Predictive Modeling for Decision Support

CIS432 Summer

Predictive Analytics Using Python

Final Project

Team B16

Group Members

Zekun Li

Ruiqi Yao

Kecen Liu

Chengyu Dong

Mingjie Lai

1. The Big Picture of the Project

The main purpose of the project is to develop a predictive model and a decision support system (DSS) that evaluates the risk of Home Equity Line of Credit (HELOC) applications, ie. predict whether an applicant will be able to repay their HELOC account, and therefore, help to decide on accepting or rejecting applications.

The data used in the project is the ***HELOC Dataset***. After data exploration and preprocessing, cross validation was applied to different machine learing algrothms to find the best predictive models that yeilds the highest fitting accuracy.

A prototype of an interactive interface was then developed so that sales representatives in a bank/credit card company can used to make decision on whether accpet the credit line application.

2. Explore the Data

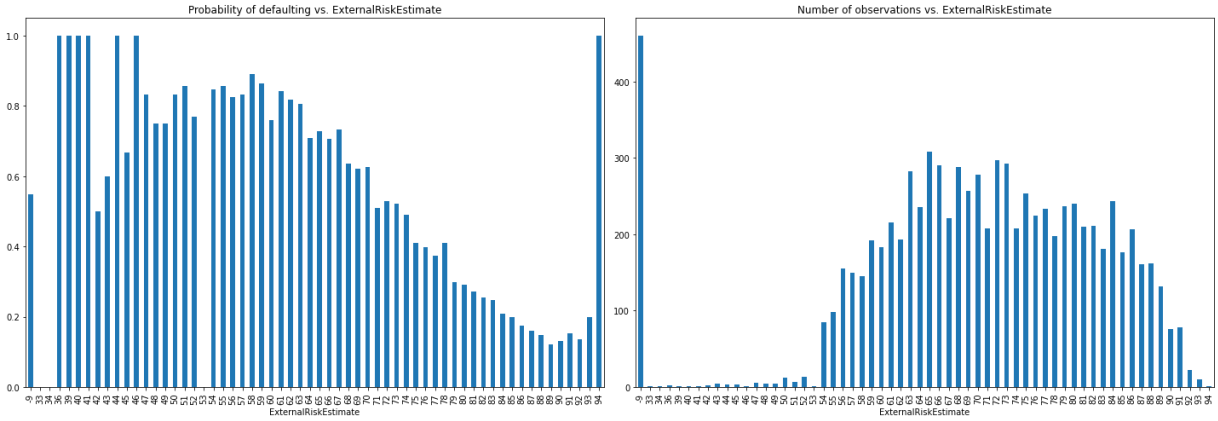
Data exploration was completed first to better understand the varibles and their features. It should be noted that the risk performance is the traget of prediction, which is a categorical variable. The value Bad means that consumer was 90 days past due or even worse. On the contraty, the value Good indicates that this consumer made their payments without being more than 90 days overdue. In the next analysis, Bad would be equal to 1, and Good is equal to 0. Other variables contain individual information like ExternalRiskEstimate, which could be helpful for making prediction. This variable is a numeric variable,and could indicate the other risk estimation for this consumer.

	RiskPerformance	ExternalRiskEstimate	MSinceOldestTradeOpen	MSinceMostRecentTradeOpen	AverageMInFile	NumSatisfactoryTrades
0	Bad	55	144	4	84	20
1	Bad	61	58	15	41	2
2	Bad	67	66	5	24	9

3. Prepare the Data for ML Algorithms

3.1 Replace missing values

After understanding the dataset, data was preprocessed for learning, which mainly focus on handling missing value and transforming data matrix. We firstly plotted the risk versus external risk estimates, and found that there are some missing values.



Considering that the data is going to be utilized for prediction model, instead of dropping the rows of missing value, the respective mean values are selected to replace missing values with means by applying the pipeline function.

3.2 Data separation for training and validation

After preprocessing, the data was separated into two parts: train and validation set, with 25% in the validation set, and the parameter random_state is set to a fix number 1234.

4. Model Selection and Tuning

4.1 Select the best model

Five kinds of models were chosen for machine learning and model evaluation, which are classification tree, logistic regression, KNN, SVM and Naïve Bayes. Accuracy score of the model was computed as the key metrics to evaluate the model performance.

Simple cross validation is completed to estimate the model fits first and the accuracy scores are as followed.

model type	accuracy score
Decision tree	0.643
Logistic regression accuracy	0.744
KNN accuracy	0.679
SVM accuracy	0.743
Naive Bayes accuracy	0.676

Five models are then applied to k-folds cross validation for more accurate evaluation of model fit. From the result, logistic regression yields the highest accuracy of 0.740 before tuning the hyperparameters.

We chose to focus on tuning four kinds of model which are classification tree, logistic regression, KNN and SVM. Grid-search method was then applied to find the best hyper parameters in each kind of model to compute the fit accuracy. The results were as follow.

model type	accuracy score before tuning	the best hyper parameters	accuracy score after tuning
Classification tree	0.633	{'criterion': 'entropy', 'max_depth': 4, 'max_features': 20}	0.715
Logistic regression	**0.740**	**{'C': 1, 'penalty': 'l2', 'solver': 'newton-cg'}**	**0.741**

KNN	0.677	{'algorithm': 'ball_tree', 'n_neighbors': 9}	0.691
SVM	0.732	{'C': 0.01, 'kernel': 'linear'}	0.740

4.2 Test the best model on the test set

Logistic regression model was selected for prediction for its highest accuracy score it obtains. This model is then used to finally test on the test set and the accuracy score is 0.723.

```
# so we choose logistic regression
# test on the test set
clf_log_regbest = linear_model.LogisticRegression(max_iter=10000,C=1, penalty= 'l2', solver= 'newton-cg').fit(X_train_t, Y_train)
print('Logistic regression accuracy: %.3f'%accuracy_score(Y_test, clf_log_regbest.predict(X_test_t)))
```

```
Logistic regression accuracy: 0.723
```

5. Develop an interactive interface for future prediction

Based on the model, we designed an interactive interface shown in the followed. We assume that all users need to know is that how to start the python script in Terminal to apply our predictive model. After running the Python script with streamlit, the user can input the value used for prediction by dragging the value bar, and the credit forecast for the situation will be given automatically.

Our model will give two results: 1&0, 1 for bad credit and 0 for good one. In this case, the predict value of the model is 1. In other words, the risk of this application is relatively high and our forecasting model does not recommend passing this application.