Prediction of the Success of Bank Telemarketing and Model Comparison

Zhiying Li(zl2697)
Yuting He(yh3054)
Xiaoming Chen(xc2479)

1.Introduction

Marketing selling constitute a typical strategy to enhance business. Companies not only use direct marketing when targeting segments of customers by contacting them to meet a specific goal, but also through various indirect channels, telephone (fixed-line or mobile) being one of the most widely used. Using data driven methods like the machine learning techniques, it is more efficient and effective for the companies to grasp the information of their customers, thus improving their selling process. In this paper, we use data driven methods to predict the success of bank telemarketing and learn which kind of customers are more likely to buy the products.

We analyze a recent and large dataset (45,211 records) from a Portuguese bank. The data were collected from 2008 to 2013, thus including the effects of the global financial crisis that peaked in 2008. We compare four Machine Learning models (Logistic Regression, Random Forest, Neural Network and SVM) using a AUC-ROC and classification metric. We find that the best model (Random Forest) could most accurately predict the possibility of the customers buying the products of the bank.

It is without a shadow of doubt that there are several classification models appropriate for the setting of our problem, such as the classical Logistic Regression (LR), Random Forest (RF), the more recent Neural Networks (NNs) and Support Vector Machines (SVMs). RF, NN and SVM are more flexible (i.e., no a priori restriction is imposed) when compared with classical statistical modeling (e.g., LR) presenting learning capabilities that range from linear to complex nonlinear mappings. Due to such flexibility, RF,NN and SVM tend to provide more accurate predictions, but the obtained models are difficult to be understood by humans, which is a loss of interpretability. In this paper, a binary classification task was modeled using the four machine learning algorithm that was fed with 16 attributes (after a feature selection step), using 4/5 randomly selected customers for training and 1/5 for testing. In the training samples, we cross validated for different models.

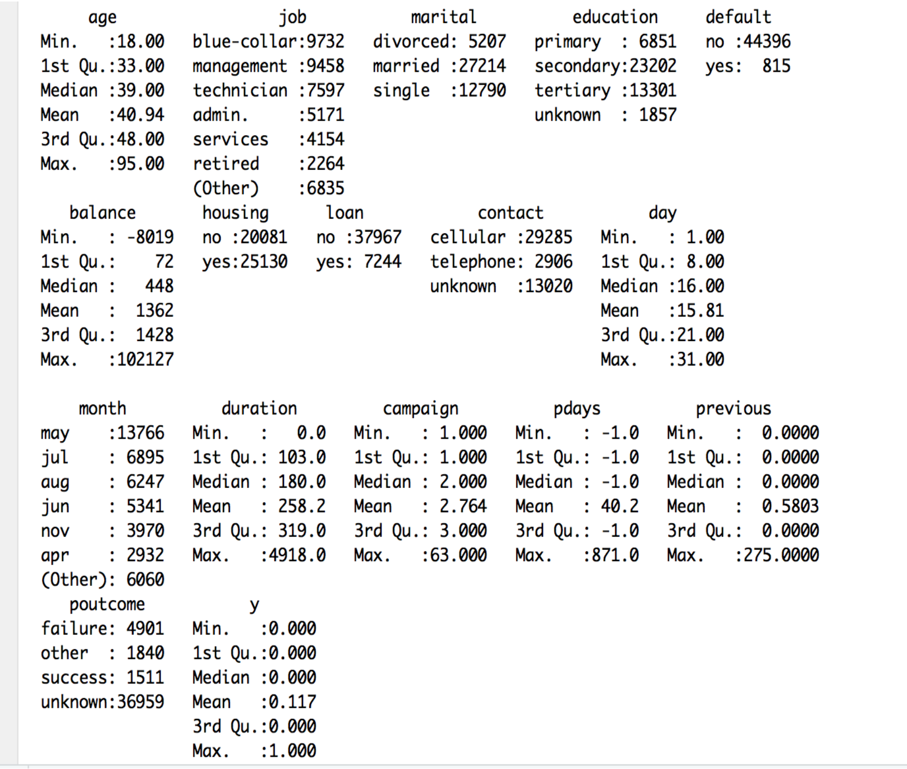
2. Data Description

Our bank data set is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be (or not) subscribed.

The data contains 45,211 instances and 16 features for each instance. Our response variable is whether the customer will make a deposit or not. We want to build different models to our data in order to illustrate how important the role that different features play in predicting our target problem. What's more, we will also compare the performances of different models to find the best model that fits our data set.

The response variable is a categorical variable, for which 1 represents that the customer will buy this product and 0 represents that the customer will not buy the product. As for the categorical predictors, the data contains job (types of job), marital (marital status), education, default (whether customer has credit in default), housing(whether customer has housing loan), loan (has personal loan or not), contact (contact communication type), month (last contact month of year) and poutcome (outcome of the previous marketing campaign).

Also, there are seven numerical predictors: age, balance (average yearly balance, in euros), day (last contact day of the month), duration (last contact duration, in seconds), campaign (number of contacts performed during this campaign and for this client), pdays (number of days that passed by after the client was last contacted from a previous campaign, -1 means client was not previously contacted) and previous (number of contacts performed before this campaign and for this client). The predictors are summarized in figure 2-1 below:



```
      age                 job           marital          education      default  
 Min.   :18.00   blue-collar:9732   divorced: 5207   primary  : 6851   no :44396  
 1st Qu.:33.00   management :9458   married :27214   secondary:23202   yes:  815  
 Median :39.00   technician :7597   single  :12790   tertiary :13301              
 Mean   :40.94   admin.     :5171                    unknown  : 1857              
 3rd Qu.:48.00   services   :4154                                                 
 Max.   :95.00   retired    :2264                                                 
                 (Other)    :6835                                                 
    balance       housing      loan            contact           day       
 Min.   : -8019   no :20081   no :37967   cellular :29285   Min.   : 1.00  
 1st Qu.:    72   yes:25130   yes: 7244   telephone: 2906   1st Qu.: 8.00  
 Median :   448                           unknown  :13020   Median :16.00  
 Mean   :  1362                                             Mean   :15.81  
 3rd Qu.:  1428                                             3rd Qu.:21.00  
 Max.   :102127                                             Max.   :31.00  

     month           duration         campaign          pdays          previous       
 may    :13766   Min.   :   0.0   Min.   : 1.000   Min.   : -1.0   Min.   :  0.0000  
 jul    : 6895   1st Qu.: 103.0   1st Qu.: 1.000   1st Qu.: -1.0   1st Qu.:  0.0000  
 aug    : 6247   Median : 180.0   Median : 2.000   Median : -1.0   Median :  0.0000  
 jun    : 5341   Mean   : 258.2   Mean   : 2.764   Mean   : 40.2   Mean   :  0.5803  
 nov    : 3970   3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.: -1.0   3rd Qu.:  0.0000  
 apr    : 2932   Max.   :4918.0   Max.   :63.000   Max.   :871.0   Max.   :275.0000  
 (Other): 6060                                                                       
    poutcome           y        
 failure: 4901   Min.   :0.000  
 other  : 1840   1st Qu.:0.000  
 success: 1511   Median :0.000  
 unknown:36959   Mean   :0.117  
                 3rd Qu.:0.000  
                 Max.   :1.000  
```

Figure 2-1

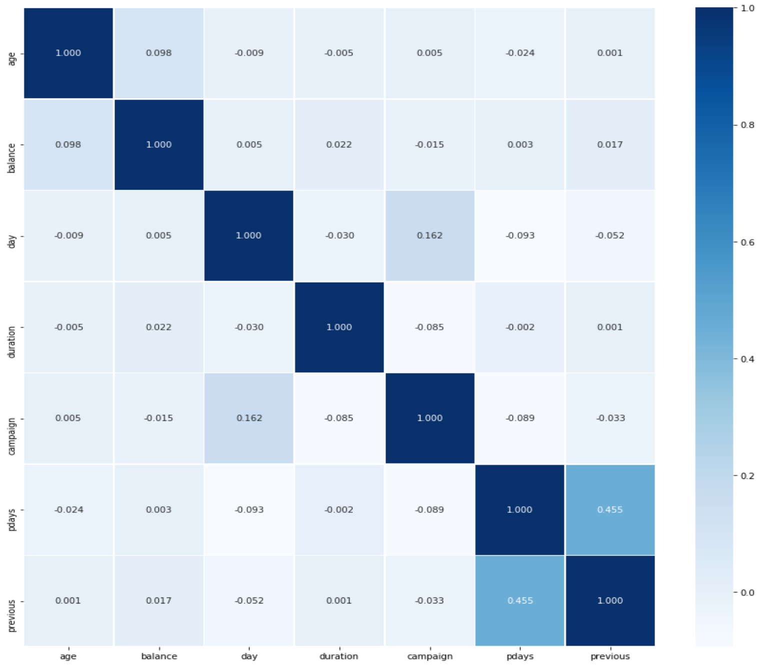


Figure 2-2

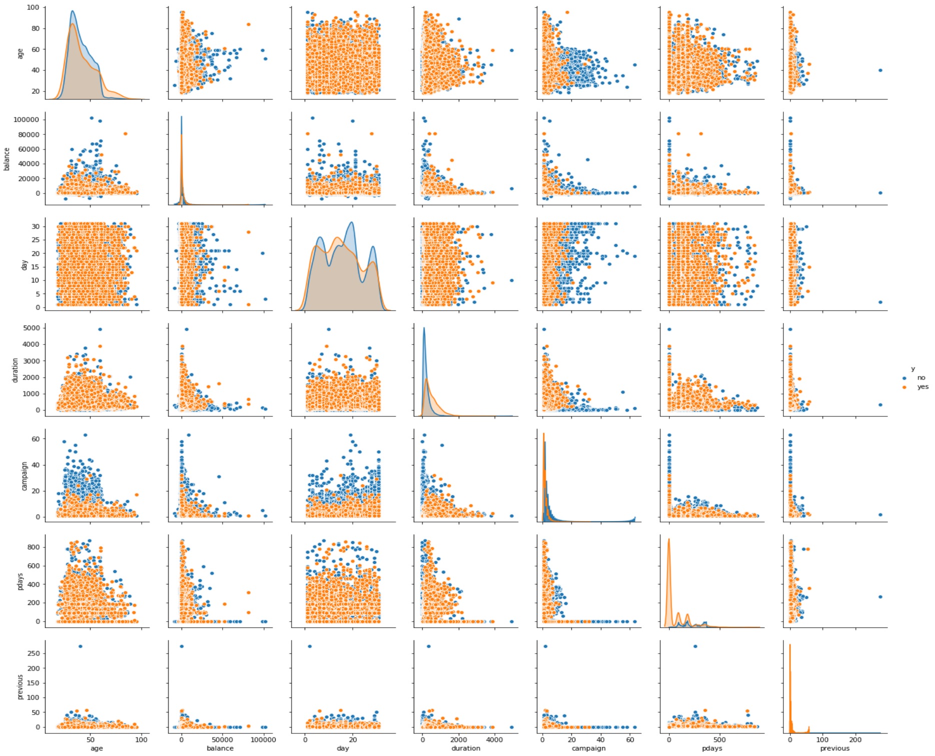


Figure 2-3

Figure 2-2 is a heatmap of the correlation between the numerical variables. The figure indicates that there does exist small correlation between variables. However, the correlations are not very strong except the correlation between p-days and previous, which is 0.455.

Figure 2-3 is the pair plot of numerical predictors based on whether the customers buy the product or not. From the pair plot, we cannot observe obvious correlation between the numerical predictors. In addition, we can observe that the average balance of the people who buy the product is lower than the people who do not buy the product, which accords with the fact since people use their deposit to buy financial product. And the duration of people who buy the product is lower than the people who do not buy the product since the marketing personnels will spend more time to persuade the clients who are not willing to buy the product. The number of contacts performed before this campaign and for those clients who will buy the product is much more than the clients who will not buy the product, since they have more interests in the product and so they are intended to get more information about the product.

3. Models

For our models, we use mainly five supervised learning models: Lasso/Ridge Regression, Logistic Regression, Support Vector Machine, Random Forest and Neural Networks. We will build these models to fit our bank data set to figure out which predictors are significant in the prediction of whether clients of the bank will buy the deposit product. Also, we will evaluate these five models based on their error rates to figure out which model is the best model that can describe the bank data set best.

3.1 Logistic Regression

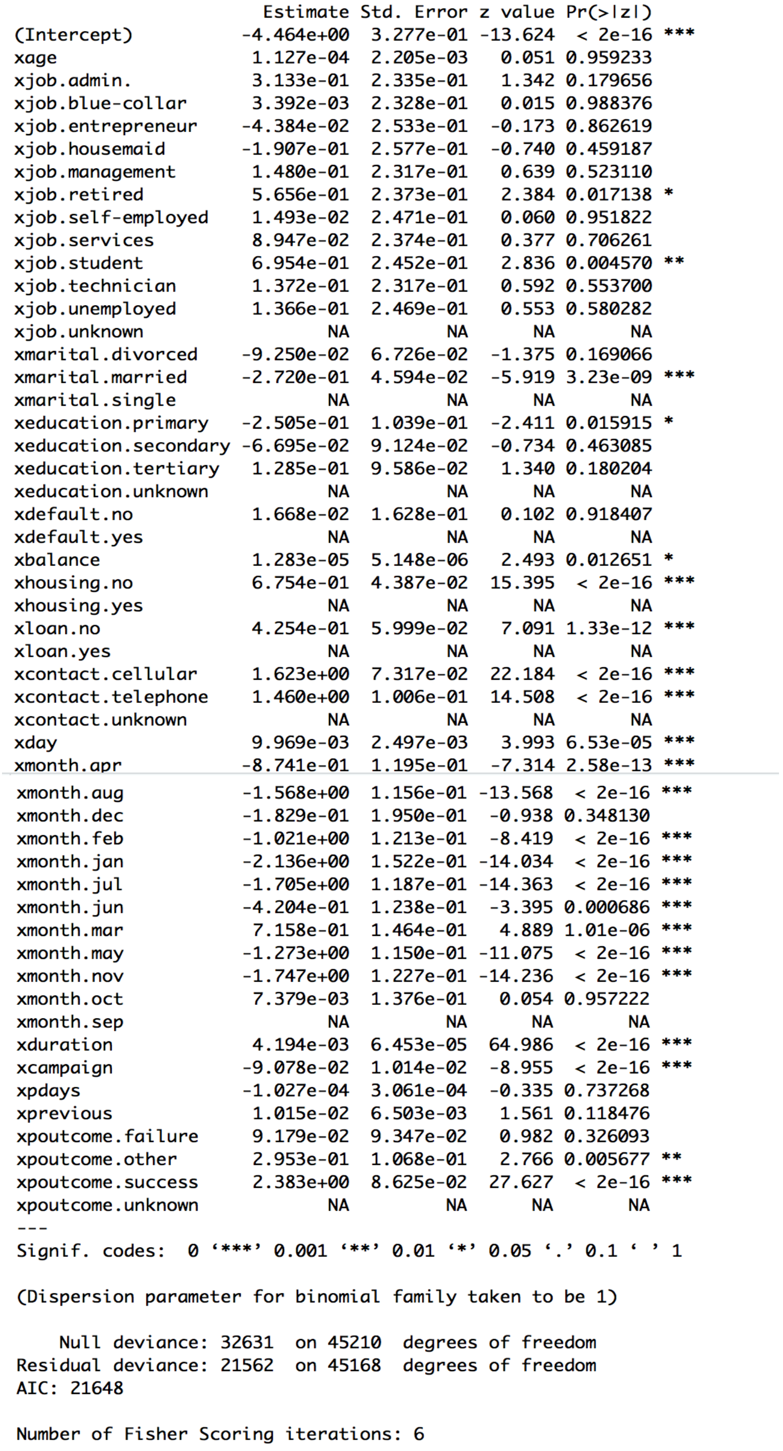
3.1.1 Logistic Regression Methodology

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Mathematically, logistic regression estimates a multiple linear regression function defined as:

$$\text{logit(p)} = \log\left(\frac{p(y=1)}{1-(p=1)}\right) = \sum_{i=1}^{p} \beta_i x_i$$

3.1.2 Logistic Regression Analysis

Figure 3-1 below is the result of fitting logistic regression model. From the p-values of the predictors, we can conclude that the most important features are married status, no housing loan, no personal loan, cellular and telephone contact status, last contact day, last contact month, contact duration, number of contacts and the success outcome status of the last campaign. The predictors age, job, divorced marital status, education, default, pdays, previous and failure outcome status of the last campaign are not significant in the logistic regression model.



```
                      Estimate Std. Error z value Pr(>|z|)    
(Intercept)         -4.464e+00  3.277e-01 -13.624  < 2e-16 ***
xage                 1.127e-04  2.205e-03   0.051 0.959233    
xjob.admin.          3.133e-01  2.335e-01   1.342 0.179656    
xjob.blue-collar     3.392e-03  2.328e-01   0.015 0.988376    
xjob.entrepreneur   -4.384e-02  2.533e-01  -0.173 0.862619    
xjob.housemaid      -1.907e-01  2.577e-01  -0.740 0.459187    
xjob.management      1.480e-01  2.317e-01   0.639 0.523110    
xjob.retired         5.656e-01  2.373e-01   2.384 0.017138 *  
xjob.self-employed   1.493e-02  2.471e-01   0.060 0.951822    
xjob.services        8.947e-02  2.374e-01   0.377 0.706261    
xjob.student         6.954e-01  2.452e-01   2.836 0.004570 ** 
xjob.technician      1.372e-01  2.317e-01   0.592 0.553700    
xjob.unemployed      1.366e-01  2.469e-01   0.553 0.580282    
xjob.unknown                NA         NA      NA       NA    
xmarital.divorced   -9.250e-02  6.726e-02  -1.375 0.169066    
xmarital.married    -2.720e-01  4.594e-02  -5.919 3.23e-09 ***
xmarital.single             NA         NA      NA       NA    
xeducation.primary  -2.505e-01  1.039e-01  -2.411 0.015915 *  
xeducation.secondary -6.695e-02  9.124e-02  -0.734 0.463085    
xeducation.tertiary  1.285e-01  9.586e-02   1.340 0.180204    
xeducation.unknown          NA         NA      NA       NA    
xdefault.no          1.668e-02  1.628e-01   0.102 0.918407    
xdefault.yes                NA         NA      NA       NA    
xbalance             1.283e-05  5.148e-06   2.493 0.012651 *  
xhousing.no          6.754e-01  4.387e-02  15.395  < 2e-16 ***
xhousing.yes                NA         NA      NA       NA    
xloan.no             4.254e-01  5.999e-02   7.091 1.33e-12 ***
xloan.yes                   NA         NA      NA       NA    
xcontact.cellular    1.623e+00  7.317e-02  22.184  < 2e-16 ***
xcontact.telephone   1.460e+00  1.006e-01  14.508  < 2e-16 ***
xcontact.unknown            NA         NA      NA       NA    
xday                 9.969e-03  2.497e-03   3.993 6.53e-05 ***
xmonth.apr          -8.741e-01  1.195e-01  -7.314 2.58e-13 ***
xmonth.aug          -1.568e+00  1.156e-01 -13.568  < 2e-16 ***
xmonth.dec          -1.829e-01  1.950e-01  -0.938 0.348130    
xmonth.feb          -1.021e+00  1.213e-01  -8.419  < 2e-16 ***
xmonth.jan          -2.136e+00  1.522e-01 -14.034  < 2e-16 ***
xmonth.jul          -1.705e+00  1.187e-01 -14.363  < 2e-16 ***
xmonth.jun          -4.204e-01  1.238e-01  -3.395 0.000686 ***
xmonth.mar           7.158e-01  1.464e-01   4.889 1.01e-06 ***
xmonth.may          -1.273e+00  1.150e-01 -11.075  < 2e-16 ***
xmonth.nov          -1.747e+00  1.227e-01 -14.236  < 2e-16 ***
xmonth.oct           7.379e-03  1.376e-01   0.054 0.957222    
xmonth.sep                  NA         NA      NA       NA    
xduration            4.194e-03  6.453e-05  64.986  < 2e-16 ***
xcampaign           -9.078e-02  1.014e-02  -8.955  < 2e-16 ***
xpdays              -1.027e-04  3.061e-04  -0.335 0.737268    
xprevious            1.015e-02  6.503e-03   1.561 0.118476    
xpoutcome.failure    9.179e-02  9.347e-02   0.982 0.326093    
xpoutcome.other      2.953e-01  1.068e-01   2.766 0.005677 ** 
xpoutcome.success    2.383e+00  8.625e-02  27.627  < 2e-16 ***
xpoutcome.unknown           NA         NA      NA       NA    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 32631  on 45210  degrees of freedom
Residual deviance: 21562  on 45168  degrees of freedom
AIC: 21648

Number of Fisher Scoring iterations: 6
```

Figure 3-1

3.2 . Lasso/ Ridge Regression

3.2.1 Lasso/ Ridge Regression Methodology

Ridge and Lasso regression are some of the simple techniques to reduce model complexity and prevent overfitting which may result from simple linear regression. The difference between ridge regression, lasso regression and linear regression is that ridge regression and lasso regression add a penalty term on the loss function.

In ridge regression, the cost function is altered by adding a penalty equivalent to square of the magnitude of the coefficients. The object function of ridge regression is to minimize the function:

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^{p} w_j^2 \quad .$$

Ridge regression puts constraint on the coefficients (w). The penalty term (lambda) regularizes the coefficients such that if the coefficients take large values the optimization function is penalized. So, ridge regression shrinks the coefficients and it helps to reduce the model complexity and multi-collinearity. The cost function for Lasso (least absolute shrinkage and selection operator) regression can be written as:

.

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^{p} |w_j|$$

This type of regularization (L1) can lead to zero coefficients i.e. some of the features are completely neglected for the evaluation of output. So Lasso regression not only helps in reducing overfitting but it can help us in feature selection.

As for the bank data, there does exist collinearity between the features, which increases the variance of estimation. So we construct ridge and lasso regression model instead of building linear regression model in order to stabilize our estimation.

3.2.2 Lasso/ Ridge Regression Parameter Tuning

In the object functions of lasso and ridge regression, the penalty coefficient (lambda) need to be chosen in order to achieve the minimum prediction error.

Figure 3-2 is the result of cross validation of the penalty parameter of ridge regression. Here, we choose the lambda which can achieve the maximum AUC (lambda = 0.01391628).

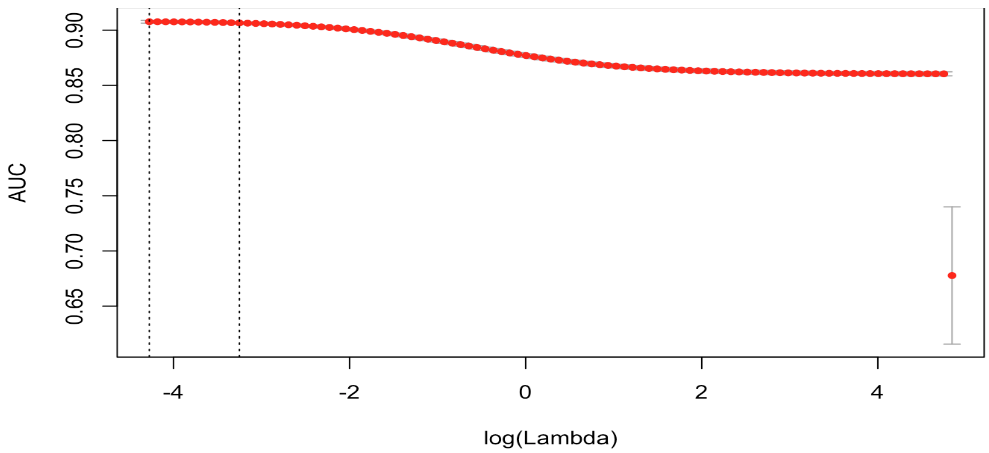


Figure 3-2

Figure 3-3 is the result of cross validation of the penalty parameter of lasso regression. Here, we choose the lambda = 0.003068721. Figure 3-4 is the plot of coefficients against log lambda, it shows that L1 penalty can shrink some coefficients directly to zeros.

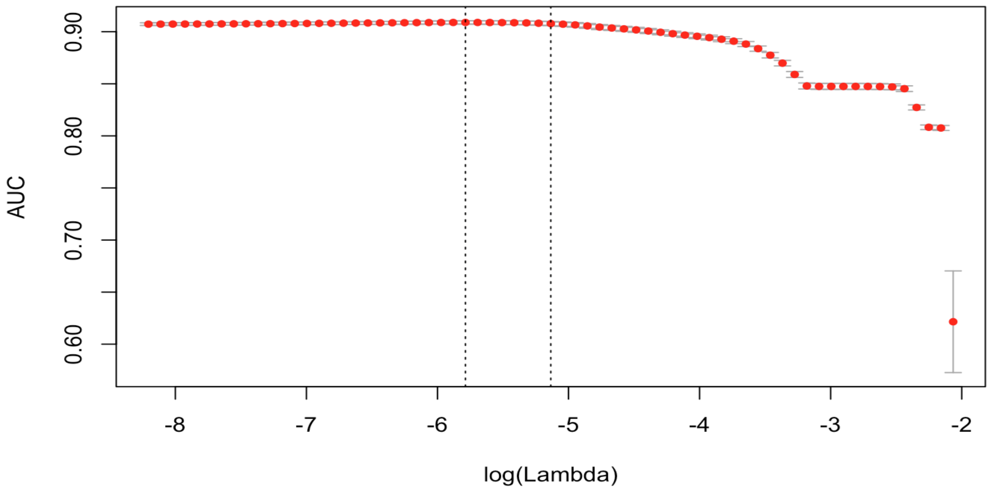


Figure 3-3

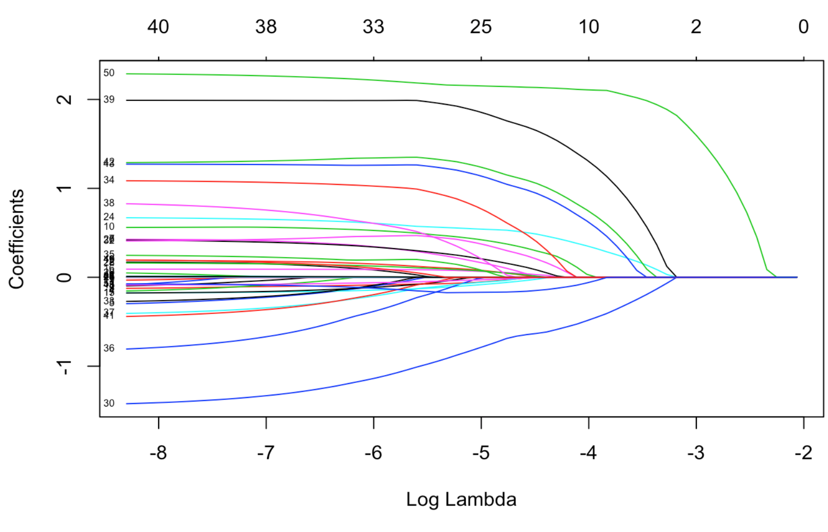
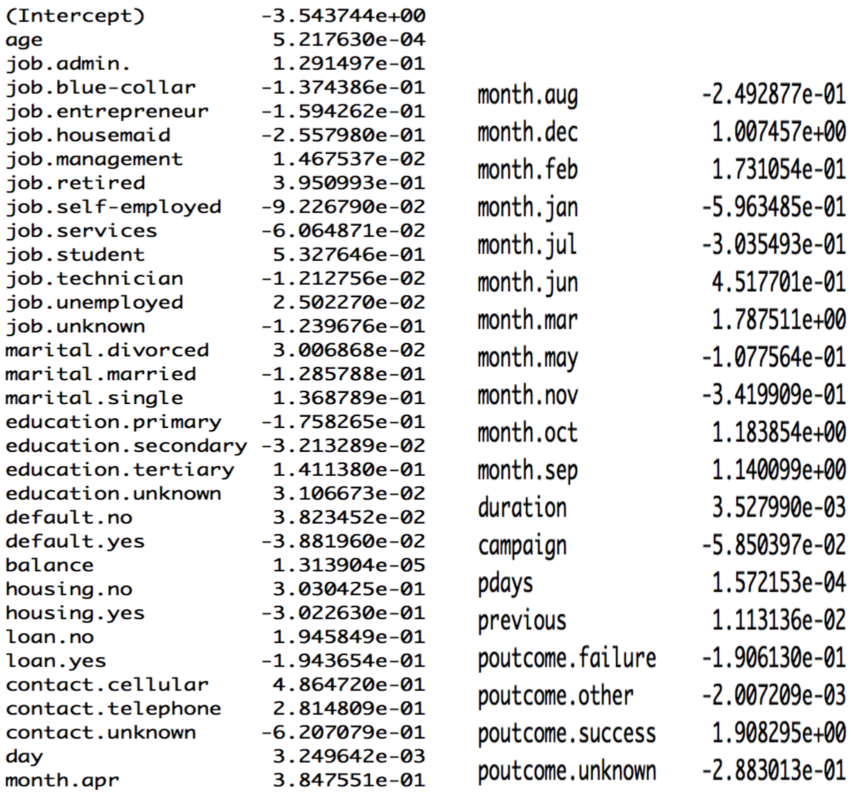


Figure 3-4

3.2.3 Lasso/ Ridge Regression Analysis

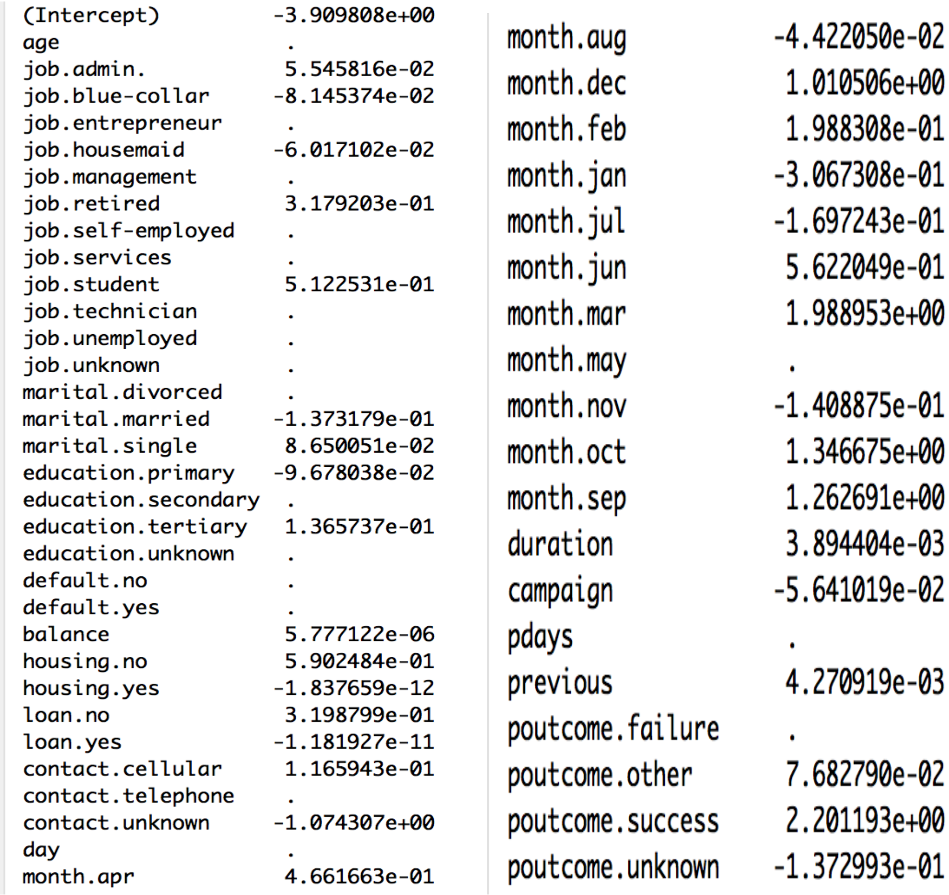


```
(Intercept)          -3.543744e+00
age                   5.217630e-04
job.admin.            1.291497e-01
job.blue-collar      -1.374386e-01
job.entrepreneur     -1.594262e-01
job.housemaid        -2.557980e-01
job.management        1.467537e-02
job.retired           3.950993e-01
job.self-employed    -9.226790e-02
job.services         -6.064871e-02
job.student           5.327646e-01
job.technician       -1.212756e-02
job.unemployed        2.502270e-02
job.unknown          -1.239676e-01
marital.divorced      3.006868e-02
marital.married      -1.285788e-01
marital.single        1.368789e-01
education.primary    -1.758265e-01
education.secondary  -3.213289e-02
education.tertiary    1.411380e-01
education.unknown     3.106673e-02
default.no            3.823452e-02
default.yes          -3.881960e-02
balance               1.313904e-05
housing.no            3.030425e-01
housing.yes          -3.022630e-01
loan.no               1.945849e-01
loan.yes             -1.943654e-01
contact.cellular      4.864720e-01
contact.telephone     2.814809e-01
contact.unknown      -6.207079e-01
day                   3.249642e-03
month.apr             3.847551e-01
```

```
month.aug          -2.492877e-01
month.dec           1.007457e+00
month.feb           1.731054e-01
month.jan          -5.963485e-01
month.jul          -3.035493e-01
month.jun           4.517701e-01
month.mar           1.787511e+00
month.may          -1.077564e-01
month.nov          -3.419909e-01
month.oct           1.183854e+00
month.sep           1.140099e+00
duration            3.527990e-03
campaign           -5.850397e-02
pdays               1.572153e-04
previous            1.113136e-02
poutcome.failure   -1.906130e-01
poutcome.other     -2.007209e-03
poutcome.success    1.908295e+00
poutcome.unknown   -2.883013e-01
```

Figure 3-4

Figure 3-4 is the coefficient of ridge regression. The result indicates that dec, mar, oct, sep categories of month and the success category of poutcome have the largest partial effect on the prediction of whether a client will buy product. It means that the last contact happened in these months and the success outcome from the previous campaign have the most positive effect on predicting the client will buy the product. On the contrast, the predictor age has the smallest partial effect on the prediction.



```
(Intercept)          -3.909808e+00
age                   .
job.admin.            5.545816e-02
job.blue-collar      -8.145374e-02
job.entrepreneur      .
job.housemaid        -6.017102e-02
job.management        .
job.retired           3.179203e-01
job.self-employed     .
job.services          .
job.student           5.122531e-01
job.technician        .
job.unemployed        .
job.unknown           .
marital.divorced      .
marital.married      -1.373179e-01
marital.single        8.650051e-02
education.primary    -9.678038e-02
education.secondary   .
education.tertiary    1.365737e-01
education.unknown     .
default.no            .
default.yes           .
balance               5.777122e-06
housing.no            5.902484e-01
housing.yes          -1.837659e-12
loan.no               3.198799e-01
loan.yes             -1.181927e-11
contact.cellular      1.165943e-01
contact.telephone     .
contact.unknown      -1.074307e+00
day                   .
month.apr             4.661663e-01
```

```
month.aug            -4.422050e-02
month.dec             1.010506e+00
month.feb             1.988308e-01
month.jan            -3.067308e-01
month.jul            -1.697243e-01
month.jun             5.622049e-01
month.mar             1.988953e+00
month.may             .
month.nov            -1.408875e-01
month.oct             1.346675e+00
month.sep             1.262691e+00
duration              3.894404e-03
campaign             -5.641019e-02
pdays                 .
previous              4.270919e-03
poutcome.failure      .
poutcome.other        7.682790e-02
poutcome.success      2.201193e+00
poutcome.unknown     -1.372993e-01
```

Figure 3-5

Figure 3-5 is the coefficients of lasso regression. The L1 penalty of lasso regression shrinks the coefficients of age, job (entrepreneur, job.management, self-employed, services, technician, unemployed categories), divorced marital status, secondary education status, default, telephone contact status, etc to zeros. In lasso regression, the predictors month and poutcome have the largest partial effect on the prediction, which is the same as the result from ridge regression.

However, since ridge and lasso regression may lead to the predicted response not equal to 0 or 1. So we also built support vector machine model, random forest and neural network model to fit the bank data.

3.3. Support Vector Machine

3.3.1 Methodology

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. The objective of SVM is to find a hyperplane which maximize the margin separating the two categories of data. It is a classification method of the supervised learning.

The following formula formally describes SVM.

$$\begin{array}{ll} \min_{\mathbf{v_H},c} & \|\mathbf{v_H}\|_{\mathcal{F}}^2 + \gamma \sum_{i=1}^{n} \xi^2 \\ \text{s.t.} & \tilde{y}_i \left(\langle \mathbf{v_H}, \phi\left(\tilde{\mathbf{x}}_i\right) \rangle_{\mathcal{F}} - c \right) \geq 1 - \xi_i \quad \text{and } \xi_i \geq 0 \end{array}$$

Where □□ is the support vectors, γ is the penalty term of the misclassified object, ϕ is a transforming function of the data to a higher dimensional space. The variables ξi are slack variables. To solve the problem implicitly, we can solve a dual optimization problem:

$$\begin{array}{ll} \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} & W(\boldsymbol{\alpha}) := \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \tilde{y}_i \tilde{y}_j \left(k\left(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j\right) + \frac{1}{\gamma} \mathbb{I}\{i = j\} \right) \\ \text{s.t.} & \sum_{i=1}^{n} \tilde{y}_i \alpha_i = 0 \quad \text{and} \quad \alpha_i \geq 0 \end{array}$$

The classifier is:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \tilde{y}_i \alpha_i^* k\left(\tilde{\mathbf{x}}_i, \mathbf{x}\right) - c \right)$$

Where *k* is the kernel of the SVM. There a many choices of the kernels, such as the linear kernel, the Gaussian kernel, etc. In this paper, we choose the Gaussian kernel:

$$k_{\mathrm{RBF}}\left(\mathbf{x}, \mathbf{x}'\right) := \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\sigma^2}\right)$$

3.3.2 Parameter tuning

Once the model is designed and determined, we will evaluate the parameters using the cross validation. We will choose the appropriate γwhich is the penalty term, the larger γ, the larger cost of allowing a point on the wrong side of the SVM classifier. The metric we use here is the classification accuracy. Figure 3-6 shows the classification accuracy rate of different penalty terms. We can see all the results appear similar, indicating that the SVM model is quite robust. The best result is 89.13, when $\gamma$is equal to 10.

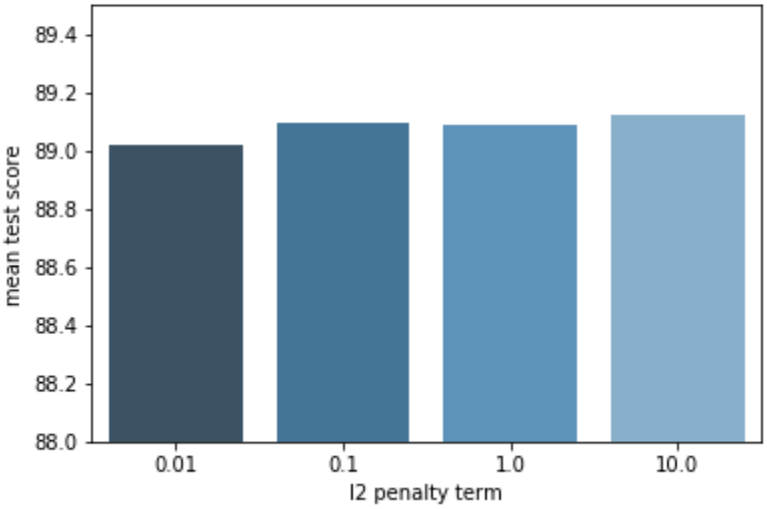


Figure 3-6

3.4. Random Forest

Random forests, or random decision forests are an ensemble learning method for classification and regression. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The advantage of Random forests is that it correct for decision trees' habit of overfitting to the training-set, which decreases the variance of the model and thus improve the MSE of the model, whereas the disadvantage is that every time we fit a tree to a bootstrap sample, we get a different tree, and this cause the loss of interpretability.

3.4.1 Methodology

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging to tree learners. Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1, ..., B$:

1. Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.
2. Train a classification or regression tree f_b on X_b, Y_b.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual classification trees on x':

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

For each individual tree, we use CART based on the split criterion of Gini impurity, which is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.

3.4.2 Parameter tuning

As what we have done in the previous model of logistic regression and SVM, we use the cross validation for different hyper-parameters of Random Forest. We choose to test the n_estimators, which is the number of trees as well as the maximum depth of each sub trees.

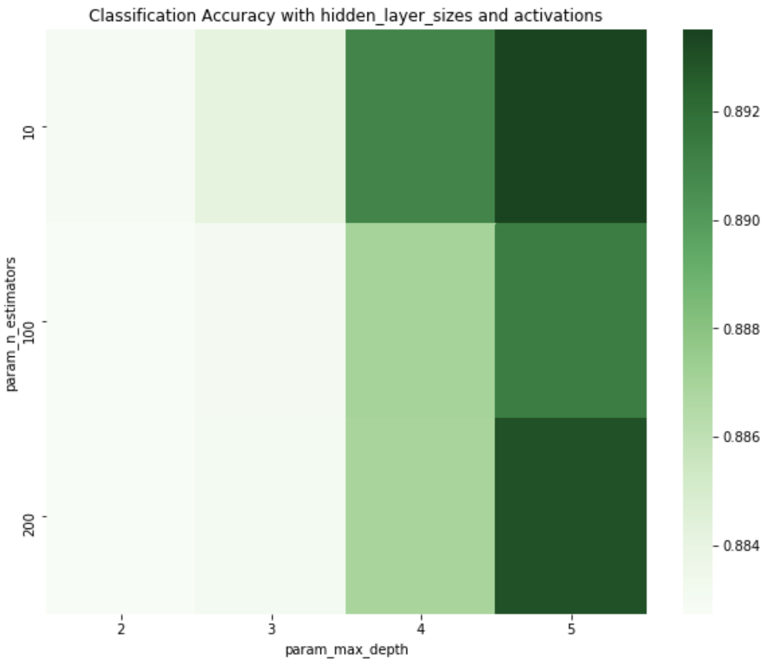


Figure 3-7

Fig 3-7 shows a heatmap of the combination of the number of trees and maximum depth of each tree. The metric we use is still the classification accuracy. The overall model seems to be good since the last accuracy is 0.884. The largest accuracy is 0.895, when the maximum tree depth is 5 and amount of trees is 10.

3.4.3 Feature importance

Different with the coefficients or the significance level of the logistic regression which is interpretable, it is hard to give a "coefficient" to the Random Forests mode. However, what we can do is to calculate the feature importance. When a tree is built, the decision about which variable to split at each node uses a calculation of the Gini impurity. For each variable, the sum of the Gini decrease across every tree of the forest is accumulated every time that variable is chosen to split a node. The sum is divided by the number of trees in the forest to give an average. The scale is irrelevant: only the relative values matter.

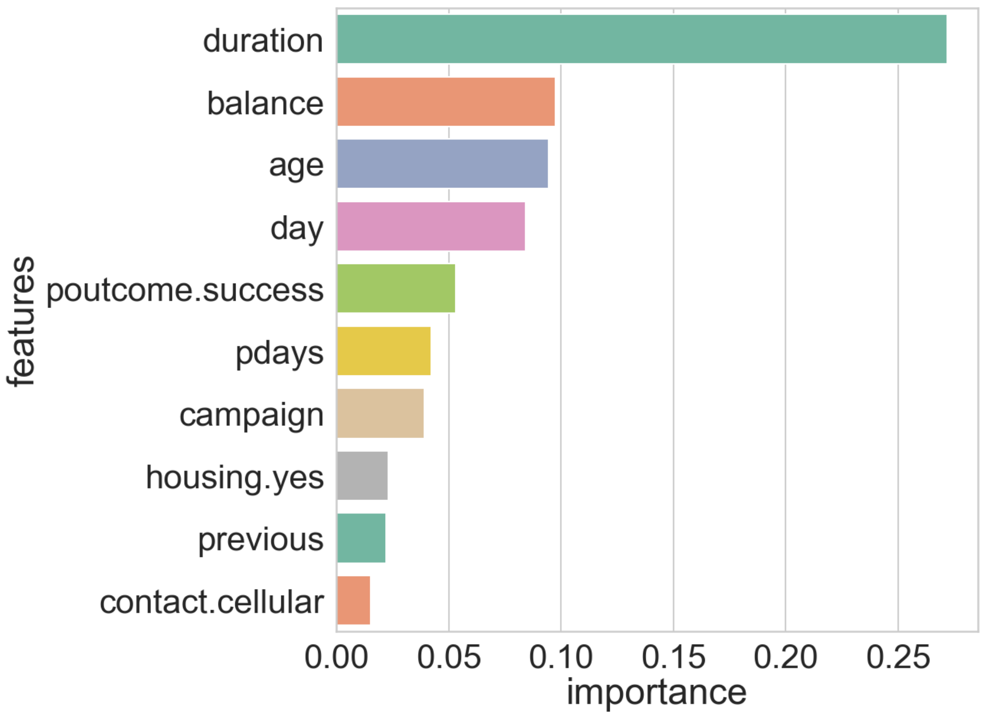


Figure 3-8

Figure 3-8 shows the importance of the features. It should be emphasized that we do not know the "direction" of each variable, but just the magnitude of importance. Amongst the features, duration plays the most significant role, followed by balance, age and day. Duration is the time duration of the sales person contacting with the client. It makes sense that the longer time they talk, the larger possibility the client will buy the product.

3.5 Neural Network

A neural network is a supervised learning method which can be applied to both regression and classification problems. The central idea of neural network is to extract linear combinations of the inputs as derived features, and then model the target as a nonlinear function of these features.

3.5.1 Methodology

Figure 3-9 shows the structure of a feedforward neural network. There are three layers: the input layer, the hidden layer and the output layer.

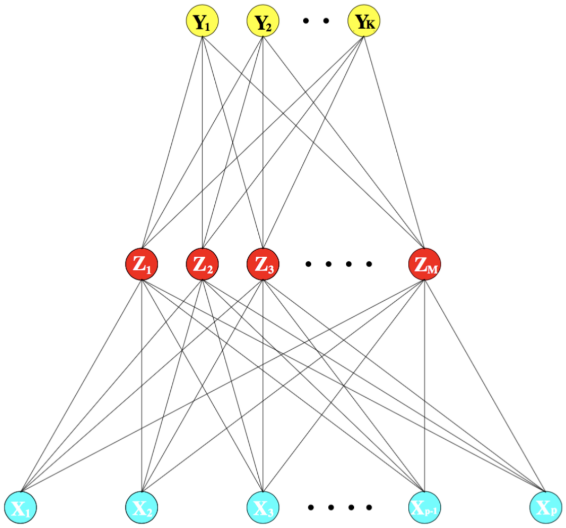


Figure 3-9

A general description of the neural network is:

In the input layer, derived features Zm are obtained by applying the activation function σ to linear combinations of the inputs:

$$Z_m = \sigma\left(\alpha_{0m} + \alpha_m^T X\right), m = 1, \ldots, M$$

In the hidden layer, the target Tk is modeled as a function of linear combinations of Zm:

$$T_k = \beta_{0k} + \beta_k^T Z, \quad k = 1, \ldots, K$$

In the output layer, the output function gk(T) allows a final transformation of the vector of outputs T:

$$f_k(X) = g_k(T), \quad k = 1, \ldots, K$$

Where $\sigma$is the activation function. There are several choices of the activation function, such as the sigmoid function, the relu activation, and the tanh function, etc. For the K-classification, we use the softmax function:

$$g_k(T) = \frac{e^{T_k}}{\sum_{l=1}^{K} e^{T_l}}$$

When fitting the weight of the model, we use the back-propagation and the gradient descent.

3.5.2 Parameter tuning

We use 10 fold cross validation to choose the parameters of the hidden layer size and the activation function. From Figure 3-10, we can see that the logistic function, which is a sigmoid function does a better job than the others. The best parameters combination are hidden layer size 50 and logistic activation function.

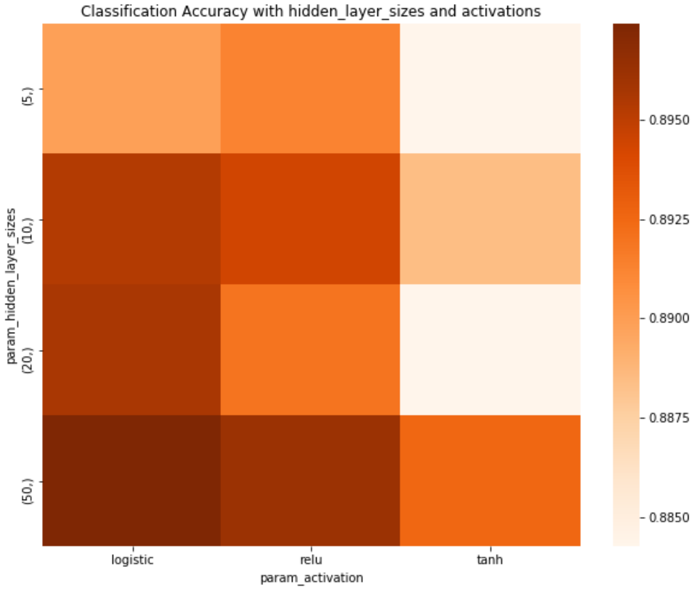


Figure 3-10

3.6 Model Comparison

After we have trained the models we compare with each other. In this model comparison process, we split the data into the training data and test data with the percentage of 80% training and 20% test data. We first train the data for each model and use two metrics: ROC-AUC and the classification accuracy.

AUC-ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s.

We generate both the 0-1 prediction for measure the classification accuracy, and the probability for calculating AUC-ROC for the four models: Logistic regression with penalty, Random Forest, SVM and Neural Network. We pick the best performances of the parameters achieved by the cross validation of them relatively. We find that the Random Forest performs the best in both ROC-AUC and Classification Accuracy. The result is shown in Table 3-1 and Figure 3-11.

Table 3-1

Comparison of models.(bold denotes the best value).

Metric	Logistic Regression	Random Forest	SVM	Neural Network
ROC-AUC	0.81	**0.89**	0.75	0.86
Classification Accuracy	0.85	**0.92**	0.88	0.90

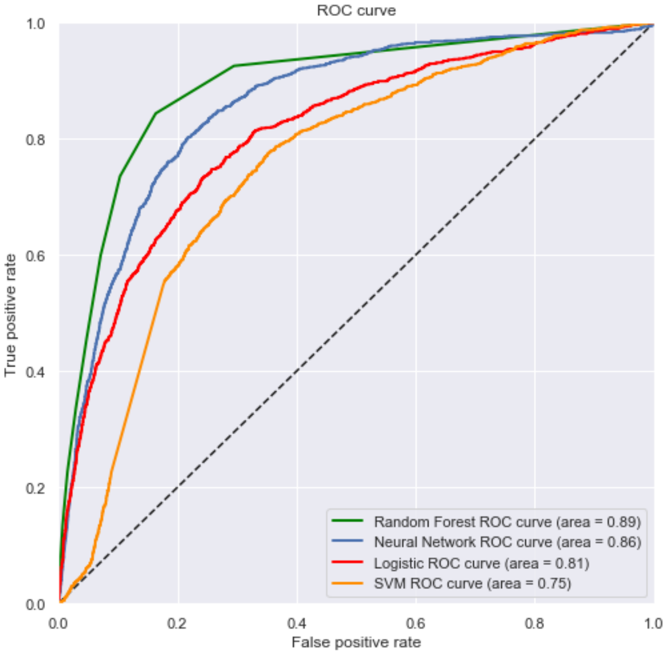


Figure 3-11

4. Conclusion

Under this context, we predict the result of a telemarketing phone call to sell long term deposits is a valuable tool to support client selection decisions of bank campaign managers. In this study, we uses a machine learning approach for the selection of bank telemarketing clients. We analyzed a recent and large Portuguese bank dataset, collected from 2008 to 2013,with a total of 45,211 records. The goal was to model the success of subscribing a long-term deposit using attributes that were known.

During the modeling phase, we used 16 relevant features with no scaling of data. In the logistic regression model, we can conclude that the most important features are married status, no housing loan, no personal loan, cellular and telephone contact status, last contact day, last contact month, contact duration, number of contacts and the success outcome status of the last campaign. For example, married status has a negative effect to the model. As an interesting outcome from Random Forest model, the duration of the call, which highly affects the probability of a success contact or the amount that is deposited in the bank, is outstandingly significant. Also the average yearly balance, and age variables.

Finally. four ML models were compared: logistic regression, random forest, neural networks and support vector machines. These models were compared using two metrics, area of the receiver operating characteristic curve (AUC) and the prediction accuracy. The best results were obtained by the RFs, which resulted in an ROC-AUC of 0.89, and Classification Accuracy of 0.92. Such AUC corresponds to a very good discrimination.

Reference

Saptashwa. (Sep 26, 2018). Ridge and Lasso Regression: A Complete Guide with Python Scikit-Learn. Retrieved from https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b

Ricardo Carvalho. How to use Ridge Regression and Lasso in R. Retrieved from http://ricardoscr.github.io/how-to-use-ridge-and-lasso-in-r.html

Paulo Cortez, Data mining with neural networks and support vector machines using the r/rminer tool, Advances in Data Mining. Applications and Theoretical Aspects, 6171, Springer, 2010, pp. 572–583.

Ian H.Witten, Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition Morgan Kaufmann, 2005.

Appendix

- Link to the data set: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#

- Group contribution
  Yuting He: data description, lasso regression, ridge regression, logistic regression
  Zhiying Li: SVM, Random forest, Neural Network, model comparison
  Xiaoming Chen: report introduction, report conclusion, poster

R code

```
---
```

~~~
title: "5223project"
author: "Yuting He"
date: "2019/4/27"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r}
setwd("~/Desktop")
data <- read.csv("raw_data.csv")
data$y <- ifelse(data$y == "no", 0, 1)
Y <- data$y
head(data)
summary(data)
```
```{r}
library(dummies)
M <- data[,-ncol(data)]
M  <- dummy.data.frame(M, sep = ".")
head(M)
data <- cbind(Y,M)
head(data)
```
##ridge regression
```{r}
library(glmnet)
x <- as.matrix(data[,-1]) # Removes class
y <- as.double(as.matrix(data[, 1])) # Only class
```
```{r}
# Fitting the model (Ridge: Alpha = 0)
set.seed(999)
cv.ridge <- cv.glmnet(x, y, family='binomial', alpha=0, parallel=TRUE, standardize=TRUE,
type.measure='auc')

# Results
plot(cv.ridge)
~~~

~~~
cv.ridge$lambda.min
cv.ridge$lambda.1se
coef(cv.ridge, s=cv.ridge$lambda.min)

```
##lasso
```{r}
cv.lasso <- cv.glmnet(x, y, family='binomial', alpha=1, parallel=TRUE, standardize=TRUE, type.measure='auc')

# Results
plot(cv.lasso)
plot(cv.lasso$glmnet.fit, xvar="lambda", label=TRUE)
cv.lasso$lambda.min
cv.lasso$lambda.1se
coef(cv.lasso, s=cv.lasso$lambda.min)
```
##logistic regression

```{r}
glm.fit <- glm(y ~ x, data = data, family = binomial)
summary(glm.fit)
```
~~~

Python code

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Apr 25 19:36:07 2019

@author: lizhiying
"""



import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
```

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.model_selection import GridSearchCV

from tqdm import tqdm

####read the dataset####
data =  pd.read_csv('/Users/lizhiying/Desktop/Multivariate/bank/raw_data.csv')


#---------------------------
'''EDA'''

sns.set_style("whitegrid")
sns.set_palette(sns.color_palette("Paired"))

ax = plt.axes()
sns.countplot(x='y', data=data, ax=ax)
ax.set_title('Target class distribution')
plt.show()

cols = ['age','balance','day','duration','campaign','pdays','previous','y']
eda = data[cols]




'''Correlation heatmap'''
f,ax = plt.subplots(figsize=(15, 15))
sns.heatmap(eda.corr(), annot=True, linewidths=.5, fmt= '.3f',ax=ax,cmap = 'Blues')
plt.show()


'''Scatter plot'''
```

```
sns.pairplot(eda,diag_kind="kde")

sns.pairplot(eda,hue='y')


'''Paired plot'''
f, axarr = plt.subplots(2, 2, figsize=(15, 15))

sns.boxplot(x='age', y='y', data=data, showmeans=True, ax=axarr[0,0])
sns.boxplot(x='balance', y='y', data=data, showmeans=True, orient = 'h',ax=axarr[0, 1]).set(xscale = "log")
sns.boxplot(x='duration', y='y', data=data, showmeans=True, ax=axarr[1, 0])
sns.boxplot(x='campaign', y='y', data=data, showmeans=True, ax=axarr[1, 1])


axarr[0, 0].set_title('age')
axarr[0, 1].set_title('education')
axarr[1, 0].set_title('duration')
axarr[1, 1].set_title('campaign')

#plt.tight_layout()
plt.show()


#------------------------------------------------------------
'''Model'''

df =  pd.read_csv('/Users/lizhiying/Desktop/Multivariate/bank/processed_data.csv',sep = ',')


y = df["Y"]
X = df.ix[:,1:44]



X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)



#------------------------------------------------------------
```

```
'''Logistic'''

lr = LogisticRegression(solver = 'lbfgs')
parameters = {'C':[0.01,0.1,1,10]}
clf = GridSearchCV(lr, parameters, cv=10)
clf.fit(X_train,y_train)

#clf.cv_results_
plot = sns.barplot(parameters.get('C'),clf.cv_results_['mean_test_score']*100,palette="Blues_d")

plot.set_ylim(88.8,89.3)
plot.set_ylabel("mean test score")
plot.set_xlabel("l2 penalty term")
plot.show()

#-------------------------------------------------------------
'''Random Forest'''

rfc = RandomForestClassifier(n_estimators=100)
rfc.fit(X_train,y_train)

# Feature importance
feature_importances = pd.DataFrame(rfc.feature_importances_,index = X_train.columns,
                                   columns=['importance']).sort_values('importance', ascending =False)

feature_importances['features'] = feature_importances.index.values



f,ax = plt.subplots(figsize=(15, 15))
sns.set_context("poster",font_scale = 2)
sns.barplot(y="features", x="importance", data=feature_importances.iloc[0:10,:],
            label="Feature Importance", palette="Set2")


#cross validation
rfc = RandomForestClassifier()
parameters = {'n_estimators':[10,100,200],'max_depth':[2,3,4,5]}
clf = GridSearchCV(rfc, parameters, cv=5)
clf.fit(X_train,y_train)
```

```
pvt = pd.pivot_table(pd.DataFrame(clf.cv_results_),
    values='mean_test_score', index='param_n_estimators', columns='param_max_depth')

plt.figure(figsize=(10,8))
plt.title('Classification Accuracy with hidden_layer_sizes and activations')
ax = sns.heatmap(pvt,cmap="Greens")
plt.show()


#---------------------------------------------------
'''SVM'''

svc = SVC(kernel = 'rbf')
parameters = {'C':[0.01,0.1,1]}
clf = GridSearchCV(svc, parameters, cv=5)
clf.fit(X_train,y_train)

plot = sns.barplot(parameters.get('C'),clf.cv_results_['mean_test_score']*100,palette="Blues_d")

plot.set_ylim(88.26,88.28)
plot.set_ylabel("mean test score")
plot.set_xlabel("l2 penalty term")




#----------------------------------------------------
''' Neural Network '''

mlp = MLPClassifier(activation='relu', solver='adam')
parameters = {'hidden_layer_sizes':[(5,),(10,),(20,),(50,)],'activation' : ["relu","logistic","tanh"]}
clf = GridSearchCV(mlp, parameters, cv=3)
clf.fit(X_train,y_train)

pvt = pd.pivot_table(pd.DataFrame(clf.cv_results_),
    values='mean_test_score', index='param_hidden_layer_sizes', columns='param_activation')
```

```
plt.figure(figsize=(10,8))
plt.title('Classification Accuracy with hidden_layer_sizes and activations')
ax = sns.heatmap(pvt,cmap="Oranges")
plt.show()

#--------------------------------------------------
''' AUC ROC'''

lr = LogisticRegression(solver = 'lbfgs',C = 10).fit(X_train,y_train)
y_pred_lr = lr.predict_proba(X_test)[:,1]
fpr_lr, tpr_lr, _ = roc_curve(y_test, y_pred_lr)
auc_lr = roc_auc_score(y_test,y_pred_lr)



rfc = RandomForestClassifier(n_estimators=10).fit(X_train,y_train)
y_pred_rf = rfc.predict_proba(X_test)[:,1]
fpr_rf, tpr_rf, _ = roc_curve(y_test, y_pred_rf)
auc_rf = roc_auc_score(y_test,y_pred_rf)



svc = SVC(kernel = 'rbf',C = 0.01, probability =True).fit(X_train,y_train)
y_pred_sv = svc.predict_proba(X_test)[:,1]
fpr_sv, tpr_sv, _ = roc_curve(y_test, y_pred_sv)
auc_sv = roc_auc_score(y_test,y_pred_sv)



mlp = MLPClassifier(activation='relu', solver='adam',hidden_layer_sizes =
(10,)).fit(X_train,y_train)
y_pred_nn = mlp.predict_proba(X_test)[:,1]
fpr_nn, tpr_nn, _ = roc_curve(y_test, y_pred_nn)
auc_nn = roc_auc_score(y_test,y_pred_nn)
```

```
plt.subplots(figsize=(8, 8))
plt.xlim(0, 1)
plt.ylim(0, 1)

plt.plot([0, 1], [0, 1], 'k--')
plt.plot(fpr_rf, tpr_rf,color = 'green',label = 'Random Forest ROC curve (area = {0:0.2f})'
          ''.format(auc_rf),lw=2)
plt.plot(fpr_nn, tpr_nn, label = 'Neural Network ROC curve (area = {0:0.2f})'
          ''.format(auc_nn),lw=2)
plt.plot(fpr_lr, tpr_lr,color = 'red',label = 'Logistic ROC curve (area =
{0:0.2f})'''.format(auc_lr),lw=2)

plt.plot(fpr_sv, tpr_sv ,color = 'darkorange',label = 'SVM ROC curve (area = {0:0.2f})'
         ''.format(auc_sv),lw=2)

plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('ROC curve')
plt.legend(loc='best')
ax.grid(True)
plt.show()
```