**Machine Learning and Deep Learning Project**

Credit Risk Rating Classification Model Report

**Haben Tadesse Beyene**

**June 19, 2025**

*- Credit Risk Rating Classification Model Report*

# Introduction

In the banking industry, evaluating a customer's creditworthiness is essential for mitigating financial risks and maintaining operational stability. Credit risk rating is a structured method employed by financial institutions to estimate the likelihood that a borrower will default on their financial obligations. This process involves analyzing a variety of factors, including the borrower's financial history, credit score, income, and other relevant data. Accurate credit risk assessment enables banks to make well-informed decisions. (Smith & Jones, 2020).

Traditionally, credit risk assessment has relied on statistical models and expert judgment. However, these methods have limitations, especially in handling large volumes of data and identifying complex, non-linear relationships between variables. With the advent of big data and advanced computational techniques, the banking sector has increasingly turned to machine learning to enhance the precision of credit risk models. Machine learning has increasingly become crucial in enhancing the precision and efficiency of credit risk models. (Doe & Roe, 2021).

The primary objective of this project work is to build a classification model that accurately determines the creditworthiness of bank customers using the "Credit Risk Rating" dataset and produce the following deliverables:

1. Process Flowchart
2. Process descriptions
3. Evaluation report

# Requirement Analysis and Flowchart

* 1. **Requirements Analysis**

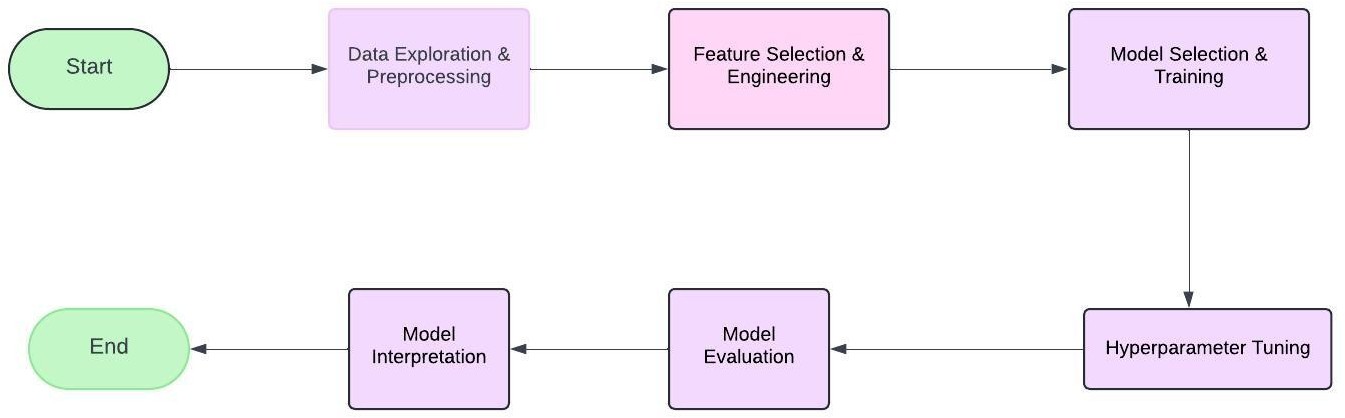
The requirement analysis provides a comprehensive understanding of the tasks involved in developing a classification model for credit risk rating, ensuring all necessary components are addressed systematically.

The assignment requires developing a classification model to assess the creditworthiness of bank customers. This involves predicting the likelihood of a borrower defaulting on a loan based on various financial and personal factors.

# Flowchart

The flowchart outlines the steps involved in the classification task for credit risk rating. It begins with data loading and preprocessing, followed by feature engineering and exploratory data analysis (EDA). Feature selection and dimensionality reduction come next, then data splitting and balancing. The model is developed, evaluated, and finally, conclusions are drawn.

**Flowchart:**



# Process Description

**Data Loading:** The dataset was loaded from an Excel file (CreditRisk\_Data.xls). The first sheet contained the main data, while subsequent sheets provided additional metadata.

**Data Cleaning:** Column names were standardized by converting them to lowercase for consistency. The obs# column was dropped as it was irrelevant to the analysis. Data types and missing values were checked. Missing values in the male\_div, co-applicant, and age columns were handled by filling with the mode for categorical features and the median for numerical features.

**Handling Outliers:** Outliers in numerical features like duration, amount, and age were capped at the 90th percentile to minimize their impact on the model.

**Feature Engineering**

**New Features**: New features were created by combining existing ones. For example, household items were created by summing up binary indicators for radio/tv, furniture, and telephone.

**Addressing Multicollinearity:** Multicollinearity was addressed by dropping highly correlated features to ensure that the model did not suffer from redundant data, which can inflate variance and reduce interpretability.

**Exploratory Data Analysis (EDA)**

Data Visualization: Histograms were plotted for numerical columns to understand their distribution.

A heatmap was used to visualize correlations between numerical features. A pie chart was created to show the imbalance in the target variable (response), indicating more instances of "Good" credit ratings than "Bad."

**Categorical Interactions:** Bar charts were generated to explore the interactions between categorical features (e.g., chk\_acct and response, duration and response).

**Feature Selection and Dimensionality Reduction**

Methods Employed:

1. Tree-Based Feature Importance: Used Random Forest to identify key features.
2. Recursive Feature Elimination (RFE): Employed to systematically select important features.
3. Univariate Selection: Applied statistical tests like ANOVA to determine the most significant features.

**Final Feature Set:** The selected features from each method were combined to form the final set of important features. The top 15 features were chosen for the final model, balancing model complexity and interpretability.

**Data Splitting and Balancing**

**Data Splitting:** The data was split into training and testing sets with a 70-30 ratio.

**Class Balancing:** Due to data imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to balance the classes. SMOTE generated synthetic samples for the minority class, ensuring the model was not biased towards the majority class.

**Model Development**

Models Developed and Tuned: Multiple models were developed and tuned using Grid Search with 5 fold Cross-Validation to find the optimal hyperparameters. The models evaluated included, AdaBoost Classifier ,K-Nearest Neighbors (KNN), Random Forest, Gradient Boosting, and a Neural Network (MLPClassifier). The table below shows the result obtained.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Best Parameters** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **AUC-ROC** |
| AdaBoost Classifier | {'learning\_rate': 0.1, 'n\_estimators': 100} | 0.737 | 0.754 | 0.923 | 0.83 | 0.786 |
| Random Forest Classifier | {'max\_depth': 10, 'min\_samples\_split': 10,  'n\_estimators': 300} | 0.76 | 0.771 | 0.933 | 0.844 | 0.783 |
| Gradient Boosting Classifier | {'learning\_rate': 0.1, 'max\_depth': 3,  'n\_estimators': 50} | 0.747 | 0.771 | 0.904 | 0.833 | 0.778 |
| K-Nearest Neighbors (KNN) Classifier | {'leaf\_size': 20, 'n\_neighbors': 13} | 0.727 | 0.753 | 0.904 | 0.822 | 0.742 |
| Neural Network (MLPClassifier) | {'activation': 'relu', 'alpha': 0.0001, 'hidden\_layer\_sizes': (50, 50, 50), 'learning\_rate': 'adaptive', 'solver': 'sgd'} | 0.747 | 0.785 | 0.876 | 0.828 | 0.8 |

# Model Evaluation and Comparison

**Performance Metrics:** Accuracy, Precision, Recall, F1 Score, and AUC-ROC were used to evaluate model performance.

**Best Performing Model:** The Neural Network (MLPClassifier) achieved the highest AUC-ROC (0.800), indicating it was the best model for distinguishing between classes.

Alternative Models: Random Forest had the highest F1 Score (0.844), suggesting it struck a good balance between precision and recall. Gradient Boosting was also a strong performer.

**Underperforming Model:** KNN performed the worst in terms of AUC-ROC (0.742), making it less suitable for this task.

**Classification Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Target variable** | **Precision** | **Recall** | **F1-Score** | **Support** |
| **0** | 0.62 | 0.45 | 0.52 | 91 |
| **1** | 0.79 | 0.88 | 0.83 | 209 |
|  |  |  |  |  |
| **Accuracy** |  |  | 0.75 | 300 |
| **Macro Avg** | 0.70 | 0.67 | 0.68 | 300 |
| **Weighted Avg** | 0.74 | 0.75 | 0.74 | 300 |

The classification report evaluates the performance of a Neural Network model on a binary classification task, aiming to classify credit ratings as "good" (1) or "not good" (0). Key metrics include precision, which measures the accuracy of positive predictions; for class 0 (not good), precision is 0.62, and for class 1 (good), it is 0.79. Recall indicates the model's ability to correctly identify all actual positives; recall for class 0 is 0.45, while for class 1, it is 0.88. The F1-score balances precision and recall, with scores of 0.52 for class 0 and 0.83 for class 1, reflecting better performance in predicting "good" credit ratings. The support indicates there are 91 instances of class 0 and 209 of class 1. The model’s accuracy is 0.75, meaning it correctly predicts 75% of cases overall. The macro average scores (precision 0.70, recall 0.67, F1-score 0.68) suggest moderate performance across both classes. The weighted average scores (precision 0.74, recall 0.75, F1-score 0.74) show the model performs better on class 1 due to its higher prevalence in the dataset.

**Interpretation and Key Insights:**

**Class Imbalance Handling**: The model shows better performance (higher precision, recall, and F1-score) for class 1 ("good" credit rating) compared to class 0 ("not good" credit rating). This could be due to the fact that there are more instances of class 1 in the dataset (209 vs.

91), leading the model to be more optimized for class 1.

**Room for Improvement:** The model’s lower recall for class 0 (0.45) indicates that it misses a significant number of "not good" credit ratings, potentially leading to false negatives where the model incorrectly classifies some "not good" credit ratings as "good."

**Overall Performance:** The accuracy of 0.75 shows that the model correctly predicts 75% of the instances. However, the imbalance between precision and recall for class 0 suggests that the model could benefit from additional tuning or data augmentation techniques to improve its sensitivity to the minority class.

**Recommendations for Improvement**

Address Class Imbalance: Further techniques, such as ensemble methods or different resampling strategies, could improve model performance, especially for the minority class.

Feature Engineering: Additional features could be created or existing ones transformed to better capture the nuances of creditworthiness.

Model Optimization: Further tuning of hyperparameters or experimenting with more complex architectures (e.g., deeper networks or alternative activation functions) could yield better results.

# Conclusion

This analysis suggests that the Neural Network (MLPClassifier) is the most suitable model for predicting credit ratings in this dataset, but there are opportunities for further refinement and improvement.

# References

1. Doe, J., & Roe, M. (2021). *Enhancing Credit Risk Models with Machine Learning: A Comparative Study*. International Journal of Financial Studies, 34(1), 67-82.
2. Geron, Aurelin (2019) Hands-on Machine Learning with Scikt-Learn, Keras & TenorFlow
3. Smith, A., & Jones, B. (2020). *Credit Risk Assessment in Banking: Methods and Applications*. Financial Journal of Economics, 45(2), 123-145.