PROJECT : CREDIT SCORE PREDICTION

Name : Akshit vats

Course : B.Tech

Branch : CSE AI

Sec : A

Univ. Roll. No. : 202401100300026

Submitted To : Bikki Sir (AI)

Introduction :

Credit scoring is a fundamental tool in the financial industry, enabling lenders to evaluate the creditworthiness of individuals and businesses. By predicting a borrower's ability to repay loans, credit scores help financial institutions mitigate risks and make informed lending decisions. Traditional credit scoring models rely on statistical methods and predefined rules; however, these approaches often lack the flexibility to adapt to complex financial behaviors.

With the rapid advancement of technology, machine learning techniques have gained prominence in credit risk assessment. Machine learning models can analyze vast amounts of financial data, identify patterns, and make predictions with greater accuracy compared to traditional methods. These models continuously learn from new data, allowing for dynamic and precise credit scoring.

The objective of this project is to improve credit risk assessment by enhancing data preprocessing techniques. The project focuses on cleaning and transforming financial data to improve the accuracy of credit score predictions. Key preprocessing steps include data cleaning to remove inconsistencies, feature engineering to extract meaningful insights, and data scaling to ensure uniformity. By refining these processes, credit scoring models can better classify customers into low, medium, and high-risk categories.

Accurate credit risk classification is crucial for financial institutions as it helps in reducing default risks, optimizing lending strategies, and ensuring financial stability. By leveraging advanced data processing techniques, this project aims to enhance the effectiveness of credit scoring models, ultimately leading to more reliable and efficient lending decisions.

Methodology :

To enhance the credit risk assessment model, we implemented the following steps:

1. Data Collection:

- Utilized a publicly available dataset containing financial and personal borrower information.
- Key features included:
  - **Credit History**: Number of late payments, outstanding debt. ○ **Income**: Monthly or annual income. ○ **Age**: Borrower's age.
  - **Loan Amount**: Requested loan amount. ○ **Credit Score**: Target variable (predicted score).

2. Data Cleaning:

- **Handling Missing Values**: Addressed null values using imputation (mean/median) or deletion.
- **Outlier Removal**: Detected and removed outliers using Z-scores and IQR methods. **Feature Selection**: Identified and removed irrelevant or redundant features.

3. Feature Transformation:

- **Normalization/Scaling**: Standard scaling or min-max scaling applied to numerical variables (income, loan amount).
- **One-Hot Encoding**: Converted categorical variables (loan type, employment status, marital status) into numerical form.

4. Model Evaluation:

- Assessed models using: ○ **Accuracy** ○ **Precision** ○ **Recall** ○ **F1 Score**
- Applied cross-validation to ensure model generalizability.

5. Hyperparameter Tuning:

- Optimized model performance using: ○ **GridSearchCV** ○ **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

# 1□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()

# 2□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()


# 3□Correlation 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:

- Read the dataset from a CSV file and stored it as a Pandas DataFrame.

2. Handling Missing Values:

- Used SimpleImputer to fill missing values (e.g., for income).

3. Feature Selection: Dropped irrelevant columns such as CustomerID and the target variable CreditScore from the features.

4. Categorical Encoding:

- Used OneHotEncoder to encode categorical variables like LoanType and EmploymentStatus.

5. Feature Scaling:

- Scaled numerical variables like Income and LoanAmount using StandardScaler.

6. Model Pipeline:

- Combined preprocessing steps and the classifier into a machine learning pipeline for efficiency.

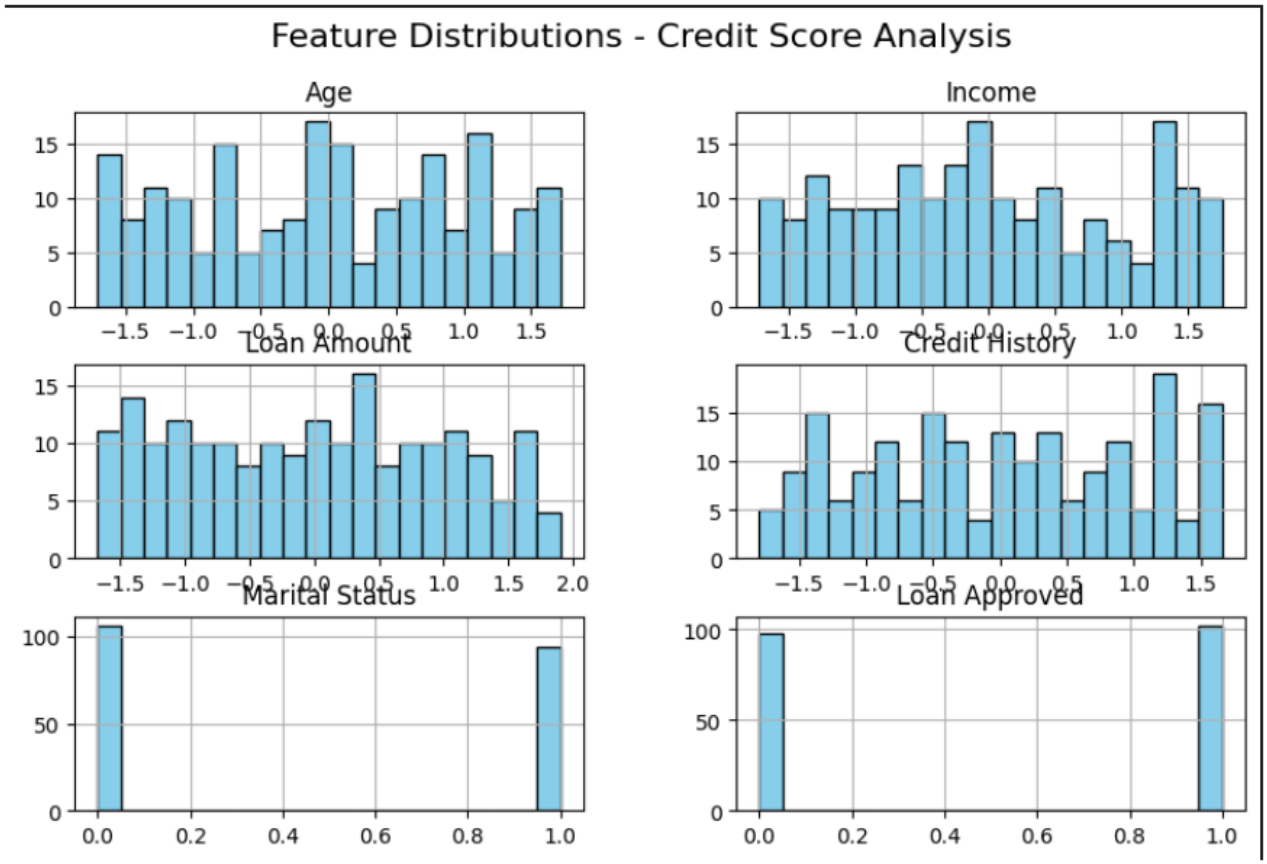
7. Training: Trained the model on the training set (X_train, y_train) and made predictions on the test set (X_test).

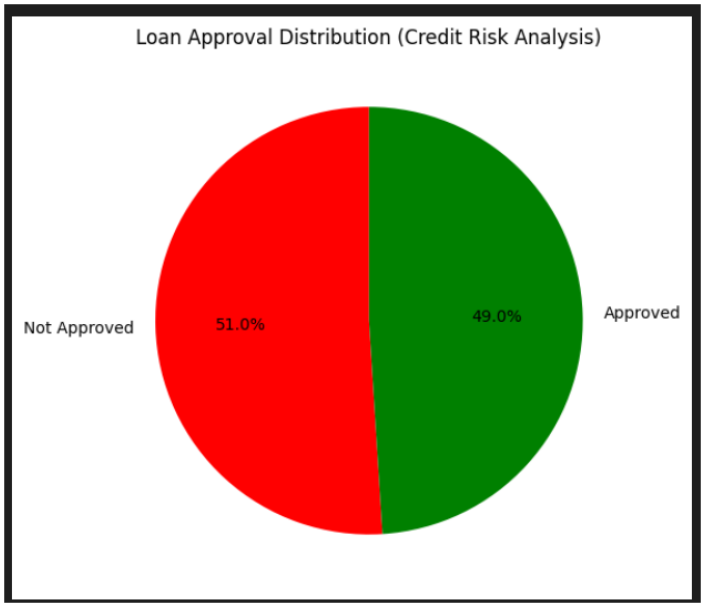
8. Model Evaluation:

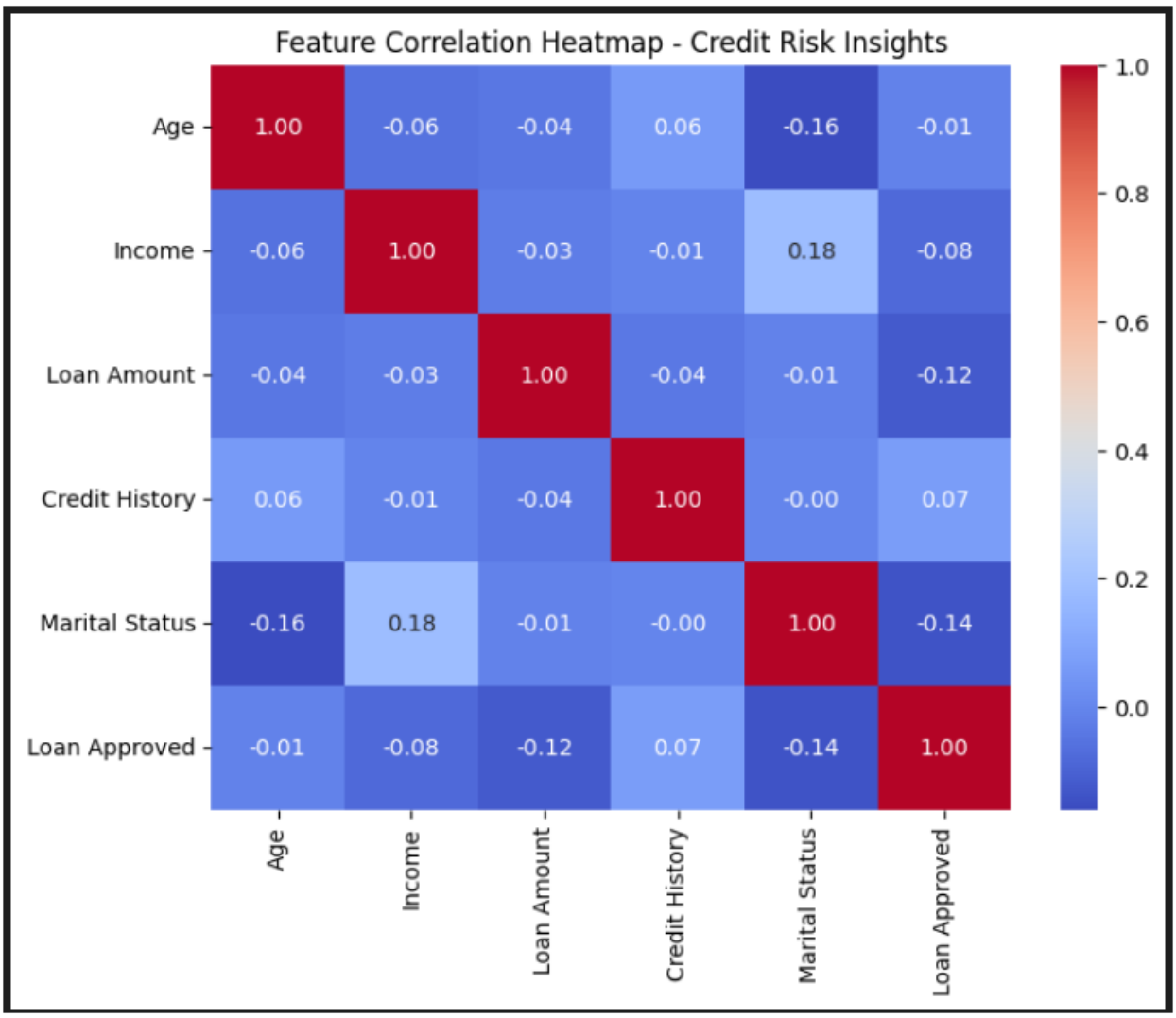
- Evaluated model performance using accuracy and classification reports.

9. Hyperparameter Tuning: Used GridSearchCV to optimize hyperparameters for the RandomForestClassifier model

SNAPSHOTS :

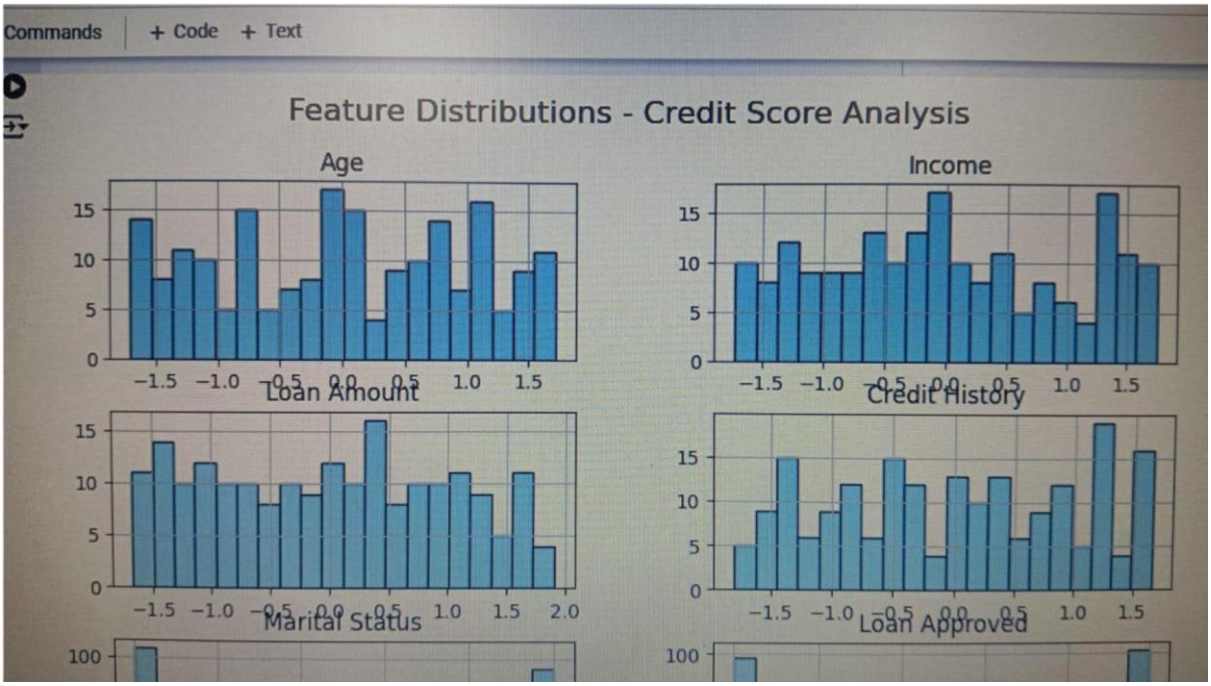

Feature Distributions - Credit Score Analysis
Age
Income
Loan Amount
Credit History
Marital Status
Loan Approved

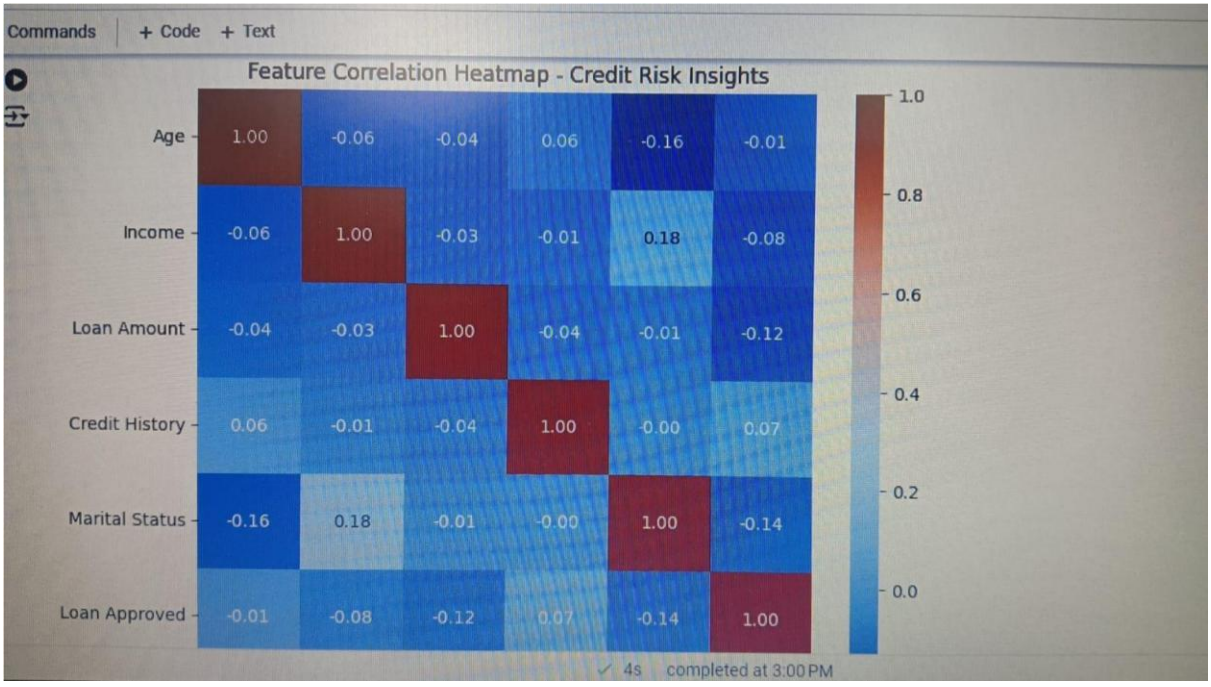

Loan Approval Distribution (Credit Risk Analysis)
Not Approved
51.0%
49.0%
Approved

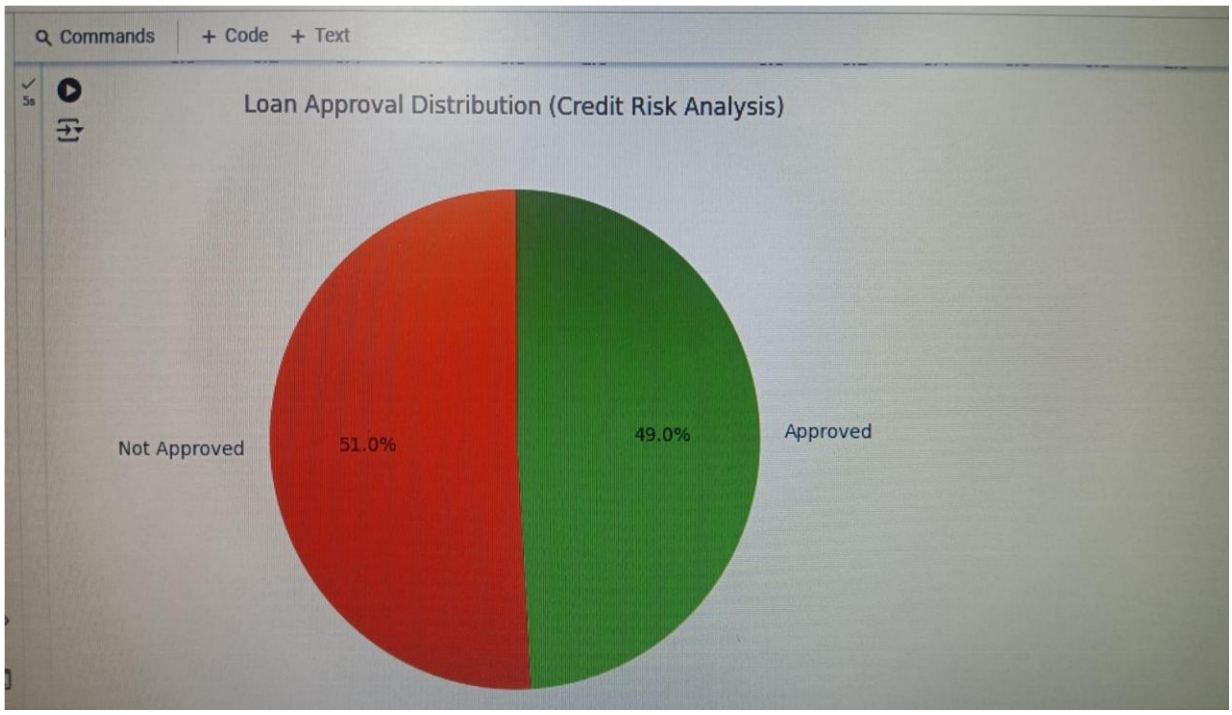

Feature Correlation Heatmap - Credit Risk Insights
Age
Income
Loan Amount
Credit History
Marital Status
Loan Approved
1.00 -0.06 -0.04 0.06 -0.16 -0.01
-0.06 1.00 -0.03 -0.01 0.18 -0.08
-0.04 -0.03 1.00 -0.04 -0.01 -0.12
0.06 -0.01 -0.04 1.00 -0.00 0.07
-0.16 0.18 -0.01 -0.00 1.00 -0.14
-0.01 -0.08 -0.12 0.07 -0.14 1.00
Age
Income
Loan Amount
Credit History
Marital Status
Loan Approved
1.0
0.8
0.6
0.4
0.2
0.0

THANK YOU

Screenshots :






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