performance

Murugesan Nagarajan

5/8/2018

Analysis

The dataset contains 20 features which are used to predict the target. Our aim is to predict whether a customer with these features will subscribe for a Banking product. The data set is ';' separated data set and we can directly read and use it for our analysis. Caret package will be used for this model development and performance evaluation. We will try to build GBM, Random Forest, Neural Network and Logistic regression from the caret package and then we will display the performance metrics.

The below code will read the data from the CSV file and create an R data frame objects.

```
library("caret")
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
print(getwd())
```

```
## [1] "/Projects/R/CSX415-project/04-more-data"
```

```
raw_data <- read.csv('/Projects/R/CSX415-project/phone_mark/data/bank.csv',sep=';')
```

Attribute Information

The dataset has the below features and we need to identify the feature importance and then need to use these features for classification.

Input variables:

```
#bank client data:
1 - age (numeric)
2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
5 - default: has credit in default? (categorical: 'no','yes','unknown')
6 - housing: has housing loan? (categorical: 'no','yes','unknown')
7 - loan: has personal loan? (categorical: 'no','yes','unknown')
#related with the last contact of the current campaign:
8 - contact: contact communication type (categorical: 'cellular','telephone')
9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11 - 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:
12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
#social and economic context attributes
16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

### Output variable (desired target):
21 - y - has the client subscribed a term deposit? (binary: 'yes','no')
```

Data cleaning

```
sum(is.na(raw_data))
```

```
## [1] 0
```

The dataset does not have any empty value since the above function returned the empty count as 0

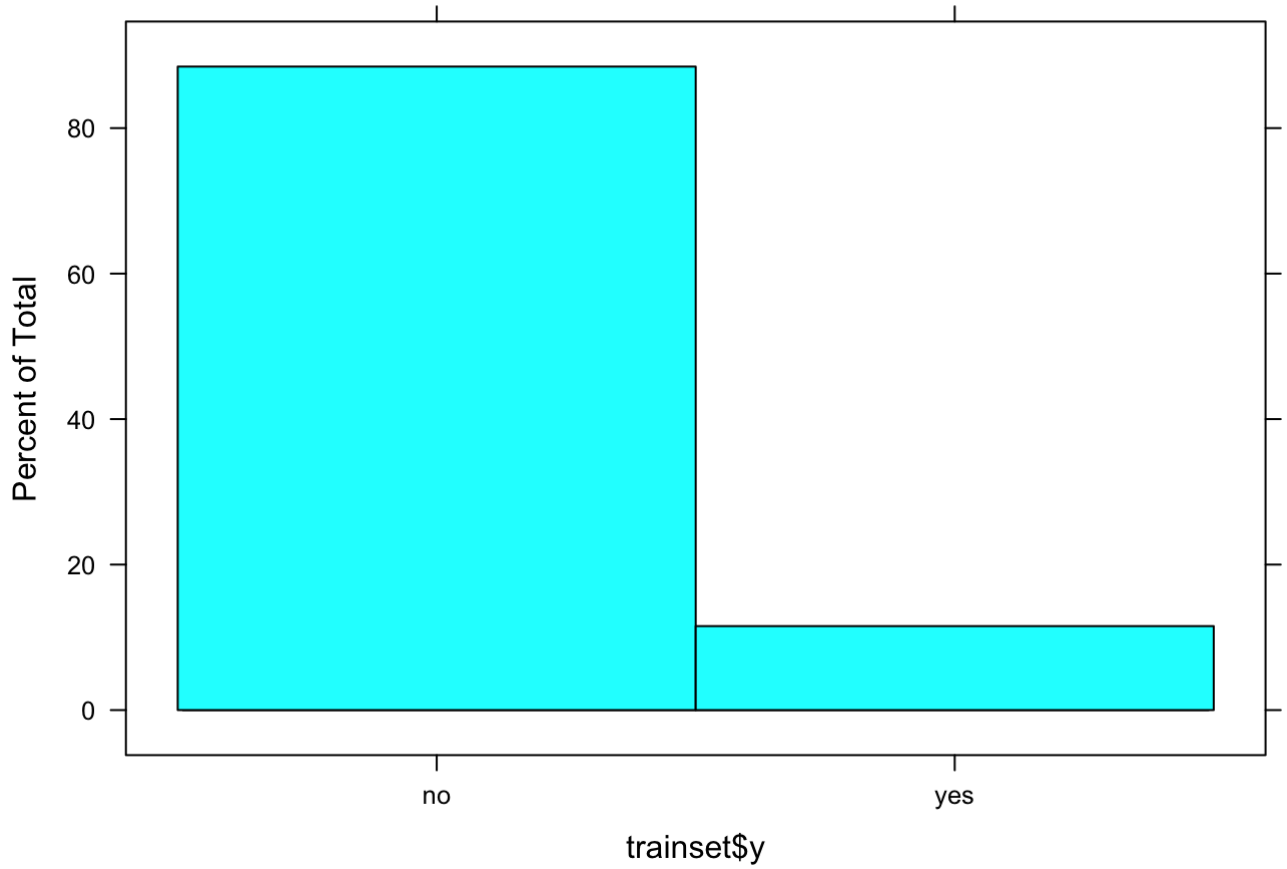
Data split

Use the caret function to divide the data into two set 75% of data as training set and 25% of data as testset

```
index<-createDataPartition(raw_data$y,p=0.5,list=FALSE)
trainset<- raw_data[index,]
testset<-raw_data[-index,]
outcomeName<-'y'
```

Plot the target variable as a histogram chart and find what percentage of the data is yes and what percentage of data is No. It seems we have less than 20% of data is having the target as Yes and more than 80% of the data are having the target as No

```
histogram(trainset$y)
```

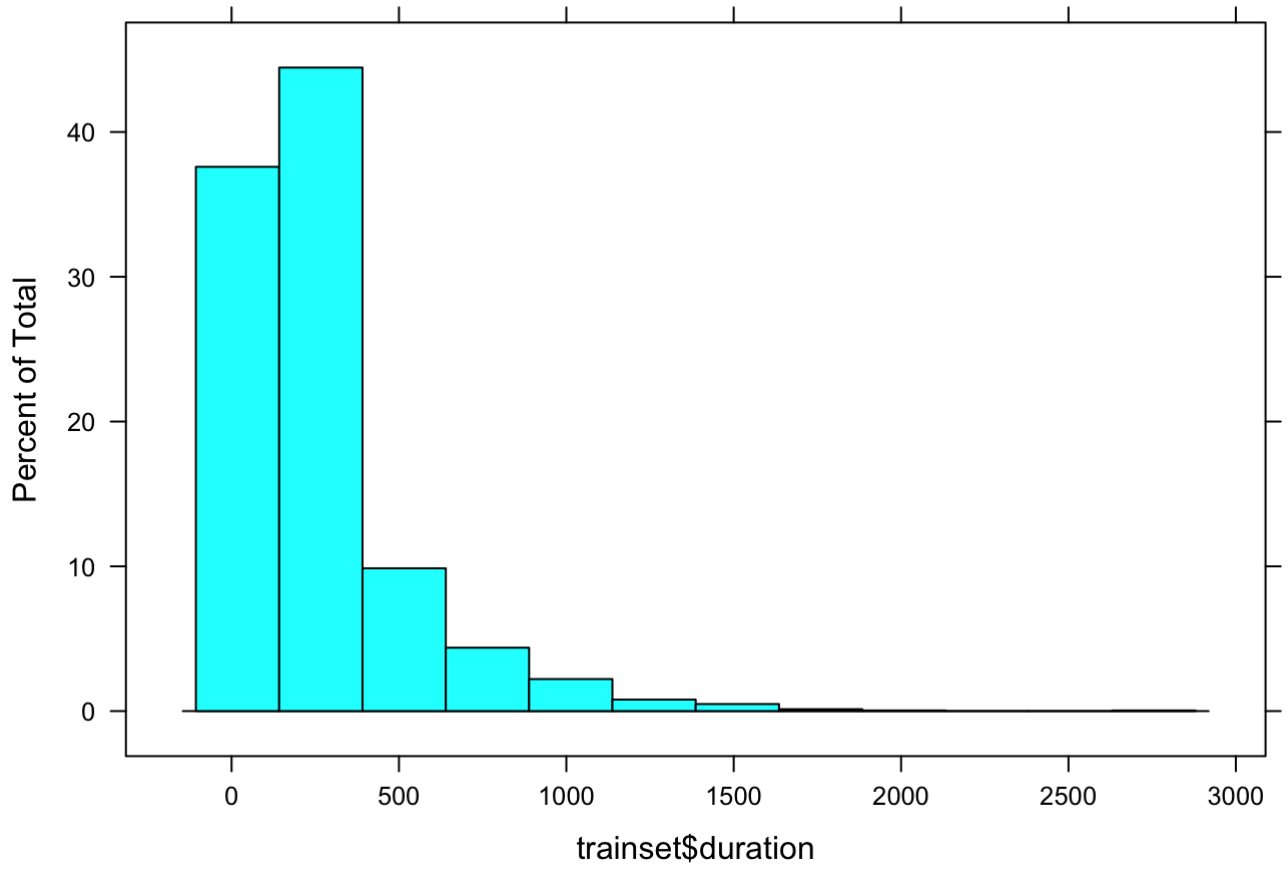


Feature selection using caret

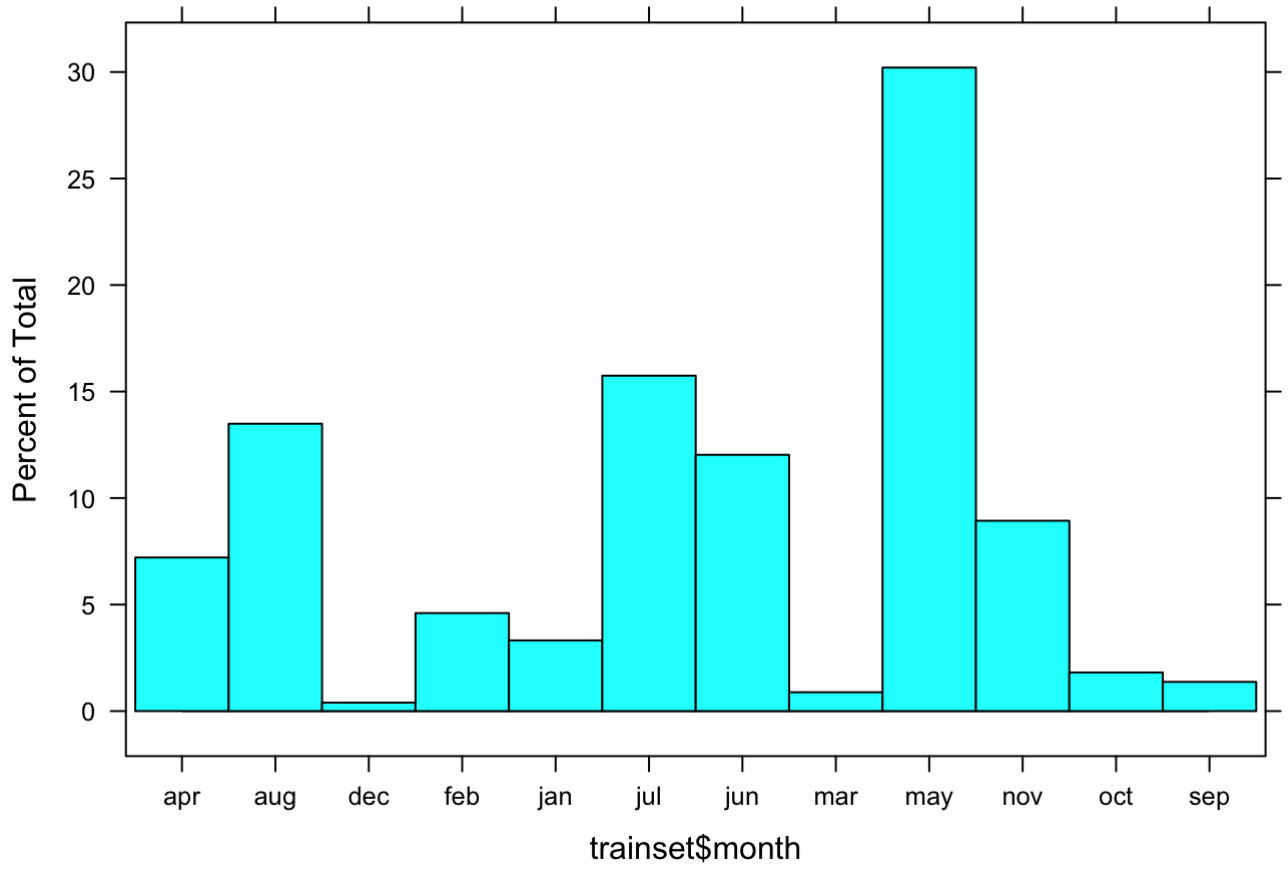
```
control <- rfeControl(functions = rfFuncs,method = "repeatedcv",repeats = 3,verbose = FA
LSE)

predictors<-names(trainset)[!names(trainset) %in% outcomeName]
Loan_Pred_Profile <- rfe(trainset[,predictors], trainset[,outcomeName],rfeControl = cont
rol)
```

consider only the top 5 features as per the above feature selection.

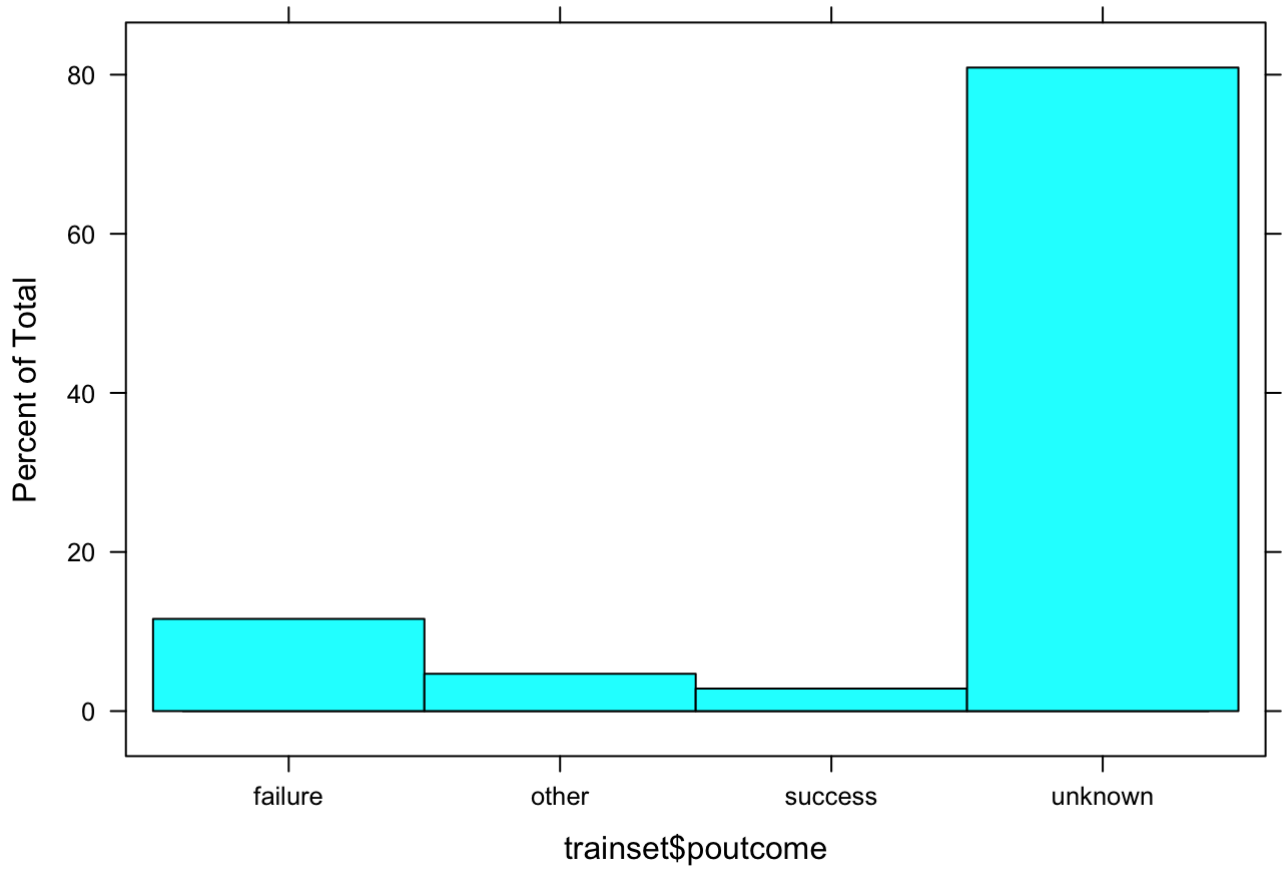
```
predictors<-c("duration", "month", "poutcome", "pdays", "previous")

histogram(trainset$duration)
```



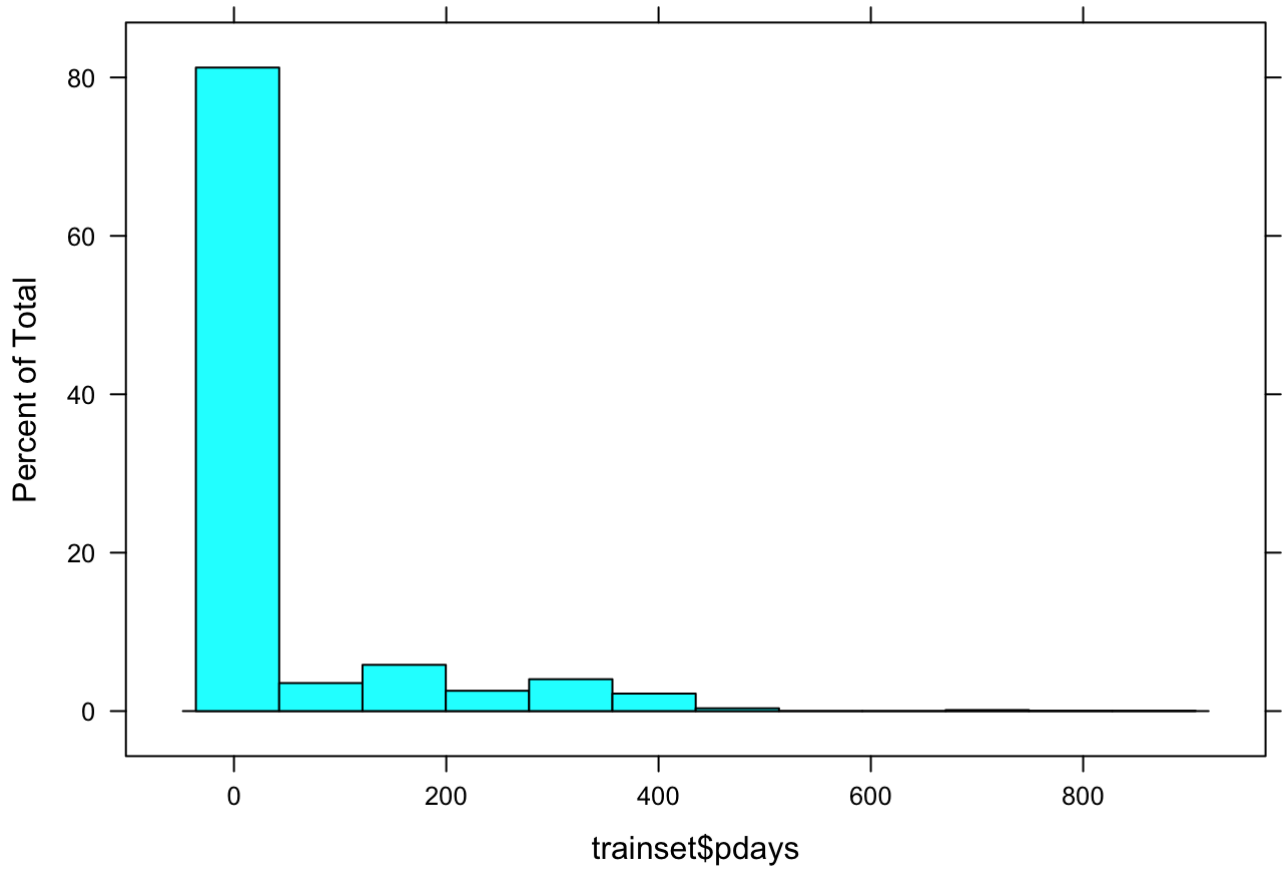

```
histogram(trainset$month)
```



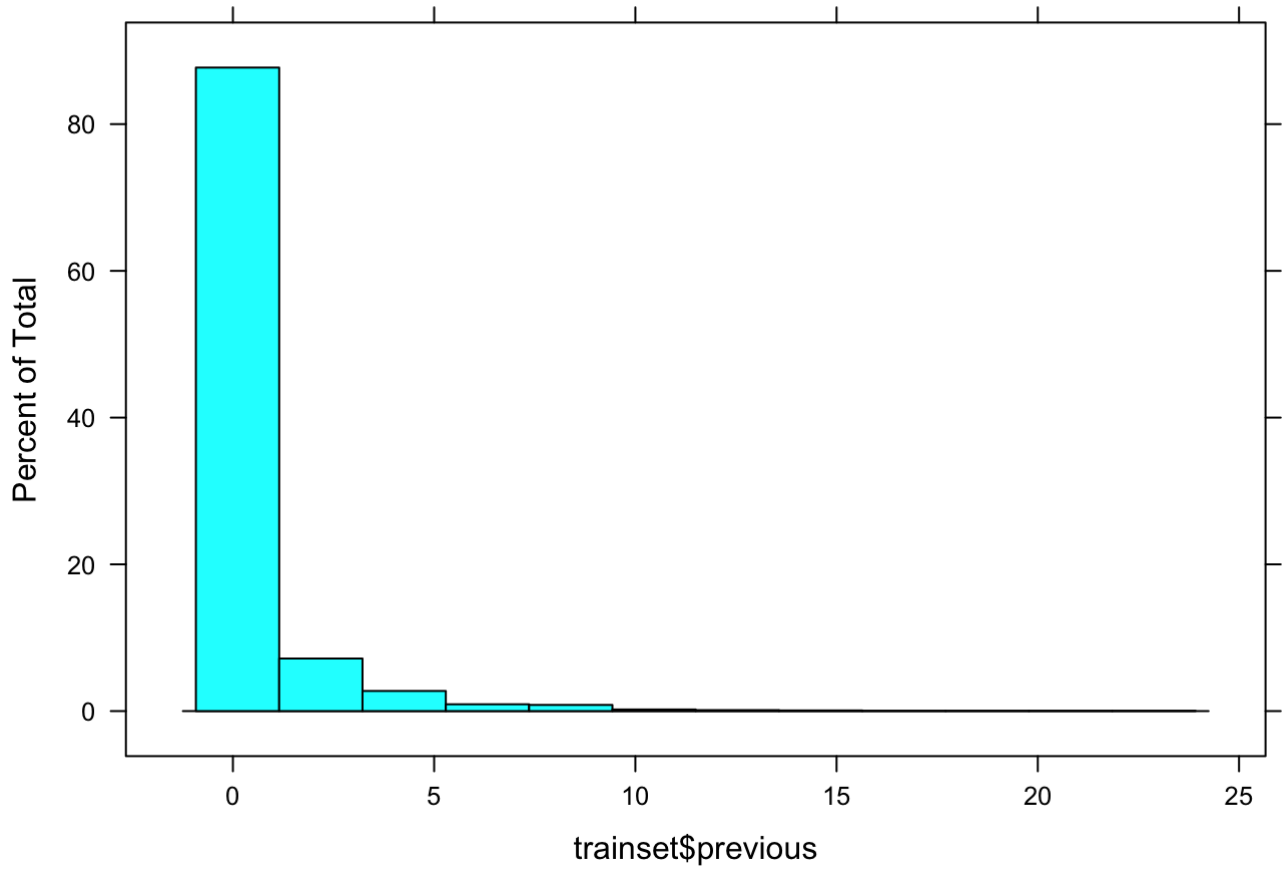

```
histogram(trainset$poutcome)
```




```
histogram(trainset$pdays)
```




```
histogram(trainset$previous)
```



Training the model

Apply GBM, Random forest, Neural Net and Logistic Regression

```
model_gbm<-train(trainset[,predictors],trainset[,outcomeName],method='gbm')
model_rf<-train(trainset[,predictors],trainset[,outcomeName],method='rf')
model_nnet<-train(trainset[,predictors],trainset[,outcomeName],method='nnet')
model_glm<-train(trainset[,predictors],trainset[,outcomeName],method='glm')
```

Prediction

Predict the outcome of test data and then evaluate the model performance

Evaluate the random forest model using the metrics

```
predict<-predict.train(object=model_rf,testset[,predictors],type="raw")
table(predict)
confusionMatrix(predict,testset[,outcomeName])
```

Model performance for Random Forest

confusion matric for the Random Forest model

Prediction	no	yes
No	1927	161
Yes	73	99

evaluate the GBM model and find the metrics

```
predict<-predict.train(object=model_gbm,testset[,predictors],type="raw")
table(predict)
confusionMatrix(predict,testset[,outcomeName])
```

Model performance for GBM

confusion matric for the GBM model

Prediction	no	yes
No	1954	189
Yes	46	71

evaluate the Neural Net model and calculate the metrics.

```
predict<-predict.train(object=model_nnet,testset[,predictors],type="raw")
table(predict)
confusionMatrix(predict,testset[,outcomeName])
```

Model performance for Neural Networks

confusion matric for the Neural Networks

Prediction	no	yes
No	1930	147
Yes	70	113

evaluate the glm and calculate the metrics.

```
predict<-predict.train(object=model_glm,testset[,predictors],type="raw")
confusionMatrix(predict,testset[,outcomeName])
```

Model performance for Logistic Regression

confusion matric for the Logistic Regression

Prediction	no	yes
No	1962	179
Yes	38	81

Conclusion

Calculated the performance metrics, accuracy, Sensitivity,Specificity, and Kappa. As per these metrics, We can either Random Forest or Neural Network model can be used for this classification problem.

Model	Accuracy	Sensitivity	Specificity	Kappa
GBM	90	97	31	37
Random Forest	90	97	33	39
Neural Network	90	96	37	41
Logistic Regression	89	97	28	35