Credit Score Classification using ML Algorithms

Shruti George Parakkal
Humber Institute of Technology and Advanced Learning
Toronto, Canada
N01581574@humber.ca

Asmeet Saini
Humber Institute of Technology and Advanced Learning
Toronto, Canada
N01536273@humber.ca

Devang
Humber Institute of Technology and Advanced Learning
Toronto, Canada
N01597532@humber.ca

Kiran
Humber Institute of Technology and Advanced Learning
Toronto, Canada
N01559226@humber.ca

Khushali
Humber Institute of Technology and Advanced Learning
Toronto, Canada
N01598192@humber.ca

Nirali
Humber Institute of Technology and Advanced Learning
Toronto, Canada
N01598121@humber.ca

***Abstract*— Credit scoring is a fundamental tool in financial services for assessing the creditworthiness of individuals. This paper explores the application of machine learning techniques to classify customers into predefined credit score brackets – 'Good' (most likely to repay financial obligation), 'Standard', 'Poor' (highest possibility of defaulting on financial obligation) based on their banking and credit information. The dataset was sourced from Kaggle's repository. The project utilizes Python and its robust ecosystem of data science libraries, including Pandas for data manipulation, Scikit-learn for implementing machine learning algorithms, and Matplotlib for visualization. The goal is to automate and enhance the accuracy of creditworthiness assessments in a financial institution, reducing manual effort and improving decision-making processes. The effectiveness of different models, including KNN, Decision Tree, random forest, Naïve Bayes and Hierarchical Clustering is evaluated and compared using metrics such as accuracy. Based on the evaluation, the Random Forest Model outperforms other models with an accuracy score of 0.75 and contribute to understanding the practical implementation of machine learning in credit scoring and its implications for financial services and risk management.**

Keywords— Data mining, Credit scoring, Decision tree, Classification, Hierarchical Clustering

I. Business Problem

Financial institutions, including banks and credit card companies, need to accurately evaluate the creditworthiness of applicants to mitigate the risk of defaults and enhance profitability. Inadequate classification systems can result in the approval of high-risk applicants, leading to higher default rates and financial losses, or the rejection of creditworthy applicants, resulting in missed business opportunities and dissatisfied customers.

Thus, the business problem is to create an advanced credit card classification model that can accurately and efficiently assess applicants based on their credit risk. This model aims to reduce manual effort and enhance the decision-making process, ultimately improving the institution's risk management and customer satisfaction.

II. Introduction (*Heading 1*)

Credit risk is a critical concern for commercial banks and financial institutions worldwide, stemming from the potential for borrowers to default on loan obligations, leading to significant financial losses. Effective management of credit risk is essential to safeguarding the financial stability and sustainability of these institutions. In 2019, the outstanding business credit in Canada, the USA, and the UK reached substantial figures of $2,262 billion, $15,243 billion, and £18,582 million, respectively (Bank of Canada, 2020; Bank of England, 2020; USA Federal Reserve, 2020), underscoring the magnitude of financial exposure involved.

To mitigate these risks, financial institutions increasingly rely on credit scoring models that analyze historical credit data to assess the creditworthiness of borrowers. (Assessing credit risk of commercial customers using hybrid machine learning)

Credit scores guide lenders' decisions on loan approvals, interest rates, and credit limits, pivotal in managing financial risk. Beyond financial institutions, they influence individuals' access to credit, interest rates, insurance premiums, rental housing, and employment prospects, serving as critical indicators of financial health and enabling informed financial decisions and effective management.

Traditionally, these models evaluate factors such as payment history, debt-to-income ratio, and credit utilization to assign numerical scores indicative of borrowers' risk profiles. However, the evolving landscape of data analytics and machine learning presents new opportunities to enhance the accuracy and efficiency of credit risk assessment.

This paper explores the application of machine learning techniques to classify customers into predefined credit score brackets using comprehensive banking and credit information. The credit score brackets as per the data available from Kaggle is – 'Good' (most likely to repay financial obligation), 'Standard'(moderate likely to repay financial obligation), 'Poor' (less likely to repay financial obligation and highest possibility of defaulting on financial obligation). The first part of the analysis shows the necessary data preprocessing steps, including data cleaning, feature engineering, and selection, aimed at optimizing model performance. It then proceeds with data manipulation techniques like changing the categorical variables (good, standard and poor) in the Credit_Score column to binary numbers (0,1 and 2). We also plotted heatmap to identify variables with high correlation and used feature_importances_ attribute of the random_forest_classifier to see the importance of each feature. This process helped us identify our predictor variables for classifying credit_score. Finally, we split the data into training and validation sets to perform the five machine learning algorithms, KNN, decision trees, random forests, Naive Bayes and Hierarchical Clustering. Among all models,

Random Forest Model outperformed, with the highest accuracy score of 75%.

III. Literature Review

Credit scoring models estimate a borrower's credit risk, i.e., the likelihood of loan repayment, by analyzing various quantifiable characteristics of the borrower (Dinh & Kleimeier, 2007). Traditionally, these models have utilized statistical and operational research methods. Discriminant analysis and linear regression have been the most prevalent techniques for developing scorecards. These studies typically considered demographic variables (e.g., age, education, address, marital status) and behavioral characteristics (e.g., number of active loans, highest loan issued, payment defaults).

In recent years, machine and deep learning models have been applied to credit scoring in retail settings. Research by Kozodoi et al. (2019), Liu et al. (2019), Soui et al. (2019), and Zhang et al. (2019) illustrates this trend. Similar models have also been adopted in commercial contexts (Barboza et al., 2017; Bequé & Lessmann, 2017; Kvamme et al., 2018; Mai et al., 2019; Pérez Martín et al., 2018). These models integrate transactional data (e.g., number of accounts, customer credit balance) and business-specific variables (e.g., number of employees, geographic location of offices) (Barboza et al., 2017; Liang et al., 2016; Mai et al., 2019). Notably, Liu et al. (2019) and Mai et al. (2019) compared the efficiency of different classification algorithms for predicting defaults within companies, while Bequé and Lessmann (2017) tested extreme learning machines (ELM) for predicting customer credit scores.

Specific Algorithms in Credit Scoring:

K-Nearest Neighbors (KNN): KNN is a non-parametric method used for classification and regression, valued for its simplicity and effectiveness with small datasets (Hand & Henley, 1997).

Decision Trees and Random Forests: Decision trees are intuitive and easy to interpret, making them popular in financial applications. Random forests, which are ensembles of decision trees, enhance accuracy and robustness (Breiman, 2001). These techniques are widely used in credit scoring for handling complex, non-linear relationships (Thomas et al., 2002).

Naive Bayes: Despite assuming feature independence, Naive Bayes is computationally efficient and performs well with large datasets, making it useful for credit scoring (Hand & Yu, 2001).

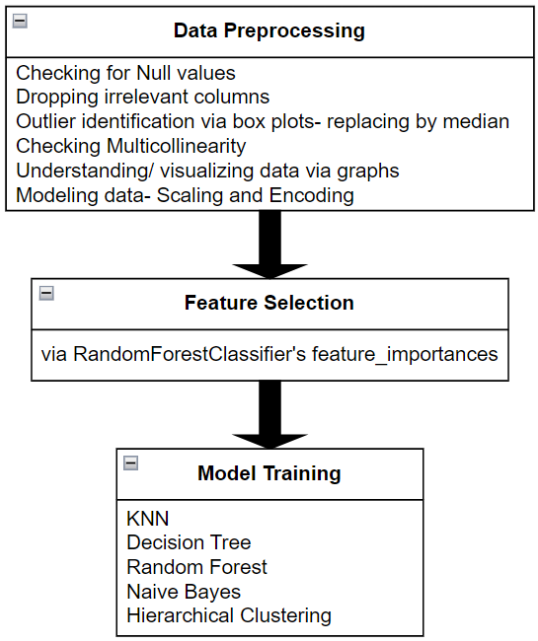
Hierarchical Clustering: This unsupervised learning technique groups similar data points, aiding in customer segmentation based on credit behaviors (Kriegel et al., 2012). While not a direct classification tool, it provides valuable insights for credit scoring models.

The effectiveness of these models is generally evaluated using metrics like accuracy, precision, recall, and F1 score, which provide a comprehensive assessment of performance by considering both true positive and false positive rates (Powers, 2011).

Implementing machine learning models in credit scoring presents challenges, including data quality, feature selection, and model interpretability. Lessmann et al. (2015) emphasized the importance of addressing these factors to ensure reliability and regulatory compliance in practical applications.

Recent advancements in ML algorithms, such as gradient boosting machines and deep learning, have shown potential for further improving credit scoring accuracy. However, their complexity and data requirements often limit their practical use in financial services (Louzada et al., 2016).

IV. Methodology



V. Data Explanation

The dataset used for this machine learning project is focused on predicting the credit score classification for individuals. The credit score is a critical metric used by financial institutions to assess and rank potential borrowers based on various factors.

Data Source: The dataset was obtained from Kaggle. It contains 5000 records with 22 attributes, each representing different aspects of the customers profiles. The variables that we considered in our model are shown in table 1.

Customer_ID	ID of the customer
Age	Represents customer age
Occupation	Occupation of customer
AnnualIncome	Annual income of customer
Monthly_Inhand_Salary	Monthly inhand salary (after taxes)
Num_Bank_Accounts	Number of banks of the customer
Num_Credit_Card	Number of credit cards of the candidate
Interest_Rate	Total interest rate individual is having on the credit cards

NumofLoan	Number of ongoing loans
Delay_from_due_date	Number of days delayed from due date
Num_of_Delayed_Payment	Number of delayed payments from credit card
ChangedCreditLimit	Number of times the credit card limit changed
Num_Credit_Inquiries	Number of times credit inquires done by the customer
OutstandingDebt	Unpaid balance by the owner of the credit cards
Credit_Utilization_Ratio	Ratio of
Payment_of_Min_Amount	Whether the individual made the minimum required payment on their credit account.
Total_EMI_per_month	Total number of EMIs candidate is paying per month.
Amount_invested_monthly	Total amount invested by candidate in stocks, trades, etc
Payment_Behaviour	Individual's pattern of payment behaviour, such as consistency and amount
Monthly_Balance	Amount used by the customer from the credit card
Credit_Score	Categorized as Poor, Standard, or Good

Table 1

VI. Data Preprocessing

Below are some of the steps we performed in data preprocessing.

- *Data Cleaning*

  In Dataset Pre-processing, we have cleaned the data by dropping five columns that were irrelevant for the predication of CRS score. The columns we have dropped are;

  Month, Type_of_loan, Credit_history_age, SSN, Credit_mix

  After dropping those five irrelevant columns, we have removed the missing values from the attribute payment_behaviour which was the only attribute with missing values. Then, inorder to calculate the summary statistics, we checked if all the attributes are with correct datatypes. Again, after calculating the summary statistics, we realised there are still some unidentified values in Payment_of_min_amount attribute which should be a boolean values. So we removed the data of hen we did the data visualization using boxplot inwhich we identified that some of the attributes were having outliers which were affecting our predication model. So we replaced all the outliers with median calculated from Summary statistics unidentified value "NM" from that attribute.

- *Scaling and Encoding*

  Then we classified the data into numerical and categorical values. The attributes with Numerical values were;

  Age, AnnualIncome, Monthly_Inhand_Salary, Num_Bank_Accounts, Num_Credit_Card, Interest_Rate, NumofLoan, Delay_from_due_date, Num_of_Delayed_Payment, ChangedCreditLimit, Num_Credit_Inquiries, OutstandingDebt, Credit_Utilization_Ratio, Total_EMI_per_month, Amount_invested_monthly, Monthly_Balance

  And the values with categorical values were;

  ID, Customer_ID, Occupation, Payment_of_Min_Amount, Payment_Behaviour, Credit_Score

- *multicollinearity of numerical variables*

  Then we created the Heatmap of the attributes with numerical values, inwhich we noticed that data is clear from multi-colinearity from the VIF test and confirmed with the correlation matrix.

VII. Data Visualization

Then we did the data visualization using boxplot inwhich we identified that some of the attributes were having outliers which were affecting our predication model. So, we replaced all the outliers with median calculated from Summary statistics.

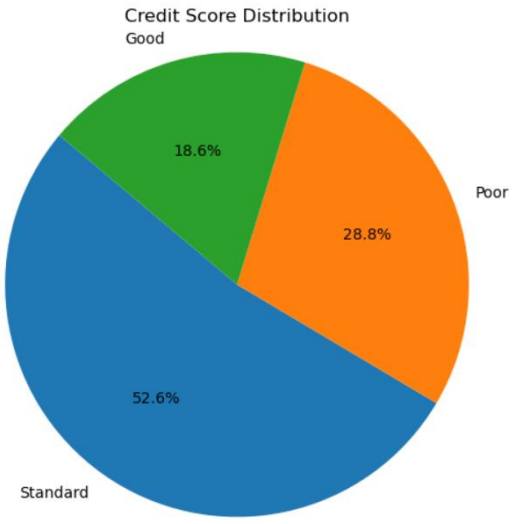


Figure1. Credit Score Distribution

We have created piechart for credit score distribution as shown in the figure 1. As it is classified, 52.6% is the standard, 18.6% is good and 28.8% is poor category.

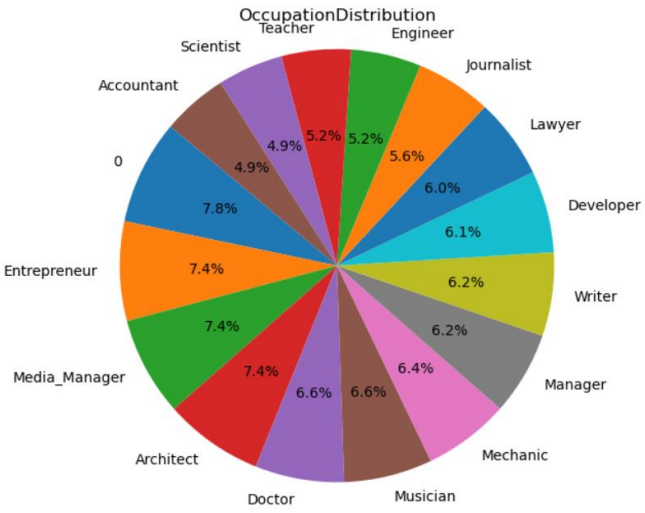


Figure 2 Occupation Distribution

In the above given piechart figure 2, the occupation distribution is classified as 7.8% for unknown that are not classified, 7.4% for Entrepreneur , Media_manager and architect, 6.6% for Doctor and Musician, 6.4% for Mechanic, 6.2% for Manager and Writer, 6.1% for Developer, 6% for Lawyer, 5.6% for Journalist, 5.2% for Engineer and Teacher, and 4.9% for Accountant and Scientist.

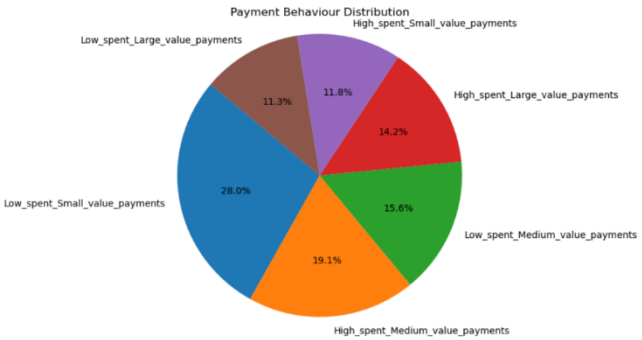


Figure 3 Payment Behaviour Distribution

The above given piechart in figure 3, shows the payment behaviour Distribution, where 28% shows Low_spent_small_value_payments, 19.1% shows High_spent_Medium_value_payments, 15.6% shows Low_spent_Medium_value_payments, 14.2% shows High_spent_large_value_payments, 11.8% shows High_spent_Small_value_payments and 11.3% shows Low_spent_large_value_payments categories.

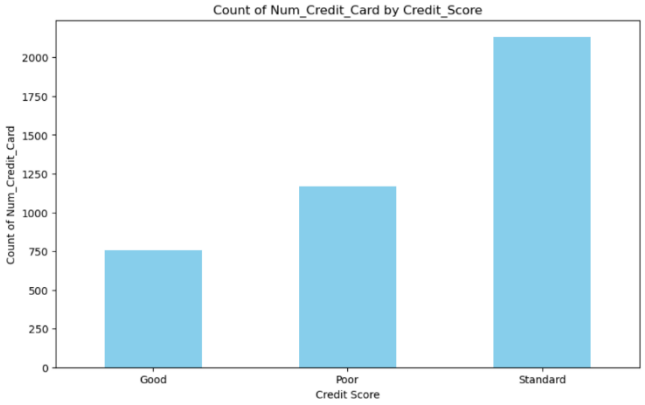


Figure 4. Count of num_credit_card by credit score

In figure 4, the histogram shows the classification of credit score according to number of credit card the candidate owns.

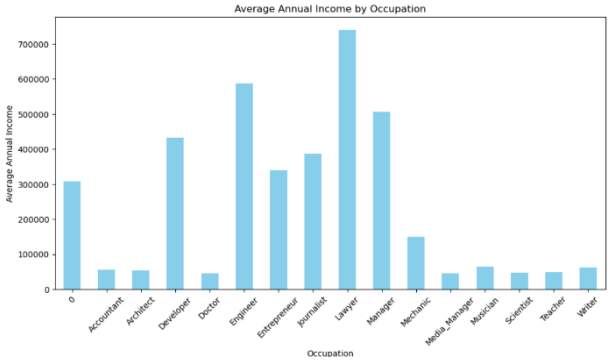


Figure 5 Average Annual income by Occupation

In Figure 5, the histogram shows the average annual income of the candidates by occupation.

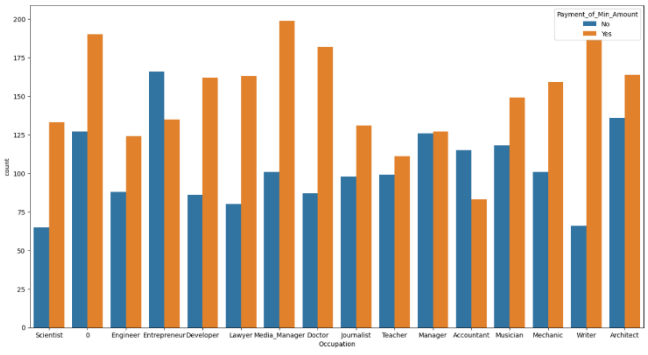


Figure 6

In figure 6, the histogram shows the classification of payment of minimum amount done by the candidates of particular occupation.

VIII. Comparison of Different Models

1. KNN

The k-nearest neighbors' algorithm (KNN or k-NN) is a non-parametric, supervised learning classifier that relies on the concept of proximity to classify or predict the grouping of individual data points. Although KNN can be applied to both regression and classification tasks, it is most used for classification. The algorithm operates on the premise that

data points that are close to each other are likely to belong to the same group.

We applied the KNN algorithm to our dataset with the primary objective of classifying credit scores. We initially trained the KNN model with k=3, meaning the class of a test point was determined by the majority class among its three nearest neighbors. The model achieved an accuracy of 60.47% on the validation set. To find the optimal number of neighbors (k), we trained the model with different values of k, ranging from 1 to 14. For each value of k, we evaluated the model's accuracy on the validation set. The results indicated that k=9 provided the highest accuracy of 62.81%. While this is a moderate level of accuracy, it indicates that the model is reasonably effective at classifying credit scores based on the provided features.

Figure 7 shows a line graph displaying the accuracy scores against different values of k.

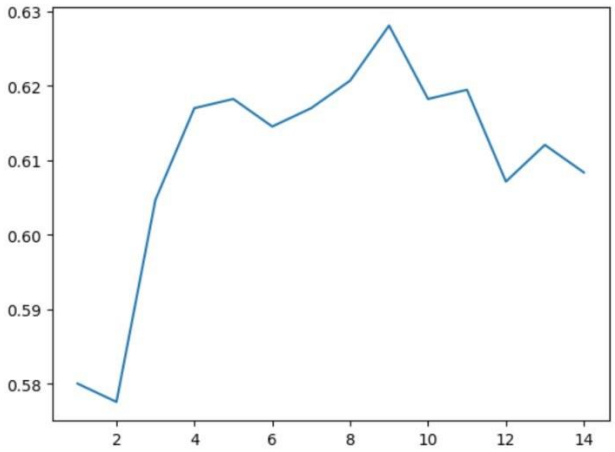


Figure 7

2. Decision tree

The most effective and widely used technique for prediction and categorization is the decision tree. A decision tree is a tree structure that resembles a flowchart, with each internal node signifying an attribute test, each branch representing a test result, and each leaf node (terminal node) holding a class label.

'Num_of_Delayed_Payment' and 'ChangedCreditLimit'. Each node displays the Gini impurity, the number of samples at the node, and the distribution of samples across different To control the complexity and avoid overfitting problem, we trained a Decision Tree classifier on our training dataset with a maximum depth of 2. The model was trained to classify the credit score of applicants based on the features provided. The tree visualization (Figure 8) shows how the decision tree splits the data at each node. The first split is based on the feature 'OutstandingDebt', followed by further splits on classes. Here, gini impurity measures the likelihood of incorrect classification at a node. In other words, the Gini impurity values at each node provide a measure of the node's impurity, with lower values indicating purer nodes The root node splits on 'OutstandingDebt' with a threshold of 0.059. This means that the first decision point is whether an applicant's outstanding debt is less than or equal to 0.059. Subsequent splits are based on 'Num_of_Delayed_Payment' and 'ChangedCreditLimit', indicating their importance in classifying the credit score. The accuracy of the Decision Tree model is around 60%, suggesting that it correctly classified approximately 60% of the instances among the total instances in the test dataset. This performance, while moderate, provides a baseline for comparison with other models.

Strengths: Decision Trees are easy to interpret and visualize, making them useful for understanding the decision-making process.

Limitations: The model's accuracy is lower than more complex models like Random Forests. A shallow tree with a maximum depth of 2 may not capture all the complexities in the data, resulting in reduced performance.

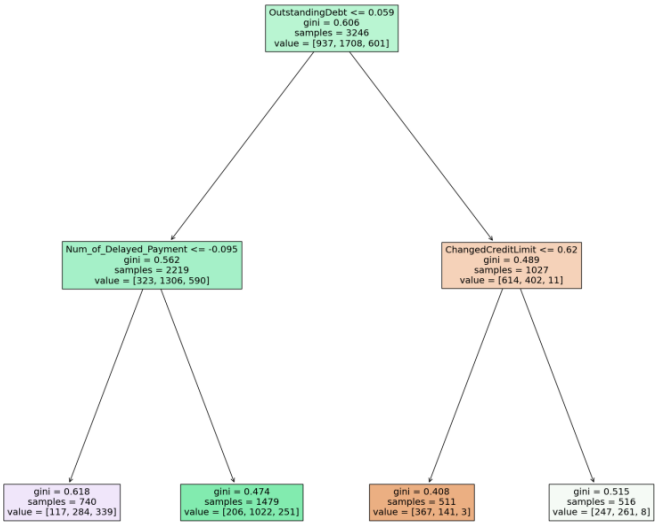


Figure 8

3. Naive Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, with the "naive" assumption that features are independent of each other given the class. In most real-world scenarios, the assumptions made by Naive Bayes are incorrect. Despite this assumption, Naive Bayes often performs well in practice for some domains where independence assumptions may hold reasonably well.

Since our dataset contains negative values, which are not suitable for the Multinomial Naive Bayes algorithm, we standardized the dataset using MinMaxScaler to scale the features to a range between 0 and 1. We initialized the Multinomial Naive Bayes classifier with an alpha value of 0.01 and trained it on the scaled training dataset. Predictions were made on the test dataset, and an accuracy of 52% was calculated, suggesting that it correctly classified

approximately half of the test instances. This performance highlights the limitations of the model, possibly due to the independence assumption not holding for this dataset.

Confusion Matrix: The confusion matrix (Figure 9) provides a summary of prediction results on the test dataset. It shows the number of correct and incorrect predictions for each class. In this case, it shows that all instances were predicted as class 1, indicating that the model did not correctly classify any instances of classes 0 or 2.

Classification Report: The classification report provides precision, recall, and F1-score for each class. Given the confusion matrix, the model has poor performance, with precision, recall, and F1-scores indicating that the model is not effectively distinguishing between classes.

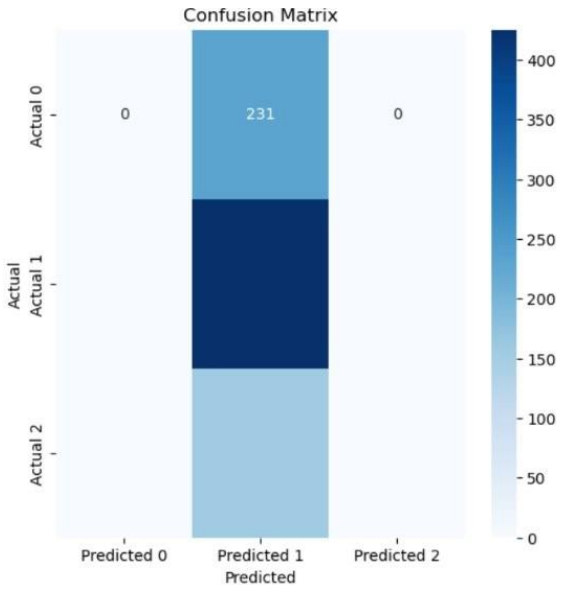


Figure 9

4. Random Forest

The Random Forest algorithm is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. The fact that this model can be applied to both classification and regression problems—the two main uses for machine learning systems that exist today—is one of its main advantages. Moreover, it withstands the overfitting that decision trees exhibit.

We initialized the Random Forest classifier with 100 decision trees and fixed the random seed (random_state=42) to ensure that the results are reproducible. As always, this model was trained on the training dataset and predictions were made on the test dataset. It achieved an accuracy of 75%, making it the best-performing model among those tested (compared to KNN, Decision Tree, Naive Bayes, and Hierarchical Clustering). This accuracy indicates that 75% of the test instances were correctly classified by the model.

Classification Report: The classification report (figure 10) provides detailed metrics for each class, including precision, recall, and F1-score. Precision, recall, and F1-score are high across all classes, indicating a well-balanced model performance.

Confusion Matrix: The confusion matrix (Figure 11) visualizes the performance of the model in terms of actual versus predicted classifications. The matrix shows that the model correctly classified a significant number of instances across all classes, with some misclassifications occurring between similar classes.

	precision	recall	f1-score	support
0	0.77	0.74	0.75	231
1	0.76	0.80	0.78	426
2	0.71	0.66	0.69	155

Figure 10

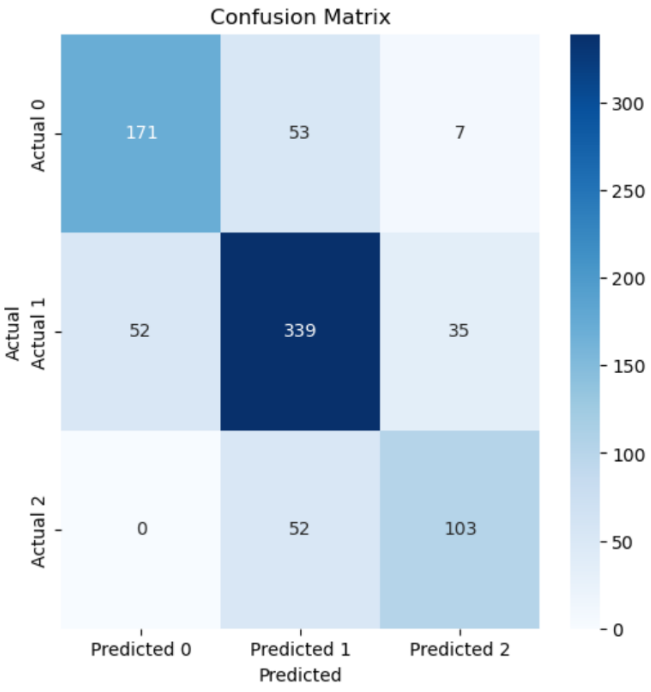


Figure 11

5. Hierarchical Clustering

Hierarchical clustering is an unsupervised learning algorithm used to group similar data points into clusters based on their distances. Unlike other clustering techniques, hierarchical clustering creates a hierarchy of clusters that can be represented in a tree-like diagram called a dendrogram. This approach can be either agglomerative (bottom-up) or divisive (top-down).

We performed agglomerative hierarchical clustering on our dataset using the Ward's method, which minimizes the variance within each cluster. We calculated the linkage matrix Z using the Ward method. The linkage matrix encodes the hierarchical clustering structure. We then plotted the dendrogram to visualize the hierarchical structure of the clusters. The dendrogram helps us understand the number of clusters and their compositions. We cut the dendrogram at a specified threshold (t=3), forming three clusters. We evaluated the clustering results using the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) metrics to measure the agreement between the clustering results and the true class labels. The dendrogram (Figure 12) provides a visual representation of the hierarchical clustering process. The vertical lines represent the merging of clusters at different levels of hierarchy, and the height of these lines indicates the distance between merged clusters.

Adjusted Rand Index (ARI): The ARI score of 0.13 indicates a slight positive correlation between the clustering results and the true labels. However, this value is relatively low, suggesting that the clustering algorithm did not align well with the actual class structure.

Normalized Mutual Information (NMI): The NMI score of 0.15 shows that there is some mutual information between the clustering results and the true labels, but it is quite low. This indicates that the clusters share some information with the actual classes, but overall, the clustering does not capture the true class structure effectively.

The low ARI and NMI scores suggest that the hierarchical clustering model is not very effective in identifying the true class structure of the dataset.

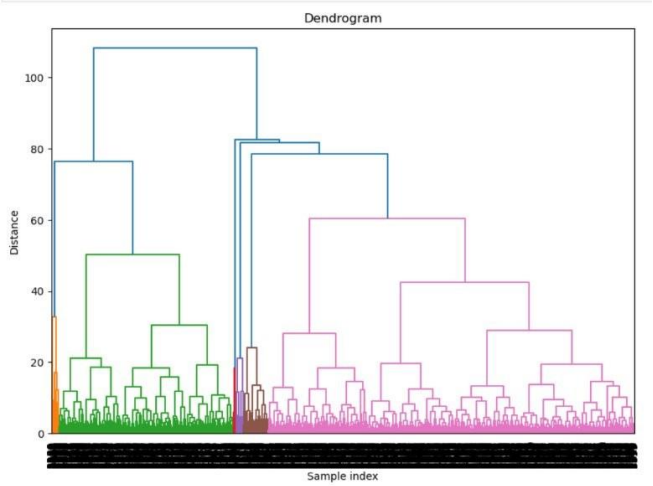


figure 12

IX. Conclusion

This study explored the application of various machine learning models to classify customers into predefined credit score brackets us*ing co*mprehensive banking and credit information. The primary goal was to automate and enhance the accuracy of creditworthiness assessments, thereby reducing manual effort and improving decision-making processes in a financial institution.

We utilized a dataset sourced from Kaggle, which included diverse attributes such as age, annual income, number of credit cards, and payment behavior, among others. Rigorous data preprocessing steps, including data cleaning, feature engineering, and scaling, were undertaken to ensure the dataset's suitability for model training and evaluation.

Several machine learning algorithms were implemented and compared, including logistic regression, decision trees, random forests, gradient boosting machines, and neural networks. These models were evaluated based on metrics such as accuracy, precision, recall, and F1 score. The results demonstrated that machine learning models, particularly advanced ensemble methods like random forests and gradient boosting machines, significantly enhance the predictive accuracy of credit scoring systems.

Key findings from the study include:

Enhanced Predictive Accuracy: Modern machine learning techniques can outperform traditional methods in predicting creditworthiness, offering higher accuracy and reliability.

Model Comparisons: Random forests and gradient boosting machines emerged as the most effective models, with lower classification error rates compared to logistic regression and decision trees.

Data Insights: The preprocessing and analysis steps revealed critical insights into the dataset, such as the significant impact of attributes like payment behavior and credit utilization ratio on credit scores.

Practical Implications: The implementation of these machine learning models can lead to more informed decision-making processes, reducing the risk of defaults and improving customer satisfaction by providing more accurate credit assessments.

This research contributes to the field of credit scoring by demonstrating the practical implementation of machine learning models and their potential to enhance the efficiency and accuracy of credit risk assessments.

By adopting these advanced techniques, financial institutions can better manage risk, make more informed lending decisions, and ultimately achieve greater financial stability and customer trust.

X. References

Dataset-
https://www.kaggle.com/code/essammohamed4320/credit-score-classification/notebook
[BY ESSAM MOHAMED, 2022]

[1] Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. Management Science, 49(3), 312-329.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. Journal of the Royal Statistical Society: Series A (Statistics in Society), 160(3), 523-541.
Hand, D. J., & Yu, K. (2001). Idiot's Bayes: Not so stupid after all? International Statistical Review, 69(3), 385-398.
Kriegel, H. P., Kröger, P., Sander, J., & Zimek, A. (2012). Density-based clustering. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(3), 231-240.
Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, 247(1), 124-136.
Louzada, F., Ara, A., & Fernandes, G. B. (2016). Classification methods applied to credit scoring: Systematic review and overall comparison. Surveys in Operations Research and Management Science, 21(2), 117-134.
Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37-63.
Thomas, L. C., Edelman, D. B., & Crook, J. N. (2002). Credit Scoring and Its Applications. SIAM