Predictive Analysis of HELCO Case

Contributors: Shiwei Chen, Yicheng Gao, Aijia Li, Lili Wang, Tianyao Xin

Case objectives

1.1 Background

Credit scores are an important factor that financial institutions consider when deciding whether to approve a loan. The scores are designed to predict the likelihood of repayment of a loan. In this report, our group aims to use machine learning techniques that offer the promise of increased accuracy with interpretability, which means greater access to credit for qualified borrowers and lower risk for financial institutions.

For HELOC, the primary goal of this model is to improve prediction accuracy. This includes precisely blocking those who do not have the ability to repay their loans on time to avoid bad debt. In addition, avoiding oversensitivity that results in blocking those groups of customers who can repay is also an important factor to consider. If some customers with good background and ability to repay are excessively intercepted, although the bad debt rate will decrease, it also means that HELOC will be more conservative in their daily operations, and the expected rate of return and market share will also decrease.

1.2 Objectives

We will develop a data-driven credit risk model in Python to predict the probabilities of default (Risk Performance) of existing or potential borrowers and build an interactive interface to sales representatives in a bank/credit card company. In this case study, we assume that the interface user has a working Python knowledge, as well as a basic understanding of certain statistical and credit risk concepts.

On this basis, we will classify the importance and meaning of features, and the degree of influence on the model through feature engineering. Considering that these features might be related to each other, we will further judge and process this to avoid bias. Besides, normalization or scaling of the feature to better adapt to the specific model are also used.

After this, we trained a total of 4 models, including decision tree, random forest, LGBM, and logit regression. After turning these models, the performance of these models in predicting HELOC data will be derived by referring to their prediction accuracy and ROC-AUC performance. Together with some consideration of the characteristics of these models, a final best fit model will be selected to ensure that it has a high accuracy and is easy to understand and interpret.

Feature Engineering

2.1 Overview of features

There are 10,459 observations in the dataset. The original dataset contains 23 features to measure banking activities of applicants. All features can be grouped by their practical meaning in financial activities. The groups by meaning are external risk, account opening, transaction pattern, delinquency, trading activity, installment, inquiry, revolving, and balance.

2.2 Feature engineering

For all features in the dataset, some of them are numeric, but others have categorical meanings. To improve the efficiency of feature utilization of models, we processed feature engineering at the first step.

Firstly, we would like to know the importance of each feature. Rather than compute the importance of features in different models, we compare the accuracy of new model without a feature with the original accuracy. Therefore, we tried to remove those features that have positive impact to accuracy. However, when the random state changes in the process of training set splitting, the impact of dropping out of a feature change totally. (See appendix “removed features impact”)

Thus, we can conclude that there is not a significant difference in the importance among features, which can be accounted for highly related in unknown relationships between all features. (They are not correlated in linear relationship) (See below “correlation heatmap”)

Secondly, we check the density distribution of every feature. Although half of them is normally distributed, some features are extremely right-skewed, whereas others are left-skewed. The skewness of features is shown in the appendix. (See appendix) Therefore, we transform the right-skewed feature “NumInqLast6M” by taking the natural logarithm of the values.

We also tried to reduce the left skewness of feature “PercentTradesNeverDelq”, but all transformations make accuracy worse. However, all these transformations do not improve accuracy significantly and the transformation makes model even more unstable.

Thirdly, we transformed all special missing values in the dataset that are marked as “-7”, “-8”, and “-9” by creating a series of dummy variables for each feature. As a result, we got a training set containing 34 features.

Model Selection

3.1 Model Overview

When we are selecting predictive models, four different statistical models are considered. They are Decision Tree, Random Forest, LGBM, and Logistics Regression. Some of these models are black boxes for decision making, while others give us some instructions for solving the problem.

Decision Tree

Compared to other models, decision tree model does not require normalization or scaling of the data. The decision tree's results will have relatively little influence on the process of building the model, even if the data has missing values. However, HELOC can easily obtain all the information needed from the applicant, so one of the advantages of this model will not be competitive. The instability and overfitting of the model are also disadvantages for decision making.

Random Forest

Random forest use bagging and ensemble learning techniques to solve the problem of overfitting by creating many trees and structuring their outputs on subsets of the data. This not only reduces variance but also improves accuracy. However, when random forests create more trees, it also takes longer training time and more computing resources. It is also relatively difficult to understand and explain, which has a negative impact on the user's understanding and interpretation of the model of HELOC.

LGBM

LGBM uses a histogram-based algorithm to speed up the training process and achieve lower memory usage by replacing continuous values with discrete bins. Since LGBM generates more complex trees by following leaf-level segmentation methods, it has a higher accuracy than other boosting algorithms. It is also worth mentioning that LGBM performs equally well on large datasets, and the training time is significantly reduced compared to other models.

However, since LGBM splits the tree leaf-by-leaf, it will generate a very complex tree, so there is a problem with overfitting. Although this problem can be avoided as much as possible by adjusting max_depth, LGBM is still prone to overfitting on small datasets.

Logistic Regression

In low-dimensional datasets, logistic regression is less prone to overfitting. Through the predicted parameters, one can infer the importance of each feature, including the positive or negative correlation. More importantly, logit regression allows models to be easily updated to reflect new data, via stochastic gradient descent. For HELOC, this is a very important advantage, which means that if HELOC wants to adjust the model, logit regression helps save time and labor costs. Moreover, logistic regression outputs well-calibrated probabilities along with classification results, inferring which training examples are more accurate for a formulation problem than other models that only take the final classification as the result.

However, on high-dimensional data sets, logit regression may be over-fitting, and regularization is needed to solve this problem, but excessive regularization will make the model more complex and difficult to explain. Logit regression also requires a large data set with enough samples, and if there are data values that deviate from the expected range, the algorithm can be sensitive to these outliers. It also pays more attention to those training examples that are correlated, thus requiring each training example to be independent of all other examples in the dataset.

3.2 Accuracy comparation

Result of Accuracy and ROC-AUC of best model for each grid search (see details in appendix).

Model	Best Model from Grid Search	ROC-AUC	Cross-Validation Accuracy
Decision Tree	DecisionTreeClassifier (max_depth=4, max_leaf_nodes=20, min_samples_leaf=100)	0.78	0.713
Random Forest	RandomForestClassifier(max_depth=6, max_leaf_nodes=16, min_samples_leaf=10, n_estimators=10)	0.80	0.728
LGBM	LGBMClassifier(colsample_bytree=0.8, learning_rate=0.02,max_depth=5, num_leaves=40,reg_alpha=0.03, reg_lambda=0.08,subsample=0.8)	0.81	0.735
Logit Regression	LogisticRegression(C=2.212216291070449, max_iter=10000, solver='liblinear')	0.81	0.738

The Home Credit (HELOC) will face losses if the model prediction is wrong in two scenarios. Scenario 1: If the model has predicted the client will repay the loan but he has defaulted. Scenario 2: If the model has predicted the client will default but he can pay the loan, the bank will lose in return interest. In reality, the loss will be much more in Scenario 1. Thus, False Positive Rate (measurement of scenario 2) and False Negative Rate (measurement of scenario 1) are important metrics to evaluate the risk performance of our existing or potential borrowers.

3.3 Model selection

The ROC-AUC score includes True Positive Rate and False Positive Rate. We take the accuracy, ROC-AUC score and confusion metrics (see appendix) as benchmarks in model selection.

Here we need to select the most suitable model from the four models with best estimators generated from the grid search analysis. The confusion metrics show that both Logit regression model and LGBM model have low FPR and FNR. Besides. In the table above, we can find that the Logit Regression model performs the best in both terms of ROC-AUC and Accuracy.

In this case, the aim of the decision support system is to lower the possibility of giving credit to people with bad records, which corresponds to the False Negative Rate in the confusion matrix.

Additionally, the system is to predict which category that a candidate belongs to, which corresponds to the binary function of logistic model. The non-multicollinearity of all features in this case fulfills the requirement of logistic model. The output of logistic model gives us a prediction of the probability that the candidate belongs to “bad” group, which provides more information when making financial decision compared to other models.

Therefore, taking accuracy, false negative rate, the logic behind the case and multicollinearity between features into consideration, we chose Logistic Regression as the most suitable model for the decision support system.

3.3 Summary

Based on the above considerations, we decided to choose the logit regression model, not only because it has both the highest prediction accuracy and ROC-AUC performance, which means it can avoid the prediction errors as much as possible, but also because it has suitable characteristics compared to other models that can well meet the business needs for HELOC.

Logistic regression outputs well-calibrated probabilities along with classification results, which coincides with the final judgment requirement of HELOC. The easy update and upgrade features can also better meet the needs of HELOC to update the model in the face of policy adjustments in the future.

	Decision Tree	Random Forest	LGBM	Logistic Regression
Basic logic	Split a population of data into smaller segments	Bagging and ensemble learning techniques	Histogram-based algorithm	Stochastic gradient descent
Overfitting	More prone to	More prone to	Less prone to	Less prone to
Noise influence	Very sensitive	Not sensitive	Not sensitive	Not sensitive
Data requirement	Low	Low	Low	High
Time consumption	Long	Medium	Short	Short
Interpretation	Medium	Hard	Medium	Easy
Limitation	Instable model; High variance; Fail to perform equally well on new data.	Training time and computational resources grow with the number of trees created.	Generate a very complex tree. Overfitting on small datasets.	Requires a large data set; Sensitive to outliers; Requires each training example to be independent.
Advantage	Less affected by missing values.	Less affected by the introduction of new data points.	Lower memory usage; Performs equally well on large datasets with less training time.	Easily updated to reflect new data; Outputs well-calibrated probabilities along with classification results.

Interface design

The interface is a decision support system targeting users in financial industries to evaluate candidates' creditworthiness or risk of default. It can give users scientific decision driven by data science and machine learning, in which the output is clear and easy to understand.

There are five important sections in the interface: input panel, output panel, data visualization, case-based explanation and appendix.

Input panel

There are 23 features to describe the financial and credit activities of a candidate. The user can input features collected by moving sliders or typing numbers in text boxes. Generally, all features should be greater than zero, but missing values marked as "-7", "-8", and "-9" are acceptable in this model. It can recognize special missing values and transform them into dummy variables before predicting. When entering features, the model automatically runs and returns the output in "Output" area.

Output

The output gives straightforward decision indicating whether the candidate is "Low Risk" or "High Risk". It also returns an overall risk score and the probability of risk for the candidate, and compares it with the average in database, showing by what percentage the candidate is riskier than the average. There is a plot displayed in the output area, in which the risk score is marked on the curve corresponding with its associated risk, so that the user can clearly know how the model gets the probability of risk to default.

Data visualization

In this part, all data distributions of features are shown in the form of box plot.

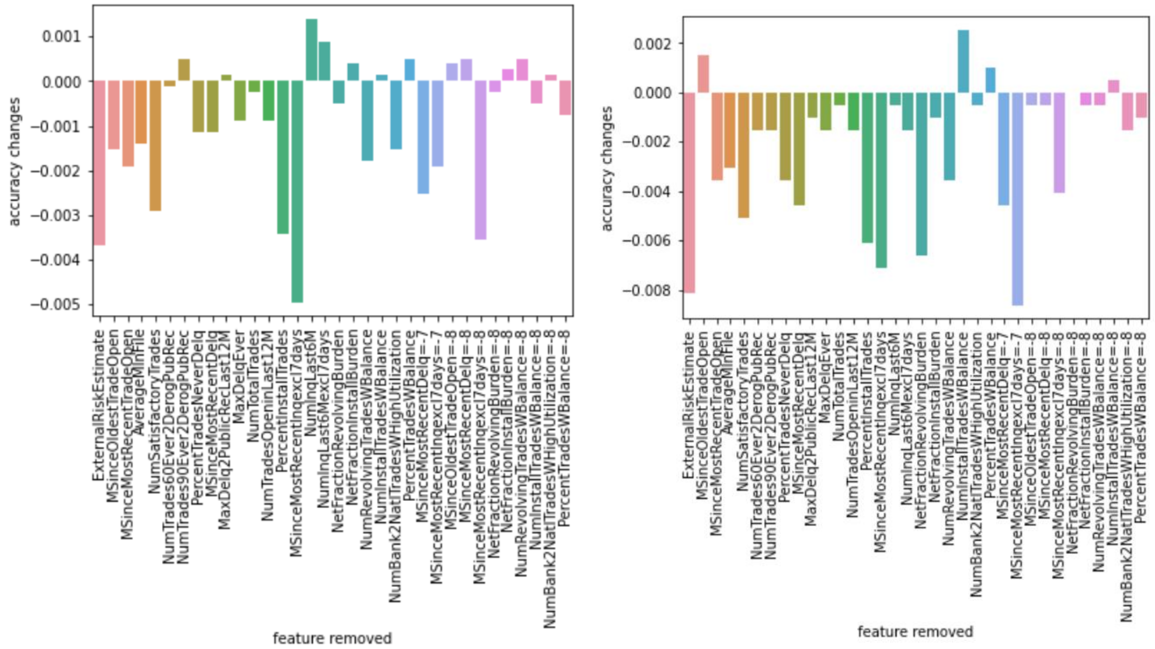
Case-based explanation

In addition to the logistic regression model, the interface presents five cases in historical dataset that are most close to features of the candidate input to the model. Knowing how applicants sharing similar features with the candidate behaved in the past, it would not be groundless for users to infer whether this candidate is good enough to get credit.

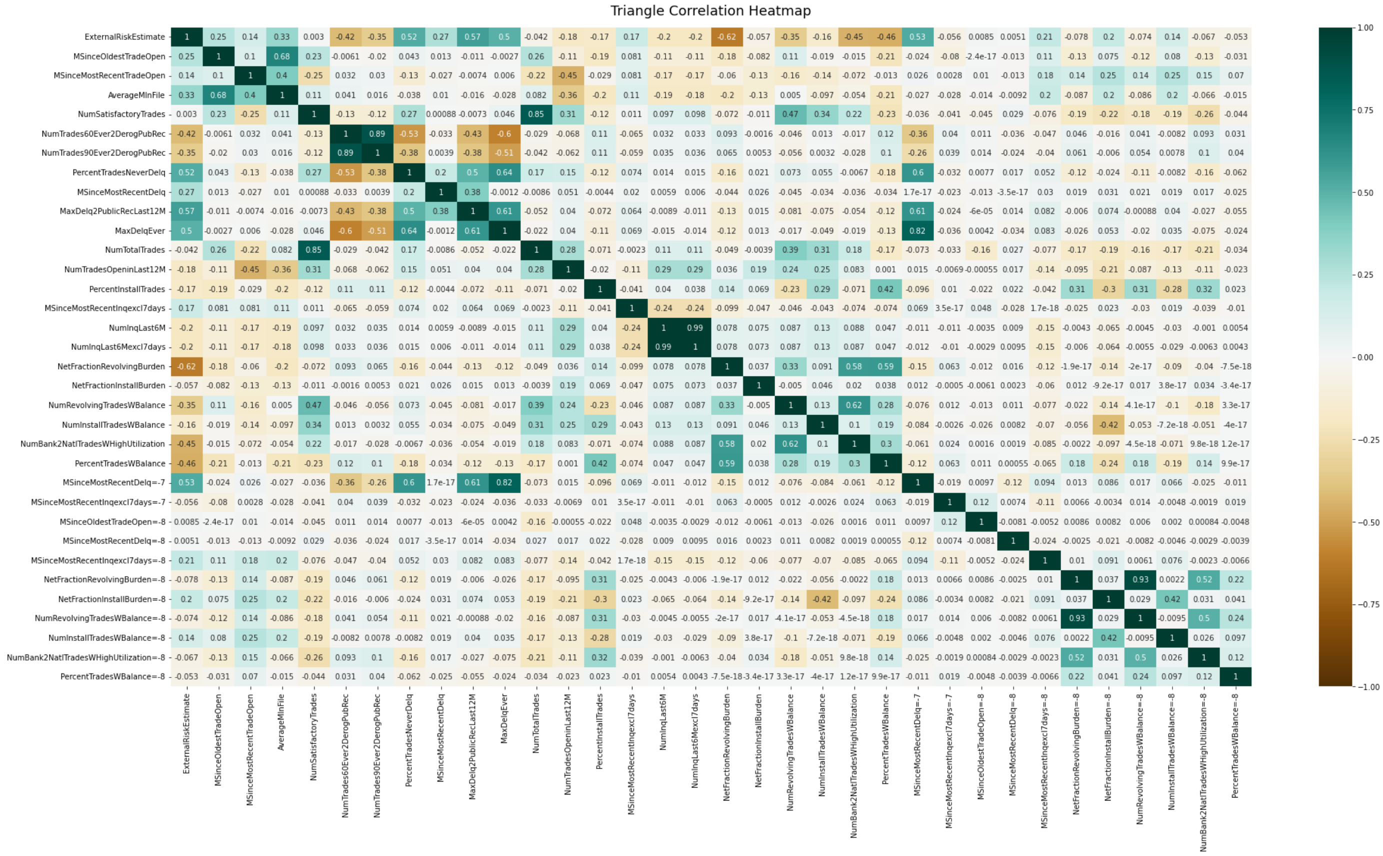
Appendix

Dictionary and explanation of all features and special values are attached as appendix.

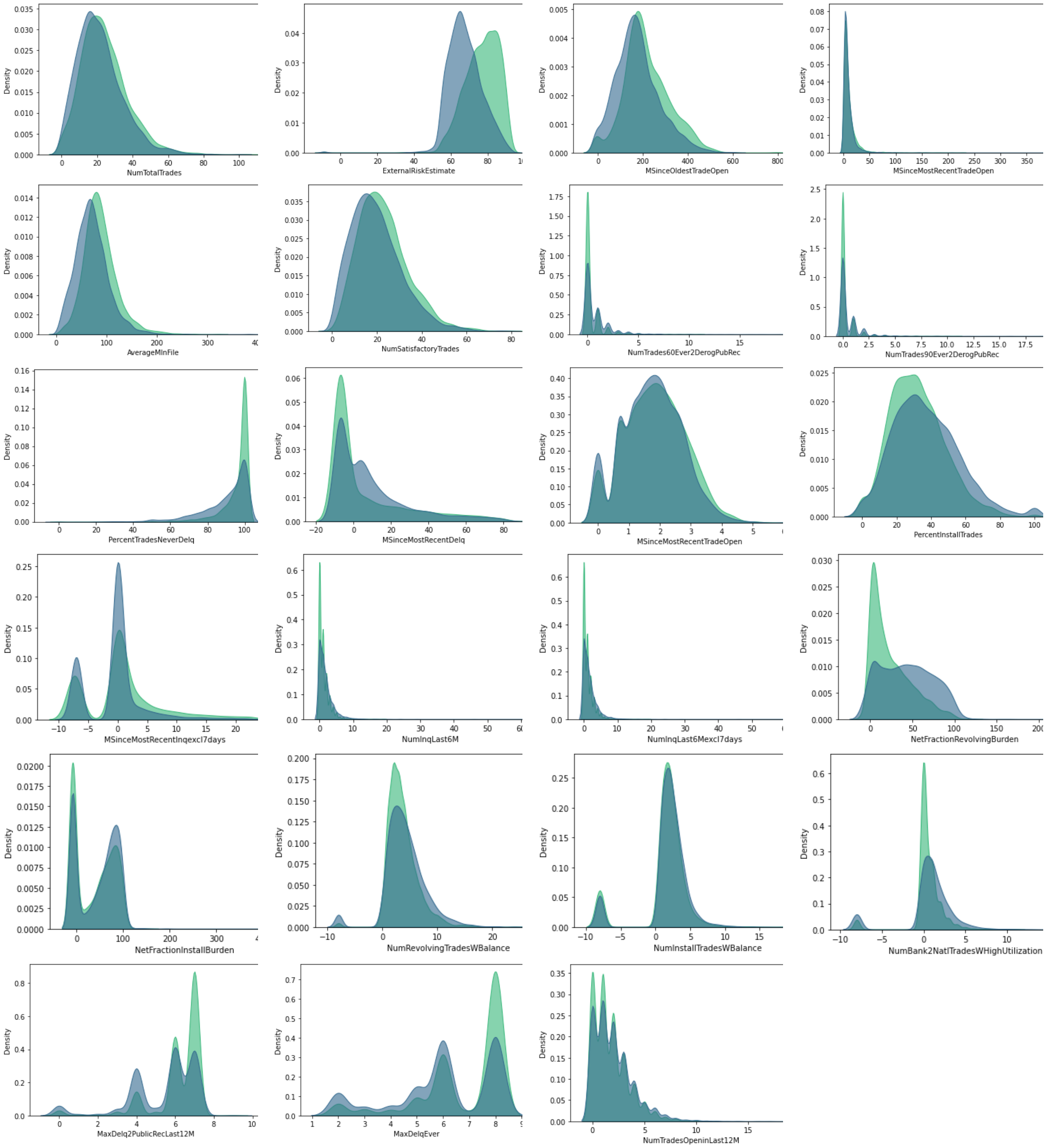
Appendix



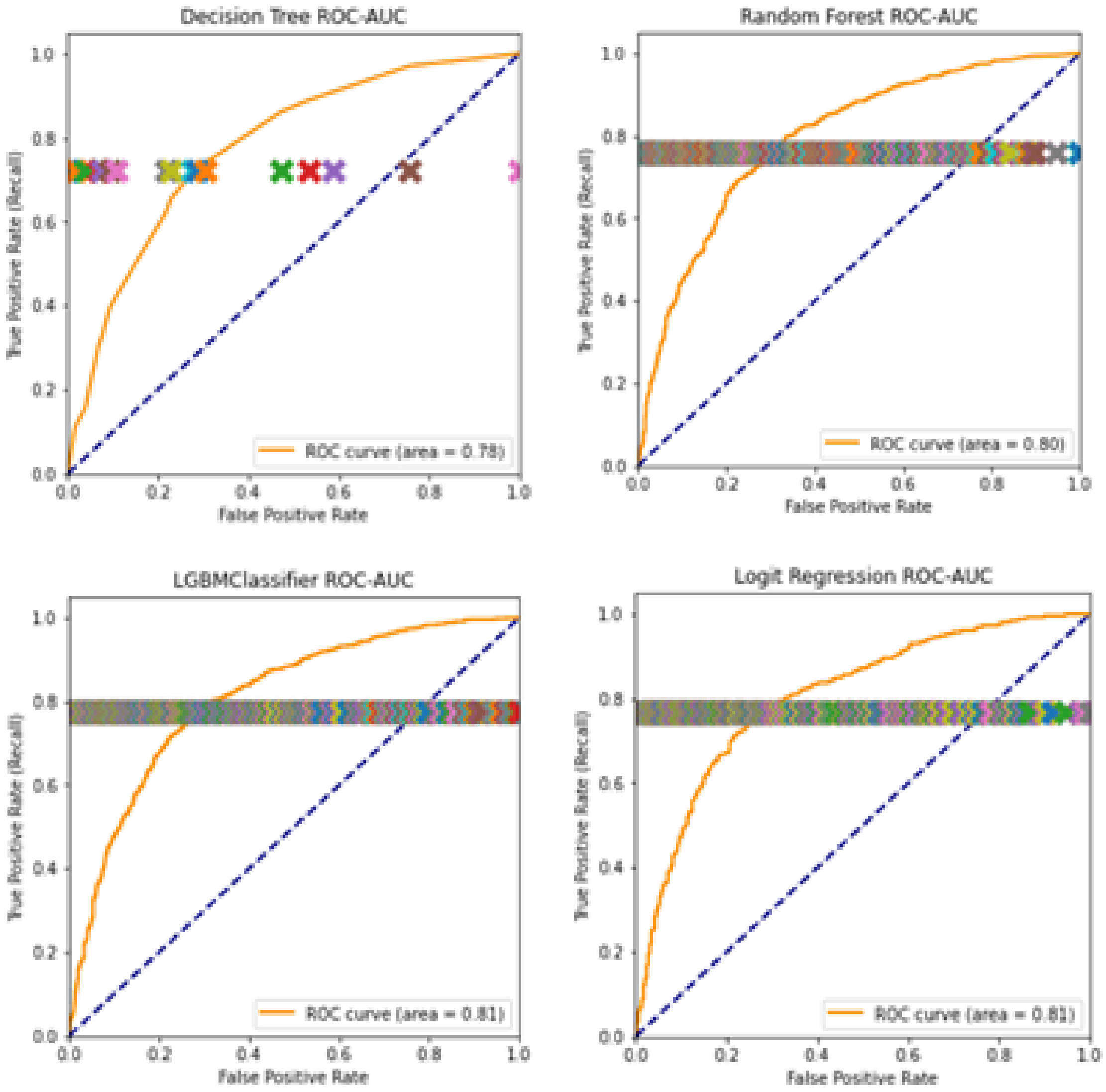
(Removed features impact)



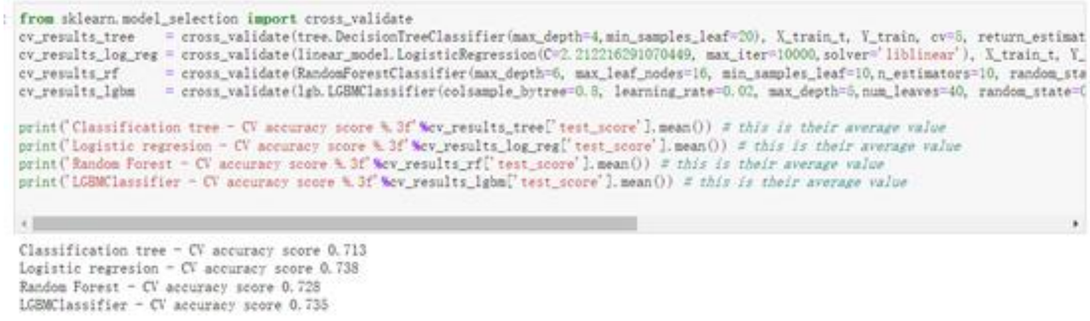
(Correlation heatmap for all features)



(Distribution of original features by Risk Performance groups)



(ROC-AUC Curves of Four Models)

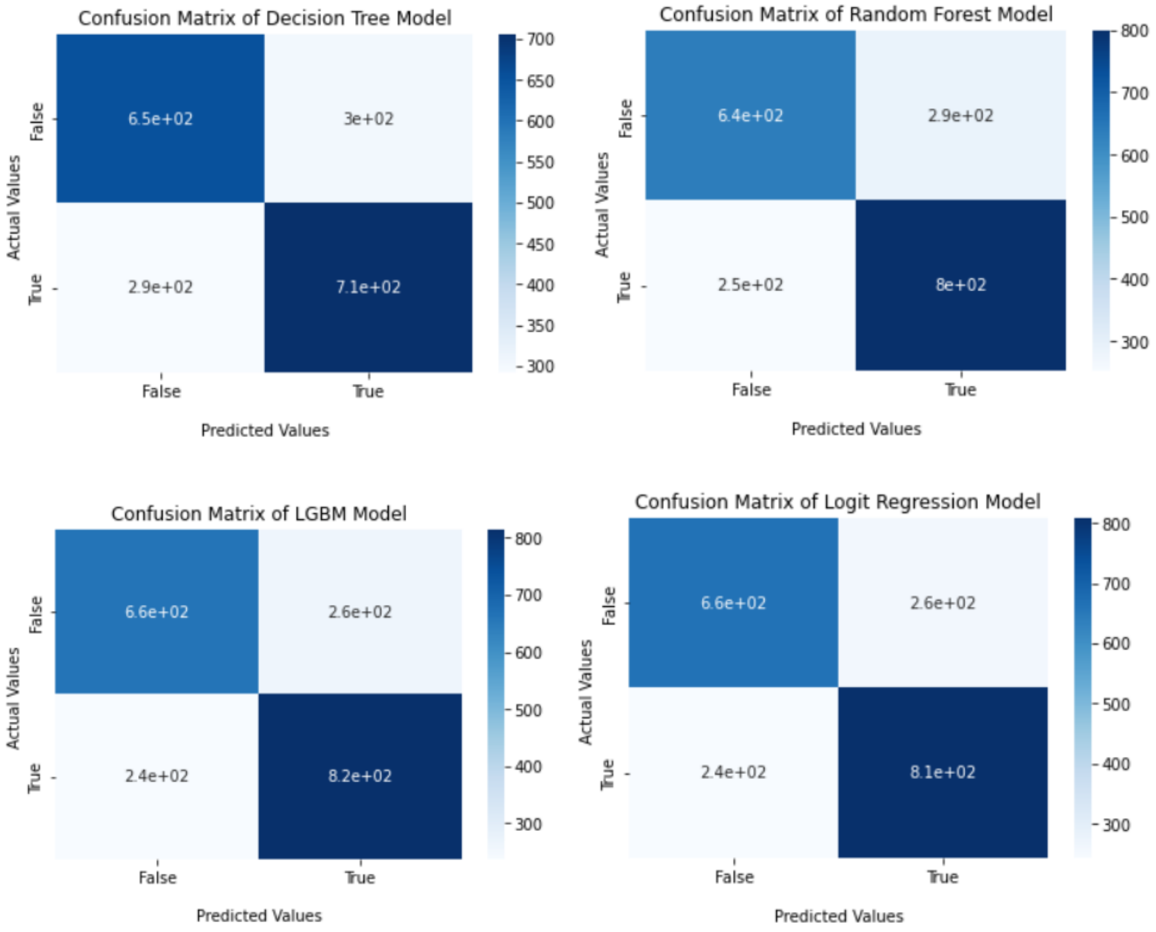


```
from sklearn.model_selection import cross_validate
cv_results_tree    = cross_validate(tree.DecisionTreeClassifier(max_depth=4,min_samples_leaf=20), X_train_t, Y_train, cv=5, return_estimat
cv_results_log_reg = cross_validate(linear_model.LogisticRegression(C=2.212216291070449, max_iter=10000,solver='liblinear'), X_train_t, Y_
cv_results_rf      = cross_validate(RandomForestClassifier(max_depth=6, max_leaf_nodes=16, min_samples_leaf=10,n_estimators=10, random_sta
cv_results_lgbm    = cross_validate(lgb.LGBMClassifier(colsample_bytree=0.8, learning_rate=0.02, max_depth=5,num_leaves=40, random_state=(

print('Classification tree - CV accuracy score %.3f'%cv_results_tree['test_score'].mean()) # this is their average value
print('Logistic regresion - CV accuracy score %.3f'%cv_results_log_reg['test_score'].mean()) # this is their average value
print('Random Forest - CV accuracy score %.3f'%cv_results_rf['test_score'].mean()) # this is their average value
print('LGBMClassifier - CV accuracy score %.3f'%cv_results_lgbm['test_score'].mean()) # this is their average value
```

```
Classification tree - CV accuracy score 0.713
Logistic regresion - CV accuracy score 0.738
Random Forest - CV accuracy score 0.728
LGBMClassifier - CV accuracy score 0.735
```

(Cross Validation Results of Four Models)



(Confusion Matrix of Four Models)