Credit Risk Prediction

Credit_Risk Dataset

Description

Credit risk analysis is the process of evaluating the creditworthiness of individuals, businesses, or other entities to assess the likelihood that they will default on their financial obligations, such as loans, credit card payments, or bonds. This analysis is crucial for banks, financial institutions, and investors to make informed decisions about lending money or extending credit. Credit risk analysis gathers and analyzes a variety of data sources to assess a borrower's creditworthiness. This includes financial statements, credit reports, income and employment information, payment history, and more.

Business Problem Statement

1. How can we assess the creditworthiness of new customers or clients?
2. How can we develop an accurate credit scoring model to assess the creditworthiness of loan applicants?
3. Can we identify borrowers who may be at risk of defaulting on their loans in the near future so that proactive measures can be taken?
4. Can we identify instances of fraudulent loan applications or borrowers misrepresenting their financial information?

More Problem Statements will be added as we proceed further with this Project.

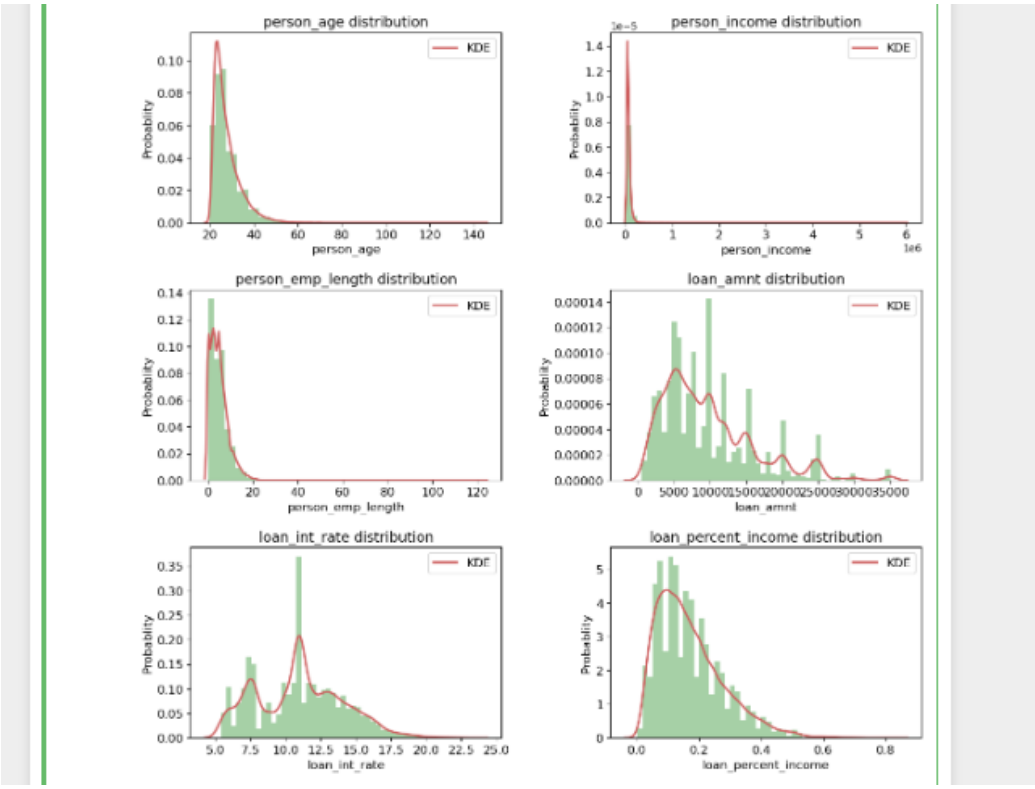
Tools that will be used in Project.

Excel: Data Understanding.
Python: Data Cleaning & Data Preprocessing.
Data Analysis & Machine Learning.
Power BI: Data Visualization.

Data Cleaning & Data Preprocessing.

Firstly, We imported all the necessary libraries and data into the Jupyter Notebook.And then We Connected the sqlite3 database with the Juypter Notebook. We Started the Data Cleaning and Data Preprocessing Steps:

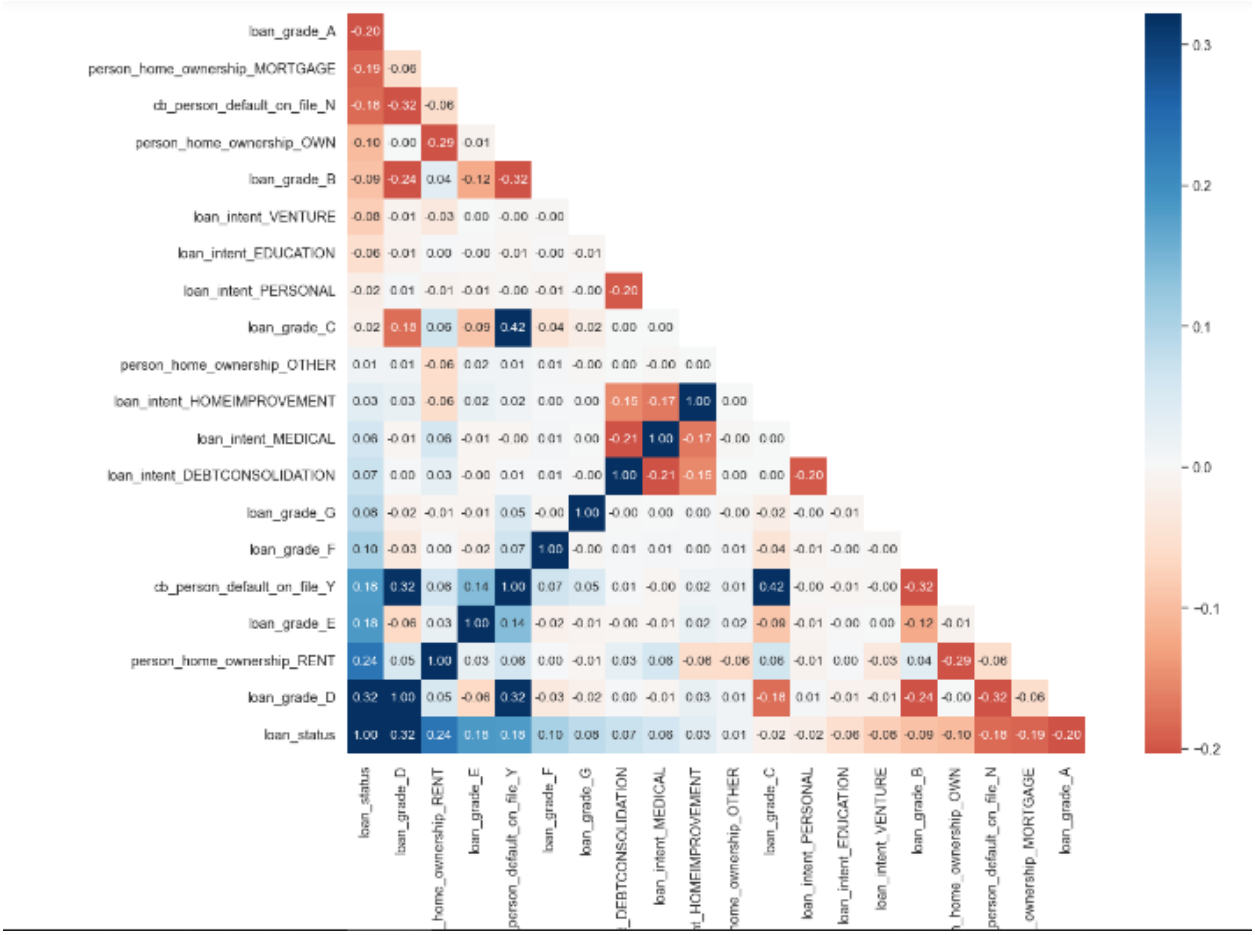
- We started with finding if there are any duplicated rows in the dataset. And we found out that there are 165 rows of duplicates in the dataset. So before removing the duplicated rows, we confirmed the total rows and columns in the dataset. (32581, 12).Now we removed the duplicated rows in the dataset. And after removing the duplicates rows we have this many rows (32416, 12).
- Next we looked for null values in the dataset and we found out that there are null values in the 'person_emp_length' and 'loan_int_rate' columns in the dataset. So we filled the null values in the 'person_emp_length' column by mode values and filled the 'loan_int_rate' by median values of the columns.
- Then we grouped all the numerical attributes of the dataset into one dataframe and named the dataframe as num_cols for the visualization distribution of these attribues in the graph. And then we used the bar graph of the visualization of the distribution of these attributes in the dataset.



- Then we observed that All of the distributions are positive skewed. **person_age:** Most people are 20 to 60 years old. In the following analysis, to be more general,

people age > 100 will be droped. **person_emp_length:** Most people have less than 40 years of employment. People with employment > 60 years will be droped. **person_income:** It seems that there are outliers which has to be removed (> 4 million). For all other variables, the distribution is more uniform across the whole range, thus they will be kept.

- Thus according to the above observations we cleaned the columns and created a new dataframe according to this new cleaned data and named the dataframe cleaned_num_cols
- Then visualized the corelations between the attributes in the new cleaned_num_cols dataset. And we observed that person_income, person_emp_length, and person_age: has negative effect on loan_status being default, which means the larger these variebles, the less likely the person is risky. loan_percent_income, loan_int_rate, and loan_amnt has positive effect on loan_status being default, which means the larger these variables, the more likely the person is risky.
- After that We dealt with the categorical variables using one hot encode for the co relations between them and then named the dataframe as encoded_cat_cols .



- Finally We merged the 2 dataframe cleaned_num_cols and encoded_cat_cols and the named new dataframe as cleaned _data

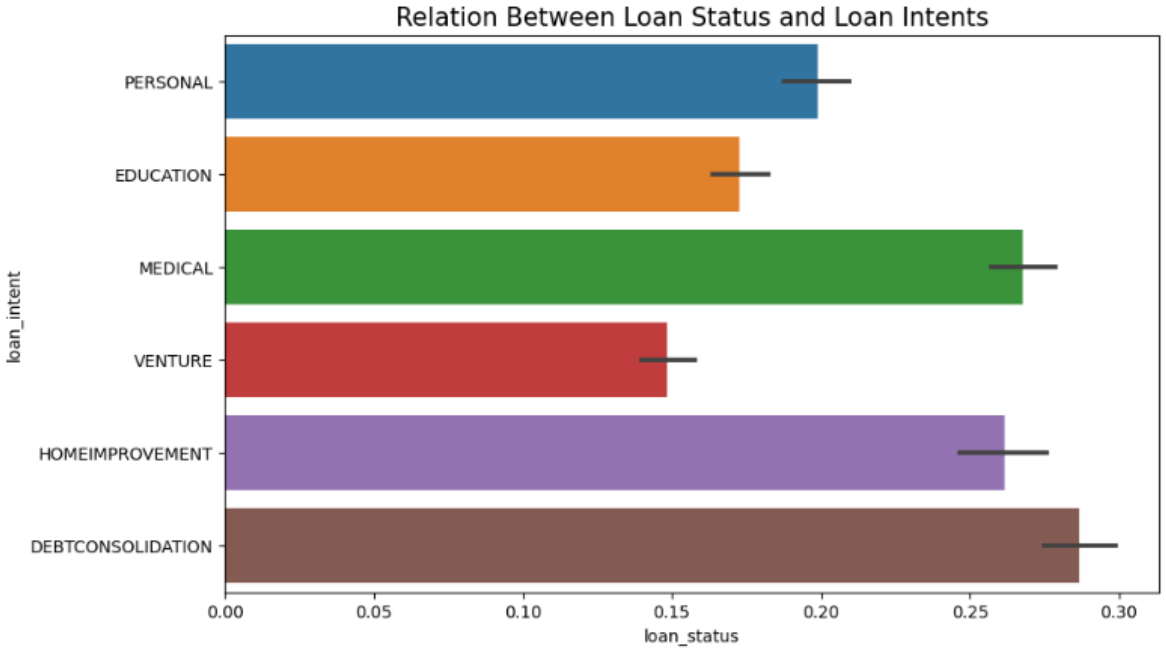
- And We checked the rows and columns and this new dataframe and The cleaned dataset has 31522 rows and 27 columns and The cleaned dataset has 7 numerical features and 19 categorical features

Exploratory Data Analysis (EDA)

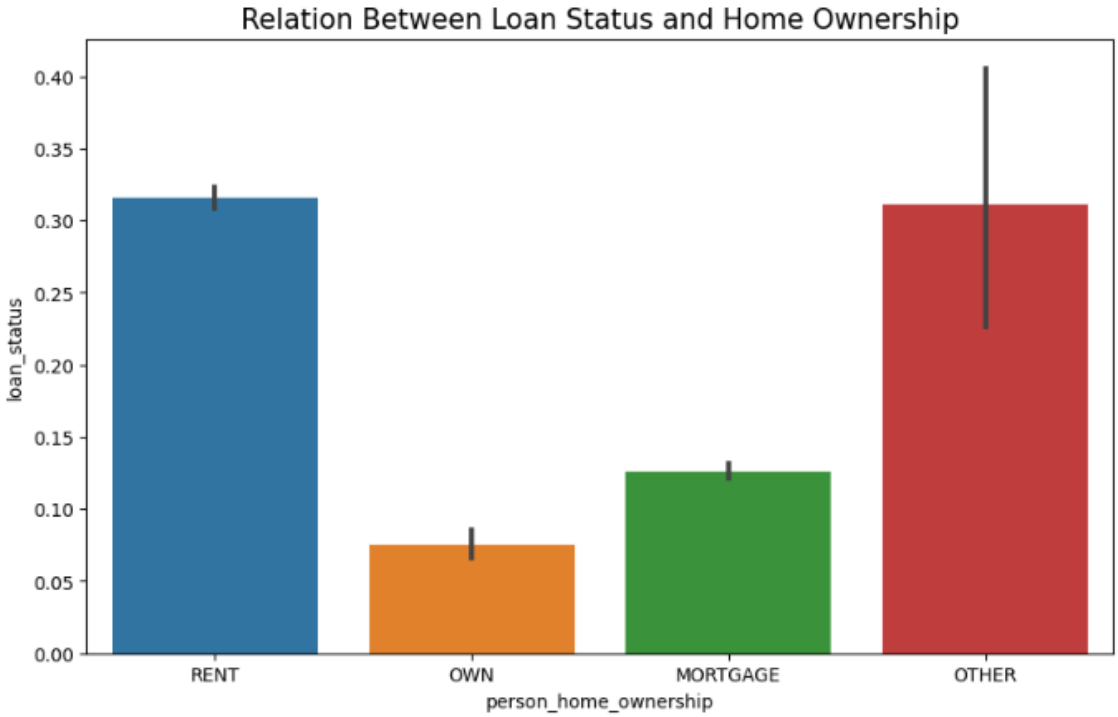
First We displayed the new cleaned_data diplaying all the columns of the dataset. And then we displayed the info of the dataset which displayed the column names and its number of non-null rows and its datatype and its shows that there are all together 26 attributes in the dataset.

Then we started with the visualization:

- First visualized the count of approved loan status with respect to Loan Intent

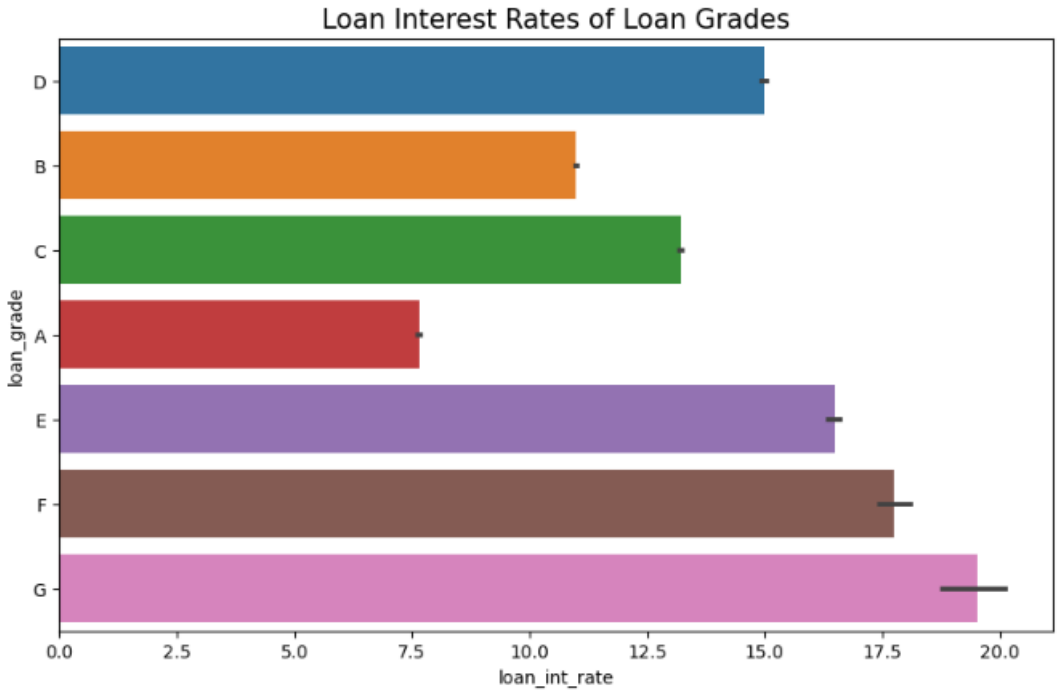


  Here We can see that most approved loan intent is Debt Consolidation and then on the second comes the medical and then on the third comes the Home improvement.
- Visualizing count of approved loan status with respect to person_home_ownership



Here You can see that at the top comes the rent and then comes the Other and the least count of approved loans comes to Own category.

- Visualizing the loan interest rate with respect to Loan Grades



Here the Highest interest rate goes to the G Grade and the Lowest interest rate goes to the A Grade But its Not in the Decreasing order.

Credit Risk Prediction (Data Modeling)

Now we will be starting with the main part of the Project, the Data Modeling Part. So firstly we will be starting with the Splitting the data by Train set and Test set and for that we will be splitting the data by 70/30 %. And so Here are the dimensions of the Train dataset and Test dataset. The train dataset has 22065 data the test dataset has 9457 data

Now that We have segregated the dataset we will be implementing the different algorithms in the train dataset. These algorithms will be

- KNN
- Logistic Regression
- Decision tree Algorithm
- XGB

By implementing these algorithms we returned the accuracy of the dataset with each respective model. The Accuracy like precision score and F1-score of the algorithms.

KNN

	precision	recall	f1-score	support
0	0.85	0.96	0.90	7447
1	0.71	0.39	0.50	2010
accuracy			0.84	9457
macro avg	0.78	0.67	0.70	9457
weighted avg	0.82	0.84	0.82	9457

Logistic Regression

	precision	recall	f1-score	support
0	0.81	0.98	0.89	7447
1	0.71	0.17	0.28	2010
accuracy			0.81	9457
macro avg	0.76	0.58	0.58	9457
weighted avg	0.79	0.81	0.76	9457

DecisionTree Classifier

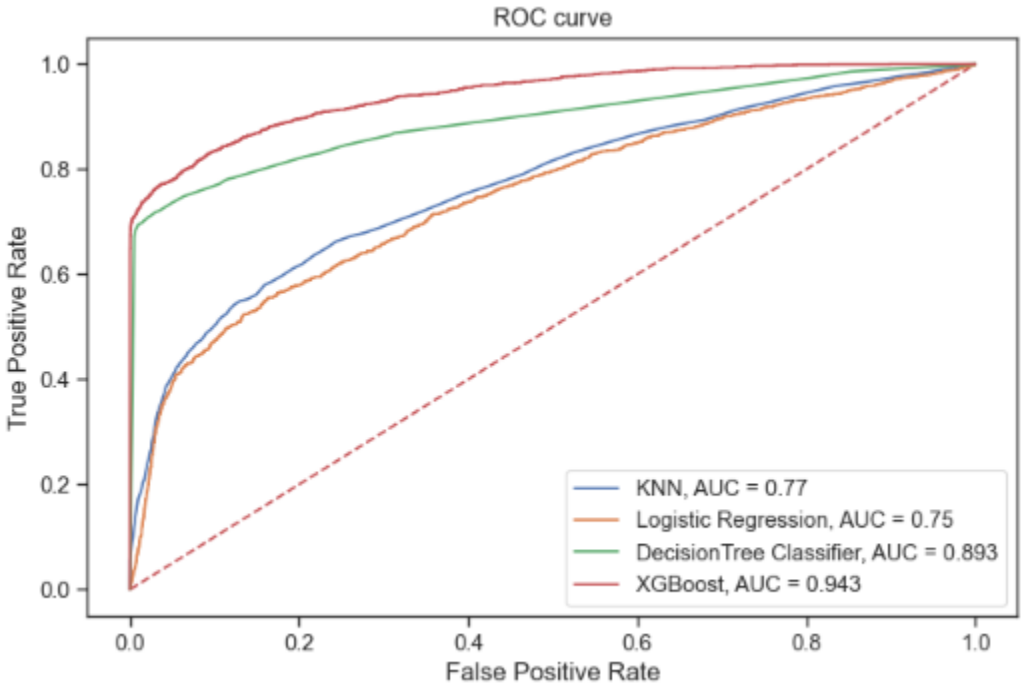
	precision	recall	f1-score	support
0	0.92	0.99	0.96	7447
1	0.95	0.69	0.80	2010
accuracy			0.93	9457
macro avg	0.94	0.84	0.88	9457
weighted avg	0.93	0.93	0.92	9457

XGBoost

	precision	recall	f1-score	support
0	0.93	0.99	0.96	7447
1	0.95	0.73	0.82	2010
accuracy			0.93	9457
macro avg	0.94	0.86	0.89	9457
weighted avg	0.93	0.93	0.93	9457

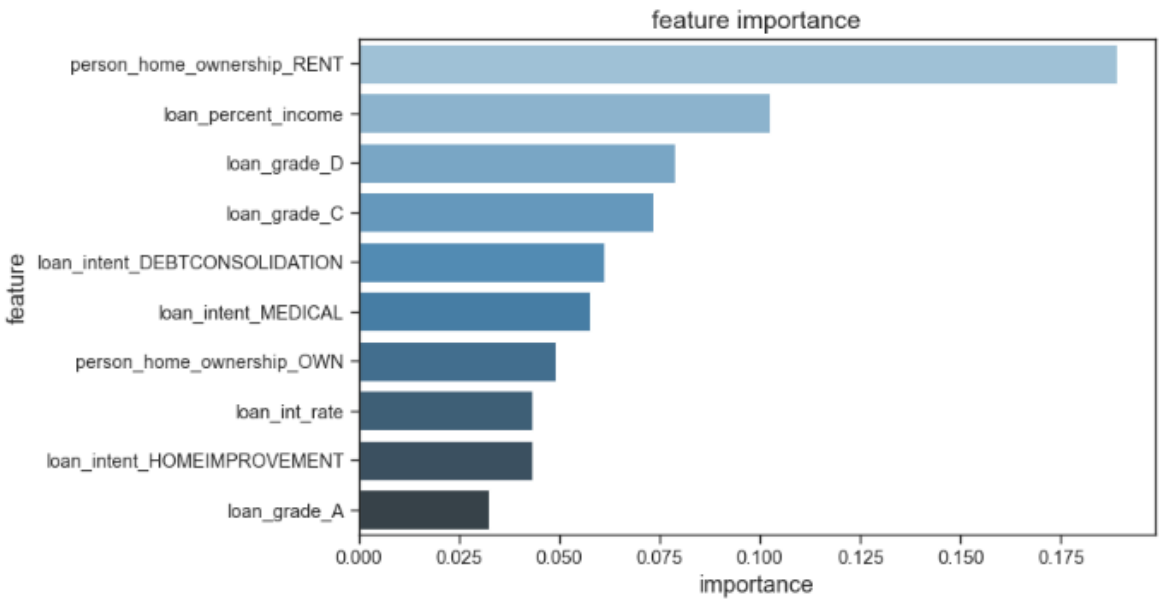
And then We displayed the graph using the ROC and AUC method. ROC (Receiver Operating Characteristic) and AUC (Area Under the Curve) are metrics commonly used to evaluate the performance of classification models, particularly in binary

classification problems. They help assess how well a model distinguishes between the classes (positive and negative) by analyzing the trade-off between true positive rate (sensitivity) and false positive rate.



Feature Importance

As for the feature importance , Using the package Xgb , feature_importance_, we created the dataframe called feature importance an plotted a bar graph with repected to the importance of the features that are required for the credit risk analysis.



Hyperparameter Tunning

Using RandomizedSearchCV package and XGBClassifier we can do the Hyperparameter tunning and create models.

```
Model with rank: 1
Mean validation score: 0.936 (std: 0.002)
Parameters: {'colsample_bytree': 0.9082891321693355, 'gamma': 0.35851103638916276, 'learning_rate': 0.2872671663213112, 'max_de
pth': 5, 'n_estimators': 143, 'subsample': 0.9977900668754272}

Model with rank: 2
Mean validation score: 0.936 (std: 0.002)
Parameters: {'colsample_bytree': 0.9403836171058041, 'gamma': 0.21946767413269447, 'learning_rate': 0.25078308278686895, 'max_d
epth': 5, 'n_estimators': 138, 'subsample': 0.9497248505892386}

Model with rank: 3
Mean validation score: 0.935 (std: 0.001)
Parameters: {'colsample_bytree': 0.9023271935735826, 'gamma': 0.4443405447766807, 'learning_rate': 0.25637095495468, 'max_dept
h': 5, 'n_estimators': 157, 'subsample': 0.9696737165364151}

Model with rank: 3
Mean validation score: 0.935 (std: 0.002)
Parameters: {'colsample_bytree': 0.9493025657771805, 'gamma': 0.2249853233585779, 'learning_rate': 0.21834082945169622, 'max_de
pth': 5, 'n_estimators': 124, 'subsample': 0.9553703052345801}
```

We have also used the the GridSearchCV hyperparameter tuning and it also returned almost same validation score.

```
Model with rank: 1
Mean validation score: 0.936 (std: 0.002)
Parameters: {'colsample_bytree': 0.9, 'gamma': 0.45, 'learning_rate': 0.26, 'max_depth': 5, 'n_estimators': 157, 'subsample':
0.97}

Model with rank: 1
Mean validation score: 0.936 (std: 0.002)
Parameters: {'colsample_bytree': 0.91, 'gamma': 0.45, 'learning_rate': 0.26, 'max_depth': 5, 'n_estimators': 157, 'subsample':
0.97}

Model with rank: 3
Mean validation score: 0.936 (std: 0.002)
Parameters: {'colsample_bytree': 0.9, 'gamma': 0.45, 'learning_rate': 0.26, 'max_depth': 5, 'n_estimators': 150, 'subsample':
0.97}

Model with rank: 3
Mean validation score: 0.936 (std: 0.002)
Parameters: {'colsample_bytree': 0.91, 'gamma': 0.45, 'learning_rate': 0.26, 'max_depth': 5, 'n_estimators': 150, 'subsample':
0.97}
```

And we got the AUROC Score for the model.

xgb base model AUROC socre: 0.9434990349715101

xgb best model using RandomizedSearchCV AUROC socre: 0.9448045458219845

xgb best model using GridSearchCV AUROC socre: 0.9438279263010848

And finally, we plotted the Threshold Optimization using a function and ploted the line graph of the Default Recall, Non-default Recall, Accuracy of the Model

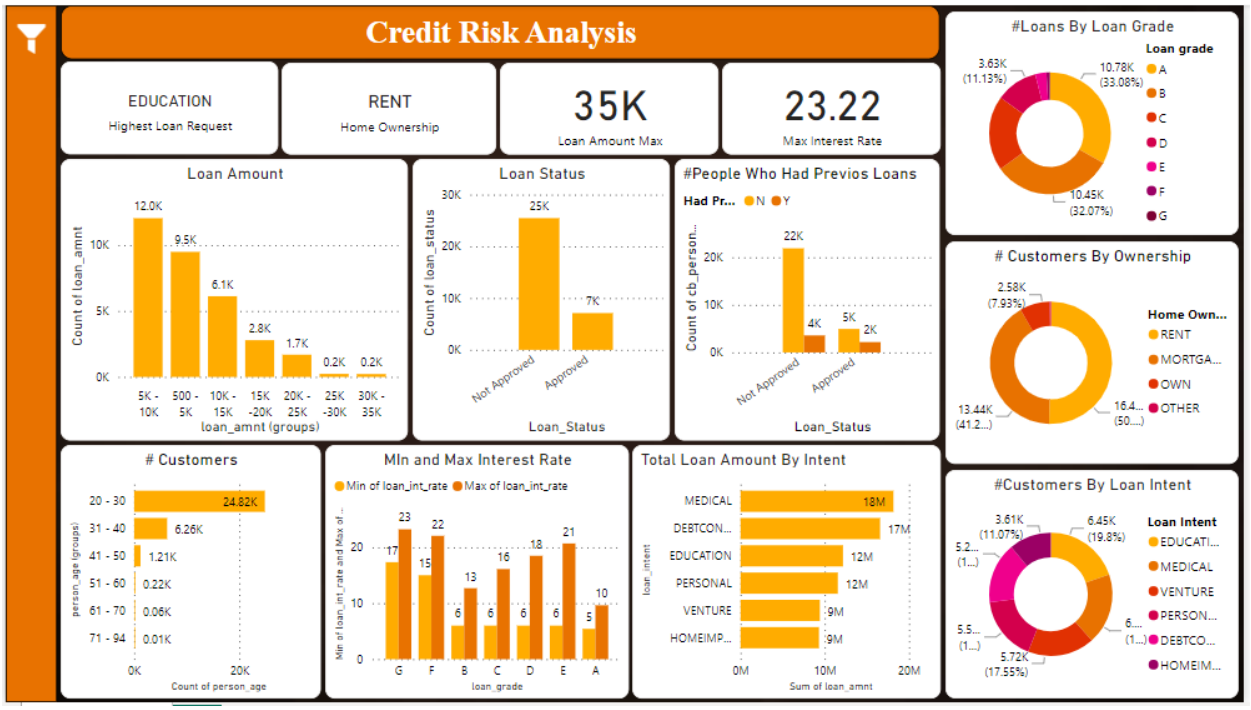
Discussion

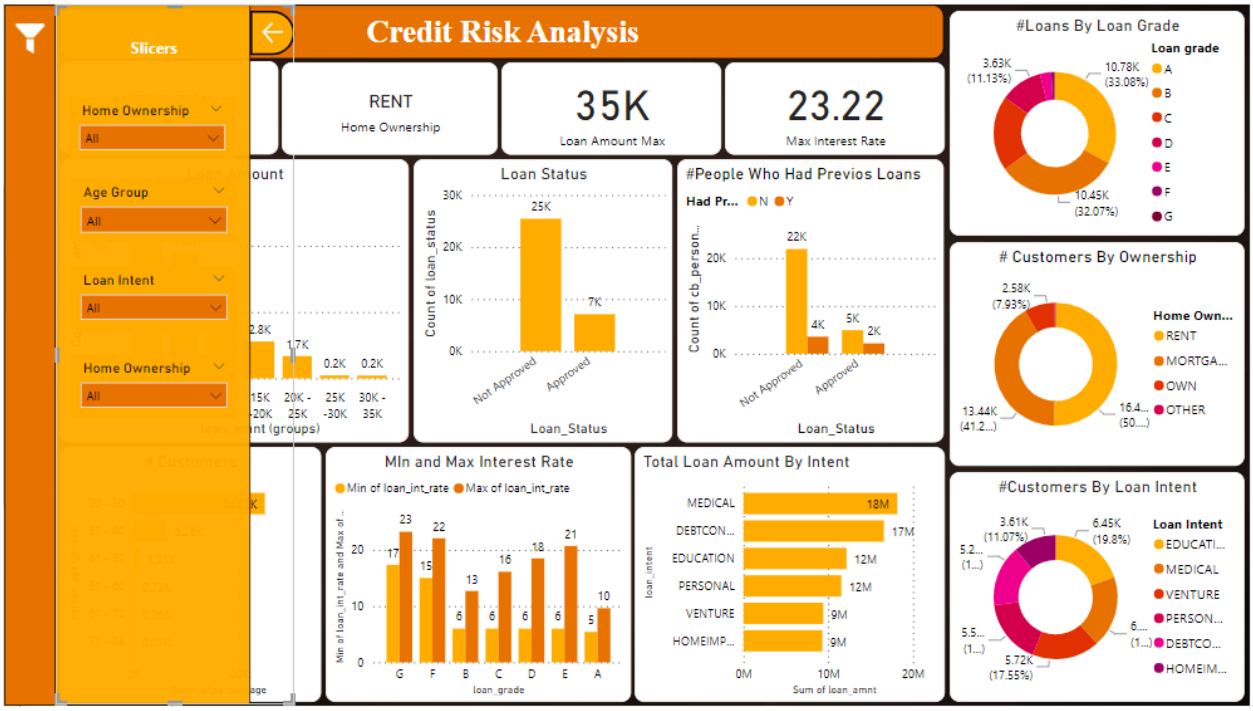
The XGBClassifier has the best performance with 0.947 AUROC score compared to other three classifiers KNN, Logistic regression, and decision tree using the base model.

Using RandomizedSearchCV to fast optimize hyperparamters, the model AUROC is improved to 0.9483

The optimal probability threshold for the best model is 0.525 resulting accuracy 0.937.

Data Visualization





Thank You