Loan Approval Prediction, Interpretability and Credit Risk Segmentation Using XGBoost, SHAP and KMeans

School of Computer Science & Applied Mathematics
University of the Witwatersrand

Abel Letoaba 1579292
Ivy Chepkwony 2431951
Ntokozo Mhlola 3042550
Kgetja Mphekgwane 2593733

Instructor: Prof Clint Van Alten

May 30, 2025

A project submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Bachelor of Science with Honours/Masters

Abstract

This project investigates the application of XGBoost for loan approval prediction and the explainability of loan decisions using a large, synthetic dataset. The dataset, inspired by the original Credit Risk dataset on Kaggle and enriched with variables from Financial Risk for Loan Approval data, contains 45,000 records and 14 variables, including both categorical and continuous features. To ensure sufficient data diversity and balance, SMOTENC was used to simulate additional data points. The workflow encompasses comprehensive data preprocessing, model training, hyperparameter optimization, and interpretability analysis. The XGBoost classifier demonstrated high predictive accuracy, with performance further enhanced by hyperparameter tuning. SHAP values were employed to provide individualized explanations for loan outcomes, highlighting key features and offering actionable advice for applicants. Additionally, KMeans clustering was used to segment applicants into risk categories, supporting a deeper understanding of applicant profiles. The results confirm that XGBoost, combined with interpretability tools, offers a robust and explainable solution for automated credit risk assessment.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Loan approval is a critical function within financial services, playing a key role in determining access to credit and promoting broader financial inclusion. With the increasing adoption of automated decision-making, there is a growing demand for models that not only deliver high predictive accuracy but also offer transparency and interpretability in their decisions. This project addresses these needs by leveraging XGBoost, a powerful ensemble learning technique, for credit risk assessment.

The project methodology includes data acquisition data preprocessing, model training with XGBoost, hyperparameter optimization, and post hoc interpretability using SHAP values. Furthermore, KMeans clustering is applied to segment applicants into distinct risk categories, providing actionable insights for both lenders and applicants. The integration modeling with explainability ensures that the loan approval system remains both accurate and transparent, promoting fair and well-informed lending decisions.

1.1 Project Question and Hypothesis

Project Question: How can XGBoost be utilized to accurately and transparently predict loan eligibility, and what are the most influential factors driving automated loan approval or rejection decisions?

Hypothesis: Applying XGBoost to the task of loan eligibility prediction will achieve high predictive accuracy, and the integration of SHAP-based interpretability methods will yield clear, actionable explanations for automated loan decisions, thereby enhancing transparency and stakeholder trust. KMeans clustering will further segment applicants by risk, offering practical insights to support risk-adjusted loan approvals.

Chapter 2

Methodology

2.1 Workflow

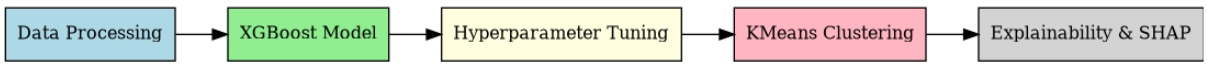

Figure 2.1: Work Flow

Workflow Steps shown in Figure 2.1

2.1.1 Data Processing

- This section covers how raw data is cleaned, transformed, and prepared for use in the machine learning model.

2.1.2 XGBoost Model

- Here, the focus is on training a supervised learning model (XGBoost) using the processed data.

2.1.3 Hyperparameter Tuning

- This part discusses the process of optimizing the XGBoost model by adjusting its hyperparameters to achieve the best performance.

2.1.4 KMeans Clustering

- This section explains how KMeans clustering is used to group data.

2.1.5 Explainability & SHAP

- Finally, this section describes how SHAP values are used to interpret and explain the predictions made by the XGBoost model, providing insights into feature importance.

2.2 Data Preprocessing

The data preprocessing process in the project involves several crucial steps to prepare the dataset for effective analysis and model building. First, the dataset was loaded from a CSV file named loan_data.csv using Pandas. Basic inspection was performed using the info() and head() functions to understand the data structure, identify missing values, and preview sample records. The next step was data cleaning, which included removing duplicate records using the drop_duplicates() method. This helped maintain data quality and ensured that repeated entries do not skew model performance. Following this, categorical variables in the dataset were converted to numerical format using Label Encoding. This was necessary because machine learning models typically require numerical inputs. LabelEncoder was used to map each unique categorical value to an integer. These steps ensured that the dataset is well-organized, consistent, and suitable for input into machine learning algorithms. Additional preprocessing such one-hot encoding was also considered depending on model needs especially when applying the KMeans. Overall, this preprocessing phase played a vital role in setting a strong foundation for accurate and reliable predictions and risk categorization.

Library Imports:

- Numpy, Pandas, Seaborn

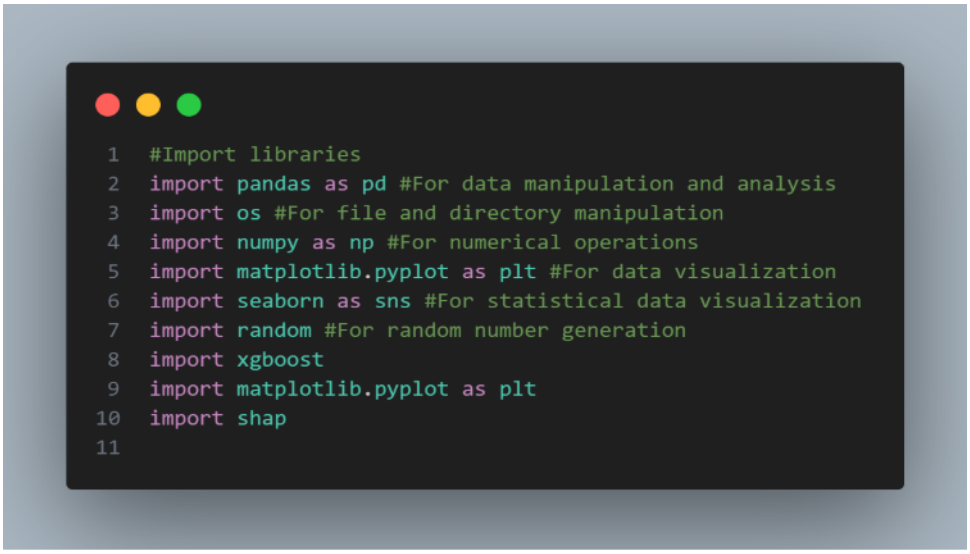

Figure 2.2: Library Imports

Inspection:

- Used df.info() and df.head() to examine column types and values
- 14 Columns and 45000 Entries
- Data types [Float64, Int64, Object]

Duplicates:

- Removed using df.drop_duplicates() to eliminate redundant records
- The were no duplicates, the shape of the data remained at (45000, 14)

Encoding:

- Applied LabelEncoder to convert categorical variables into integers
- Ensures model compatibility with non-numeric data

Data Splitting:

- Divided dataset into training, validation, and test sets using train_test_split()
- Typical split used: 70% training, 15% validation, 15% test

2.3 XGBoost model

For our dataset a Supervised Learning model was chosen specifically XGBoost (Extreme Gradient Boosting) a highly efficient and scalable machine learning algorithm that is particularly well-suited for structured (tabular) data.

Why XGBoost Was Chosen

- **Performance on Tabular Data:** XGBoost is recognized for its exceptional accuracy and efficiency on tabular datasets, which are prevalent in financial and credit risk applications. XGBoost has consistently outperformed models like logistic regression and decision trees in financial applications. For example, Ouyang [2024] found that XGBoost achieved a higher AUC (0.6973) compared to logistic regression (0.5601). Furthermore, unlike deep learning models such as neural networks, which often require large volumes of data and extensive feature engineering to excel on tabular data, XGBoost can natively handle both numerical and categorical variables with minimal pre-processing. See [Shwartz-Ziv and Armon 2022] and [Gorishniy *et al.* 2023].
- **Feature Importance and Interpretability:** XGBoost provides built-in methods for evaluating feature importance. This aligns with the project's requirement to not only predict outcomes but also to interpret which factors most influence loan approval decisions. As reflected in the notebook outputs, features such as previous loan defaults, annual income, loan amount, and previous loan defaults were identified as key contributors to approval or rejection. For further reading, refer to MDPI (2023) and DIVA Portal (2023).

Table 2.1: Comparison of Model Strengths and Weaknesses

Model	Strengths	Weaknesses
XGBoost	Excels on tabular data with mixed feature types	Requires careful hyperparameter tuning for optimal performance
Random Forest	Handles non-linearities and feature interactions well	Can be slower to train and less efficient than XGBoost on large datasets
CNNs	Powerful for spatial or image data; able to learn complex patterns	Not naturally suited for tabular data; often underperforms on such data

2.4 Implementation of the model

Now that the data was processed into 70% training, 15% validation, 15% test data split, which helps to prevent overfitting and provides a realistic estimate of how the model will perform in real-world scenarios. Then the next step was to ensure all the necessary libraries were imported, libraries such as xgboost, scikit-learn, shap as seen in Figure 2.2.

2.5 Model Configuration

Configured the **XGBoost Classifier** with several key parameters, such as max_depth, learning_rate, n_estimators, and others that control the model's behavior during training.

```python
# Initialize and train XGBoost model
xgb_model = XGBClassifier(
    objective='binary:logistic',
    n_estimators=100,
    max_depth=5,
    learning_rate=0.8,
    subsample=0.5,
    colsample_bytree=0.9,
    random_state=42
)

# Train the XGBoost model on the training set
xgb_model.fit(X_train, y_train)
```

Figure 2.3: Model Config

Applying evaluation matrices : Accuracy Score, Classification_Report, Confusion Matrix

```python
# Make predictions
y_pred = xgb_model.predict(X_test)

# Store evaluation results
accuracy = accuracy_score(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

# Evaluate model (output remains the same)
print("Accuracy:", accuracy)
print("\nClassification Report:")
print(class_report)
print("\nConfusion Matrix:")
print(conf_matrix)
```

Figure 2.4: Evaluation Matrices

2.6 Hyperparameter Tuning

Fine-tuning a machine learning model involves adjusting its hyperparameters to improve generalization and performance. In this project, instead of conventional methods like GridSearch or RandomizedSearch, we applied a more strategic approach: 4-Vector Optimization. This approach outperformed traditional grid/random search methods, echoing the findings of Yu *et al.* [2024] who achieved a test accuracy of 91.71% with reduced overfitting. This method optimizes four key hyperparameters simultaneously to find an optimal balance between model complexity and predictive power.

For the XGBoost model, the following parameters were considered:

- **max_depth:** Controls the maximum depth of each decision tree. A deeper tree allows the model to learn more complex patterns, but may also overfit.
- **learning_rate:** Determines how quickly the model adapts to the problem. Lower values typically improve performance but require more trees.
- **n_estimators:** The number of boosting rounds. More estimators can improve performance but increase training time and risk of overfitting.
- **subsample:** The fraction of training data used in each boosting round. Lower values can help prevent overfitting.

Then the best values suggested by the 4-Vector were stored then fed back into the model autonomously. Though we applied the 4-vector we initially tried fine tuning the model manually, testing various values into the model to observe its performance. It provided extremely low accuracy scores which did not assist the objective of our project. Hence we decided to resolve the issue using the 4-vector optimization.

2.7 KMeans

The K-Means algorithm was used to cluster loan applicants based on a variety of features into High Risk, Moderate Risk, and Low Risk categories. As an unsupervised learning technique, K-Means groups data points into distinct clusters by analyzing their inherent structure and distribution [GeeksforGeeks 2025b]. The algorithm works by assigning each data point to the nearest centroid (a randomly initialized center point) on each iteration and then recalculating the central points based on the average of the assigned points. This process continues until the central points stabilize, minimizing within-cluster variance and maximizing separation between clusters.

The key advantages of using K-Means in this context include its scalability, efficiency with large datasets, and ability to reveal hidden patterns in unlabeled data [The IoT Academy 2025]. By segmenting loan applicants into risk-based clusters, stakeholders gain a data-driven perspective that enhances decision-making, improves risk assess-

ment, and supports targeted financial strategies. However, K-Means has limitations, such as sensitivity to initial centroid placement and the assumption of spherical, equally sized clusters, which may not always hold true in real-world financial data [The IoT Academy 2025]. Despite these constraints, K-Means remains a powerful tool for exploratory data analysis, providing useful insights into customer risk profiles.

After the model is optimized and applied, the KMeans algorithm is implemented. First the cleaned data is loaded and the features are separated between categorical columns and numeric columns for pre-processing because KMeans does not accept data of object or categorical types, only numeric types. Then through one-hot encoding, the categorical columns are converted. Finally, the numeric columns are scaled, before combining the features again to create a matrix of features for KMeans.

KMeans is implemented making use of the KMeans function in the sklearn.cluster library and is applied on the matrix of features X_cluster. Three clusters are defined and then, the clusters are analysed. A feature called "credit_risk_label" is defined to classify the group of clusters and the clusters are dynamically assigned the labels "high risk", "low risk" and "medium risk" based on the default rate such that customers with higher default rates are assigned high risk. In the analysis, the clustering of KMeans is measured against various metrics such as the silhouette score, adjusted rand index (ARI), normalized mutual information (NMI), homogeneity, completeness and v-measure.

2.8 Explainability & SHAP

To better understand the model's decision-making, we used SHAP (SHapley Additive ExPlanations) values to explain feature importance. As highlighted in Mamun *et al.* [2022], SHAP enables financial professionals to understand the driving features behind each decision, which is key in domains like lending. When we combine these individual explanations, we get an overall view of feature importance. For example, income, credit history, and loan defaults might be the most influential features in deciding whether a loan is approved or rejected.

Library Import : Shap

In addition to that we went a step feather to give reasons why an applicant was rejected or approved based on the various influential features affecting the outcome. Furthermore we supplemented the SHAP with tailored advice to the applicant on how they can improve their credit eligibility or continue their current habits.

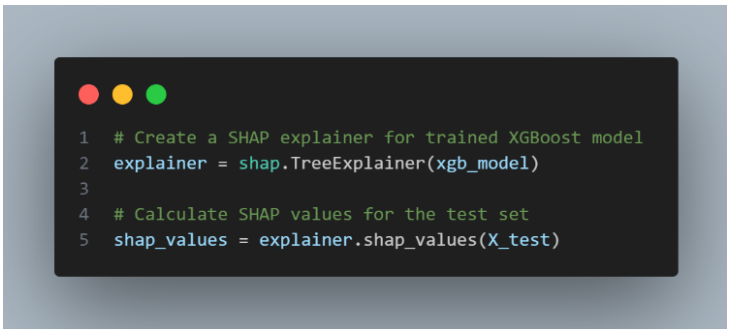

Figure 2.5: SHAP Explainer

Chapter 3

Results and Analysis

3.1 Dataset

The dataset contains 45,000 loan records with 14 columns representing demographic, financial, and loan-related attributes. All columns are complete with zero missing values, which is advantageous for analysis and modelling. Categorical variables are correctly stored as object types and Continuous variables values that are generally within plausible ranges for finance-related data. Several notable outliers and anomalous values were observed during preprocessing this include maximum age of 144, person income exceeding $7 million and Experience ranges up to 125 year. Outliers and anomalous values are caped to reduce noice and . The dataset doesn't have duplicates records. The target(loan_status) is imbalance(22% of borrowers defaulted), this will be critical to address during model training to avoid bias.Overall preprocessing stage reveals that the dataset is rich and largely clean.

Key Characteristics:

- **Size**: 45,000 records
- **Features**: 14 total, with a mix of demographic, financial, and credit-related variables
- **Target Variable**: loan_status (1 = approved, 0 = rejected)
- **Synthetic Generation**: Data was expanded using SMOTENC (a variation of SMOTE that supports categorical variables), typically for handling class imbalance.

Table 3.1: Loan Approval Classification Dataset

Column	Description	Type
person_age	Age of the person	Float
person_gender	Gender of the person	Categorical
person_education	Highest education level	Categorical
person_income	Annual income	Float
person_emp_exp	Years of employment experience	Integer
person_home_ownership	Home ownership status (e.g., rent, own, mortgage)	Categorical
loan_amnt	Loan amount requested	Float
loan_intent	Purpose of the loan	Categorical
loan_int_rate	Loan interest rate	Float
loan_percent_income	Loan amount as a percentage of annual income	Float
cb_person_cred_hist_length	Length of credit history in years	Float
credit_score	Credit score of the person	Integer
previous_loan_defaults_on_file	Indicator of previous loan defaults	Categorical
loan_status	Loan approval status: 1 = approved; 0 = rejected	Integer (Target)

3.1.1 Dataset Analysis

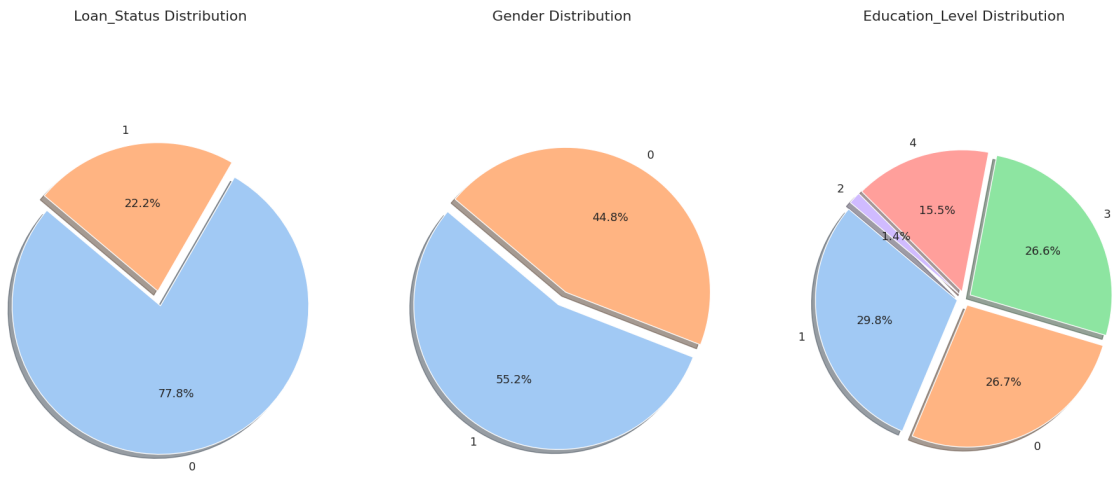

Figure 3.1: PIE CHARTS for Loan_Status Distribution, Gender Distribution and Education_level Distribution

The analysis begins with a series of pie charts, which illustrate the distribution of individual features within the dataset.Figure 3.1 reveals that the majority of loans, specifically 77.8%, fall under Loan_Status 0, while 22.2% are categorized as Loan_Status 1. With the provided description, Loan_Status 0 means “rejected” and Loan_Status 1 means “approved”. This indicates that the vast majority of loan applications over three-quarters are rejected, while less than a quarter are approved. This highlights a significant class imbalance in the target variable, which is a common challenge in loan approval prediction. The gender distribution shows a slight leaning towards Gender 1 at 55.2%, compared to Gender 0 at 44.8%, necessitating an understanding of the categorical encoding to interpret further. Furthermore, the Education_Level distribution indicates that level 1 is the most common at 29.8%, with levels 0 and 3 also being significant, whereas levels 4 and 2 are less frequent, suggesting a diverse educational background among loan applicants.

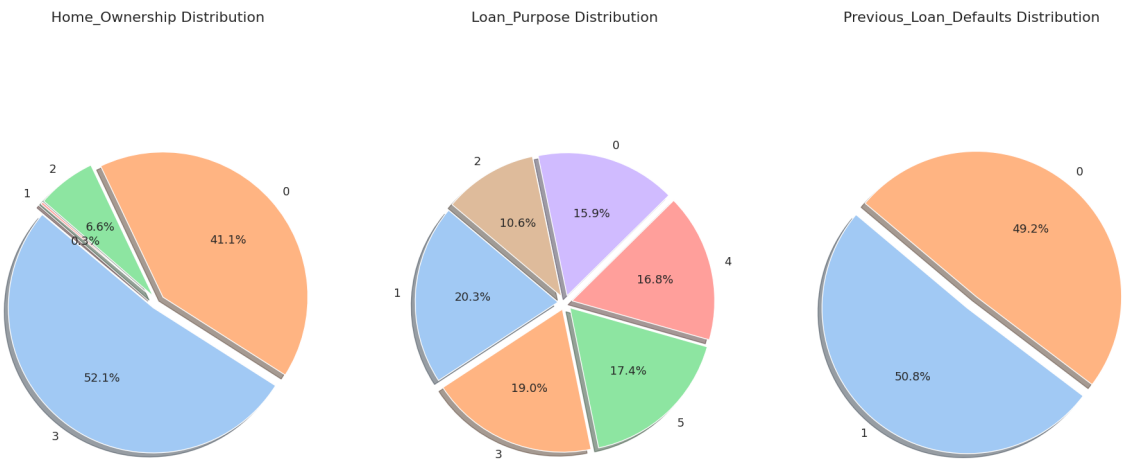

Figure 3.2: PIE CHARTS for Home_Ownership Distribution, Loan_Purpose Distribution and Previous_Loan_Defaults Distribution

Figure 3.2 presents insights into home ownership, loan purpose, and previous loan defaults. The Home Ownership distribution shows that categories 3 and 0 are the most prevalent, accounting for 52.1% and 41.1% respectively, with categories 1 and 2 being much less common. The Loan Purpose distribution is more varied, with different purposes showing distinct percentages, indicating a range of reasons for loan applications. Finally, the Previous Loan Defaults chart reveals an almost even split, with 49.2% of applicants having no prior defaults category 0 and 50.8% having experienced previous defaults category 1. This even split suggests that previous loan history is a critical factor to consider for risk assessment, as a substantial portion of the applicant pool has a history of default.

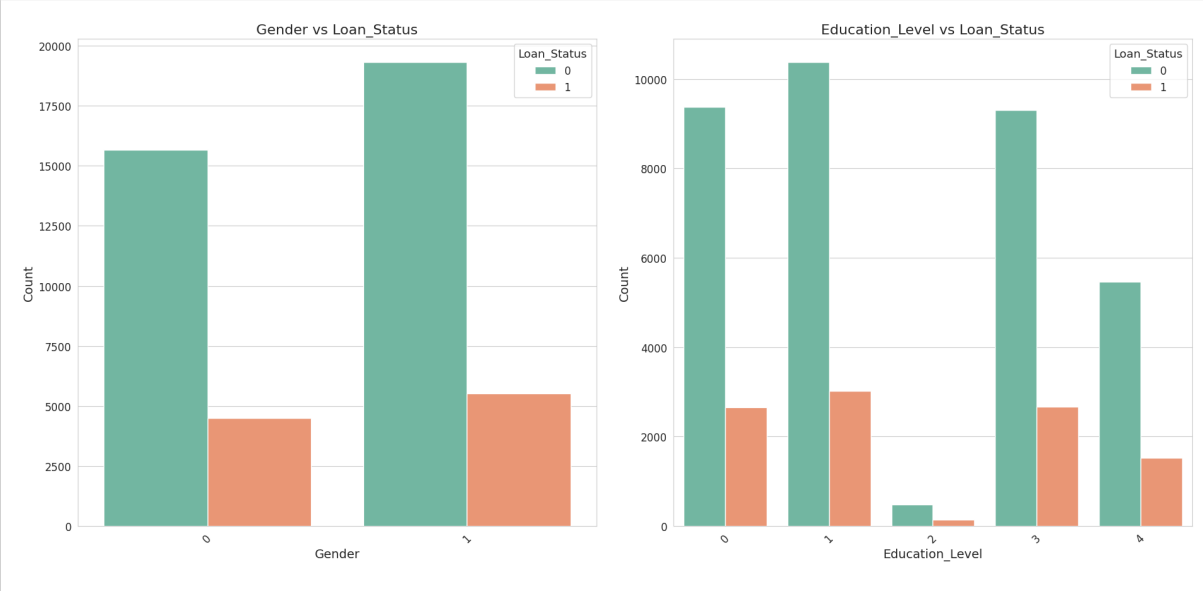

Figure 3.3: BAR CHARTS for Gender VS Loan_Status and Education_Level VS Loan_Status

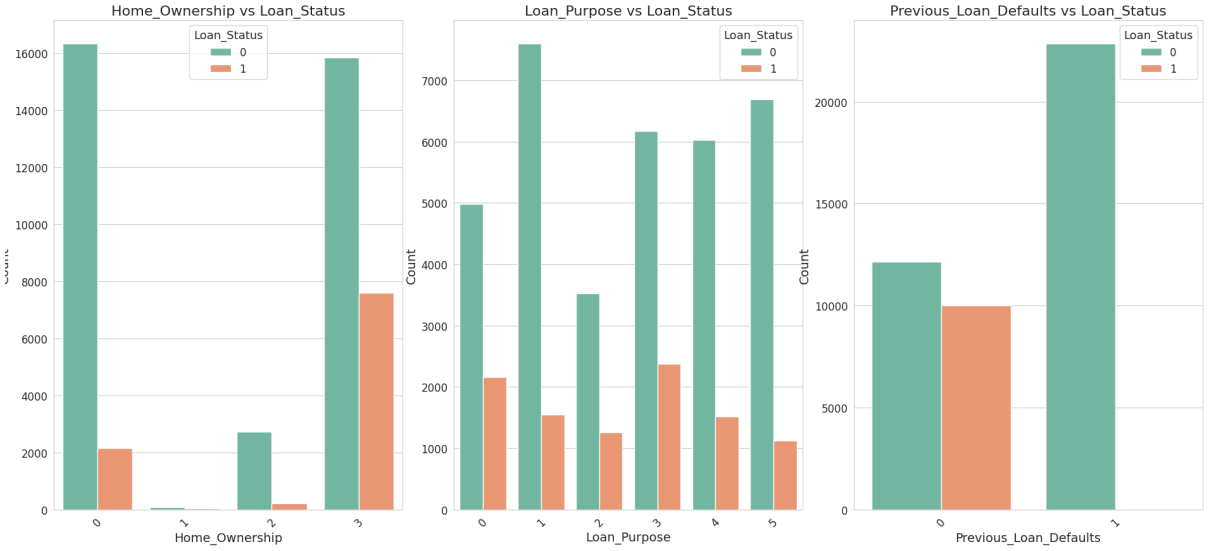

Figure 3.4: BAR CHARTS for Home_Owenership VS Loan_Status, Loan_Purpose VS Loan_Status and Previous_Loan_Defaults VS Loan_Status

The subsequent bar charts Figure 3.3 and Figure 3.4 delve into the relationships between various features and the Loan_Status, providing crucial information for identifying potential predictors of loan approval or rejection. In these visualizations, Loan_Status 0 now unequivocally represents “rejected” and Loan_Status 1 signifies “approved.” The first bar chart Figure 3.3 examines Gender vs Loan_Status, showing that both gender categories 0 and 1 have a significantly higher count for rejected applications Loan_Status 0 than for approved applications Loan_Status 1. This finding aligns with the overall

low approval rate observed in the initial Loan_Status distribution. To ascertain if gender plays a disproportionate role in approval rates, a comparison of the proportion of approvals within each gender category would be necessary. The Education_Level vs Loan_Status chart demonstrates that for most education levels 0, 1, 3, 4, the number of rejected loans Loan_Status 0 substantially outweighs the number of approved loans Loan_Status 1. Education levels 1 and 3 appear to have the highest overall application counts, and consequently, the highest number of rejections. Education_Level 2, with its very low overall count, shows very few approvals. To determine which education levels are associated with a higher approval rate, calculating the proportion of approvals relative to total applications for each level would be essential.

The second set of bar charts Figure 3.4, the Home_Ownership vs Loan_Status chart reveals that Home_Ownership categories 0 and 3 account for a large volume of applications, with a clear dominance of rejected loans Loan_Status 0. Interestingly, Home_Ownership 1, despite having a low overall application count, shows a relatively low number of approved loans Loan_Status 1 compared to its rejected counterparts, potentially indicating a higher likelihood of rejection for this category. Home_Ownership 2, with its extremely low application volume, shows almost no approvals. Further analysis of approval rates per home ownership type would be beneficial for deeper understanding.

The Loan_Purpose vs Loan_Status chart illustrates that for most loan purpose categories, the number of rejected loans Loan_Status 0 far exceeds the number of approved loans Loan_Status 1. Loan_Purpose 0 and 5 seem to have a larger share of total applications and rejections. Again, calculating the approval rate for each loan purpose would clarify which purposes are associated with a higher approval likelihood. Most strikingly, the Previous_Loan_Defaults vs Loan_Status chart strongly indicates that prior defaults are a powerful predictor of loan rejection. Applicants with no previous defaults Previous_Loan_Defaults 0 overwhelmingly have their loans rejected Loan_Status 0, though a substantial number are also approved. Conversely, for applicants with previous defaults Previous_Loan_Defaults 1, the number of approved loans Loan_Status 1 is significantly lower compared to the number of rejected loans Loan_Status 0, unequivocally demonstrating that having previous loan defaults is strongly associated with a higher likelihood of loan rejection.

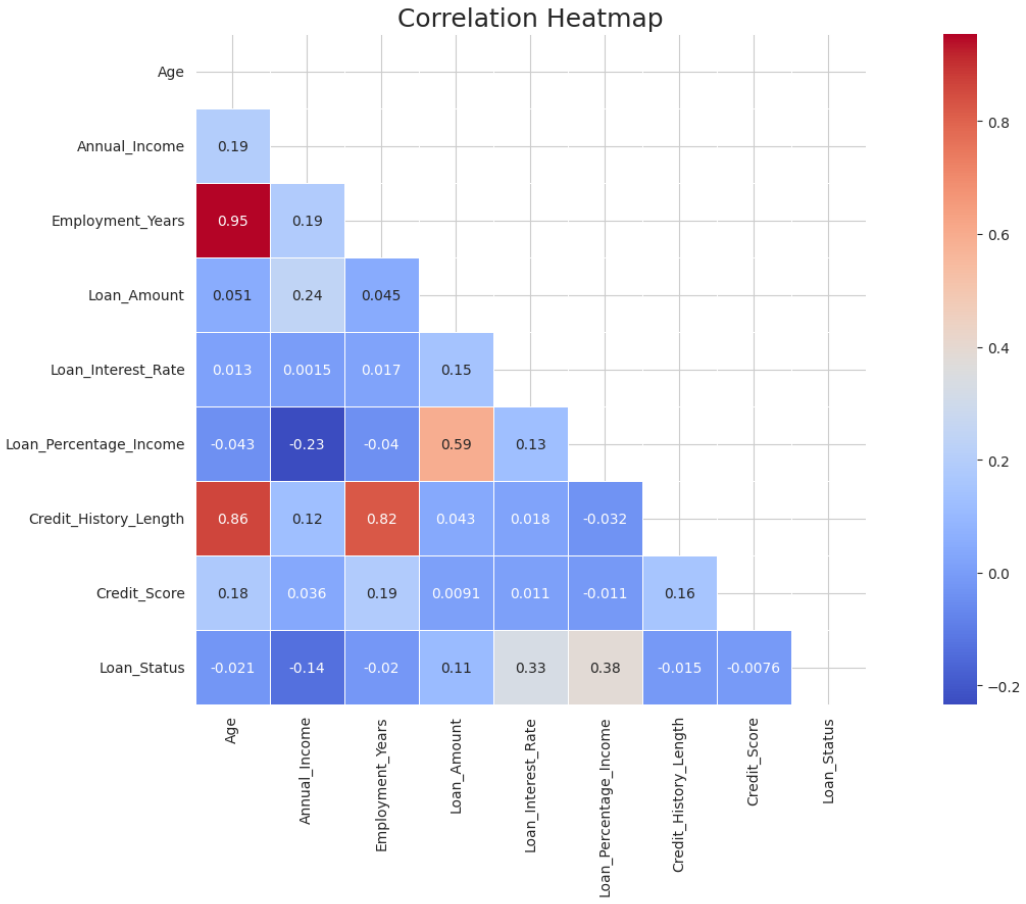

Figure 3.5: Correlation Heatmap

Figure 3.5 provides insights into the linear relationships between numerical features and the Loan_Status. The correlation values range from -1 perfect negative correlation to 1 perfect positive correlation, with 0 indicating no linear relationship. The color scale illustrates the strength and direction of the correlation reds for positive, blues for negative. Examining the correlations with Loan_Status the last row/column, a moderate positive correlation of 0.33 is observed with Loan_Interest_Rate, suggesting that higher interest rates are associated with an increased likelihood of loan approval Loan_Status 1. This is a particularly intriguing finding, as it could imply that loans with higher perceived risk are still being approved, but with a higher interest rate to compensate for that risk, or that the bank strategically approves loans that yield higher returns, even if they carry a somewhat elevated risk. A weak positive correlation of 0.13 with Loan_Percentage_Income indicates that as the loan amount as a percentage of income increases, there's a slight tendency for the loan to be approved.
Loan_Amount also shows a weak positive correlation of 0.11 with Loan_Status.

Conversely, several variables exhibit counter-intuitive weak negative correlations with Loan_Status meaning they are slightly associated with rejection. person_emp_exp (Employment_Years) shows a very weak negative correlation of -0.045, suggesting that

more years of employment experience are slightly associated with loan rejection. Annual_Income exhibits a weak negative correlation of -0.14, implying that higher annual incomes are slightly associated with rejected loans. Similarly, Age shows a very weak negative correlation of -0.021. These negative correlations are surprising, as traditionally, longer employment, higher income, and generally older age up to a certain point are considered favorable attributes for loan approval.

Most strikingly, Credit_Score shows an extremely weak negative correlation of -0.0076 with Loan_Status. Given that Credit_Score is a cornerstone of credit risk assessment, this near-zero linear correlation with loan approval is highly unexpected and warrants extensive further investigation. It could suggest a non-linear relationship not captured by Pearson correlation, that other factors overwhelmingly dictate the approval decision in this dataset, or an issue with how the Credit_Score variable itself is behaving. Among other features, person_emp_exp (Employment_Years) and Age display a very strong positive correlation of 0.95, as expected, since older individuals typically have more years of employment experience. Credit_History_Length shows a very strong positive correlation with Credit_Score (0.86) and a strong positive correlation with person_emp_exp (Employment_Years) (0.82), highlighting logical inter-dependencies within these financial and employment history variables.

These correlations are crucial for understanding potential multi-collinearity in model building and identifying features that might be redundant or require transformation. The insights derived from these charts, especially the presence of counterintuitive correlations, are paramount for constructing an explainable and interpretable loan prediction model. They underscore the need for a more in-depth investigation into the specific decision-making process for loan approvals and rejections within this particular dataset, moving beyond simple linear relationships.

3.2 XGBoost results

The XGBoost model demonstrated robust performance on the loan approval task, achieving high accuracy and balanced precision/recall across both classes. On the test set, it correctly classified the majority of applications, with strong detection rates for both approved and rejected loans. Moreover, interpretability analysis via SHAP confirmed that the model's decisions were driven by financial indicators in the cleaned data set—such as credit score, previous loan defaults, annual income, highlighting both the effectiveness and transparency of the approach.

XGBoost Model Performance **Before Hyperparametization**

```
Accuracy: 0.8860740740740741

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.93      0.93      5250
           1       0.76      0.72      0.74      1500

    accuracy                           0.89      6750
   macro avg       0.84      0.83      0.83      6750
weighted avg       0.88      0.89      0.88      6750
```

Confusion matrix before **Hyperparametization.**

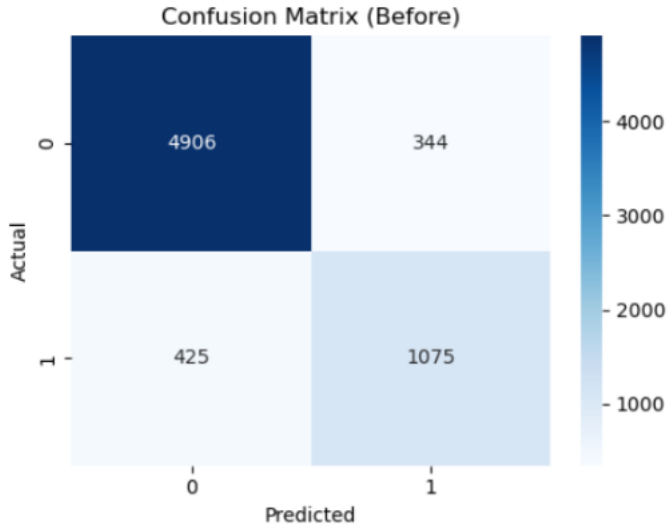

Figure 3.6: Confusion matrix Before

Model Hyperparametization **4Vector**

After several iterations, the function returns the best hyperparameters found and their corresponding validation accuracy. **Best Hyperparameters:**

```
 Max_depth: 9
 Learning_rate: 0.1624
 Subsample: 0.5925
 Colsample_bytree: 0.7206
Validation Accuracy: 0.9047
```

XGBoost Model Performance **After Hyperparametization.**

```
Accuracy: 0.902074074074074

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.95      0.94      5250
           1       0.80      0.74      0.77      1500

    accuracy                           0.90      6750
   macro avg       0.87      0.84      0.85      6750
weighted avg       0.90      0.90      0.90      6750
```

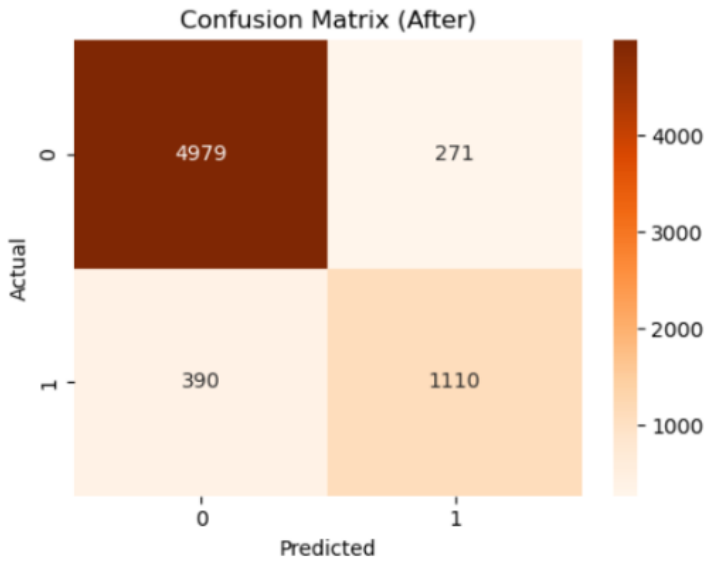

Figure 3.7: Confusion matrix After

The above 2 Confusion Matrix results show exactly where the XGBoost model is making the correct predictions and making mistakes:

- True Positives (TP — Top Left) Approved applicants correctly identified **4906 — 4979**
- True Negatives (TN — Bottom Right) Rejected applicants correctly identified **1075 — 1110**
- False Positives (FP — Top Right) Rejected applicants wrongly approved **334 — 271**
- False Negatives (FN — Bottom Left) Approved applicants wrongly rejected **425 — 390**

Accuracy Comparison Summary

The accuracy of the the **XGboost model** showed a clear improvement after fine-tuning its hyperparameters. The base model, trained using default or initial parameter values, achieved an accuracy of 0.8861. After applying **4-vector optimization** to adjust key hyperparameters such as max_depth, learning_rate, n_estimators, and subsample, the accuracy increased to 0.9021. This improvement of approximately 1.6 percentage points demonstrates that targeted hyperparameter tuning significantly enhanced the model's ability to generalize and make correct predictions on unseen data. It validates the effectiveness of the optimization strategy and underscores the importance of model refinement in achieving high-performance results.

3.2.1 Interpretation and Eligibility Explainer

	Loan_Id	Outcome	Reasons	Advice	Risk Profile	Risk Profile Summary
0	Loan_1	Rejected	**Previous_Loan_Defaults** contributed negatively to loan approval. **Loan_Amount** contributed negatively to loan approval. **Credit_Score** contributed negatively to loan approval. **Loan_Purpose** contributed negatively to loan approval.	Loan purpose is HighRisk. Avoid Loan defaults and maintain a clean repayment record to boost your chances. Try applying for a lower loan amount or extending your repayment period. Work on improving your credit score by paying bills on time and reducing outstanding debts.	High Risk	This segment is characterized by moderate incomes, weaker credit scores, and shorter credit histories. They experience the highest default rates and lowest approval rates, indicating significant financial risk.
1	Loan_2	Approved	**Annual_Income** contributed positively to loan approval. **Loan_Amount** contributed positively to loan approval. **Previous_Loan_Defaults** contributed positively to loan approval. **Credit_Score** contributed positively to loan approval. **Credit_History_Length** contributed positively to loan approval.	No specific advice. Keep up the good financial habits.	Low Risk	Applicants in this group typically have steady incomes, strong credit scores, and long credit histories. They show the lowest default rates and highest loan approval rates—representing the safest financial profile.
2	Loan_3	Rejected	**Loan_Amount** contributed negatively to loan approval. **Annual_Income** contributed negatively to loan approval. **Home_Ownership** contributed negatively to loan approval.	Consider increasing your annual income or adding a co-applicant to strengthen your application. Try applying for a lower loan amount or extending your repayment period. Having stable home ownership can improve eligibility. Provide additional documentation if possible.	Low Risk	Applicants in this group typically have steady incomes, strong credit scores, and long credit histories. They show the lowest default rates and highest loan approval rates—representing the safest financial profile.

Figure 3.8: explanations

Human-readable explanations are produced for individual loan applicants in the test set, using SHAP values to identify the most influential features for each prediction. For each applicant, it retrieves their SHAP values and feature values, predicts their loan outcome, and determines whether the loan was approved or rejected. It then identifies the top features that most strongly influenced the prediction and, for each, records whether it contributed positively or negatively. If a feature contributed negatively to a rejected application, the code adds tailored advice on how the applicant might improve their chances in the future (for example, by increasing income or improving credit score).This makes it easy to review, for each applicant, the key factors behind the model's decision and any actionable feedback. The results produced by this explainer are directly based on the model's results. The explainer uses the trained XGBoost model (xgb_model) and its predictions on the test set (X_test). It leverages the SHAP values, which quantify how much each feature contributed to each individual prediction made by the model. **The Reasons , Advice and Risk Profile are tailored for each applicant**.

3.3 KMeans results

For analysis purposes, a cluster profile Table 3.5, 3.4, 3.7, and 3.6 is outputted in which each cluster under the heading "credit_risk_label" is compared against the following features: Annual_Income, Loan_Amount, Credit_History_Length, Credit_score, Previous_Loan_Defaults, Home_Ownership and Loan_Status.

Cluster 0 which is "Low risk" has the 2nd most individuals, cluster 1 which is "Medium risk" has the least individuals and cluster 2 which is "High Risk" has the most individuals.

Based on Table 3.5, 3.4, 3.7, and 3.6 it can be observed that the "Low risk" cluster are individuals with the lowest income, moderate Loan amounts, shortest credit history lengths, moderate credit scores, lowest previous loan defaults, likely to mortgage homes, and likely to use loans for medical purposes. They are additional individuals with the highest loan repayments at a rate of 46%. When combined with the results in Table 2, it can be seen that Low Risk individuals have a 53% default rate and 46 non-default rate. Overall, these individuals have a fairly stable financial profile.

Additionally, "Medium Risk" individuals are people that typically have the highest annual income, request high loan amounts, have a high credit history length, high credit scores and a moderate rate of previous loan defaults (47%). These individuals typically own homes and mortgage homes. People in the medium risk category also typically use loans for educational, venture, and home improvement purposes. This group also has a loan successful repayment rate of 21%. When observing these figures along with the figures in Table 3.2 it is logical to include that such individuals while they are not high risk, they pose a moderate risk likely due to their previous loan defaults.

Furthermore, those grouped into 'high risk' are people who can be characterized by their moderate annual income and low loan_amounts. However, these individuals have a moderate credit history length and the lowest credit score. In addition, such individuals have the highest previous loan defaults. These individuals typically own and mortgage their homes. They also use their loans for personal purposes and home improvement purposes. These individuals therefore pose the highest risk.

Overall it can be deduced that the individuals in cluster 0 typically pose the lowest risk in loan eligibility because although these individuals earn the least annual income, they are typically individuals that have the least amount of previous loan defaults making them the most reliable. The applicants in cluster 2 pose the highest risk in loan eligibility, because although they are individuals with a moderate annual income, they are also individuals that typically have lower credit scores and high previous loan defaults making them unreliable customers. The applicants in cluster 1, pose a moderate risk because although they typically earn the most in annual income, they request higher loan_amounts, have a higher credit history length, and the second highest loan defaults making them reliable or unreliable depending on other factors.

The clustering was evaluated further by looking at common metrics used for KMeans such as the silhouette score, ARI, NMI, homogeneity, completeness and v-measure. The silhouette score is used to measure how well grouped the clusters are by checking if points are closer to their own cluster than to other clusters. A high score means well-separated clusters, while a low score suggests overlap or poor grouping [GeeksforGeeks 2025a].The ARI metric is used to measure how accurate the clustering is to ground truth [GeeksforGeeks 2025a]. The NMI metric determines the level of agreement among assigned clusters. V-measure measures how well clusters are formed by looking at information overlap. A higher score means better clustering. It combines homogeneity, which ensures each cluster only has members of one group (like precision), and completeness, which makes sure all members of a group are placed in the same cluster (like recall). V-measure balances these two, similar to how the F-score balances precision and recall in classification [Towards Data Science 2025].

The clustering evaluation metrics in Table 3.3 are as follows: Silhouette Score of 0.20, ARI of 0.15, and NMI of 0.21—reflect the nuanced and overlapping nature of the clusters.

Although these scores indicate modest cluster cohesion and limited alignment with any predefined labels, the cluster profile in Table 3.5, 3.4, 3.7, and 3.6 above still provides meaningful insights into the financial behaviors and risk categories of the individuals. The relatively low Silhouette Score suggests overlap between clusters, which is reinforced by the PCA projection, which is expected given the complexity of financial risk factors and the continuous nature of the features involved.

The homogeneity score (0.31) being higher than completeness (0.16) aligns with the observation that clusters tend to be internally consistent but do not fully capture all variations across the dataset. For example, the "Low Risk" cluster, despite having the lowest income group, shows the fewest previous loan defaults and highest loan non-default rates, indicating a stable financial profile. The "Medium Risk" cluster, with the highest incomes but moderate defaults, and the "High Risk" cluster, characterized by moderate income but poor credit scores and high defaults, illustrate overlapping but distinct financial behaviors that naturally limit crisp cluster boundaries. These results suggest that while the clusters are not perfectly separable, they capture relevant financial risk patterns that can inform loan eligibility decisions.

Table 3.8 and 3.9, along with Figure 3.9, show how loan applicants are grouped into clusters based on actual loan defaults (ground truth) and the model's predictions (XGBoost). The data shows a clear pattern, where the High Risk cluster is largely associated with loan defaults, while the Low Risk and Medium Risk clusters contain a balanced mix of non-defaults and defaults. This strong alignment suggests that the clustering method effectively reflects actual loan decisions. This suggests the model effectively captures risk patterns, making its predictions reliable.

The results of the KMeans clustering are summarized and incorporated into the SHAP explainer to reinforce the model's decisions and provide deeper insights into the dataset.

In this section, each cluster is assigned a risk profile label (High Risk, Medium Risk, Low Risk), accompanied by a summary that characterizes the typical behavior and financial attributes of individuals within that cluster. This integration helps to contextualize the model's predictions by linking cluster-based risk profiles with feature importance explanations, thereby offering a more comprehensive understanding of both the data and the model's decision-making process.

Table 3.2: Default Rate per Cluster

credit_risk_label	Loan_Status 0	Loan_Status 1
High Risk	1.000000	NaN
Low Risk	0.536314	0.463686
Medium Risk	0.792424	0.207576

Table 3.3: Clustering Evaluation Metrics (KMeans)

Metric	Score
Silhouette Score	0.1998
Adjusted Rand Index (ARI)	0.1482
Normalized Mutual Information (NMI)	0.2146
Homogeneity	0.3143
Completeness	0.1630
V-Measure	0.2146

Table 3.4: Cluster Profile: Part 1

credit_risk_label	count	Annual_Income	Loan_Amount
High Risk	19453	77484.68	8396.27
Low Risk	18340	66181.92	8989.12
Medium Risk	7207	123944.97	14298.45

Table 3.5: Cluster Profile: Part 2

credit_risk_label	Credit_History	Credit_Score	Prev_Defaults
High Risk	4.73	618.53	1.00
Low Risk	4.69	637.95	0.00
Medium Risk	11.95	657.02	0.47

Table 3.6: Cluster Profile: Part 3

credit_risk_label	RENT	OWN	MORTGAGE
High Risk	0.00	0.08	0.47
Low Risk	0.00	0.05	0.62
Medium Risk	0.00	0.07	0.40

Table 3.7: Cluster Profile: Part 4

Label	Pers	Edu	Med	Vent	HomeImp	Status
High Risk	0.23	0.09	0.18	0.17	0.19	0.00
Low Risk	0.19	0.11	0.21	0.16	0.15	0.46
Medium Risk	0.16	0.15	0.17	0.18	0.18	0.21

Table 3.8: Cross-tabulation (Cluster vs Outcome)

credit_risk_label	Loan_Status 0	Loan_Status 1
High Risk	1.000000	0.000000
Low Risk	0.536314	0.463686
Medium Risk	0.792424	0.207576

Table 3.9: Cross-tabulation of Cluster vs XGBoost Prediction

credit_risk_label	0	1
High Risk	1.00	0.00
Low Risk	0.57	0.43
Medium Risk	0.81	0.19

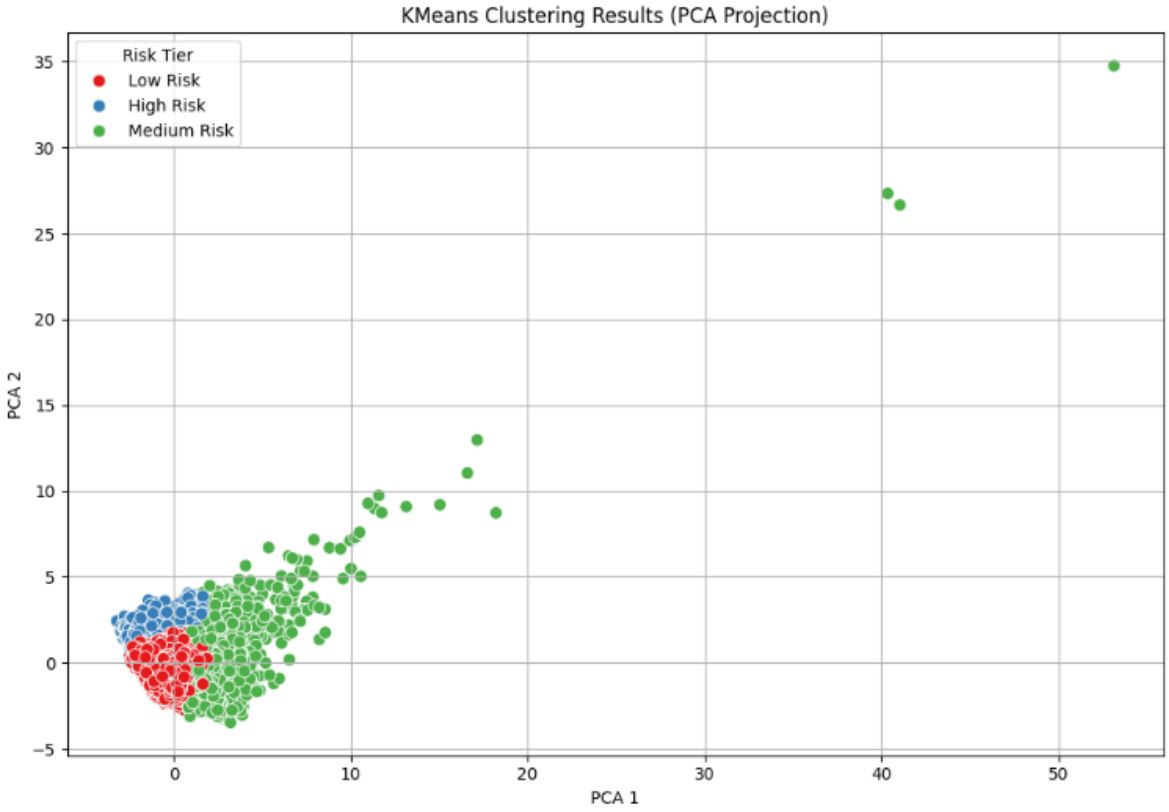

Figure 3.9: KMeans Clustering Results(PCA projection)

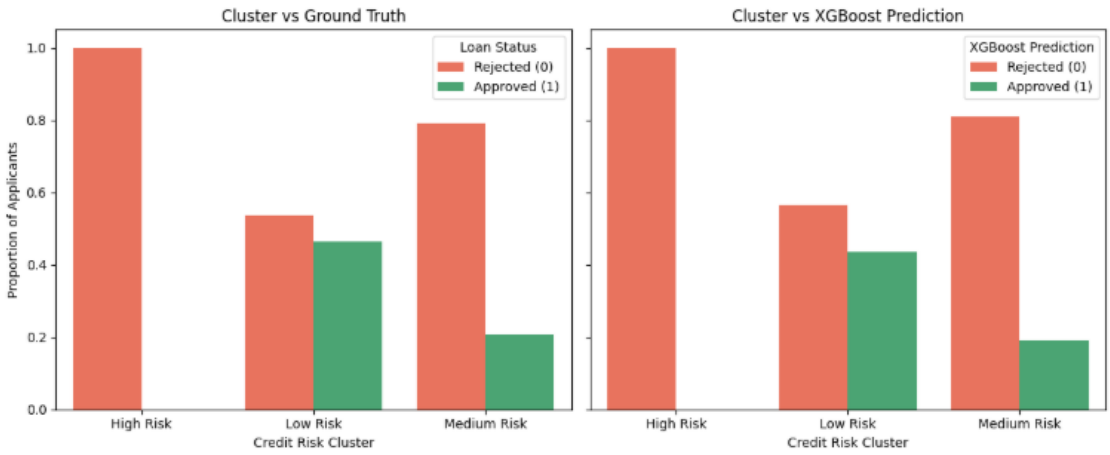

Figure 3.10: Cluster VS Ground Truth and Cluster VS XGboost Prediction

Chapter 4

Conclusion

The project results show that an optimized XGBoost model, combined with interpretability and risk segmentation tools, can provide both high accuracy and clear, understandable decisions for loan approval. The base model reached an accuracy of 88.6%, with strong precision and recall for approved loans (0.92 and 0.93) and decent performance for rejected loans (precision 0.76, recall 0.72). After applying a population-based hyperparameter optimization, the accuracy improved to 90.2%, with F1-scores of 0.94 for approved and 0.77 for rejected loans. The confusion matrix supported these results, showing low numbers of false positives and negatives.

More than just accuracy, the use of SHAP values helped generate easy-to-understand explanations for each loan decision, which is important for building trust and ensuring transparency. KMeans clustering was also used to group applicants into different risk levels, helping lenders better understand borrower profiles and make more informed decisions. Overall, the combination of strong performance, clear explanations, and useful borrower groupings shows that a well-tuned XGBoost model can be a practical and reliable tool for loan approval, while also addressing fairness and accountability.

References

[GeeksforGeeks 2025a] GeeksforGeeks. *Clustering Metrics*. https://www.geeksforgeeks.org/clustering-metrics/, 2025. Accessed: 2025-05-30.

[GeeksforGeeks 2025b] GeeksforGeeks. *K-Means Clustering Introduction*. https://www.geeksforgeeks.org/k-means-clustering-introduction/, 2025. Accessed: 2025-05-30.

[Gorishniy *et al.* 2023] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. *Revisiting Deep Learning Models for Tabular Data*, 2023.

[Mamun *et al.* 2022] MA Mamun, Afia Farjana, and Muntasir Mamun. Predicting bank loan eligibility using machine learning models and comparison analysis. In *Proceedings of the 7th North American International Conference on Industrial Engineering and Operations Management*, pages 12–14, 2022.

[Ouyang 2024] Yi Ouyang. Loan default prediction based on logistic regression and xgboost modeling. In *2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT)*, pages 1145–1149. IEEE, 2024.

[Shwartz-Ziv and Armon 2022] Ravid Shwartz-Ziv and Amit Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[The IoT Academy 2025] The IoT Academy. *K Means Clustering in Machine Learning*. https://www.theiotacademy.co/blog/k-means-clustering-in-machine-learning/, 2025. Accessed: 2025-05-30.

[Towards Data Science 2025] Towards Data Science. *7 Evaluation Metrics for Clustering Algorithms*. https://towardsdatascience.com/7-evaluation-metrics-for-clustering-algorithms-bdc537ff54d2/, 2025. Accessed: 2025-05-30.

[Yu *et al.* 2024] Keke Yu, Siwei Xia, Yitian Zhang, and Shikai Wang. Loan approval prediction improved by xgboost model based on four-vector optimization algorithm. *Applied and Computational Engineering*, 82:35–44, 2024.