Matt Tilley

Capstone 2

Credit Modeling in the Sub-Prime Market

**Problem**:

When people need extra money, they apply for a loan. In order to get a loan, one must meet some standards of credit worthiness. If one does not meet these standards, then one does not get a loan from a major institution. People who meet these standards typically have a relationship with a large institution that can outcompete new firms on price. This advantage makes it very difficult for a new company to loan money to people with established credit and assets. However, there is still a very large number of people who do not have access to credit but need it.

This is where Home Credit comes in. They are a company that looks at making loans to people who have traditionally been considered high risk. By using data outside of credit scores, they hope to find new customers that are thought to be high risk but actually are capable of paying back a loan. This new model could also be helpful in places where credit scores are not available such as developing third world nations. By using unconventional credit scoring means, the company hopes to discover profitable new customers in groups that previously could not get loans.

**Dataset: Where it was acquired and how it was cleaned:**

This data was acquired on Kaggle.com. It is from the 2018 credit modeling challenge by Home Credit. One of the first issues with cleaning it was the size of the data itself. There were several spreadsheets with millions of rows, so I could not load and merge the entire spreadsheets at the same time. To counter this, I took the first 100,000 unique ids from the training set. Several of the other sheets had duplicate entries though so I had to group each id together. Since these accounts were for different loan amounts, I decided to group them together by the mean of each id. After this, I merged all the sheets together to create one big sheet.

The next cleaning issue came from the large number of columns. After creating the consolidated sheet, there were 490 columns of data. This would take a very long time to go through manually so instead I decided to test each column for its information value.[[1]](#endnote-1) Information value takes a ratio of the positive and negative outcomes to determine how significant variables are. This removes variables that happen so infrequently that they would just be noise in the model. After using this formula, I kept all values above 0.02. The highest value was 0.335 which was described as a strong predictor but not a suspiciously strong predictor. This reduced the number of columns down to 76.

Now that the data was in one place and had been reduced to only the significant columns, it was time to fill in missing values. There were several key assumptions that were made when filling in the missing data. The first was to use 0 when dealing with missing values of money. Using any statistical value would indicate that the bank was missing millions of dollars in which the numbers being used in the credit model itself would probably be useless. So, we will trust that the accountants are not missing money but rather just didn’t record the 0.

There are several different types of loans that are listed in the spreadsheet such as home loans and credit loans. Not every customer has had every type of loan, so those missing values will be replaced with 0. This will help in our models but will make mean, median, and mode somewhat useless in the case of categories with few entries.

**List Other potential datasets that could be used**

One of the more interesting variables in the data set is ‘Emergency state mode’. It is a variable that happens less than 1% of the time but is still a significant factor. I think the model could also be improved by having overall GDP data during the time of the loan. This would allow for a more accurate understanding of who is a good borrower at what time in the economic cycle.

**Explain Initial Findings**

The three most related values were the 3 pieces of external data. I suspect that these might be credit scores even though it is never clearly specified. There were also more payment problems with younger people as opposed to older people. This could be due to older people having less need for debt (not as many student loans or taking on a new mortgage) as well as being more organized. People with higher education were also less likely to have payment problems.

Overall though, the correlations between any group and payment problems were very small with none being higher than 0.07 or lower than -0.07 so unfortunately there is no single smoking gun variable we can point to as a key determinant in an applicant’s ability to avoid payment problems. The same was true with our information value results. According to the medium article:

|  |  |
| --- | --- |
| **Information Value** | **Predictive Power** |
| < 0.02 | Useless for Prediction |
| 0.02 to 0.1 | Weak Predictor |
| 0.1 to 0.3 | Medium Predictor |
| 0.3 to 0.5 | Strong Predictor |
| > 0.5 | Suspicious or too Good to be True |

Our dataset consists of two strong predictors (external source 3 and 2) and one medium strength predictor (external source 1). The other 73 variables are weak predictors with 30 of them being lower than 0.03. If I am right that the three external sources are existing credit ratings, then there are no medium or higher variables to construct a new credit rating model. However, there are a lot of customers in this data set so maybe we are just looking too broadly.

VIF scores

Now that we have the final set of data, it is time to check for multicollinearity in the data. To do this, we will perform variance inflation testing on the remaining variables. In this process, we will check to see which variables are most inflating the variance by removing the variable with the highest VIF score until all VIF scores are below 5. This removes variables that are highly correlated with other variables so there is less risk of overfitting.

Random Forest Modeling

To create this model, we will be using random forest. This model will be especially helpful because there are so many variables with small significance. If we used logistic regression, the model would be very prone to overfitting. For the model, we will use 100 trees.

**Sampling Issues**

One of the biggest issues with this dataset is that it is very heavily skewed towards loans with no payment problems. As a result, the first time we ran the Random Forest Model, the model just guessed no payment problems every time. This led to a model that was accurate 91% of the time. While in theory, a model with 90% percent accuracy sounds like a good thing but if you told a board of bank executives to just give everyone money because they’ll usually pay it back then you would probably be asked to leave.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.91 | 1.00 | 0.95 | 16354 |
| 1 | 0.23 | 0.00 | 0.00 | 1618 |
| Avg/total | 0.85 | 0.91 | 0.87 | 17972 |

To fix this issue and make our model actually model, we tried two different approaches. The first was oversampling. This method keeps all the data from the dataset and replicates the rows with payment problems repeatedly until there is an equal number of 1s and 0s. The issues with this is that the model essentially memorizes the data and correctly predicts everything in the training set but doesn’t do well at all on a new data set. Because there are basically 10 copies of each row with payment problems, the data for those rows becomes very heavily weighted in the model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 1.00 | 0.99 | 0.99 | 16172 |
| 1 | 0.99 | 0.99 | 0.99 | 16516 |
| Avg/total | 0.99 | 0.99 | 0.99 | 32688 |

|  |  |
| --- | --- |
| Feature | Importance Score |
| External Source 2 | 0.0945 |
| External Source 3 | 0.0906 |
| Days Since Birth | 0.0594 |
| Age of ID | 0.0505 |
| Amount of Goods Price x | 0.0472 |
| Days Since Last Phone Change | 0.0454 |
| Seller place Area | 0.0452 |
| Amount Payment | 0.0450 |

Unfortunately, because the rows with payment problems were duplicated so much, the model just recognized those values and picked them to have payment problems. Once we take the model outside of the limited environment, it is likely that its accuracy will drop significantly.

The next method we will try will be under sampling. This drops some of the rows that don’t have any payment problems until there is the same number of rows between payment problems and no payment problems. This would make it seem like we are about to lose a lot of data. To counter this, we will randomly select the data to take from the larger dataset. Because it is random, we will still have the general overall characteristics of the population, but we won’t have to worry about overfitting the model. This method feels a lot more statistically sound. The bad news is that it had an accuracy of 67.4%. This is much less accurate than the other two models but is going to be the best model since it will work better on new data than the first two attempts.

Ideally, we would be able to re-create this model on a cloud, so we could run the entire data set that contained millions of rows. This would reduce the information loss from under sampling. When the data goes from 100,000 to 16,000 thousand and there are 70 independent variables, it is very difficult to train the model. If we were able to under-sample the full data set, we would still have several hundred thousand rows of payment problems to work with. We believe that this would greatly increase the accuracy of the model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.67 | 0.69 | 0.68 | 1626 |
| 1 | 0.68 | 0.66 | 0.67 | 1629 |
| Avg/total | 0.68 | 0.68 | 0.68 | 17972 |

Under Sampling Feature Importance:

|  |  |
| --- | --- |
| Feature | Importance Score |
| External Source 3 | 0.0897 |
| External Source 2 | 0.0889 |
| Days Since Birth | 0.0582 |
| Age of ID | 0.0512 |
| Amount Goods Price x | 0.0457 |

Validation:

To test the validity of our models, we will use five-fold cross validation. This process divides the data into two sets. One set is the training set while the other set is the testing set. By breaking up the data, we can create a holdout set to test the model on once we have finished training the model. As can be seen below, all three models were very consistent in the different tests.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data Set | 1 | 2 | 3 | 4 | 5 | Mean |
| Original | 0.919 | 0.919 | 0.919 | 0.919 | 0.919 | 0.919 |
| Over Sample | 0.997 | 0.998 | 0.997 | 0.997 | 0.997 | 0.997 |
| Under Sample | 0.675 | 0.670 | 0.680 | 0.669 | 0.673 | 0.673 |

Next Steps and Recommendations:

The next steps going forward will be to apply the model to loans made to historically high-risk customers. The model we recommend is going to be the model created by under-sampling (preferably with the increased amount of data and processing power.) The segment that this model will target typically does not have access to credit so there should be an open market with many business opportunities. By having better predicting abilities, the bank will be able to offer lower interest rates to those customers which can greatly increase those people’s quality of life.

1. https://medium.com/@sundarstyles89/weight-of-evidence-and-information-value-using-python-6f05072e83eb

   https://github.com/Sundar0989/WOE-and-IV/blob/master/WOE\_IV.ipynb [↑](#endnote-ref-1)