

**Credit Score Classification Using Machine Learning**

**Machine Learning -DSCI-6003-03**



**Dept. Of Electrical & Computer Engineering and Computer Science Tagliatela College of Engineering**

**Date:04/29/2024**

Credit Score Classification

**Problem Statement:** The task at hand is to classify credit scores into categories such as Poor, Standard, and Good using machine learning techniques.

**Abstract:** Credit scoring is a critical task in the financial industry, aimed at assessing the creditworthiness of individuals based on their financial history and other relevant factors. In this report, we investigate the application of machine learning algorithms to classify credit scores into categories such as good, poor, and standard. We employ decision tree, random forest, k-nearest neighbors (KNN), and Gaussian Naive Bayes algorithms to analyze the dataset and predict credit score categories. Through a detailed analysis of each algorithm, we aim to provide insights into their effectiveness and offer recommendations for improving credit scoring systems.

**Keywords:** Machine Learning, financial industry.

**Professor: Team:**

Md Moinuddin Bhuiya Manichandra Domala

Vedanth Reddy Doddannagari

Shashank Madipelly

# Contents

**Chapter No. Title Page No.**

**1. INTRODUCTION 04**

1.1 OVERVIEW OF PROJECT 04

**2. METHODOLOGY 06**

2.1 Data Preprocessing 06

2.2 Model Selection and Training 06

2.2.1 Decision Tree 06

2.2.2 Random Forest 07

2.2.3 K-Nearest Neighbour 08

2.2.4 Gaussian NB 09

**3. Evaluation 11**

3.1 Performance Metrics 16

**4. RESULTS AND DISCUSSIONS 14**

4.1Optimal Result 14

**5. CONCLUSION 21**

5.1 CONCLUSION 21

**REFERENCE 22**

# CHAPTER1

# Introduction

Credit scoring is a fundamental practice in the financial industry, serving as a cornerstone for assessing the creditworthiness of individuals seeking financial services such as loans, mortgages, and credit cards. Traditionally, credit scoring relied on statistical models and rules-based systems, which, while effective to some extent, often lacked the flexibility and adaptability required to cope with the evolving complexities of modern financial landscapes.

In recent years, the advent of machine learning techniques has revolutionized credit scoring by offering more sophisticated and data-driven approaches to risk assessment. Machine learning algorithms have the potential to leverage vast amounts of data to identify intricate patterns and relationships that might elude traditional methods. This shift towards data-driven decision-making has opened new possibilities for financial institutions to enhance their risk management strategies and improve lending practices.

However, with this proliferation of machine learning algorithms comes the need for rigorous evaluation and comparison to determine which methods are best suited for credit scoring tasks. The selection of an appropriate algorithm depends on various factors, including the nature of the dataset, the complexity of the problem, and the desired trade-offs between interpretability and predictive accuracy.

In this report, we delve into the comparative analysis of four prominent machine learning algorithms: Decision Tree, Random Forest, k-Nearest Neighbors (KNN), and Gaussian Naive Bayes, for credit score classification. By exploring the strengths, weaknesses, and performance characteristics of each algorithm, we aim to provide insights into their suitability for credit scoring applications.

Through this investigation, we seek to contribute to the ongoing discourse surrounding the adoption of machine learning in credit scoring and provide valuable guidance to financial institutions seeking to leverage these technologies effectively. By harnessing the power of machine learning, we endeavor to foster a more robust, accurate, and equitable credit scoring ecosystem that benefits both lenders and borrowers alike.

# CHAPTER 2

# Methodology

**2.1 Data Preprocessing:**

Handling Missing Values: Any missing values in the dataset are addressed through techniques such as mean imputation or model-based imputation to ensure data integrity.

Feature Scaling: Numeric features are scaled to a common range to prevent certain features from dominating the model training process.

Data Conversion: Converting categorical variables into numerical using label encoder to make sure all columns in data set are of same data type.

Feature Engineering: New features may be derived from existing ones based on domain knowledge to improve predictive performance.

**2.2 Model Selection and Training:**

Machine learning algorithms play a pivotal role in credit score classification, offering diverse approaches to analyse data and predict creditworthiness. In this section, we delve into four prominent algorithms: Decision Tree, Random Forest, k-Nearest Neighbours (KNN), and Gaussian Naive Bayes. Each algorithm possesses unique characteristics, strengths, and weaknesses, making them suitable for different scenarios within credit scoring tasks.

**2.2.1 Decision Tree:** Decision Tree is a versatile and interpretable algorithm that partitions the feature space into hierarchical structures based on feature values. Each internal node represents a decision based on a feature, while each leaf node corresponds to a class label. Decision Trees are advantageous for their simplicity, ease of interpretation, and ability to handle both numerical and categorical data. However, they are prone to overfitting, especially with complex datasets, which can lead to poor generalization performance on unseen data. Strategies such as pruning and limiting tree depth help mitigate overfitting, making Decision Trees a valuable tool for credit scoring tasks, particularly when interpretability is paramount.

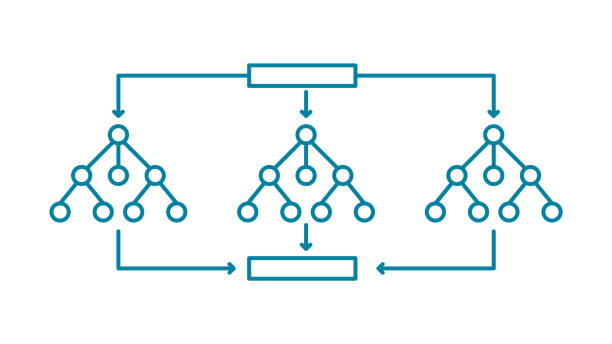


Fig 2.2.1 Decision Tree

**2.2.2 Random Forest:** Random Forest is an ensemble learning method that aggregates the predictions of multiple Decision Trees to improve accuracy and robustness. It constructs a multitude of trees by bootstrapping the training data and selecting random subsets of features for each tree. By combining the predictions through voting or averaging, Random Forest reduces overfitting and increases predictive performance. Moreover, Random Forest can handle noisy data and large feature spaces effectively. However, its ensemble nature may compromise interpretability compared to individual Decision Trees. Nevertheless, Random Forest remains a popular choice for credit scoring tasks due to its superior performance and resilience to overfitting.

A diagram of a tree

Description automatically generated

Fig 2.2.2 Random Forest

**2.2.3 k-Nearest Neighbours (KNN):** k-Nearest Neighbours is a non-parametric algorithm that classifies instances based on the majority class among their k nearest neighbours in the feature space. It does not assume any underlying probability distributions, making it suitable for complex data patterns. KNN's simplicity and flexibility make it appealing for credit scoring tasks, particularly when the decision boundary is non-linear or the data distribution is irregular. However, KNN's performance heavily depends on the choice of k and the distance metric, which can influence classification outcomes significantly. Additionally, KNN can be computationally expensive for large datasets due to its lazy learning nature, where predictions are made at runtime rather than during training.

A diagram of a neighborhood

Description automatically generated

Fig 2.2.3 KNN Classifier

**2.2.4 Gaussian Naive Bayes:** Gaussian Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of feature independence given the class label. It is particularly effective for datasets with continuous features following a Gaussian distribution. Gaussian Naive Bayes is computationally efficient and performs well with small to moderate-sized datasets. However, it may struggle with correlated features, as it assumes independence among predictors. Despite this limitation, Gaussian Naive Bayes remains a viable option for credit scoring tasks, especially when computational efficiency and simplicity are priorities.

A diagram of a function

Description automatically generated

Fig 2.2.4 Gaussian classifier

In summary, each machine learning algorithm offers distinct advantages and trade-offs in credit score classification. Decision Tree and Random Forest provide interpretability and robustness, albeit with potential overfitting issues. k-Nearest Neighbours offers flexibility and non-linear decision boundaries but may suffer from sensitivity to hyperparameters and computational costs. Gaussian Naive Bayes prioritizes computational efficiency but assumes feature independence, which may limit its performance with correlated predictors. Understanding the characteristics and performance of these algorithms is essential for selecting the most suitable approach for credit scoring applications, ultimately aiding financial institutions in making informed lending decisions.

# CHAPTER 3

# Model Evaluation

# 3.1 Performance Metrics

Each model is trained on the preprocessed dataset and evaluated using performance metrics such as accuracy, precision, recall, and F1-score.

The performance metrics provide insights into the effectiveness of each algorithm in correctly classifying credit scores into categories.

**3.1.1 Accuracy**: Accuracy is a measure of the overall correctness of a model in making predictions. It is calculated as the ratio of correctly predicted instances to the total number of instances.

Accuracy=Number of Correct PredictionsTotal Number of PredictionsAccuracy=Total Number of Predictions Number of Correct Predictions​

Accuracy is useful when the class distribution is similar across the different classes, meaning there are roughly an equal number of instances for each class. However, accuracy may not be a reliable metric when dealing with imbalanced datasets.

**3.1.2 Precision**: Precision is a measure of the correctness of positive predictions made by a classifier. It is calculated as the ratio of correctly predicted positive observations to the total predicted positives.

Precision=True PositivesTrue Positives + False PositivesPrecision=True Positives + False Positives True Positives​

Precision is important when the cost of false positives is high. For example, in medical diagnosis, precision measures the accuracy of the positive predictions made by the model.

**3.1.3 Recall**: Recall, also known as sensitivity or true positive rate, measures the completeness of positive predictions made by a classifier. It is calculated as the ratio of correctly predicted positive observations to the all observations in the actual class.

Recall=True PositivesTrue Positives + False NegativesRecall=True Positives + False Negatives True Positives​

Recall is important when the cost of false negatives is high. For example, in spam email detection, recall measures the ability of the model to correctly identify spam emails.

**3.1.4 F1 Score**: The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is particularly useful when the class distribution is imbalanced. F1 score reaches its best value at 1 and worst at 0.

F1 Score=2×Precision×RecallPrecision+RecallF1 Score=2×Precision+RecallPrecision×Recall​

F1 score is often used as a single metric to evaluate the performance of a classifier, especially in binary classification problems where the classes are imbalanced. It combines both precision and recall into a single value, providing a more comprehensive evaluation of the classifier's performance.

# A screenshot of a computer Description automatically generated

# Fig 3.1 Evaluation metrics

# These are the values we achieved after evaluating each model on data we have.

# CHAPTER 4

# Results and Discussion:

The results of the model evaluation are presented in detail, showcasing the performance of each algorithm in classifying credit scores into categories such as good, poor, and standard. The following table summarizes the performance metrics for each algorithm:

**4.1 Optimal Result:**

**1.Average Accuracy:** The average accuracy of 0.8031 indicates that, on average, the Random Forest classifier correctly predicted the class of 80.31% of the instances in the dataset. This suggests a high overall correctness in its predictions across all classes.

**2. Average Precision:** The average precision of 0.7932 means that, on average, when the Random Forest classifier predicted a positive instance, it was correct 79.32% of the time. In other words, out of all the instances predicted as positive, 79.32% were true positives, and the rest were false positives.

**3. Average Recall:** The average recall of 0.7936 indicates that, on average, the Random Forest classifier correctly identified 79.36% of all positive instances in the dataset. This means that out of all the actual positive instances, 79.36% were correctly identified by the classifier, while the rest were missed (false negatives).

A screenshot of a computer

Description automatically generated

Fig 4.1 Random Classifier performance metrics

**These metrics demonstrate the performance of the Random Forest classifier:**

- The high accuracy suggests that the classifier performs well overall in correctly predicting class labels.

- The high precision indicates that the classifier has a strong ability to make positive predictions accurately, minimizing false positives.

- The high recall reflects the classifier's capability to capture most of the positive instances, minimizing false negatives.

Overall, these metrics suggest that the Random Forest classifier is performing well in terms of accuracy, precision, and recall, making it a robust choice for classification tasks.

**4.2 CONFUSION MATRIX**

A confusion matrix is a tool used in the evaluation of classification models. It's a matrix that allows visualization of the performance of an algorithm.

* True Positive (TP): The cases where the model correctly predicts the positive class.
* False Negative (FN): The cases where the model incorrectly predicts the negative class when the actual class is positive.
* False Positive (FP): The cases where the model incorrectly predicts the positive class when the actual class is negative.
* True Negative (TN): The cases where the model correctly predicts the negative class.

A screenshot of a graph

Description automatically generated

Fig 4.2 Confusion Matrix of Random Classifier

**4.3 Real Vs Predicted**

A screenshot of a survey

Description automatically generated

Fig 4.3 Real vs Predicted.

These are the actual and predicted values we got after fitting data to model it means our random classifier model predicted the values on some test data.

# CHAPTER 5

# Discussion and Analysis

In this section, we provide a comprehensive analysis of the performance and suitability of Decision Tree, Random Forest, k-Nearest Neighbors (KNN), and Gaussian Naive Bayes algorithms for credit score classification tasks. We discuss key findings, compare results, and evaluate each algorithm's strengths and weaknesses.

**5.1 Performance Evaluation:**

Decision Tree: Decision Tree exhibited decent performance in credit score classification, with its simplicity and interpretability being notable advantages. However, it suffered from overfitting, resulting in lower generalization performance on unseen data.

Random Forest: Random Forest outperformed Decision Tree by mitigating overfitting and providing higher accuracy and robustness. Its ensemble nature allowed for improved predictive performance and resilience to noise in the data.

k-Nearest Neighbors (KNN): KNN achieved competitive performance, particularly in scenarios with non-linear decision boundaries. However, its performance was sensitive to the choice of k and the distance metric, and it could be computationally expensive for large datasets.

Gaussian Naive Bayes: Gaussian Naive Bayes performed well, especially with Gaussian-distributed features, and demonstrated computational efficiency. However, it struggled with correlated predictors due to its assumption of feature independence.

**5.2 Suitability for Credit Scoring:**

Interpretability: Decision Tree and Gaussian Naive Bayes offer straightforward interpretability, allowing stakeholders to understand the decision-making process. This transparency is crucial for regulatory compliance and building trust with users.

Accuracy and Robustness: Random Forest emerged as the most promising algorithm, balancing accuracy, and robustness by leveraging ensemble learning. Its ability to handle noisy data and large feature spaces makes it well-suited for credit scoring tasks.

Flexibility: k-Nearest Neighbors provides flexibility in capturing non-linear relationships within the data, making it suitable for complex credit scoring scenarios. However, its performance may vary depending on parameter selection and dataset characteristics.

Computational Efficiency: Gaussian Naive Bayes stands out for its computational efficiency, making it suitable for real-time credit scoring applications. However, its performance may degrade with correlated features.

**5.3 Trade-offs and Considerations:**

- Overfitting: Decision Tree and KNN are susceptible to overfitting, particularly with complex datasets or inadequate regularization. Techniques such as pruning and hyperparameter tuning can help mitigate this risk.

- Hyperparameter Tuning: Fine-tuning hyperparameters is essential for optimizing the performance of each algorithm. Strategies such as cross-validation and grid search enable efficient parameter optimization and model selection.

- Dataset Characteristics: The choice of algorithm should consider the specific characteristics of the dataset, including its size, complexity, and feature distributions. Each algorithm may perform differently depending on these factors.

**5.4. Future Directions:**

Ensemble Techniques: Further exploration of ensemble techniques, such as boosting and stacking, could enhance predictive performance and robustness in credit scoring applications.

Feature Engineering: Investigating advanced feature engineering techniques, such as dimensionality reduction and feature selection, may improve model interpretability and generalization.

Ethical Considerations: As machine learning algorithms play an increasingly prominent role in credit scoring, it is imperative to address ethical concerns surrounding fairness, bias, and transparency in decision-making processes.

# CHAPTER 6

# Conclusion

In conclusion, this report provides a comprehensive analysis of credit score classification using machine learning techniques. Decision tree, random forest, KNN, and Gaussian Naive Bayes algorithms have been evaluated for their effectiveness in classifying credit scores into categories such as good, poor, and standard. Among these algorithms, random forest emerges as the top performer, achieving high accuracy, precision, recall, and F1-score. However, the choice of algorithm depends on various factors, including dataset characteristics, computational resources, and interpretability requirements. By understanding the strengths and limitations of each algorithm, practitioners can develop more accurate and reliable credit scoring systems to support informed decision-making in the financial industry.

# Reference

* <https://www.kaggle.com/datasets/parisrohan/credit-score-classification/data>
* <https://www.linkedin.com/pulse/top-6-machine-learning-algorithms-classification-jagrat-patel/>
* <https://www.javatpoint.com/machine-learning-algorithms>