Personal Loan Forecasting Based on Machine Learning

Xiaotong Liu 001062480

1. Problem description

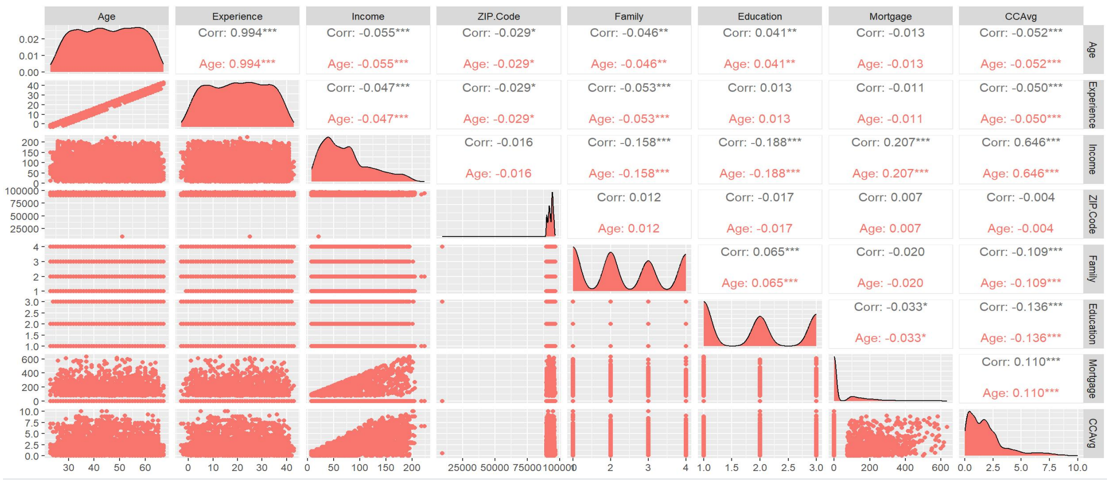
1.1 Introduction

Machine learning plays an important role in today's society. In the financial field, loans are banks, credit unions and other institutions that lend money to units or individuals that need money. Loans can meet the needs of the society to expand reproduction for supplementary funds and promote economic development. At the same time, banks can also obtain loan interest income from them. We can use machine learning to train a suitable model to predict whether a bank customer will accept a personal loan. What kind of model to choose and how to train it is a challenge. In this report we will discuss several common classification models.

1.2 Data detail

The data set we obtained contains these features: *Age, Experience, Income, ZIP.Code, Family, Education, Mortgage, CCAvg, Personal.Loan, Securities.Account, CD.Account, Online, CreditCard,*where *Personal.Loan* is the response variable and other variables are predictors.

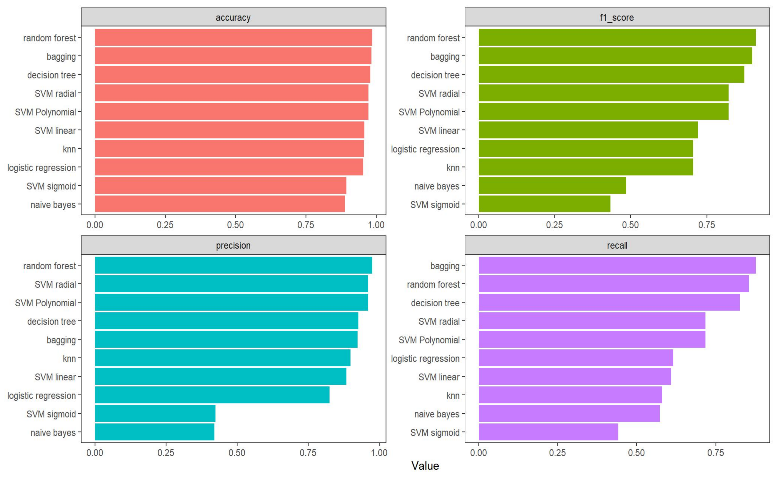
The figure below shows the relationship between different feature variables.



By looking at the picture we can clearly see that there is an obvious linear relationship between age and experience. And they are proportional to each other. This is reasonable.

1.3 Model performance

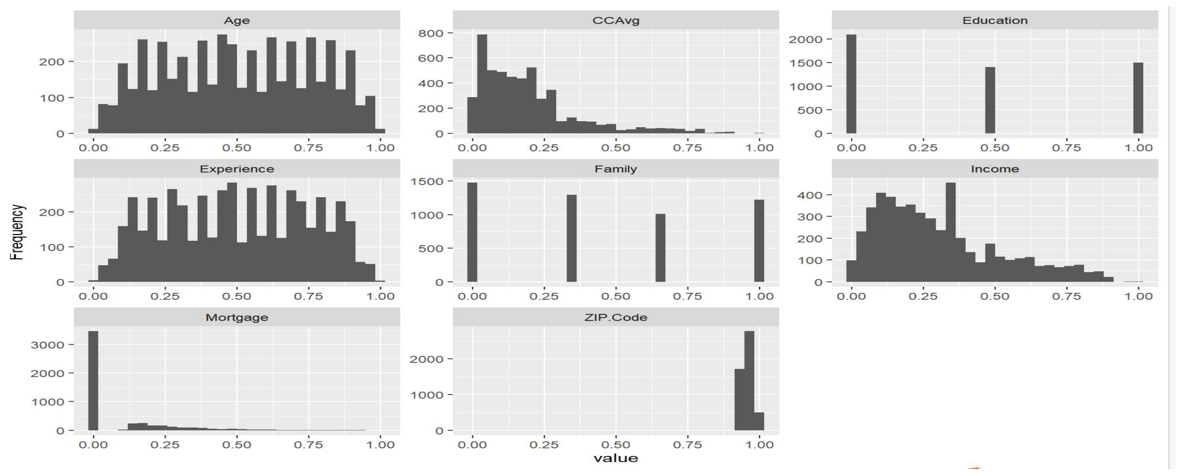
When we first use different simple machine learning models to fit data, we can find that different models have different performances through test data and prediction results. Different models perform differently under four different evaluation metric. The figure at right shows the ranking of model performance under evaluation metric such as *accuracy, f1_score, precision, recall.*



2. Technical Summary

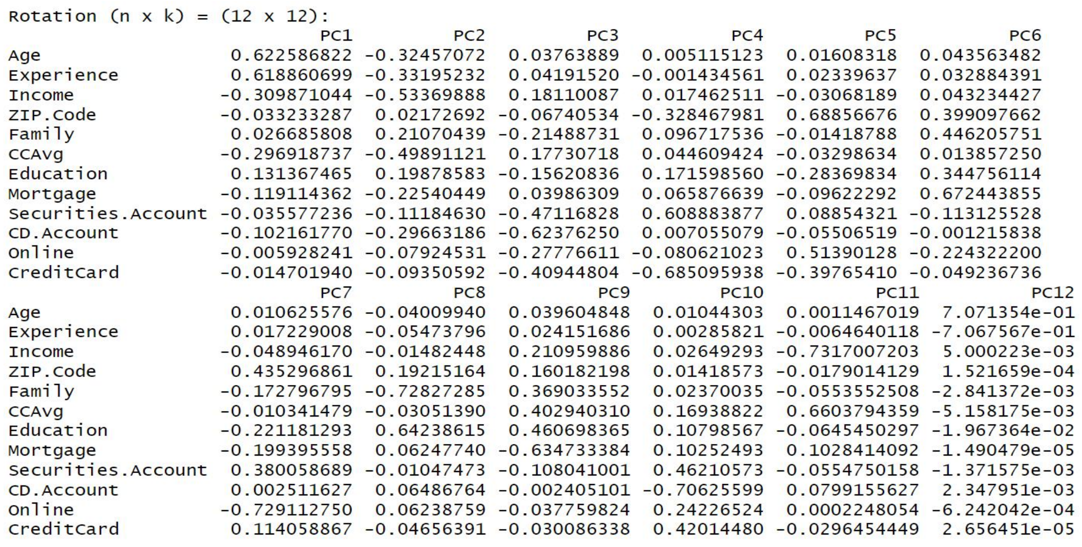
2.1 Data Preprocessing

First, let’s have a look of the frequency of the original data.



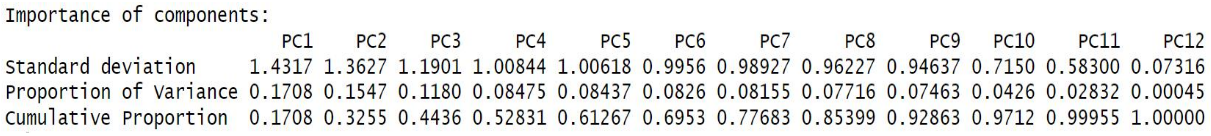
we perform principal component analysis on the data and we can obtain the loading values of each principal component on all variables. Taking the first principal component as an example, the relationship between the first principal component and each variable is shown in the following formula:

*PC1 = 0.622586822*Age + 0.618860699*Experience - ... - 0.005928241*Online - 0.014701940C*reditCard*



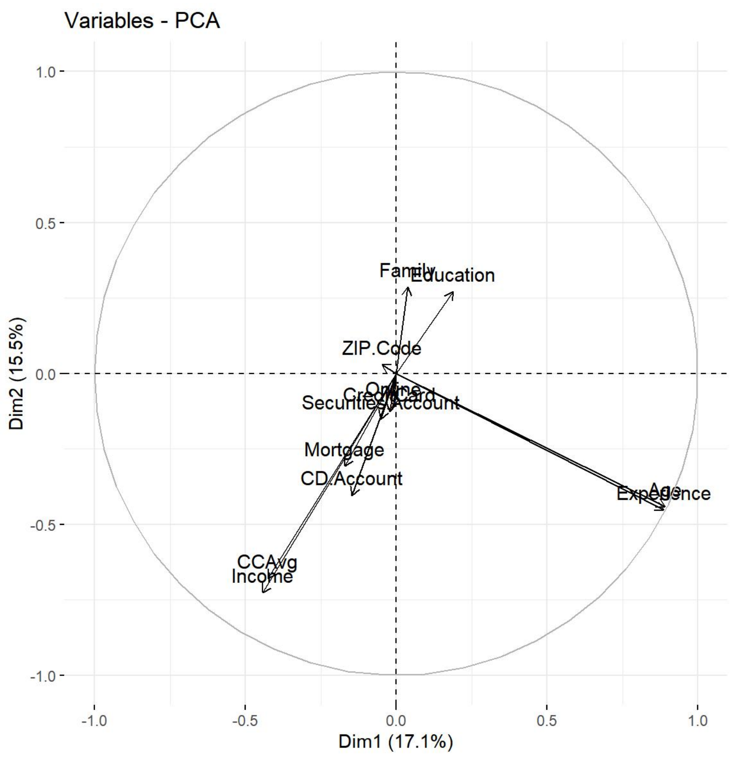
```
Rotation (n x k) = (12 x 12):
                          PC1         PC2         PC3          PC4         PC5          PC6
Age                0.622586822 -0.32457072  0.03763889  0.005115123  0.01608318  0.043563482
Experience         0.618860699 -0.33195232  0.04191520 -0.001434561  0.02339637  0.032884391
Income            -0.309871044 -0.53369888  0.18110087  0.017462511 -0.03068189  0.043234427
ZIP.Code          -0.033233287  0.02172692 -0.06740534 -0.328467981  0.68856676  0.399097662
Family             0.026685808  0.21070439 -0.21488731  0.096717536 -0.01418788  0.446205751
CCAvg             -0.296918737 -0.49891121  0.17730718  0.044609424 -0.03298634  0.013857250
Education          0.131367465  0.19878583 -0.15620836  0.171598560 -0.28369834  0.344756114
Mortgage          -0.119114362 -0.22540449  0.03986309  0.065876639 -0.09622292  0.672443855
Securities.Account -0.035577236 -0.11184630 -0.47116828  0.608883877  0.08854321 -0.113125528
CD.Account        -0.102161770 -0.29663186 -0.62376250  0.007055079 -0.05506519 -0.001215838
Online            -0.005928241 -0.07924531 -0.27776611 -0.080621023  0.51390128 -0.224322200
CreditCard        -0.014701940 -0.09350592 -0.40944804 -0.685095938 -0.39765410 -0.049236736
                          PC7         PC8          PC9        PC10          PC11          PC12
Age                0.010625576 -0.04009940  0.039604848  0.01044303  0.0011467019  7.071354e-01
Experience         0.017229008 -0.05473796  0.024151686  0.00285821 -0.0064640118 -7.067567e-01
Income            -0.048946170 -0.01482448  0.210959886  0.02649293 -0.7317007203  5.000223e-03
ZIP.Code           0.435296861  0.19215164  0.160182198  0.01418573 -0.0179014129  1.521659e-04
Family            -0.172796795 -0.72827285  0.369033552  0.02370035 -0.0553552508 -2.841372e-03
CCAvg             -0.010341479 -0.03051390  0.402940310  0.16938822  0.6603794359 -5.158175e-03
Education         -0.221181293  0.64238615  0.460698365  0.10798567 -0.0645450297 -1.967364e-02
Mortgage          -0.199395558  0.06247740 -0.634733384  0.10252493  0.1028414092 -1.490479e-05
Securities.Account  0.380058689 -0.01047473 -0.108041001  0.46210573 -0.0554750158 -1.371575e-03
CD.Account         0.002511627  0.06486764 -0.002405101 -0.70625599  0.0799155627  2.347951e-03
Online            -0.729112750  0.06238759 -0.037759824  0.24226524  0.0002248054 -6.242042e-04
CreditCard         0.114058867 -0.04656391 -0.030086338  0.42014480 -0.0296454449  2.656451e-05
```

Then we get the importance of each principal component through the summary function. It can be seen that the cumulative proportion of the first nine principal components reached 92%.



```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5    PC6     PC7     PC8     PC9   PC10    PC11    PC12
Standard deviation     1.4317 1.3627 1.1901 1.00844 1.00618 0.9956 0.98927 0.96227 0.94637 0.7150 0.58300 0.07316
Proportion of Variance 0.1708 0.1547 0.1180 0.08475 0.08437 0.0826 0.08155 0.07716 0.07463 0.0426 0.02832 0.00045
Cumulative Proportion  0.1708 0.3255 0.4436 0.52831 0.61267 0.6953 0.77683 0.85399 0.92863 0.9712 0.99955 1.00000
```

Finally use the summary function to visualize the loading of all variables under the 1st principal component and the 2nd principal component.



2.2 Model fitting

In this experiment I used several machine learning training models. Next, the algorithm of the mechanism model will be introduced.

2.2.1 Decision Tree Classification Algorithm

Divide a dataset into segments based on some characteristic variables in the dataset. The thresholds for these divisions are usually the mean or mode (if they are numeric) of the respective feature variables. Since a tree can represent a set of segmentation rules for segmenting a data set, this algorithm is called a decision tree. Metrics to test the purity of splits:

$$D = -\sum_{k=1}^{k} p_{mk} \log\left(p_{mk}\right)$$

(pmk represents the proportion of training variables belonging to the k-th class in the m-th segment)

2.2.2 Random Forest

A forest consists of a large number of trees. Likewise, random forests involve processing many decision trees. Each tree predicts the probability value of the target variable. We then average the probabilities of producing the final output.We evaluate each tree as follows:

a. Create the first sample of the dataset by selecting data points with replacement.

b. Next, we create a decision tree without using all input variables. We only use a subset of what is available.

c. Each tree was allowed to grow to its greatest possible length, and no pruning was involved.

2.2.3 Logistic Regression

The algorithm is similar to a Bayesian classifier in that it also predicts the probability that Y is associated with the input variable X. It uses the logical function,

$$P(X)=\frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

maximize the likelihood function given by,

$$l(\beta_0,\beta_1)=\prod_{l:y_i=1}p(x_i)\prod_{l':y_{i'}=1}(1-p(x_i))$$

2.2.4 SVM

The algorithm utilizes support vector classifiers with exciting variations, making it suitable for evaluating non-linear decision boundaries. This is made possible by enlarging the feature variable space using special functions called kernels. The decision boundary considered by the algorithm allows to label feature variables as target variables. The mathematical function it uses to evaluate the bounds is given by

$$f(x)=\beta_0-\sum_{i\in s}\alpha_i K(x,x_i)$$ *where K represents the kernel function.*

2.2.5 KNN

The KNN algorithm works by identifying the K nearest neighbors for a given observation point. It then evaluates the proportion of each type of target variable using K points, and then predicts the target variable with the highest proportion.

2.2.6 Naive Bayes

Naive Bayesian classifier, one of the simplest and most effective classification algorithms. It is based on Bayes' theorem, which describes how the probability of an event can be assessed based on prior knowledge of the conditions likely to be associated with the event. Mathematically, this theorem states that—

$$\mathrm{P}(Y|X)=\frac{P(X|Y)P(Y)}{P(X)}$$

We can use the following formula to estimate the probability.

$$P(Y|X_1=x_1,X_2=x_2)=\frac{P(X_1=x_1,X_2=x_2|Y)P(Y)}{P(X_1=x_1,X_2=x_2)}$$

2.2.7 Cross-Validation

In this experiment we chose five-fold cross-validation. 5-fold cross-validation is to divide the data set into 5 parts, 4 parts are used to train the model, and 1 part is used to test the model, and then repeat this process 5 times, each time selecting a different part of the data as the test set.

2.3 Model improvements

2.3.1 Regularisation Methods

We use min_max_scale regularisation methods, the function is

(x-min(x))/(max(x)-min(x)).

After regularisation, we get the standard deviation of all features.

```
Personal.Loan               Age        Experience            Income          ZIP.Code            Family
   0.29462071        0.26052649        0.24930334        0.21311912        0.02429305        0.38255436
        CCAvg         Education          Mortgage Securities.Account       CD.Account            Online
   0.17476590        0.41993454        0.16017922        0.30580933        0.23825027        0.49058933
   CreditCard
   0.45563749
```

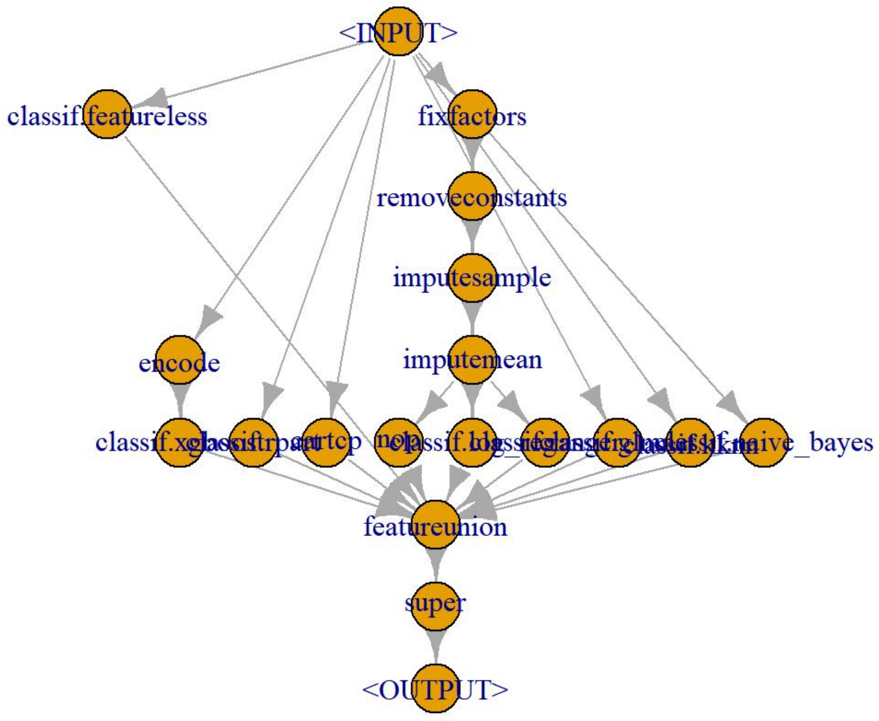
2.3.2 Data Augmentation Approaches

In this case, we choose super learning. We'll go for cross validation, and we'll arbitrarily choose 5 folds. We first create the resampling strategy and then 'instantiate' it on the task. Instantiating means that the folds get fixed: we can then use this object on multiple models and the same set of folds have been used so that comparisons are fair. We can get many error measures though aggregate() and msr("") function.

Firstly we define a task. Here, we are going to define a 'positive' outcome to be 1 meaning customer accepted a personal load offered. This will be important to specify for when we assess the model performance. Secondly we choose Cross validation resampling strategy. Then, define a collection of base models and a super model. Next, define a full pipeline to group different models with different conditions. Finally fit the super model and get measurement.

2.4 Performance report

The model below is the super model we obtained.



By evaluating the model we found that the model performs well. We consider the following measures: Classification Error, Classification Accuracy, False Positive Rate, False Negative Rate and Log Loss. The value of them are:

```
res_spr$aggregate(list(msr("classif.ce"),
                       msr("classif.acc"),
                       msr("classif.fpr"),
                       msr("classif.fnr"),
                       msr("classif.logloss")))
   classif.ce     classif.acc     classif.fpr     classif.fnr classif.logloss 
  0.011200000     0.988800000     0.004422067     0.077133169     0.039947847 
```

Code link: https://github.com/AdiyiahL/Classification-Coursework