2020

DATA MINING II PROJECT PROPOSAL

SECTION 001, GROUP 5

LI, HENG (M13457579)
KAVADI, SANDEEP (M13497714)
CHEN, ZHUO (M13500813)
JUMANI, ALI (M13469318)

Executive Summary

This bank dataset contains 41188 rows, 20 variables, these variables can be separated into four categories including client data, socio economic factors and campaign contact data. The dataset is reasonably well established. However, there are some missing values in various variables that need to be cleaned.

We have built five models to predict whether the client has subscribed a term deposit. There are 3 models that have been covered in our class, they are GLM, Random Forest and Neural Network. We also performed XGBoost and Support Vector Machine models as additional models that have not been covered in our lecture. The XGBoost method is a fine-tuned implementation of gradient boosting method which returns higher accuracy but with shorter process duration and lower storage requirement. The Support Vector Machine is a heavily used classification algorithm which works especially well for Binary classifications by constructing non-linear class boundaries. All of our models are performing very well and all of them have beat one of the industry rules of thumb—70% area under the curve.

Here is the summary performance matrix of the 5 models we applied:

Table 1.1

	Train					**Test**				
	Accuracy	*Sensitivity*	*Specificity*	*F1*	*AUC*	*Accuracy*	*Sensitivity*	*Specificity*	*F1*	*AUC*
GLM	80.4%	94.8%	78.4%	54.9%	92.9%	80.1%	93.4%	78.1%	54.8%	92.7%
Random Forest	83.1%	95.1%	81.3%	58.5%	94.0%	82.8%	94.5%	81.1%	58.6%	94.2%
Neural Network	83.7%	95.4%	82.1%	59.5%	94.2%	82.3%	89.4%	81.3%	56.7%	92.6%
XGBoost	86.9%	96.7%	85.4%	64.9%	96.6%	85.8%	91.3%	85.0%	62.4%	94.4%
SVM	83.5%	97.2%	81.5%	59.7%	95.4%	82.1%	93.3%	80.4%	57.3%	93.0%

Introduction

Most of the companies use telephone marketing to connect with their customers or potential clients directly. From small firms to multinational giants, any business that sells or deals directly, issue quotes, handles enquiries and even take orders on the phone.

There are many reasons companies still use phone calls as the way of marketing its products. Unlike text marketing or email, this technology allows you to talk to the customer directly and gauge their level of interest. Some potential customers still prefer being informed by phone calls rather than emails, mails or automated messages. They prefer brands who have their contact and are readily available to talk with people instead of robots, so the trend of automated messages or mails is often frustrating for the customers. It is also possible for businesses to gauge which customers lean towards which products by having a phone conversation with them and is a better and personalized way of marketing than many other approaches.

Just like many other firms, banking companies have relied on phone calls as an essential form of marketing. Bank customers often maintain long relationships and rely heavily on these institutions to make them aware of the new products that are launched from time to time. After advancement in technology, which allows us to leverage the data and turn it into actionable insights, these Call Centers have now looked like Goldmines of Marketing Data. Many banks have used the technology to optimize their marketing. This study is also based on one such data set.

The data belongs to a Portuguese banking institution and is related to its direct marketing campaign, which includes phone calls as its sole marketing technique. Some clients were required to be contacted more than once to identify if they will opt for the product. We have demographic data of the customers, their financial history, activity, the socioeconomic conditions of the time when the campaign was in place and finally whether they conduct business with the bank.

STATEMENT OF PROBLEM

The problem in contention is *Binary Classification,* where we have to analyze the data that was gathered or rather resulted from the previous marketing campaign (where the mode of communication was phone calls only and each customer was often contacted more than once) and figure out the factors which influenced the conversion of the customers the most. In other words, we can determine whether the contacted customer will opt for the term deposit product or not.

We can then use our results to predict or make recommendations as to which marketing strategy will yield maximum conversion amongst customers and make the best use of our resources.

DATA DESCRIPTION

The data set comprises the following 21 variables:

Input variables

- **bank client data:**
  - **Age**: (age of customer - numeric)
  - **Job**: type of job - (categorical: admins, blue-collar, entrepreneur, housemaid, management, retired, self-employed, services, student, technician, unemployed, unknown)
  - **Marital**: marital status (categorical: divorced, married, single, unknown; note: divorced means divorced or widowed)
  - **Education**: (categorical: basic.4y, basic.6y, basic.9y, high school, illiterate, professional course, university degree, unknown)
  - **Default**: has credit in default? (categorical: no, yes, unknown)
  - **Housing**: has housing loan? (categorical: no, yes, unknown)
  - **Loan**: has personal loan? (categorical: no, yes, unknown)

- **Related with the last contact of the current campaign:**
  - **Contact**: contact communication type (categorical: cellular, telephone)

- **Month**: last contact month of year (categorical: Jan, Feb, Mar, ..., Jul, Aug, Sep, Oct, Nov, Dec)
- **Day of week**: last contact day of the week (categorical: Mon, Tue, Wed, Thu, Fri)
- **Duration**: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

Other attributes:

- **Campaign**: number of contacts performed during this campaign and for this client (numeric, includes last contact)
- **Pdays**: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means this client was not previously contacted)
- **Previous**: number of contacts performed before this campaign and for this client (numeric)
- **Poutcome**: outcome of the previous marketing campaign (categorical: failure, nonexistent, success)

Social and economic context attributes

- **Emp.var.rate**: employment variation rate - quarterly indicator (numeric)
- **Cons.price.idx**: consumer price index - monthly indicator (numeric)
- **Cons.conf.idx**: consumer confidence index - monthly indicator (numeric)
- **Euribor3m**: euribor 3-month rate - daily indicator (numeric)
- **nr.employed**: number of employees - quarterly indicator (numeric)

Output variable (desired target):

- **y** - has the client subscribed a term deposit? (binary: 'yes','no')

EXPLORATORY DATA ANALYSIS

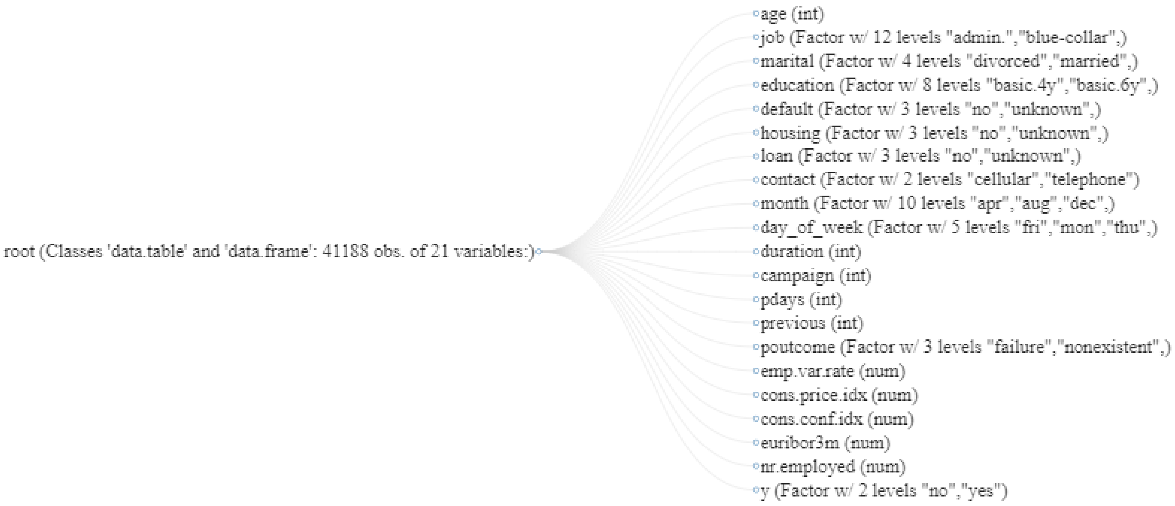
root (Classes 'data.table' and 'data.frame': 41188 obs. of 21 variables:)
age (int)
job (Factor w/ 12 levels "admin.","blue-collar",)
marital (Factor w/ 4 levels "divorced","married",)
education (Factor w/ 8 levels "basic.4y","basic.6y",)
default (Factor w/ 3 levels "no","unknown",)
housing (Factor w/ 3 levels "no","unknown",)
loan (Factor w/ 3 levels "no","unknown",)
contact (Factor w/ 2 levels "cellular","telephone")
month (Factor w/ 10 levels "apr","aug","dec",)
day_of_week (Factor w/ 5 levels "fri","mon","thu",)
duration (int)
campaign (int)
pdays (int)
previous (int)
poutcome (Factor w/ 3 levels "failure","nonexistent",)
emp.var.rate (num)
cons.price.idx (num)
cons.conf.idx (num)
euribor3m (num)
nr.employed (num)
y (Factor w/ 2 levels "no","yes")

Figure 2.1

There are 41,188 observations and 21 columns (in the dataset), out of which 10 features are categorical variables and 11 continuous variables. On a broader scale, we have four categories of variables, namely, bank client data (7 features), Contact data (4 features), Socioeconomic Data with 5 features and finally other variables. In terms of the type of data, we found that we have 10 Numeric (5 continuous & 5 discrete) while 7 Categorical and 5 Binary Indicators. The data set has both temporal and cross-sectional data.

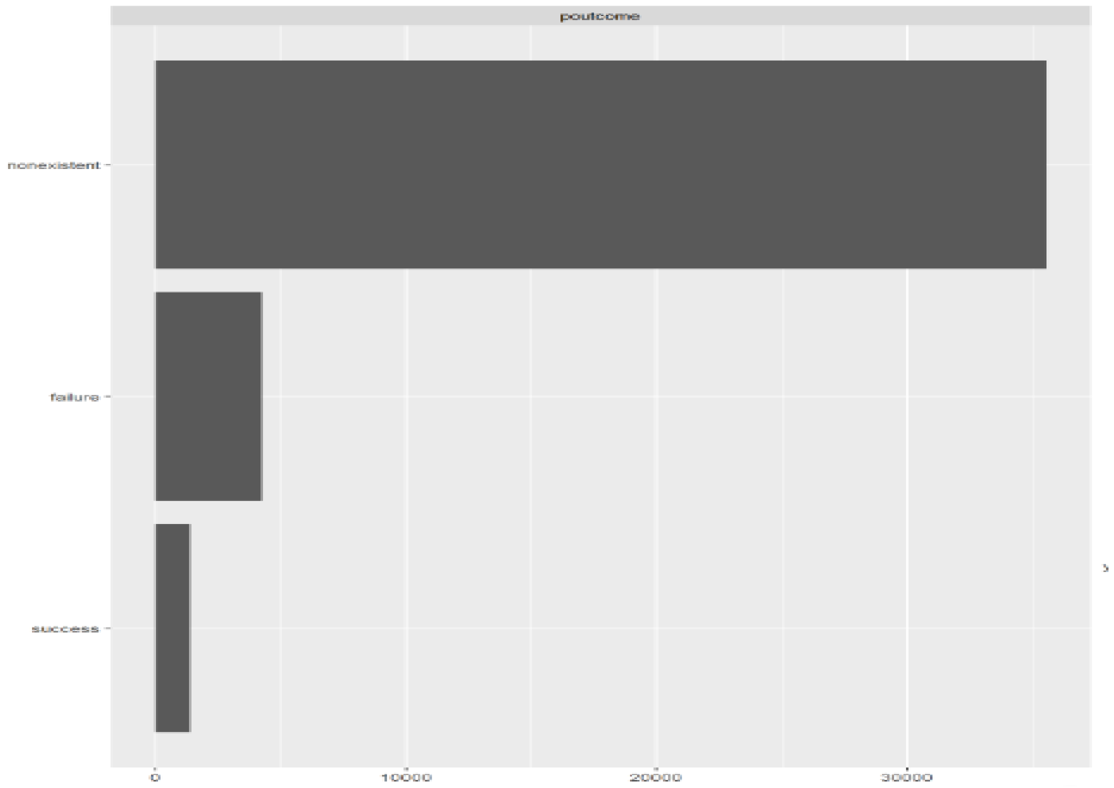

Figure 2.2

As far as the missing data is concerned, the variable pdays is missing around 96.32% observations so it makes little sense to include it in model even though we might lose some valuable information in terms of a feature of customers who opted for the product. Similarly, poutcome has a lot of missing values but would be in contention to be used in the model. The default column also has 20.87% values missing. The rest of the dataset has pretty much no issues in terms of the missing values.

We can see that the product was mostly marketed to people are between the age of 25 and 50, and the majority were contacted less than 3 times in this campaign. The variable 'previous' has a lot of 0's, so most of this campaign was targeted towards new customers or people who were not contacted previously. People with admin and blue-collar jobs among the most approached group while the product was less promoted among retired people and students which reiterates our observation we made by looking at the age column where we found that middle age (25-50) were the most targeted segments. Most of the customers in this list are married which makes sense as they would be highly motivated to choose this product and save for expense that they can incur later like education for their children, house, etc. A lot of these people are also highly educated (university degree) which is in line with most people being approached having blue-collar jobs.

Distribution of variables (for which the customer had converted)

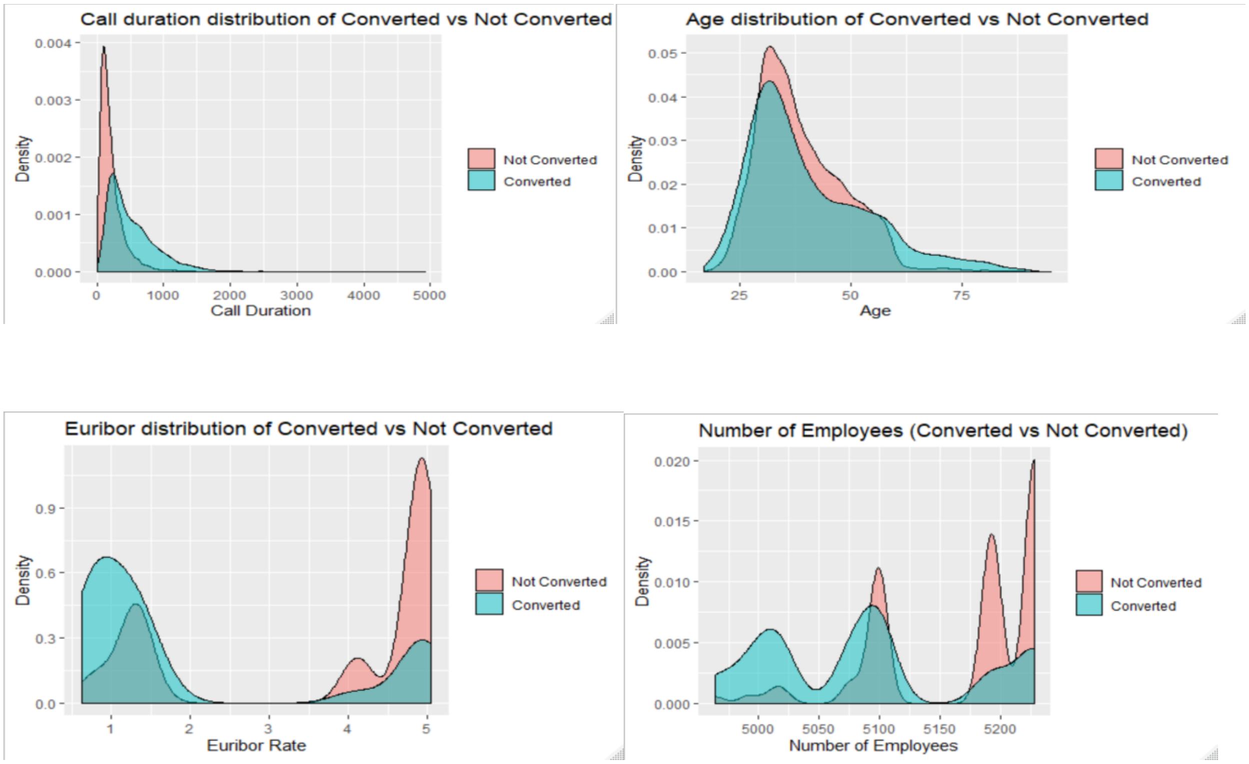

Figure 2.3

When we looked at the distribution of variables for the records in which the customer had converted or took the loan, we found that the age of the customers who had opted for the loan was skewed to the right which indicates the fact that most of the loan was taken by older population. Same as the case with the duration of phone calls (with the clients who opted for the loan), It was observed that longer duration of phone call meant that the customer was most likely to opt for the loan. On the other hand, the two variables, Euribor3m value (Euribor 3-month rate) and the number of employees (for the clients who converted) were skewed to the left which mean the customers are more likely to convert when the number of employees is lower vis-a-vis when the number of employees is higher.

As far as the social and economic factors are concerned, they correspond to different reporting frequency (quarterly, monthly and daily). We found that that employment variation rate for when customers converted had a negative mean, median and even third quartile which makes sense as most people will save when employment is not doing well in the economy. Same is the case with consumer price and conf. index. What is counter-intuitive is that people opted for this product when Euribor too had a lower mean and median as generally, people save when interest rates are high.

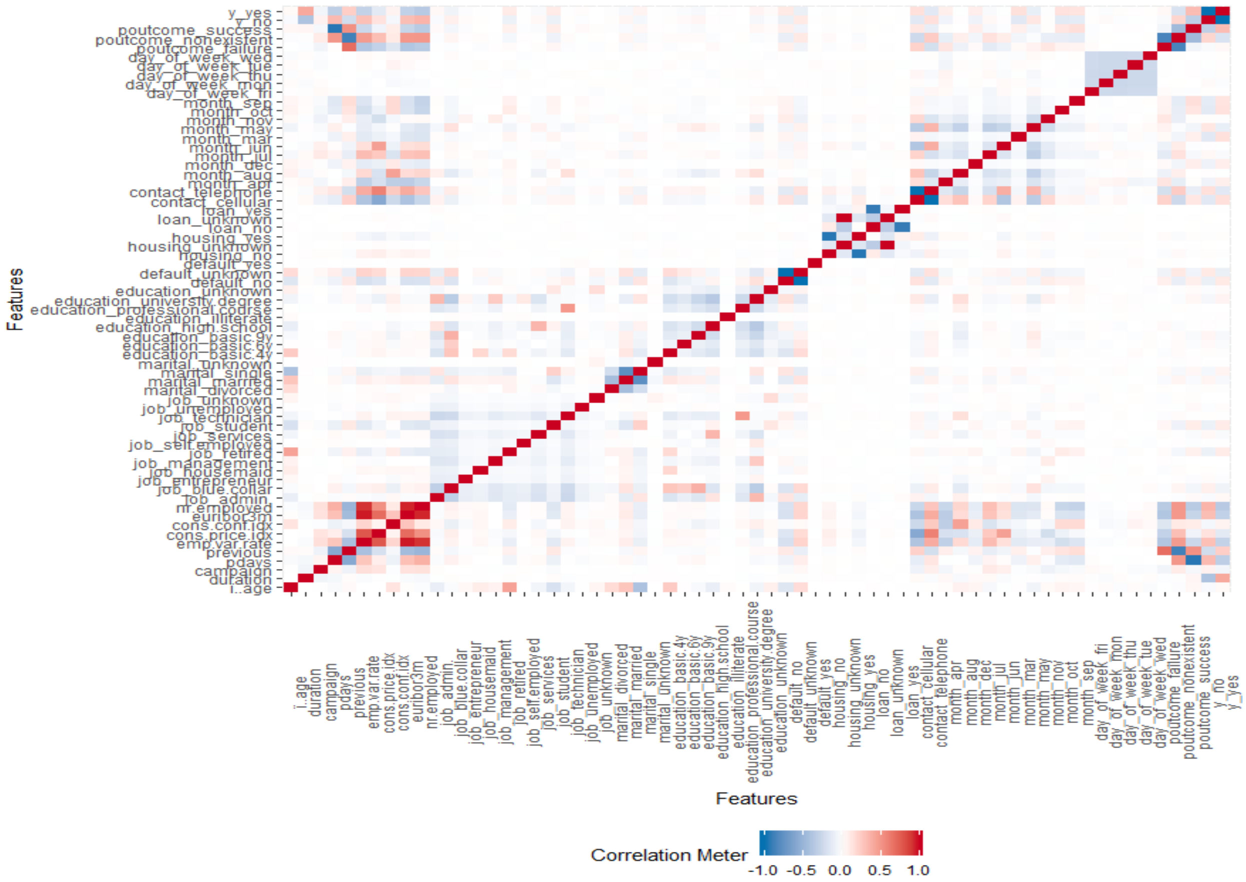

Figure 2.4

There is some high correlation between most of these social and economic factors which was partly due to the fact that they were recorded on different time scales.

We now move to modeling techniques that we employed.

Model Building

We split the original dataset into 2 parts, 70% of the original dataset was set to training data and the rest 30% was set to testing data. On this basis, we have built 5 predictive models including: Generalized Linear Model for logistic regression, Tree based Random Forests model, Neural Network model, XGBoost model and Support Vector Machines model. For each model, our general workflow is: 1. Model fitting and variable selection use training data; 2. Calculate optimal cut-off probability and perform the in-sample and out-of-sample prediction; 3. Model evaluation by calculating Accuracy Rate, Sensitivity and Specificity Score, F1 Score, and AUC; 4.

Goodness-of-fit test to make sure our model is in a good fit. The simulation seed was set to 13500813.

Generalized Linear Model

Model Fitting

We first built a binominal logistic regression model; we used the default logit link function for the model fitting. Then we used the forward BIC method for model selection. According to the BIC score, our final model is : $y \sim nr.employed\ +\ duration\ +\ month\ +\ poutcome\ +\ emp.var.rate\ +\ \ cons.price.idx\ +\ contact\ +\ cons.conf.idx\ +\ campaign$

Optimal Cut-off Probability

We used customized cost function to calculate the cut-off probability for all models. In generalized linear model, we used 10:1 as the asymmetric cost ratio. The cost plot is shown as in figure 3.1:

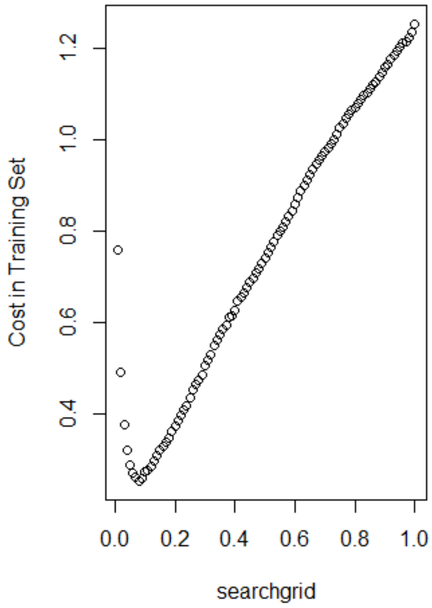

Figure 3.1

We then calculated the optimal cut-off probability is 0.08. Based on the optimal cut-off probability, we performed the in-sample and out-of-sample prediction.

Here is the confusion matrix for training data：

GLM: Training Data	***Predicted***

		Interested	Not Interested
Observed	Interested	*14625*	*4038*
	Not Interested	*139*	*2539*

Table 3.1

The misclassification rate for training data set is: 20.98%

Here is the confusion matrix for testing data:

GLM: Test Data		***Predicted***	
		Interested	Not Interested
Observed	Interested	*6223*	*1743*
	Not Interested	*78*	*1103*

Table 3.2

The misclassification rate for testing data is 19.91%

Model Evaluation

Here is the performance matrix of the generalized linear regression model.

Table 3.3

	Training	Testing
Accuracy	0.8042	0.8009
Sensitivity	0.9481	0.9340
Specificity	0.7836	0.7812
F1 Score	0.5487	0.5478
Area Under the Curve	0.9293	0.9268

From the evaluation matrix, we can see that the generalized linear regression model can well predict the objective, the accuracy rate and area under the curve are very high and can beat the industry rule of thumb.

Random Forests

We then performed the Tree based Random Forests test to predict whether the contacted customer will opt for the term deposit product or not.

Model Fitting

We fitted the model by using the randomForest function, and then we plotted the Out of Bag error, False Positive Rate and False Negative Rate in the following plot:

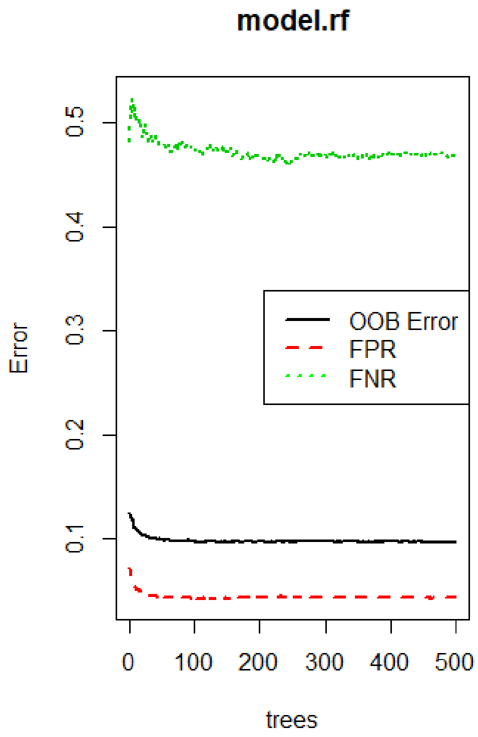

Figure 3.2

With the increase of the number of trees all 3 error measurements are decreasing, and both OOB error and FPR are constantly very low. However, the FNR is pretty high. It can be a little dangerous when we observed a high FNR. Therefore, we need to further tune the model by using the optimal cut-off probability and have a better prediction.

Optimal Cut-off Probability

Again, we used the same customized cost function that we defined in GLM; the optimal cut-off probability is 0.1 for Random Forest model. Here is the cost plot:

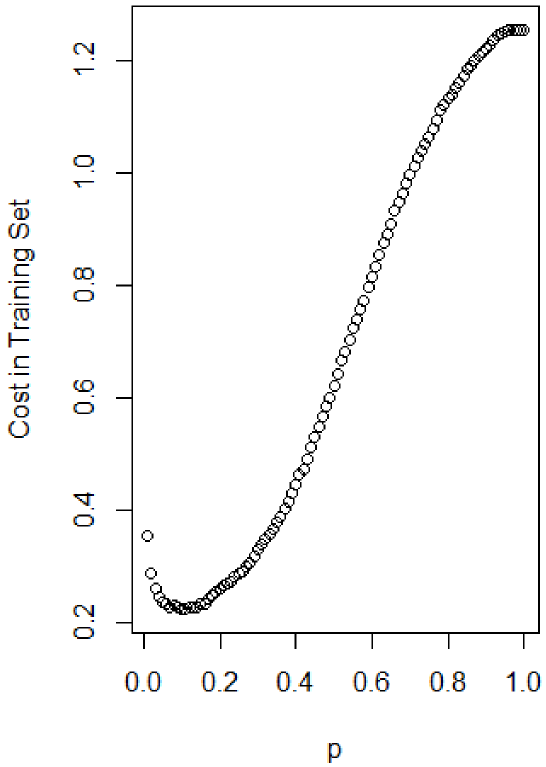

Figure 3.3

Here is the confusion matrix for training data：

RF: Training Data		***Predicted***	
		Interested	Not Interested
Observed	Interested	*15178*	*3485*
	Not Interested	*131*	*2547*

Table 3.4

The misclassification rate for training data is: 16.94%

The FNR is now only 4.89%.

Here is the confusion matrix for testing data:

RF: Test Data	***Predicted***	
	Interested	Not Interested

Observed	Interested	*6457*	*1509*
	Not Interested	*65*	*1116*

Table 3.5

The misclassification rate for training data is: 17.21%

The FNR is now only 5.50%.

Model Evaluation

Here is the performance matrix of the Random Forest model.

Table 3.6

	Training	Testing
Accuracy	0.8305	0.8279
Sensitivity	0.9511	0.9450
Specificity	0.8133	0.8106
F1 Score	0.5848	0.5864
Area Under the Curve	0.9399	0.9420

Here we can see, the Random Forest model also performed very well. The overall performance is even better than GLM model. Therefore, we believe both models are very good in predicting the objective.

Neural Network

Model Fitting

For a neural networks model, we need to first standardize the data, in order to convert the data to the format that neural network models can you. Also, there are many packages for building neural network models, here we are using the nnet function which the number of hidden layers is fixed to 1 by default.

The model we built has 3 neurons, and it is shown as follows, due to the number of input variable is too large, so the plot may look unpleasant:

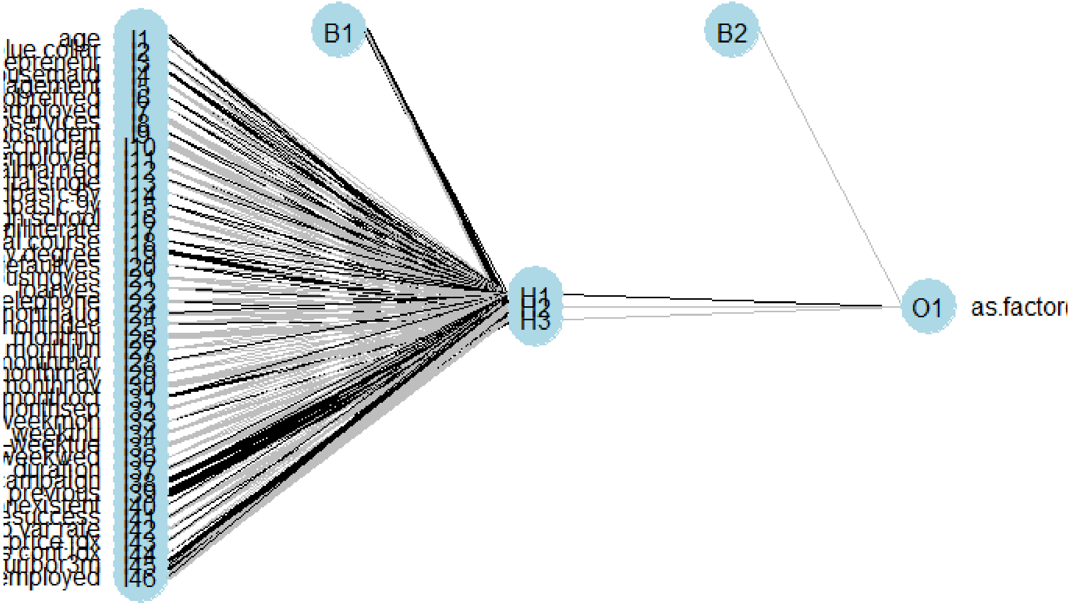

Figure 3.4

Optimal Cut-off Probability

The optimal cut-off probability for neural network model is 0.09. The cost plot is shown below:

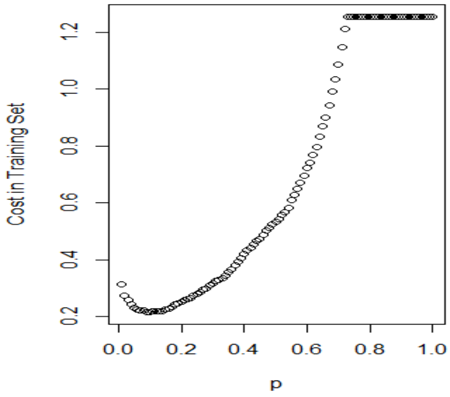

Figure 3.5

The confusion matrix for training data is shown below:

NNet: Training Data		***Predicted***	
		Interested	Not Interested
Observed	Interested	*15314*	*3349*
	Not Interested	*123*	*2555*

Table 3.7

The misclassification rate is: 16.30%

The confusion matrix for testing data is shown below:

NNet: Test Data		***Predicted***	
		Interested	Not Interested
Observed	Interested	*6476*	*1490*
	Not Interested	*125*	*1056*

Table 3.8

The misclassification rate is 17.67%.

Model Evaluation

Here is the performance matrix of the Random Forest model.

Table 3.9

	Training	Testing
Accuracy	0.8370	0.8233
Sensitivity	0.9536	0.8941
Specificity	0.8202	0.8128
F1 Score	0.5949	0.5665
Area Under the Curve	0.9421	0.9264

Here we find the neural net work model returns a similar accuracy to Random Forest model, where the Random Forest model has slightly better performance on testing dataset.

XGBoost

XGBoost is an ensemble learning method which is a popular implementation of gradient boosting. XGBoost has lots of unique features, for this project specifically, XGBoost has some main benefits like: time saving and storage efficiency.

Model Fitting

Like the neural networks model, we need to standardize the dataset at the beginning, then we built the model on the standardized dataset. We then specified the parameters with learning rate (eta) to 0.1 to shorten the processing time, other parameters are using the default value.

The model stopped fitting when the training error is equal to 0.071787 where it cannot be further reduced in the next 10 rounds.

Optimal Cut-off Probability

With the same cost function, the optimal cut-off probability is 0.13 for XGBoost. The cost plot is shown below:

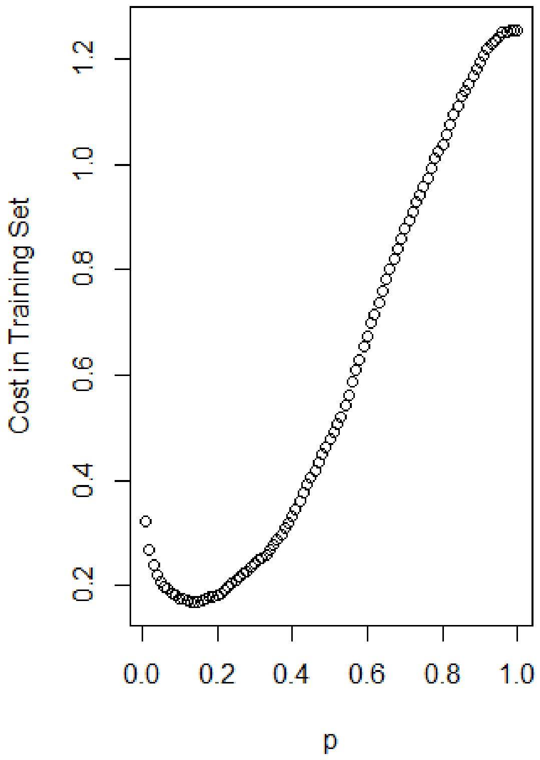

Figure 3.6

Here is the confusion matrix for training data：

XGB: Training Data		***Predicted***	
		Interested	Not Interested
Observed	Interested	*15946*	*2717*
	Not Interested	*88*	*2590*

Table 3.10

The misclassification rate for training data is: 13.14%

Here is the confusion matrix for testing data:

XGB: Test Data		***Predicted***	
		Interested	Not Interested
Observed	Interested	*6772*	*1194*
	Not Interested	*103*	*1078*

Table 3.11

The misclassification rate for testing data is :14.17%

Model Evaluation

Here is the performance matrix of the Random Forest model.

Table 3.12

	Training	Testing
Accuracy	0.8685	0.8582
Sensitivity	0.9671	0.9127
Specificity	0.8544	0.8501
F1 Score	0.6487	0.6243
Area Under the Curve	0.9665	0.9438

From the performance matrix, we can see that XGBoosting so far has the best performance among the first 4 models.

Support Vector Machines

Model Fitting

With SVM, we can specify the asymmetric cost function into the model building function. We specified the class.weights equal to 10 for class “1” and 1 for class “0”. Cost and gamma arguments were using the default values where gamma= 1/ (data dimension), cost=1.

Prediction

Since the optimal cut-off probability has been calculated and applied in the model building phase. Here we directly performed the in-sample and out-of-sample prediction.

Here is the confusion matrix for training/testing data：

SVM: Training Data		***Predicted***	
		Interested	Not Interested
Observed	Interested	*15218*	*3445*
	Not Interested	*74*	*2604*

Table 3.13

SVM: Test Data		***Predicted***	
		Interested	Not Interested
Observed	Interested	*6404*	*1562*
	Not Interested	*79*	*1102*

Table 3.14

The misclassification rate for training data is: 16.48%

The misclassification rate for testing data is: 17.94%

Model Evaluation

Here is the performance matrix of the Random Forest model.

Table 3.15

	Training	Testing
Accuracy	0.8351	0.8206
Sensitivity	0.9724	0.9331
Specificity	0.8154	0.8039
F1 Score	0.5968	0.5732
Area Under the Curve	0.9541	0.9296

The overall accuracy of SVM model is slightly

Gain Chart and Lift Chart

Because of the length requirement of the report, we placed the actual charts in appendix 2. Here is a summary table of the output values from gain chart and lift chart. We can see the overall performance of each model is very close.

Table

Percentiles	GLM	Neural Network	XGBoost	SVM	GLM	Neural Network	XGBoost	SVM
10	49.03%	50.38%	53.01%	48.09%	4.91	5.04	5.3	4.81

20	78.75%	78.66%	82.98%	79.34%	3.94	3.93	4.15	3.97
30	92.72%	91.53%	95.43%	93.82%	3.09	3.05	3.18	3.13
40	97.88%	97.54%	98.90%	98.65%	2.45	2.44	2.47	2.47
50	99.41%	99.15%	99.75%	99.66%	1.99	1.98	2	1.99
60	99.41%	99.83%	99.92%	99.83%	1.66	1.66	1.67	1.66
70	99.58%	99.92%	100.00%	100.00%	1.42	1.43	1.43	1.43
80	99.75%	99.92%	100.00%	100.00%	1.25	1.25	1.25	1.25
90	99.83%	99.92%	100.00%	100.00%	1.11	1.11	1.11	1.11
100	100.00%	100.00%	100.00%	100.00%	1.00	1.00	1.00	1.00

Table 3.16

Goodness of Fit Test

We also applied Hosmer-Lemeshow test to evaluate our logistic models, which is one of the most popular goodness of fit test due to its clear calculation and interpretation. We need to assign a number of groups (g) in the function firstly. The dataset would be sorted according to the value of estimated success probability, and then split into g groups with an equal size. A well-fitting model should show negligible difference between the observed data and the fitted model. In the Hosmer-Lemeshow test, the p-value of a model should be greater than 0.05 to pass the test. Using the “hoslem.test” function in the R package “ResourceSelection”, we evaluated the goodness of fit for each model as following:

Table 3.17

Table. P-values summary of Hosmer-Lemeshow test (g = 10)		
Method	Training Set	Testing Set
GLM	< 2.2e-16	< 2.2e-16
Tree Model	0.0003049	**0.4186**
Neural Network	**0.5604**	2.2e-16
XGBoost	< 2.2e-16	**0.321**
SVM	< 2.2e-16	**0.4524**

The P-values summary above shows that only one model using training dataset (Neural Network) and three models (Tree Model, XGBoost, and SVM) using testing dataset passed the test. All other models failed that is obviously not we expected.

We tried to find a satisfying explanation to these unsatisfying results. Like other chi-square tests, the evaluating power of Hosmer-Lemeshow test is strongly affected by the sample size. Dahiya and Gurland (1973) illustrated how model performance is affected by an increasing sample size. According to the Yu, et al. (2017), this dataset usually loses the power when the sample size is larger than 25,000. Our training set and testing set has 28,831 and 12,357 observations, exclusively. This might be an explanation for our testing results that most of the tests for training set were failed, and the success rate for testing set is higher.

Reference

Dahiya RC, Gurland J (1973). How many classes in the Pearson chi-square test? Journal of the American Statistical Association, 68(343):707–712.

W. Yu, W. Xu and L. Zhu (2017). A modified Hosmer–Lemeshow test for large data sets. Communications in Statistics - Theory and Methods 46(23):11813-11825

Appendix

1. Variable importance bar chart—illustrate the importance of each variable in the model.

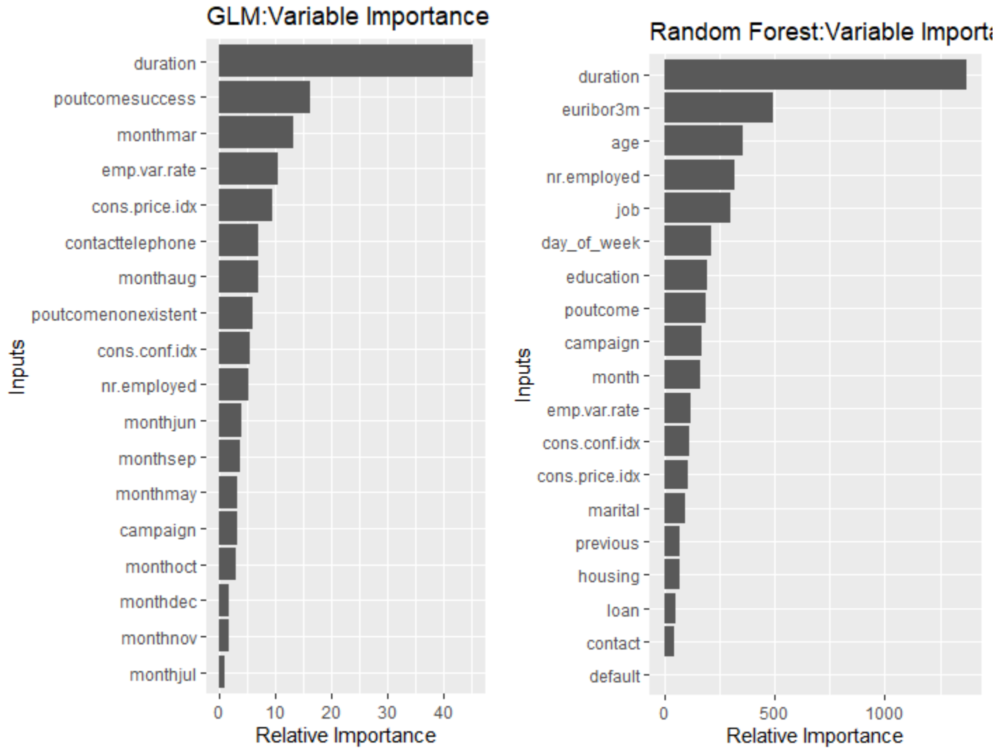

figure a1.1-1.2

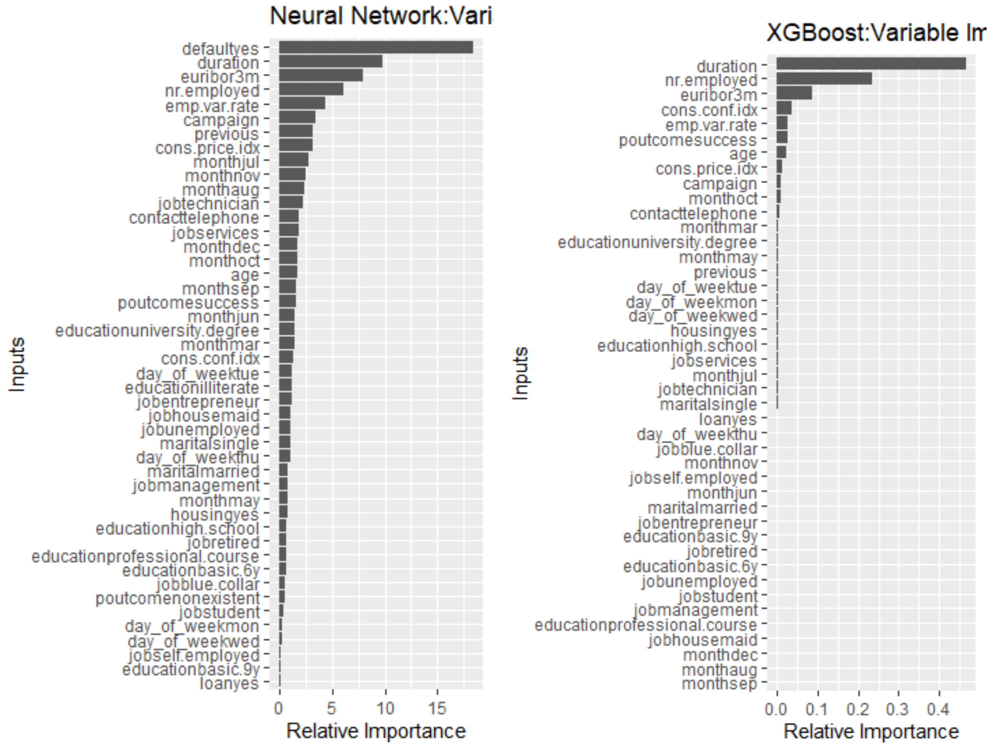

figure a1.3-1.4

2. Gain and Lift Plot

These 2 plots illustrate the vertical comparison between models.

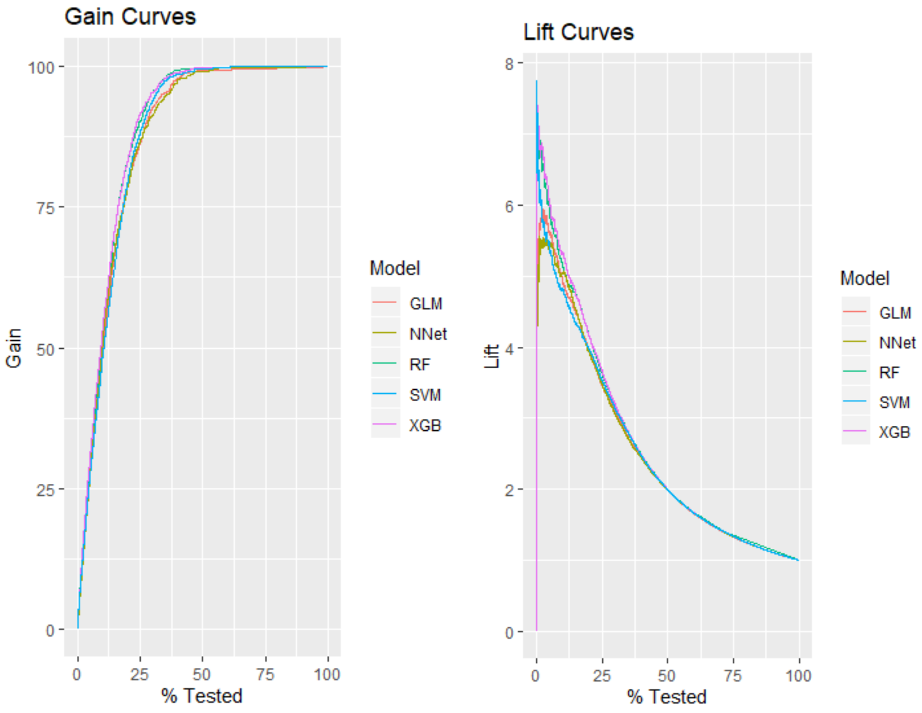

Figure a2.1-2.2