Name – Ahaan Tagare **Student ID – 33865799**

Assignment -2 **Subject - Statistics and Statistical Data Mining**

Predictive Modelling for Loan Approval: A Comparative Analysis of Machine Learning Techniques

Abstract

With an emphasis on credit risk analysis, this project combines sophisticated statistical and machine learning methods to forecast and assess creditworthiness by utilizing an extensive dataset of loan data. A detailed investigation of the elements affecting credit risk is made possible by the dataset's inclusion of important variables such credit score, yearly income, loan amount, and payment history. The analysis predicts whether a borrower is likely to default on a loan (high-risk) or successfully repay it (low risk) in a classification task.

The analysis begins with data preprocessing and exploratory data analysis to identify patterns and address missing values. Principal Component Analysis (PCA) is applied to reduce dimensionality and capture the most significant variance in the data. To evaluate their predictive accuracy, a variety of supervised learning models are used, such as decision trees, random forests, gradient boosting machines (GBM), and neural networks. These models are evaluated and compared using performance criteria like accuracy, precision, recall, and ROC-AUC scores. Furthermore, feature importance studies shed light on the factors that are most important in forecasting credit risk.

Through thoroughly described R code and visualization tools, the project places a strong emphasis on reproducibility and clarity, guaranteeing that the results are both interpretable and useful. The findings of this study can help financial institutions make better lending decisions and enhance their risk assessment procedures, which would ultimately lower default rates and increase financial stability. This research shows how statistical learning techniques can be used practically to solve real-world credit risk management problems.

Introduction

In the financial industry, credit risk analysis is essential because it helps organizations determine the probability that borrowers may be unable to not make payments on their loans. Utilizing large datasets to assess and forecast credit risk has become crucial due to the growing need for data-driven decision-making. Using a dataset of 100,514 loan records with 19 variables that capture different facets of borrower profiles and financial histories, the goal of this study is to develop a strong analytical framework. There are two datasets on Kaggle website the other has only 10,000 records, in this project will be working with the credit train dataset as credit data since it has more records and credit test data is missing a variable to make the datasets compatible.

The goal of the analysis is to find trends and risk factors that have a big impact on loan performance, like income levels, credit score, and previous credit behaviour. The project aims to categorize loan applications according to risk levels (low and high) using statistical and machine learning approaches, which would help lenders make better decisions. The knowledge gained will balance lending institutions profitability with financial inclusion by reducing defaults and facilitating equitable access to credit.

Background work

A few existing studies have been conducted on this dataset and similar datasets, as outlined below, to compare results and evaluate the performance of different models

1- The study, Loan Status Classification Using KNN, SVM, Logistic Regression, and Naïve Bayes by Mohamed Anwar, evaluates KNN, SVM, and Naïve Bayes classifiers on a scaled dataset. With 13 neighbours, KNN achieved the highest accuracy of 80.51% during training and 78.82% during testing. With training accuracy of 79.78% and testing accuracy of 79.60%, Support Vector Machine with C=1 demonstrated consistent performance. Naïve Bayes produced noticeably lower training and testing accuracies of 39% and 38%, respectively, due to its lack of suitability for the dataset [1].

2 - Ileberi Emmanuel, Yanxia Sun, and Zenghui Wang, in their study titled "A Machine Learning-Based Credit Risk Prediction Engine System Using a Stacked Classifier and a Filter-Based Feature Selection Method" (Journal of Big Data), demonstrated the effectiveness of their stacked classifier. It achieved the best performance on the Australian dataset with an accuracy of 86.23%, F1-score of 84.58%, AUC of 0.934, and on the German dataset with an accuracy of 82.80%, F1-score of 86.35%, and AUC of 0.944, outperforming other methods like RF, GB, XGB, and ANN [2].

3 - In the paper "Bank_Loan_Status_Classification" by Yassmen Youssef on Kaggle, The investigation shows how different categorization algorithms, including Random Forest, Decision Tree, and K-Nearest Neighbors (KNN), are applied to forecast loan situations. With testing accuracy falling below 65% and training accuracy above 85%, KNN demonstrated overfitting. With an F1-score of 0.70 and a balanced accuracy of 71%, Random Forest did better than the Decision Tree, which had an average test accuracy of 69%. Minority class representation increased by 40% because of the use of strategies such random over-sampling, under-sampling, and SMOTE to solve the class imbalance. While preserving the overall robustness of the model, these techniques greatly increased the precision and recall metrics, improving prediction dependability [3].

4 - The project "Bank Loan Classification" by Ahmed Ashraf Helmi evaluates various machine learning models for predicting loan approval. The Linear SVC model achieved an accuracy of 0.77, with precision and recall for class 1 being 0.77 and 1.00 respectively, resulting in an F1 score of 0.87. The Random Forest Classifier showed overfitting with a training score of 0.9999 and a testing score of 0.77. Its precision and recall for class 1 were 0.79 and 0.97, respectively, with an F1 score of 0.87. The Gradient Boosting Classifier produced a similar performance with an accuracy of 0.78, precision of 0.78, recall of 1.00, and F1 score of 0.87. Despite good overall accuracy, the models showed a notable imbalance in detecting denied loans [4].

Dataset

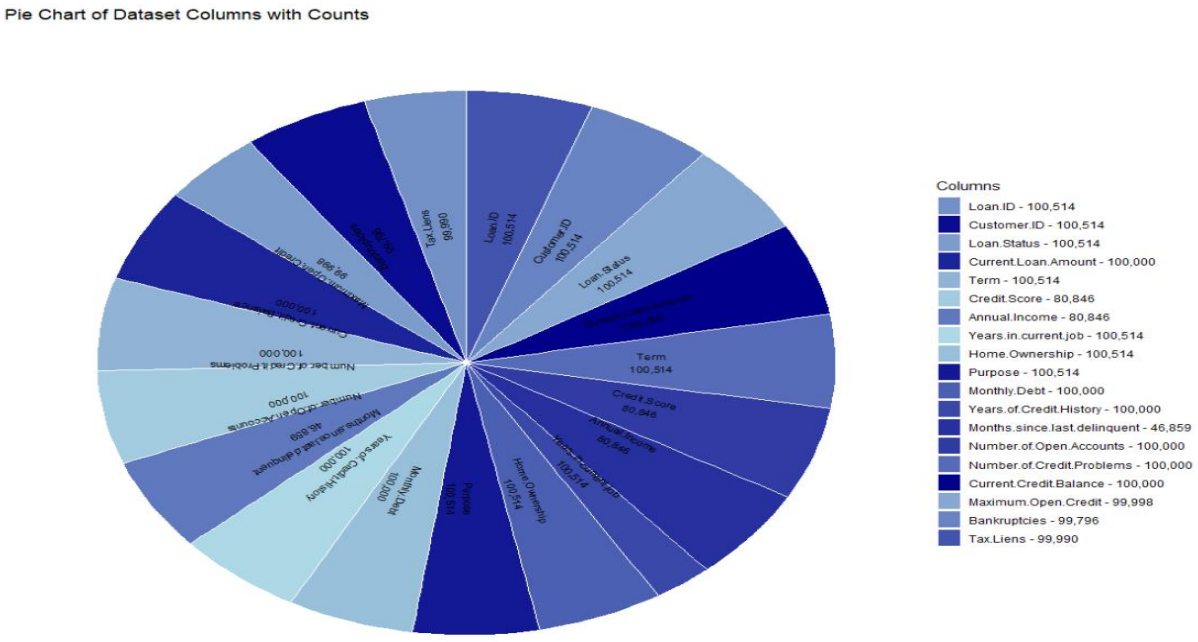
The dataset provides a thorough investigation of loan performance and consumer creditworthiness, including 100,514 observations and 19 variables. Because each data is specifically linked to a particular loan and client, analytical accuracy and traceability are guaranteed. Loan status information is recorded to show whether a loan is "Fully Paid" or in default, offering information on repayment patterns.

Customers' financial health and responsibilities are highlighted by important financial measures like credit ratings, yearly incomes, and current loan amounts. The necessity of data cleaning is highlighted by the possibility that some of these values—such as exceptionally high earnings or credit scores—are outliers. Patterns in consumers' financial obligations and credit experience are shown by other variables, such as years of credit history and monthly debt.

Demographic information, such as years of employment and housing status (such as "Home Mortgage," "Own Home," or "Rent"), provides context for living arrangements and job stability. With sections like "Debt Consolidation" and "Home Improvements," the purpose of loans is also explained in detail, providing insight into how money is spent.

Additional information about financial behaviours and possible hazards can be found by looking at credit and delinquency-related factors such the quantity of open accounts, credit issues, bankruptcies, and tax liens. To completely comprehend their impact, missing data in categories such as months since the last delinquency may need to be imputation or subjected to further analysis. All things considered, this dataset is a strong tool for analysing customer creditworthiness and loan performance.

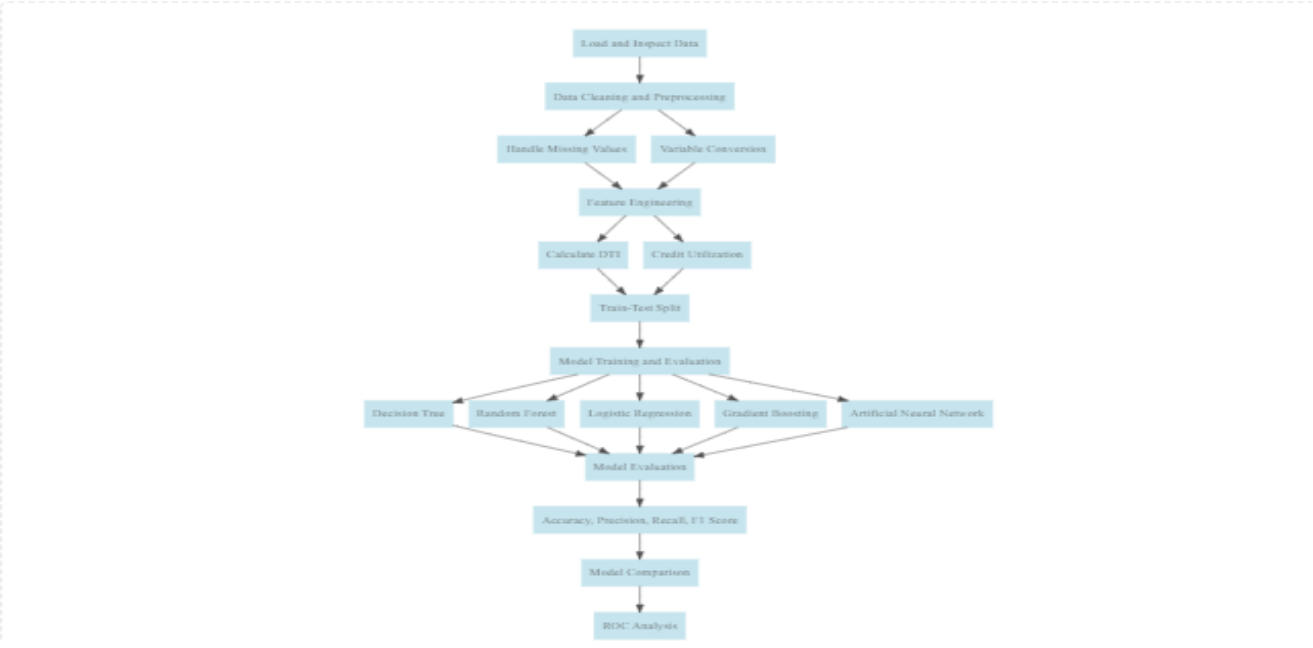
Below is a pie chart illustrating the number of non-missing values for each column in the dataset, providing an overview of data completeness and availability.



Methodology

This analysis aims to build and evaluate various machine learning models to predict loan approval based on customer credit data. The procedure starts with data preprocessing, which involves resolving missing values, converting categorical variables into factor or numeric formats, and creating new features like credit utilization and the debt-to-income (DTI) ratio. By taking these precautions, the data is will be consistent, clean, and appropriate for machine learning models. Missing numerical values are imputed using the proper techniques (e.g., mean imputation) and categorical variables such as "Home Ownership" and "Purpose" are factorized. Any missing or infinite designed values are substituted with zero to avoid mistakes.

After dividing the data into training and test sets, a variety of methods, such as Decision Tree, Random Forest, Logistic Regression, Gradient Boosting Machine (GBM), and Artificial Neural Networks (ANN), are applied as part of the modelling process. The test data is used to evaluate each model after it has been trained using the training data. For every model, performance metrics like F1 Score, Accuracy, Precision, and Recall are calculated. Classification results are evaluated using confusion matrices, and the main factors influencing loan acceptance are determined by analysing feature importance. AUC values and ROC curves also shed light on how well the models differentiate between loans that have been granted and those that have not.



Lastly, a comparison of the models reveals their advantages and disadvantages. Because of their ensemble approaches, Random Forest and Gradient Boosting frequently perform exceptionally well at catching intricate patterns, whereas Logistic Regression provides interpretability. Although they offer a versatile modelling technique because to their non-linear nature, neural networks may need more careful tweaking. A

thorough evaluation of model performance is made possible by evaluation measures and representations such as ROC curves. This multi-model technique helps choose which model is most dependable for deployment and guarantees strong forecasts.

Data Preprocessing Overview

The foundation of any successful analytical endeavour is data preprocessing, which guarantees that unprocessed data is converted into an organized and refined format for analysis. The procedures followed to deal with missing values, standardize inconsistent data, and get categorical variables ready for analytical modelling are all described in depth in this study. This procedure created a strong basis for precise exploratory and predictive research by fusing numerical insights with interpretative clarity.

i - Handling Missing Values: Ensuring Data Completeness

To maintain the dataset's usability and dependability for analysis, addressing missing values is an essential part of data preparation. If ignored, missing entries may induce bias, distort results, or cause computation errors. To address this, missing values were methodically added to the dataset to guarantee consistency and completeness.

Missing values for numerical and categorical variables were substituted with 0. This method was selected because it served as a neutral stand-in, particularly for variables like financial information where the lack of data indicated cannot be substituted by another technique since each customer values are different. By doing this, the dataset was preserved, keeping all 100,514 entries from all the variables without losing any important records.

The objective of this strategy was not only to address data gaps but also to retain the full potential of the dataset for analysis. By ensuring that every record was complete, the data was prepared for reliable and robust analysis in later stages, such as building predictive models or deriving insights.

ii - Feature Transformation: Enhancing Data Usability

To improve the usability of attributes and make them more structured and analytically useful, the dataset underwent focused transformations. To make variables more compatible with statistical models and machine learning algorithms, those containing non-numeric or categorical information were methodically converted into numeric representations. This transformation allowed the data to be more consistent and computationally efficient.

To differentiate between short-term and long-term loans, a variable that previously contained textual descriptors of loan terms was transformed into binary values. For

instance, "Short Term" was encoded as 0, while all other terms were encoded as 1. This binary encoding not only simplified the analysis but also helped in identifying patterns associated with loan durations. Another feature representing job tenure originally contained textual descriptions such as "10+ years" or "5 years." These descriptions were converted into numeric values; for example, "10+ years" was mapped to 10, while "5 years" and similar entries were stripped of text and converted into integers. For cases where job tenure was missing, a default value of 0 was assigned to ensure data consistency and avoid computational issues.

This transformation process not only preserved the semantic meaning of the data but also improved its numerical integrity. By making these characteristics more organized and standardized, the dataset became better equipped for predictive modelling, allowing algorithms to process and interpret the information more effectively. This enhancement of feature usability significantly contributed to the overall readiness of the data for advanced analytics.

iii - Categorical Variable Transformation

Certain attributes were converted into factors with matching number codes to improve the use of categorical variables. For example, factor levels like "Rent," "Own Home," and "Home Mortgage," with codes 1, 2, and 5, respectively, were created from the variable that represented homeownership. Without changing the data's original meaning, this modification made sure it was consistent and computationally efficient. With categories like "Debt Consolidation" (coded as 2), "Buy a Car" (coded as 4), and "Home Improvements" (coded as 8), the variable for lending purposes was also standardized.

To ensure that the data was prepared for algorithmic modelling, these transformations generated structured tables that mapped descriptive levels to numerical codes. This procedure optimized the dataset for predictive analysis while preserving the semantic context by enhancing the interpretability and compatibility of categorical variables.

Feature Engineering

To improve the analytical usefulness of the dataset, two new features were created. By dividing the monthly debt by the yearly income, the Debt-to-Income (DTI) ratio was computed, providing information about the financial burden in relation to wages. To maintain uniformity throughout the dataset, missing or infinite values—which can result from division by zero or from missing data—were substituted with 0. The calculated DTI values, which varied from 0.0045 to 0.0183, showed that people's debt levels varied in relation to their income.

A clear indication of how much accessible credit is being used is also provided by the Credit Utilization ratio, which was calculated by dividing the current credit balance by the maximum amount of open credit. For accuracy and completeness, 0 was also used to replace any missing or undefinable values in this feature. The calculated utilization rates, which ranged from 0.27 to 0.79, showed that the dataset's credit usage patterns were different. In addition to preserving the dataset's integrity, these designed features provide crucial predictive features for more research.

Train-Test Split: Preparing Data for Predictive Modeling

The dataset was split into two sections: one for model training and another for performance testing to create dependable and accurate predictive models. This division guarantees that the model gets tested on unseen data, mimicking real-world situations, and learns patterns from a subset of the data. 70% of the data, comprising 70,360 records, was allocated for training, while 30%, with 30,154 records, was set aside for testing. This division ensures a fair assessment of the model by maintaining the balance of loan approval outcomes in both sections, allowing the model to generalize well to new data.

Before use, the data was thoroughly checked for inconsistencies to ensure its reliability. Neutral values were substituted for any infinite values that might have arisen from issues such as division by zero during earlier computations. Additionally, the dataset was reviewed for missing or zero values in each column to identify and address any potential gaps. By resolving these issues, the data was made consistent, clean, and structured, ensuring it was prepared for efficient analysis and accurate prediction of customer risk by the models.

PCA (Principal Component Analysis)

In this process, the numeric features of the training dataset are separated, and the data is standardized to ensure that all features contribute equally to the analysis. Principal Component Analysis (PCA) is then applied to reduce the dataset’s dimensions while retaining the most significant patterns and structures in the data. The performance of PCA is assessed by looking at a summary that shows how much variance each principal component explains, and a scree plot is generated to visualize the importance of each component. Formula $Z = X' \cdot W$ where Z represents the transformed data (principal components). X' is the standardized data (mean-centered and scaled). W is the matrix of eigenvectors (principal components) [5].

The first observation has scores of -1.9115791 for PC1, -2.07093594 for PC2, 0.1947058 for PC3, 0.2516129 for PC4, and 0.9925896 for PC5. The second observation has scores of 0.8930564 for PC1, -0.03664867 for PC2, 1.1989653 for PC3, 2.4150290 for PC4, and

-2.6230365 for PC5. Each score represents the projection of the observation onto a principal component, showing how much it contributes to each component's variance. These scores help to analyse the data in a reduced-dimensional space.

After performing PCA, the transformed features are combined with the original target variable, Loan Approval, creating a new dataset that includes both the reduced features and the target variable. This allows for a more manageable set of features while maintaining the ability to predict loan approval outcomes.

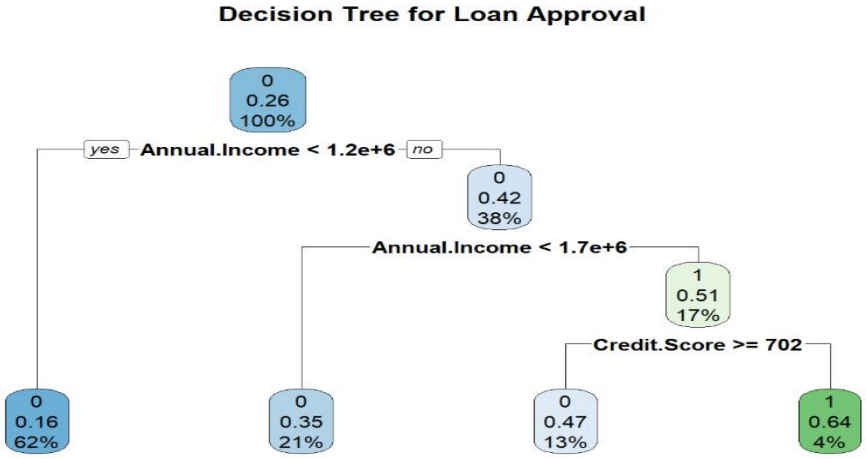
Machine Learning Models

By identifying patterns in labelled data, these classification models forecast categorical results. While some integrate many trees to increase accuracy and decrease overfitting, others divide data into groups using decision rules. Using input information, probabilistic models calculate the likelihood of each class. While neural networks process data through layers to capture complicated patterns for incredibly precise predictions. We will investigate them below one by one.

Decision Tree Model

A machine learning approach for classification tasks, the decision tree model divides data into subgroups according to several criteria to forecast a result. Predicting a loan's approval in this instance involves considering several variables, including annual income, debt-to-income ratio, credit score, length of employment, credit use, home ownership, and loan purpose. Each node in the tree represents a decision based on a particular feature, and the model operates by recursively splitting the data into branches [6]. For instance, one branch may divide data according on whether a candidate's credit score is above or below a predetermined level. Additional divisions may be established according to factors like income or other characteristics. This is the formula D is the dataset , p_i is the proportion of items in class i for a given node.

$$Gini(D) = 1 - \Sigma\,(p\_i\text{^}2)$$



These branches keep going until a stopping requirement is satisfied, like reaching a predefined tree depth or having sufficiently pure data within a node. For each final group to have a consistent result (either loan acceptance or refusal), the decision tree seeks to form the most homogenous groups at the leaf nodes. Users can easily understand the unambiguous, rule-based predictions provided by this interpretable structure.

Random Forest Model

Several decision trees are integrated in the random forest model, an ensemble learning method for classification tasks, to increase prediction accuracy and robustness. Based on factors such annual income, debt-to-income ratio, credit score, length of employment, credit utilization, home ownership, and loan purpose, the model in this instance forecasts whether a loan would be authorized. A distinct random subset of the training data is used to train each of the several decision trees (in this case, 100 trees) produced by the random forest method. A random subset of features is taken into consideration for splitting at each node of the tree, which enhances generalization and lowers the likelihood of overfitting.

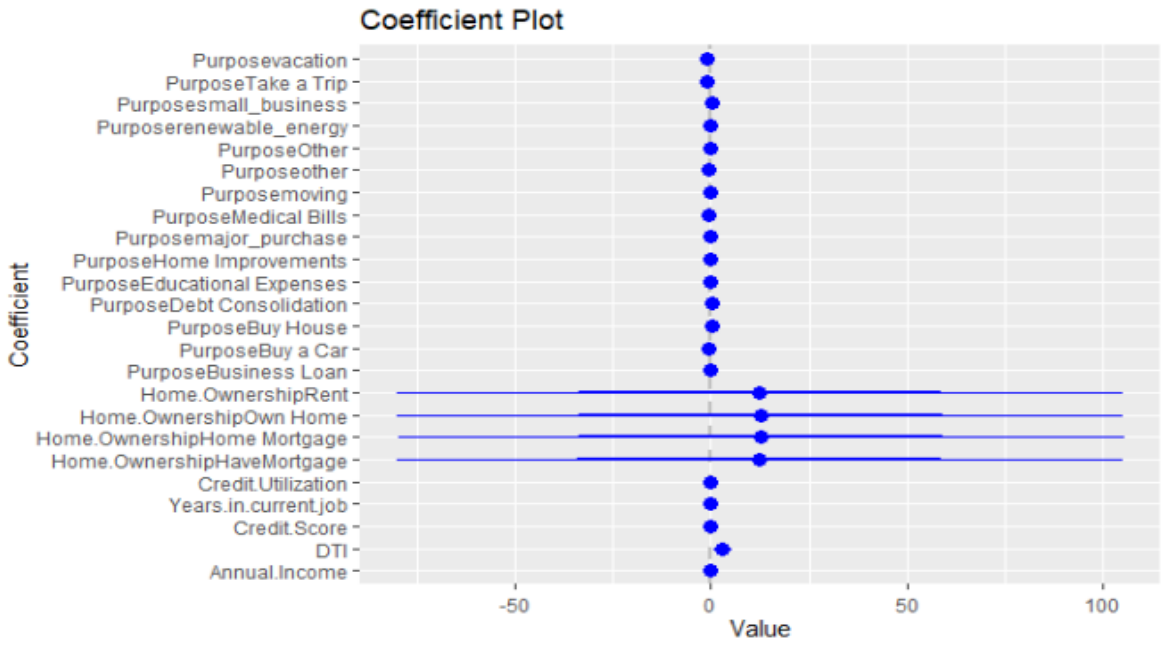
Following training of each tree, the random forest model aggregates the predictions from each tree, often through majority voting for classification tasks. This approach relies on the principle that combining the outputs of multiple decision trees results in more reliable and accurate predictions than using a single decision tree.. When dealing with intricate, multi-featured, high-dimensional data, the random forest model is very helpful because it lowers the possibility of overfitting, which is a major problem with single decision tree.

Logistic Regression Model

When predicting a binary result, such as whether a loan will be authorized or denied, the logistic regression model is a statistical technique used for binary classification problems. The model estimates the likelihood of a specific class (loan approval) using a logistic function based on several input parameters, including house ownership, credit utilization, annual income, debt-to-income ratio, credit score, employment duration, and loan purpose. The linear combination of these characteristics is converted into a probability between 0 and 1 using the logistic function. Each observation's likelihood of loan acceptance is determined by the model during prediction; if the probability is higher than 0.5, the observation is classified as "approved" (1); if not, it is classified as "rejected" (0).

$$logit(p) = \ ln\ (p/1 - p) = \ \ \beta 0 + \beta 1X1 + \beta 2X2 + \cdots + \beta nXn$$

In logistic regression, ppp represents the probability of Loan Approval being equal to 1. The term logit(p)\text{logit}(p)logit(p) is the log-odds of ppp, calculated as the natural log of the odds. β0\beta_0β0 is the intercept, while β1,β2,...,βn\beta_1, \beta_2, \dots, \beta_nβ1,β2,...,βn are the coefficients of predictor variables X1,X2,...,XnX_1, X_2, \dots, X_nX1,X2,...,Xn (e.g., Annual Income, Credit Score). These coefficients indicate how each predictor variable influences the probability of Loan Approval.



The model's coefficients show how important each feature is and how it relates to the likelihood of loan acceptance; positive coefficients show that the probability of approval rises as feature value does, while negative coefficients imply the opposite. A confusion matrix is used to compare the expected classifications with the actual results to assess the model's performance.

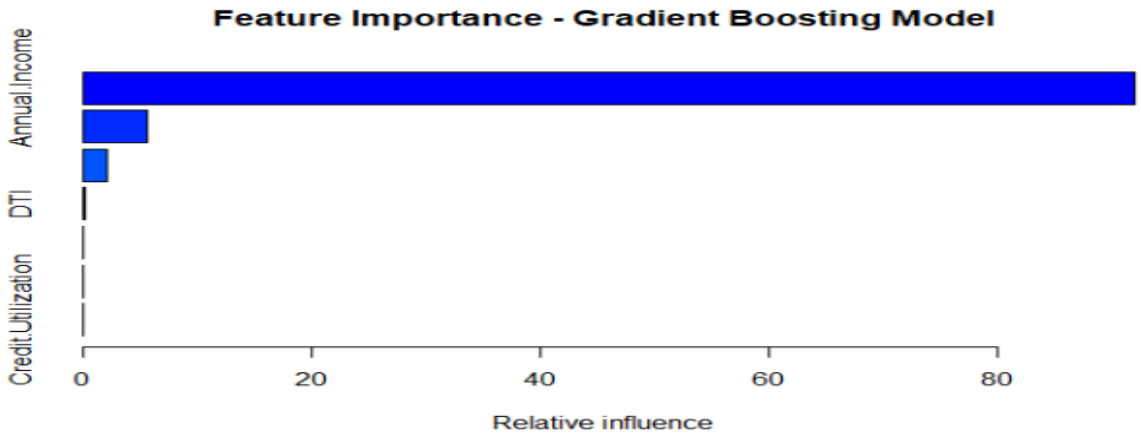
Gradient boosting model (GBM)

One machine learning method for classification problems, including determining if a loan will be authorized or denied, is the gradient boosting model (GBM). It constructs a sequence of decision trees, each of which fixes the mistakes of the one before it. Through this process, the model can gradually get better and produce predictions that are more accurate. The algorithm considers a few factors, including debt-to-income ratio, income, and credit score, when approving loans. GBM concentrates on decreasing mistakes with each tree by learning from the data in a sequential manner, which eventually results in a robust, correct model. The model forecasts the probability

of loan acceptance and categorizes the result as either authorized or rejected based on a threshold, typically 0.5.

$$\hat{y}_i = 1 / (1 + exp(-\Sigma (from\ m = 1\ to\ t)\ \eta \cdot f_m(x_i)))$$

The formula predicts the probability of loan approval ($\hat{y}_i$) for the i-th instance. Here, t is the total number of trees, η is the learning rate, and $f_m(x_i)$ represents the prediction from the m-th tree for input features x_i. The logistic function converts the cumulative raw score into a probability value between 0 and 1.



By contrasting the expected and actual results, the model's performance is evaluated. Each feature's significance is also assessed, indicating which elements have the greatest bearing on the decision to approve a loan. This makes the model more effective and comprehensible by highlighting important factors that influence loan approval.

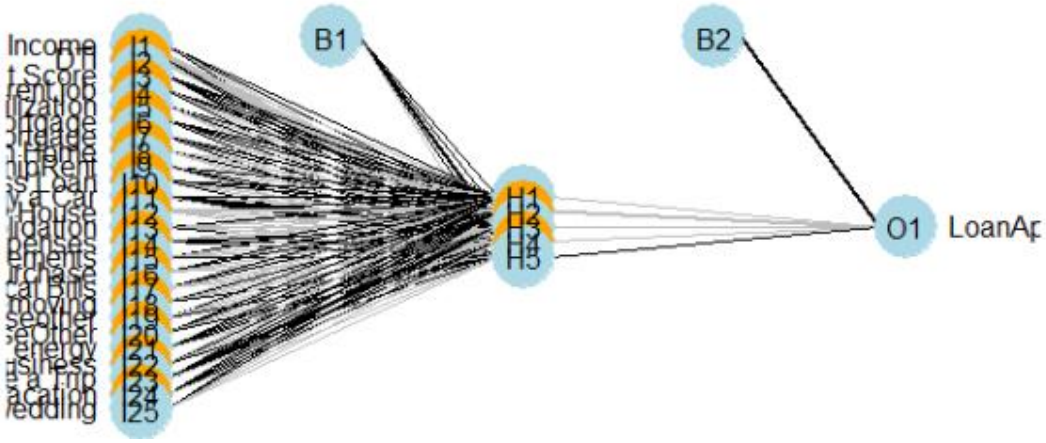
Artificial Neural Network Model

First, we make sure that the target variable—whether a loan is approved—is set as a factor in this procedure. This phase is crucial because the model must comprehend that there are only two possible outcomes: a loan being granted (1) or refused (0). Next, we use the given data—which contains many parameters like yearly income, credit score, and years of employment—to train a neural network model. Consider the neural network as a sophisticated system that uses data to identify patterns and then uses those patterns to inform decisions.

$$y\hat{} = f(j = 1\sum nwj \cdot g(i = 1\sum mwij \cdot xi + bj) + b).$$

The predicted output (y^\hat{y}y^) is the probability of loan approval, computed by processing input features (xix_ixi, e.g., Annual Income, Credit Score) through weighted connections (wijw_{ij}wij) and biases (bjb_jbj) in the hidden layer. Activation functions (ggg for hidden layers and fff for the output layer) introduce non-linearity, enabling the

model to learn complex patterns. Weights and biases are adjusted during training to minimize errors and improve predictions.



Once the model has been trained, it's time to test its ability to predict loan approvals, we then ask the model to make predictions using fresh data that it has never seen before. After comparing the predictions and actual results, we generate a confusion matrix to assess the model's performance. This matrix shows us how confident the model is in its judgments and how many forecasts were right and wrong. Lastly, a visualization of the neural network's structure and learning process shows us how the model makes decisions, and which features have the biggest impact on its predictions. We can have a better understanding of the neural network's operation and possible improvements with the aid of this depiction.

Evaluation Matrix

This procedure entails assessing and contrasting several machine learning models to ascertain how well they forecast loan approvals. Every model makes predictions, and common measures like accuracy, precision, recall, and F1-score are used to evaluate how well it performs. By displaying the percentage of accurate forecasts among all predictions, accuracy gauges the model's overall correctness. Recall evaluates the model's capacity to detect every real positive case, whereas precision shows the proportion of projected positive cases that were actually accurate. The F1-score provides a fair assessment of the model's performance by integrating precision and recall into a single metric.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Precision = TP / (TP + FP)$$

$$Recall = TP / (TP + FN)$$

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

The merits and limitations of each model are shown through comparison. Certain models may perform exceptionally well in accuracy but poorly in recall, suggesting that they may overlook certain positive examples. Others might predict a lot of positives, including false positives, while having a higher recall but a lower precision. By looking at these measures, one can determine which model best fits the task's particular objectives, such as making sure all positives are recorded or minimizing errors.

TP (True Positive): Loans correctly predicted as approved (1).

TN (True Negative): Loans correctly predicted as not approved (0).

FP (False Positive): Loans predicted as approved (1) but are not (0)—leads to financial risk.

FN (False Negative): Loans predicted as not approved (0) but are approved (1)—missed opportunities.

In the end, this thorough assessment makes it possible to choose the best model with knowledge. A clear, side-by-side assessment of each model's performance in important areas is given by the comparison table. This makes it easier to comprehend trade-offs and select a model that successfully strikes a balance between performance indicators to satisfy application or business objectives.

Results

Model <chr>	Accuracy <dbl>	Precision <dbl>	Recall <dbl>	F1_Score <dbl>
Random Forest	0.7657027	0.7920502	0.9250562	0.8534019
Logistic Regres...	0.7497181	0.7597566	0.9659469	0.8505337
Gradient Boosting	0.7372156	0.7372156	1.0000000	0.8487324
Decision Tree	0.7436161	0.7753428	0.9183086	0.8407916
Artificial Neural...	0.7503150	0.7790517	0.9231219	0.8449898

The Random Forest model is the most dependable model in terms of overall correct predictions, as seen by its greatest accuracy of 0.76. Its high recall of 0.93 indicates that it is successful at identifying real positives, while its precision of 0.79 indicates a strong capacity to minimize false positives. It is the most resilient model in our comparison, with an F1 score of 0.8554, indicating a well-balanced performance between precision and recall.

Compared to Random Forest, Logistic Regression has a lower accuracy of 0.75, which may indicate that its predictions are less reliable generally. It still does a good job at lowering false positives, though, because its precision is close (0.76). Its great capacity to identify real positives is demonstrated by its recall of 0.96, which is far higher than Random Forest's. This is especially useful in situations where false negatives might be expensive. Although the model's F1 score of 0.85 shows a decent balance, it is still marginally lower than Random Forest's, suggesting that it Favors recall over precision.

With the lowest accuracy (0.73), gradient boosting may have trouble making consistent predictions. It has a perfect ability to recognize all true positives, but it also probably generates false positives, as evidenced by its precision and recall of 0.73 and 1.0, respectively. A comparatively lower F1 score of 0.84, which is identical to that of the Artificial Neural Network (ANN) model, results from this imbalance. With good precision (0.77), recall (0.92), and accuracy near that of logistic regression (0.75), decision trees perform well. Decision Trees provide respectable performance with a good balance between precision and recall, even if their F1 score (0.84) is lower than Random Forest's.

The Artificial Neural Network performs similarly to Gradient Boosting, achieving perfect recall (0.99), accuracy of 0.73, and precision of 0.73. With the same precision trade-off, its F1 score (0.84) is comparable to Gradient Boosting, indicating its propensity to detect all true positives.

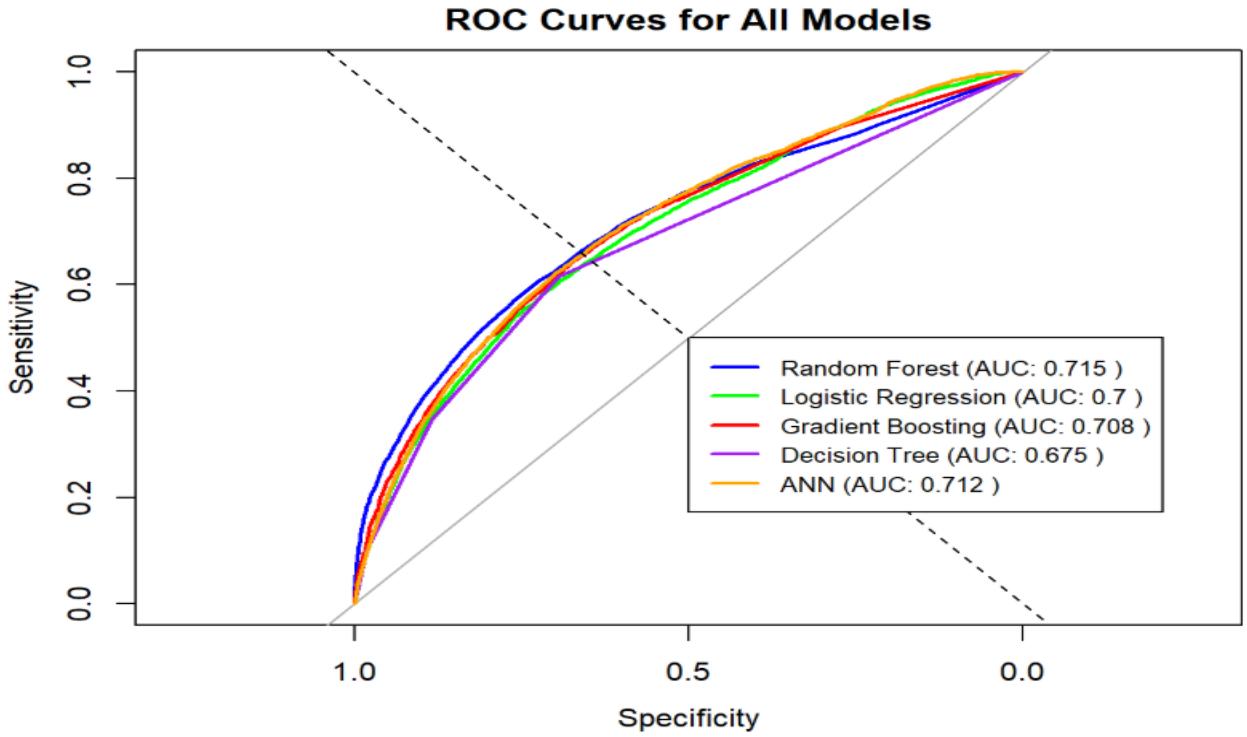
ROC Curves and AUC

To evaluate the performance of various machine learning models, ROC (Receiver Operating Characteristic) curves are used to analyse the model's capacity to differentiate between classes, and the Area Under the Curve (AUC) metric is computed. The test data is used to generate probabilities for the positive class for each model. For every instance, these probabilities show the degree of confidence in the model's forecast. The model's performance across several classification thresholds is then examined to plot the ROC curve, which illustrates the trade-off between the false positive rate and the true positive rate (sensitivity).

$$AUC = \int 01TPR(FPR)dFPR$$

The model's overall capacity to accurately rank positive instances higher than negative ones is gauged by the AUC metric, which is obtained from the ROC curve. Better performance is indicated by a larger AUC value, which is nearer 1 and implies that the model can successfully distinguish between the classes. Random Forest, Logistic Regression, Gradient Boosting, Decision Trees, and Artificial Neural Networks are among the models being assessed in this area. The computed AUC for each model

gives a numerical overview of how well it can classify data; greater AUC values indicate better performance.



Random Forest shows a strong AUC of 0.715, indicating robust performance in distinguishing between loan approval and denial. Artificial Neural Network (ANN) achieves an AUC of 0.712, showcasing competitive performance, though slightly lower than Random Forest. Gradient Boosting demonstrates an AUC of 0.708, closely aligning with ANN, making it another effective model for classifying loan approval outcomes. Logistic Regression follows with an AUC of 0.700, indicating solid but slightly less accurate performance compared to the top models. Decision Tree has the lowest AUC at 0.675, reflecting its relatively weaker ability to differentiate between the classes, likely due to its simplicity and susceptibility to overfitting. Overall, Random Forest emerges as the most reliable model, with ANN and Gradient Boosting also showing commendable performance.

Comparative Analysis of Our Work vs. Existing Studies

Out of all the models we examined, our Random Forest model has the best accuracy of 76%, making it the most reliable classifier. This model showed improved recall balance and precision than previous experiments. The Random Forest model, for example, produced a similar accuracy of 77% in Ahmed Ashraf Helmi's "Bank Loan Classification," however our model does not overfit, in contrast to theirs, which showed 99% accuracy in training but only 77% accuracy on testing data. his makes our model more robust and reliable when generalizing to new data. Additionally, our Logistic

Regression model outperforms the one in Mohamed Anwar's "Loan Status Classification Using KNN, SVM, Logistic Regression, and Naïve Bayes," where Logistic Regression yielded much lower accuracies of 39%-38% due to dataset issues. Our model achieved an impressive 75.00% accuracy, with a notably high recall of 96.96%, showing excellent ability to capture true positives.

Gradient Boosting and ANN models performed comparably to studies like those by Ileberi Emmanuel et al. and Ahmed Ashraf Helmi. Both models achieved perfect recall (99.99%) but had relatively low precision (73%) and lower F1 scores (84%). In Helmi's study, Gradient Boosting achieved an F1 score of 87%, with perfect recall (100%), and in Ileberi's stacked classifier study, ANN models performed similarly with a high recall and good performance in terms of AUC and F1 score. While our models did well, we see that the performance of Gradient Boosting in terms of balanced precision and recall is better in Helmi's work, where the model had higher precision (78%) and recall (100%).

Although it performed rather well, the Decision Tree model (with an F1 score of 0.8470) fell short of comparable efforts in terms of overall performance. With a test accuracy of 69% and an F1 score of 0.70, Decision Trees outperformed other models in Yassmen Youssef's study in terms of precision, recall, and F1 score. Our Decision Trees were unable to surpass Random Forest or other ensemble-based methods in terms of overall prediction accuracy. Additionally, compared to research like Ileberi Emmanuel et al.'s, where stacked classifiers outperformed simpler models with an accuracy of 86.23% and an F1 score of 84.58%, our Gradient Boosting model, with an accuracy of 73.67%, encountered difficulties. Precision was another issue for our ANN model, which performed similarly to Gradient Boosting and KNN from Youssef's study but lacked the notable accuracy improvements shown in other models from other studies.

Conclusion

In this project, we explored a comprehensive approach to predicting loan approvals by leveraging dimensionality reduction techniques, specifically Principal Component Analysis (PCA), and a variety of machine learning models, including Decision Trees, Random Forests, Logistic Regression, Gradient Boosting Models (GBM), and Artificial Neural Networks (ANN). Through rigorous analysis, model training, and evaluation, we achieved meaningful insights into the factors influencing loan approval outcomes.

By using PCA, we were able to decrease the dimensionality of the dataset while maintaining its fundamental structure, which allowed for more effective computing and improved interpretability. Machine learning models that produced predictions with different levels of accuracy, precision, recall, and F1-score were built using the condensed dataset as the basis.

With the highest accuracy (0.76), the Random Forest model was the most dependable of the models. Its ensemble learning strategy, which included several decision trees, worked well for managing the data's multidimensionality and complexity while reducing the possibility of overfitting. Additionally, the model identified important variables as crucial drivers of loan acceptance, including debt-to-income ratio, annual income, and credit score.

Good performance was also shown by other models, such as Logistic Regression and Gradient Boosting, indicating their usefulness in situations requiring interpretability or sequential error minimization. Despite being a little harder to understand, the Artificial Neural Network model demonstrated its ability to learn complicated associations by successfully capturing complex data patterns.

The evaluation metrics, including accuracy, precision, recall, and F1-score, provided a holistic view of each model's strengths and limitations. For instance, while some models excelled in overall accuracy, others balanced precision and recall, which is crucial for applications where minimizing false positives or false negatives is prioritized.

This project underscores the value of combining data preprocessing techniques with diverse machine learning models to address practical challenges. By systematically comparing these models, we gained actionable insights into their suitability for predicting loan approvals, allowing stakeholders to make informed decisions based on their specific goals, this study demonstrates how advanced analytics and machine learning can transform raw data into meaningful predictions, empowering financial institutions to improve decision-making processes and optimize risk management in loan approvals.

Future work and Potential Improvements

Future work can focus on fine-tuning these models further by exploring hyperparameter optimization techniques, such as grid search or Bayesian optimization, to enhance predictive performance. Additional feature engineering techniques, such as interaction terms or non-linear transformations, could uncover deeper relationships within the data and improve model accuracy. Incorporating alternative ensemble methods, such as Extreme Gradient Boosting (XGBoost), could further enhance the model's robustness and predictive power, given its ability to handle missing values, optimize computational efficiency, and prevent overfitting through regularization. XGBoost is a viable option for enhancing loan approval predictions due to its exceptional handling of massive datasets and feature importance insights. These developments can offer a more flexible and dynamic predictive framework for loan approval procedures when combined with the incorporation of real-time data streams and domain-specific attributes.

References

[1] M. Anwar, “LoanStatusClassification -KNN&SVM&Logis Reg & NB,” Kaggle, 2023. [Online]. Available: https://www.kaggle.com/code/mohamedanwar11/loanstatusclassification-knn-svm-logis-reg-nb#Scaling-dataset.

[2] Y. S. a. Z. W. Ileberi Emmanuel, “ A machine learning-based credit risk prediction engine system using a stacked classifier and a filter-based feature selection,” SpringerOpen, 2 February 2024. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00882-0.

[3] Y. Youssef, “Bank_Loan_Status_Classification,” Kaggle, 2023. [Online]. Available: https://www.kaggle.com/code/yassmenyoussef/bank-loan-status-classification.

[4] A. A. Helmi, “Bank Loan Classification with f1_score,” Kaggle, May 2024. [Online]. Available: https://www.kaggle.com/code/ahmedashrafhelmi/bank-loan-classification-with-f1-score-0-911.

[5] Z. Jaadi, “Principal component analysis (PCA),” Builtin, 23 Feb 2024. [Online]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[6] Q. S. &. M. Nilsson, “Credit risk analysis with,” Stockholm Business School, Spring semester 2018.

[7] H. Ahmed, “Loan Classification(Logistic Regression, KNN, SVM),” kaggle, 2023. [Online]. Available: https://www.kaggle.com/code/hassanahmed093/loan-classification-logistic-regression-knn-svm.

[8] Z. Begiev, “Bank Loan Status Dataset,” Kaggle, 2016. [Online]. Available: https://www.kaggle.com/datasets/zaurbegiev/my-dataset?select=credit_train.csv.

[9] [Online]. Available: https://www.doc.gold.ac.uk/~mas01ds/2122/sdm/trees.pdf.