Home Credit Repayment Difficulty: A consolidated approach in extrapolating an applicant's payment abilities

Amauris De Jesus, Eyob Tadele-Manhardt, Leandro Quezada, Tabahani Hayles

***Abstract*—In 2019 a record high was hit as housing loan debts reached a total of $9.406 trillion. The debt was attributed to a total of 56.1 million accounts which were opened during the course of that year. Of those 56.1 million accounts about 27% of the borrowers struggled to repay their loans. In order to better predict which applications will likely experience difficulties during their repayment period an existing set of current/historical applications and their associated statuses were analyzed and processed with the goal of determining which attributes can aid in foreseeing applicants that will have trouble with repayments. Improving the predictability of an applicant to repay the loan will not only benefit the bank but also improve the loaning experience for the applicant who would otherwise have to deal with penalties and liquidation if they fail to pay the loan.**

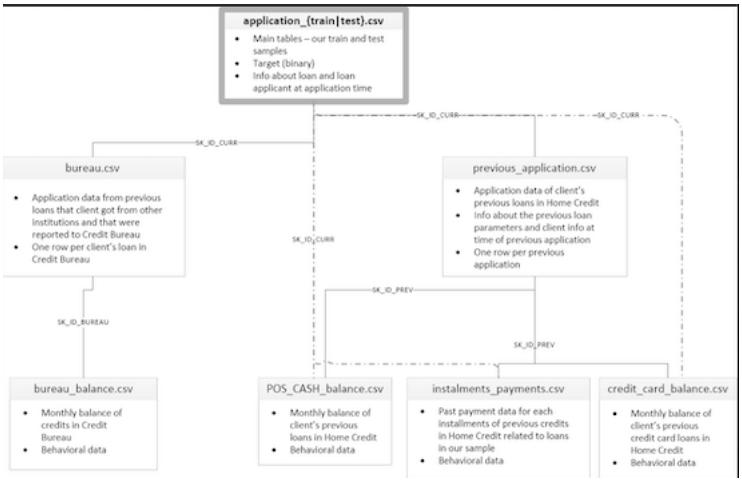
This problem was selected from Kaggle and the model developed was submitted after every iteration in order to test its performance. Using the model developed a score of 74% (%80 is the best score in kaggle) was achieved. A total of 140 columns were used in the model (out of a total of 422 columns). Ways to improve this prediction may be reliant on analyzing every column of the dataset or trying to feature engineer multiple columns into one.

I. Introduction

IS it possible to extrapolate an applicant's repayment ability based on a set of given attributes? The aim of this study is to develop a model capable of establishing whether a loan applicant will experience difficulties in repaying a loan. Determining a loan applicant's likelihood of encountering issues while repaying a loan is a critical business need for banks/lenders worldwide. In the United States alone, debt related to housing accounts for the largest category debt ($9.406 trillion). If a reliable model can be created which can better predict any future issues during the repayment period, banks/lenders will be better equipped to facilitate a more thorough selection process for applicants that allows them to successfully pay off a loan with little to no difficulty.

The data used to compose the model comes from a service called Home Credit which offers loans to customers in 9 countries. This data was composed of a total of 7 CSVs. The CSVs are broken down as follows:

- **application_train/application_test**: the main training and testing data with information about each loan application at Home Credit.
- **bureau**: data concerning client's previous credits from other financial institutions.
- **bureau_balance**: monthly data about the previous credits in the bureau.
- **previous_application**:previous applications for loans at Home Credit of clients who have loans in the application data.
- **POS_CASH_BALANCE**: monthly data about previous point of sale or cash loans clients have had with Home Credit.
- **credit_card_balance**:monthly data about previous credit cards clients have had with Home Credit.
- **installments_payment**: payment history for previous loans at Home Credit.



The process used in developing the model followed a simple set of 5 steps that were repeated in order to further model refinement. The steps implemented include:

1) Each team member choose one or two datasets to investigate
2) Clean and aggregate i.e choose/create specific features for main model
3) Join table with the main data frame
4) Run main data frame through the preprocessing pipeline
5) Evaluate and tune model
6) Repeat steps 2-5

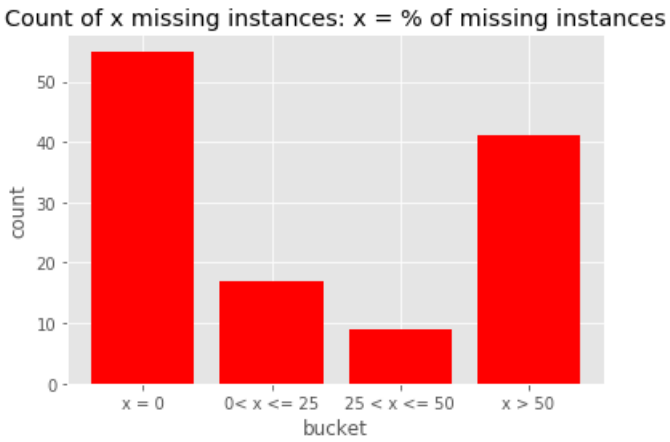
II. data

A. Application

The application dataset is the main table for this analysis, broken into two files: training and testing. The training application file contains the Target labels, whereas the testing application file doesn't. The primary purpose of the testing

file is to test the performance of our model on unseen data. However, since we didn't possess the actual labels for the test data, we were able to test the performance of our model by submitting our predictions to Kaggle. The training application dataset consists of 307,511 instances and 122 features, where 16 are categorical and 106 are numerical, including the target. In general, most machine learning models can't handle missing values. Most importantly, missing values can significantly affect the performance of the model. Therefore, we devised a function that groups the features in five buckets based on missing values. The buckets we choose are the following:

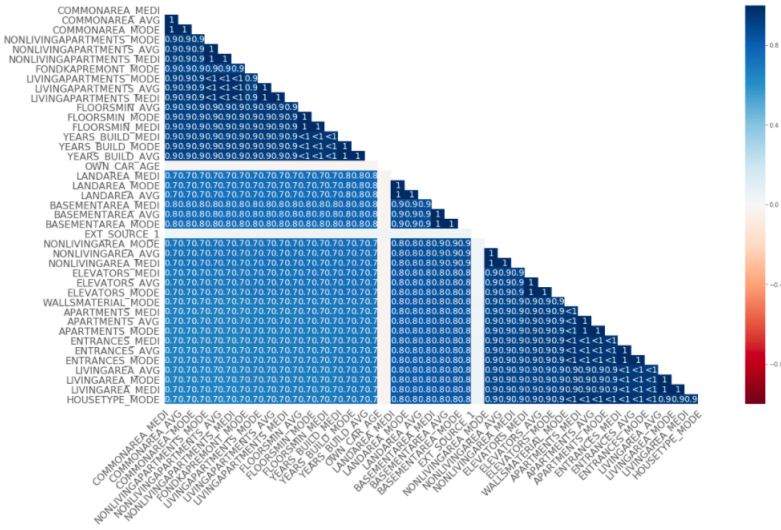
- 0% missing
- 0% $<$ missing data $\leq$ 25%
- 25% $<$ missing data $\leq$ 50%
- missing data $>$ 50%



The plot above the number of features that lie within the bucket groups. Specifically, the bar graph shows that 55 features contain no missing values, and 67 has some. Of the 67 features with missing values, 41 belong to the "x $>$ 50%". After further investigation, we discovered that most of the features that contain more than 50% of missing values are aggregated statistics about the clients' housing information. The list below displays the first 10 features that has over 50% of missing values:

- COMMONAREA_MEDI
- COMMONAREA_AVG
- COMMONAREA_MODE
- NONLIVINGAPARTMENTS_MODE
- NONLIVINGAPARTMENTS_AVG
- NONLIVINGAPARTMENTS_MEDI
- FONDKAPREMONT_MODE
- LIVINGAPARTMENTS_MODE
- LIVINGAPARTMENTS_AVG
- LIVINGAPARTMENTS_MEDI

The actual computation or calculation for the statistics above is unknown to us. There was no description in the dataset nor on the HOME CREDIT web page describing the procedure, i.e., HOME CREDIT did not make this information public. Moreover, we hypothesized that the high volume of missing values among the features is highly correlated due to their name and description. Specifically, we believed that if feature A has a missing value at instance, X, then features B, C, ..., will most likely have a missing value at instance X too. We tested this hypothesis using a heatmap where the color indicates the correlation coefficients between all features with a high volume of missing values. As the plot depicts, the missing values amongst the different features are highly correlated.



Note, EXT_SOURCE_1 and OWN_CAR_AGE are the only two features in the graph below that are not correlated with the other features.

In our final model, we decided to drop most of these features; however, for the baseline assessments we kept all of the features and imputed them accordingly.

The application dataset also holds vital information about the clients. For instance, approximately 2/3 of the clients are female. More generally, 2/3 of the clients don't own a car. On average, the clients have worked six years at a particular company, and the average age of the clients is about 43 years old. Also, the majority of the clients are married. Interestingly, when applying for a loan, only one member of the family usually shows up. On the technical side, the majority of clients request cash loan contracts instead of revolving loans and about 9% of the clients in this dataset have difficulties paying their loans. In other words, the class distribution is highly imbalanced. In the method section, we'll discuss how we handled this particular problem.

B. Bureau and Bureau Balance

The Bureau data contains behavioral loan data from external lending entities. The goal of including this dataset is to find bad behavioral patterns for an applicant in other banks that can be a good indication that the applicant will also fail to make payments for the current application. The bureau.csv data file contains 17 columns, and bureau_balances.csv contains 3 columns. Exploring the bureau.csv there was missing data for

- DAYS_CREDIT_ENDDATE
- DAYS_ENDDATE_FACT

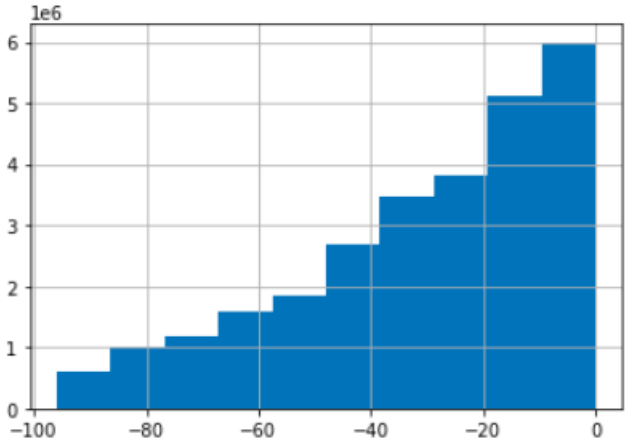
- AMT_CREDIT_MAX_OVERDUE
- AMT_CREDIT_SUM_DEBT
- AMT_CREDIT_SUM_LIMIT
- AMT_ANNUITY

For now we will only consider AMT_CREDIT_SUM_DEBT and fill nas with 0. We can adjust and come back to it later. For the bureau_balances.csv, there was no missing data and we can leave as is. Also one thing to keep in mind is that not all loans in bureau.csv necessarily have historical data, that's because bureau_balances.csv only contains a subset of historical data. This is troubling since we will need to keep that in mind when merging both datasets to make sure we don't exclude some rows in bureau.csv. In addition, the status of a loan in bureau.csv (whether it has Day Pass Due (DPD) does not necessarily mean it had DPD in historical data and vice versa. That's because the data in bureau.csv only represents the current status of a loan. For example, a loan may have previously had DPD, but if the applicant has paid all the dues then the status will be good. This is important to note since we will have to keep track of both historical DPD, and current DPD to make sure we take into consideration historical DPD and current DPD. Intuitively, some important columns that may lead to good target predictions can be:

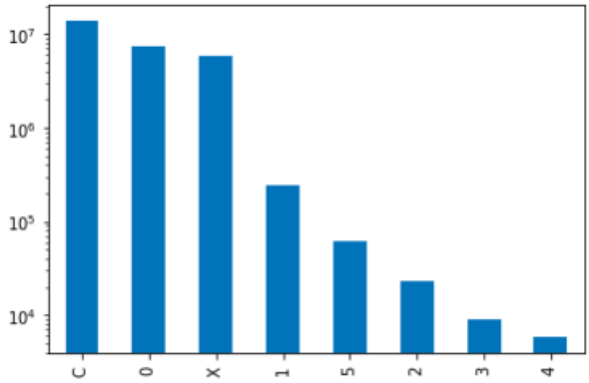
- DPD (Days past due) counts in both bureau_balances.csv (by aggregating it) and in bureau
- Current sum/debt ratio (and potentially debt/income ratio when joined with applications table)
- MONTHS_BALANCE (how far back from application day the balance is for)

but whether we include this in the training set will be determined by feature selection techniques discussed later in the feature selection section. Some charts that help explain the distribution of certain key features are listed below. This helps us understand the values of these metrics.

Distribution of Month Balances:



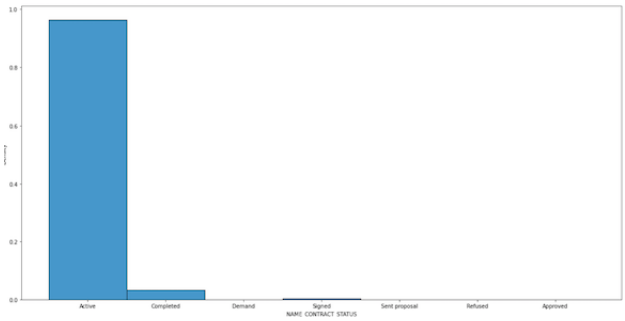
Distribution of Statuses (DPD)



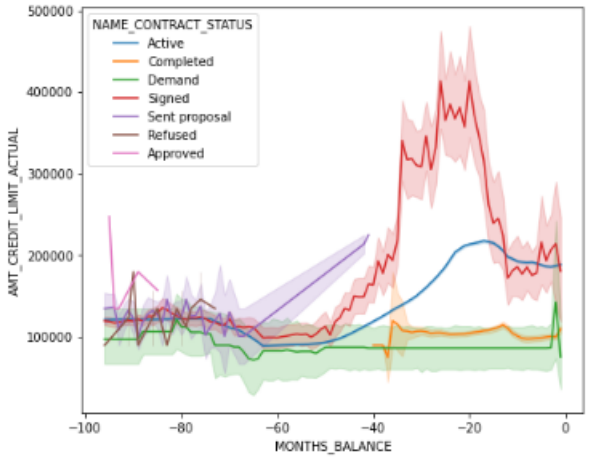
C. Credit Card Balance

We all know the implicant importance of credit history as it relates to mortgage loans. We are often told that we need to have a credit score higher than 700 for us to be a good lender and get lower interest rates. There are several factors that contribute to a FICO score, like payment history, length of credit, and amount used are all models of the payment behavior of an individual . In the simplest sense an individual that constantly pays their bills on time and has a longer history of credit is deemed the better lender. Using this idea of a good lender as context for our dataset 'credit_card_balance.csv', we can see trends that will be useful in our model building process. We are trying to establish the relationship between the credit_card_balance and application data namely, the relationship to the target classes.

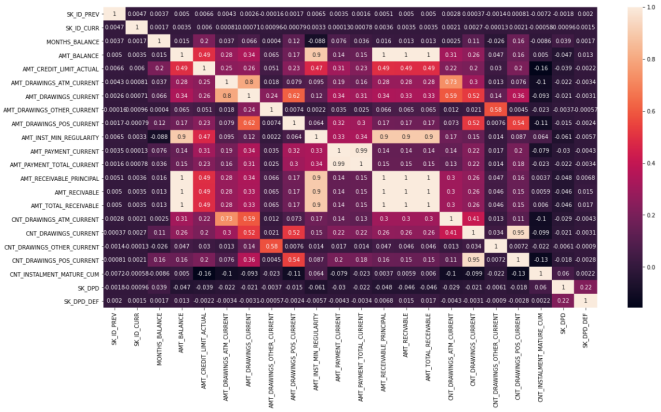
The credit_card_balance data contains a significant amount of information. This sheet comprises 23 columns of which only one is categorical, and there are 3,840,312 entries. The numerical columns give us details of an individual's credit history such as how much money they have remaining on their accounts on a month-to-month basis, and the age of the account. This table does have missing values, but consider the context, if it was not in the columns of 'MONTHS_BALANCE', or 'NAME_CONTRACT_STATUS', the value was replaced with a zero otherwise that information was dropped. Our categorical column, namely the 'NAME_CONTRACT_STATUS', which tells us the status of the account. To no surprise out of this 'NAME_CONTRACT_STATUS', the majority of contracts are active meaning it's currently being used. The ACTIVE value contributes more than 80 percent of this column's value, for this particular column we have to encode the values so we can use them for our model.



Looking at the metadata for our table, we should note that the column 'MONTHS_BALANCE' should be treated as a time series for our data; it describes the month of balance relative to the application date, in which -1 one means most recent data and and less than means older data. Graphing this as a time series we see some particular trends in relation to other columns, credit limit increases as an ACTIVE account gets older, a simple explanation for this is that as you use your card and you don't have late payment credit card companies tend to increase your limit.



Given how numerical this particular dataset is, we can test how correlated these columns are to one another.From the graph we can see we have highly correlated columns, this makes sense as the columns is capturing common attributes of credit behavior, it very likely that an individual will start spending more money (more account balance), when their credit limit goes up, or that the balance increases as the accounts gets older. If we move onto feature selection we will have to deal with these highly correlated fields, in order to make sure our final model doesn't overfit based on the attributes.

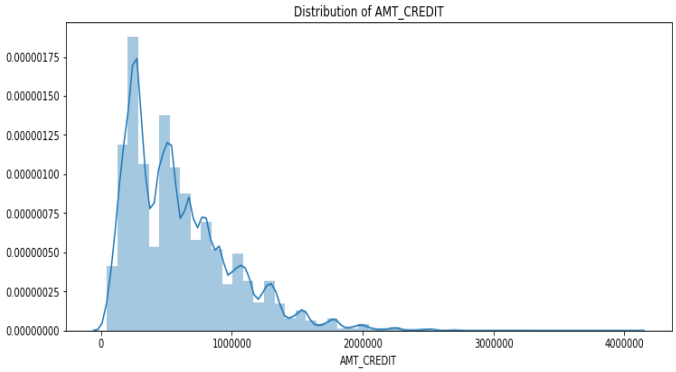


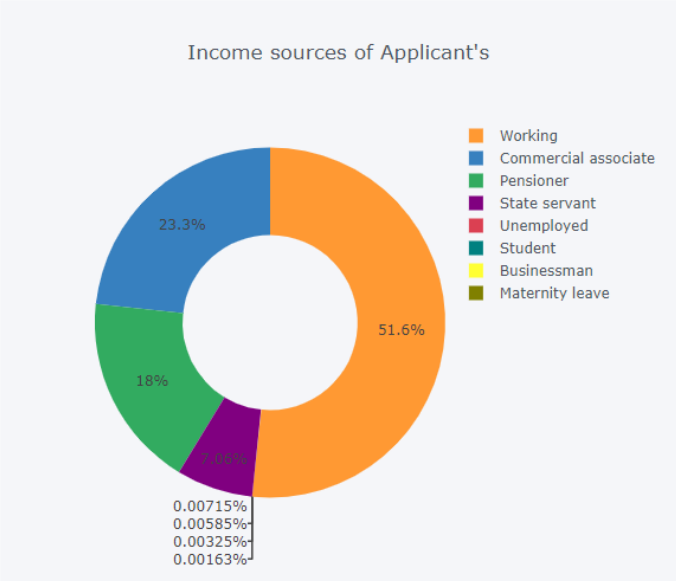
D. Previous Applications

The data contained within the previous_application CSV pertains to prior applications for clients using the Home Credit service. Of the total 37 columns contained within this dataset, 16 were missing over 40% of the data. Taking a brief look at some of the columns we can observe the following:

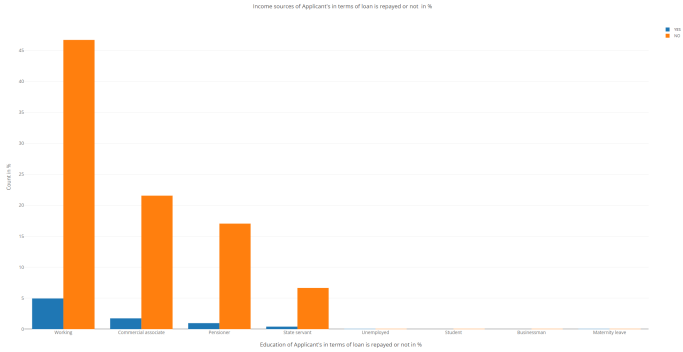
- The majority of previous applications were rejected for rejection code XAP
- Cash loans are normally made for the purpose codes XAP & XNA
- There are 3x as many repeat applicants as there are new ones
- The majority of previous applications were approved
- Just about the same amount of cash loans were made as consumer loans
- Most payments were made as cash through the bank

We can also take a look at the distribution of the amount of credit applied for in prior applications along with some of the income sources for those applications filed:





In attempting to determine which factors may contribute to whether or not a loan was repaid we can take a look at some of the columns that may have an impact in repayment such as income source, education, & occupation.The following three plots display the distributions of income source, education, and occupation.



III. Methods

A. Prepossessing

As mentioned above, this dataset is highly imbalanced. Therefore, we decided to use the Synthetic Over Sampling technique (SMOTE), where we created synthetic data from the minority class using the KNN algorithm. This method cost the least in throwing away vital information and increased the accuracy of our baseline model by a significant amount. Regarding the other preprocessing steps, we created a pipeline that standardized all of the continuous variables and imputed the missing values depending on the features' datatype.

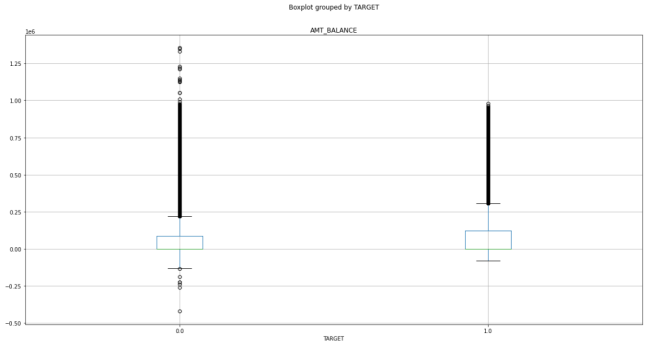
B. Feature selection

For feature selection we went through a 2 pronged approach, first we removed the highly correlated columns, first by making the correlation matrix and then setting a threshold (.90) items higher than that we removed, once we removed those correlated we fed that into the sklearn RFE using LogisticRegression, RandomForest, and Support Vector Machine as estimators for the class.

We used a brute-force approach for removing features using the RFE utility in sklearn, we started with our full set of 23 properties, we generated the features for N-1 for all 3 models, so we would have a rfe for logit, random_forest, and svm each list of features would be then pipe into our training models (Logistic Regression, Random Forest, and Support Vector Machines). Once we got the testing accuracy for our model, we compared the f1 scores, and if we didn't reduce the accuracy, we would repeat this process reducing the number of features by 1, until our testing accuracy was below the original accuracy score. Using this method for selecting features for the credit card dataset set we were left with the following useful columns :

- 'MONTHS_BALANCE'
- 'AMT_BALANCE'
- 'AMT_CREDIT_LIMIT_ACTUAL'
- 'AMT_RECEIVABLE_PRINCIPAL'
- 'AMT_TOTAL_RECEIVABLE'
- 'SK_DPD_SUM'
- 'SK_DPD_DEF_SUM'

Comparing the features we've extracted to our target variables we can see that there is some metric to our choices. Looking at the AMT_BALANCE, we can see that when compared to the target we can see that there tends to be amounts that are negative when the individual has no issue paying their loan (Class 0), an explanation for this behavior is that your are more likely to pay your current balance and the previous balance (paying more than the currently monthly balance), this would be an indication that this person has the ability to fully pay all the amount they owe.

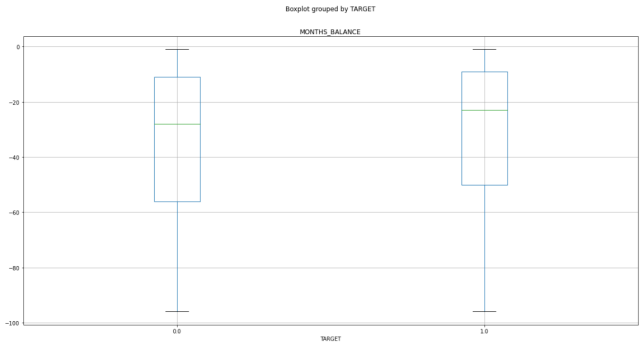


In addition, we can look at the MONTHS_BALANCE as it relates to the TARGET class. Applications that have no problem paying their loan (Class 0) tend to have longer length of accounts, these tend to be older accounts. Which makes sense once we apply this result to the real world. An account that is older and active gives an indication that an individual is more likely to pay their loan. [months_blance]

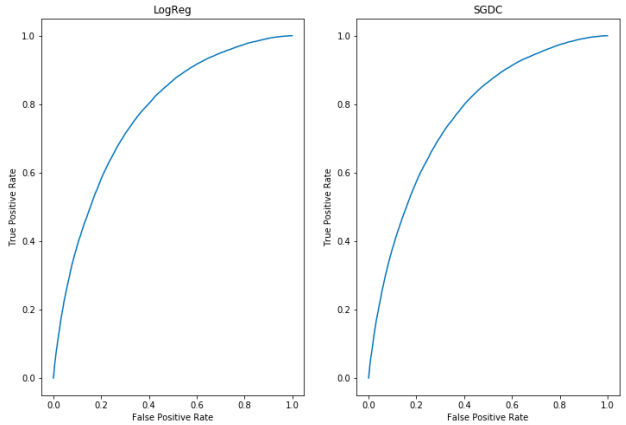
For brevity, the graphs showing the distribution of the selected features for all the tables used (bureau,bureau_balances, instalments_payements, Credit_card_balance), is shown in the appendix. The process described earlier was used for the remaining tables in our dataset.

IV. Evaluation

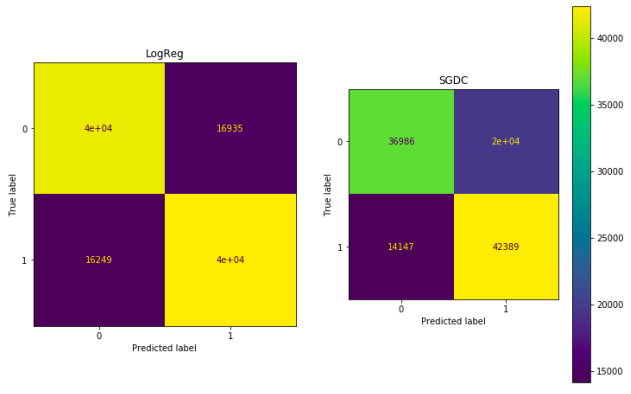
We attempted to use various models such as random forest classifier, logistic regression, SVM, SGDClassifier, and KNN. Due to the computation, the only two contestants' models



(i.e., models that could run with reasonable time in our local computers) were Logistic Regression and SGDClassifier. Without tuning their parameters, both models' performance is relatively the same, where Logistics' Regression accuracy is slightly higher but not enough to show an impact. Their ROC curves are nearly identical.



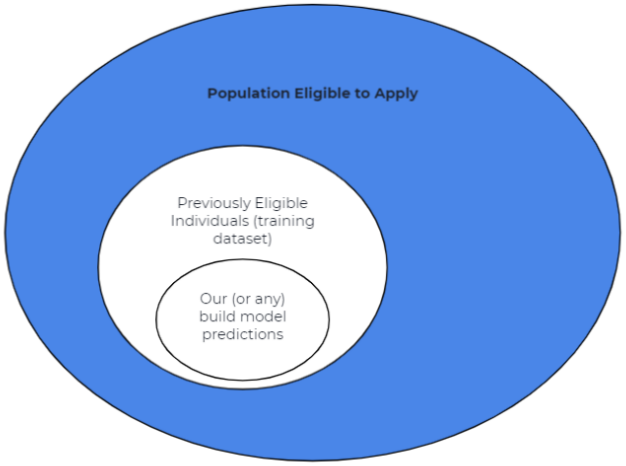
However, if we look at the heatmap of the confusion matrix for the two models, we can see that the logistic regression model outperforms the SGDClassifier model.



For instance, the confusion matrix for the SGDClassifer displays that this model misclassifies the clients who have payment difficulties by a more considerable margin. The objective of this project is to identify the clients who will have problems paying their loans. Therefore, Logistic Regression will be a better choice. Furthermore, after optimizing its' parameters using grid search, we could score 74% accuracy on unseen data on the Kaggle platform.

V. Conclusion

We were able to achieve a score of 74% (%80 is the best score in kaggle), using 140 columns out of 422 columns (we didn't get to all of them). This data set is unique from usual ML problems we have encountered because not a single feature contributes more than 1% (outside of EXT_SOURCE). Therefore, getting a higher prediction really depends on analyzing every column of the dataset or trying to feature engineer multiple columns into one. One of the problems with this dataset that may prevent the model from being as robust for future prediction is that the data may be biased due to different behaviors such as those from COVID/quarantining. The data has behavior patterns from 2020 which can be drastically different from those pre/post covid. For example, throughout the course of 2020 banks were more reluctant to lend out money due to increased risk as many people lost jobs etc. (According to a report from Experian 4/1/21) In the future, it will probably be best to re-train the model while omitting any data from the covid period to get a better representation of normal/average lending behaviors. In addition, this dataset assumes that the organization's initial/previous loan model is ideal and therefore no one in the excluded population (in the blue background) is qualified.



Therefore, any future iterations of the model will only target the sub sample previously qualified. It assumes previous qualifications are perfect. Maybe there were errors in previous qualification models and there can exist a small percentage of those that can now qualify with new training techniques?

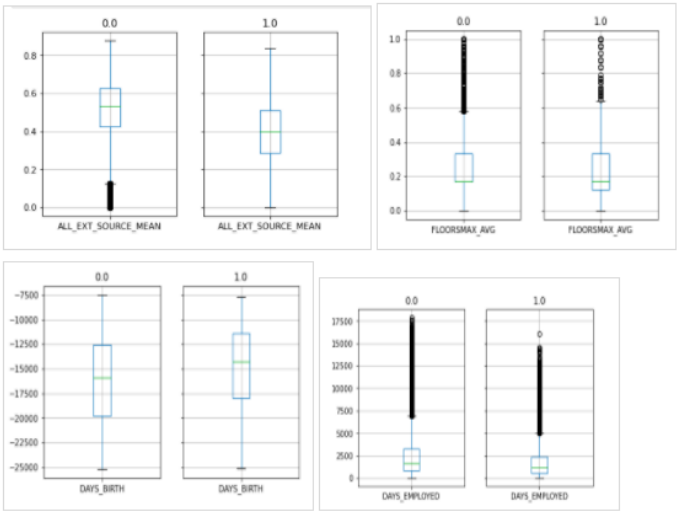
VI. Bibliography

1) Household debt rises to $14.6 trillion due to record-breaking rise in mortgage loans - https://www.cnbc.com/2021/02/17/household-debt-rises-to-14point6-trillion-due-to-record-breaking-rise-in-mortgage-loans.html

2) American Debt: Mortgage Debt Reaches $10.04 Trillion in Q4 2020 - https://www.investopedia.com/personal-finance/american-debt-mortgage-debt/
3) What Credit Score Do You Need to Buy a House in 2021? - https://www.quickenloans.com/learn/credit-score-to-buy-a-house

VII. Appendex

Some of the top contributing features had different means/distributions when broken down by target. For example, these features were the top contributors to the target variable and the box plots on the next page shows why.



In addition to checking for correlation between features and targets, the differences in mean divided by standard deviation was a good metric to sort by in order to identify top performing features (which sums up, to some extent, what the box plot visualizes).