Credit Card Approval Prediction

Using Classification Method

MATH 509 Final Report | Fall 2022

Zhengyuan Liu, Ruike Xu, Steven Ge

TABLE OF CONTENTS

INTRODUCTION

Background

The credit score is an essential determination of personal credit evaluation in modern society. When proceeding with credit card applications, credit companies employ the applicants' personal information and credit history to evaluate future defaults and overdue payments. It's critical to predicting whether the card applicants will pay on time.

The dataset we used contains transactions made by credit cards in September 2013 by European cardholders. The data set includes the basic information of the cardholder, such as marriage status, property ownership, number of family members, number of children, annual income, etc.

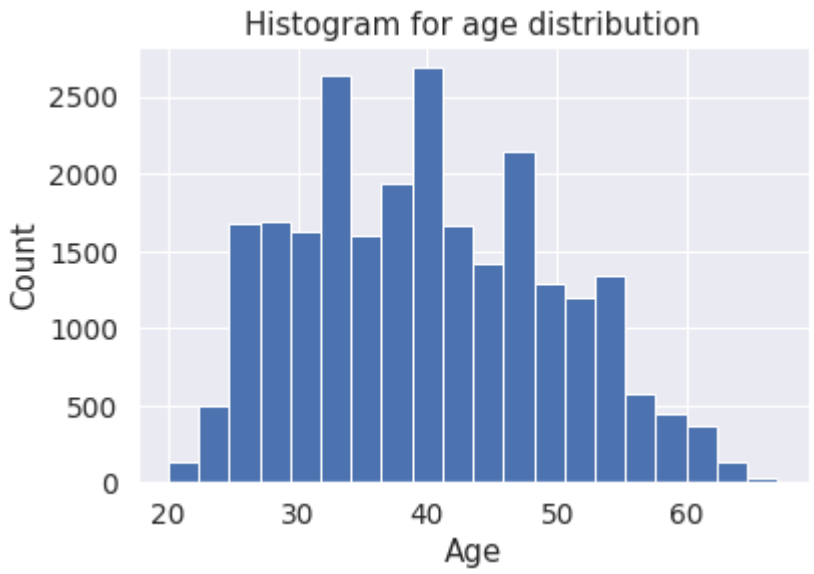
Objective

We aim to employ demographic and personal information to determine whether the application should proceed. We would employ several classification methods and compare the performance of each modeling technique to evaluate the best possible model for our problem.

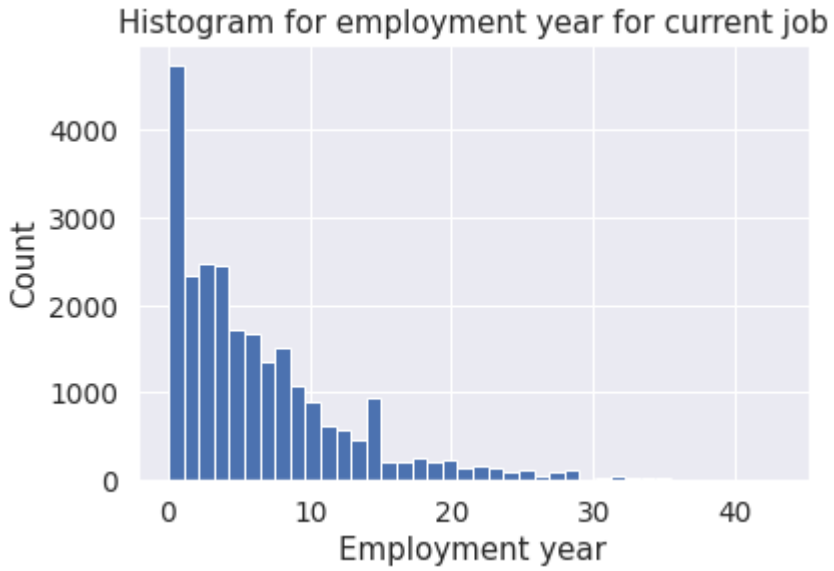
Description statistic

In our data set, there are 17 independent variables and one dependent variable. The dependent variable is the loan repayment status of each individual client in different periods. There are 5 numerical variables in the independent variable, which are the number of children; total annual income; age; total working days for the current job; the number of family members, and 7 binary variables, which are: gender; car ownership; realty ownership; possession of the phone and a work mobile phone; whether they have an email, and 5 categorical variables, which are: job types; education level; marital status; housing type; job position.

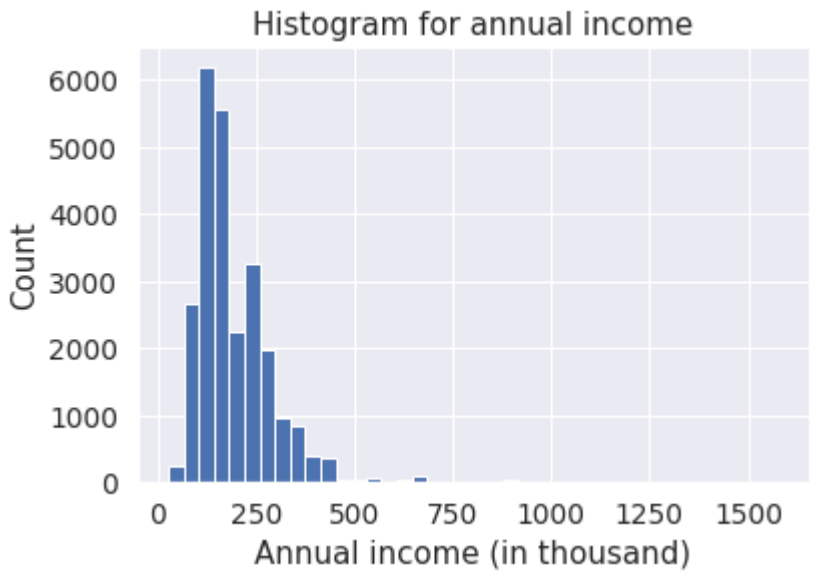
Based on preliminary analysis, variables such as property and total income should have a strong impact on loan repayment status, while variables such as whether there is a mobile phone and email address should have a weak impact. We will verify this point of view in the subsequent analysis.



From the above histogram of the age distribution for our data, we can observe that the age of the clients is roughly normally distributed around 45 years old. One thing that we find interesting to mention is the number of cardholders under 30 years old. We thought that clients around the age of 20-30 would have strong demands for financial support. However, the data claims that they are not the major proportion of clients.



The above histogram is the employment year distribution for all clients. We could observe that the most dominant proportion of clients are unemployed or have been working at their current job for less than a year. The employment time for the current job declines exponentially for our distribution. There are few clients exceeding 30 years of employment for the current job.



This histogram for annual income shows the number of clients that fall into each income level. We can find that the data contains clients with extremely high incomes, creating a right-skewed distribution. The plot indicates that a large proportion of clients had an income of around 170k.

Data cleaning

We first proceed with the data cleaning process. We would first adjust the credit status of clients to the binary categorical variable using a scoring system. We have 6 states that indicate whether the clients paid past loans. We define a scoring system based on the state of the clients that pay their owed money or don't owe money at all. If the client doesn't owe money or repay on time during this period, we will grant this client the highest score of 6. For the state of customers who are overdue for more than one day, we mark them from high score to low score as their loan overdue periods increase. (From 5 to 1) We then split the credit status of clients based on the quantile of their scores. In this way, we split clients with 'good' and 'bad' credit status using a 50% quantile of scoring.

After this, we have 12568 ‘good’ clients and 12566 ‘bad’ clients.

We also change categorical variables in the form of one-hot encoding in order to make sure we can fit the model in the later part. According to the documentation, we adjust the 'DAYS_EMPLOYED' and 'DAYS_BIRTH' to be an appropriate format. The positive values indicate the number of days lived / number of days employed for the current job. We then create a new variable called 'WORK_AGE_RATIO,' which is the ratio of the client's current working days to the client's time of living. This information would demonstrate the stability of the client's current employment.

In order to stabilize the model and reduce the biases, we also normalize three numerical variables with large magnitude differences: total annual income, days of birth, days employed, and work age ratio.

FORMULATION

For our classification methods, we would need our response variable to be a discrete binary discrete variable. We also need to eliminate confounding effects in our models, which would require examining correlations in our features. We also assume that the observations we collected are independent observations.

Logistic regression

Logistic regression is a type of generalized linear regression often used as a classification model to predict binary variables, like '0' and '1', based on the independent variables. The outcome of the model is the probability that the dependent variable is predicted to be a certain condition P(y=1). When the probability is less than 0.5, we say y is equal to 0, and on the contrary, y is predicted to equal 1. The formula is

$$P(y=1) = \frac{1}{1+e^{-(\beta 0+\beta 1X1+\beta 2x2+\ldots\ldots+\beta nXn)}}$$

Decision Tree

A decision tree is a tree-like model, which is a decision-making process based on multiple decisions. In each decision tree, each node asks a question to help classify the data, whereas each branch represents different probabilities that this node could lead to.

Since it is hard for us to make decisions if the split of data is not pure(each split has more than one category of response), we would employ Gini Index to classify each node in the decision tree. We would take that feature as the root node which gives the lowest impurity(lowest Gini index)

$$Gini\ Index\ = 1 - \sum_{i=1}^{n} (P_i)^2 = 1 - [(P_+)^2 + (P_-)^2]$$

Where P+ is the probability of a positive class and P- is the probability of a negative class.

Entropy is also a way to measure the impurity of the split,

$$E(S) = - p_{(+)} log p_{(+)} - p_{(-)} log p_{(-)}$$

We usually use the Gini index since it is computationally efficient, and it saves time complexity since there is no logarithmic term like entropy.

Random forest

Random forest is a supervised learning model based on a series of decision trees. The random forest combines different decision trees and produces the outcome by calculating the average of all decision trees. Random forest uses ensemble learning, which combines many weak classifiers to provide solutions for a complex problem.

Neural network

A neural network is a computational learning system that uses a network of functions to understand and translate a data input of one form into the desired output, which was inspired by human biology and the way neurons of the human brain function together to understand input from the human senses.

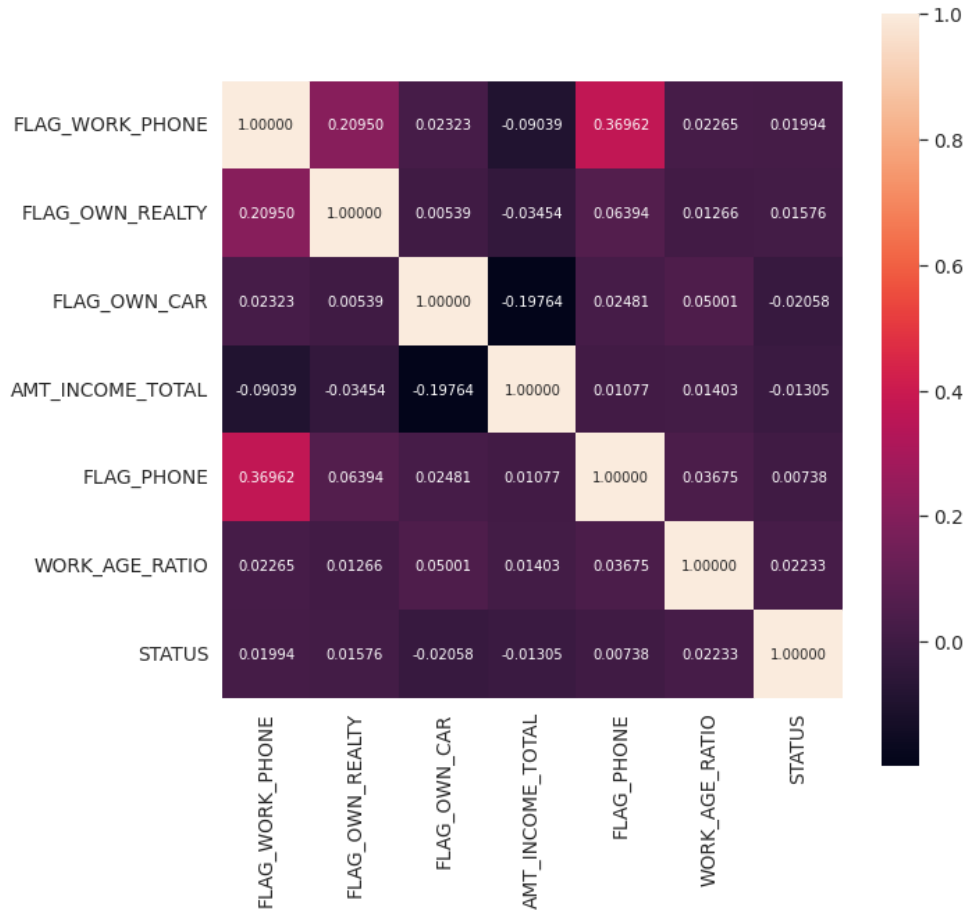
Support vector machine

The support vector machine algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The algorithm aims to find the hyperplane with the maximum margin, which means that the distance between data points of two classes is maximized. In this case, the classifier provides more predictive power for future observations. The algorithm we used is the radial basis function kernel SVM.

SOLUTION

Feature selection

We would use the correlation of potential predictors to the response variable to conduct a preliminary feature selection. We first find the absolute correlation of all potential predictors to the credit status of the customer, then sort out the first few predictors with high correlations to the credit status.



From the above correlation matrix with dominant correlation predictors, we can observe that all potential predictors have moderate contributions to the credit status, but there exists a confounding effect between 'FLAG_PHONE' and 'FLAG_WORK_PHONE'; we would remove the 'FLAG_PHONE' predictor since it has relatively less contribution to the credit status. Therefore, we finalize our feature selection with 'FLAG_OWN_CAR,' 'FLAG_WORK_PHONE,' 'FLAG_OWN_REALTY,' 'AMT_INCOME_TOTAL,' and 'WORK_AGE_RATIO' as our predictors.
After doing these, We randomly shuffle and divide the data set into two parts, the 80% data to be the training set and 20% data to be the test set. We use the training set to fit the model and then use the test set to test the performance and generalization of the model.

Modeling

Logistic regression classification

We can first do the logistic model to make the prediction; the logistic regression is going to return the probability that the customer is good.
We first fit the model by using the training dataset model and then use the test model to calculate the accuracy. Here is the confusion matrix of the training dataset of this model:

```
Accuracy Score is 0.51703
      0     1
0  5389  4664
1  5047  5007
```

And the confusion matrix of the testing dataset is as follows:

```
Accuracy Score is 0.51502
      0     1
0  1369  1144
1  1294  1220
```

We can see the accuracy of both of the confusion matrices is about 0.5, which shows that the regression model is not very useful.
It might be because the correlation of each variable is not very strong, which means that we should not use a logistic regression model in our data.

Decision Tree

In this model, we divide the eigenvalues of each variable by recursion and then classify the data into one category. This model does not need the data to follow linear regression.
Here's the confusion matrix of the training dataset of this model:

```
Accuracy Score is 0.80067
      0     1
0  8380  1673
1  2335  7719
```

And the confusion matrix of the testing dataset is as follows:

```
Accuracy Score is 0.66382
      0     1
0  1778   735
1   955  1559
```

We can see now the accuracy has clearly increased to 0.80067 now, which is much better than logistic regression.
And there's one thing that has happened, which is that the accuracy score has been decreasing from training data to testing data, which implies that our data are not very suitable for general data.

Random forest

More advanced, we can do a random forest model. We fit the model of the decision tree multiple times and then select the best model as the final model by voting.

We let the number of trees equal 200 for random forest modeling. And calculate the confusion matrix by using the training dataset:

```
Accuracy Score is 0.80067
      0     1
0  7933  2120
1  1888  8166
```

And the confusion matrix of the testing dataset is as follows:

```
Accuracy Score is 0.66421
      0     1
0  1647   866
1   822  1692
```

We see the result of the random forest is very similar to the decision tree model. Just the accuracy score of the testing dataset is a little bit higher than the decision tree model.

Neural network

In addition, we try to use the Neural network model to contrast with the random forest model. We use 4 Hidden layers; each is 200, 150, 100, 50 in our model.

Here is the confusion matrix of training and testing dataset are:

```
Accuracy Score is 0.64893
      0     1
0  6025  4028
1  3031  7023
```

```
Accuracy Score is 0.56734
      0     1
0  1307  1206
1   969  1545
```

We see the accuracy score of both of them is about 60%. This score is just better than logistic a little bit, but not as good as the random forest model.

SVM | Support vector machine

We also try to use the SVM model, And the confusion matrix of the training and testing dataset is:

```
Accuracy Score is 0.54339
      0     1
0  5502  4551
1  4630  5424
```

```
Accuracy Score is 0.52457
      0     1
0  1349  1164
1  1226  1288
```

We see it just like the Neural network model, which means it's better than the logistic model but not better than the random forest model.

INTERPRETATION

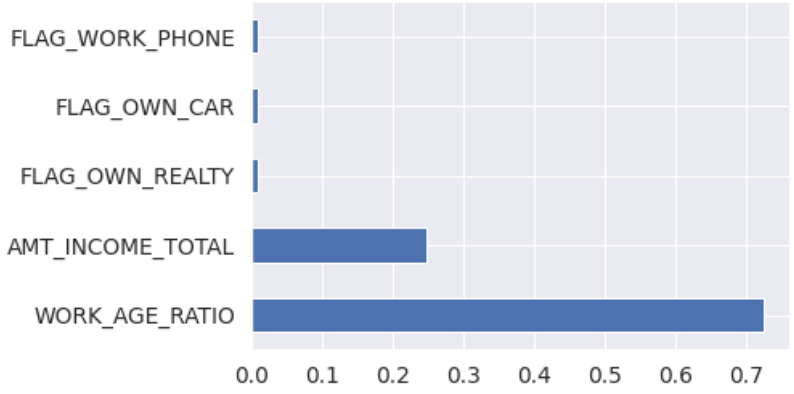
Among all models that we used for the data, the random forest model has the highest performance. In order to make further analysis of the performance of the model, we generate the classification report, which includes the following variables:

- Precision is the accuracy of all cases that are predicted as positive
- Recall the percentage of all positive cases that are correctly predicted
- F1-score, a weighted harmonic mean of precision and recall
- Support, the number of cases in the actual data with the specific class (0 and 1 in this report)

	precision	recall	f1-score	support
0	0.67	0.66	0.66	2513
1	0.66	0.67	0.67	2514
accuracy			0.66	5027
macro avg	0.66	0.66	0.66	5027
weighted avg	0.66	0.66	0.66	5027

The data appears to be equally split by the two classes, and the score results are identical. Thus the model performed equivalently when predicting each of the classes.

To better understand the model, we also export the feature importance plot.



The plot indicates that the "WORK_AGE_RATIO" variable created earlier has the largest contribution to the model, with "AMT_INCOME_TOTAL" (total annual income) listed in the second. Based on the results, we believe that the stability of the client's current employment would be a great explanatory variable for other industrial institutions to consider when building their own model.

CRITIQUE

Our decision tree and random forest models have similar performance on the testing data, the decision tree has an accuracy score of 0.65765, and the random forest has a score of 0.66183. The decision tree has limitations on its unstableness, meaning that any small changes to the data may result in a largely different model structure. Also, the decision tree may overfit the data, making it difficult to make good predictions on another dataset. Random forest is able to overcome this problem most of the time by combining random subsets with the cost of more computational power. Furthermore, an increase in its accuracy will require a larger number of trees, which slows down the model and makes it unsuitable for predictions in real time.

For our project, tweaking the minimum leaf split samples and maximum depth for the decision tree may avoid overfitting. Increasing the number of trees and a maximum number of features using hyperparameters would improve our random forest model's accuracy. Another method that we would consider in the future to avoid overfitting is using cross-validation.

If we had more time, we would also like to make adjustments to our data acquisition process. Our current data contains information about each client’s income level. However, it would be better if we could find relative information on the client's level of consumption. Furthermore, the client's relationship with the bank, such as their money-saving habits, are also interesting variables that we think would have great contributions towards our final predictions. As mentioned before, our data include cardholders from European countries only. Thus the result may not apply to the global population. We would like to find more relative data from other countries while using better sampling methods such as stratified random sampling and weighted random sampling. Our data also appears to be collected in the year 2013, which means that the model might not fit the current society (considering the effect of the pandemic). Thus if we have more time, we would like to find data that are collected from a more recent timeline.

APPENDICES

GitHub repository link: https://github.com/Stelvlen/MATH-509.git

Reference (APA)

Schott, M. (2020, February 27). *Random Forest Algorithm for Machine Learning*. Medium. Retrieved December 4, 2022, from https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb#:~:text=The%20R%20andom%20Forest%20Algorithm,of%20all%20these%20decision%20trees.

Saini, A. (2022, August 26). *Random Forest Algorithm for absolute beginners in Data Science*. Analytics Vidhya. Retrieved December 4, 2022, from https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-random-forest-algorithm-for-beginners/

DeepAI. (2019, May 17). *Neural network*. DeepAI. Retrieved December 4, 2022, from https://deepai.org/machine-learning-glossary-and-terms/neural-network

Gandhi, R. (2018, July 5). *Support Vector Machine - introduction to machine learning algorithms*. Medium. Retrieved December 6, 2022, from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Corporate Finance Institute. (2022, November 30). Decision Tree. Retrieved December 6, 2022, from https://corporatefinanceinstitute.com/resources/data-science/decision-tree/#:~:text=One%20of%20the%20limitations%20of,get%20in%20a%20normal%20event.

(n.d.). The Ultimate Guide to Decision Trees for Machine Learning. Retrieved December 6, 2022, from https://www.keboola.com/blog/decision-trees-machine-learning

Donges, N. (n.d.). Random forest classifier: A complete guide to how it works in Machine Learning. Built In. Retrieved December 6, 2022, from https://builtin.com/data-science/random-forest-algorithm

Kohli, S. (2019, November 18). *Understanding a classification report for your machine learning model*. Medium. Retrieved December 8, 2022, from https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397