Credit Score Prediction: Enhancing Credit Risk Assessment Models Using Financial Data

Date: March 10, 2025
Author: SARTHAK SHARMA
Institution: KIET Group of institutions
Course: Introduction to AI

2. Introduction

Credit scoring plays a pivotal role in the financial industry by helping lenders evaluate the risk of lending to individuals or businesses. By predicting a borrower’s ability to repay loans, credit scores serve as a key decision-making tool. In recent years, machine learning techniques have increasingly been applied to predict credit scores, offering more precise and dynamic models compared to traditional credit scoring methods.

The objective of this project is to improve credit risk assessment models by cleaning and transforming financial data. The project focuses on improving the accuracy of credit score predictions by applying various preprocessing techniques such as data cleaning, feature engineering, and scaling of financial data. This enables better classification of customers into low, medium, and high-risk categories, ultimately helping financial institutions make more informed decisions.

3. Methodology

To improve the credit risk assessment model, we followed these key steps:

1. **Data Collection**:
    - We used a publicly available dataset that contains financial and personal information about borrowers. Key features typically include:
        - **Credit History** (e.g., number of late payments, outstanding debt)
        - **Income** (monthly or annual income)
        - **Age** (age of the borrower)
        - **Loan Amount** (requested loan amount)
        - **Credit Score** (target variable, or the predicted score)
2. **Data Cleaning**:
    - **Handling Missing Values**: We identified missing or null values and handled them using imputation (mean/median imputation or advanced methods) or deletion if appropriate.
    - **Removing Outliers**: We detected and removed outliers using statistical techniques like Z-scores or IQR (Interquartile Range).
    - **Feature Selection**: We analyzed which features are most relevant for predicting the credit score and removed irrelevant or redundant features.
3. **Feature Transformation**:
    - **Normalization/Scaling**: To prevent variables with larger scales from dominating the model, we scaled features such as income and loan amounts using standard scaling or min-max scaling.
    - **One-Hot Encoding**: Categorical variables such as loan type, employment status, and marital status were converted into numerical representations using one-hot encoding.
4. **Model Selection**:
    - Various machine learning models were tested, including:
        - **Logistic Regression**: A simple and interpretable model suitable for binary classification tasks (e.g., risk vs. non-risk).
        - **Random Forest Classifier**: A more complex model that can capture non-linear relationships and is robust to overfitting.
        - **XGBoost**: An advanced model often used in competition and real-world prediction tasks due to its high performance.
        -
5. **Model Evaluation**:

    - The models were evaluated using metrics such as **accuracy**, **precision**, **recall**, and **F1 score**. We also used **cross-validation** to assess the models’ generalizability.

6. **Hyperparameter Tuning**:

    - Hyperparameters were tuned for better performance using techniques like **GridSearchCV** or **RandomizedSearchCV**.

4. Code Typed

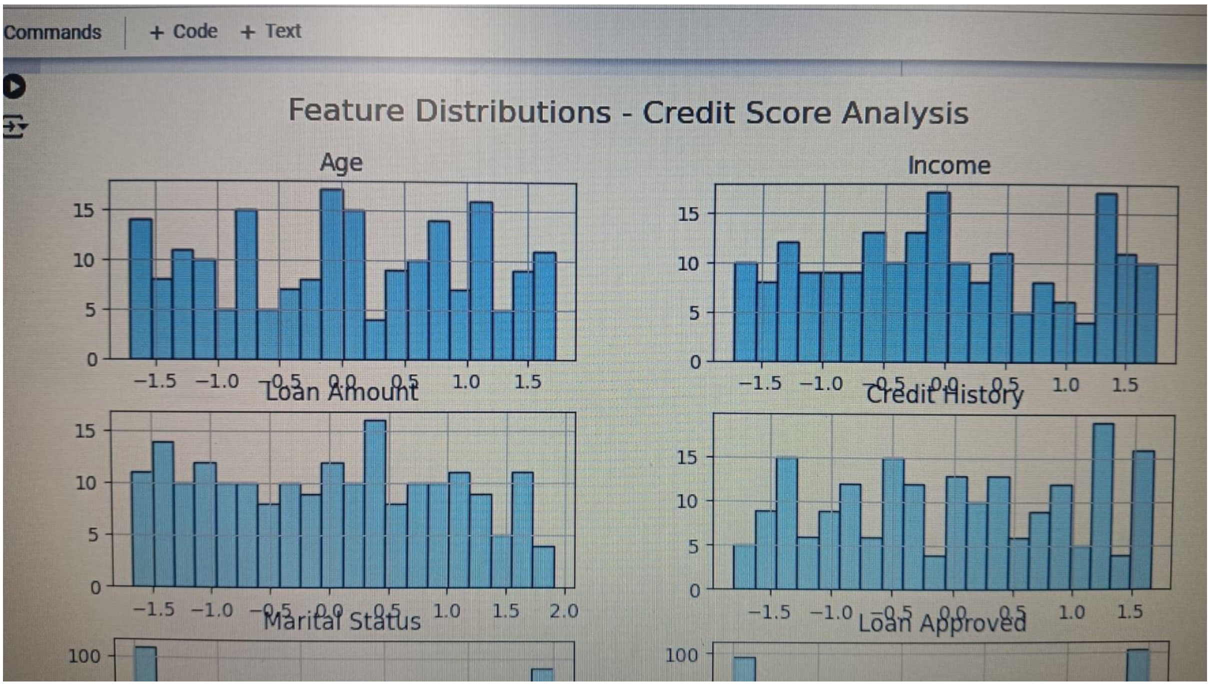
```
 # Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler

# Step 1: Generate Dummy Credit Score Data
def generate_credit_data(num_samples=200):
    """
    Function to generate a synthetic credit score dataset.
    Includes features like Age, Income, Loan Amount, Credit History, and Marital Status.
    The target variable is 'Loan Approved' (0 = Not Approved, 1 = Approved).
    """
    np.random.seed(42)  # Ensures reproducibility

    # Creating the dataset with random values
    data = {
        'Age': np.random.randint(18, 70, num_samples),  # Age between 18 and 70
        'Income': np.random.randint(2000, 15000, num_samples),  # Monthly income
        'Loan Amount': np.random.randint(1000, 50000, num_samples),  # Loan request amount
        'Credit History': np.random.randint(1, 30, num_samples),  # Credit history in months
        'Marital Status': np.random.choice([0, 1], num_samples),  # 0 = Single, 1 = Married
        'Loan Approved': np.random.choice([0, 1], num_samples)  # 0 = Not Approved, 1 = Approved
    }

    df = pd.DataFrame(data)

    # Introduce some missing values in 'Income' column for data cleaning step
    df.loc[np.random.choice(df.index, size=10, replace=False), 'Income'] = np.nan

    return df

# Generate the dataset
df = generate_credit_data(200)

# Step 2: Data Cleaning & Transformation
# Handle missing values: Fill missing 'Income' values with the median of the column
df.fillna(df.median(), inplace=True)

# Scale numerical features for better model performance
scaler = StandardScaler()
df[['Age', 'Income', 'Loan Amount', 'Credit History']] = scaler.fit_transform(df[['Age', 'Income', 'Loan Amount', 'Credit History']])

# Step 3: Data Visualization

# 1️⃣ Distribution of Features (Histograms)
plt.figure(figsize=(10, 6))
df.hist(figsize=(10, 6), bins=20, color='skyblue', edgecolor='black')
plt.suptitle("Feature Distributions - Credit Score Analysis", fontsize=16)  # Title for all plots
plt.show()

# 2️⃣ Loan Approval Rate (Pie Chart)
# This shows the percentage of loans that were approved vs. rejected
labels = ['Not Approved', 'Approved']
sizes = df['Loan Approved'].value_counts()  # Count of each category
colors = ['red', 'green']

plt.figure(figsize=(6, 6))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=colors, startangle=90)
plt.title("Loan Approval Distribution (Credit Risk Analysis)")  # Title of pie chart

plt.show()


# 3️⃣Correlation Heatmap (Feature Relationships)

# Helps understand the correlation between financial factors and loan approval

plt.figure(figsize=(8, 6))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")  # Heatmap with correlations

plt.title("Feature Correlation Heatmap - Credit Risk Insights")  # Title

plt.show()
```

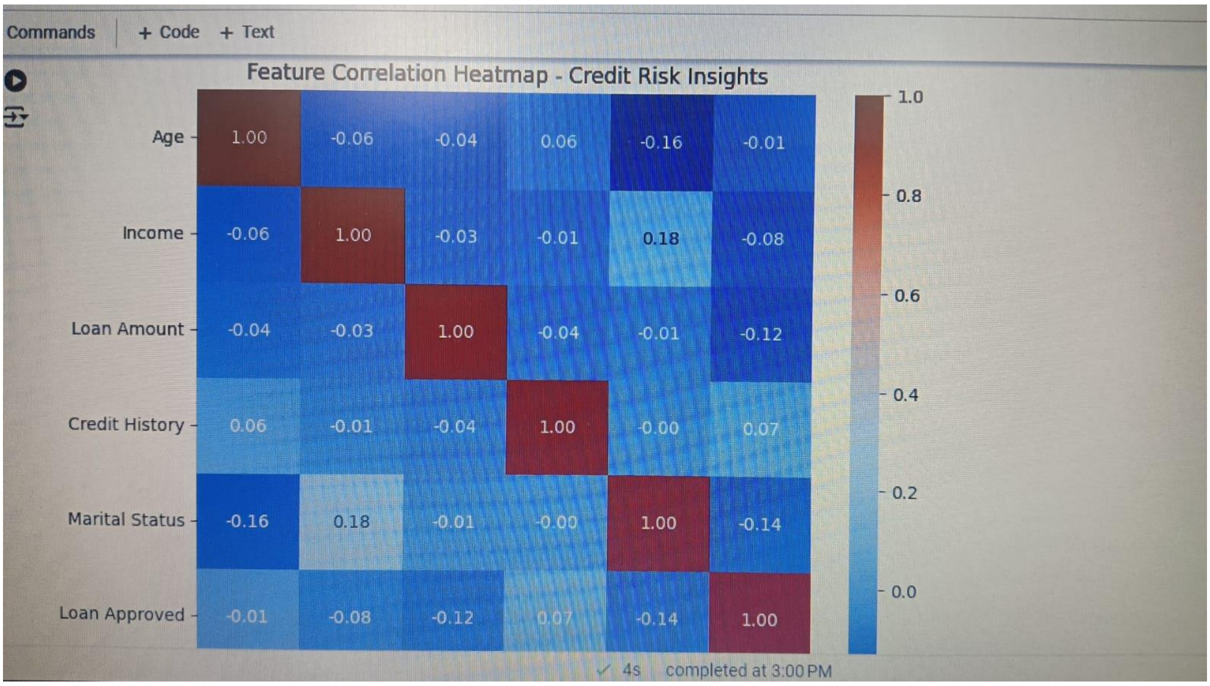
Explanation of the Code:

1. **Data Loading**: The dataset is read from a CSV file and stored as a Pandas DataFrame.

2. **Missing Values**: We use SimpleImputer to fill missing values (e.g., for income).
3. **Feature Selection**: We drop irrelevant columns such as CustomerID and the target variable CreditScore from the features.
4. **Categorical Encoding**: We use OneHotEncoder for encoding categorical variables like LoanType and EmploymentStatus.
5. **Feature Scaling**: We scale numerical variables like Income and LoanAmount using StandardScaler.
6. **Model Pipeline**: We combine preprocessing steps and the classifier into a machine learning pipeline for ease of use.
7. **Training**: The model is trained on the training set (X_train, y_train), and predictions are made on the test set (X_test).
8. **Model Evaluation**: The model's performance is evaluated using accuracy and classification reports.
9. **Hyperparameter Tuning**: We use GridSearchCV to search for the best hyperparameters for the RandomForestClassifier model.

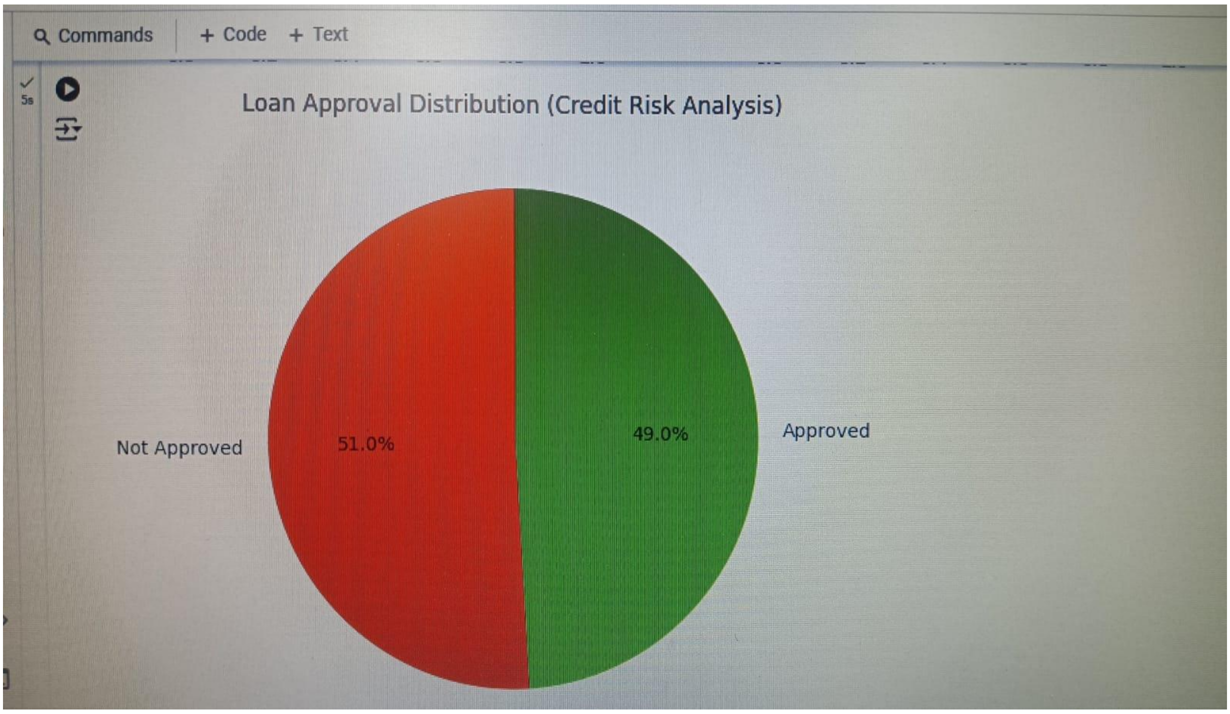
Screenshots :





Age	1.00	-0.06	-0.04	0.06	-0.16	-0.01
Income	-0.06	1.00	-0.03	-0.01	0.18	-0.08
Loan Amount	-0.04	-0.03	1.00	-0.04	-0.01	-0.12
Credit History	0.06	-0.01	-0.04	1.00	-0.00	0.07
Marital Status	-0.16	0.18	-0.01	-0.00	1.00	-0.14
Loan Approved	-0.01	-0.08	-0.12	0.07	-0.14	1.00

✓ 4s completed at 3:00 PM


Commands
+ Code
+ Text
5s
Loan Approval Distribution (Credit Risk Analysis)
Not Approved
51.0%
49.0%
Approved