CSE 587: Data Intensive Computing, Fall 2024
Instructor Prof. Naeem Maroof

Project Phase I - Report

Credit Risk Analysis for Loan Approval

Appendix:

Task 1	Title Problem statement Background Significance Impact
Task 2	Data source Data description
Task 3	Data cleaning and processing tasks - 1)Detecting missing values 2)Detecting outliers 3)Handling duplicate rows 4)Handling missing data points and outliers 5)Normalization 6)Label encoding
Task 4	EDA tasks - 1)Data types, unique values 2)Data description 3) Duplicate rows 4)Outliers visualization plots 5)Univariate Visualization plots 6)Bivariate Visualization plots 7)Multivariate Visualization plot 8)Distribution of target variable (loan_status) 9)Correlation matrix
References	

Team Information:

TEAMMATE_1: Shantam Srivastava - 50604167 - ss693@buffalo.edu
TEAMMATE_2: Saloni Kamlesh Redij - 50611349 - salonika@buffalo.edu
TEAMMATE_3: Manas Peshwe - 50592351 - mpeshwe@buffalo.edu
TEAMMATE_4: Naman Mawandia - 50610535 - namanmaw@buffalo.edu

Task 1

Title Credit risk analysis for loan approval

Problem Statement

Credit risk analysis is important for financial institutions to reduce loss and ensure a well-managed, profitable loan profile. The problem is to build an accurate and dependable model for financial institutes to predict loan defaults using historical data. Financial institutes usually rely on traditional methods to analyse the creditworthiness of a candidate. The traditional methods often overlook the complex patterns in customer behavior that leads to poor laon approval decisions. Therefore, this project seeks to answer the following questions:

1. How can we effectively evaluate credit risk using machine learning models?
2. How to ensure fair lending practices without compromising risk assessment standards?
3. When optimizing loan approval processes, how can financial institutions reduce default rates and maintain profitability?

Background

Loan defaults can have a significant impact on a financial institution's stability and profitability. Traditional credit risk assessment models, which primarily focus on credit scores, income, and employment history, may fail to identify the complex patterns in customer behavior. Rapidly evolving economic conditions and shifting consumer behaviors demand more dynamic and adaptive risk assessment models. The rise of big data and machine learning provides us with an opportunity to improve risk prediction by employing advanced techniques and incorporating a wide range of features.

Significance

With better identification of high-risk applicants, financial institutions can lower default rates and minimize loss. Implementing advanced models will enhance the decision-making in loan processes. This data-driven approach will allow for more precise customer segmentation and targeted marketing. Big data and machine

learning approaches will lead to improved customer experience due to faster decisions aided by predictions.

Impact

This project will identify and extend loans to a customer segment previously overlooked by conventional models, resulting in an influx of new borrowers and increased business for financial institutions. Overall this project will lead to reduced systemic risk, improved credit allocation, and inclusive and responsible lending practices through the use of advanced data-driven models.

Task 2

Data source

We sourced our data which is available on Kaggle here.

About the data

The dataset contains 32,581 samples 12 columns/features. The datatypes of these features are as below:

Numerical data types(int, float): Income, age, loan amount, employment length, interest rate, credit history, percent of loan income

Categorical features: Homeownership, loan intent, loan grade, previous default.

Feature information

- person_age: Applicant's age in years.
- person_income: Annual income of the applicant in USD.
- person_home_ownership: Homeownership Status (e.g., Rent, Own, Mortgage).
- person_emp_length: Length of employment in years.
- loan_intent: Purpose of the loan (e.g., Education, Medical, Personal).
- loan_grade: Risk grade assigned to the loan, assessing the applicant's creditworthiness.
- loan_amnt: Total loan amount requested by the applicant.
- loan_int_rate: Interest rate associated with the loan.
- loan_status: The approval status of the loan (approved or not approved).

- loan_percent_income: Percentage of the applicant's income allocated towards loan repayment.
- cb_person_default_on_file: Indicates if the applicant has a history of default ('Y' for yes, 'N' for no).
- cb_person_cred_hist_length: Length of the applicant's credit history in years.

Task 3

Data cleaning and processing are crucial steps as they ensure that the data is accurate and free from errors and inconsistencies. These processes improve the performance of the model and prevent bias leading to reliable and better predictions overall.

Data cleaning and processing steps employed on our dataset for credit risk analysis:

1) Detecting Missing values

```
person_age                      0
person_income                   0
person_home_ownership           0
person_emp_length             895
loan_intent                     0
loan_grade                      0
loan_amnt                       0
loan_int_rate                3116
loan_status                     0
loan_percent_income             0
cb_person_default_on_file       0
cb_person_cred_hist_length      0
dtype: int64
```

Although most of the features in our dataset were free from missing or null values we found two such features person_emp_length and loan_int_rate that had missing or null values. Out of which loan_int_rate had a significantly higher number of missing values that needed to be addressed.

2) Outliers Detection

For the data, we have implemented z_scores for outlier detection.
With a threshold of 4, all the detected outlier points are replaced with N/A. The state of the data after we implement this process is as below:

```
NaN values :
person_age                    183
person_income                 126
person_home_ownership           0
person_emp_length             960
loan_intent                     0
loan_grade                      0
loan_amnt                     184
loan_int_rate                3095
loan_status                     0
loan_percent_income            60
cb_person_default_on_file       0
cb_person_cred_hist_length    171
dtype: int64
```

3) Handling duplicate rows

Through our EDA processes, we found multiple records of data that were duplicated. These rows were dropped in our data cleaning step using the drop_duplicates() method. The below snapshot shows the number of records before and after dropping the duplicate records.

```
Before dropping duplicate rows: (32581, 12)
After dropping duplicate rows: (32416, 12)
```

4)Handling Missing data and outliers

we have employed a KNN imputer with the value for number of neighbors set to 4 to handle the missing and outlier values in our dataset.

5) Normalization

Normalization is the preprocessing technique, which is used to rescale the numerical non-target data columns. This makes our final data columns value in the range of [0,1]. Normalization is done on the dataset so that our model does not learn the actual numerical data, but the pattern in the whole data. In our project, we have applied a min-max normalization technique, which brings our data in a range of [0,1]. The point to keep in mind while applying this technique is that we should not normalize our target feature.

	person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status	l
0	0.060606	0.177994	RENT	0.178571	PERSONAL	D	0.334548	0.595506	1	
1	0.030303	0.018123	OWN	0.238095	EDUCATION	B	0.014577	0.321348	0	
2	0.151515	0.018123	MORTGAGE	0.047619	MEDICAL	C	0.145773	0.418539	1	
3	0.090909	0.199029	RENT	0.190476	MEDICAL	C	0.421283	0.551124	1	
4	0.121212	0.163107	RENT	0.380952	MEDICAL	C	0.437318	0.497191	1	

6) Label encoding

Label encoding is a data preprocessing technique that is used to convert categorical data into numerical data, thus allowing us to apply Machine Learning techniques to such data. Applying encoding also helps in finding the correlation of categorical data with our target feature. In our project we have applied label encoding on columns: {person_home_ownership, loan_intent, loan_grade,cb_person_default_on_file}.

›n_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb
0.177994	3	0.178571	4	3	0.334548	0.595506	1	1.000000	
0.018123	2	0.238095	1	1	0.014577	0.321348	0	0.169492	
0.018123	0	0.047619	3	2	0.145773	0.418539	1	0.966102	
0.199029	3	0.190476	3	2	0.421283	0.551124	1	0.898305	
0.163107	3	0.380952	3	2	0.437318	0.497191	1	0.932203	

Task 4

EDA

Exploratory data analysis (EDA) is used to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.

In our project we have implemented multiple EDA methods like analyzing the datatypes of features, understanding the unique values, checking duplicate rows in the data to later remove them, visualizing the outliers in the data using scatterplots, using multiple histograms and bar graphs to understand the distribution of data using univariate and bivariate analysis. We have plotted boxplots and violin plots on the Loan status feature to visualize the intensity of records for that feature. In total we have plotted ~10 visualization graphs to analyze our data and the correlation between the features.

1) Data types, Unique values

The snapshot below shows the different features and their datatypes.

```
person_age                    int64
person_income                 int64
person_home_ownership        object
person_emp_length           float64
loan_intent                  object
loan_grade                   object
loan_amnt                     int64
loan_int_rate               float64
loan_status                   int64
loan_percent_income         float64
cb_person_default_on_file    object
cb_person_cred_hist_length    int64
dtype: object
```

The following snapshot shows all the unique values in our categorical features.

```
person_home_ownership
['RENT' 'OWN' 'MORTGAGE' 'OTHER']
loan_intent
['PERSONAL' 'EDUCATION' 'MEDICAL' 'VENTURE' 'HOMEIMPROVEMENT'
 'DEBTCONSOLIDATION']
loan_grade
['D' 'B' 'C' 'A' 'E' 'F' 'G']
cb_person_default_on_file
['Y' 'N']
```

2) Data description

The following snapshot shows the summary of non-categorical data.

	person_age	person_income	person_emp_length	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_cred_hist_length
count	32581.000000	3.258100e+04	31686.000000	32581.000000	29465.000000	32581.000000	32581.000000	32581.000000
mean	27.734600	6.607485e+04	4.789686	9589.371106	11.011695	0.218164	0.170203	5.804211
std	6.348078	6.198312e+04	4.142630	6322.086646	3.240459	0.413006	0.106782	4.055001
min	20.000000	4.000000e+03	0.000000	500.000000	5.420000	0.000000	0.000000	2.000000
25%	23.000000	3.850000e+04	2.000000	5000.000000	7.900000	0.000000	0.090000	3.000000
50%	26.000000	5.500000e+04	4.000000	8000.000000	10.990000	0.000000	0.150000	4.000000
75%	30.000000	7.920000e+04	7.000000	12200.000000	13.470000	0.000000	0.230000	8.000000
max	144.000000	6.000000e+06	123.000000	35000.000000	23.220000	1.000000	0.830000	30.000000

3) Duplicate rows

During our analysis, we found our dataset to have a significant number of duplicate records. In the following snapshot we are presenting is a small subset of records with duplicate values.

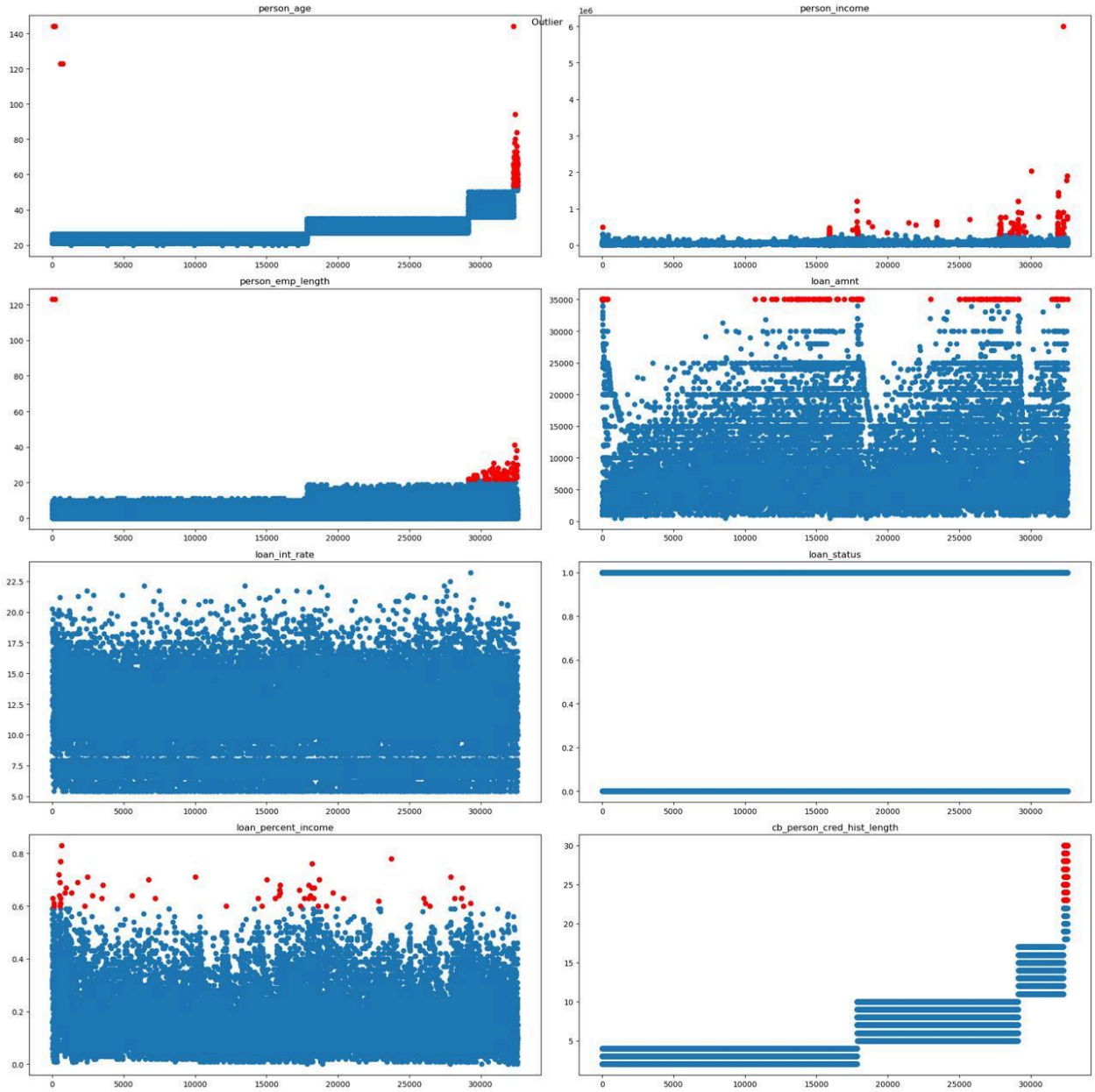
	person_age	person_income	person_home_ownership	person_emp_length	\
15944	21	8088	RENT	NaN	
16835	21	8088	RENT	NaN	
2431	21	15600	RENT	0.0	
17758	21	15600	RENT	0.0	
2498	21	18000	RENT	0.0	
...	...	...	...	...	
32010	42	39996	MORTGAGE	2.0	
29484	43	11340	RENT	4.0	
32279	43	11340	RENT	4.0	
31676	49	120000	MORTGAGE	12.0	
32172	49	120000	MORTGAGE	12.0	

The total number of such duplicate records was 330 which were later dropped as a part of the data cleaning process.

4) Outlier Visualization

To visualize the outlier data points detected in the previous step of Data cleaning and processing we have plotted scatter plots on multiple features to understand the distribution of the data.

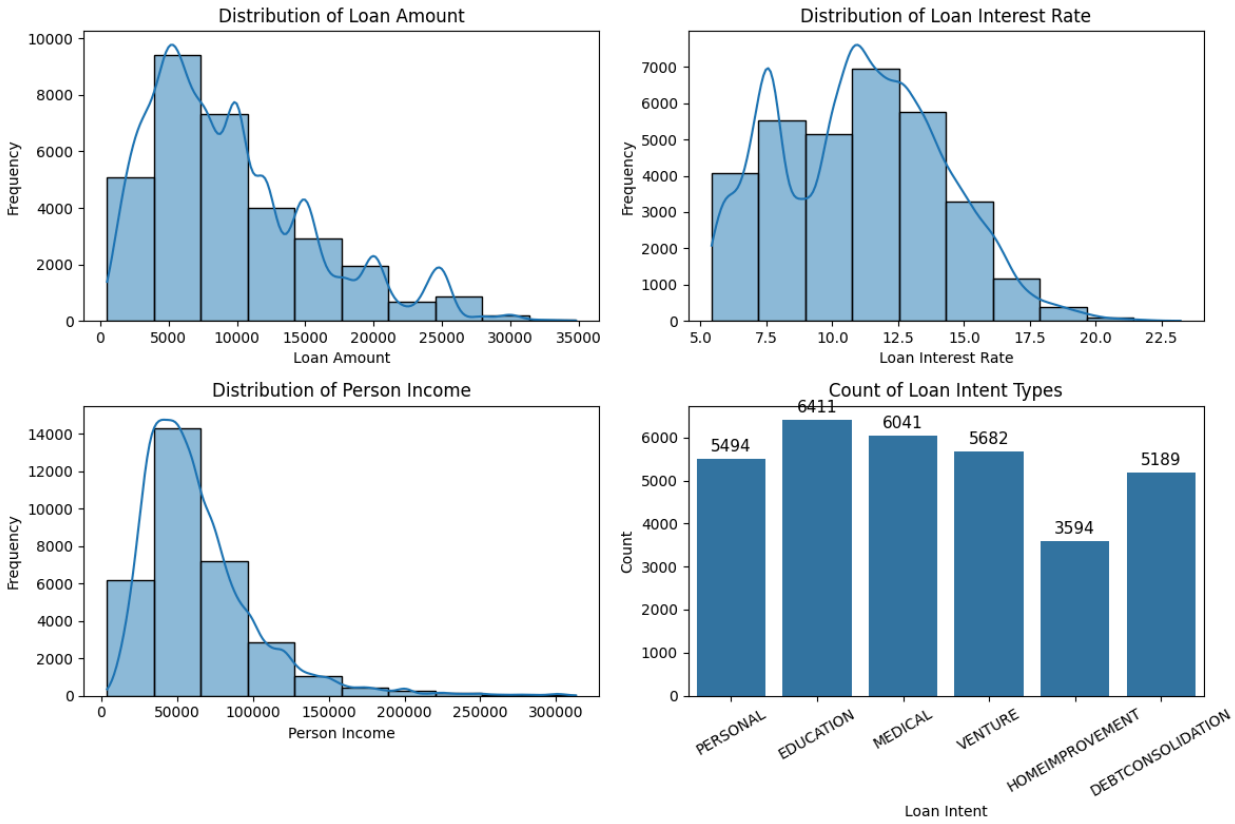
Plot 1 has plot of all the features (columns) of the dataset showing outliers(in red) in the initial data.



Plot1

5) Univariate visualization

Univariate visualizations are used to understand the distribution of a single variable in the data. For our dataset, we have plotted multiple histograms for different features, viz Loan Amount, Loan Interest Rate, Person Income, and Loan Intent.



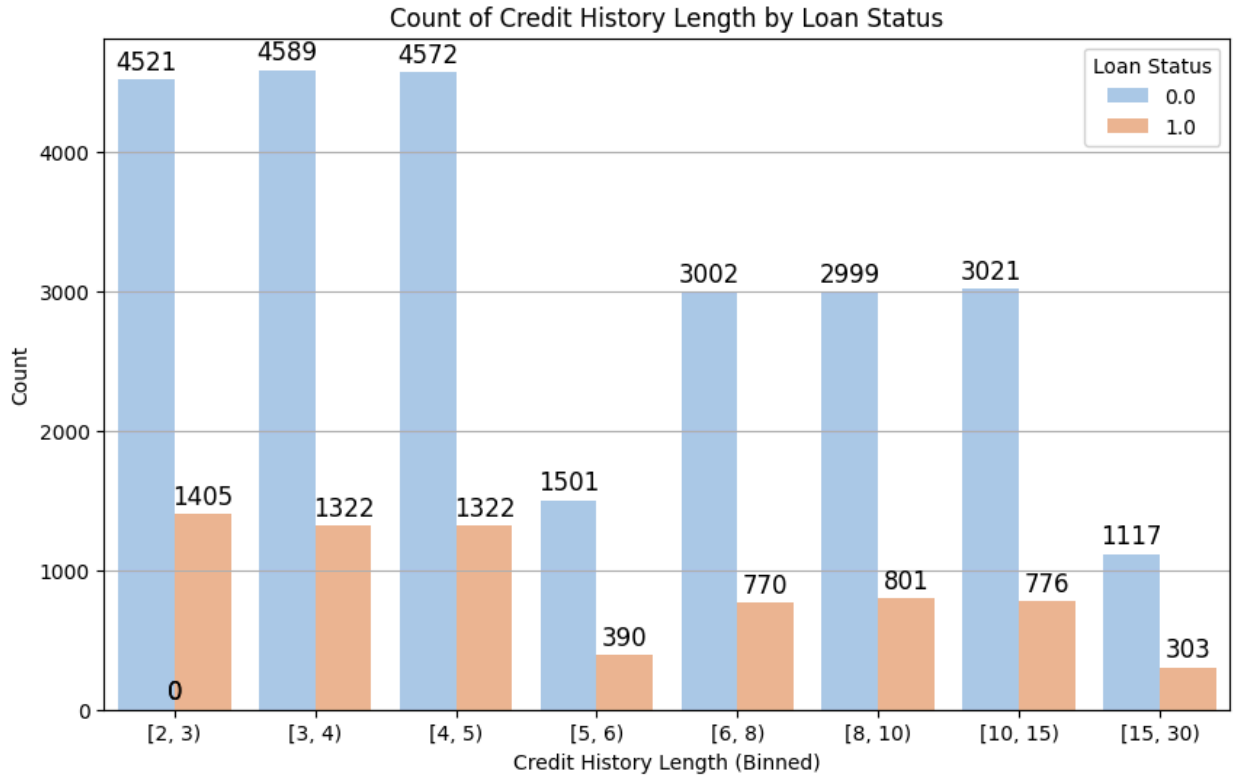
Plot2

SubPlot 1: Loan amount distribution shows that there are the highest number of records for a loan amount of 5000.
SubPlot 2: Loan interest rate distribution shows that the most number of loan records are for an interest rate of 12.5
SubPlot 3: Person income distribution shows that the highest person income in this data is 300,000 but the significant size of the population has a person income of 50,000.
SubPlot 4: Loan intentdistribution shows that the intent is distributed well and there is no polarity of any specific value although the highest is for the intent of Education for a 6411 records.

6) Bivariate visualization

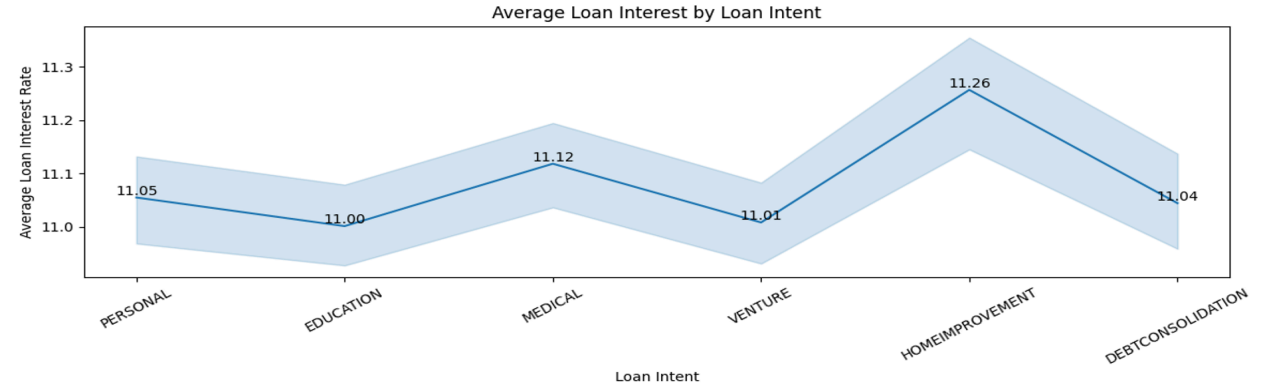
Bivariate graphs display the relationship between two variables. The type of graph will depend on the measurement level of each variable (categorical or quantitative)

Plot 3 shows the frequency of people with the credit history length and loan status.



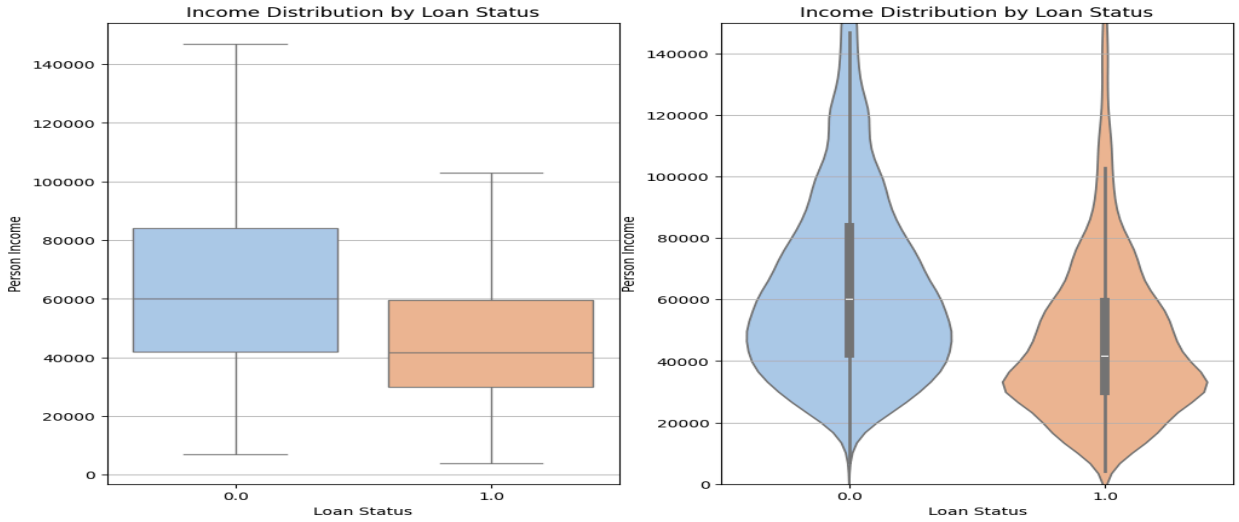
Plot3

Plot 4 shows Average Loan Intent in a line graph.



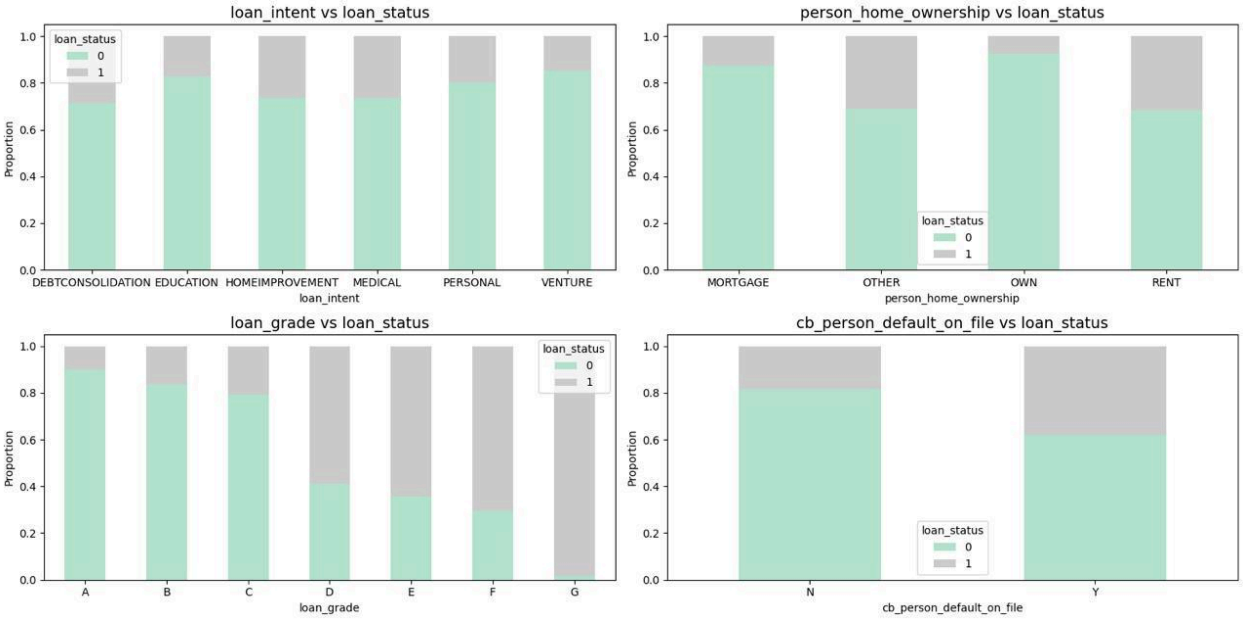
Plot 4

Plot 5, shows 2 subplots each showing a person's income with loan status. The first subplot is a box plot and the other is a violin plot which shows the density of the data also.



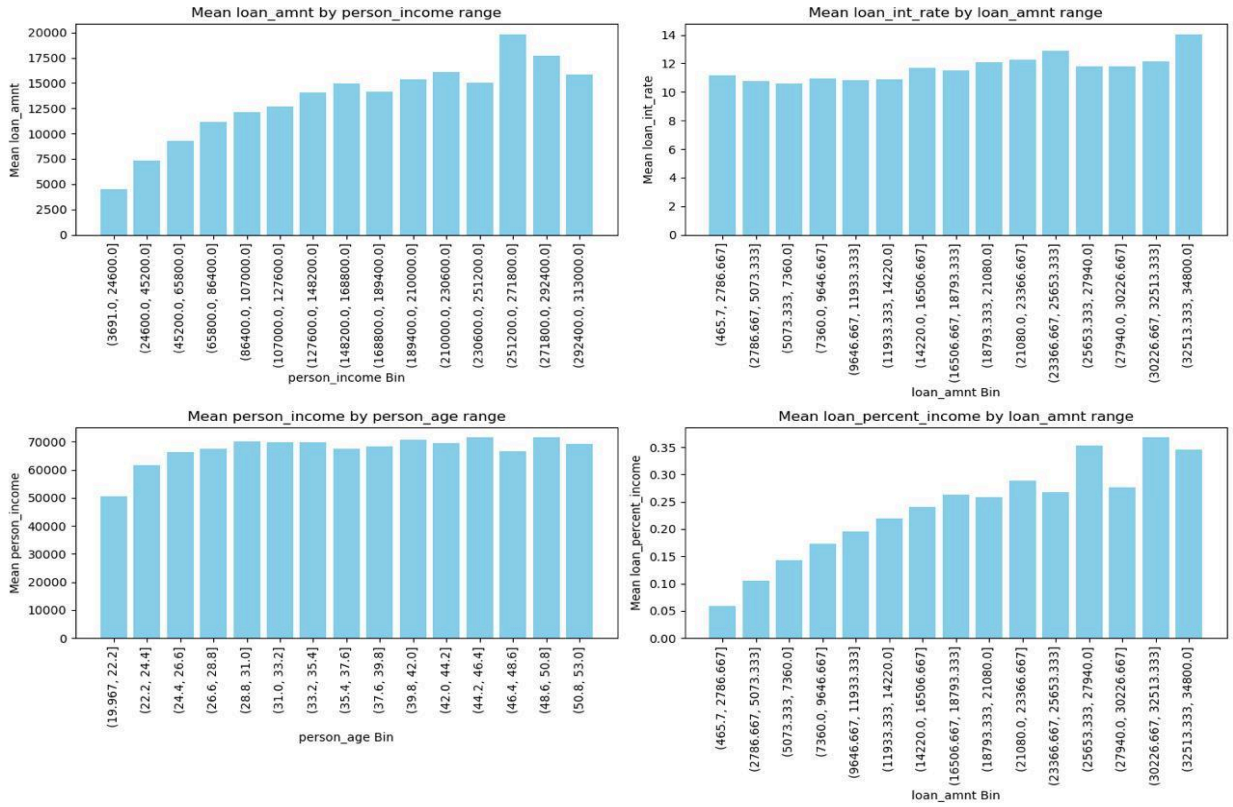
Plot5

Plot 6, shows 4 subplots of proportion with different categorical data such as loan intent, person home ownership, loan grade, and history of default.



Plot6

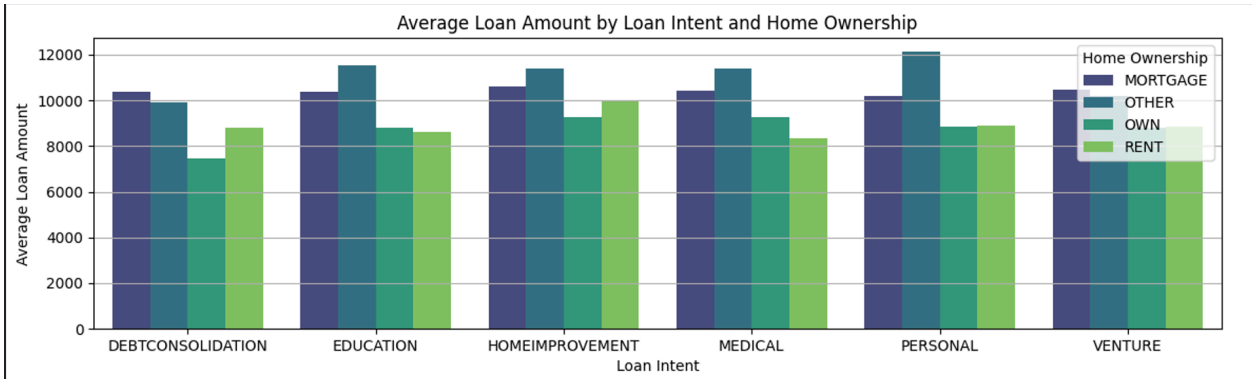
Plot 7, shows 4 bar subplots. It shows the binning of different continuous data vs. other features' mean.



Plot7

7) Multivariate visualization

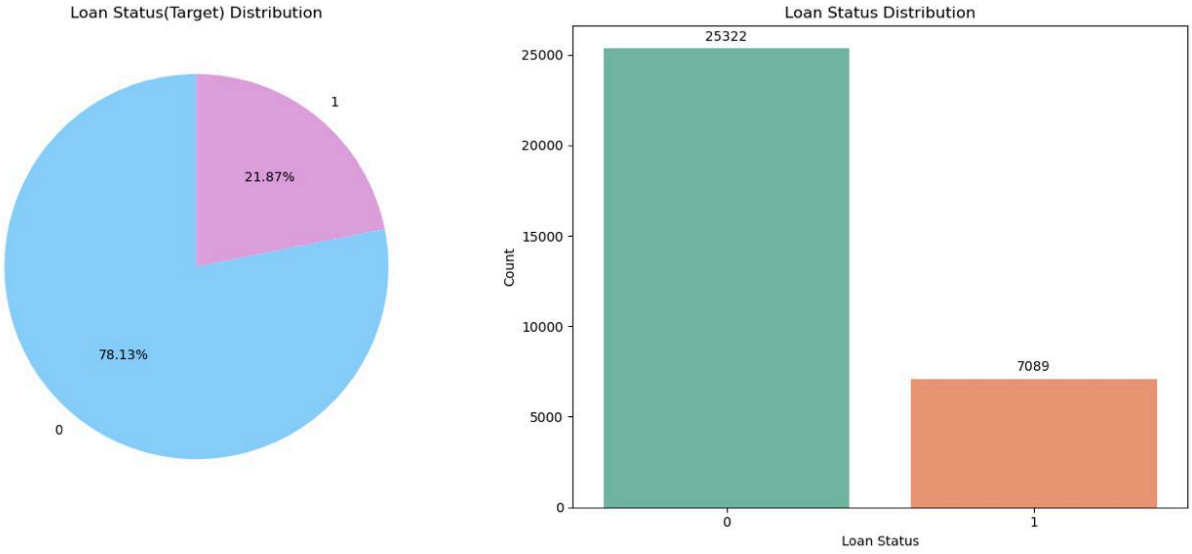
This plot shows the relationship between 3 features: {loan_intent, loan_amnt, person_home_ownership}. From the plot below, we can observe that Mortgage home ownership has a very consistent loan amount, and Rent has a lower side loan amount.



Plot 8

8) Distribution of target variable (loan_status)

While we are visualizing our other non target features and their correlation, target feature distribution should also be visualized. Here, we have target feature loan_status and below are the plots to see the distribution of it using a bar graph and pie chart. We can clearly see that we have more data for the rejection of loan status than the approval, which shows some real world scenario.

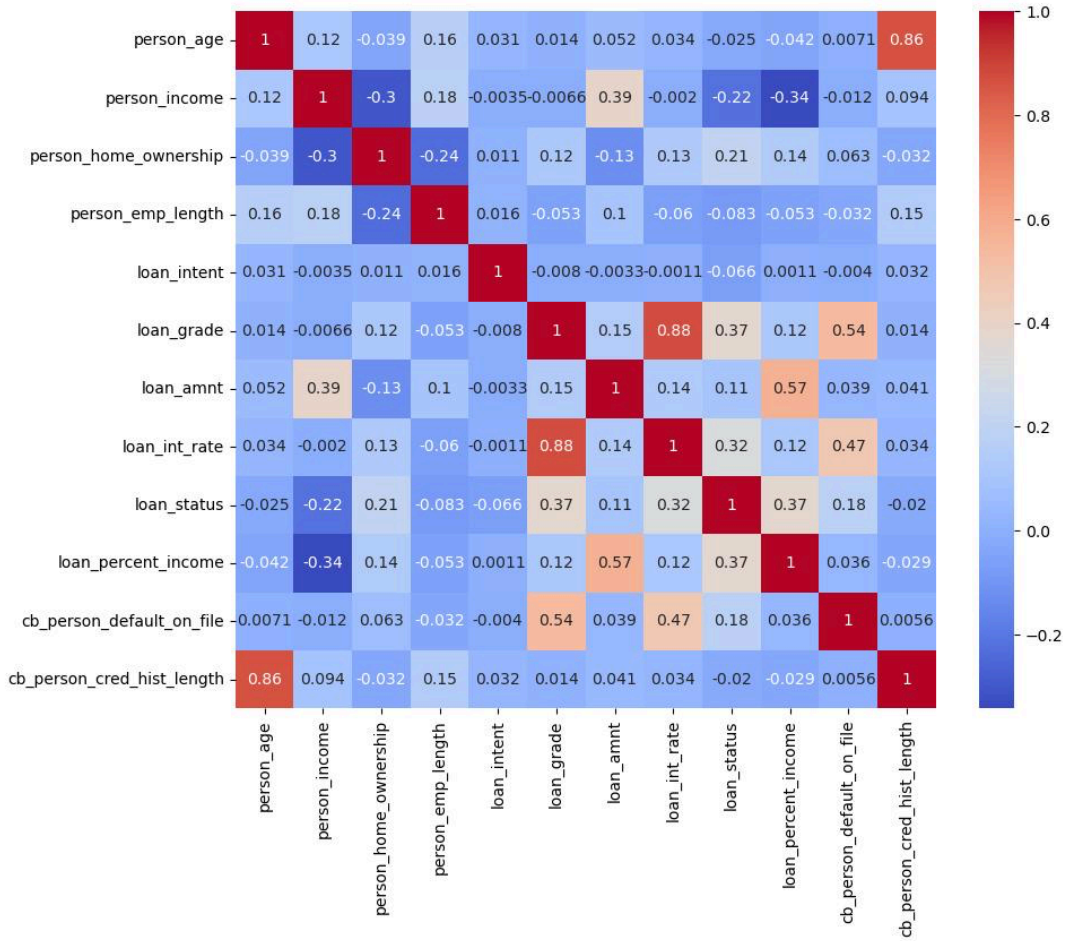


Plot 9

9) Correlation Matrix Visualized

A correlation matrix is a table showing correlation coefficients between variables in which each cell in the table represents a correlation between two features. Correlation matrix is a great tool to quickly understand the patterns in the data.

Plot 10, shows the correlation matrix between the features of our dataset. This will help us in our future steps to decide which features will be best to train our model and which one's can be dropped.


person_age 1 0.12 -0.039 0.16 0.031 0.014 0.052 0.034 -0.025 -0.042 0.0071 0.86
person_income 0.12 1 -0.3 0.18 -0.0035 -0.0066 0.39 -0.002 -0.22 -0.34 -0.012 0.094
person_home_ownership -0.039 -0.3 1 -0.24 0.011 0.12 -0.13 0.13 0.21 0.14 0.063 -0.032
person_emp_length 0.16 0.18 -0.24 1 0.016 -0.053 0.1 -0.06 -0.083 -0.053 -0.032 0.15
loan_intent 0.031 -0.0035 0.011 0.016 1 -0.008 -0.0033 -0.0011 -0.066 0.0011 -0.004 0.032
loan_grade 0.014 -0.0066 0.12 -0.053 -0.008 1 0.15 0.88 0.37 0.12 0.54 0.014
loan_amnt 0.052 0.39 -0.13 0.1 -0.0033 0.15 1 0.14 0.11 0.57 0.039 0.041
loan_int_rate 0.034 -0.002 0.13 -0.06 -0.0011 0.88 0.14 1 0.32 0.12 0.47 0.034
loan_status -0.025 -0.22 0.21 -0.083 -0.066 0.37 0.11 0.32 1 0.37 0.18 -0.02
loan_percent_income -0.042 -0.34 0.14 -0.053 0.0011 0.12 0.57 0.12 0.37 1 0.036 -0.029
cb_person_default_on_file -0.0071 -0.012 0.063 -0.032 -0.004 0.54 0.039 0.47 0.18 0.036 1 0.0056
cb_person_cred_hist_length 0.86 0.094 -0.032 0.15 0.032 0.014 0.041 0.034 -0.02 -0.029 0.0056 1
person_age
person_income
person_home_ownership
person_emp_length
loan_intent
loan_grade
loan_amnt
loan_int_rate
loan_status
loan_percent_income
cb_person_default_on_file
cb_person_cred_hist_length
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Plot10

References:

- NumPy: https://numpy.org/doc/stable/
- SciPy: https://docs.scipy.org/doc/scipy/
- Matplotlib: https://matplotlib.org/stable/contents.html
- Scikit-learn: https://scikit-learn.org/stable/documentation.html
- Seaborn: https://seaborn.pydata.org/
- Pandas: https://pandas.pydata.org/pandas-docs/stable/
- https://bookdown.org/max/FES/visualizations-for-numeric-data-exploring-train-ridership-data.html
- https://rkabacoff.github.io/datavis/Bivariate.html
- https://www.ibm.com/topics/exploratory-data-analysis
- https://www.displayr.com/what-is-a-correlation-matrix/#

CSE 587: Data Intensive Computing, Fall 2024
Instructor Prof. Naeem Maroof

Project Phase II - Report

Credit Risk Analysis for Loan Approval

Appendix:

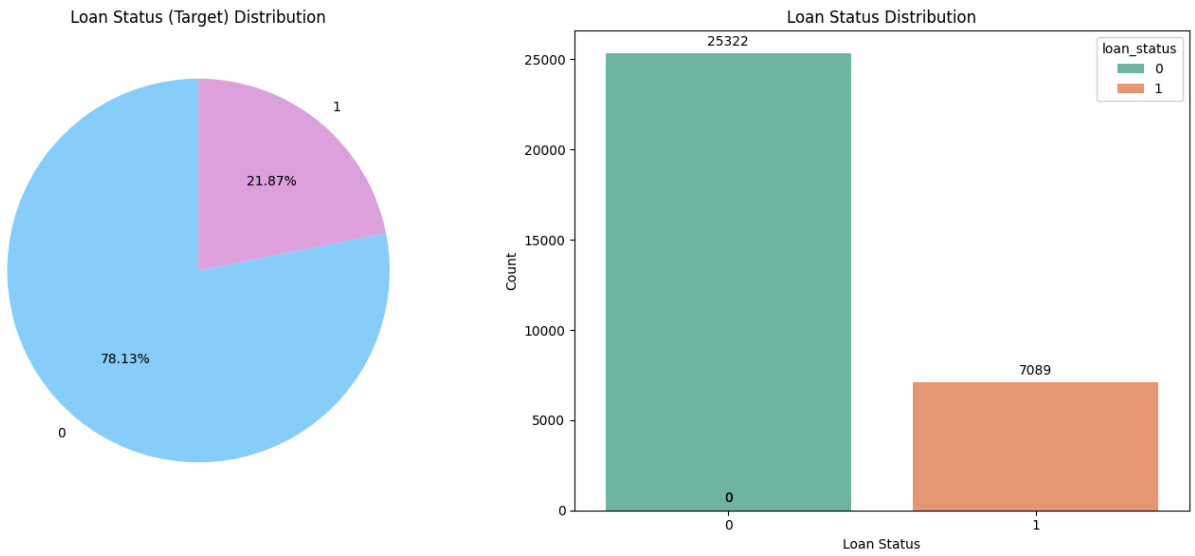
Data Preparation(continued)	1. Handling class imbalance 2. Preparing data for model training and evaluation
ML algorithms, Model training, and evaluations	1. Logistic Regression 2. Decision Tree Classifier 3. XGBoost 4. Support Vector Classifier 5. Random Forest Classifier
	Comparison of all algorithms Conclusion
References	

Team Information:

TEAMMATE_1: Shantam Srivastava - 50604167 - ss693@buffalo.edu
TEAMMATE_2: Saloni Kamlesh Redij - 50611349 - salonika@buffalo.edu
TEAMMATE_3: Manas Peshwe - 50592351 - mpeshwe@buffalo.edu
TEAMMATE_4: Naman Mawandia - 50610535 - namanmaw@buffalo.edu

Data preparation(continued)

1. Handling class imbalance in ‘Target’



Distribution of classes in target variable before applying SDG

Why is it important to handle class imbalance and the impact of class imbalance on model training?

The above visualizations highlight a significant imbalance in the dataset, where rejected loans dominate the target variable. Models trained on this dataset without addressing the imbalance may become biased toward the majority class (rejected loans), leading to poor performance in identifying the minority class (approved loans).

To mitigate this issue, techniques such as synthetic data generation (e.g., with autoencoders), oversampling (e.g., SMOTE), or algorithm-specific class weighting can be used.

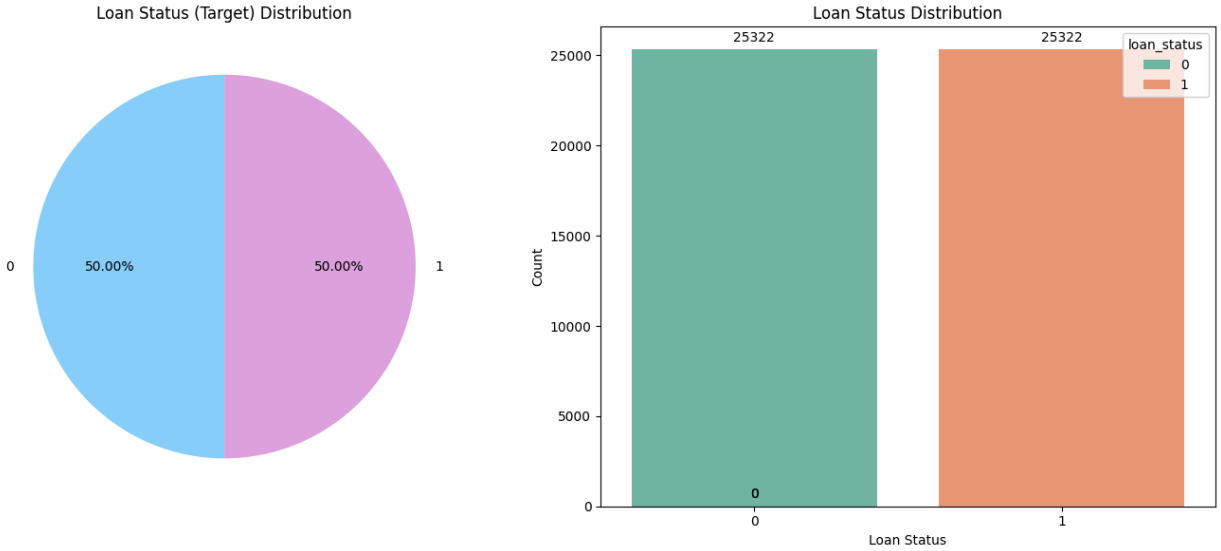
For our project, we used Synthetic data generation using autoencoders to handle the skewed data.

Synthetic data augmentation was applied to address the class imbalance, where rejected loans vastly outnumbered approved loans. Techniques such as autoencoders were employed to generate additional examples for the minority class. This significantly improves model performance, particularly for Logistic Regression and Decision Trees.

Advantage:

- Boosts recall scores across models by reducing false negatives.
- Enhances the robustness of models by exposing them to more diverse data points.

Results after applying Synthetic Data Generation



Distribution of classes in target variable after applying SDG

2. Preparing the data for model training and evaluation

Train: 80% of the processed data

Test: Remaining 20% of the data

ML Algorithms, Model Training and Evaluation

Since our problem is a binary classification task, we chose five machine learning algorithms to train and evaluate our models.

The algorithms implemented in Phase 2 of the project include:

1. Logistic Regression
2. Decision Tree Classifier
3. Support Vector Classifier
4. Random Forest Classifier
5. XGBoost

Algorithm 1: Logistic Regression

Logistic Regression is well-suited for binary classification tasks like predicting loan approval outcomes. Hence, it served as the base model for our project due to its simplicity and interpretability. However, its reliance on linear decision boundaries limits its ability to capture complex patterns in the data.

Advantage:

- simplicity and interpretability.
- Computationally efficient

Limitations:

- Limited ability to capture complex patterns in non-linear data.
- Sensitive to outliers

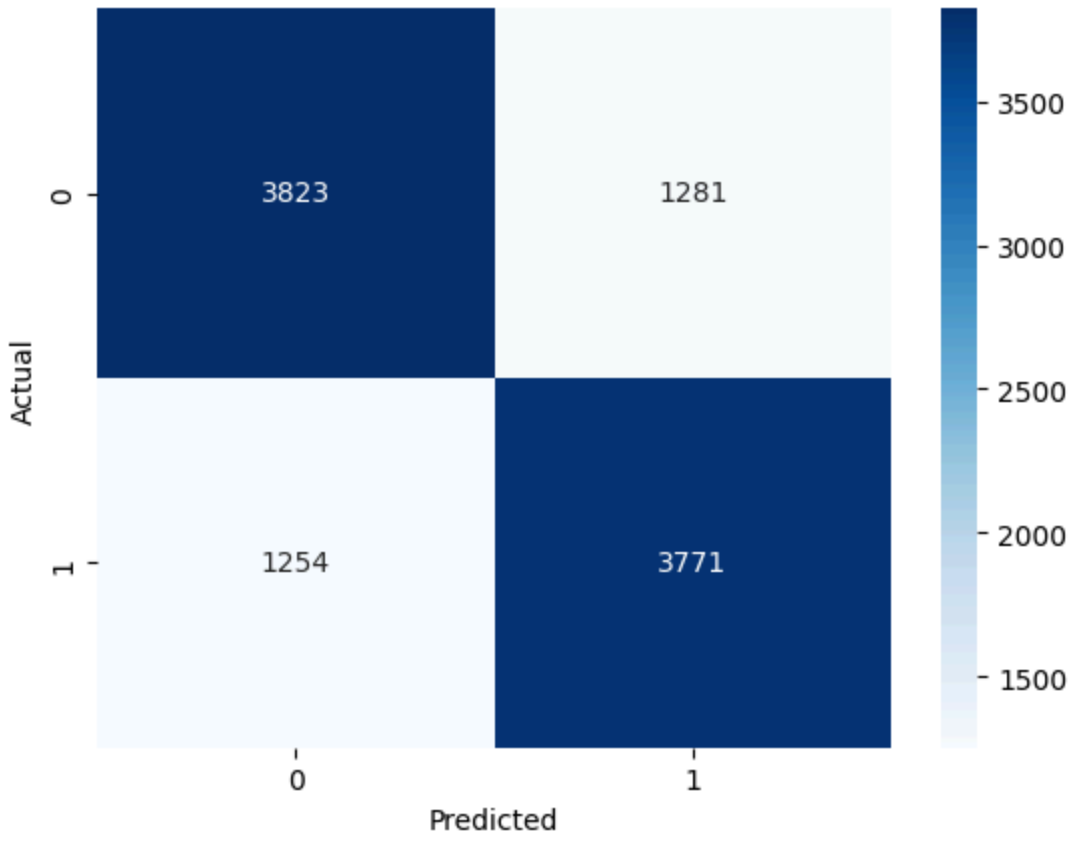
- Tends to overfit

Performance Metrics:

Accuracy: 0.75

	precision	recall	f1-score
0	0.76	0.75	0.76
1	0.75	0.76	0.75
accuracy			0.75
macro avg	0.75	0.75	0.75
weighted avg	0.75	0.75	0.75

The model achieved a training accuracy of just 75%, which is relatively low. With a recall score of 0.75, the model misses 25% of actually “approved” loans, and a precision of 0.75 suggests that 25% of loans predicted as “approved” were actually not eligible. These shortcomings can pose significant risk for a critical task like loan approval, thus Logistic Regression is not suitable for this binary classification on the given dataset.



Algorithm 2: Decision Tree Classifier

Unlike Logistic Regression, the Decision Tree model can handle non-linear patterns and provides interpretable decision paths, this makes it useful for understanding feature contributions.

Key Advantages:

- Offers clear decision paths that can help explain predictions.

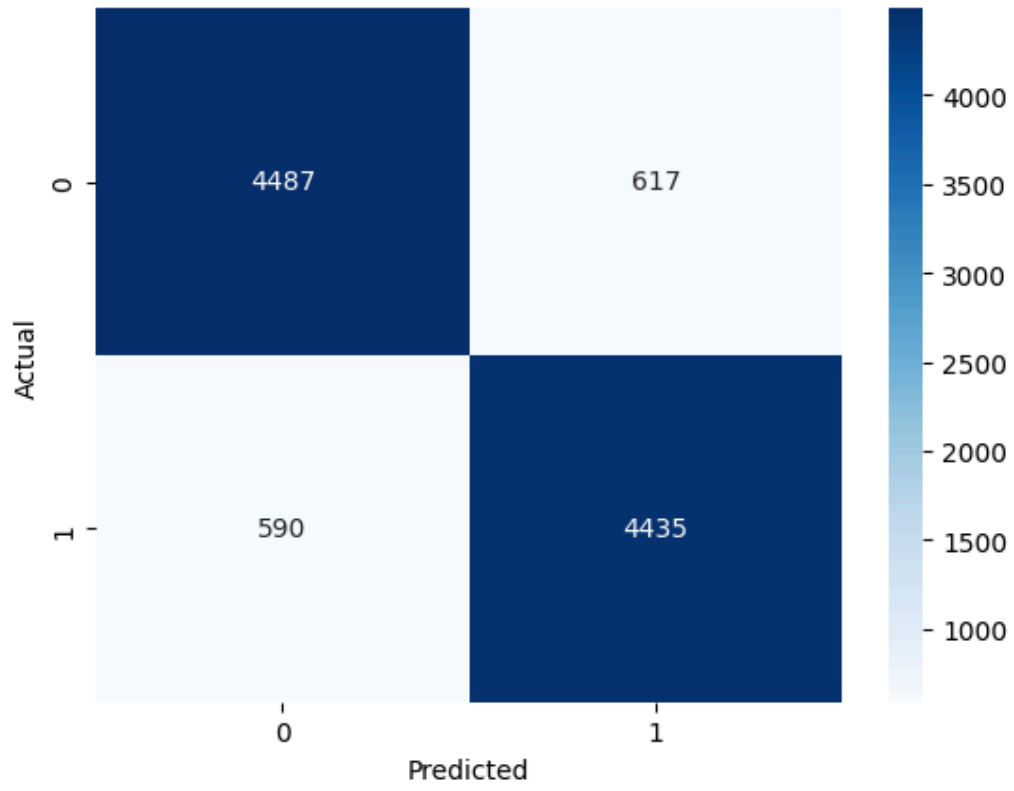
Limitations:

- Highly prone to overfitting, particularly when trained on imbalanced datasets.

Performance Metrics:

Accuracy: 0.89

	precision	recall	f1-score
0	0.89	0.89	0.89
1	0.89	0.89	0.89
accuracy			0.89
macro avg	0.89	0.89	0.89
weighted avg	0.89	0.89	0.89



The model achieved a training accuracy of 89%, with a recall of 0.89, indicating that the Decision Tree model successfully identified a larger proportion of

approved loans as compared to Logistic Regression. However, some overfitting was detected, which could limit its generalizability and impact performance on new data.

Algorithm 3: XGBoost

XGBoost is an advanced implementation of gradient boosting that enhances model accuracy and efficiency. It incorporates regularization techniques (L1 and L2) to reduce overfitting, while also utilizing parallel processing and efficient memory management for faster computation. These optimizations make XGBoost more accurate and computationally efficient.

Key Advantages:

- Excels in capturing complex relationships.
- Handles imbalanced datasets effectively with appropriate parameter tuning.

Limitations:

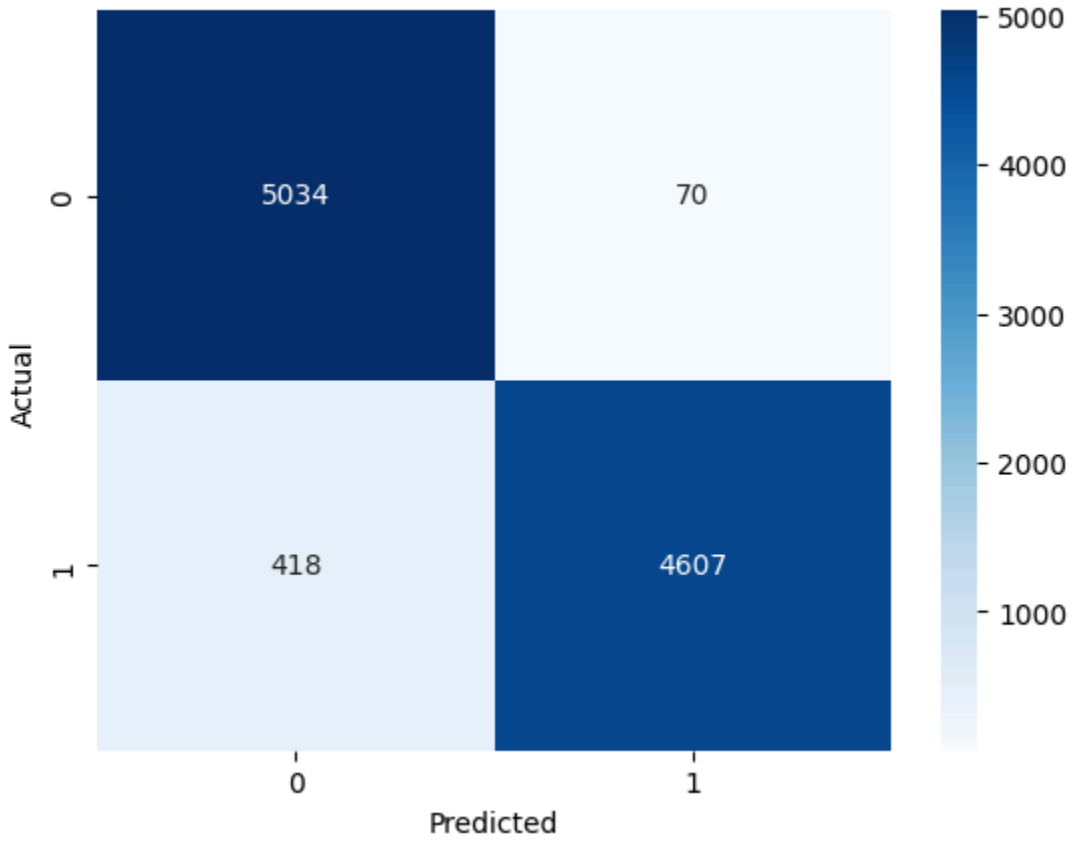
- Computationally intensive.
- Requires careful hyperparameter tuning to prevent overfitting.

Performance Metrics:

Accuracy: 0.95

	precision	recall	f1-score
0	0.92	0.99	0.95
1	0.99	0.92	0.95
accuracy			0.95
macro avg	0.95	0.95	0.95
weighted avg	0.95	0.95	0.95

Gave us the best performance of all the models that we tested, balancing both precision and recall effectively and also being robust against overfitting due to the L1 and L2 regularization techniques.



Algorithm 4: Support Vector Classifier

Support Vector Classifier is robust against overfitting for small to medium datasets. It can handle non-linear relationships with the use of kernels by maximizing the separation between classes.

Key Advantages:

- Capable of handling non-linear decision boundaries using kernels
- Robust against overfitting in small to medium datasets.

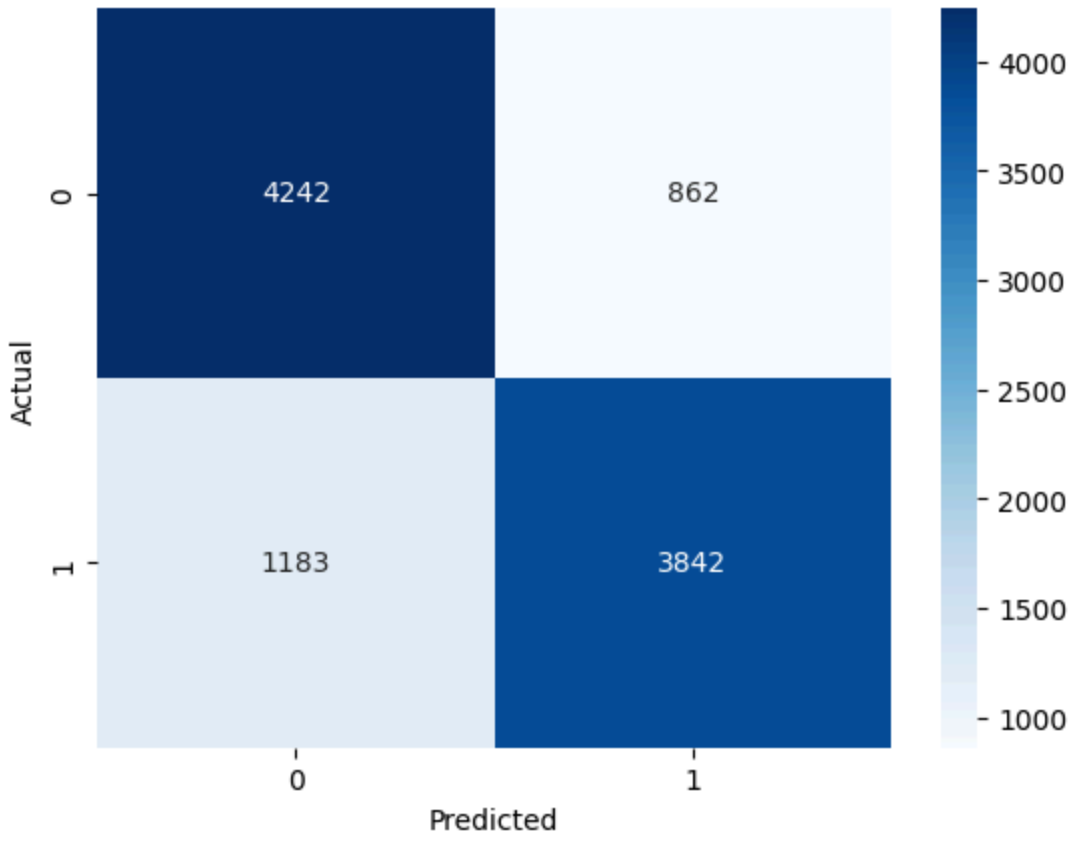
Limitations:

- Computationally expensive for large datasets.
- Less effective on imbalanced datasets without class weighting or other adjustments.

Performance Metrics:

Accuracy: 0.80

	precision	recall	f1-score
0	0.79	0.83	0.81
1	0.82	0.77	0.79
accuracy			0.80
macro avg	0.80	0.80	0.80
weighted avg	0.80	0.80	0.80



SVC performed moderately well, but its recall of 0.80 (averaged over both classes) suggests it missed more approved loans compared to XGBoost. While it demonstrated good precision, the overall computational cost and lower recall made it less favorable for this application.

Algorithm 5: Random Forest Classifier

Random Forest employs multiple decision trees to improve stability and predictive power. Aggregating results mitigate the overfitting seen in standalone decision trees.

Key Advantages:

- Reduces overfitting by averaging results across trees.
- Offers feature importance rankings.

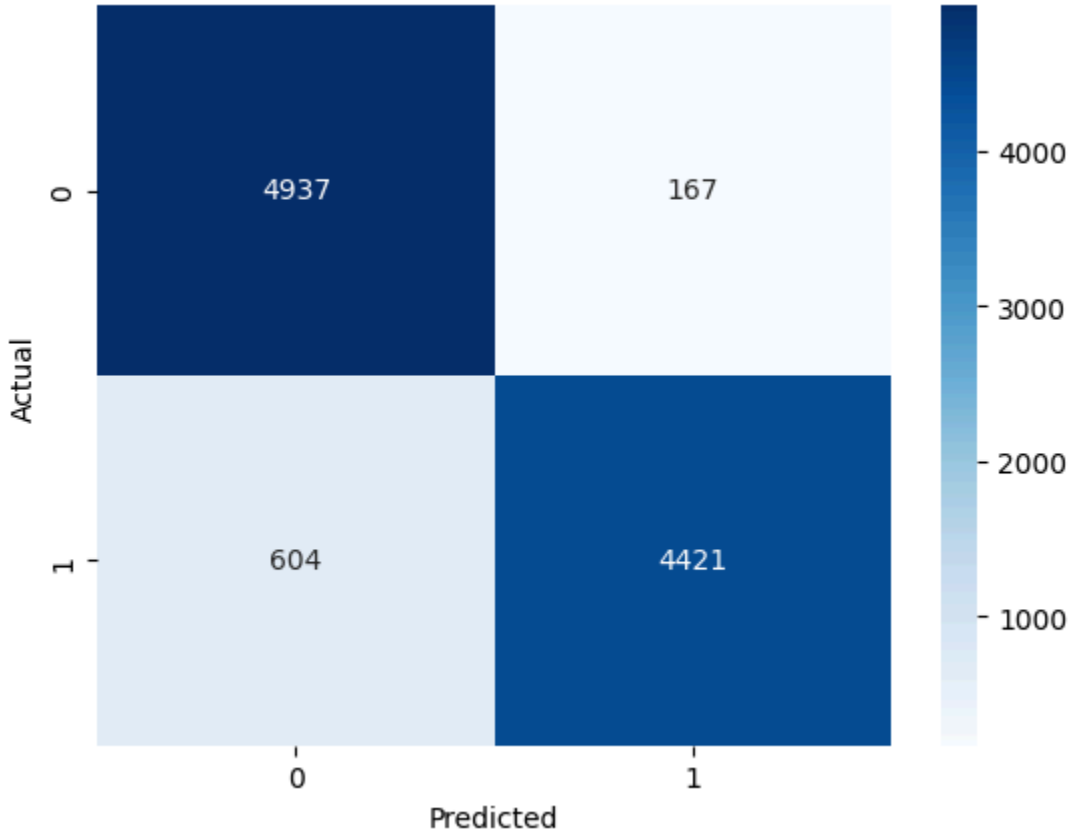
Limitations:

- Might need additional techniques to address imbalanced datasets.

Performance Metrics:

Accuracy: 0.93

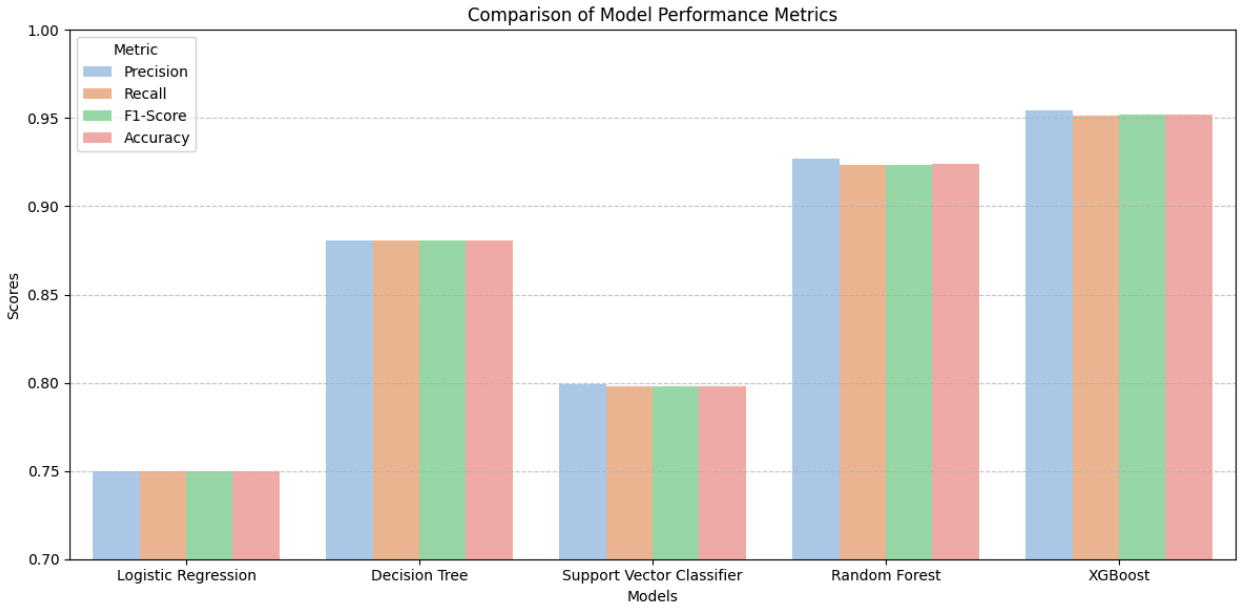
	precision	recall	f1-score
0	0.90	0.97	0.93
1	0.96	0.89	0.92
accuracy			0.93
macro avg	0.93	0.93	0.93
weighted avg	0.93	0.93	0.93



Random Forest showed balanced performance across all metrics, demonstrating reliability and generalization but still can’t beat XGBoost’s performance, which is understandable since it focuses on correcting errors in each subsequent tree built.

Final Comparison

In conclusion, our study demonstrated that advanced models like XGBoost, which capture nuanced relationships in the data, outperformed simpler models like Logistic Regression, which are limited to linear relationships. This distinction is particularly crucial for a critical task like loan approval, where precision in predicting credit risk directly impacts financial outcomes and decision-making reliability for institutions.



References:

- https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/
- https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- https://xgboost.readthedocs.io/en/stable/tutorials/model.html
- https://xgboost.readthedocs.io/en/stable/tutorials/model.html
- https://www.geeksforgeeks.org/support-vector-machine-algorithm/
- https://www.geeksforgeeks.org/understanding-logistic-regression/
- https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes.
- https://www.geeksforgeeks.org/decision-tree/
- https://visualstudiomagazine.com/Articles/2021/05/06/variational-autoencoder.aspx
- https://scikit-learn.org/stable/auto_examples/model_selection/index.html
- https://scikit-learn.org/stable/