Capstone Proposal for Home Credit Default Risk Assessment

Sandhya Raghavan
January 7th 2019

Domain Background

Every bank has a several strategies to acquire customers for their different line of businesses like home loans, auto loans, credit cards etc. Before providing any form of loan to the customers the financial institutions need to assess the Credit Risk. Credit risk is the probable risk of loss resulting from a borrower's failure to repay a loan or meet contractual obligations. For the assessment of Credit Risk, the lenders calculate it based on the borrower's overall ability to repay. To assess credit risk on a consumer loan, lenders look at the five C's: credit history, capacity to repay, capital, the loan's conditions and associated collateral. Based on the customers value for the five C's they provide a risk score for the customers and decide on whether they should provide the loan or not and if provided at what interest rate. In the traditional methods, statistical learning used these values to predict the defaulters and this was in a more linear fashion. In todays world, when so much data is available there are variable factors, we can use to asses the customers and the points are not linearly separable to make predictions. A machine learning model, unconstrained by some of the assumptions of classic statistical models, can yield much better insights that a human analyst could not infer from the data. In this link you can see a clear example of how machine learning improves the defaulters prediction in Moody Analytics

Problem Statement

For the capstone project, I would like to try attempting to solve the Home Credit Default Risk Assessment problem posted as a Kaggle competition. Details of the competition are available here. Home Credit strives to broaden financial inclusion for the unbanked population by providing a positive and safe borrowing experience. As explained in the domain background using the five C's and performing statistical analysis to determine credit risk scores has always been the traditional approach which financial institutions. Home credit would like to use machine learning for this task, so that we could consider as many features as possible not just the five C's to determine whether the customer is a potential credit defaulter or not. This is clearly a classification problem to classify customers as credit defaulters or not. The inputs to this problem will be data containing the details of the credit history and payment cycles and balances of customer and the output will be a probability prediction wit value between 0 to 1 if the particular applicant is a defaulter or not , with 1 representing a 100% defaulter.

Datasets and Inputs

Below are the available source files provided in the competition

- application_{train|test}.csv : This is the main table, broken into two files for Train (with TARGET) and Test (without TARGET). One row represents one loan in the data sample.
- bureau.csv: All client's previous credits provided by other financial institutions that were reported to Credit Bureau (for clients who have a loan in the sample). For every loan in our sample, there are as many rows as number of credits the client had in Credit Bureau before the application date.
- bureau_balance.csv: Monthly balances of previous credits in Credit Bureau.
- POS_CASH_balance.csv: Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
- credit_card_balance.csv: Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
- previous_application.csv: All previous applications for Home Credit loans of clients who have loans in the sample.
- installments_payments.csv: Repayment history for the previously disbursed credits in Home Credit related to the loans in the sample.
- HomeCredit_columns_description.csv: This file contains descriptions for the columns in the various data files. This is just a data dictionary and we would not be using this for assessment.

All the files above (except the data dictionary) will be used by the project.

Solution Statement

The default risk problem is a classification problem where we would need to classify each customer as a defaulter or not. The solution to this kind of problem is to do an exploratory analysis on the fields provided in each of the input datasets and find the important features. And then using those features we can use machine learning classification algorithms to learn from the training dataset provided. We can measure the correctness of our solution by using the right evaluation metric (which will be discussed in a later section). Based on the metric scores if there is still room for improvement, we can try different algorithms for classification or we can try tuning the parameters of the existing model to attain acceptable performance.

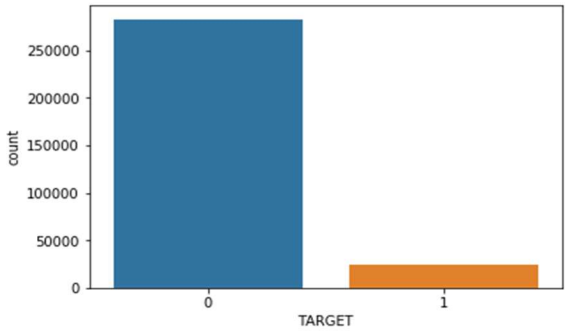
Benchmark Model

This being a classification problem we can use the Decision trees classifier algorithm as the baseline model. Why Decision Tress Classifier?

- They are well suited for these kind of problems
- They can be understood, they are more of a Whitebox model. Interpretation is very much required in financial domain.
- They can be validated by statistical methods making them more reliable. Using our chosen evaluation metric, we will be able to measure the performance of our decision tree model which will tell how well the model performs on unseen data.

Evaluation Metrics

The distribution of the 'Target' feature (Credit defaulter or not) is highly biased.



In the 'Target' field distribution, we can see that there are about 90% of 0's and 10% 1's. The evaluation metric provided in the Kaggle competition is area under the ROC Curve (AUC-ROC). This is definitely a good evaluation metric considering the target variable bias as it takes into consideration the sensitivity and the specificity of the predictions. The Receiver Operating Characteristics curve (ROC) plots the False Positive Rate (FPR) on the x-axis vs True Positive Rate (TPR) on the y-axis. The area under this curve depicts the degree of separability between the classes thus measuring how well the model is able to split the data into the right classes. If the AUC - ROC value is close to 1, it means the model is performing very well and if it close to 0 it means the model is performing poorly.
The TPR value is the Recall or Sensitivity value which is calculated as below:

$$TPR(or)Sensitivity(or)Recall = \frac{TP}{TP + FN}$$

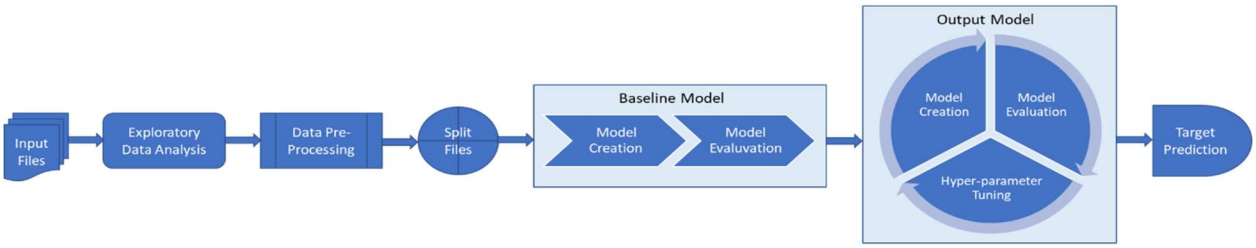
The FPR value is complement of the specificity which is calculated as below:

$$Specificity = \frac{TN}{TN + FP}$$

$$FPR = (1 - Specificity) = \frac{FP}{TN + FP}$$

Project Design

Below picture represents the design approach for the solving the Home Credit Default Risk Assessment Problem.



1. Exploratory Data Analysis: Before using the data for building our solution it is critical that we have a complete understanding of the data we have in hand. We would need to understand the below:

- Understand the dimensions of the data files and data type of the fields in each file
- Get a sense of the missing values in the data so that we can see how to handle them later.
- Profile the data fields to check the distribution of the values, checking the distinct values, get the key statistics for each field
- Check for correlation between fields as they could be either used to create a new feature or be handled correctly to not affect the model's performance
- Check for outliers and sparsity of the data as they will have significant impact on the model's performance if not handled correctly.

2. Data Pre-processing: Based on the information we have from our EDA, we can then work on the below pre-processing steps.

- Handling Missing Values
- Normalizing and Transforming fields
- Perform One hot encoding of features
- Merging the data files

3. Feature Importance and Selection: Once we have the data pre-processed, we can use the same to check which features are important for the model. We will calculate the feature importance in the merged files using feature importance parameter in a decision tree classifier and based on it make a feature selection

4. Splitting of Data: We need to split the data for training, testing and validation sets so that it would help analyze and improve performance so as to achieve a bias-variance tradeoff We will be splitting data into train test and cross validation sets using K-fold validation.

5. Benchmark Model Creation: We will create the benchmark model with Decision trees classifier as mentioned in the previous section. We will use the sklearn library for the same and since we are looking to predict probabilities, we will be using the predict_proba method for the same.

6. Benchmark Model Evaluation: We will evaluate the benchmark model with the AUC-ROC metric and see how well we are performing on the datasets. Based on the results we will make some minor adjustments to the model and see if we can improve the performance of the benchmark model further. If not, we would call the same the baseline and move with the solution for the project.

7. Output Model Creation: For predicting the credit defaulters, we will be predicting the probabilities and ensemble techniques are very efficient for these problems. The common ensemble techniques are bagging and boosting. For this project, we would be checking on both the techniques and see which performs better.

- Create a model using the Random Forest Classifier: This uses the bagging method
- Create a model using Gradient Boosting Machine Classifier: This uses the boosting method

8. Output Model Evaluation: We will then evaluate the models with the AUC-ROC metric and see how well we are performing on the training, cross-validation and testing sets. We would plot the results to compare and analyze the performance of the two models on the various datasets. Based on analysis we will check which parameters need further tuning to improve the performance of the model.

9. Hyper-parameter Tuning: Hyper-parameter tuning is the way we can minimize the error and improve the performance of the model. We would employ two different methods to do the tuning.

- For the Random Forest model, we will use the Grid Search technique to test different parameter sets.
- For Gradient Boosting Machine model, we will use automated tuning using Bayesian optimization Techniques.

We will create functions for all process from step 4 through 8 and we create a pipeline to do more of a rinse and repeat of the steps till the model provides a much better improvement in performance over our score from the benchmark model.

10. Prediction: Using the best model fitted we will predict the values for the test dataset provided by Kaggle and submit the same to see what our score is on the leaderboard.