**Data Mining on Banking Tele-Marketing Campaign**

**Project Team 10**

**Group Members:** Anuj Agrawal, Genghua Li, Shunya Miyatake

**INDEX**

|  |  |  |
| --- | --- | --- |
| Section No. | Title | Page No. |
|  | Executive Summary | 3 |
| 1 | Introduction | 4 |
|  |  |  |
| 2 | Data Modification | 4 |
|  |  |  |
| 3 | Data Exploration | 4 |
|  |  |  |
| 4 | Models |  |
| 4.1 | Decision Tree | 7 |
| 4.2 | Random Forest | 9 |
| 4.3 | Logistic Regression | 12 |
| 4.4 | Discriminant Analysis | 14 |
| 4.5 | Naïve Bayes | 18 |
|  |  |  |
| 5 | Model Comparison | 19 |
| 6 | Conclusion |  |
|  |  |  |

**Executive Summary**

A confidential Portuguese bank conducted a marketing campaign by phone to get people to subscribe to their term deposit. They contacted **45211** people; about **12%** of these people subscribed the term deposit. The bank determined that the campaign was not efficient in getting people to respond, considering the amount of work and labor they input. Our group has a solution for the bank to improve efficiency of tele-campaign in the future. Based on the dataset from the campaign, we built a few predictive models on variables such as duration of the call, balance of loans and selected the best predictive model to help the bank to identify potential customers. At the early stage of our model production process, we employed oversampling technique to train models that can better predict responders. By using this technique, the best model from random forest improved its precision by **30%** and the model from discriminant analysis improved precision by about **10%**.

Since the response variable we are interested in is binary (whether a person will subscribe or not), we based our evaluations of models upon confusion matrices. In particular, we focus on precision which tells us the proportion of the responders in the testing dataset that was correctly identified; the reason is that the benefit of correctly identifying responders is so much greater than that of correctly identify non-responders. Below is a table that compares the accuracies and precisions of different models.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy | Precision |
| Decision Tree | **89.89 %** | **40.25%** |
| Random Forest | **88.70%** | **65.70%** |
| Logistic Regression | **89.71%** | **50.63%** |
| Discriminant Analysis | **81.9%** | **61.58%** |
| Naïve Bayes | **85.64%** | **40.13%** |

The random forest model correctly identified **65.7%** of the responders; the implication is, if the bank was to use this model to conduct similar campaign in the future, they would be more than 50% more efficient than randomly calling people. We believed the bank will be satisfied with our solution.

Given more time and computation power, we would also like to test the stability of our models; we sensed that some models may be more stable than others and we believe the bank manager can make more informed decision if a comprehensive comparison of stabilities of models can be provided. We would carry it out by iteratively randomly partition our dataset and trace the trend of precisions for each model.

1. **Introduction**

In this project, we seek to help a confidential Portuguese bank with its tele-marketing campaign to increase its asset by getting more people to subscribe its term deposits. The bank has historical data for its existing customers, with their economic and social characteristics captured in various variables. Using this dataset, we are to help the bank to improve confidence in identifying the customers who will be subscribing the term deposit from the bank, thereby, increasing the bank’s asset and fund for investment. The relevant response variable is the binary variable ‘y’–whether or not a person will subscribe term deposit from the bank.

**The data set has 45211 observations and 17 variables including the response variable ‘y’**.

The other variables are: age, job, marital, education, default, balance, housing, loan, contact, day, month, duration, campaign, pdays, previous, poutcome. The link for the dataset and its full description can be found at the below URL:

[**http://archive.ics.uci.edu/ml/datasets/Bank+Marketing**](http://archive.ics.uci.edu/ml/datasets/Bank+Marketing)

1. **Data Modification**

The dataset does not have any missing values. Also the variables already present in the dataset do not require any transformations and are used directly in various models. We decided that the variables ‘day’ and ‘month’ are ignorable variables in our training and testing of the model because the variable ‘pdays’ already captures the information of the number of days that have passed since the last contact with the customer. So, the variables ‘day’ and ‘month’ have not been included as predictor variables in any of the models.

**We first partitioned the dataset randomly into training dataset and testing dataset with the ratio of 60:40 in R. [1]**

**The proportion of ‘yes’ response is about 12% which is pretty rare.** This could result in over-fitting of our model to records with ‘no’ response. We found this to be highly likely as we discovered that the proportion of ‘yes’ response we detected were very low. In response, **we oversampled the records with ‘yes’ response from our training dataset**. It will also be discussed later in the report. [2]

1. **Data Exploration**

Using the summary function in R, we looked at the basic statistics for various variables.

> summary(bank.full)

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 no :39922

other : 1840 yes: 5289

success: 1511

unknown:36959

The above output gives the mean values and range of values for all the continuous variables. For categorical observations it gives the number of observations under each category.

Based on some knowledge of banking domain we first identified the variables which might have influence on the response variable – y. The Exploratory data analysis is summarized below:

1. **Poutcome vs Y:** Poutcome gave the result of any such previous marketing campaign with the customer. A customer who responded favorably to an earlier campaign was expected to do so for the current one:

> table(poutcome, y)

y

poutcome no yes

failure 4283 618

other 1533 307

**success 533 978**

unknown 33573 3386

> prop.table(table(poutcome,y),1)

y

poutcome no yes

failure 0.87390329 0.12609671

other 0.83315217 0.16684783

**success 0.35274653 0.64725347**

unknown 0.90838497 0.09161503

The above tables confirmed that the proportion of customers responding ‘yes’ was much more (64 % for Success rate) than other categories of poutcome. This indicated that this variable could be very important in predicting y.

1. **Job**: The job variable had 12 categories. We wanted to see whether some category of people working a specific job were likely to respond ‘yes’ to bank term deposit campaign.

> prop.table(table(job,y),1)

y

job no yes

admin. 0.87797331 0.12202669

blue-collar 0.92725031 0.07274969

entrepreneur 0.91728312 0.08271688

housemaid 0.91209677 0.08790323

management 0.86244449 0.13755551

**retired 0.77208481 0.22791519**

self-employed 0.88157061 0.11842939

services 0.91116996 0.08883004

**student 0.71321962 0.28678038**

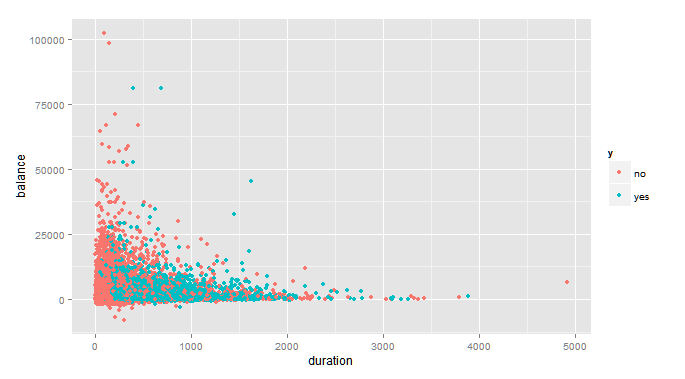
technician 0.88943004 0.11056996

unemployed 0.84497314 0.15502686

unknown 0.88194444 0.11805556

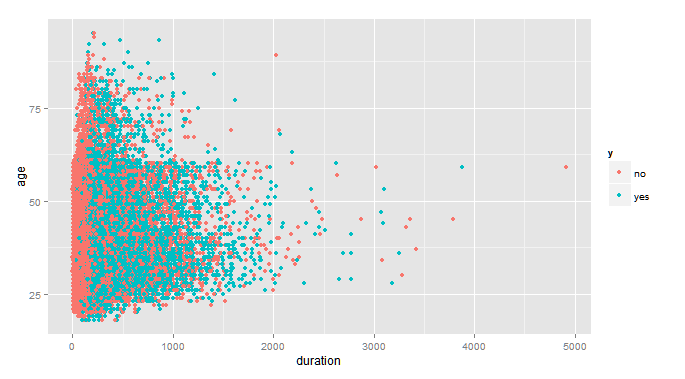
The above table gave an interesting insight that students with a response rate of 28.6 % and retired people with a response rate of 22.7 % were more interested in subscribing bank term deposit than other categories.

1. **Duration:** The variable Duration which indicates the time duration of phone call, is an important variable since if the customer is interested, the time duration will be more; and if the customer is not interested, he or she will hang up quickly. We made two scatter plots of relationship of two continuous variables color coded by response.



More Concentration of ‘yes’ responders for longer duration

Less Concentration of ‘no’ responders for shorter duration



From the both of the above plots we can clearly see the concentration of the cluster of red dots (‘no’ responders) that was separated from the blue dots (‘yes’ responders). The yes responders tend to have longer duration.

1. **Models**

**4.1 Decision Tree**

**Decision Tree Model**

We used the **rpart** package in R to make a classification tree. This tree was then tested on the test dataset. **The criteria R used to automatically stop the expansion of tree is cp;**

**We used the cutoff of cp = 0.001, meaning a split must decrease the overall lack of fit by a factor of 0.001 before being attempted.** We **selecting a tree size that minimizes the mis-classification error, labeled ‘xerror’ column printed by printcp( ). After a certain tree size, the xerror starts to increase.** We then use the prune function to prune back the tree using cp value corresponding to that xerror**.** Another important parameter was the **maxdepth,** this indicates the maximum depth to grow the tree up to. We have used the **gini method** for splits using **parms parameter**.

All the variables except the day and month have been included as predictors in this decision tree model. The formula passed as parameter to rpart function is:

> myformula

y ~ age + job + marital + education + default + balance + housing +

loan + contact + duration + campaign + pdays + previous +

poutcome

We tried several parameters for the cp and max depth. A max depth = 5 lead to too much granularity and too few observations in many leaf nodes and over fitting was observed. Finally the tree that gave us the best results was this:

rp1 <- rpart(myformula, data = train.oversampled, method = "class", control = rpart.control(maxdepth = 4, cp=0.0001), parms = list(split = "gini"))

The parameter **method = “class”** indicates that we are building a classification tree where the response variable is binary. Other parameters have been explained above.

**The Cp table for the above classification tree model rp1:**

> printcp(rp1)

Classification tree:

rpart(formula = myformula, data = train.oversampled, method = "class",

parms = list(split = "gini"), control = rpart.control(maxdepth = 4,

cp = 1e-04))

Variables actually used in tree construction:

[1] contact duration housing poutcome

Root node error: 3160/12746 = 0.24792

n= 12746

CP nsplit rel error xerror xstd

1 0.15537975 0 1.00000 1.00000 0.015427

2 0.09683544 1 0.84462 0.84652 0.014549

3 0.03117089 2 0.74778 0.74937 0.013895

4 0.00348101 4 0.68544 0.69873 0.013521

5 0.00126582 5 0.68196 **0.69525** 0.013494

6 0.00094937 7 0.67943 0.69715 0.013509

7 0.00010000 8 0.67848 0.69304 0.013477

As we can see from the above output: the variables actually used in tree construction are: **duration, contact, job and poutcome**. In the cptable, the xerror is the least for the 5th row and again increase at the 6th row. **This means that the 5th iteration corresponds to the tree size that minimizes the classification error. This will be the point where the tree has to be pruned using the below code:**

rp1p <- prune(rp1, cp =0.00126582)

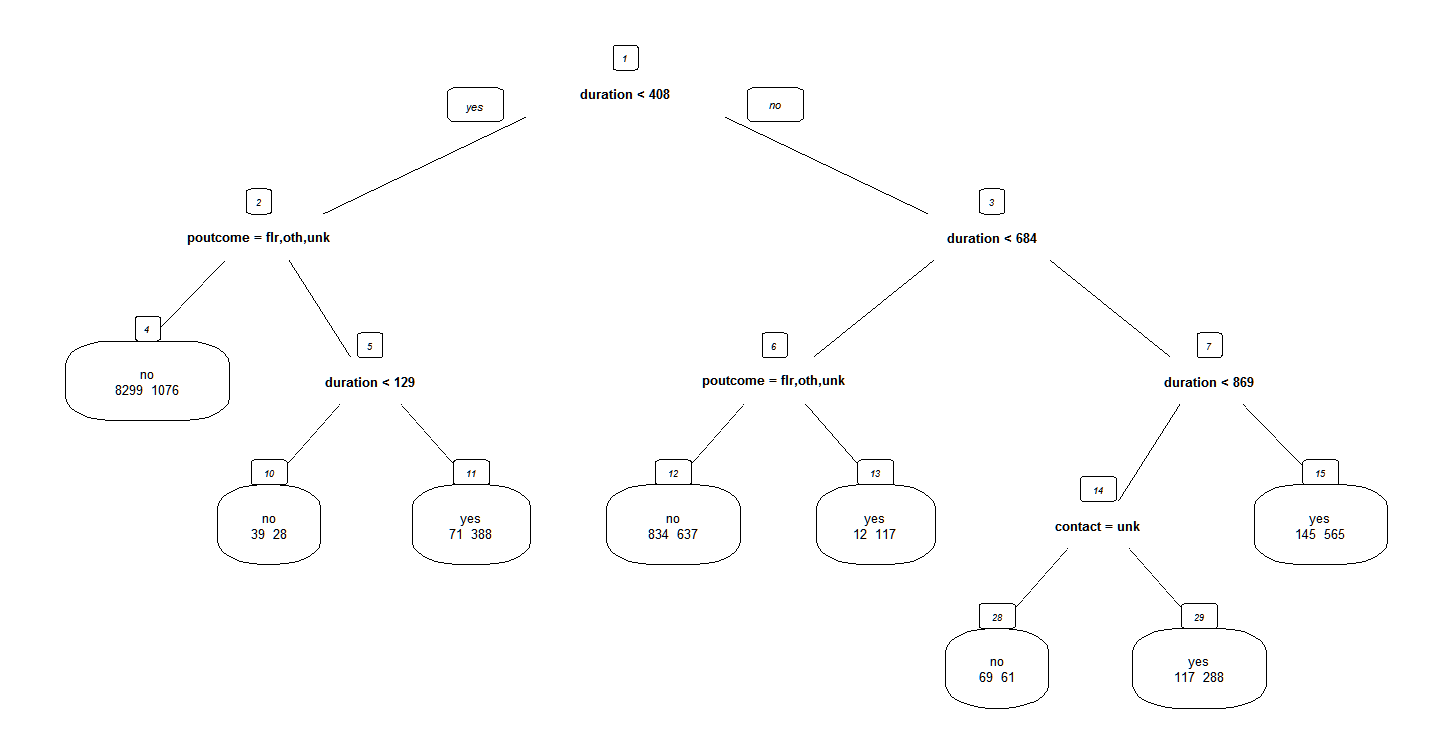
This completes the fitting the decision tree model on the train dataset. Next, we will visualize decision tree and check its accuracy on the test dataset.

**Decision Tree Model Testing**

Finally to get a better visualization of the tree we use the **rpart.plot** package, and use its function prp:

library(rpart.plot)

prp(rp1p, type =0, extra = 1, nn = TRUE)



**Shown above is the final pruned tree obtained by the model.** The yes / no written on the leaf nodes indicates the nodes classification based on majority. By default, the left split is yes for the condition tested (indicated in bold) and the right split is no. We can see the variable duration and poutcome play a major role. The **leaf node 11** (Classification Rule: duration < 408+ poutcome = success + duration > 129) – classified as yes – gives us a very pure node with 348/407 classfied as yes. Similarly the **leaf node 15** (Classification Rule: duration >869) is also very pure. The **leaf node 4** (Classification Rule: duration < 408 + poutcome = failure, other, unknown) is classified as no, it is also very pure. These rules can be used by the bank to classify ‘bad’ customers, ones which are most likely to not to respond to marketing campaign.

**To check the accuracy and precision of the above classification tree, we use the predict function in r** and use a classification confusion matrix.

> dtp1 <- predict(rp1p, newdata = banktest, type = "class")

> table(banktest$y, dtp1)

dtp1

no yes

no 15401 555

yes 1272 857

> prop.table(table(banktest$y, dtp1),1)

dtp1

no yes

no 0.96521685 0.03478315

yes 0.59746360 0.40253640

**Model Accuracy = (15401 + 857) / (15401 + 857+ 555 + 1272) = 89.89 %**

**We get a 89.89% accuracy in the model**, but this is because the response variable has a

Dominant number of ‘no’ responses.

**Model Precision = 40.25% - this means the model is able to successfully predict 40.25% of the ‘yes’ responders.**

**4.2 Random Forest**

**Random Forest Model:**

Since our dataset has many predictor variable with some having many levels or categories, a random forest is a good choice, as it selects the most important variables at each split based on majority voting and thus gives us a more robust and more accurate model.

We **used ‘randomForest’ package in R** for this. Below is the Rcode to fit this model:

> rf1 <- randomForest (myformula, data = train.oversampled, importance =TRUE, do.trace=TRUE, mtry = 4)

The parameter importance = TRUE indicates that the variable importance needs to be assessed for analysis later. **do.trace parameter** gives a more verbose R output for the number of trees built. By default it runs **500 different classification trees.** This can be configured using parameter ntree, but is not required in this case. The parameter **mtry, indicates the number of variables that the algorithm will try at each split**, by default sqroot of the number of predictor variables, increasing mtry will also increase the model accuracy but at the cost of computation speed. **We obtained the best model accuracy by keeping mtry =4.**

**Here is the output of the rf1 random forest:**

> rf1

Call:

randomForest(formula = myformula, data = train.oversampled, importance = TRUE, do.trace = TRUE, mtry = 4)

Type of random forest: classification

**Number of trees: 500**

**No. of variables tried at each split: 4**

**OOB estimate of error rate: 15.32%**

Confusion matrix:

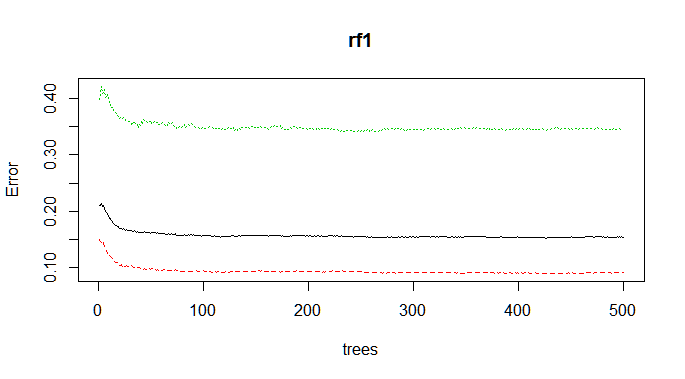
no yes class.error

no 8729 857 0.08940121

yes 1096 2064 **0.34683544**

**The classification error rate for ‘yes’ response in the train dataset is about 34.6 %.**

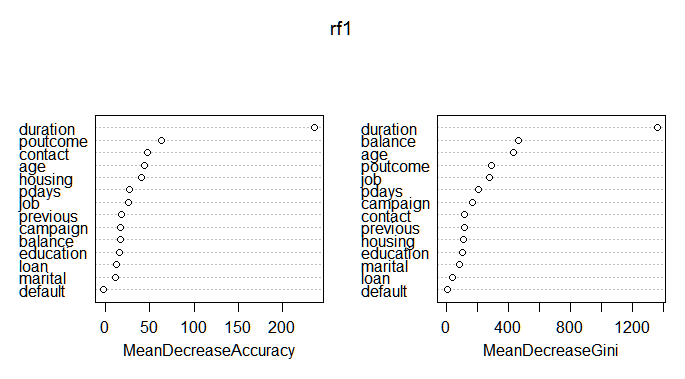
> plot(rf1)



The above plot shows how the OBB (Out of Bag) Classification error rate goes down as the number of tree is increased. Here we observe that for our case the classification error rate falls sharply from **zero to about 100 trees** and stays constant after 100 trees. A very important point to note is that for a random forest algorithm to give good results, the algorithm should be run for sufficient number of trees.

> varImpPlot(rf1)

**Variable Importance PLOT:**



The above chart indicates a **comparison of variables for their importance**. The plot on the right indicates the variables ranked according to the **mean decrease in gini coefficient if the split happens using that variable**. Not surprisingly, the **variable duration is the most important variable**. The other important variables are balance, age, poutcome and job.

This completes the fitting the random forest model on the train dataset. Next, we will check its accuracy on the train dataset.

**Random Forest Model Testing**

To check the accuracy and precision of the above classification tree, we use the predict function in r and use a classification confusion matrix.

> p1 <- predict(rf1, newdata = banktest)

> table(banktest$y, rfp1)

rfp1

no yes

no 14650 1306

yes 730 1399

> prop.table(table(banktest$y, rfp1),1)

rfp1

no yes

no 0.91814991 0.08185009

yes 0.34288398 0.65711602

**Model Accuracy = (14650 + 1399) / (14650 + 1399 + 1306 + 730) = 88.70 %**

**We get a 88.70% accuracy in the mode**l, but this is because the response variable has a

Dominant number of ‘no’ responses.

**From the proportion table above, Model Precision = 65.71% - This is an improvement of**

**Over 25% from the decision tree model. Also unlike decision tree model, we don’t**

**need to tweak the model parameters and choose the best one that fits our data. This is**

**the advantage of the random forest model since it automatically chooses the best**

**variables based on majority votes of trees with the trees of the forest.**

**4.3 Logistic Regression**

**Logistic Regression Model:**

Since the **response variable – y in the dataset is a binary variable**. We can apply a logistic regression model. Based on the variable importance plot of the random forest we got to know the most important variables for predicting the response variable. Out of the 14 predictor variables, we have selected the **7 most important variables – duration, poutcome, balance, age, contact, housing, job** and built a logistic regression model of those variables in R. The **oversampled data used** in the first decision tree model and the random forest model was again used here since the results on the original non-oversampled train data were not good.

**Building the logistic regression model in R using the glm function:**

> logistic2 <- glm(formula2, data = train.oversampled, family = "binomial")

> summary(logistic2)

Call:

glm(formula = formula2, family = "binomial", data = train.oversampled)

Deviance Residuals:

Min 1Q Median 3Q Max

-5.1827 -0.5584 -0.3457 -0.1277 2.9781

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.585e+00 1.562e-01 -10.147 < 2e-16 \*\*\*

duration 4.727e-03 1.084e-04 43.608 < 2e-16 \*\*\*

poutcomeother 2.201e-01 1.287e-01 1.711 0.087163 .

poutcomesuccess 2.413e+00 1.264e-01 19.092 < 2e-16 \*\*\*

poutcomeunknown -4.229e-01 7.909e-02 -5.347 8.94e-08 \*\*\*

balance 1.771e-05 7.044e-06 2.513 0.011961 \*

age -5.459e-03 2.835e-03 -1.925 0.054175 .

contacttelephone 3.677e-02 1.055e-01 0.349 0.727343

contactunknown -1.272e+00 8.578e-02 -14.826 < 2e-16 \*\*\*

housingyes -8.642e-01 5.837e-02 -14.805 < 2e-16 \*\*\*

jobblue-collar -5.417e-01 1.026e-01 -5.282 1.28e-07 \*\*\*

jobentrepreneur -5.018e-01 1.782e-01 -2.816 0.004856 \*\*

jobhousemaid -6.117e-01 1.918e-01 -3.190 0.001423 \*\*

jobmanagement -4.948e-02 9.347e-02 -0.529 0.596535

jobretired 3.566e-01 1.402e-01 2.543 0.010979 \*

jobself-employed -4.079e-01 1.572e-01 -2.594 0.009486 \*\*

jobservices -5.149e-01 1.222e-01 -4.215 2.50e-05 \*\*\*

jobstudent 5.636e-01 1.620e-01 3.479 0.000504 \*\*\*

jobtechnician -2.345e-01 9.935e-02 -2.361 0.018237 \*

jobunemployed 2.230e-02 1.613e-01 0.138 0.890084

jobunknown -6.390e-01 3.577e-01 -1.786 0.074024 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 14276.5 on 12745 degrees of freedom

Residual deviance: 9360.6 on 12725 degrees of freedom

AIC: 9402.6

Number of Fisher Scoring iterations: 5

The model output above shows that most of the predictor variables are significant, indicated by their **P-values which are < 0.05 (Level of Significance)**. The insignificant (0.05 significance level) predictor variables are poutcomeother, age, contacttelephone, jobmanagement, jobunempoyed and job unknown.

**Interpretation of Model Coefficients**:

**Duration- Coefficient = 4.72 x 10^-3 = 0.00472**

The interpretation of the coefficient corresponding to the variable Duration is that a **1 increase in duration leads to an increase of 0.00472in Log Odds of ‘event yes’** in the Response variable. **But, this is interpretation is not very intuitive.**

**Let us interpret this in terms of odds ratio:**

**A 100 secs increase in duration will lead to e^0.00472\*100 = e^0.472 = 1.59 increase in the odds of event ‘yes’ in the Response Variable.** Since Odds = p / 1-p , so p = odds / 1+odds,

So, p = 1.59 / 1+1.59 = 0.613. **That , is p = 0.613**

**So, finally we can say that a 100 seconds increase in the duration will lead to a 0.613 increase in the probability of the event ‘yes’ happening, that is the customer subscribing for the term deposit.**

The model coefficients corresponding to other predictor variables can be interpreted in a similar manner.

The **Null Deviance = 14276.5 on 12745 degrees of freedom**. **Residual Deviance = 9360.6 on 12725 degrees of freedom**. The Residual Deviance is calculated after the predictor variables have been accounted for, while the Null Deviance is calculated without any predictor variables. There is a significant drop in the Residual Deviance, which indicates that the model performance is good. These **Deviances are actually Chi-Sq values and we can calculate the corresponding Chi-Sq P-value** to check if the model is significant:

> pchisq(14276.5-9360.6, 12745-12725, lower.tail = FALSE)

[1] 0

The P-value for the Chi-Sq test = 0 < 0.05 (Alpha – Level of Significance), this indicates that there is a statistically significant difference between the Model without Predictors and the Model with Predictors. In other words, the Model with these Predictor variable is Statistically Significant better than the null model.

**Logistic Regression Model Testing**

To check the accuracy and precision of the above classification tree, we used the predict function in r and used a classification confusion matrix.

> predict.test <- predict(logistic2, newdata = banktest, type = "response")

Since the output of the Logistic Regression Model is probability values, we used 2 cutoff values > 0.5 and > 0.6 to consider a response as ‘yes’.

**Cutoff = 0.5**

> predict.test <- predict(logistic2, newdata = banktest, type = "response")

>

> predict.cutoff1 <- ifelse(predict.test > 0.5 , 'yes', 'no')

> table(banktest$y, predict.cutoff1)

predict.cutoff1

no yes

no 15058 898

yes 1051 1078

> prop.table(table(banktest$y, predict.cutoff1),1)

predict.cutoff1

no yes

no 0.94372023 0.05627977

yes 0.49365899 0.50634101

>

**Cutoff = 0.6**

> predict.cutoff2 <- ifelse(predict.test > 0.6 , 'yes', 'no')

> table(banktest$y, predict.cutoff2)

predict.cutoff2

no yes

no 15300 656

yes 1221 908

> prop.table(table(banktest$y, predict.cutoff2),1)

predict.cutoff2

no yes

no 0.95888694 0.04111306

yes 0.57350869 0.42649131

**The Cutoff = 0.5 is giving us a better Precision. Since, in this study we are more interested in correctly identifying the ‘yes’ response we will go with Cutoff = 0.5**

**Model Accuracy = (1078+ 15300) / (1078+ 15300+ 656 + 1221) = 89.71%**

**We get a 89.71 % accuracy with this mode**l. **Model Precision = 50.63. This is less that the Precision of the Decision Tree and the Random Forest Model.**

**4.4 Discriminant Analysis**

With discriminant analysis, I will use scorings of discriminant function to classify records. The

performance of the classification depends on the followings: Mulivariate Normality of variables for both classes (‘yes’ / ‘no’), the absence of outliers and similarity of correlation matrices across classes.

With some experimenting with different variables to fit my model, I finally selected three

variables into my model: balance, duration and previous. I first explored the multivariate

normality condition of the vector of these three variables. If the vector of the variables is

multi-variate normal, the scatter plot of U (a function of Mahalanobis distance) and V

(a beta distribution statistic dependent on sample size and number of variables) should be

linear. The U vs V plot for class of y=no is in red and that for class of y=yes is in blue.

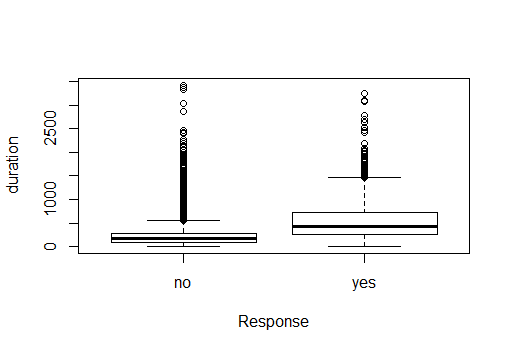
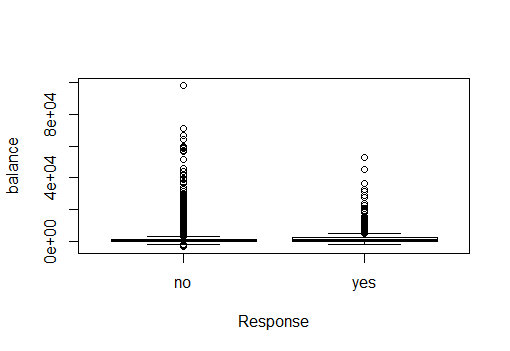
****

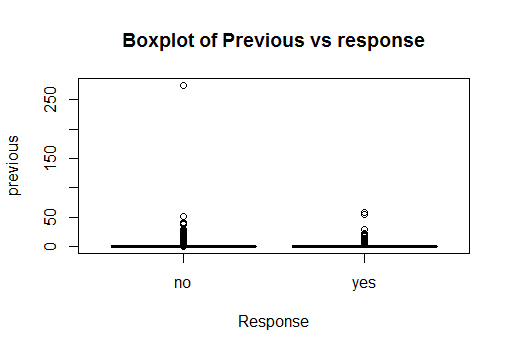
****

We can see the vector of variables is not satisfyingly multivariate normal; but we are not very discouraged because discriminant analysis is very robust to deviation of normality and the vector of variables do not depart from normality terribly. Comparing to the deviation of normality of

our variables, the prevalent presence of outliers for each variable in different classes concern

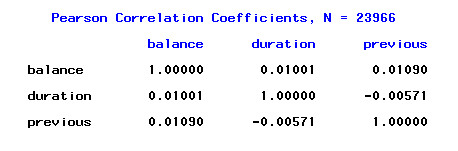
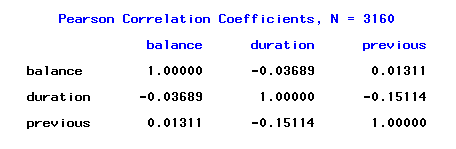
us much more. From the boxplot below, you can see how blatant these outliers are.

****

****

**Next, we will compare the covariance matrices of the three variables with response=yes and response=no.**

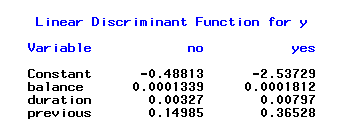
**Y=yes Y=No**

****

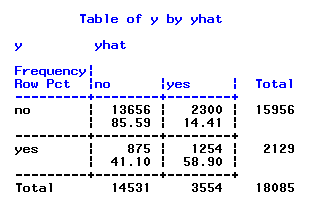
Comparing the correlation structure of the three variables for y=yes and y=no, we found that **they are very similar**. For both classes, **all the variables are almost independent from each other.**

**With the situation of outliers, our expectation of the performance of discriminant analysis in predicting response is rather low**. Employing the winsoring (percentile capping) technique will help us reduce the impact of outliers on the model. We will train the data with and without winsoring technique and compare performances of both.

Using the original training dataset, we trained the following linear discriminant functions.

****

**The confusion matrix for discriminant function analysis follows below:**

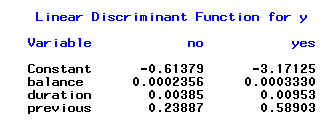
****

**The overall accuracy is 82.44%, the precision is 58.9%.**

****

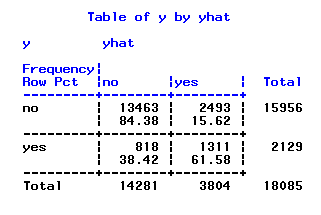
The scatter plot of discriminant scores color coded with response also shows that the linear **discriminant functions did a good job at discriminating** **responses** because we do see a pattern that records with response of yes tends to have larger score on the second discriminant function.

It appears that discriminant analysis does a decent job at classification given the presence of outliers. It only makes us more curious if we can improve classification by using winsoring technique. To winsor a variable simply means to set values of outliers at the value of a certain percentile. For balance, duration and previous, we set the values of the low outliers to be the value of first percentile and the values of high outliers at the value of 99th percentile.

****

**Shown Above are the coefficients of linear discriminant functions after winsoring.**

**Below is the confusion matrix from the classifications using these discriminant functions on the testing data.**

****

We see that by using the winsoring technique, precision improved by almost 3% and the specificity decreased by 0.91%. Considering that the bank benefits much more by calling those who will subscribe term deposit than not calling those who will refuse them, we believe this improvement amenable to profitability.

**4.5 Naïve Bayes**

The R-package KlaR allowed us to use Naïve Bayes analysis

######## naive bayes ##########

library(klaR)

Since Naïve Bayes Model requires the variables to be categorical, we convert variables into categorical variables by using factor function in R-package.

# convert columns to factor.

for(item in name){

train[,item] <- as.factor(train[,item])

To check the accuracy and precision of the Naïve Bayes, we use the predict function in r and use a classification confusion matrix.

pred <- prediction(predictions2,bank\_test$y)

R gave us the best cut off value for this model. With the best cutoff value, 0.8823517 or 0.8823517 gives us the accuracy 86%.

cutoffs <- data.frame(cut=perf@alpha.values[[1]], fpr=perf@x.values[[1]], tpr=perf@y.values[[1]])

cutoffs\_acc <- data.frame(cut=perf@alpha.values[[1]], cut2=perf2@alpha.values[[1]], acc=perf2@y.values[[1]])

cutoffs\_acc[which(cutoffs\_acc$acc==max(cutoffs\_acc$acc)), ]

cut cut2

0.8823517 0.8823517

Since, in this study we are more interested in correctly identifying the ‘yes’ response we will go the best cutoff value, the classification matrix below gives us the overall result of the Naïve Bayes model.

> table(results)

bank\_vali.y

predictions.class no yes

no 14539 1179

yes 1417 950

**Model Accuracy = (950+ 14539) / (950+14539+1179+1417) = 85.64%**

**We get a 85.64 % accuracy in the mode**l

**Model Precision = 950 / (1417+950) = 40.13%**

**Model Precision = 40.13% - This is less that the Precision of the Decision Tree and the Random Forest Model, logistic Regression and Discriminant Analysis.**

1. **Model Comparison**

In this case of Bank Marketing campaign, the bank is more interested in identifying ‘yes’ responders than ‘no’ responders. So, Model Precision that is – how successfully the model predicts 1 (‘yes’) is critically important in our evaluation of models.

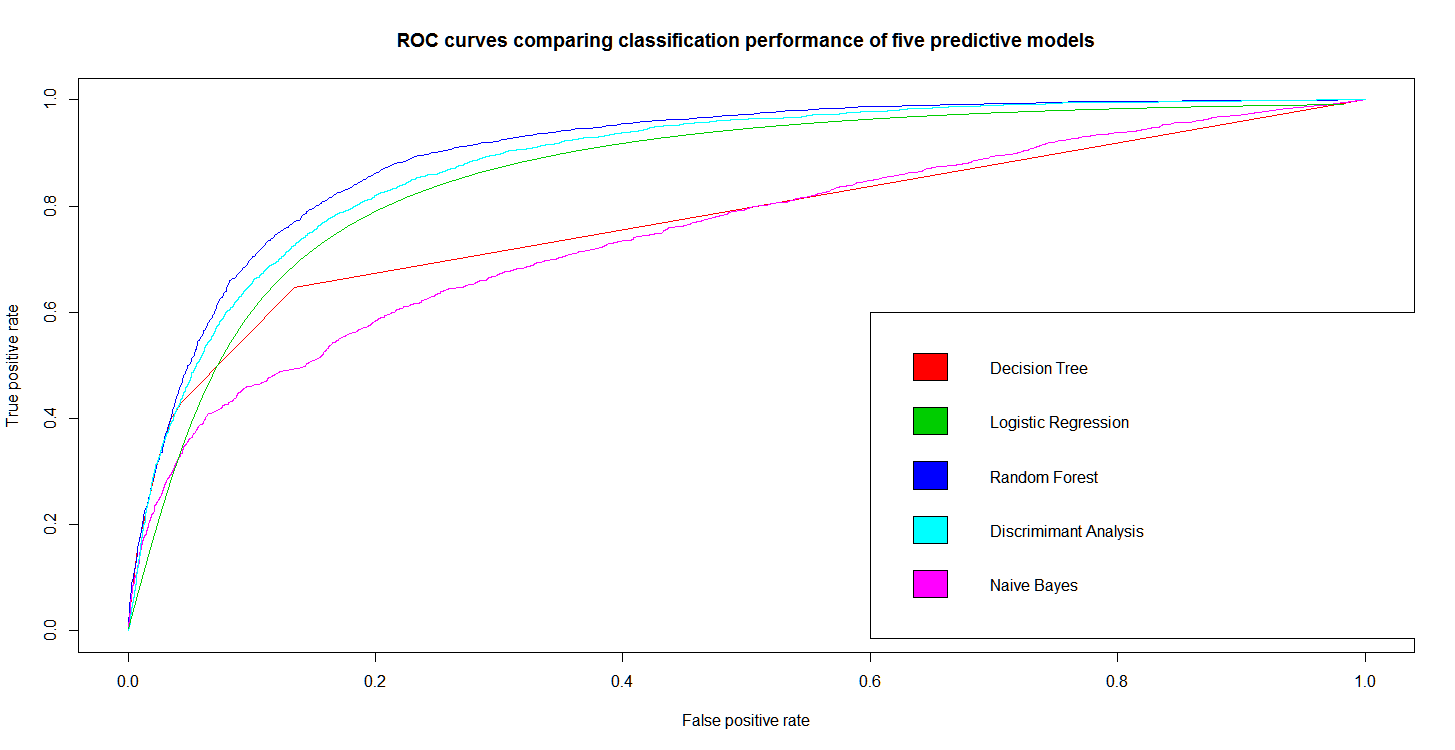
**The below table summarizes the Accuracy, Precision of the Models discussed above:**

|  |  |  |
| --- | --- | --- |
| Model | Accuracy | Precision |
| Decision Tree | **89.89 %** | **40.25%** |
| Random Forest | **88.70%** | **65.70%** |
| Logistic Regression | **89.71%** | **50.63%** |
| Discriminant Analysis | **81.9%** | **61.58%** |
| Naïve Bayes | **85.64%** | **40.13%** |

**The highest Precision was obtained by the Random Forest Model, the Model Accuracy is more or less similar in all the 3 models.**

**The Discriminant Analysis Model has the lowest accuracy = 81.9 % but its Precision is just slightly lower than that of the Random Forest Model.**

**ROC Curve comparing Classification Performance:**

****

**6. Conclusion**

From the above **Summary Tables of Accuracy & Precision and ROC Curve** we can see that the **classification performance of the Random Forest Model is clearly superior than the other models**. The **area under curve (AUC) of the ROC Curve for Random Forest is = 90%** which is the highest and an area greater than 80 % is typically considered a very successful model.

The interpretation from the Random Forest model is that the **variables- duration, poutcome, balance, age, and loan are the most important variables** when it comes to classification of customers as 1 (‘yes’) or 0 (‘no’). The bank should use these variables in deciding whether a potential customer is a 1 or 0.

**Cited Websites**

[**http://www.listendata.com/2015/04/oversampling-for-rare-event-with-sas.html**](http://www.listendata.com/2015/04/oversampling-for-rare-event-with-sas.html)

[**http://www.listendata.com/2015/01/detecting-and-solving-problem-of-outlier.html**](http://www.listendata.com/2015/01/detecting-and-solving-problem-of-outlier.html)