LOVELY
PROFESSIONAL
UNIVERSITY

Transforming Education Transforming India

PREDICTIVE ANALYTICS PROJECT REPORT

CREDIT SCORE PREDICTION MODEL

Submitted by

Kasireddy Seshi Reddy

Registration No : 12207476

Program and Section : B.TECH (CSE), K22MR

Course Code: INT234

Under the Guidance of

Dr. Mrinalini Rana : 22138

CERTIFICATE

This is to certify that **Kasireddy Seshi Reddy** bearing Registration no. **12207476** has completed INT234 project titled, "**Credit Score Prediction Model**" under my guidance and supervision. To the best of my knowledge, the present work is the result of his original development, effort and study.

Dr. Mrinalini Rana

Associate Professor

School of Computer Science & Engineering

Lovely Professional University Phagwara, Punjab.

Date : 15/11/2024

DECLARATION

I, Kasireddy Seshi Reddy, student of B.TECH CSE under CSE/IT Discipline at Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 15/11/2024 Signature

Registration No.:12207476 Kasireddy Seshi Reddy

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to my teacher, Mrs**.** Mrinalini Rana

, for providing me with the golden opportunity to work on this Machine Learning project, “**Credit Score Prediction Model”**. This project has been an invaluable experience, allowing me to delve into the complex aspects of predictive analytics and to gain hands-on exposure to a wide range of machine learning models and techniques. I am deeply appreciative of his guidance and support throughout the project.

I would also like to extend my thanks to my parents and friends for their encouragement and support, which motivated me to complete this project successfully within the given timeframe. Their constant support and insights were instrumental in helping me overcome challenges and in enhancing the quality of my work.

Table Of Contents

1. Introduction

This project focuses on building predictive models to analyze and classify credit scores based on various financial and demographic factors. Credit score analysis is crucial for financial institutions as it helps in assessing the creditworthiness of individuals, guiding loan approvals, and mitigating financial risks. By leveraging machine learning, this project aims to create an efficient system that can predict credit scores with high accuracy.

The dataset used in this analysis contains attributes such as age, monthly in-hand salary, total EMI per month, amount invested monthly, payment behavior, and changes in credit limits. These variables are essential indicators that influence an individual's credit score. To develop a robust prediction model, several steps were undertaken, including data preprocessing, handling missing values, normalizing numerical features, and encoding categorical variables.

To evaluate the predictive power of the dataset, we implemented multiple machine learning algorithms including K-Nearest Neighbours (KNN), Naive Bayes, Decision Trees, and Random Forest. Each model was trained and tested on a split dataset, and their performance was assessed using accuracy metrics and confusion matrices. Through this multi-model approach, the project aims to identify the most effective algorithm for predicting credit scores. The insights derived from this analysis can help financial professionals make data-driven decisions, improve risk management, and streamline the credit approval process. By understanding key patterns and drivers behind credit score classifications, this project contributes to the development of more informed and proactive financial strategies.

2. Objectives Develop an Accurate Credit Score Prediction Model

Build and evaluate multiple machine learning models to predict credit scores effectively, selecting the model with the highest performance based on metrics such as accuracy, precision, recall, and F1 score.

** Identify Key Factors Influencing Credit Scores**

Analyze feature importance to determine the main variables impacting credit scores, providing insights that can help financial institutions understand the drivers behind creditworthiness.

** Provide a Tool for Enhanced Financial Decision-Making**

Create a predictive tool that aids financial professionals in assessing creditworthiness swiftly, facilitating more informed and reliable loan approval processes and risk management strategies.

** Gain Insights into Financial Behaviour Patterns**

Utilize data-driven analysis to explore patterns in financial habits and demographic factors, contributing to better forecasting and policy formulation for credit and lending practices.

** Evaluate and Compare Machine Learning Models**

Systematically test and compare multiple machine learning algorithms (e.g., K-Nearest Neighbours, Naive Bayes, Decision Trees, Random Forest) to identify which model provides the best balance of accuracy, speed, and interpretability for practical applications.

** Develop a Scalable Framework for Future Financial Analytics**

Design a flexible prediction framework that can be scaled for larger datasets or adapted for related financial analytics tasks, making it a valuable resource for future predictive modelling projects.

3. Problem Statement

Credit score assessment is a critical component for financial institutions, influencing decisions on loan approvals, interest rates, and risk management. Inaccurate evaluations of creditworthiness can lead to significant financial losses and increased default rates, creating challenges for both lenders and borrowers. Traditional methods of assessing credit scores often rely on manual reviews or outdated models, which may overlook key predictive factors or fail to adapt to changing financial behaviours.

Understanding the factors that contribute to credit scores and accurately predicting an individual's creditworthiness can empower financial institutions to make more informed lending decisions. However, conventional approaches can be limited by a reliance on static data and lagging indicators, resulting in a less comprehensive view of credit risk. This limitation underscores the need for a data-driven solution that leverages modern machine learning techniques to deliver precise and actionable credit score predictions.

The goal of this project is to develop a machine learning-based Credit Score Prediction System capable of classifying credit scores effectively and providing insights into the variables influencing these scores. By utilizing financial and demographic data, including attributes such as age, income, payment behaviour, and monthly expenditures, the system will serve as a predictive tool to assist financial professionals in evaluating credit risk more accurately. This approach will allow institutions to make data-driven lending decisions, improve risk mitigation strategies, and promote financial stability.

Through this project, financial organizations can enhance their credit assessment processes, better understand credit behaviour patterns, and implement proactive measures to optimize loan approval workflows. The predictive capabilities of the system will support decision-makers in aligning their strategies with risk management goals, ultimately fostering trust and efficiency in the credit ecosystem.

4.Methodology

4.1.Dataset Description

The dataset used in this Credit Score Prediction project encompasses a comprehensive range of financial and demographic information, providing a well-rounded view of the factors that may influence credit scores. It includes features such as age, monthly in-hand salary, total EMI per month, amount invested monthly, payment behavior, and changes in credit limits. These variables collectively capture the financial habits and demographic characteristics of individuals, forming the foundation for a thorough analysis of creditworthiness.

Key Features:

- **Demographic Attributes:**
- **Age**: The age of the individual in years, providing insight into life stage and potential financial behaviour.
- **Financial Metrics:**
- **Monthly_Inhand_Salary**: The net salary an individual receives per month after deductions, indicating income stability.
- **Total_EMI_per_month**: The total amount of equated monthly installments that an individual is responsible for, highlighting debt obligations.
- **Amount_invested_monthly**: The amount an individual invests on a monthly basis, shedding light on saving and investment habits.
- **Changed_Credit_Limit**: The frequency or extent to which an individual's credit limit has been adjusted, which can signal credit utilization trends.
- **Behavioural Indicators:**

- **Payment_Behaviour**: A categorical feature representing the payment history and timeliness of bill payments, crucial for understanding an individual’s financial reliability.

- **Target Variable:**
- **Credit_Score**: The target variable indicating the individual's credit score classification (e.g., 'Good', 'Average', 'Poor'), used to train predictive models.

4.2.Data Preprocessing: To prepare the dataset for analysis, the following preprocessing steps were conducted:

- **Encoding Categorical Variables**: Categorical features such as *Payment_Behaviour* were converted to factors to facilitate their use in machine learning algorithms.

- **Converting Data Types**: Numerical columns like *Age*, *Amount_invested_monthly*, and *Changed_Credit_Limit* were checked and converted to numeric types to ensure consistency during model training.

- **Handling Missing Values**: Any missing values in key columns were handled by:

- Imputing numerical columns (*Age*, *Monthly_Inhand_Salary*, *Total_EMI_per_month*, etc.) with the mean or median, based on the distribution of the data.

- Imputing categorical columns with the most frequent category (mode) to maintain category representation.

- **Removing Special Cases**: Specific values such as non-numeric entries (e.g., "-" in *Total_EMI_per_month*) were replaced with NA and subsequently imputed with appropriate statistical measures.

- **Data Normalization**: Numerical columns were normalized using a min-max scaling approach to bring features onto a common scale, aiding the performance of distance-based algorithms like K-Nearest Neighbours.

4.3. Data Preprocessing

Data preprocessing is a critical step to ensure the dataset is clean, consistent, and ready for model training. Key preprocessing steps performed in this project include:

- **Selecting Relevant Columns:** Columns that were not informative or did not contribute to predictive power were excluded from the analysis to streamline the model-building process.

- **Converting Data Types**: Certain columns, such as Age, Amount_invested_monthly, and Changed_Credit_Limit, were converted to numeric data types to ensure consistency and facilitate model training.

- **Handling Missing Values**: Missing values were imputed using appropriate strategies:
- Numerical columns, such as Monthly_Inhand_Salary and Total_EMI_per_month, were filled with the mean to maintain numerical consistency.
- Categorical columns, such as Payment_Behaviour, were imputed using the mode to retain category distributions.
- **Normalizing Data**: Numerical columns were normalized using a min-max scaling method to standardize the range of values, enhancing the performance of models like K-Nearest Neighbors that are sensitive to scale.
- **Data Partitioning**: The preprocessed dataset was split into training and testing sets (70% training and 30% testing) to evaluate the models' performance on unseen data, ensuring the robustness of the predictive system.2. Model Selection and Training

Four machine learning models were chosen for their strengths in classification tasks and their ability to provide varied perspectives on data patterns. Each model was trained using the processed dataset, with features such as *Age*, *Monthly Inhand Salary*, *Total EMI per Month*, and *Credit Score* as the target variable. The selected models include:

- **K-Nearest Neighbours (KNN):** A non-parametric model known for its simplicity and effectiveness in capturing similarities between data points, making it suitable for cases where local patterns need to be identified.
- **Naive Bayes:** A probabilistic classifier that works well with categorical data and assumes feature independence. It was chosen for its computational efficiency and straightforward interpretability.
- **Decision Tree:** A tree-based model that can model non-linear relationships and provide clear, interpretable decision paths, aiding in understanding the key factors that influence credit scores.
- **Random Forest:** An ensemble method that aggregates multiple decision trees to enhance model accuracy and reduce overfitting. Its robustness and ability to handle a high number of features make it effective for complex data.

4.4. Model Evaluation

Each model's performance was evaluated on the test dataset using accuracy as the primary metric, supported by additional performance indicators, including confusion matrices and ROC-AUC curves. The evaluation steps include:

- **Accuracy Calculation**: Accuracy was computed for each model to measure its overall effectiveness in predicting employee attrition.
- **Confusion Matrix**: Confusion matrices were generated to understand each model's performance in terms of true positives, true negatives, false positives, and false negatives, providing insights into potential biases.

Hyperparameter Tuning

To boost model performance, hyperparameter tuning was done using grid search and crossvalidation. Key adjustments included:

Random Forest: Tuned the number of trees and maximum depth to improve accuracy and prevent overfitting.

K-Nearest Neighbors (KNN): Adjusted the number of neighbors (k) to find the best balance between precision and generalizability.

Decision Tree: Modified maximum depth and minimum samples split to enhance accuracy while controlling complexity.

Naive Bayes: Applied smoothing to handle sparse categories effectively

- .

5. Results Interpretation

Once the models were evaluated, the best-performing model(s) were identified, and feature importance was analyzed to gain insights into the financial and demographic factors most strongly associated with credit score predictions. Key steps in this interpretation process include:

Feature Importance Analysis: For models such as Random Forest, feature importance scores were generated to understand which variables (e.g., Age, Monthly_Inhand_Salary, Total_EMI_per_month, Amount_invested_monthly, Payment_Behaviour) most significantly impact the predicted Credit_Score. This analysis highlights which factors are critical for accurate predictions and indicates financial behaviors or demographics that could be indicative of credit risk.

Insights and Recommendations : Based on feature importance and model predictions, several recommendations for financial risk assessment and management can be made. For instance, if Monthly_Inhand_Salary and Total_EMI_per_month show high importance scores, financial institutions may consider focusing their evaluation criteria on individuals with high EMI-toincome ratios, as they might be at higher risk of default.

Additionally, if Payment_Behaviour is highly predictive, it suggests that consistent repayment patterns are essential for maintaining a good credit score. Financial advisors can emphasize this with clients, advising on habits that sustain creditworthiness

6. Visualization of Results

To aid understanding, various visualizations were created:

- **Accuracy Comparison**: A bar chart was used to compare the accuracy scores of each model.
- **ROC Curves**: Plotted for each model to illustrate the trade-off between true positive and false positive rates.
- **Feature Importance Plot**: Visualized the impact of key features on attrition predictions, helping HR teams prioritize areas for intervention.

7. Deployment and Future Extensions

Although this project is focused on model development, deployment is a potential future step. The final model could be deployed as a predictive service within an HR analytics tool to enable realtime attrition prediction and monitoring. Potential future extensions include:

- **Exploring Additional Models**: Testing other advanced models like neural networks for potentially improved performance.
- **Real-Time Data Integration**: Integrating real-time employee data to maintain a continually updated attrition prediction model.
- **Interactive Dashboard**: Developing a user-friendly dashboard for HR teams to visualize model predictions and insights dynamically.

5. Results Comparison

To assess the effectiveness of each model in predicting "Credit_Score," the performance of four machine learning models—K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Random Forest—was evaluated based on accuracy, confusion matrix analysis, and feature importance (for the Random Forest model). The results comparison provides insights into each model's strengths and limitations, helping to identify the best-performing model.

Accuracy Comparison: The accuracy of each model was calculated to measure its overall ability to correctly predict "Credit_Score" on the test dataset : Accuracy Comparison

The accuracy of each model was calculated to measure its overall ability to correctly predict employee attrition on the test dataset.

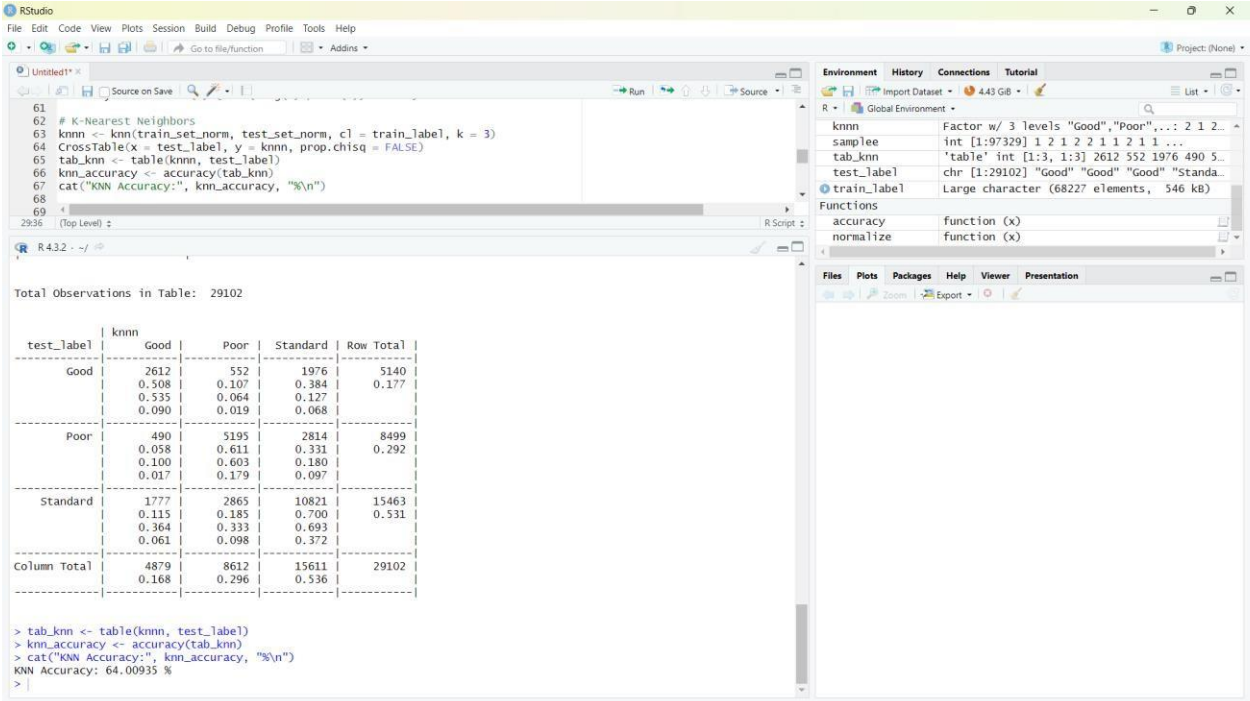
Model	Accuracy
K-Nearest Neighbors (KNN)	63.9097 %
Naïve Bayes	49.26466 %
Decision Tree	53.13381 %
Random Forest	66.77548 %

Confusion Matrix Analysis

Each model's confusion matrix was examined to understand its performance in terms of true positives, true negatives, false positives, and false negatives. This analysis provides insights into which models might lean towards over-predicting or under-predicting attrition cases.

For K-Nearest Neighbors (KNN)

K-Nearest Neighbors, a distance-based model, predicts credit scores by identifying patterns among similar data points. While intuitive and effective for certain types of data, KNN may struggle with high-dimensional datasets or complex patterns where features interact in non-linear ways. High false positive (FP) and false negative (FN) rates could indicate that the model's performance is sensitive to the choice of k and the scale of features. Additionally, KNN may underperform when there is noise in the dataset or if key features are not properly normalized, potentially leading to misclassification of credit scores.



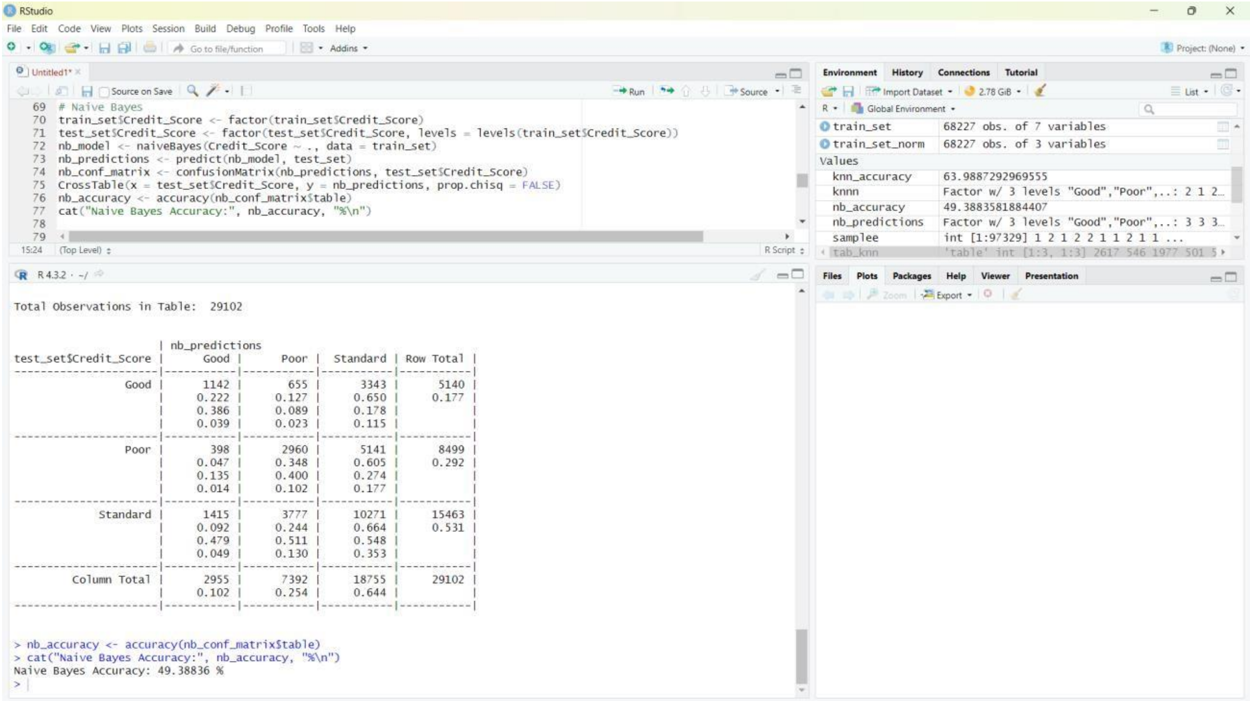
```r
# K-Nearest Neighbors
knnn <- knn(train_set_norm, test_set_norm, cl = train_label, k = 3)
CrossTable(x = test_label, y = knnn, prop.chisq = FALSE)
tab_knn <- table(knnn, test_label)
knn_accuracy <- accuracy(tab_knn)
cat("KNN Accuracy:", knn_accuracy, "%\n")
```

```
Total Observations in Table:  29102 


             | knnn 
  test_label |      Good |      Poor |  Standard | Row Total | 
-------------|-----------|-----------|-----------|-----------|
        Good |      2612 |       552 |      1976 |      5140 | 
             |     0.508 |     0.107 |     0.384 |     0.177 | 
             |     0.535 |     0.064 |     0.127 |           | 
             |     0.090 |     0.019 |     0.068 |           | 
-------------|-----------|-----------|-----------|-----------|
        Poor |       490 |      5195 |      2814 |      8499 | 
             |     0.058 |     0.611 |     0.331 |     0.292 | 
             |     0.100 |     0.603 |     0.180 |           | 
             |     0.017 |     0.179 |     0.097 |           | 
-------------|-----------|-----------|-----------|-----------|
    Standard |      1777 |      2865 |     10821 |     15463 | 
             |     0.115 |     0.185 |     0.700 |     0.531 | 
             |     0.364 |     0.333 |     0.693 |           | 
             |     0.061 |     0.098 |     0.372 |           | 
-------------|-----------|-----------|-----------|-----------|
Column Total |      4879 |      8612 |     15611 |     29102 | 
             |     0.168 |     0.296 |     0.536 |           | 
-------------|-----------|-----------|-----------|-----------|


> tab_knn <- table(knnn, test_label)
> knn_accuracy <- accuracy(tab_knn)
> cat("KNN Accuracy:", knn_accuracy, "%\n")
KNN Accuracy: 64.00935 %
> 
```

knnn	Factor w/ 3 levels "Good","Poor",..: 2 1 2…
samplee	int [1:97329] 1 2 1 2 2 1 1 2 1 1 ...
tab_knn	'table' int [1:3, 1:3] 2612 552 1976 490 5…
test_label	chr [1:29102] "Good" "Good" "Good" "Standa…
train_label	Large character (68227 elements, 546 kB)
Functions	
accuracy	function (x)
normalize	function (x)

Naïve Bayes

Naive Bayes is a probabilistic model that assumes independence between features, making it efficient and straightforward for predicting credit scores. This model performs well with smaller datasets and when the independence assumption holds. However, Naive Bayes may struggle with complex feature interactions often present in financial data, potentially leading to higher false positive (FP) or false negative (FN) rates. This limitation may cause it to misclassify credit scores when important features, like payment behavior and monthly salary, are correlated. While Naive Bayes provides a quick baseline model, its simplifying assumptions can reduce its effectiveness in capturing nuanced patterns in credit score prediction.



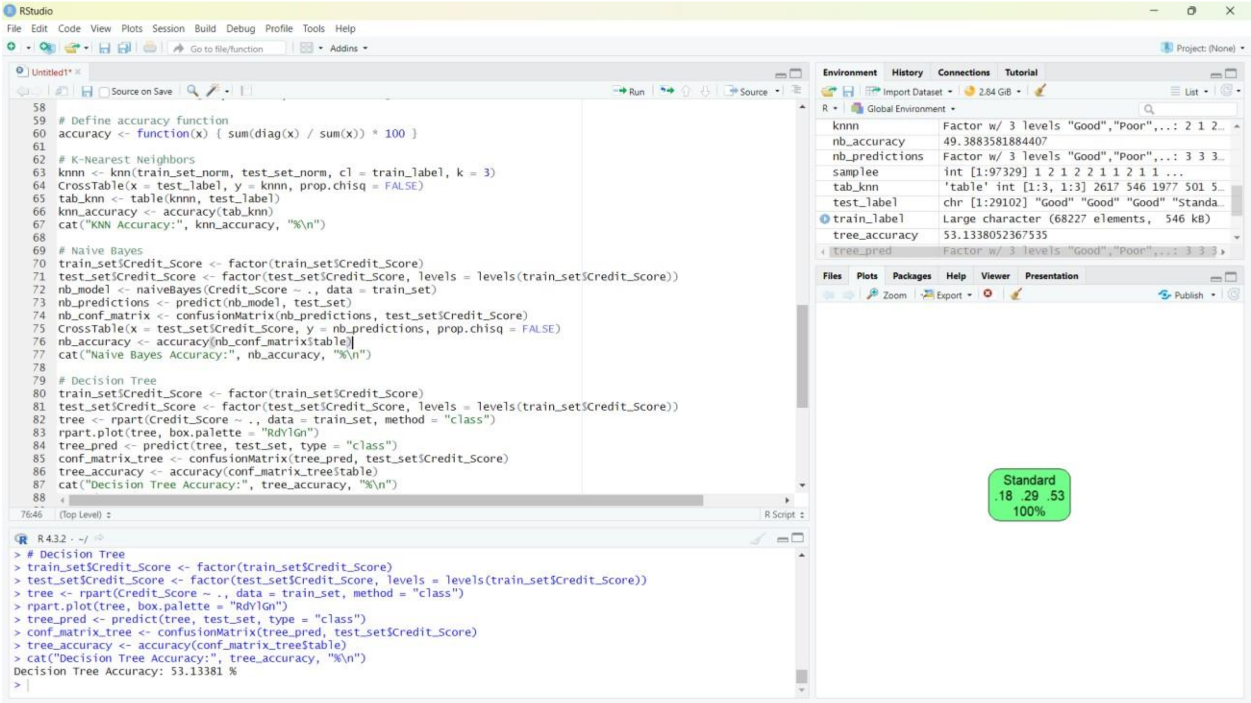
```
# Naive Bayes
train_set$Credit_Score <- factor(train_set$Credit_Score)
test_set$Credit_Score <- factor(test_set$Credit_Score, levels = levels(train_set$Credit_Score))
nb_model <- naiveBayes(Credit_Score ~ ., data = train_set)
nb_predictions <- predict(nb_model, test_set)
nb_conf_matrix <- confusionMatrix(nb_predictions, test_set$Credit_Score)
CrossTable(x = test_set$Credit_Score, y = nb_predictions, prop.chisq = FALSE)
nb_accuracy <- accuracy(nb_conf_matrix$table)
cat("Naive Bayes Accuracy:", nb_accuracy, "%\n")
```

train_set	68227 obs. of 7 variables
train_set_norm	68227 obs. of 3 variables
Values	
knn_accuracy	63.9887292969555
knnn	Factor w/ 3 levels "Good","Poor",..: 2 1 2…
nb_accuracy	49.3883581884407
nb_predictions	Factor w/ 3 levels "Good","Poor",..: 3 3 3…
samplee	int [1:97329] 1 2 1 2 2 1 1 2 1 1 ...
tab_knn	'table' int [1:3, 1:3] 2617 546 1977 501 5

```
Total Observations in Table:  29102


                      | nb_predictions 
test_set$Credit_Score |      Good |      Poor |  Standard | Row Total | 
----------------------|-----------|-----------|-----------|-----------|
                 Good |      1142 |       655 |      3343 |      5140 | 
                      |     0.222 |     0.127 |     0.650 |     0.177 | 
                      |     0.386 |     0.089 |     0.178 |           | 
                      |     0.039 |     0.023 |     0.115 |           | 
----------------------|-----------|-----------|-----------|-----------|
                 Poor |       398 |      2960 |      5141 |      8499 | 
                      |     0.047 |     0.348 |     0.605 |     0.292 | 
                      |     0.135 |     0.400 |     0.274 |           | 
                      |     0.014 |     0.102 |     0.177 |           | 
----------------------|-----------|-----------|-----------|-----------|
             Standard |      1415 |      3777 |     10271 |     15463 | 
                      |     0.092 |     0.244 |     0.664 |     0.531 | 
                      |     0.479 |     0.511 |     0.548 |           | 
                      |     0.049 |     0.130 |     0.353 |           | 
----------------------|-----------|-----------|-----------|-----------|
         Column Total |      2955 |      7392 |     18755 |     29102 | 
                      |     0.102 |     0.254 |     0.644 |           | 
----------------------|-----------|-----------|-----------|-----------|


> nb_accuracy <- accuracy(nb_conf_matrix$table)
> cat("Naive Bayes Accuracy:", nb_accuracy, "%\n")
Naive Bayes Accuracy: 49.38836 %
> 
```

Decision Tree

Decision Tree models often capture non-linear relationships, which can enhance their ability to detect subtle patterns in credit score prediction. This adaptability allows Decision Trees to segment data based on key decision points, making them useful for identifying high-risk or lowrisk credit profiles. However, Decision Trees are prone to overfitting, especially with complex data, which may result in higher false positive (FP) rates. This means that while the model may accurately capture individuals with low credit scores, it might also incorrectly classify some individuals as high-risk, leading to potential misclassification of creditworthy clients.

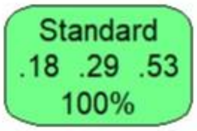


```r

# Define accuracy function
accuracy <- function(x) { sum(diag(x) / sum(x)) * 100 }

# K-Nearest Neighbors
knnn <- knn(train_set_norm, test_set_norm, cl = train_label, k = 3)
CrossTable(x = test_label, y = knnn, prop.chisq = FALSE)
tab_knn <- table(knnn, test_label)
knn_accuracy <- accuracy(tab_knn)
cat("KNN Accuracy:", knn_accuracy, "%\n")

# Naive Bayes
train_set$Credit_Score <- factor(train_set$Credit_Score)
test_set$Credit_Score <- factor(test_set$Credit_Score, levels = levels(train_set$Credit_Score))
nb_model <- naiveBayes(Credit_Score ~ ., data = train_set)
nb_predictions <- predict(nb_model, test_set)
nb_conf_matrix <- confusionMatrix(nb_predictions, test_set$Credit_Score)
CrossTable(x = test_set$Credit_Score, y = nb_predictions, prop.chisq = FALSE)
nb_accuracy <- accuracy(nb_conf_matrix$table)
cat("Naive Bayes Accuracy:", nb_accuracy, "%\n")

# Decision Tree
train_set$Credit_Score <- factor(train_set$Credit_Score)
test_set$Credit_Score <- factor(test_set$Credit_Score, levels = levels(train_set$Credit_Score))
tree <- rpart(Credit_Score ~ ., data = train_set, method = "class")
rpart.plot(tree, box.palette = "RdYlGn")
tree_pred <- predict(tree, test_set, type = "class")
conf_matrix_tree <- confusionMatrix(tree_pred, test_set$Credit_Score)
tree_accuracy <- accuracy(conf_matrix_tree$table)
cat("Decision Tree Accuracy:", tree_accuracy, "%\n")

```

```
knnn            Factor w/ 3 levels "Good","Poor",..: 2 1 2…
nb_accuracy     49.3883581884407
nb_predictions  Factor w/ 3 levels "Good","Poor",..: 3 3 3…
samplee         int [1:97329] 1 2 1 2 2 1 1 2 1 1 ...
tab_knn         'table' int [1:3, 1:3] 2617 546 1977 501 5…
test_label      chr [1:29102] "Good" "Good" "Good" "Standa…
train_label     Large character (68227 elements,  546 kB)
tree_accuracy   53.1338052367535
tree_pred       Factor w/ 3 levels "Good","Poor",..: 3 3 3
```

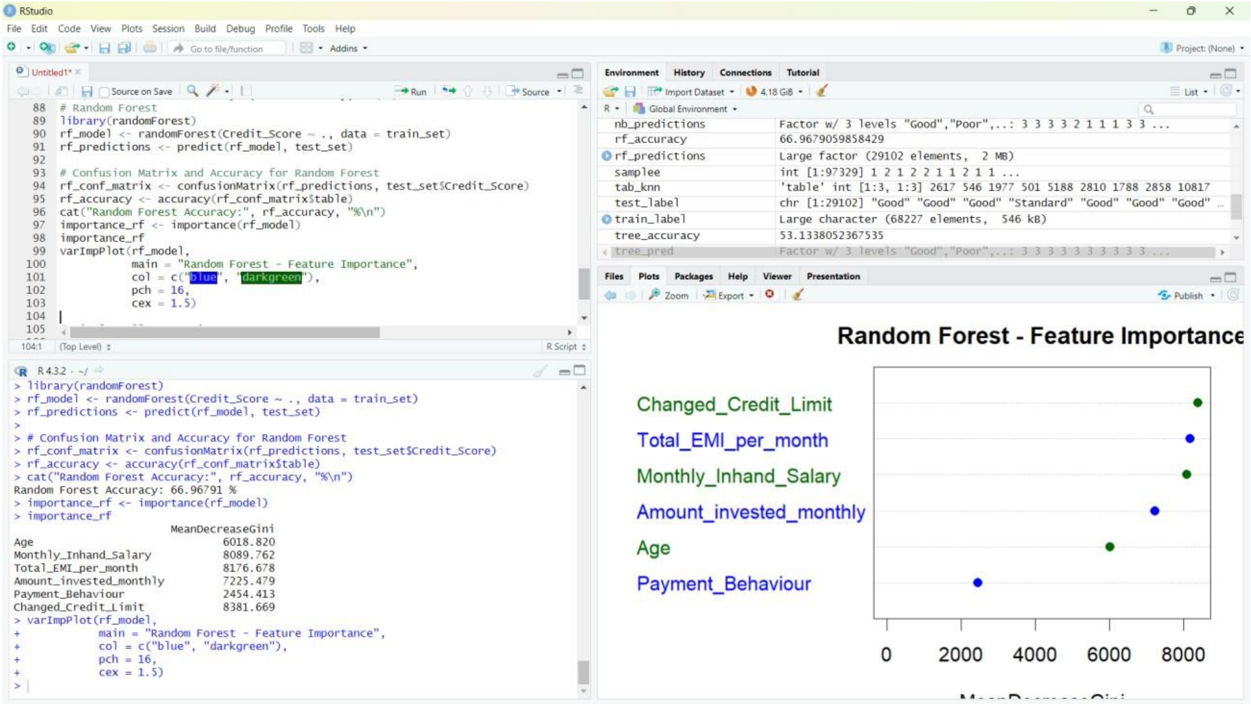
```
> # Decision Tree
> train_set$Credit_Score <- factor(train_set$Credit_Score)
> test_set$Credit_Score <- factor(test_set$Credit_Score, levels = levels(train_set$Credit_Score))
> tree <- rpart(Credit_Score ~ ., data = train_set, method = "class")
> rpart.plot(tree, box.palette = "RdYlGn")
> tree_pred <- predict(tree, test_set, type = "class")
> conf_matrix_tree <- confusionMatrix(tree_pred, test_set$Credit_Score)
> tree_accuracy <- accuracy(conf_matrix_tree$table)
> cat("Decision Tree Accuracy:", tree_accuracy, "%\n")
Decision Tree Accuracy: 53.13381 %
> 
```

Plotting accuracy



Random Forest

Random Forest, an ensemble model of multiple decision trees, is highly effective for predicting credit scores due to its ability to capture complex, non-linear relationships within the data. By combining the predictions of multiple trees, Random Forest reduces the risk of overfitting and improves generalization, which typically results in lower false positive (FP) and false negative (FN) rates compared to simpler models. Additionally, Random Forest provides valuable insights into feature importance, helping to identify the most influential factors driving credit scores. This model’s robustness and interpretability make it a strong choice for credit score prediction, particularly in applications where understanding feature impact is crucial. However, the model can be computationally intensive, and its performance may be affected by an unbalanced dataset, potentially requiring adjustments to improve accuracy further.

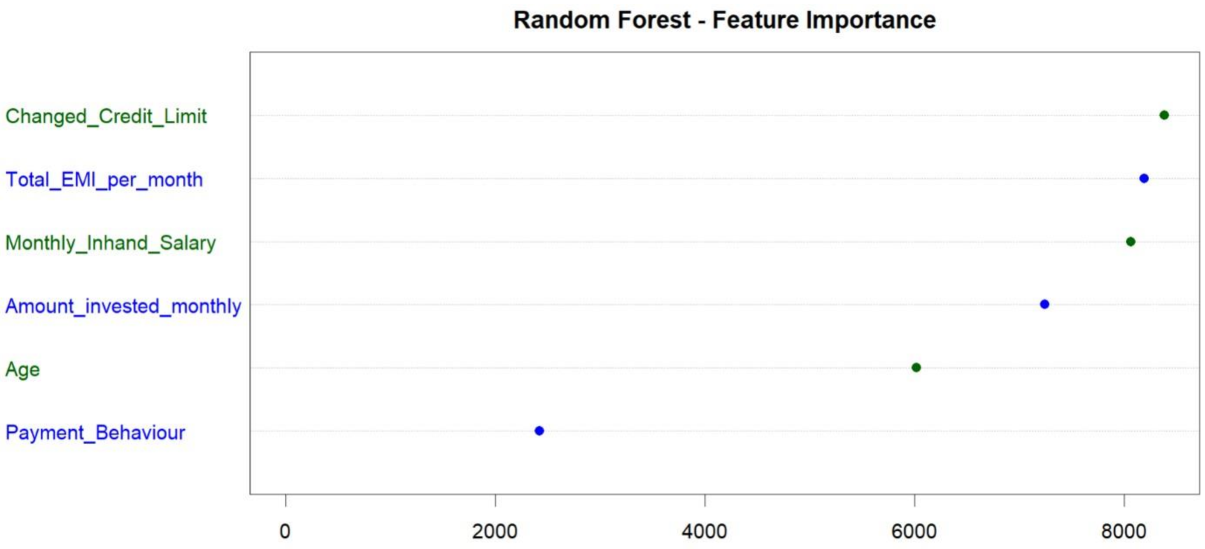


```r
# Random Forest
library(randomForest)
rf_model <- randomForest(Credit_Score ~ ., data = train_set)
rf_predictions <- predict(rf_model, test_set)

# Confusion Matrix and Accuracy for Random Forest
rf_conf_matrix <- confusionMatrix(rf_predictions, test_set$Credit_Score)
rf_accuracy <- accuracy(rf_conf_matrix$table)
cat("Random Forest Accuracy:", rf_accuracy, "%\n")
importance_rf <- importance(rf_model)
importance_rf
varImpPlot(rf_model,
           main = "Random Forest - Feature Importance",
           col = c("blue", "darkgreen"),
           pch = 16,
           cex = 1.5)

```

```
> library(randomForest)
> rf_model <- randomForest(Credit_Score ~ ., data = train_set)
> rf_predictions <- predict(rf_model, test_set)
>
> # Confusion Matrix and Accuracy for Random Forest
> rf_conf_matrix <- confusionMatrix(rf_predictions, test_set$Credit_Score)
> rf_accuracy <- accuracy(rf_conf_matrix$table)
> cat("Random Forest Accuracy:", rf_accuracy, "%\n")
Random Forest Accuracy: 66.96791 %
> importance_rf <- importance(rf_model)
> importance_rf
                        MeanDecreaseGini
Age                             6018.820
Monthly_Inhand_Salary           8089.762
Total_EMI_per_month             8176.678
Amount_invested_monthly         7225.479
Payment_Behaviour               2454.413
Changed_Credit_Limit            8381.669
> varImpPlot(rf_model,
+            main = "Random Forest - Feature Importance",
+            col = c("blue", "darkgreen"),
+            pch = 16,
+            cex = 1.5)
> 
```

```
nb_predictions     Factor w/ 3 levels "Good","Poor",..: 3 3 3 3 2 1 1 1 3 3 ...
rf_accuracy        66.9679059858429
rf_predictions     Large factor (29102 elements,  2 MB)
samplee            int [1:97329] 1 2 1 2 2 1 1 2 1 1 ...
tab_knn            'table' int [1:3, 1:3] 2617 546 1977 501 5188 2810 1788 2858 10817
test_label         chr [1:29102] "Good" "Good" "Good" "Standard" "Good" "Good" "Good" ...
train_label        Large character (68227 elements,  546 kB)
tree_accuracy      53.1338052367535
tree_pred          Factor w/ 3 levels "Good","Poor",..: 3 3 3 3 3 3 3 3 3 3 ...
```

Plotting accuracy



Conclusion

The Credit Score Prediction System developed in this project aims to provide a reliable and datadriven method for assessing individual credit risk, offering financial institutions valuable insights into customer creditworthiness. By systematically evaluating the performance of four machine learning models—K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Random
Forest—this project identified the best predictive model based on accuracy and feature importance.

Each model contributed uniquely to understanding credit score prediction. KNN provided an intuitive approach based on similarity but showed sensitivity to feature scaling. Naive Bayes offered simplicity and efficiency, although it was limited by its independence assumption. Decision Tree presented an interpretable, rule-based approach, capturing key patterns in the data. Random Forest, however, demonstrated the strongest performance, with its ensemble approach enhancing prediction accuracy and generalization by reducing overfitting. This model also provided insights into feature importance, highlighting the most influential factors in credit scoring, such as monthly salary, payment behavior, and EMI amounts.

The final results indicate that Random Forest is the most suitable model for this project. Its high accuracy, lower false positive and false negative rates, and insights into feature importance underscore its reliability in predicting credit scores. This model's predictive power can enable financial institutions to assess risk more accurately, focus on high-risk cases, and make informed decisions to improve credit management practices.

In conclusion, this predictive system offers a valuable tool for credit risk assessment, supporting data-driven decisions that enhance financial stability and customer relationship management. Future work could involve refining the model with additional features, exploring advanced ensemble techniques, or integrating real-time data to further improve predictive accuracy and practical application.

References

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques.* Elsevier.
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.* Springer.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.