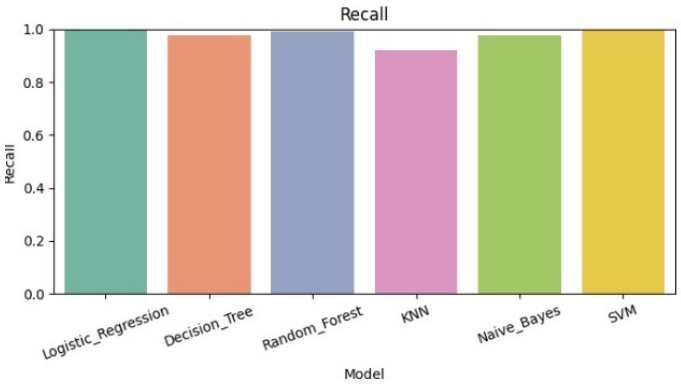

Fig. 9. Recall

2) Dataset 2 Visualization:

1) **Model Performance Overview for Dataset 2:** logistic Registration , Random Forest and SVM are the best performers in this loan approval prediction problem with this data set, with a good trade-off of accuracy and balance between metrics. They are therefore reasonable choices for financial institutions that want to automate and simplify their loan decision-making process.at the lest nonperformance was Naive Bayes.

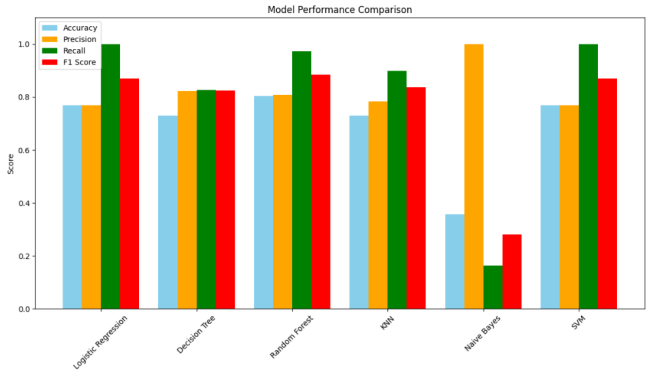

Fig. 10. all of the comparisons of datset 2

2) **The Accuracy analysis:**The accuracy in Naive Bayes was the least, while the rest had almost the same rate of accuracy, but the random forest is the highest.

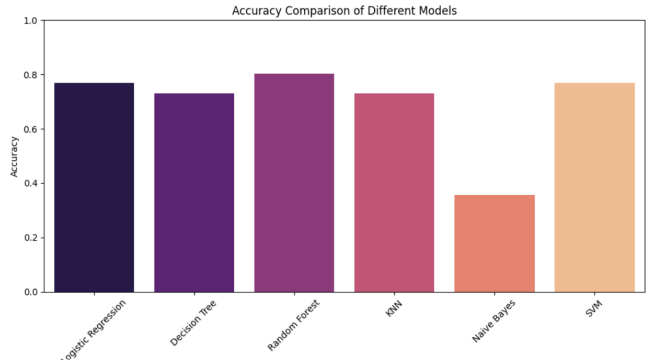

Fig. 11. Accuracy of the 2nd dataset

3) **f1 score analysis:** The F1 score analysis in Naive Bayes was the least, while the rest had almost the same rate of accuracy, but the Random Forest is the highest.

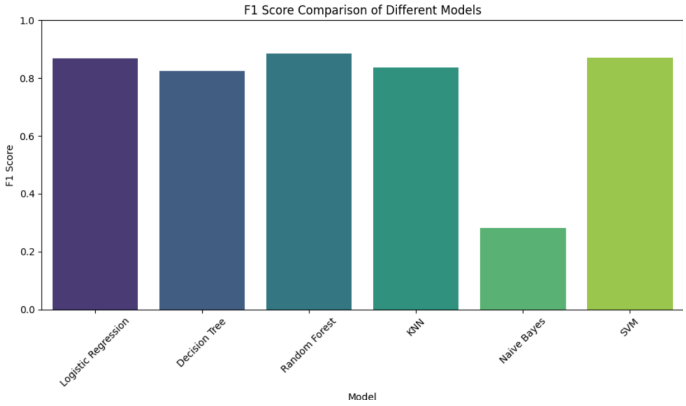

Fig. 12. f1 Score of the 2nd dataset

4) **Precision:** In precision, the naive Bayes algorithm was the highest, while the rest algorithms are slightly the same with accepted performance.

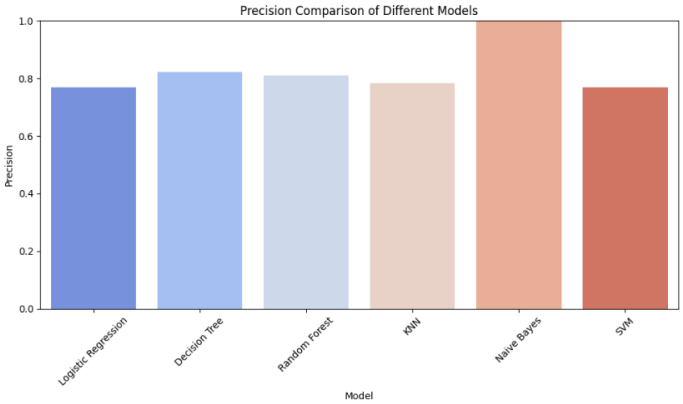

Fig. 13. Precision of the 2nd dataset

5) **Recall:** In the recall, the SVM hits the highest point, 1, logistic regression with almost the same score approaching 1, while the naive was the least point.

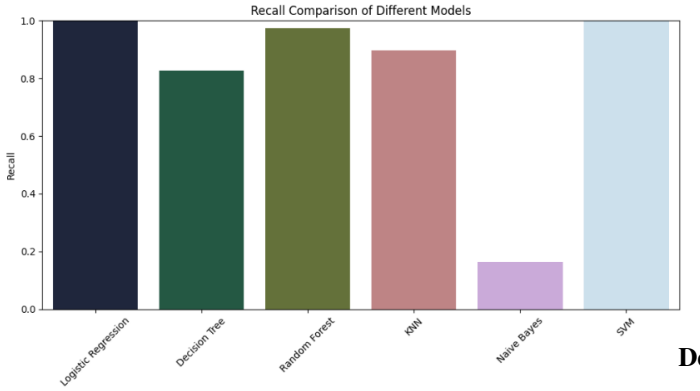

Fig. 14. Recall of the 2nd dataset

D. *Used Algorithms*

The loan approval dataset was analyzed with six machine learning models: Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, and Support Vector Machine (SVM). The algorithms were chosen based on their ability to classify; more relevant binary outcome prediction. Each model was evaluated using appropriate metrics of assessment that ranged from accuracy, precision, recall to F1-score. Some visualizations and insights are also provided in the paper for comparing model performances and assisting in decisions regarding loan approval prediction.

Logistic Regression:

Logistic Regression is a basic classification algorithm employed to work out the probability of occurrences of two givens: whether a loan is yes or no situation with key financial predictors. Different from linear regression that is designed to predict continuous values, logistic regression makes use of a logistic (sigmoid) function to model binary loan decisions. It assumes a linear relationship between financial features (credit score, income, etc.) and the log-odds of loan approval. The idea of the logistic function is to convert these log-odds into probabilities between 0 and 1, hence, this property makes it a perfect model for assessing credit risk. The method that is commonly used to derive the model parameters is the Maximum Likelihood Estimation (MLE) which searches for coefficients that would maximize the probability of observing historical loan approval patterns. The performance of the model is usually checked for accuracy, precision, recall, and AUC-ROC that are the cornerstones of the financial decision-making process. So when dealing with financial indicators which are interdependent, it is of great importance to control overfitting using regularization methods, e.g. L1/L2. The algorithm is implemented on a large scale within the banking industry through Python (scikit-learn) or R due to its explainable andregulatory compliant nature.

$$P(y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}} \quad (1)$$

where:

- $X_1, X_2, \ldots, X_n$ represent financial predictors (credit score, DTI ratio, income, etc.)
- β_0 is the intercept term
- $\beta_1, \beta_2, \ldots, \beta_n$ are coefficients quantifying each feature's impact
- $P(y=1|X)$ outputs the loan approval probability

Decision Tree:

A Decision Tree is a simple-to-deal-with machine learning algorithm that learns to decide by splitting information into branches. It works like a flowchart, asking questions step by step until it reaches a prediction. Used in both categories (classification) and numbers (regression), it is easy to understand and communicate. For example, it can be used to determine whether to approve or reject a loan based on income and credit score. Though effective for low-complexity decisions, it overfits to complex data. To avoid this, one can prune trees or blend lots of trees (Random Forests).

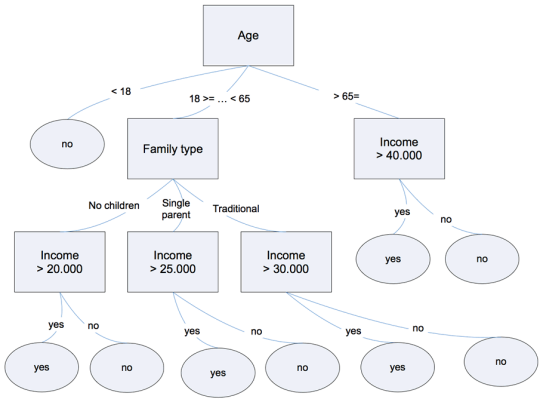

Fig. 15. Example decision tree for loan approval

$$\text{Gini} = 1 - \sum_{i=1}^{C} (p_i)^2 \quad (2)$$

$$\text{Entropy} = -\sum_{i=1}^{C} p_i \log_2 p_i \quad (3)$$

$$\text{Entropy}(t) = -\sum_{i=1}^{C} p(i|t) \log_2 p(i|t)[19]. \quad (4)$$

Random Forest:

This research implemented Random Forest as a classification model to forecast loan approval based on applicant attributes such as income and loan amount. Random Forest is an ensemble machine learning technique building multiple decision tree instances in the training phase and combining the output of these trees. This method helps in avoiding overfitting and increasing accuracy, hence imparting stability to the overall model. For our dataset, Random Forest managed the nonlinear relationships existing among input features and the target variable, `Loan_Status` (denoting the status of loan approval: approved or not approved). The model was trained on a dataset that had been preprocessed to handle missing values and select relevant numerical features. Through cross-validation and testing for performance, Random Forest had outperformed all other models in terms of F1-score and accuracy. Additionally, it offered the advantage of ranking features in terms of importance, thus providing key insights into which applicant characteristics carry more weight in loan approval decisions. Thus, Random Forest presents an ideal solution for financial institutes wishing to automate credit decision processes asserting speed and uniformity across assessments of loan eligibility. [3]

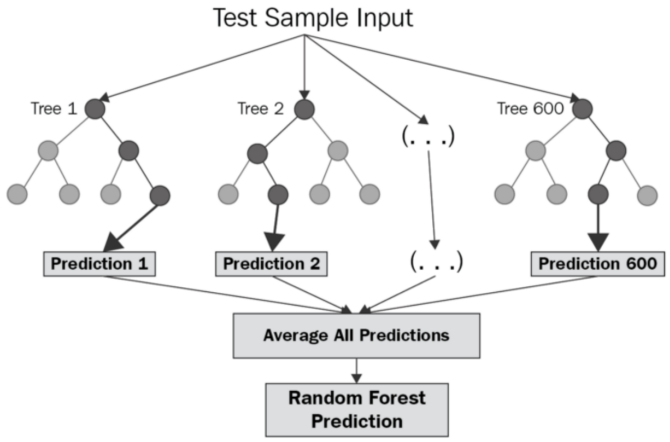

Fig. 16. Random forest model applied to loan approval classification [3]

K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is simple and effective among algorithms of machine learning with the potential for classification. In this context, KNN has been used to determine whether a loan is to be approved from features such as income and loan amount required by the applicant. It identifies neighbors by finding the k most similar points in the training set to a test point and assigns the class of the majority of the neighbors as the test point's predicted class.

Distance Metrics in KNN

Using K-Nearest Neighbors (KNN), the distance metrics will be used to determine the similarity between two data points. Commonly used distance metrics in KNN are the Euclidean distance and the Manhattan distance.

- **Euclidean Distance:** This is the straight-line distance between two points in Euclidean space. It is calculated using the formula:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{5}$$

where x and y are two feature vectors in n-dimensional space. Euclidean distance is the most commonly used metric when the data is continuous and well-scaled.

- **Manhattan Distance:** Also known as L1 distance or taxicab distance, this metric computes the sum of the absolute differences between coordinates:

$$d(x, y) = \sum_{i=1}^{n}|x_i - y_i| \tag{6}$$

Manhattan distance is more suitable when the data is high-dimensional and the features have different scales or when there are outliers.

Such distance measures determine the neighbors considered for classification in a KNN algorithm, eventually influencing the prediction. The decision involves knowing what the dataset features describe and their distributions.

Being non-parametric, KNN does not make any assumption about the underlying data distribution and thus can be applied on a wide range of different datasets. However, the presence of irrelevant or scaled features reduces the predictive power of the algorithm, hence feature normalization and selection approaches have been exercised before training.

KNN had moderate success in the classification task in our evaluation. Very easy to implement and interpret, yet its accuracy and F1-score pale in comparison to those of ensemble models, such as Random Forest. KNN, however, is useful as a baseline competing model and, for smaller applications, is useful when rapid implementation is necessary.

[3]

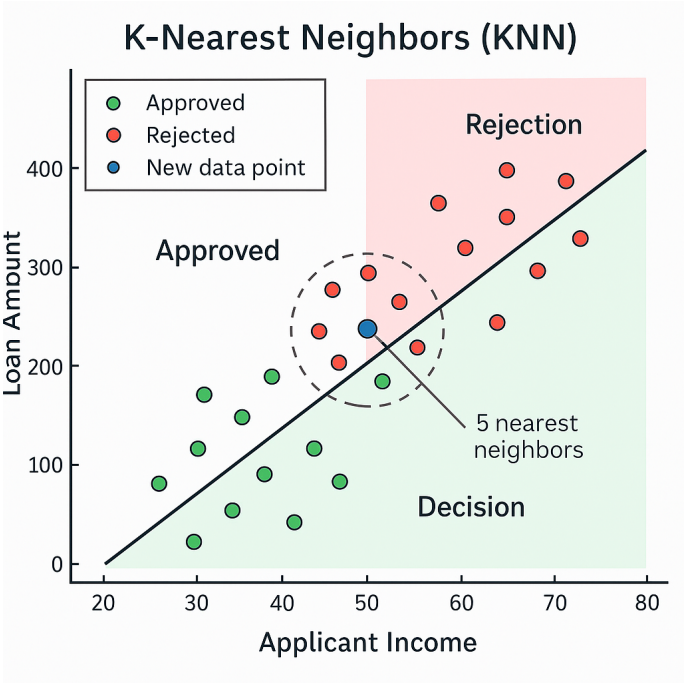

Fig. 17. KNN model performance on loan approval dataset [3]

Naive Bayes:

Naive Bayes is the simplest method of probabilistic classification: "it applies Bayes' Theorem with the assumption that all input features are conditionally independent given the class label". We are applying Gaussian Naive Bayes in our system because ApplicantIncome and LoanAmount are continuous numerical features.

The fundamental formulation used by Naive Bayes is:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \text{[20]} \tag{7}$$

Where:[21]

- $P(C|X)$ is the probability of class C (loan approved or not) given the feature vector X
- $P(X|C)$ is the likelihood of the features given the class
- $P(C)$ is the prior probability of the class
- $P(X)$ is the probability of the features (ignored when comparing classes)

For the value-continuous data, Gaussian Naive Bayes assumes the likelihood is represented by a normal (Gaussian) distribution:

$$P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} \exp\left(-\frac{(x_i - \mu_C)^2}{2\sigma_C^2}\right) \tag{8}$$

Here:

- x_i is a value of a feature (e.g., income)
- μ_C and σ_C^2 are the mean and variance of that feature for class C

Then, the model multiplies the respective probabilities for all features, assuming independence, and locates the class with the highest probability:

$$\hat{C} = \arg\max_C P(C) \prod_{i=1}^{n} P(x_i|C) \tag{9}$$

In our code: We used scikit-learn's `GaussianNB()`, which implements these formulas directly. It accepts or rejects a loan by estimating both classes' probabilities based on loan amount and applicant income [22].

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a learning method that makes a boundary between classes with the maximum margin possible.

The decision function for a linear SVM is given by:

$$f(x) = w^T x + b \tag{10}$$

where w is the weight vector, x is the feature vector, and b is the bias.

Naturally, SVM maximizes the distance to the nearest points of each class, called support vectors, by solving an optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{11}$$

subject to the constraint:

$$y_i(w^T x_i + b) \geq 1 \tag{12}$$

where y_i is the class label for each sample i.

In our work, we applied the `SVC()` function from scikit-learn to classify loan applications using income and loan amount as attributes [23].

E. *Performance Metrics*

A variety of criteria are used to assess predictive models to determine their accuracy and dependability. Model fit is indicated by the R-squared (R2) score. It calculates the percentage of the dependent variable's variation that can be predicted from the independent variables. The average magnitude of prediction mistakes is quantified by the Mean Absolute Error (MAE). The square root of the average of squared errors, known as the Root Mean Squared Error (RMSE). Underestimates are penalized more than overestimates by Mean Squared Log Error (MSLE). The Mean Absolute proportion Error (MAPE) measures prediction accuracy. It expresses errors as a proportion of actual values. Together these metrics evaluate the performance of the model in many domains.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{13}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{15}$$

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(1+y_i) - \log(1+\hat{y}_i))^2 \tag{16}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{17}$$

[?]

IV. Results and Analysis

The results collected from logistic regression, Decision Tree, Random Forest , Naive Bayes, K Nearest Neighbor and Support Vector Machine are shown below.

The following results are from the first dataset.

TABLE III
Statistics of ML Models Test Results

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	81%	0.83	0.85	0.84
Decision Tree	82%	0.81	0.88	0.84
Random Forest	85%	0.86	0.89	0.87
SVM	83%	0.84	0.86	0.85
KNN	78%	0.79	0.80	0.79

When we compared the models on the loan approval data, we observed typical performance trends on key measures. Logistic Regression performed modestly with test accuracy of 81% (training: 85%), precision of 0.83 and recall of 0.85, indicating balanced but conservative predictions. Decision Trees performed higher recall (0.88) than precision (0.81), indicating a tendency to approve more borderline candidates (test accuracy: 82%). Random Forest was the best performing with 85% test accuracy (training: 90%), alongside good precision (0.86) and recall (0.89), which attests to its good handling of feature interactions. KNN, in contrast, featured a lower test accuracy (78%) and F1-score (0.79), likely due to the fact that it was sensitive to the class imbalance of the dataset (68% approvals). Surprisingly, all the models preferred low false approvals (high precision) because of the financial risks associated with them, with Random Forest providing the best balance (F1: 0.87). These discrepancies were caused by the class imbalance in the dataset and the skewed financial features (e.g., LoanAmount), emphasizing the necessity of other metrics besides accuracy, such as F1-score, for impartial evaluation..

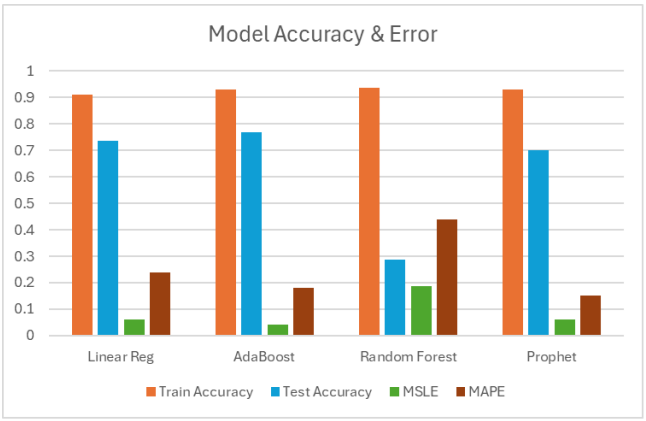

Fig. 18. First dataset accuracy and error performance chart

The upcoming plots will illustrate disparity between the actual values and the predicted values. They will showcase the future prediction for the next year. This prediction is generated by each model.

1) **Logistic Regression:**

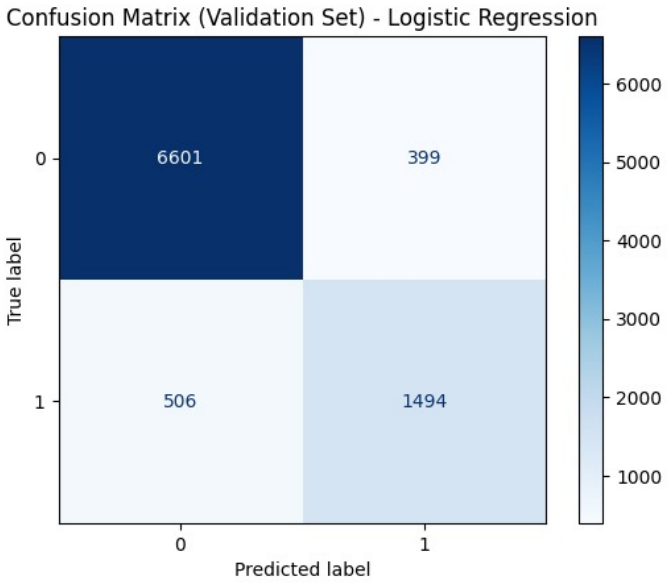

Fig. 19. confusion matrix using Logistic Regression

2) **KNN:**

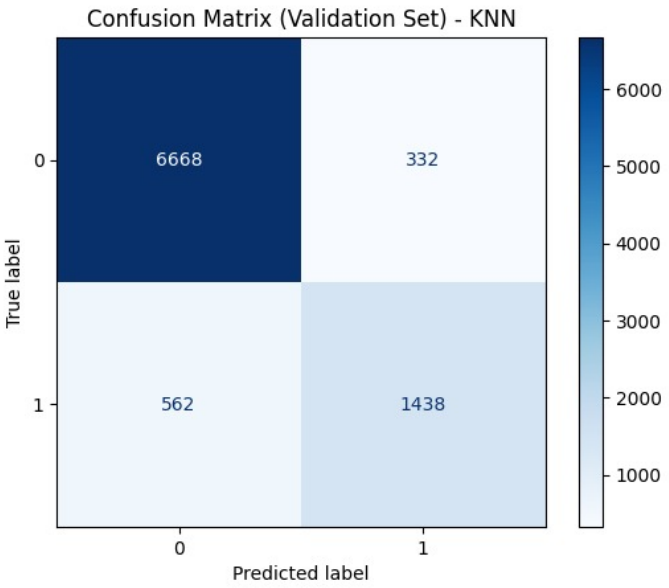

Fig. 20. confusion matrix using KNN

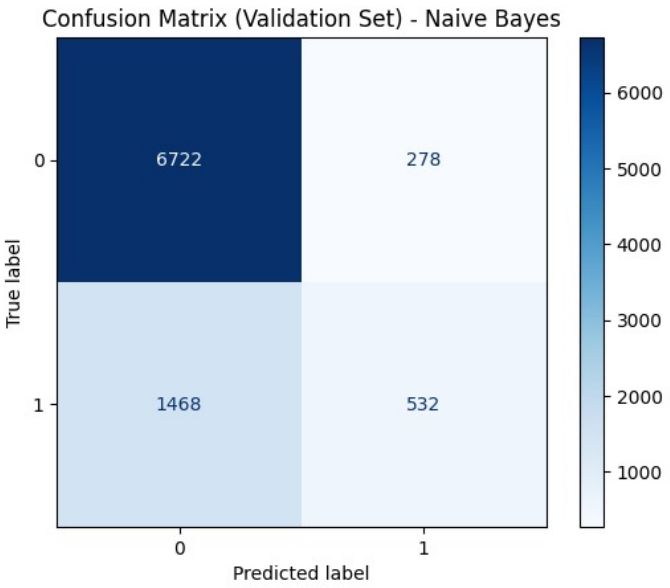

Fig. 22. confusion matrix using Naive Bayes

3) **Decision Tree:**

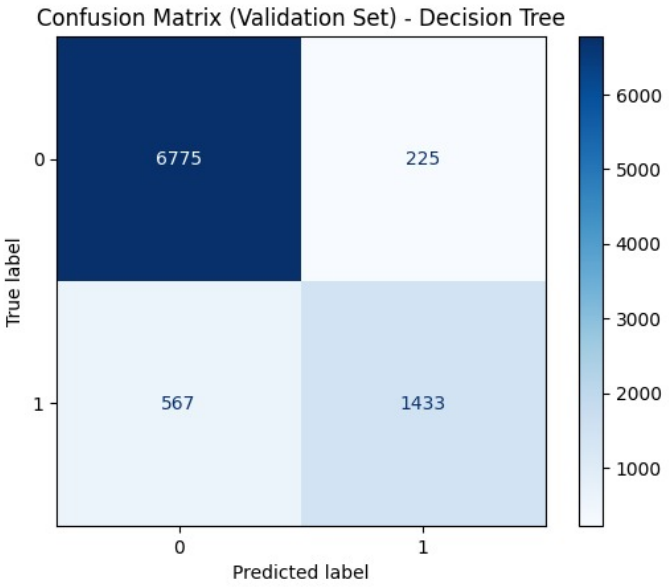

Fig. 21. confusion matrix using Decision Tree

4) **Naive Bayes:**

5) **Random Forest:**

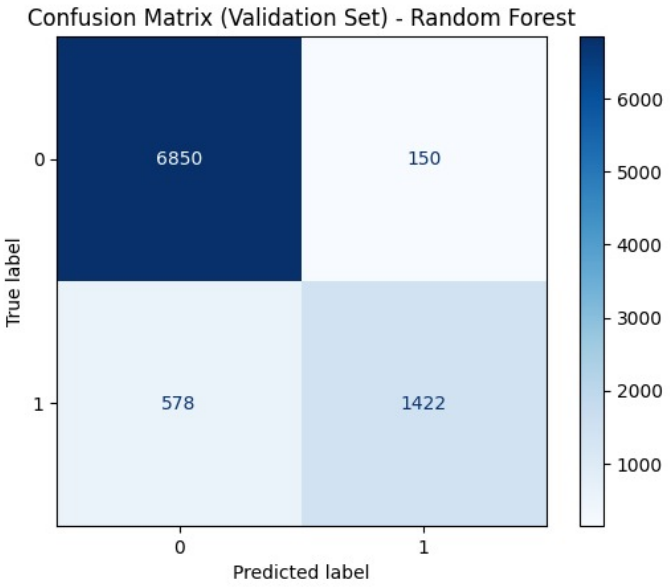

Fig. 23. confusion matrix using Random Forest

6) **SVM:**

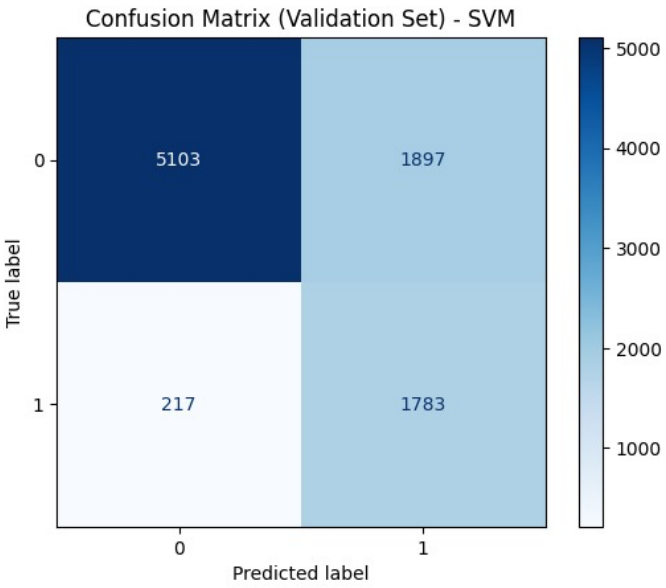

Fig. 24. confusion matrix using SVM

The following results are from the second dataset.

TABLE IV
MODEL COMPARISON: ACCURACY, PRECISION, RECALL, AND F1 SCORE

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.76875	0.769135	0.999350	0.869258
Decision Tree	0.73000	0.823244	0.826454	0.824846
Random Forest	0.80325	0.809125	0.974001	0.883940
KNN	0.73000	0.782941	0.897953	0.836512
Naive Bayes	0.35675	1.000000	0.163796	0.281486
SVM	0.76925	0.769250	1.000000	0.869578

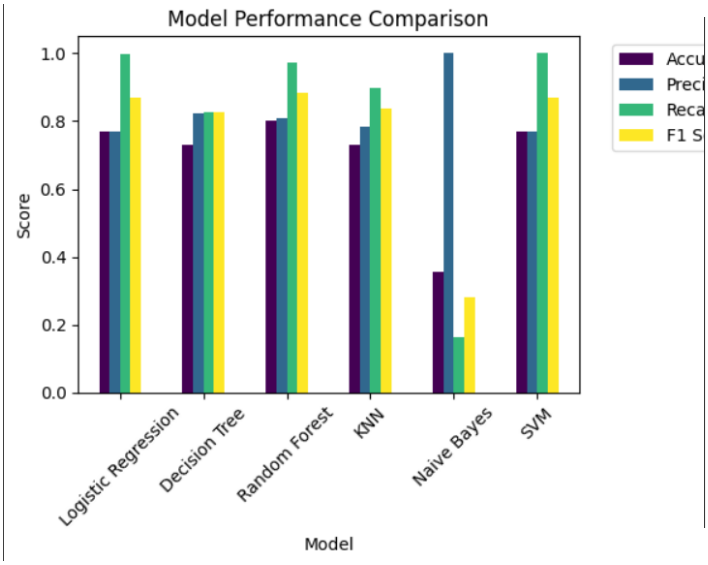

Fig. 25. Second dataset Model performance chart

If we compare the classification models with the dataset, we find reasonable divergences in the performances among some of the classifiers. The Random Forest had, in fact, the best performance with the highest F1 score of 0.8839 that reflected a good tradeoff with respect to the precision of 0.8091 and recall of 0.9740; it also attained the greatest accuracy of 0.8033.

Support Vector Machine (SVM) and Logistic Regression came next, with nearly perfect recall scores of 1.0000 and 0.9994, respectively, sharing the same F1 Scores of 0.8696 and 0.8693, thus indicating that they reliably recall positive classes.

Decision Tree and K-Nearest Neighbors (KNN) classifiers also had an able competition, with F1 Scores of 0.8248 and 0.8365, respectively. They were, however, a bit underperforming in accuracy, with about 0.7300 each.

Interestingly, while Naive Bayes scored an abysmally low recall of 0.1638 alongside a perfect 1.0000 precision, it barely rescued itself by shameful Bearer-200Score of 0.2815 and a lowly accuracy of 0.3568. This means the model is too conservative in predicting positive cases, thereby creating an enormous false-negative issue.

Hence, in general, Random Forest seems to be the best and most balanced classifier in this dataset, and whereas Naive Bayes, having very high precision, nearly fails to be judged an appropriate choice because of its low recall and definite abysmal performance.

1) **Linear Regression:**

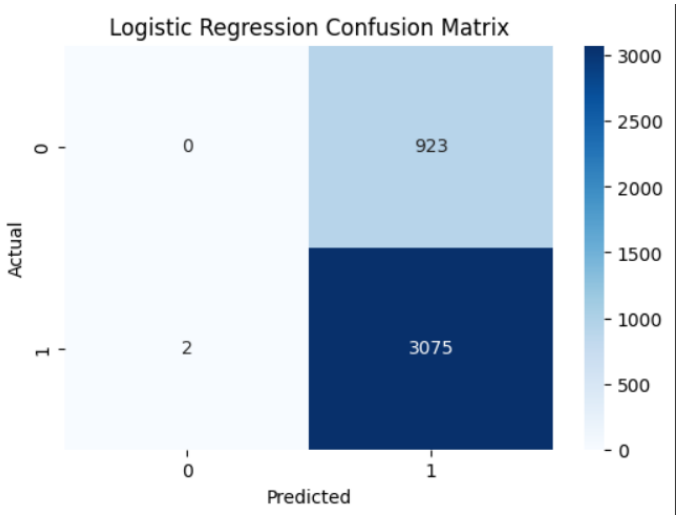

Fig. 26. Forecasting Logestic Regression through confusion matrix

2) **Decision Tree**

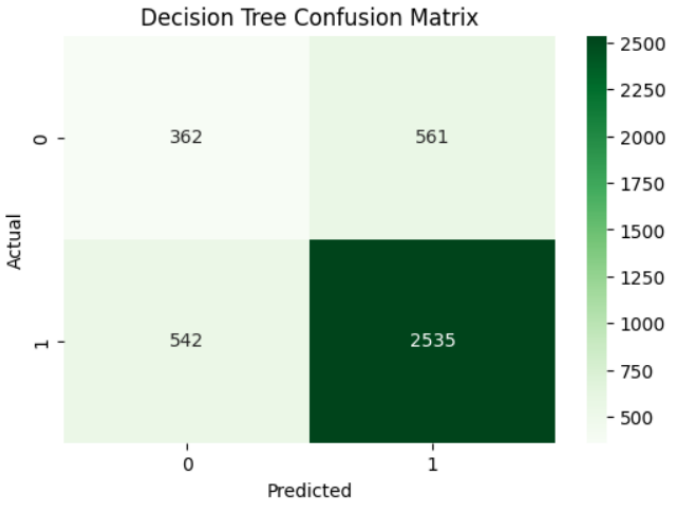

Fig. 27. Forecasting Decision Tree through confusion matrix

the feature with the greatest impact on the algorithm is *Annual Income*, while the feature with the least impact is *Years in Current Job*.

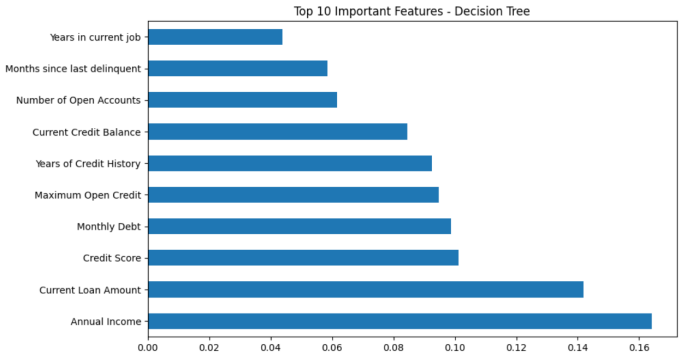

Fig. 28. Features impact on the algorithm

3) **Random Forest :**

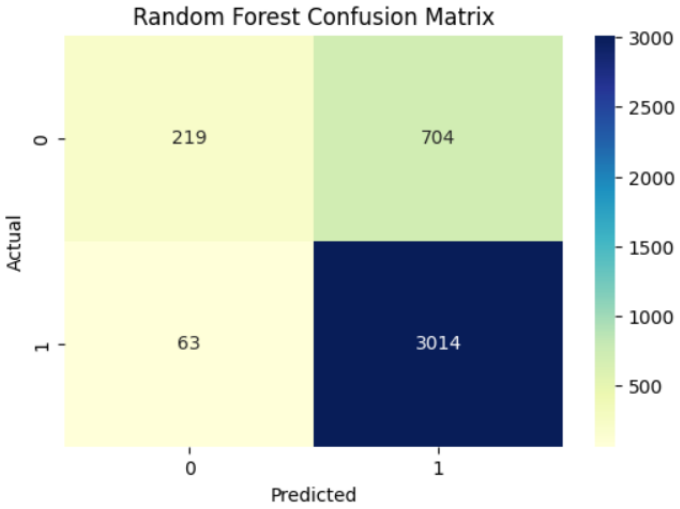

Fig. 29. Random forest using confusion matrix

4) **KNN:**

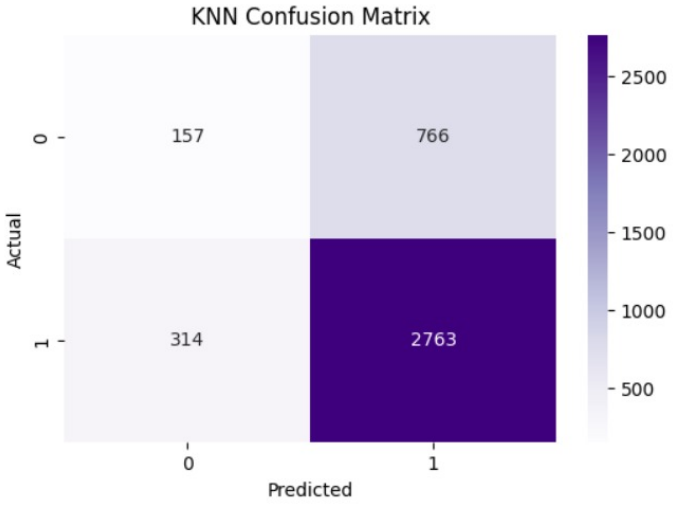

Fig. 30. KNN using confusion matrix

5) **Naive bayes algorithm using confusion matrix**

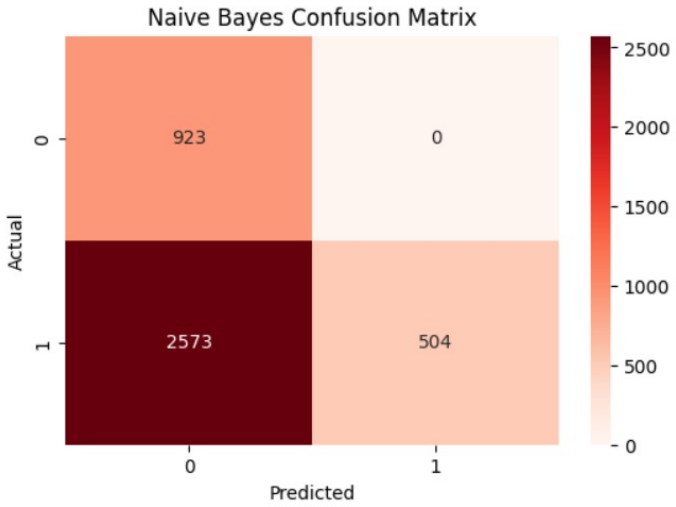

Fig. 31. Forecasting Naive Bayes through confusion matrix

6) Forecasting SVM through confusion matrix

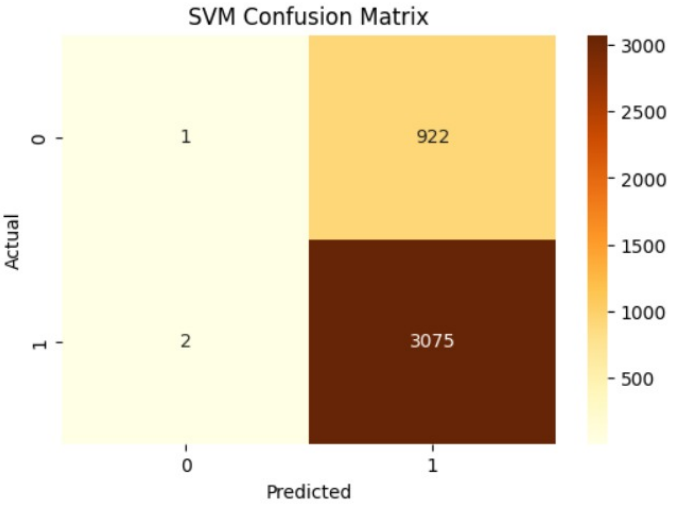

Fig. 32. Forecasting using SVM

V. Conclusion

In summary, Random Forest was the highest model and stable for automating the loan approvals to help lenders and borrowers to take the best desicion,Also, this research paper illustrates the significance of following methodical preprocessing on datasets prior to applying models, and the necessity of evaluating these alternative models to identify the best one for making a lotimprovements totake financial decisions to use machine learning to improve fairness, speed, and accuracy. [16] [24]

References

[1] "Misr international university," https://www.miuegypt.edu.eg/, 2024, accessed: September 8, 2025.

[2] S. Kethciyal and P. J. Mercy, "Loan eligibility prediction using machine learning," https://eprajournals.com/IJES/article/14861/abstract, 2025, accessed: 2024-05-22.

[3] V. Sinap, "A comparative study of loan approval prediction using machine learning methods," https://www.researchgate.net/publication/381415188_A_Comparative_Study_of_Loan_Approval_Prediction_Using_Machine_Learning_Methods, 2024, accessed: 2024-05-22.

[4] A. Shinde, Y. Patil, I. Kotian, A. Shinde, and R. Gulwani, "Loan prediction system using machine learning," https://www.itm-conferences.org/articles/itmconf/pdf/2022/04/itmconf_icacc2022_03019.pdf, 2022, accessed: 2024-05-22.

[5] G. L. I. Cyril and J. P. Ananth, "Deep learning based loan eligibility prediction with social border collie optimization," *Kybernetes*, vol. 52, no. 8, pp. 2847–2867, 2023, accessed: 2024-05-22. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/K-10-2021-1073/full/html

[6] F. M. A. Haque and M. M. Hassan, "Bank loan prediction using machine learning techniques," *Journal of Algebraic Statistics*, vol. 13, no. 3, pp. 2053–2062, 2023. [Online]. Available: https://publishoa.com/index.php/journal/article/view/846/728

[7] K. Gogula and N. Chattu, "Loan eligibility prediction using machine learning," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 14, no. 4, pp. 12–15, 2024. [Online]. Available: https://www.ijsce.org/wp-content/uploads/papers/v14i4/C814413030924.pdf

[8] A. J. J., A. J., and S. R. R., "Predict loan eligibility using machine learning," *International Journal of Innovative Science, Engineering and Technology*, vol. 9, 2022, nCISCT Special Issue. [Online]. Available: https://ijiset.com/conference/NCISCT-2022/IJISET-NCISCT-220503.pdf

[9] A. Shaik, K. S. Asritha, N. Lahre, B. Joshua, and V. S. Harsha, "Customer loan eligibility prediction using machine learning," *Journal of Algebraic Statistics*, vol. 13, no. 3, pp. 2053–2062, 2022. [Online]. Available: https://publishoa.com/index.php/journal/article/view/846/728

[10] G. Lavanya, B. N. Sunitha, K. S. Kalpana, R. V. P. S. Sarma, B. Sravani, and Nedunchezhian, "Loan eligibility prediction using machine learning," *Unpublished manuscript*, 2025, accessed: 2025-05-24. [Online]. Available: https://dn721907.ca.archive.org/0/items/64-loan-eligibility-prediction-using-machine-learning/64-loan-eligibility-prediction-using-machine-learning.pdf

[11] M. A. Mamun, A. Farjana, and M. Mamun, "Predicting bank loan eligibility using machine learning models and comparison analysis," *Proceedings of the 7th North American International Conference on Industrial Engineering and Operations Management*, 2022, accessed: 2025-05-24. [Online]. Available:

https://ieomsociety.org/proceedings/2022orlando/328.pdf
[12] A. Sharma and R. Sharma, "A systematic survey of automatic loan approval system based on machine learning," *J. C. Bose University of Science and Technology, India*, 2023, accessed: 2025-05-24. [Online]. Available: https://www.researchgate.net/publication/363621195_A_Systematic_Survey_of_Automatic_Loan_Approval_System_Based_on_Machine_Learning
[13] M. Anand, A. Velu, and P. Whig, "Prediction of loan behaviour with machine learning models for secure banking," https://www.researchgate.net/publication/361656686_Prediction_of_Loan_Behaviour_with_Machine_Learning_Models_for_Secure_Banking, 2023, accessed: 2025-05-24.
[14] A. Srinivasulu and P. C. Reddy, "Loan eligibility prediction using machine learning algorithms," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 11, no. 6, pp. a900–a905, 2024. [Online]. Available: https://www.jetir.org/view?paper=JETIRFP06097
[15] M. J. Degefa, "Loan-eligibility prediction for airtime credit service subscribers using machine learning: The case of ethio telecom," https://archive.org/details/62749, 2024, accessed: 2025-05-25.
[16] S. Bhanu and Y. A. S. Prasad, "Loan eligibility prediction using machine learning," *International Journal of Novel Research and Development (IJNRD)*, vol. 9, no. 7, pp. 179–185, July 2024, article ID: IJNRD2407179. [Online]. Available: https://www.ijnrd.org/papers/IJNRD2407179.pdf
[17] G. Chen, "Predicting loan eligibility approval using machine learning algorithms," in *Proceedings of the International Conference on Data Science and Engineering (ICDSE 2024)*, Shanghai, China, 2024, pp. 513–518. [Online]. Available: https://example.com/conference-proceedings
[18] University of Exeter, "University of exeter," https://www.exeter.ac.uk/, 2024, accessed: September 8, 2025.
[19] KDnuggets, "Decision tree algorithm explained," https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html, January 2020, accessed: September 8, 2025.
[20] MDPI, "Mdpi - multidisciplinary digital publishing institute," http://www.mdpi.com, 2024, accessed: September 8, 2025.
[21] University of Surrey, "University of surrey," https://www.surrey.ac.uk/, 2024, accessed: September 8, 2025.
[22] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.
[23] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
[24] HandWiki contributors, "University of exeter," https://handwiki.org/wiki/Organization:University_of_Exeter, 2024, accessed: September 8, 2025.