**Capstone Project 1: Conclusions and Future work**

**Conclusions:**

In this project we predicted the probability of default for each current loan for Home Credit, based on aggregated information from previous loan histories belong to the same current loan ID, Bureau credit information, demographics and social network attributes of applicants, income and education level, living area conditions and so on.

Our model reports are based on 0.5 threshold when predicting if a future loan will go default or not, and all the confusion matrix, precision and recall, F1 score etc. metrics are also based on 0.5 threshold. However, the analysts in Home Credit can leverage the model predicted probabilities and adjust the threshold in their decisions to achieve better tradeoff between precision and recall rate, to fit their risk appetite and eventually making the correct decision to decline or approve a new loan. In addition to the F1 score etc. mentioned above, we also provide top 10 features for the analysts to understand feature importance and ROC curves to help choosing the threshold.

To achieve all of these, we started by aggregating each of the 5 data sources provided by Home Credit, and conducted exploratory data analysis on both categorical and numeric variables to visualize correlations between the target and features. Some features show better separation of default vs. non-default and others not so much. We compared the performances of 6 classification models and below is what we found.

1. The model with the highest AUC (0.7748) is LightGBM built on balanced training data. XGboost comes second and Logistic regression with L2 penalty ranks the 3rd place, followed by kernel SVM and random forest. KNN and linear SVM perform worst with AUC equals 0.6333 and 0.5715, respectively.
2. In terms of computation times, tree-based methods are generally faster than other methods. Logistic regression takes similar amount of time as tree-based methods. Kernel SVM is the slowest to train and predict, while the processing time for KNN and linear SVM lie in between.

**Future work:**

We realized that the precision for all of the models are not ideal, somewhere around 0.17. We believe more thoughtful feature engineering work needs to be done and more models can be tried. Specifically, below are some thoughts for future work.

1. Feature engineering can be further explored by looking into combinations of various columns, quadratic terms or using different aggregation methods at sub-ID levels.
2. In the final dataset used in modeling, Bureau balance dataset was not included in the features due to limited time and computation power. In future work, it can be aggregated and incorporated into the training data.
3. More missing value imputation techniques can be used to better impute the missing values based on overall shape of the distribution, rather than using the median for every numeric variable.
4. More sampling techniques such as up sampling of the minority class, or SMOTE can be used to compare with the performances with down sampling.
5. PCA can be leveraged to reduce the dimension of the features, however, model interpretability will be lost.
6. In this project we only tried to use scaled features in Kernel SVM model, to speed up the convergence of the algorithm. For all other 5 models we used the original scale of the variables, as tree-based methods can deal pretty well with different scales of variables. In future work we can try using scaled / normalized features in Logistic Regression and tree models as well, to see if there is any performance gain.
7. In all of the models we trained, we throw in all of the over 300 features without picking any subset of features. In Logistic regression, L1 penalty was selected, so we automatically got some feature selection benefits. Tree-based models naturally produce the rankings of features, so we can try to use the top 100 or so to train the model again to see if there are any performance improvements. In future work, more variable selection techniques can be considered to fit a more parsimonious model rather than using the full set of features.
8. In the SVM model cross validation step to select the optimal hyperparameters, due to the prohibitive training time of SVM algorithms, we only tried random search 2-fold cross validation with 3 random parameter combinations. More combinations can be tested to achieve possibly higher AUC provided with GPU or a high computation power CPU. Similar with tree-based models, more combinations can be tried when searching for best hyperparameters.
9. Other classification models can be tested such as neural networks, linear and quadratic discriminant analysis, Naïve Bayesian classifier, other tree-based methods such as Adaptive Boosting, or other bagging and ensemble methods.

**Reference:**

[1] https://www.kaggle.com/c/home-credit-default-risk/data

[2]

[3]