The Warm-Up Project

Author: Li Shengshu
Student ID：20746064

Brief Introduction

In this project, since we need to predict the home credit default risk and the final status only have two classes: default and not default, which is decrement and known, it is better to use the supervised classification algorithm.

Besides, considering that the scikit-learn package of Python is a good tool in Machine Learning field, my programming selection is Python and I will mainly use scikit-learn package to help me predict the probability of home credit default.

Preprocessing and Analyzing Data

Firstly, we need to process the raw data to make sure that it can be used in later machine learning algorithm without losing too much original information.

To begin with, we need to deal with the missing value and find which categories of the data have the missing value and the percentage of the missing values in each category. By processing the data in 'application_train.csv', some useful information can be obtained.

Column Name	Missing Value	Percentage of Missing Value in Total Value
COMMONAREA_AVG	214865	69.87
COMMONAREA_MODE	214865	69.87
COMMONAREA_MEDI	214865	69.87
NONLIVINGAPARTMENTS_AVG	213514	69.43
NONLIVINGAPARTMENTS_MODE	213514	69.43
NONLIVINGAPARTMENTS_MEDI	213514	69.43
FONDKAPREMONT_MODE	210295	68.39
LIVINGAPARTMENTS_MEDI	210199	68.35
LIVINGAPARTMENTS_AVG	210199	68.35
LIVINGAPARTMENTS_MODE	210199	68.35
FLOORSMIN_AVG	208642	67.85
FLOORSMIN_MODE	208642	67.85
FLOORSMIN_MEDI	208642	67.85
YEARS_BUILD_AVG	204488	66.5

YEARS_BUILD_MEDI	204488	66.5
YEARS_BUILD_MODE	204488	66.5
OWN_CAR_AGE	202929	65.99
LANDAREA_MEDI	182590	59.38
LANDAREA_AVG	182590	59.38
LANDAREA_MODE	182590	59.38
BASEMENTAREA_MEDI	179943	58.52
BASEMENTAREA_AVG	179943	58.52
BASEMENTAREA_MODE	179943	58.52
EXT_SOURCE_1	173378	56.38
NONLIVINGAREA_AVG	169682	55.18
NONLIVINGAREA_MODE	169682	55.18
NONLIVINGAREA_MEDI	169682	55.18
ELEVATORS_MODE	163891	53.3
ELEVATORS_AVG	163891	53.3
ELEVATORS_MEDI	163891	53.3
WALLSMATERIAL_MODE	156341	50.84
APARTMENTS_AVG	156061	50.75
APARTMENTS_MEDI	156061	50.75
APARTMENTS_MODE	156061	50.75
ENTRANCES_AVG	154828	50.35
ENTRANCES_MEDI	154828	50.35
ENTRANCES_MODE	154828	50.35
LIVINGAREA_MEDI	154350	50.19
LIVINGAREA_MODE	154350	50.19
LIVINGAREA_AVG	154350	50.19
HOUSETYPE_MODE	154297	50.18
FLOORSMAX_MEDI	153020	49.76
FLOORSMAX_MODE	153020	49.76
FLOORSMAX_AVG	153020	49.76
YEARS_BEGINEXPLUATATION_MEDI	150007	48.78
YEARS_BEGINEXPLUATATION_MODE	150007	48.78
YEARS_BEGINEXPLUATATION_AVG	150007	48.78
TOTALAREA_MODE	148431	48.27
EMERGENCYSTATE_MODE	145755	47.4
OCCUPATION_TYPE	96391	31.35
EXT_SOURCE_3	60965	19.83
AMT_REQ_CREDIT_BUREAU_WEEK	41519	13.5
AMT_REQ_CREDIT_BUREAU_HOUR	41519	13.5
AMT_REQ_CREDIT_BUREAU_MON	41519	13.5
AMT_REQ_CREDIT_BUREAU_QRT	41519	13.5
AMT_REQ_CREDIT_BUREAU_DAY	41519	13.5
AMT_REQ_CREDIT_BUREAU_YEAR	41519	13.5

NAME_TYPE_SUITE	1292	0.42
DEF_30_CNT_SOCIAL_CIRCLE	1021	0.33
OBS_60_CNT_SOCIAL_CIRCLE	1021	0.33
OBS_30_CNT_SOCIAL_CIRCLE	1021	0.33
DEF_60_CNT_SOCIAL_CIRCLE	1021	0.33
EXT_SOURCE_2	660	0.21
AMT_GOODS_PRICE	278	0.09

Table1: Number and proportion of missing values of all categories with missing values.

After obtaining the **Table 1**, we can find out which kind of data can be imputation and which kind of data must be dropped. Some missing values of categories can be filled artificially. For example, the categories, "OBS_30_CNT_SOCIAL_CIRCLE" and "OBS_60_CNT_SOCIAL_CIRCLE" represent that how many observations of client's social surroundings with observable 30 or 60 DPD (days past due) default respectively. Most of values are 0 in these two categories and only 0.33% of values are missing. Thus, we can fill those missing values with 0 or their mean values. For another example, the category "NAME_TYPE_SUITE" represents that who accompanied client when applying for the previous application and most of the clients applied on their own, with the status "Unaccomplished". Just like the categories above mentioned, this category also only has very low percentage of missing values, which is only 0.42%, So we can fill the missing value with "Unaccomplished". By this way, I filled all the categories whose percentage of missing values are lower than 0.5% with their mode value or mean value. However, other categories can be dropped due to their higher percentage of missing value.

Then, I try to transform those no-numerical values into numerical values, which is necessary for later classification, since numerical value is the only kind of data format can be distinguished by scikit-learn package. The no-numerical categories and their number of unique values are as follow:

Column Name	NO. of Unique Values
NAME_CONTRACT_TYPE	2
CODE_GENDER	3
FLAG_OWN_CAR	2
FLAG_OWN_REALTY	2
NAME_TYPE_SUITE	7
NAME_INCOME_TYPE	8
NAME_EDUCATION_TYPE	5
NAME_FAMILY_STATUS	6
NAME_HOUSING_TYPE	6
WEEKDAY_APPR_PROCESS_START	7
ORGANIZATION_TYPE	58

Table2: The no-numerical categories and their corresponding number of unique values.

After analyzing the no-numerical categories, I find that their values just represent a kind of status, so there is no size relationship between different status, which leading to a fact that it might be not suitable to use Label Encoding, reassigning values to all different status according to 1, 2, 3, 4. Thus, I decide to One-Hot Encoding, which means that each status of each category is reassigned as a new category with only to value 0 and 1, representing "Yes" or "No" respectively.

Classification Algorithm Selection and Result Analyzing

Form our institution, lots of categories provided in this dataset, such as "Income of the client" and "Credit amount of the loan", are highly linear correlated with their probability of default, since the more they earn, the more they will pay for loan and the more credit amount they have, the less repayment pressure they we suffer. Thus, with the assumption that those features have linear correlation with default risk, Logistic Regression might be a good Classification Algorithm.

$$P(y=1|x) = \frac{e^{w^T x+b}}{1+e^{w^T x+b}} \qquad P(y=0|x) = \frac{1}{1+e^{w^T x+b}}$$

Formula1: Logistic Regression.

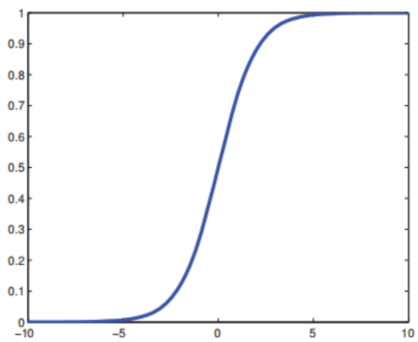

Figure1: Diagrammatic sketch of Logistic Regression.

Besides, I also tried Random Forest Algorithm to classify and predict the default risk, since it also an intuitive algorithm. For example, "Number of children the client has" might linear separate the default risk, since having no child might mean that client is too poor to raise offspring and raising one to three child might mean that client has considerable income, but if the client has too many children will bring too many unpredictable accident which increases his or her default risk.

$$H(X) = -\sum_{k=1}^{N} \frac{|C_k|}{|D|} log_2 \frac{|C_k|}{|D|} \qquad g(D,A) = H(D) - H(D|A) \qquad g_r(D,A) = \frac{g(D,A)}{H(D)}$$

Formula2: Random Forest.

Finally, after fitting the model and predicting the default probability of test dataset, I got the final score on the competition website and the score of Logistic Regression Algorithm is 0.61306 and the score of Random Forest Algorithm is 0.67440.

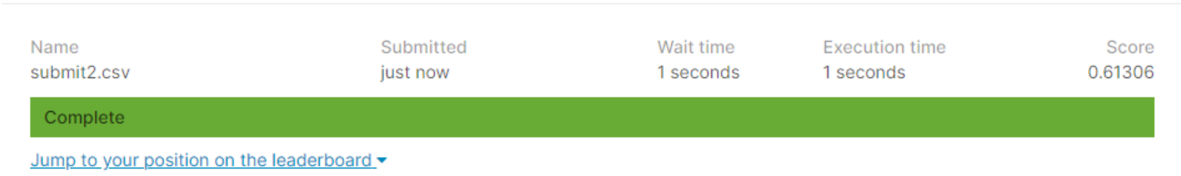

Figure2: The final score of the Logic Regression model.

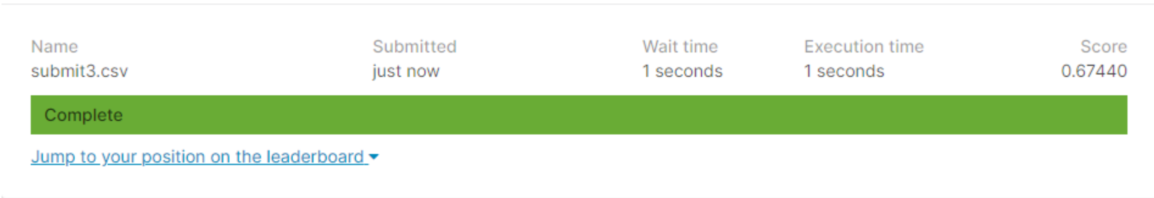

Figure3: The final score of the Random Forest model.

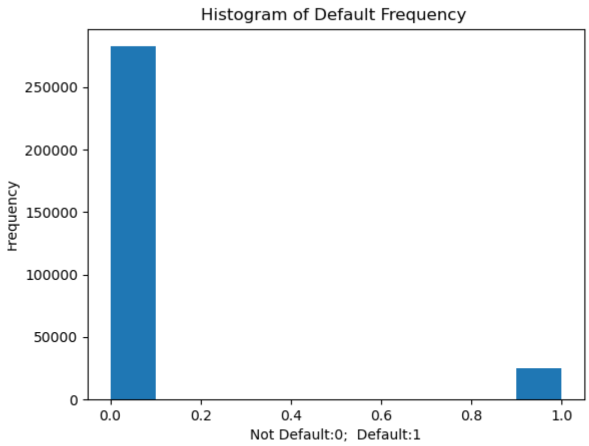

Figure4: The histogram of default frequency of train dataset.

Since the frequency distribution of default and not default is very uneven, we cannot simply use the accuracy of the prediction of the test dataset. The score, the area under the ROC curve can be regard as a more scientific way of prediction. Furthermore, 0.61306 and 0.67440 are larger than 0.5, which means that my models are effective in some extent.

Furthermore, the result of Random Forest model is better than Logistic Regression's. From my point of view, it might be cause by the reason that those features in the dataset are just linear separatable instead of linear correlated with default risk.

Summary

I used Python and scikit-learn package of Python to complete the warm-up project. At the beginning of the project, I preprocessed the data to fill the missing values and transform no-numerical value to numerical value. Then, I used the Logistic Regression and Random Forest to classify the target feature and predict the probability of the default of the test dataset. At the end, the scores I received from the competition website show that my models are effective in some extent and the Random Forest model is better than the Logistic Regression model.