Analysis of Loan Prediction Dataset

Course Title: Data Analytics

Course Id: CSE 6207 (N)

Submitted By:

Name: F. M. Shefat Hossain Niloy

ID: 0122310011

1.Introduction

1.1. Background

Loan prediction plays a crucial role in the financial sector, enabling lenders to assess the risk associated with granting loans to potential borrowers. Accurately predicting loan risk helps minimize risk and ensures the financial stability of lending institutions. This project utilizes a dataset containing various financial and personal characteristics of individuals to build a model for predicting their likelihood of being a credit risk.

1.2. Objectives

The primary objective of this project is to develop a machine learning model that can effectively predict loan risk based on an individual's financial background and demographics. The model aims to identify individuals who are more likely to return their loans, allowing lenders to make informed lending decisions and mitigate potential financial losses. Additionally, the project seeks to gain insights into the key factors that significantly influence loan risk, providing valuable information for improving credit risk assessment practices.

2. Dataset Description

2.1. Dataset Overview

The dataset used in this project was obtained from Kaggle and contains information about individuals' loan applications. It contains 20 columns and over 55,000 rows, providing details on various financial and personal characteristics of the borrowers. The dataset includes features such as:

- **Personal Finances:** This includes income, expenses, and net income, providing insights into the individual's financial standing.
- **Demographics:** Age, gender, work experience, and marital status provide information about the borrower's personal background.
- **Home and Car:** Ownership of a house or car can indicate financial stability and potential assets.
- **Job and Location:** Details about the type of job, city, state, and duration of employment provide information about the individual's career stability.

- **Family:** The number of family members can influence financial obligations and expenses.
- **Credit Information**: This includes the type of credit the individual has, their credit score, and information about co-applicants, which are crucial indicators of creditworthiness.
- **Risk Assessment:** The target variable is the 'Risk_Flag,' which indicates whether the individual is considered a credit risk (1) or not (0). This variable serves as the basis for building the predictive model.

2.2. Feature Description

Here's a detailed description of each feature in the dataset:

Feature Name	Data Type	Description
Id	Integer	Unique identifier for each individual
Gross_Income	Numerical	Total income of the individual
Expenses	Numerical	Total monthly expenses of the individual
Net_Income	Numerical	Difference between gross income and expenses
Age	Numerical	Age of the individual
Gender	Categorical	Gender of the individual
Experience	Numerical	Number of years of work experience
Marital_Status	Categorical	Marital status of the individual
House_Ownership	Categorical	Whether the individual owns a house
Car_Ownership	Categorical	Whether the individual owns a car
Profession	Categorical	Profession of the individual
City	Categorical	City of residence
State	Categorical	State of residence
Current_Job_Yrs	Numerical	Number of years in the current job
Current_House_Yrs	Numerical	Number of years living in the current house
Nos_Family_Members	Numerical	Number of family members
Credit_Type	Categorical	Type of credit the individual has
Credit_Score	Numerical	Credit score of the individual
Co-Applicant_Credit_Type	Categorical	Type of credit co-applicant has
Risk_Flag	Categorical	Indicates whether the individual is a credit risk (1) or not (0)

3. Data Preprocessing

3.1. Data Loading

After importing necessary libraries, the dataset was loaded into the Python environment using the pandas library. The *pd.read_csv()* function was used to read the CSV file and create a pandas DataFrame for further analysis.

```
# Importing the Loan_Prediction_Dataset
df = pd.read_csv('Loan_Prediction_Dataset.csv')


df.head()
```

df.head() displayed the first few rows of the DataFrame df, allowed us to quickly inspect the structure and contents of the dataset. By default, it displayed the first 5 rows.

3.2. Data Cleaning

The initial analysis revealed the presence of missing values in some features. To address this, appropriate imputation techniques were applied:

Handling Missing Values: Numerical features with missing values were filled using the mean value of the corresponding feature. Categorical features with missing values were filled using the mode value of the corresponding feature. This ensured that all features contained complete data for subsequent analysis and model training.

```
[ ]  # Identifying numerical and categorical columns
     numerical_cols = df.select_dtypes(include='number').columns
     categorical_cols = df.select_dtypes(include='object').columns

     # Filling missing values with mean for numerical columns
     df[numerical_cols] = df[numerical_cols].fillna(df[numerical_cols].mean())

     # Filling missing values with mode for categorical columns
     df[categorical_cols] = df[categorical_cols].fillna(df[categorical_cols].mode().iloc[0])
```

df[numerical_cols] = df[numerical_cols].fillna(df[numerical_cols].mean()): This line fills missing values in numerical columns (numerical_cols) with the mean of each column. Missing values are replaced with the mean value of the respective column.

df[categorical_cols] = df[categorical_cols].fillna(df[categorical_cols].mode().iloc[0]): This line fills missing values in categorical columns (categorical_cols) with the mode (most frequent value) of each column. Missing values are replaced with the mode value of the respective column.

Removing Duplicate Rows: Duplicate rows were checked to maintain data quality. But in our dataset there was no duplicate rows.

Outlier Detection: Outlier data was checked using box plot for our dataset. It doesn't contain significant outlier data.

4. Exploratory Data Analysis

4.1 Data Visualization

Various visualizations were created to explore the distribution of numerical features, identify relationships between variables, and discover potential patterns within the data. Some of the key visualizations used include Box plot, Scatter plot, heatmaps, Pie charts and bar charts.

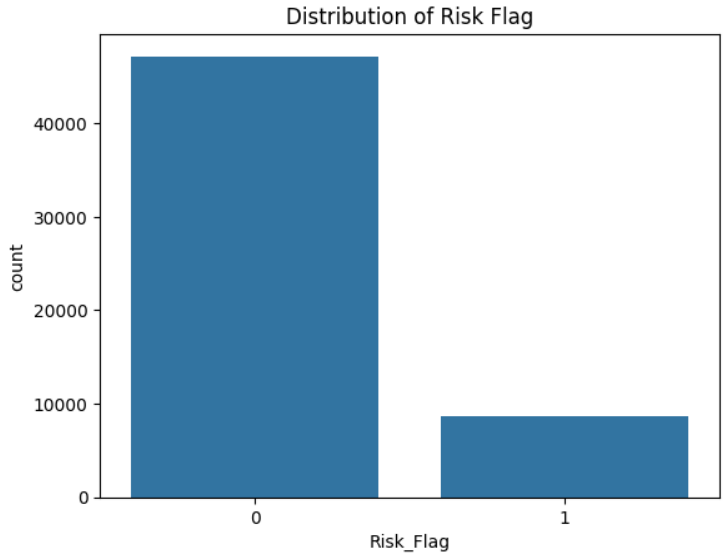


Fig: Risk Flag distribution in dataset

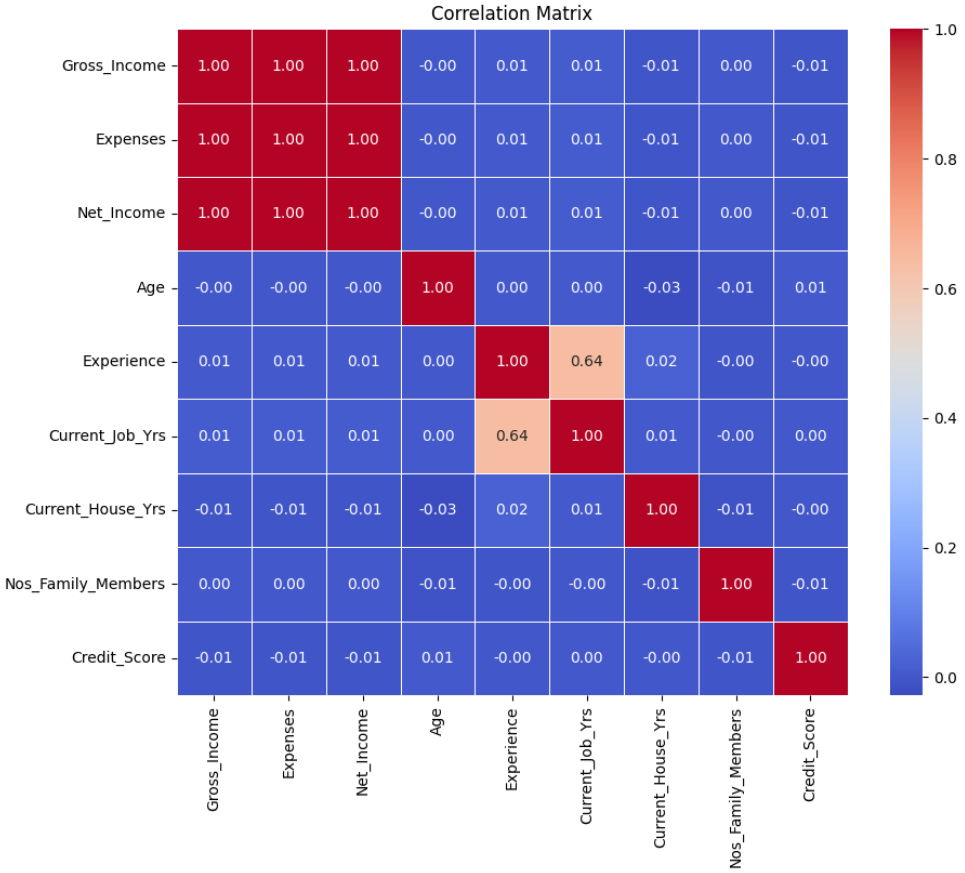


Fig: heatmap for correlation matrix

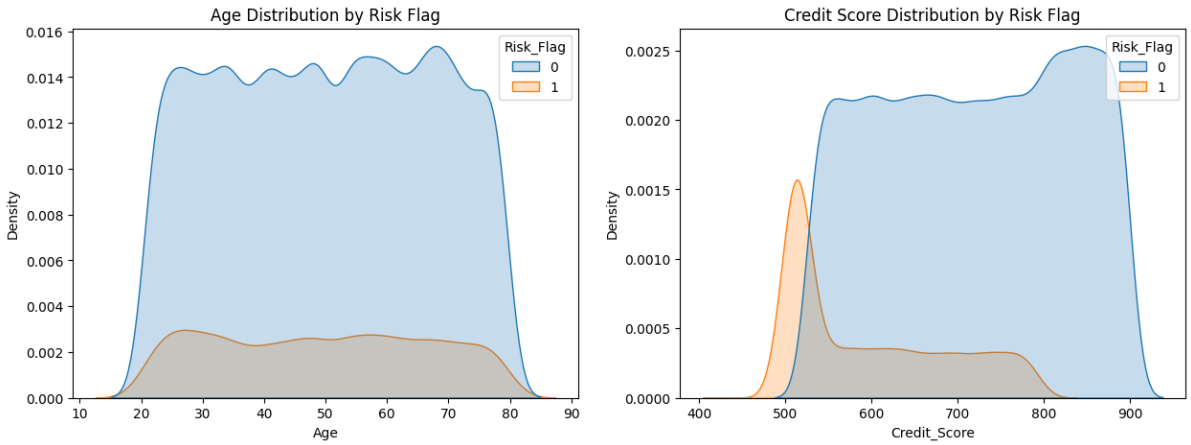


Fig: KDE plot for Age distribution and Credit Score

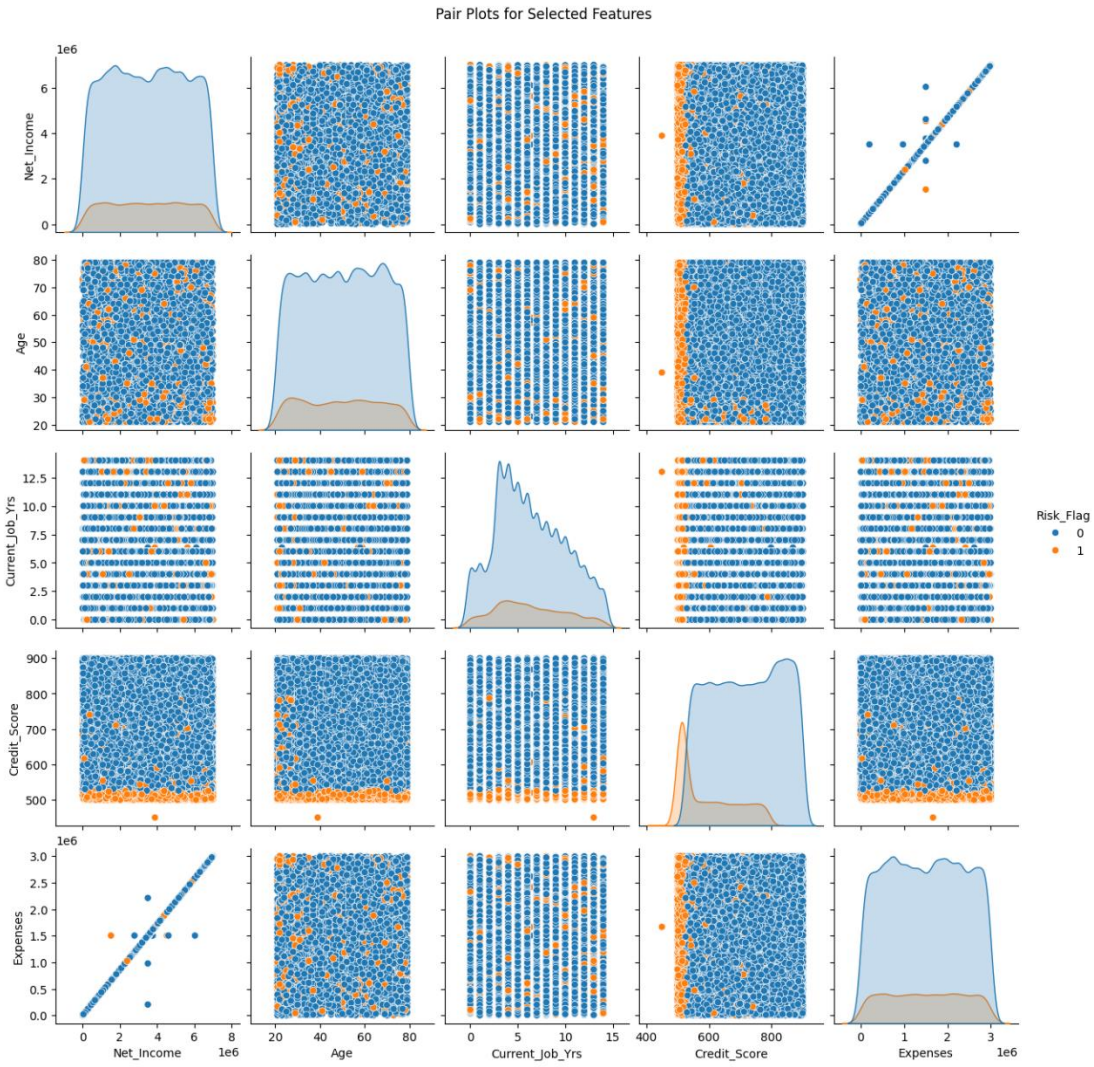


Fig: Pair Plot for different features

By analyzing these visualizations, I have gained a deeper understanding of the data characteristics, patterns, and identified potential relationships between variables that could be relevant for predicting loan risk.

4.2 Insights and Interpretations

The EDA revealed several key insights:

- The distribution of income and expenses showed that a significant portion of individuals expense increases with their income.
- Credit score exhibited a positive correlation with income and low risk, suggesting that individuals with better credit scores have low loan risk.

- There were slight variations in income distribution across different professions, with certain professions showing higher average income levels.
- There are outliers in the Net Income vs Expenses plot. These outliers represent individuals with very high expenses despite having a moderate income. This could be due to factors like debt, high cost of living, or large dependents in the family.
- These insights provided valuable preliminary information about the characteristics of individuals in the dataset and potential factors that might influence their loan risk.

5. Model Training and Evaluation

5.1 Data Preparation

The data was prepared for model training by:

- Encoding categorical features using one-hot encoding for binary variables and label encoding for others. This step converted categorical data into numerical representations suitable for machine learning algorithms.

```
# One-hot encoding for categorical columns
df = pd.get_dummies(df, columns=['Profession', 'City', 'State', 'Credit_Type', 'Co-Applicant_Credit_Type'], prefix='', prefix_sep='')

# One-hot encoding for binary columns
df = pd.get_dummies(df, columns=['Gender', 'Marital_Status', 'House_Ownership', 'Car_Ownership'], prefix='', prefix_sep='')

# Label encoding for other categorical columns
label_encoder = LabelEncoder()
df['Credit_Score'] = label_encoder.fit_transform(df['Credit_Score'])
```

I have applied one-hot encoding for the specified categorical columns: 'Profession', 'City', 'State', 'Credit_Type', and 'Co-Applicant_Credit_Type'. It creates binary variables for each category within these columns and adds them to the dataFrame. And one-hot encoding for the binary categorical columns: 'Gender', 'Marital_Status', 'House_Ownership', and 'Car_Ownership' creates binary variables for each category within these columns and adds them to the DataFrame.

label_encoder = LabelEncoder(): This part creates an instance of the LabelEncoder class, which was used to encode categorical variables into numerical labels.By using label encoding for 'Credit_Score', each category is assigned a numerical label that preserves the ordinal relationship between the categories. This allows machine learning models to learn from the ordinal

nature of the data and potentially capture meaningful patterns or trends related to credit scores.

- Removing duplicate features, if any, to avoid introducing redundancy into the model.

```
# removing duplicate feature names
df = df.loc[:,~df.columns.duplicated()]
```

This part uses the .loc indexer to select all rows (:) and columns that are not duplicated .

- Replacing special characters in column names to ensure compatibility with model training procedures.

```
# Replacing special characters in column names
df.columns = df.columns.astype(str).str.replace('[^a-zA-Z0-9]', '_', regex=True)
```

This part replaces any special characters in the column names with underscores (_). It first converts the column names to strings using .astype(str), then uses the .str.replace() method with a regular expression (regex=True) to replace any characters that are not letters (lowercase and uppercase) or digits ([^a-zA-Z0-9]) with underscores.

5.2 Model Selection

Several machine learning algorithms were chosen for building models to predict loan risk:

- **Logistic Regression:** This linear model widely used for classification tasks, suitable for predicting binary outcomes like loan risk.
- **Decision Tree:** This is a tree-based model that can capture complex relationships between features, potentially identifying non-linear patterns relevant to loan risk prediction.
- **Random Forest**: This is an ensemble method that combines multiple decision trees to improve accuracy and reduce overfitting.

- **Extra Trees Classifier:** Similar to Random Forest, but uses random splitting at each node, potentially leading to more diverse trees and potentially better performance.
- **XGBoost:** A powerful gradient boosting algorithm known for its efficiency and accuracy in handling various types of data.
- **LightGBM:** Another gradient boosting algorithm with fast training speed and potential effectiveness in classification tasks.
- **CatBoost:** A gradient boosting framework specifically designed for categorical features, potentially leading to improved performance when dealing with categorical data like professions and credit types.

5.3 Model Training

The dataset was split into training and testing sets using a common ratio (80/20 split). The training set was used to fit the models, while the testing set was used to evaluate their performance on unseen data. This split ensures that the models are not simply memorizing the training data but can generalize to new loan applications.

```
# splitting the data into features and target variable
X = df.drop('Risk_Flag', axis=1)
y = df['Risk_Flag']
```

X = df.drop('Risk_Flag', axis=1): This line creates the feature matrix X by dropping the 'Risk_Flag' column from the DataFrame df.

y = df['Risk_Flag']: This line creates the target vector y containing the 'Risk_Flag' column from the DataFrame df.

```
def classify(model, X, y):
    x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    model.fit(x_train, y_train)

    accuracy = model.score(x_test, y_test)
    print('Accuracy:', accuracy)

    score = cross_val_score(model, X, y, cv=5)
    print('CV Score:', np.mean(score))

    return accuracy
```

I have created a function named classify that takes three arguments: the machine learning model (model), the feature matrix (X), and the target vector (y).

Inside the classify function:

x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42): This line splits the dataset into training and testing sets using the train_test_split function. It assigns 80% of the data to the training set *(x_train, y_train)* and 20% to the testing set (x_test, y_test). The random_state=42 argument ensures reproducibility of the split.

model.fit(x_train, y_train): This line fits the machine learning model to the training data.

accuracy = model.score(x_test, y_test): This line calculates the accuracy of the model on the testing data using the score method of the model.

print('Accuracy:', accuracy): This line prints the accuracy score of the model on the testing data.

score = cross_val_score(model, X, y, cv=5): This line calculates the cross-validated accuracy scores using the cross_val_score function with 5-fold cross-validation (cv=5).

print('CV Score:', np.mean(score)): This line prints the mean cross-validated accuracy score.

return accuracy: This line returns the accuracy score of the model on the testing data.

Overall, the classify function trains a machine learning model, evaluates its performance on both testing data and cross-validated data, and returns the accuracy score.

Then I called the classify function within every model. The function trains the model, evaluates its accuracy on the testing data, and returns the accuracy score. Then the accuracy score is assigned to different variable on each model.

5.4 Performance Evaluation

Each model was trained and evaluated using a common split of the data into training and testing sets. The accuracy score and cross-validation score were used as the primary evaluation metrics to assess the model's performance in predicting the target variable (Risk_Flag).

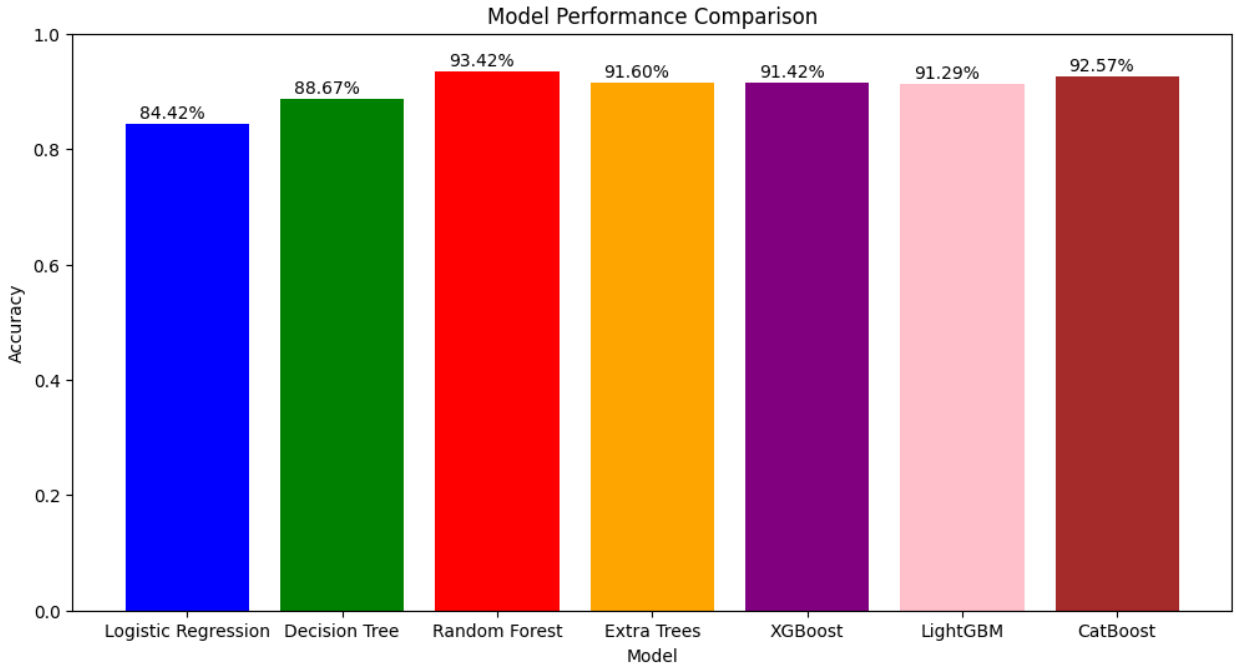
In an attempt to improve model performance, Principal Component Analysis (PCA) was applied to the dataset. PCA is a dimensionality reduction technique that aims to capture the most significant variance in the data while reducing the number of

features. However, after applying PCA, the accuracy of some of the machine learning models decreased. This suggests that the original features, despite potentially containing some redundancy, provided sufficient information for the models to make accurate predictions. Therefore, I decided to use the original features without applying PCA for further analysis and model training.

6. Results and Discussion

6.1 Model Comparison

The performance of the different machine learning models used in this project was evaluated using accuracy and cross-validation scores. The following chart summarizes the results:



As illustrated in the model performance comparison chart, Random Forest achieved the highest accuracy score (93.42%), followed by CatBoost (92.57%) and Extra Trees Classifier model (91.60%). This indicates that Random Forest outperformed the other models in predicting loan risk within this specific dataset.

Several factors might have contributed to Random Forest's effectiveness. Random Forest is an ensemble method that combines multiple decision trees, potentially leading to improved accuracy and reduced overfitting compared to individual decision trees. Additionally, its ability to handle both numerical and categorical features effectively might have been beneficial for this dataset.

6.2 Practical Implications

The findings of this project have practical implications for the financial sector. The insights gained from the data analysis and the best performing model can provide valuable insights for improving credit risk assessment practices. For example, lenders can prioritize factors like income stability, credit score, and job type when evaluating loan applications, potentially reducing the risk. Additionally, the model itself can be used as a decision-making tool, assisting lenders in classifying individuals as low-risk or high-risk based on their financial characteristics.

Furthermore, the insights gained from this project can be used to:

- Develop targeted marketing campaigns towards individuals with lower credit risk, potentially expanding the customer base for lenders.
- Design financial education programs aimed at improving financial literacy and responsible borrowing practices among individuals.
- Advocate for policy changes that promote financial inclusion and responsible lending practices within the financial sector.
- By implementing these practical applications, the findings of this project can contribute to a more responsible and sustainable lending environment, benefiting both financial institutions and individual borrowers.

7. Conclusion and Future Directions

This project successfully built and evaluated various machine learning models for predicting loan risk based on individual financial and personal characteristics. The analysis identified Random Forest as the best performing model, demonstrating its effectiveness in this specific task. Understanding the key features that influence loan risk prediction, such as income stability, credit score, and job type, can guide lenders in making informed decisions and potentially reduce the loan risk.

Future research directions could involve:

- Expanding the dataset to include additional features or data points that might further improve the model's accuracy.
- Exploring alternative machine learning algorithms or ensemble methods that might yield even better performance.
- Incorporating explainable AI techniques to gain deeper insights into the model's decision-making process and identify potential biases.
- Investigating the potential for applying this approach to other financial risk prediction tasks beyond loan risk assessment.

By continuing to refine and improve these models, we can contribute to a more efficient and responsible financial system that benefits both lenders and borrowers.

8. References

Kaggle dataset link: https://www.kaggle.com/datasets/dc04492/loan-prediction

Project Github link:

https://github.com/SifatNiloy/loan-prediction/blob/main/loan_prediction.ipynb