Group 18

Data Source

https://www.lendingclub.com/info/download-data.action

1. Introduction

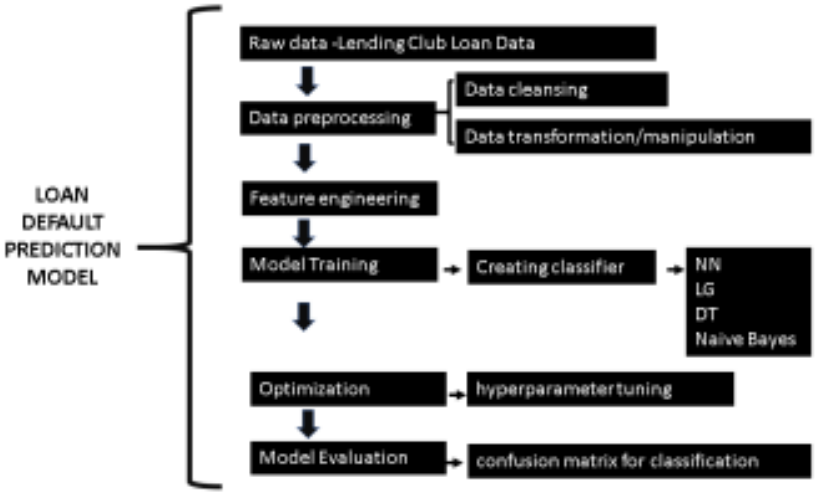
Establish a debt forecasting model that debtors can use to make better financial decisions. The second is that creditors can predict when the debtor will fall into financial difficulties. This model is used by Lending Club investors to determine whether to grant a loan. According to the information provided by this model, banks can make better decisions, and lenders can better plan financial aspects to avoid falling into debt crisis in the future.

Problem Statement

• Apply Machine Learning Algorithms to determine whether a loan application will result in a default, along with the loss that will be incurred if it should default.

• Where a simple finance-based approach would accept or reject a loan in a somewhat binary way, this multifaceted solution will anticipate a default and incorporate the severity of the loss that results.

Method:



2. Data Preparation

Read the two datasets

Data for this project comes from two datasets: "accepted_2007_to_2018Q4.csv" and "rejected_2007_to_2018Q4.csv". The 'accepted' dataset contains feature data for accepted loan application observations while the 'rejected' dataset contains feature data for rejected loan application observations.

We start by reading this data using the pd.read_csv() method, with the only parameter used being the dataset file title in both cases. Mixed data types are detected, but we ignore this for now, since we will manually preprocess the datasets next.

Data Preprocessing and Feature Selection

First, we will examine the features present in df_rej to determine the features we expect to be central to our analysis and machine learning model. Intuitively, the amount requested, risk_score, debt-to-income ratio, zip_code, state, and employment length appear relevant and are part of both datasets, so we included all of them at the beginning. Meanwhile, for the application date, loan title, and policy code, as these appear less relevant for our purpose, we dropped them from our analysis. We also appended 'loan_status' to the rows of each dataset, and assigned values of '0' for df_rej and '1' for df_acc.

In this sense, we make the best possible use of the df_rej features, which include fewer features, and attempt to find identical counterparts in df_acc, and apply a unique set of the same headers that can be applied across the two datasets:

- 'loan_amt'
- 'debt_income_ratio'
- 'zip_code'
- 'state'
- 'employment_length'
- 'loan_status'

The selection of these features results in three continuous quantitative variables ('loan_amt', 'debt_income_ratio', 'employment_length'), two categorical variables ('zip_code' and 'state'), and one binary target variable ('loan_status'). We specify the correct type of each feature in each dataset so that they can be interpreted as needed. Continuous quantitative variables are converted to float type while categorical variables are encoded to int type.

Next, the two datasets are merged, and the resulting dataset is cleaned/pre-processed.

Data Cleaning the features

Debt-to-income ratio

We remove the "%" character from the 'debt_income_ratio' feature in df_rej, before specifying the feature data types. We then convert df['debt_income_ratio'] to a float type.

Employment length

For the 'employment_length' value, if it is missing, the number of years is displayed in a combination of numeric and string characters. We start by transforming these values in these columns to a numerical format, and assign 0.5 to '<1 year' and 11 to '10+ years'. We transform the type of this column's data to float.

Zip code

We first ensure that the zip code is a string type. Next, we count the number of occurrences of ‘x’ in the zip code. An ‘x’ value occurs only when the zip code value is missing. While cleaning, we also notice that some zip codes contain the letter ‘O’ instead of the numeric ‘0’, and make sure to differentiate between the letter and numeric instances.

Encoding Categorical 'State' Feature and ‘Loan_status’ to Numeric Format

Using cat.codes, we encode the state feature into a numerical format, as well as, convert the dataframe for loan status into an integer type, so that the data is easier to work with.

Dealing with Missing Values

Finally, we print the first five rows of the dataframe and apply the info() function to check for any missing values.

Feature Engineering

There was a very limited number of intersecting columns between the rejected and accepted datasets so it was difficult to engineer new features to add to the machine learning model. We created a region column which uses the first digit of each zip code. The United States is split into 10 regions (0-9) seen below. We also group the loan amounts into bins to examine if that presents any other obvious trends.

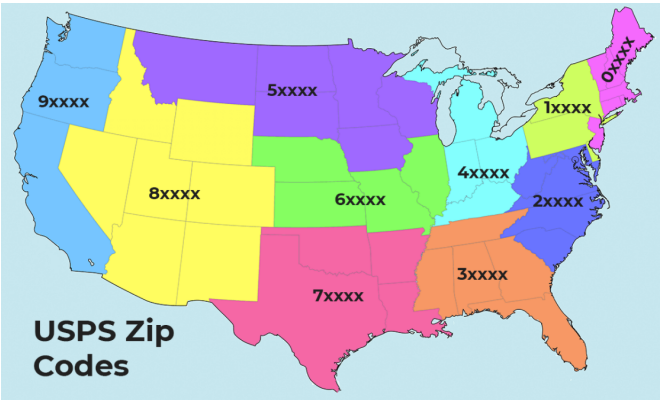


Image: USPS Zip Codes

3. Analysis

Exploratory Data Analysis

First, we examine the loan rejection database.

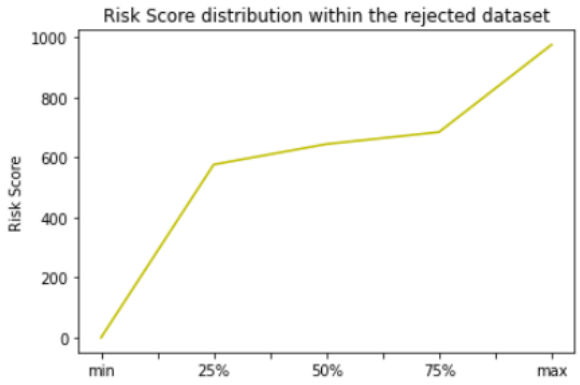
	Amount Requested	Application Date	Loan Title	Risk_Score	Debt-To-Income Ratio	Zip Code	State	Employment Length	Policy Code
0	1000.0	2007-05-26	Wedding Covered but No Honeymoon	693.0	10%	481xx	NM	4 years	0
1	1000.0	2007-05-26	Consolidating Debt	703.0	10%	010xx	MA	< 1 year	0
2	11000.0	2007-05-27	Want to consolidate my debt	715.0	10%	212xx	MD	1 year	0
3	6000.0	2007-05-27	waksman	698.0	38.64%	017xx	MA	< 1 year	0
4	1500.0	2007-05-27	mdrigo	509.0	9.43%	209xx	MD	< 1 year	0

There are 9 columns/features available in this dataset, namely:

- Amount Requested
- Application Date
- Loan Title
- Risk_Score
- Debt-To-Income Ratio
- Zip Code
- State
- Employment Length
- Policy Code

Risk Score

First, we explore the risk score feature in this dataset.



The distribution plot shows that there is a good variation in the risk score of loan applicants that were rejected and not all were "low" risk score applicants.

However, to get a better understanding of the distribution, we will further explore the risk score feature values and create an engineered feature called "Risk_Strength" by following the below metrics that explains how each range corresponds to the risk strength:

All other factors being equal, a higher credit score generally means you'll pay lower interest rates, fees and deposits. Over the lifetime of a loan, even a small reduction in rate can save you thousands of dollars in interest, so it pays to have a high credit score.

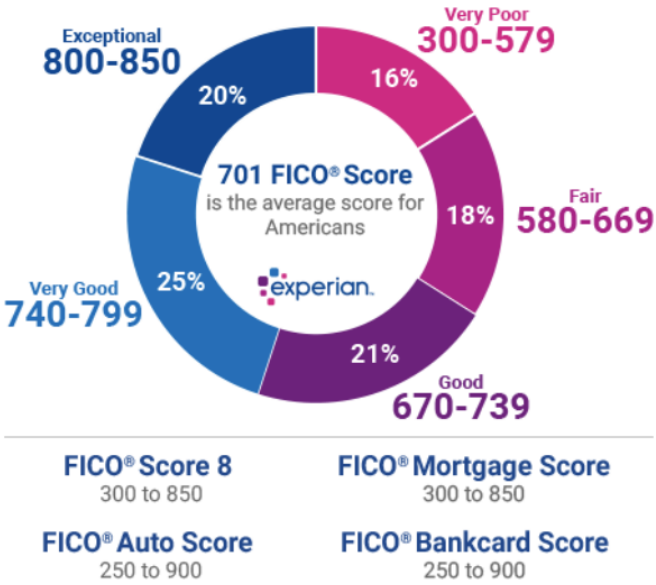
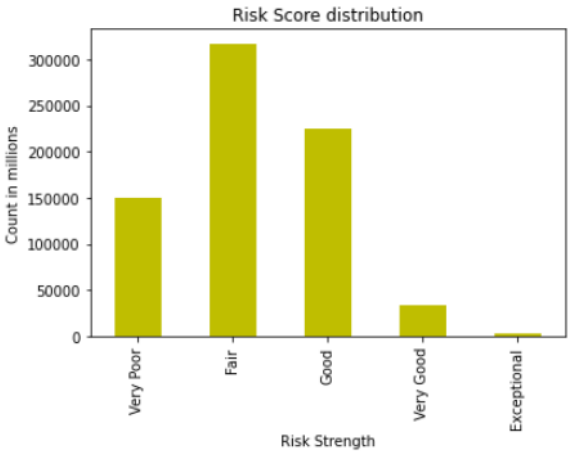


Image: Credit Score by Experian

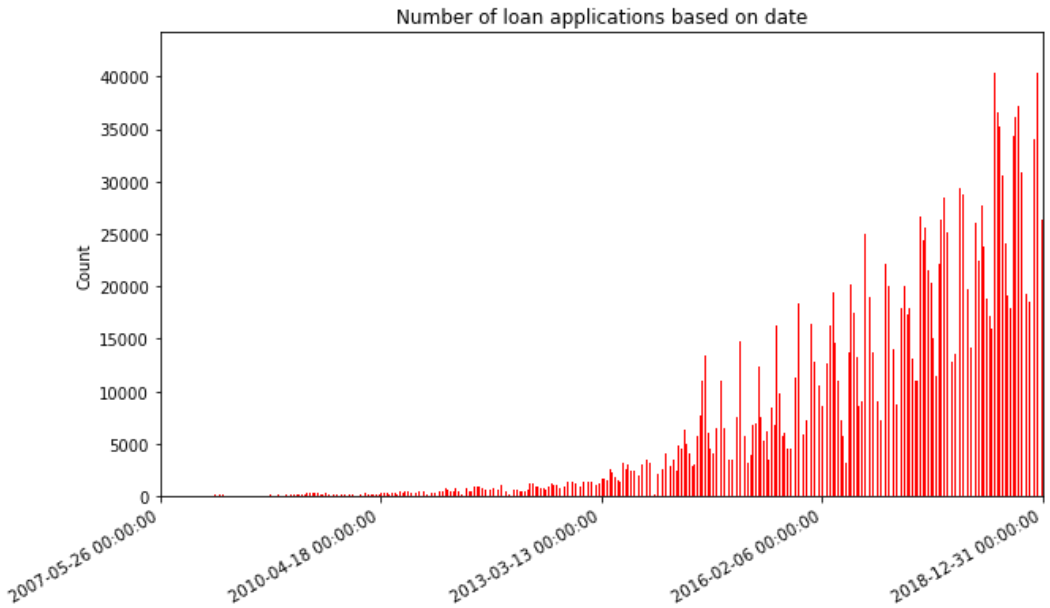


The plot gives more detail into the risk score distribution of all applicants that were rejected loan requests. Interestingly, the majority of applicants had "Fair" scores, while more applicants had "Good" scores than "Very Poor" ones but there were still some rejected loan requests.

Observation: It may be fair to say that “Risk Score” on its own was not a major factor, at least not on its own especially when seeing that the number of rejected applicants with "Good" to "Exceptional" scores were a lot.

Application Date

Next, we quickly explore the "Application Date" column just to get a feeling of the time aspect of loan requests if most rejections are happening recently or evenly across the time period of this dataset.



A few interesting observations can be seen from the plot:

- The dataset of rejected loan requests spans a time period of about 11 years from 2007 to the end of 2018
- The number of rejections interestingly follows an almost exponential increase across this time period
- Before 2013, it seems most loan applications were rarely rejected

Caution taken from the above insights: The number of loan requests per year is unavailable so it is unclear if the increase in rejections over time was due to more applicants with 'certain' features applying, or just due to a higher total number of applicants in general.

Observation: While the dataset spans a very long time period (i.e. 11 years) and considerations for loan applications may or may not have changed over this period, the distribution of the data shows more of the rejections were happening on the latter half of the dataset, thus it may be safe to conclude that whatever insights we gain from the analysis in this project work based on the impact of the different features in the dataset, will still be fairly valid for the loan request considerations of recent years.

Examining the acceptance dataset

Before we explore the remaining columns/features in the rejected dataset, let's have a look at the loan acceptance dataset.

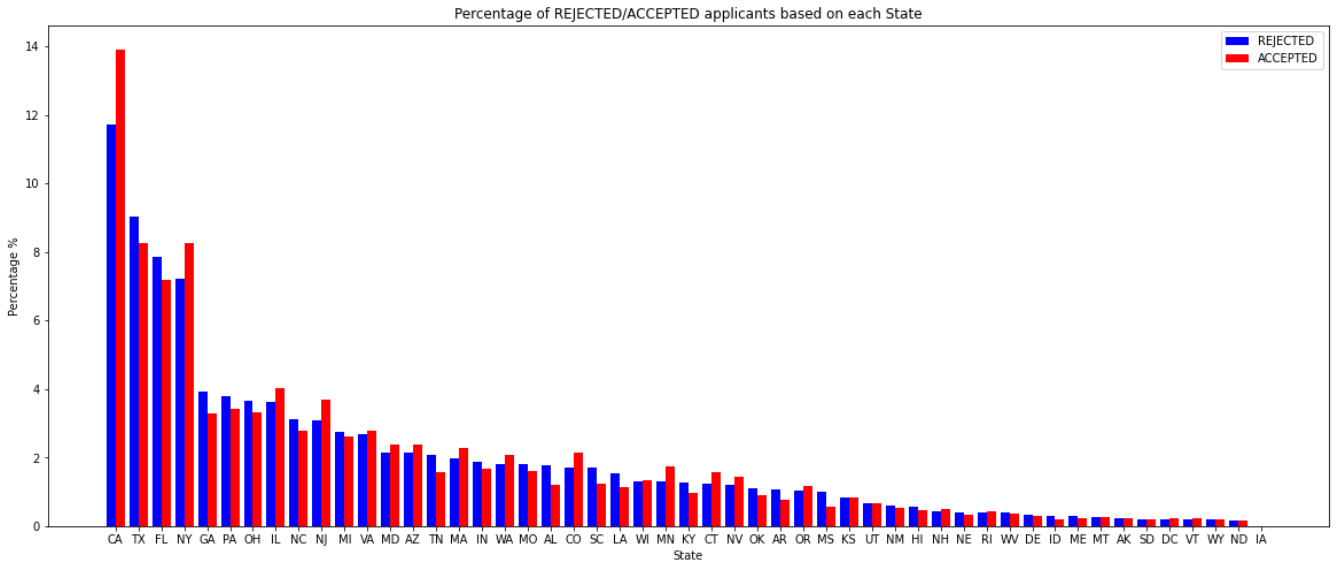
This database is more extensive with 151 columns/features compared to the 9 columns/features available in the rejected dataset. For the purpose of this analysis, to be able to determine how a feature may affect the loan decision, we will explore only the columns/features that intersect both datasets, namely:

- Amount Requested **OR** loan_amt
- Loan Title **OR** purpose
- Debt-To-Income Ratio **OR** dti
- Zip Code **OR** zip_code
- State **OR** addr_state
- Employment Length **OR** emp_length
- Policy Code **OR** policy_code

State

Now we will explore the 'State' feature. It would be interesting to examine if there are any variations in the rejected vs accepted dataset from one state to another, i.e. trying to find some sort of indicator if being in a particular state gives a loan requester any advantage of being approved.

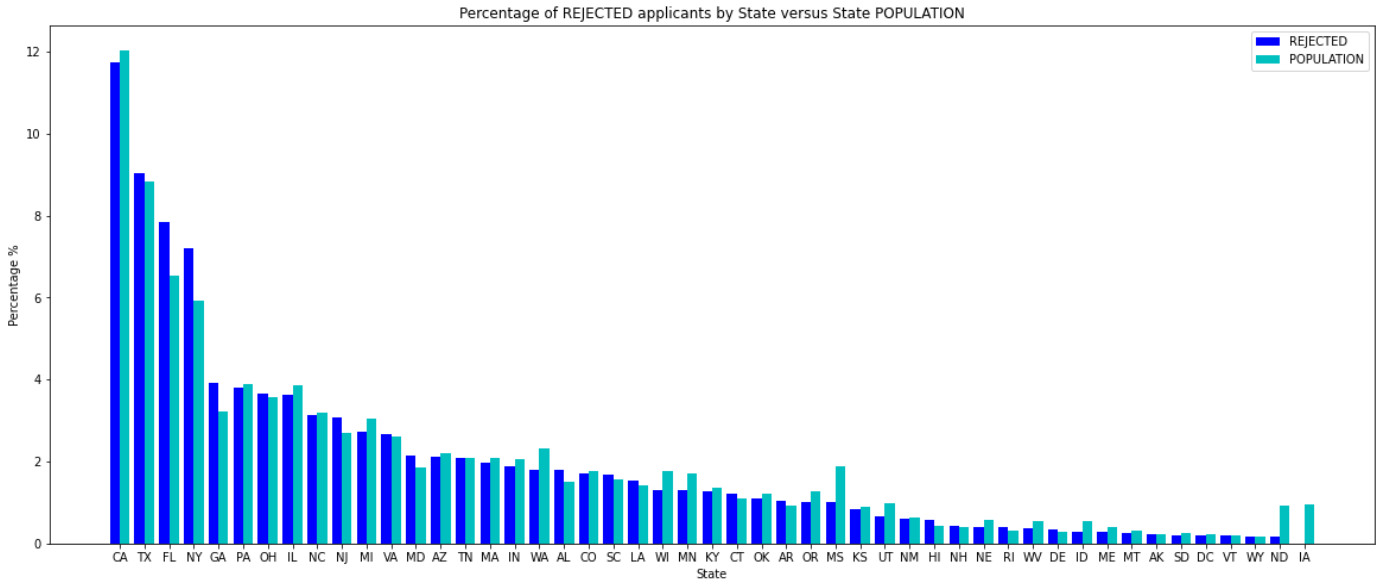
	Rejected	Accepted
CA	11.726290	13.913277
TX	9.025774	8.242475
FL	7.839727	7.165625
NY	7.201704	8.244864
GA	3.919220	3.282039

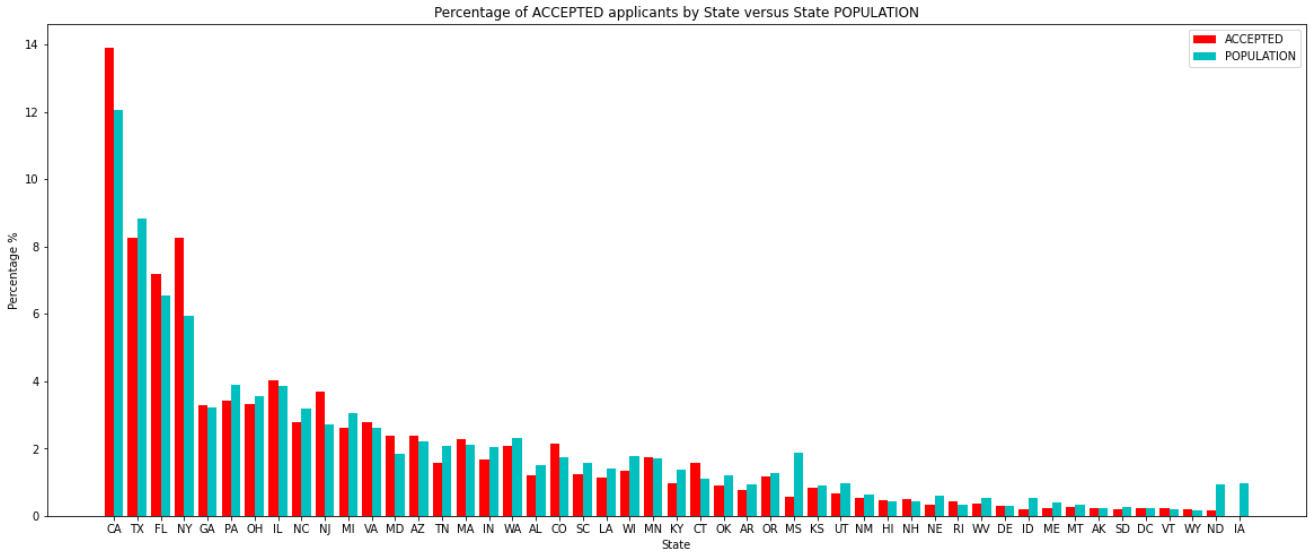


An initial consideration of the data from the rejected dataset may suggest that the majority of rejections could be seen in California (CA), Texas (TX), Florida (FL) and New York (NY), and that may suggest people applying in those states may be at a disadvantage. However, those same states appear as the top 4 states that have the highest rate of acceptance as well from the accepted dataset!

This suggests that maybe the reason why we see this similarity in rejected/accepted distributions within the states could be due to the population distribution. To verify, we pull in the 'Population' in each state and add it to the visualizations.

	Rejected	Accepted	Population	Population_percent
CA	11.726290	13.913277	39512223	12.037619
TX	9.025774	8.242475	28995881	8.833757
FL	7.839727	7.165625	21477737	6.543312
NY	7.201704	8.244864	19453561	5.926636
GA	3.919220	3.282039	10617423	3.234657

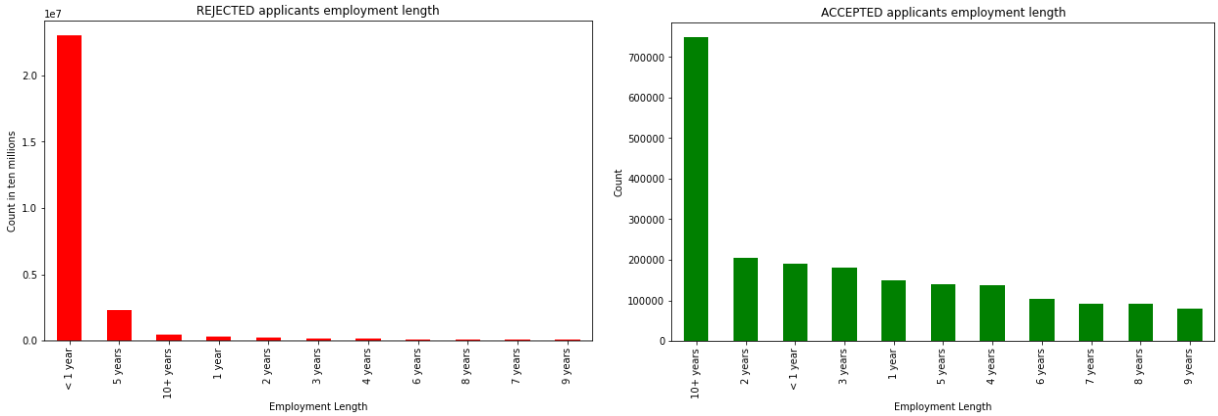




Observation: The plots show that the variation in the number of rejected/accepted loan requests across states is strongly skewed by its population distribution and thus population density needs to be taken into account. It is also important that the target population is properly defined and that the sampling frame matches the actual target population as much as possible.

Employment Length

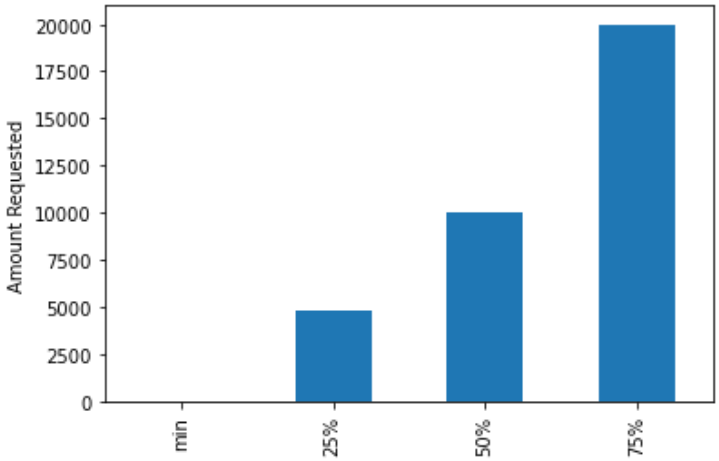
Now, we will examine how employment length relates to rejection/acceptance of a loan request.



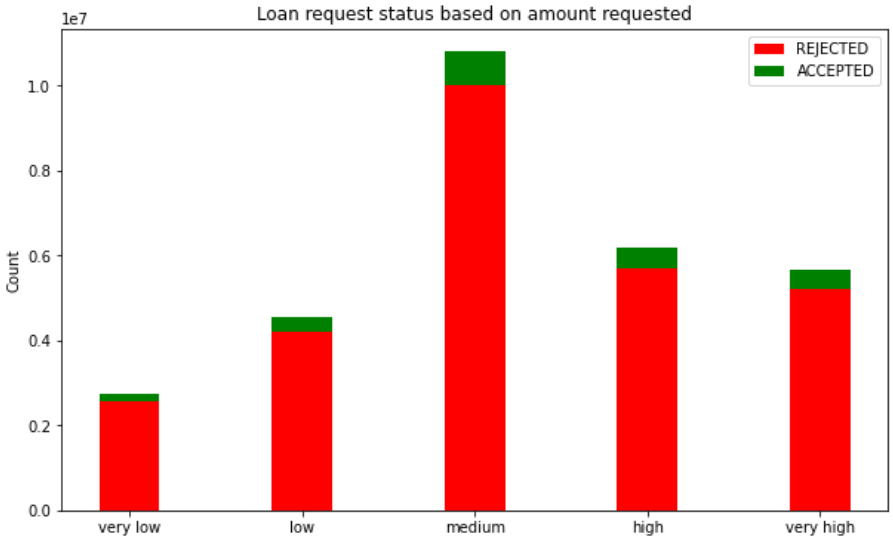
Observation: From the plots, it is clear that the overwhelming majority of rejected applicants had <1 year employment at the time of loan request, whereas the overwhelming majority in the accepted applicants had >10 years employment. This suggests that this feature could be a very important factor in determining the outcome of loan requests.

Amount requested

Now we will examine how does the amount requested relate to rejection/acceptance of a loan request?



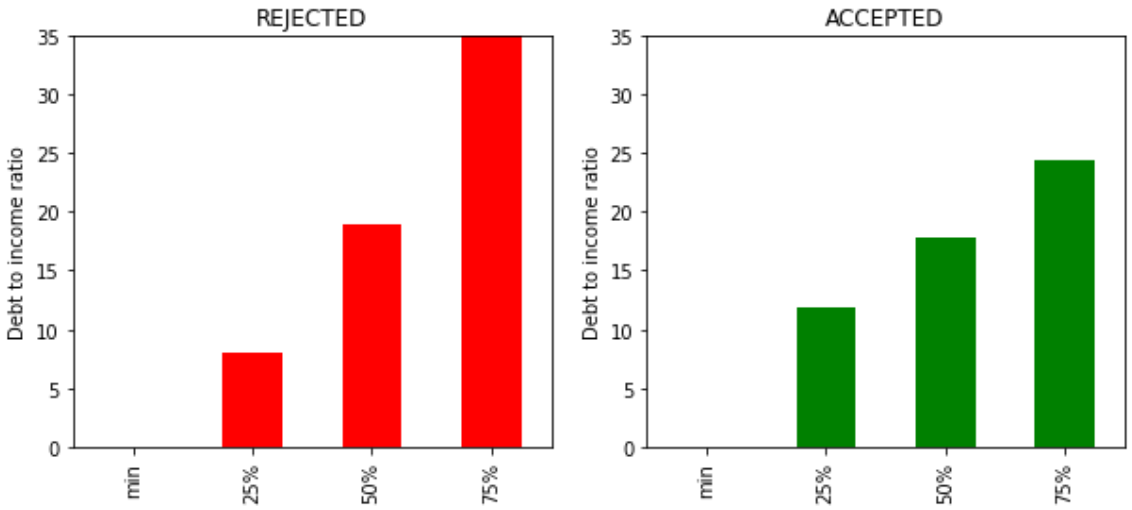
The plot shows that there is a wide range in the amount requested. Thus, we will create an engineered feature as a categorical one called 'Amount Range' that will classify the amounts requested from 'very low' to 'very high'.



Observation: Interestingly, the percentage of loan requests that were both rejected and accepted were higher for medium to very high amounts requested. While higher loan amounts requested makes sense for why it would have a higher chance to be rejected, the data shows that loan requests with higher amounts also had higher chances of being accepted. This observation suggests that this feature may be skewed by other factors such as the general number of loans >4000 (medium range) is very common rather than the smaller ranges specified.

Debt-to-income ratio

Now, we will examine how does the debt-to-income ratio relate to rejection/acceptance of a loan request?



Observation: The plots show that while there is a wide variation in applicants' debt to income ratios for both rejected and accepted ones, the general distribution for the accepted loan request applicants is lower than that of the rejected ones. This can be seen in the mean value (although the data hasn't been cleaned yet with outliers). This may suggest that lower debt-to-income ratios can give a loan applicant a higher probability of approval.

Policy code

Now, we will examine how does the policy code relate to rejection/acceptance of a loan request?

```
#Drop NA values and check how  may unique policy codes out there from the rejected dataset
df_rej['Policy Code'].dropna(inplace=True)
df_rej['Policy Code'].unique()
```

```
array([0., 2.])
```

```
#Drop NA values and check how  may unique policy codes out there from the accepted dataset
df_acc['policy_code'].dropna(inplace=True)
df_acc['policy_code'].unique()
```

```
array([1.])
```

Observation: We can see from both datasets that there are three policy codes: 0, 1 and 2. The data reveals 100% of the cleaned NA datasets that policy 1 were all accepted whereas policy 0 and 2 were all rejected. While this may strongly show that policy 1 is easier to lead to approval than policy 0 or 2, it is unclear if this data will be useful as we do not know enough about the policies offered and it is not possible that a certain policy has never been approved for 11 years!

Hence, it has been decided that this feature will not be explored further in our analysis.

Zip code

Now, we will examine how does the zip code relate to rejection/acceptance of a loan request?

```
#Find how many unique zip codes in the dataset
ziprej=len(df_rej['Zip Code'].unique())
zipacc=len(df_acc['zip_code'].unique())
print('Total number of zip codes =',ziprej+zipacc)
```

```
Total number of zip codes = 1959
```

Observation: There are 1959 different zip codes. More cleaning needs to be conducted with this feature before being able to analyze and examine it further.

Loan Title

While the purpose of the loan can be identified from this feature, and it may provide direct insights into loan request rejection or acceptance, this would require a much more intensive analysis of this very huge dataset. As such, it is decided that this feature will not be explored further in our analysis.

Trends, Correlations, Patterns in the Data, and Model Evaluation

By displaying a sequence of steps, **decision trees** give people an effective and easy way to visualize and understand the potential options of a decision and its range of possible outcomes. Each branch of the decision tree represents a possible decision, outcome, or reaction. The farthest branches on the tree represent the end results. Decision trees are frequently employed in determining a course of action in finance, investing, or business.

The **confusion matrix** for the decision tree and the **XGB**(eXtreme Gradient Boosting) model is used as a measure to describe the performance of the classification model by examining the number of true positives and true negatives on the diagonal, with a higher diagonal value on the matrix indicating greater accuracy. An XGB model is a decision tree procedure that fits the data better, so that overfitting is minimized, and produces better error rates.

Using a decision tree, the accuracy score obtained is 0.929295, while with the XGB model, the accuracy score is higher at 95%. The **precision score** is the number of correctly identified positive results divided by the number of all positive results, including those not identified properly. The **recall score** is a measure of success for making predictions, when there is significant imbalance between the classes, and is the number of correctly identified positive results divided by the number of all samples. The recall score was also fairly high with an accuracy of 81-95%. Finally, the **F-score** is a measure of the test's accuracy using the precision and recall values to calculate the F-score. The F-score was fairly high indicating a high level of accuracy ranging mostly from 83 to 97%.

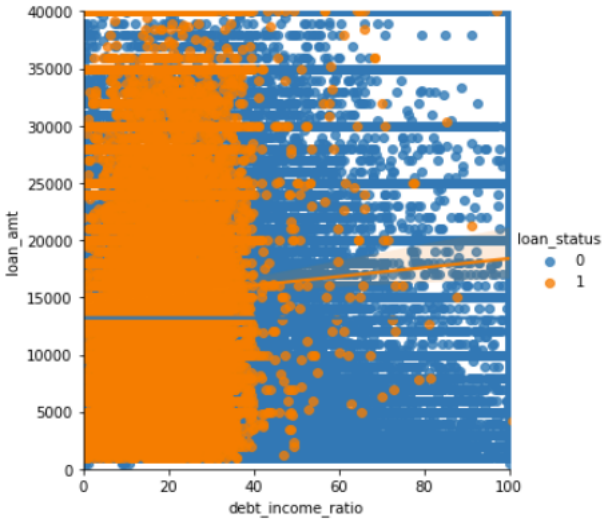
```
accuracy Score (testing) for Decision Tree:0.929295
Confusion Matrix for Decision Tree
[[934777  45607]
 [ 29933  58060]]
=== Classification Report ===
              precision    recall  f1-score   support

           0       0.97      0.95      0.96    980384
           1       0.56      0.66      0.61     87993

    accuracy                           0.93   1068377
   macro avg       0.76      0.81      0.78   1068377
weighted avg       0.94      0.93      0.93   1068377
```

```
accuracy Score (testing) for Decision Tree:0.929295
Confusion Matrix for XGB
[[930956  49428]
 [ 13560  74433]]
=== Classification Report ===
              precision    recall  f1-score   support

           0       0.99      0.95      0.97    980384
           1       0.60      0.85      0.70     87993

    accuracy                           0.94   1068377
   macro avg       0.79      0.90      0.83   1068377
weighted avg       0.95      0.94      0.95   1068377
```

We also applied a 1% random sample to the dataframe. We examined the plots to check for correlation between the features, but there wasn't a whole lot to check, as the number of contributing variables was limited in our analysis.

We couldn't find a specific relationship between the variables, and also tried to limit or engineer the other conditions, such as the product of the debt-income ratio and the loan amount, and considered varying the sampling size for the dataframe, so that there was a fine balance between the sample size and the visibility of the plot. To submit an accepted application, it's safe to assume that we need to limit our debt-to-income ratio to less than 40% and the loan amount to be less than $40,000.

Debt-to-income ratio vs. Loan Amount



4. Conclusions

Through this project, we hoped to establish a debt forecasting model that debtors could use to help make better financial decisions. By examining both the accepted and rejected loan applications by the Lending Club, we examined the possible contributing factors and model evaluation techniques, to better minimize the chance that debtors will default and reduce the severity of the losses that may result.

Through our analysis, we observed that certain factors such as the: debt-to-income ratio and employment length were most significant, while other factors were less meaningful. In general, a successful application occurred most frequently, when the debt-to-income ratio was less than 40% and the loan amount was less than $40,000.

5. References

1) Segal, T. DotDash. (2019 September). Decision Tree Definition. Retrieved from https://www.investopedia.com/terms/d/decision-tree.asp

2) DataTechNotes. (2019 July). Classification Example with XGBClassifier in Python. Retrieved from https://www.datatechnotes.com/2019/07/classification-example-with.html

3) Wood, T. DeepAI. What is the F-Score? Retrieved from https://deepai.org/machine-learning-glossary-and-terms/f-score

4) Intellipat. (2019 June). Introduction to Confusion Matrix in Python Sklearn. Retrieved from https://intellipaat.com/blog/confusion-matrix-python/#:~:text=Introduction%20to%20Confusion%20Matrix%20in%20Python%20Sklearn%20Confusion,classification%20accuracy%2C%20sensitivity%2C%20specificity%2C%20recall%2C%20and%20F1%20score

5) Kaggle. (2018 Jan). All Lending Club Loan Data. Retrieved from https://www.kaggle.com/wordsforthewise/lending-club/notebooks