CIS432
Predictive Analytics
Fall B 2019

TEAM PROJECT #2
Company 33

Group Members:
Yingyuan He(Yingyuan.He@simon.rochester.edu)
Ya Liu(Ya.Liu1@simon.rochester.edu)
Mingwei Tu(Mingwei.Tu@simon.rochester.edu)
Yuchen Yao(Yuchen.Yao@simon.rochester.edu)
Bojian Zhang(Bojian.Zhang@simon.rochester.edu)

Contents

1.Prediction goal and target users

This decision support system is designed to predict risk performance of repayment of a loan or a credit card. This prediction is an important factor to decide whether or not to approve a loan or a credit card.

The target users of this company are the sales representative roles of a bank or a credit card company. According to over 200 observations of related job descriptions on Linkedin, the target users are required to be proficient in specific bank operating system (for example, nCINO), loan related software or interface, Windows Office and Adobe Software.

So we assume that target users don't have quant background but are proficient in operating bank software or interface.

2.Predictive model

2.1 Prepare data

2.1.1 Data Scaling

The dataset Homeline of Credit contains features highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Eucledian distance between two data points in their computations, this is a problem.

If left alone, these algorithms only take in the magnitude of features neglecting the units. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes.To suppress this effect, we need to control all features on the same level of magnitudes. This can be achieved by scaling.

Here we use the most common method StandardScaler(). Standardisation replaces the values

$$x' = \frac{x - \bar{x}}{\sigma}$$

by their Z scores.

This redistributes the features with their mean = 0 and standard deviation =1 . sklearn.preprocessing.StandardScaler helps us implementing standardisation in python.

2.1.2 How to deal with value in (-7,-8,-9)

From the data dictionary file, we know that '-7' represents condition not Met (e.g. No Inquiries, No Delinquencies), -9 represents no Bureau Record or No Investigation and -8 represents no Usable/Valid Accounts Trades or Inquiries. There are a bunch of young people that do not have previous credit record before they apply for loan or credit cards. So we cannot arbitarily drop those data. Instead, we should remind the bank that though these people might be predicted as 'Good', it might be unknown risk to allow those people to apply for large number of loan.

2.2 Model Comparison

The core method of our model selection is using `GridSearchCV` function to find out the model with highest CV score.

We first tried on single model include Decision tree, Logistic regression, KNN and SVM, and fine-tuned our model by trying different hyper-parameters until we found out the best one. Among all the single models, comparing both their CV scores and accuracy, SVM(rbf) has the best performance.

In order to further improve our accuracy, we then applied the aggregation method to our model training, which includes Boosting, Random Forest and Bagging models. Since SVM(rbf) performs really well in our single model training, we decided to use SVM(rbf) with the best hyper-parameters we found in the previous step as the base estimator of our bagging model. It is not surprising that all the three models have high CV scores and accuracy comparing to the single model method. Among them, the bagging model has the best performance. So we decided to choose the bagging model as our risk evaluation model, which has approximately 71.86% accuracy.

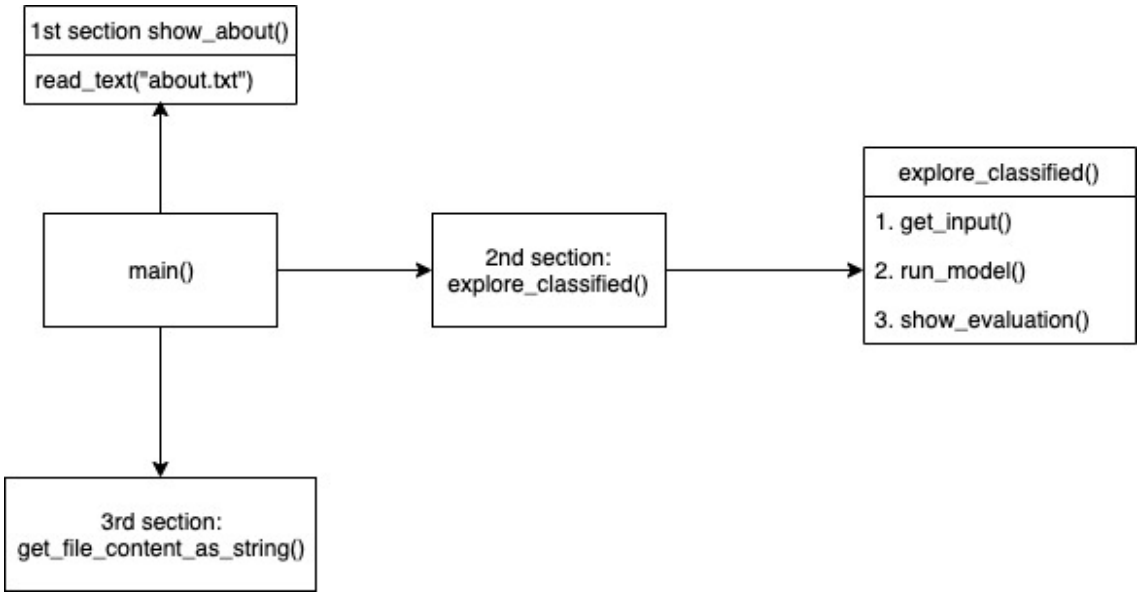
The following table shows the best estimator, CV score and accuracy of each model we have trained.

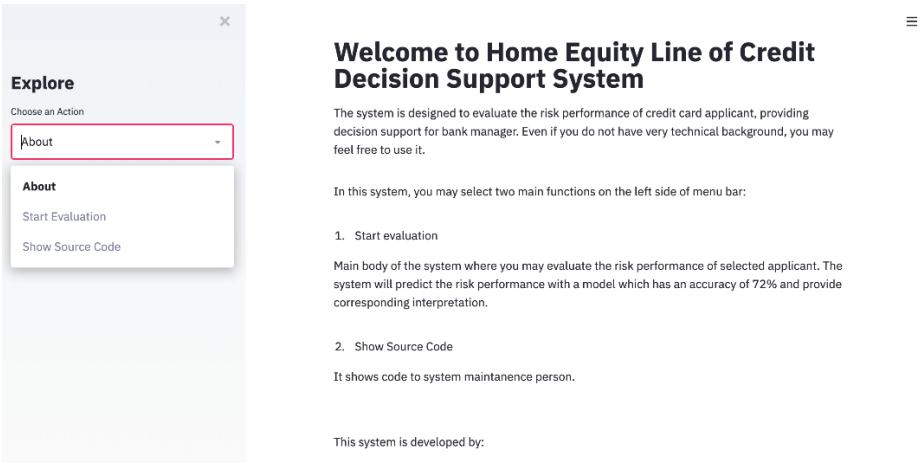
Model	Best hyper-parameters	CV Score	Accuracy
Decision tree	max_depth=6, min_samples_split=6	0.70601	0.68872
Logistic regression	C=0.1, penalty='l2'	0.71953	0.70822
KNN	n_estimators=100	0.71520	0.70746

SVM(linea)	C=1, gamma='auto_deprecated'	0.72144	0.70784
SVM(poly)	C=1, gamma='auto_deprecated', degree=3	0.71252	0.70631
SVM(rbf)	C=1, gamma='auto_deprecated'	0.72578	0.71778
Boosting	learning_rate=1, n_estimators=30	0.72616	0.71281
Random forest	n_estimators=60, max_features=5	0.72743	0.70478
Bagging (SVM base)	max_features=0.6, max_samples=0.4, n_estimators=50	**0.72769**	**0.71855**

3. Interactive interface design

The interface includes three sections on the left side of menu bar: About, Start Evaluation and Show Source Code. Main() function will call corresponding function when users choose an action.





3.1 About

3.1.1 What this page is for

This section provides brief introduction of prediction goal and 2 other sections of the decision support system.

3.1.2 Technical realization

Since default value for sidebar is “about”, main() will call show_about() automatically. show_about() will call read_text() to write “about.txt” on the screen.

3.2 Start Evaluation

3.2.1 What this page is for

This section is the main body of system for users to get decision support provided by our model. Since users in real life only use model with best accuracy and we assume that users do not have technical background, we put only one model in our system.

There are three steps for using this system. Firstly, users choose data from dataset for prediction. If necessary, for example when dataset is not updated in time or applicant doesn’t have corresponding historical record, users may input data manually. Secondly, users click “get evaluation result” to run model. Thirdly, users will get result (either “good” or “bad”) and future interpretation which explains the prediction result.

3.2.2 Technical realization

explore_classified() will call following functions conditionally:

3.2.2.1 get_input()

get_input() provides sidebar and records data users input or choose from dataset.

3.2.2.2 run_model()

run_model() will get data from get_input() and use input data to run model which performs best in our training.

3.2.2.3 show_evaluation()

show_evaluation() will receive result from run_model() and provide interpretation.

When result is 1， which means ‘Good’, the likelihood of repayment is predicted to be high. A loan or a credit card application could be approved. However, there is some problem with the missing value, which is -7,-8 and -9. These value means there is ‘ No Bureau Record or No Investigation’, or ‘No Usable/Valid Accounts Trades or Inquiries’, or condition not met. These people are usually young and this is their first time applying for loa or credit card. Considering that there is no enough historical data to refer to, we should warn banks about the risk.

When result is ‘Bad’, the likelihood of repayment is predicted to be low. A loan or a credit card application should be turned down. To find the reason that cause these people to be refused from loan or credit, we need to use information about standard normal distribution. First, we calculate the mean and standard deviation of each variable in the training set. Then we compute the possibility of X<predict value, which is Pr(X < x), to quantify how far our to-be-estimated term is from the population mean. If Pr(X < x) > 0.5, Pr(X < x) = 1 - Pr(X < x). Then we choose the first far variables and detect whether they are more than 2 standard deviation from the mean, which is the data center. However, in running the code we find that most values, even those belonging to ‘Bad’ people, are still within 2 standard deviation of mean. So we only use the possibility, and choose the first three to be the reason why these people is predicted to have low likelihood of repayment. Reasons that cause the result are provided to help applicants improve their financial status.

3.3 Show Source Code

3.3.1 What this page is for

This section prints source code of program as a reference to users with technical background and to developers who maintain and improve the system.

3.3.2 Technical realization

After calling get_file_content_as_string(), it will use urllib.request.urlopen(url) to download the raw code we uploaded and make its content available as a string.