**Analysis of Home Credit Group Credit Default Risk**

*Nirav Sheth*

**I. Introduction:**

Mortgages or home loans are a 5-30 year loans given to consumers to typically purchase property. Before the mortgage lender provides the fund, they must predict if the applicant will be able to pay back their loans with a variety of credit and related information. On the other side, the applicant purchasing the home wants to find a loan that suits their needs. Traditionally, loans are provided to applicants with adequate credit history. However, the unbanked population, consumers without adequate credit history, will go to an untrustworthy lender whom could take unfair advantage of their situation because reputable banks will not provide them with loans.

However, there could be other ways to determine the credit worthiness of applicants. For this project, I will be testing if non-traditional data sources of credit history can determine if the applicants will repay the loan?

**A: Client:**

Home Credit Group goal is to broaden financial inclusion for the unbanked population. Their vision is to provide better services to the unbanked population to ensure fair and equal opportunities to purchasing property. Overall, if they can better predict the likelihood of loan repayment, then they can expand their services to a larger population and reduce their cost to service the loans.

**B: Data:**

The primary data provided in CSV format by Home Credit Group are the current applicants financial information, previous applicants, available and relevant information by the Credit Bureau, remaining balances on existing loans, and credit card information. The current applicants has about 308K applications. In addition, they also provided ~2M to 13M lines of supporting credit history information.

A dataset has a mix of behavioral, descriptive, and credit history. The main difficulty is that about half of the clients do not have historical information, so the use of behavioral and descriptive data will be an important source to judge the clients.

**C: Methodology:**

I will need to develop the dataset for the machine learning by using data wrangling techniques, exploratory data analysis, statistical analysis, and machine learning. A common connector of data will be the applicant ID generated per client. The next steps will be to gain some deeper understanding of the data through statistical analysis and generate some baseline knowledge. Then I will use supervised machine learning to generate a probability of loan repayment. The probability for each applicant will help determine the likelihood of repayment, but the actual decision is binary (yes or no). Therefore, I will plot the probability results onto a histogram and determine the cutoff value or the value that determines full repayment of loan. The optimal cutoff value will be determined based on the value of the ROC to the application test dataset. In the end, a binary classifier will be given to each applicant to determine if they will repay the loan.

**D : Deliverables:**

The deliverables for this project are finalized proposal, repository on Github, explanation of methodology with code, and finalized report.

**Capstone Link**

<https://github.com/nervster/CapstoneProject1>

<https://www.kaggle.com/c/home-credit-default-risk>

**II. Data Cleaning**

My datasets consists of seven fairly clean because it came from Kaggle. The seven datasets are one current applicant dataset with six historical informational datasets such as the data from the Credit Bureau, credit card transactions, previous application, etc. An important cleaning step was to aggregate the historical datasets to convert from the ‘1:many’ to ‘1:1’ relationships, so I will can merge it into the current applicant dataset. For example, one current applicant could have multiple previous loans.

My first step was to drop any rows that were not required for my application dataset. Therefore, I brought in all six historical informational datasets (ranging from 500k rows to over 6M) and then dropped any row that did not match SK\_ID\_CURR with the current applicant dataset. This was a great benefit to the performance of the model.

Next, I used Pandas’ pivot\_table function to help with the aggregation. For the aggregation via the ‘aggfunc’ parameter, I created a function called ‘aggfuncdict’, which created a dictionary of various aggregation methods depending on the columns’ format. This step was important because it will allow the machine learning algorithms to determine the important columns instead of myself, which can remove the biases I may have.

As an example, the Previous\_Application dataset has 37 columns with float, int and object column formats. First, I dropped all the rows by finding all the SK\_ID\_CURR ‘isin’ (which is a Panda function) the applicant dataset. Next, I aggregated the 37 columns with min, max, sum and Numpy’s mean methods if the column was numerical. If the columns were object based, then I used ‘lambda x: len(x.unique())’ to count the unique categories that each applicant was part of. I repeated the same thought process for each of the datasets. As a result, I ended up with a dataframe with over 400 columns with a mix of Float, Integer and Object data types.

**A: Missing Values/Null Values**

This dataset has an extremely large amount of missing values (ranges from 99.6% missing values to 0%) due to the aggregation step and lack of historical information for current applicants. To reduce the missing values or null values, I removed any columns with over 50% missing values, which is an arbitrary threshold. Next, I used the fillna method with the ‘inplace=True’ parameter to replace all null values to ‘0’ before merging the datasets together. For simplicity, I decided to use ‘0’ as the dataframe’s null value because of the high volume of ‘0’ within my dataset. This step is important because it will help allow the machine learning algorithms to see which filter our the columns with large amount of null values or 0 in this case.

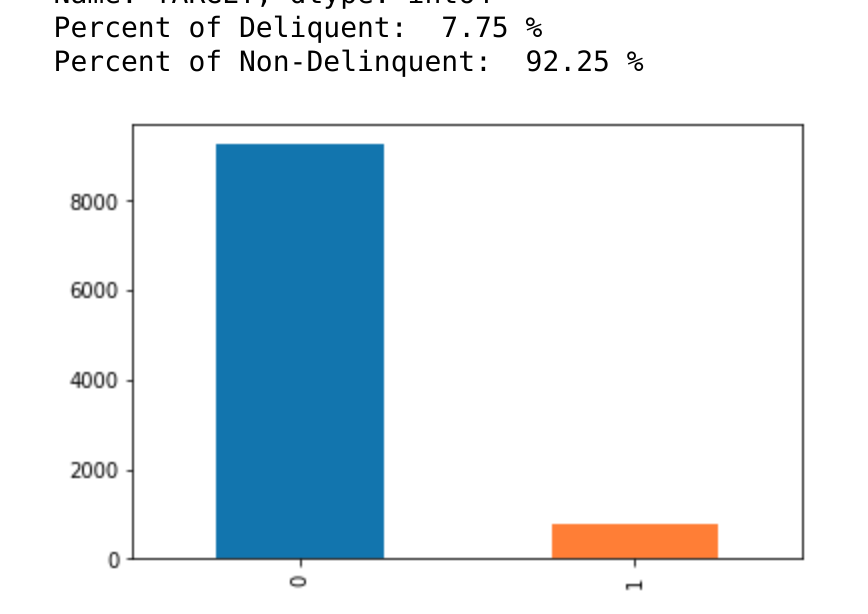
**B: Outliers/Feature Elimination**

This step would allow me to narrow the 400 columns with a mix of Float, Integer, and Object data types to a management amount. I wanted to understand which columns impacted the overall score the most, so I can dive deeper into the dataset with EDA. Therefore, I decided to use the ‘SelectFromModel’ meta-transformer with the ‘LassoCV’ classifier.

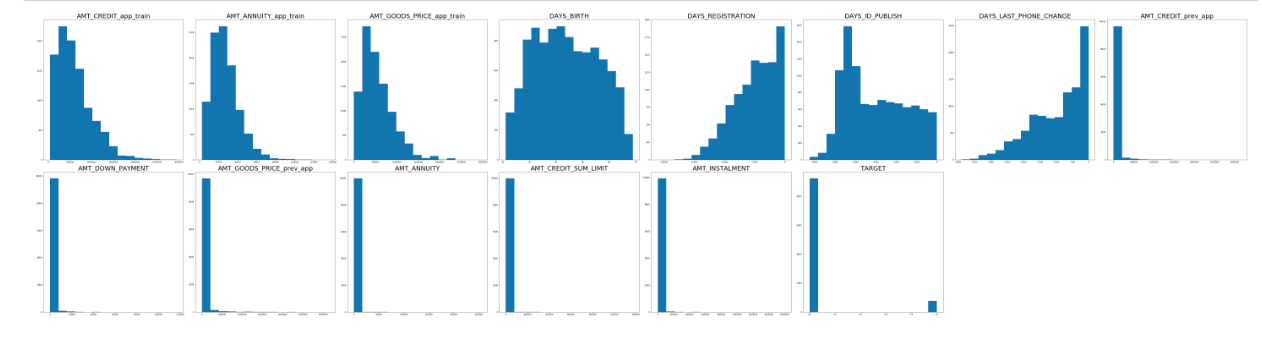
The LassoCV works with only numerical data types, so I used the Pandas’ get\_dummied method to convert all of my ‘Object’ based columns into a numerical. I was then able to use the SelectFromModel with a 1E-8 threshold limit to reduce my dataframe to 15 columns. From here, I was able to gain a better understanding of what the data means and gain additional information about the applicant pool.

**C: Data Storytelling**

After the aggregation and merging, my data has many incomplete values but I believe that is important to the overall story because we are trying to gain a better understanding of the group of people without credit information. I started by creating a correlation graph to the bar chart. I also wanted to see how many delinquent loans the dataset had. Upon further research, I noticed that Home Credit Group had twice the amount of delinquent loans than the United States’ national average. (Source: [Delinquency Rate for US](http://www.worldpropertyjournal.com/real-estate-news/united-states/irvine/corelogic-loan-performance-insights-report-june-2018-frank-martell-mortgage-loan-delinquency-rates-serious-delinquent-mortgage-loans-in-2018-real-estate-news-10920.php)) This information is useful because I can create a business case by determining if the model may have lowered their delinquency rate, which I could potentially translate to real value to the business. Based on the data below, the client’s delinquency rate is 7.75%.



Other characters about my dataset is that the average age of my population is about 44 years old. I compared the delinquent loans age to non-delinquent and noticed that younger generation are more likely to be delinquent. The lower income population tend to be less delinquent, but the price of the loan did not seem to have that large of an impact. Lastly, I generated a histogram plot of all loans. I will be able to use this plot to refer back to during the later steps.



**III. Data Storytelling**

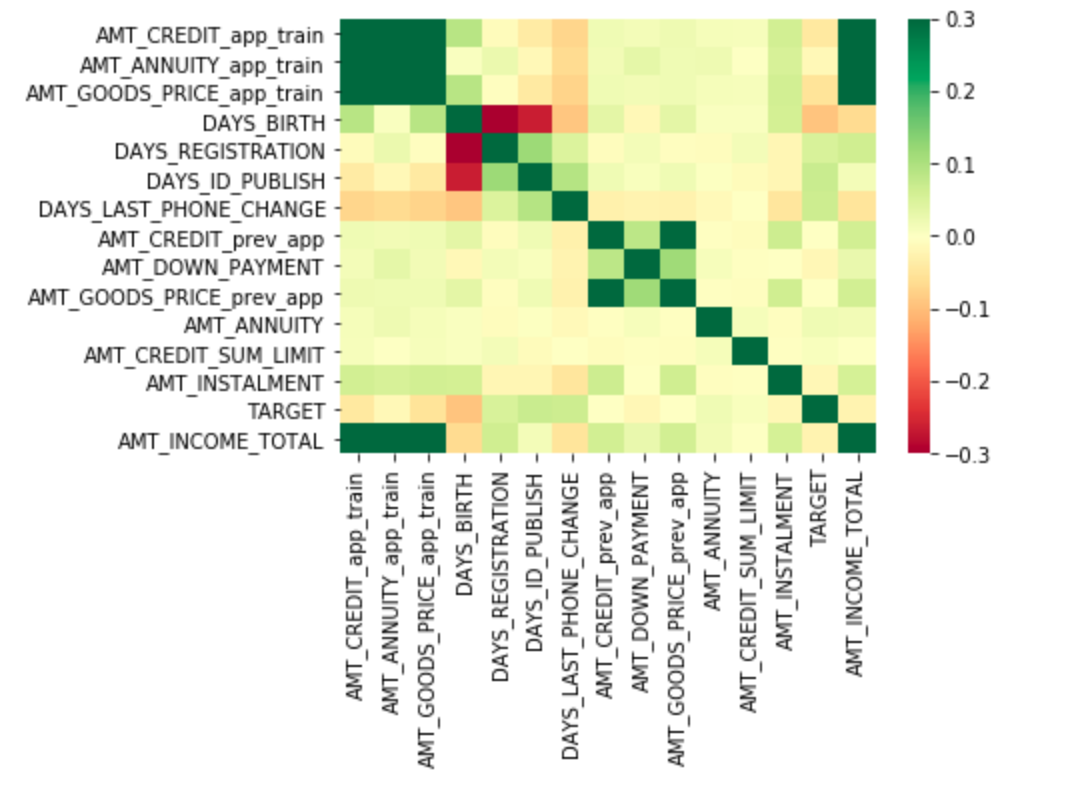
**A: Are there variables that are particularly significant in terms of explaining the answer to your project question?**

Using the Pearson R Correlation method, I have identified 4 columns that are extremely important to the TARGET data.

1. DAYS\_ID\_PUBLISH: This columns identifies the number of days the client changed the identity on the application. This column has a Pearson R correlation of .071. This dataset shows that clients with payment difficulties tend to have a larger or changed their application closer to the days published.
2. DAYS\_LAST\_PHONE\_CHANGE: This columns identifies the number of days the client changed their phone number. This columns has a Pearson R correlation of .067. This dataset shows that clients with payment difficulties tend to have a larger or changed their phone closer to the application date.
3. DAYS\_REGISTRATION: This columns identifies the number of days the client changed their application. This columns has a Pearson R correlation of .053.
4. DAYS\_BIRTH: This column identifies the client’s age. This columns has a Pearson R correlation of .093. We tend to see younger clients that are more delinquent on their loans.

This columns are quite interesting for the fact that columns 1-3 are behavioral based columns while column 4 is descriptive based. While the correlation is low, it is the highest of the features.

**B: Are there strong correlations between pairs of independent variables or between an independent and a dependent variable?**

****

Using the Seaborn library and heatmap function, I plotted the Pearson R correlations onto a heat map and saw a few observations. I decided to use .3 to -.3 as the range because this dataset has very low correlations, so it allows us to see the correlations in further detail.

1. Positive Correlation:
   1. AMT\_CREDIT, AMT\_ANNUITY, AMT\_GOODS\_PRICE, and AMT\_INCOME
   2. AMT\_INSTALMENT, AMT\_GOODS\_PRICE, AMT\_CREDIT, AMT\_ANNUITY,
2. Negative Correlation:
   1. TARGET against AMT\_CREDIT, AMT\_ANNUITY, and AMT\_GOODS\_PRICE.
   2. DAYS\_BIRTH against AMT\_CREDIT, AMT\_GOODS\_PRICE, DAYS\_REGISTRATION, DAYS\_ID\_PUBLISH

While there are strong correlations, I do not see many low correlations. With that said, I believe that identifying how the TARGET doesn’t correlate with the size of the loan is helpful to narrow the scope of where to help the business to focus their efforts. Overall, it seems like the delinquencies seems to be a behavioral factor. I will identify further testing going forward.

**C: What are the most appropriate tests to use to analyse these relationships?**

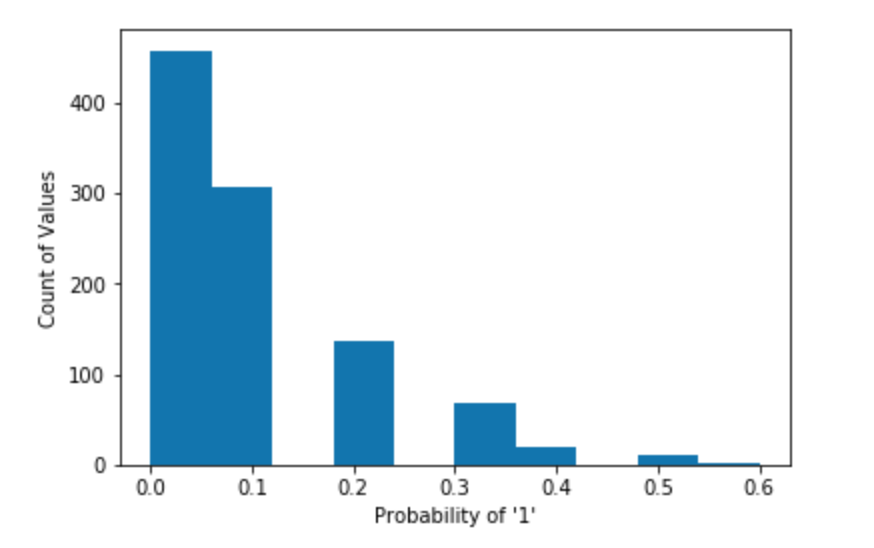
I would like to test how behavioral factors affect delinquency. To design this test, I would use columns 1-3 (identified above) against TARGET. I would Z-Test to test the difference of means between TARGET with 0 and 1 per column is significant. In other words, I will split each columns into 2 group: first column with TARGET identified with 0 and second column with TARGET identified with 1. The null hypothesis would test if there is no significant difference and the alternative hypothesis would test if there is a significant difference.

**IV. Machine Learning Models**

We will explore different machine learning algorithms to test how well a model can predict delinquency. To use the following classification models, I needed to complete two main steps. First, I ‘dummied’ the data using Pandas’ get\_dummies function. Second, I used the sklearn.model\_selection.train\_test\_split function with an 80-20 split to create the train and the test data set. To test my models, I used an AUC score because it helps compare my model to others on Kaggle. However, I believe an F1 score would be better served for this model because it is an imbalanced data set. Therefore, I used AUC and F1 score to judge the outcome of the model.

**A:** **Random Forest Classification: Generate Baseline Target**

I started with a Random Forest Classification Model because it allows to test many highly and low correlated features together. Therefore, I was able to use my greater than 400 columns data frame to generate a basic baseline. To generate the baseline, I needed to fit, predict, and score my classifier. For fitting (or training the model), I used the default parameters because alternative parameters did not make a significant difference. For predict, I needed to use predict\_proba because the threshold value to classify a certain row as ‘1’ is between .2 and .3. The following graph is a probability of the ‘1’ classification:



Based on this graph, I was able to see that .5 would not be a good threshold because it able to only classify less than 1% of my dataset as ‘1’. I decided to use .25 as my threshold to classify a row as ‘1’ or delinquent. With this change, I was able to generate a consistent score of .58 AUC and .19 F1 score.

**B: kNN Classification: Alternative Prediction Method**

Next, I tested my data set with kNN Classification to compare to the Random Forest and try alternative methods of improving my model. The benefits of kNN Classifier is that it is simple to implement and is able to handle a large multi-feature data set. Thus, I used the default parameters and generated a .51 AUC and .09 F1 score. While it didn’t provide me with a better result, it did provide me with a basic understanding that Random Forest baseline is most likely the largest score I will get without trying alternative methods.

**C: Alternative Machine Learning Methods: Hyperparameter Tuning**

I wanted to use hyperparameter tuning with GridSearchCV to try to improve my model’s AUC and F1 score. While it is computational expensive, this step is important to optimize my model. With kNN, I wanted to test 4 main parameters:

1. n\_neighbors: the number of neighbors needed for classification

2. leaf\_size: search parameter for the algorithm parameter

3. weights: determines the importance of distance to the nearest data point

4. algorithm: different methods to compute the nearest neighbors classification

I decided to choose these parameter based on research on the most common parameters to tune. Based on the results of Hyperparameter tuning, AUC and F1 scores increased slightly to .53 and .11

With Random Forest, I wanted to test 4 main parameters:

1. criterion: determines the type of split

2. n\_estimators: number of trees in a forest

3. min\_sample\_leaf: number of samples needed to split a leaf node

4. min\_sample\_split: number of samples needed to split an internal node

Based on the results of the hyperparameter tuning, the AUC and F1 scores had significant differences at .65 and .23.

**D: Machine Learning Conclusion**

Overall, there are more classifiers I could have incorporated into the model, however, I believe the Random Forest and the kNN provides a basic understanding of the model at this stage. I believe that random forest performs better than kNN due to the amount of features in the model. In the next section, I will list additional actions I should take to improve the model.

**V. Conclusion and Future Work**

This model proves to show the complexity of underwriting of credit. The highest F1 was able to achieve was .2, but there are steps that may have helped increase the score. The reason why I will not perform these steps is due to time constraints.

1. Missing Values: The datasets arrived with majority of the features with greater than 50% missing values. In my model, I dropped any columns with greater than 80% missing values. Additionally, I should have taken a deeper look at what was causing the missing values. Next, I could have resolved the missing values either by continuing to drop the feature or using a fill method.
2. Outliers: I did not deal with outliers within my model. However, I would not be surprised if the outliers were causing the decreased performances. I believe I could use a quantile based outlier function, which removes outliers that are greater than 25% or 75% of the feature. During the EDA, my data had a decent amount of outliers, and I believe that additional steps with dealing with outliers will help.
3. Feature Engineering: I could use a standard feature engineer function to create additional features based on other features. Or, I could have combine a few common ratio based on my finance background. A few I could have looked into are outstanding debt, payment history, spending history, assets, net worth, etc.
4. Machine learning: I stuck to a few common classification models (Random Forest and kNN), but there are additional methods I could have used to boost my scores. First, I could have used the ensemble method. This methods combines a plethora of models into a singular model and uses a majority vote methodology to determine the classification. Second, I could have found additional classification models that better supported an unbalanced class. On Kaggle, many competitors used LightGBM and I am sure there are plenty more models that could have been implemented if I had some additional time.

I believe the biggest takeaway from this project is that it is extremely difficult to judge the credit worthiness of an applicant if financial data is not available. As of now, historical credit information is needed to determine the likelihood of repayment. However, the train dataset had many incomplete fields, which means the applicant did not fill out the form. I question if the model could improve if the applicant provided more information on the application. Furthermore, is the data provided adequate to judge the delinquency probability of the applicant? The credit market is extremely complex and the credit market is highly correlated to economic factors. This dataset does not have any time features available, so it was unable to be tied to external economic data. With that said, I wonder if this model would be more complete with external economic data.

Overall, we were provided 7 datasets from behavioral to descriptive to financial information that was provided for majority of the applicants. With those datasets, I was able to clean and merge the data together, so I could apply machine learning techniques to create a model that can predict the probability that an applicant will repay a loan. This model can be used as a foundational model which additional data science techniques can be used to improve the model.