Midterm_Bank_Additional_Full_Dataset_Starwin

Starwin

March 11, 2018

FINC614 Introduction to Data Science

Please find below R scripts and output for the mid term questions. I have used Bank_Additional_Full_Dataset from UCI.

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
library(caret)

## Loading required package: lattice

library(ROCR)

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

1. Pick any dataset from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/index.php) suitable for building a classification model. Provide a brief description of the data (type of data, variables etc.). (2 points)

Read csv file into and create DF

```
bank <- read.csv("C:/Users/starw/OneDrive/Documents/NJCU Spring
Semester/FINC614_Intro to Data science/Data Sets/bank-additional/bank-
additional/bank-additional-full.csv", sep=";")

### This dataset is based on "Bank Marketing" UCI dataset
### Title: Bank Marketing (with social/economic context)
### The data 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 ('yes') or not ('no')
subscribed.
```

check the varaiable names and ouput variable name and position

```
colnames(bank)

##  [1] "age"            "job"            "marital"        "education"
##  [5] "default"        "housing"        "loan"           "contact"
##  [9] "month"          "day_of_week"    "duration"       "campaign"
## [13] "pdays"          "previous"       "poutcome"       "emp.var.rate"
## [17] "cons.price.idx" "cons.conf.idx"  "euribor3m"      "nr.employed"
## [21] "y"

### Observation:
### So, we have 20 predictors and 1 response.
```

Dimension of our data

```
dim(bank)

## [1] 41188    21

### Observation:
### So, we have 41188 rows of observations available in our raw data.
```

Quick glance over our data set to check different default or null values

```
str(bank)

## 'data.frame':    41188 obs. of  21 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1
8 8 1 2 10 8 ...
##  $ marital       : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2
2 2 3 3 ...
##  $ education     : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4
3 6 8 6 4 ...
##  $ default       : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1
1 ...
##  $ housing       : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3
3 ...
##  $ loan          : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1
```

```
1 ...
##  $ contact       : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2
2 2 2 2 ...
##  $ month         : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7
7 7 7 ...
##  $ day_of_week   : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2
2 2 2 ...
##  $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome      : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2
2 2 2 2 2 2 ...
##  $ emp.var.rate  : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ cons.price.idx: num  94 94 94 94 94 ...
##  $ cons.conf.idx : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -
36.4 -36.4 ...
##  $ euribor3m     : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ nr.employed   : num  5191 5191 5191 5191 5191 ...
##  $ y             : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

### Observation:
### So, we have "unknown" as one of our category in our data set.
```

Checking each of our predictors, their categories and distribution

```
summary(bank)

##       age                 job            marital     
##  Min.   :17.00   admin.     :10422   divorced: 4612  
##  1st Qu.:32.00   blue-collar: 9254   married :24928  
##  Median :38.00   technician : 6743   single  :11568  
##  Mean   :40.02   services   : 3969   unknown :   80  
##  3rd Qu.:47.00   management : 2924                   
##  Max.   :98.00   retired    : 1720                   
##                  (Other)    : 6156                   
##                education        default         housing     
##  university.degree  :12168   no     :32588   no     :18622  
##  high.school        : 9515   unknown: 8597   unknown:  990  
##  basic.9y           : 6045   yes    :    3   yes    :21576  
##  professional.course: 5243                                  
##  basic.4y           : 4176                                  
##  basic.6y           : 2292                                  
##  (Other)            : 1749                                  
##       loan            contact          month       day_of_week
##  no     :33950   cellular :26144   may    :13769   fri:7827   
##  unknown:  990   telephone:15044   jul    : 7174   mon:8514   
##  yes    : 6248                     aug    : 6178   thu:8623   
##                                    jun    : 5318   tue:8090   
##                                    nov    : 4101   wed:8134   
##                                    apr    : 2632              
```

```
##                                    (Other): 2016
##     duration         campaign          pdays          previous
##  Min.   :   0.0   Min.   : 1.000   Min.   :  0.0   Min.   :0.000
##  1st Qu.: 102.0   1st Qu.: 1.000   1st Qu.:999.0   1st Qu.:0.000
##  Median : 180.0   Median : 2.000   Median :999.0   Median :0.000
##  Mean   : 258.3   Mean   : 2.568   Mean   :962.5   Mean   :0.173
##  3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.:999.0   3rd Qu.:0.000
##  Max.   :4918.0   Max.   :56.000   Max.   :999.0   Max.   :7.000
##
##         poutcome      emp.var.rate      cons.price.idx  cons.conf.idx
##  failure    : 4252   Min.   :-3.40000   Min.   :92.20   Min.   :-50.8
##  nonexistent:35563   1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7
##  success    : 1373   Median : 1.10000   Median :93.75   Median :-41.8
##                      Mean   : 0.08189   Mean   :93.58   Mean   :-40.5
##                      3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4
##                      Max.   : 1.40000   Max.   :94.77   Max.   :-26.9
##
##    euribor3m      nr.employed     y
##  Min.   :0.634   Min.   :4964   no :36548
##  1st Qu.:1.344   1st Qu.:5099   yes: 4640
##  Median :4.857   Median :5191
##  Mean   :3.621   Mean   :5167
##  3rd Qu.:4.961   3rd Qu.:5228
##  Max.   :5.045   Max.   :5228
##

### observation:
### We have to handle this "unknow" categories in our data. This should be
treated as "NA"
```

One more sanity test by checking first few rows of our data

```
head(bank)

##   age       job marital   education default housing loan   contact month
## 1  56 housemaid married    basic.4y      no      no   no telephone   may
## 2  57  services married high.school unknown      no   no telephone   may
## 3  37  services married high.school      no     yes   no telephone   may
## 4  40    admin. married    basic.6y      no      no   no telephone   may
## 5  56  services married high.school      no      no  yes telephone   may
## 6  45  services married    basic.9y unknown      no   no telephone   may
##   day_of_week duration campaign pdays previous    poutcome emp.var.rate
## 1         mon      261        1   999        0 nonexistent          1.1
## 2         mon      149        1   999        0 nonexistent          1.1
## 3         mon      226        1   999        0 nonexistent          1.1
## 4         mon      151        1   999        0 nonexistent          1.1
## 5         mon      307        1   999        0 nonexistent          1.1
## 6         mon      198        1   999        0 nonexistent          1.1
##   cons.price.idx cons.conf.idx euribor3m nr.employed  y
## 1         93.994         -36.4     4.857        5191 no
## 2         93.994         -36.4     4.857        5191 no
```

```
## 3          93.994         -36.4     4.857        5191 no
## 4          93.994         -36.4     4.857        5191 no
## 5          93.994         -36.4     4.857        5191 no
## 6          93.994         -36.4     4.857        5191 no

### observation:
### Again "unknow" category of data needs to be handled. This should be treated as "NA"
```

2. Count the number of rows that have missing data. Remove the missing data. (2 points)

Count the “NA” in our data

```
sum(is.na(bank))

## [1] 0

### observation:
### There is no missing data.
```

Count the complete cases in our data

```
sum(complete.cases(bank))

## [1] 41188

### observation:
###  There is no missing cases in  our data.
```

But, I don’t want to keep “unknown” category in our DF. I believe that this will misguide our predictions. So, I am going to remove this “unknown” catergory rows from our DF.

```
bank[bank=="unknown"] <- NA

### observation:
### We changed all "unknown" value int "NA" in our data.
```

Now again count the “NA” in our data

```
sum(is.na(bank))

## [1] 12718

### observation:
### There is 12718 "NA" are available in our data now.
```

Again count the complete cases in our data

```
sum(complete.cases(bank))
```

```
## [1] 30488

### observation:
###  There are only 30488 complete cases in  our data now. Remianing 10700
are having "NA".
```

Removing "NA" from our data.

```
bank_cleaned <- na.omit(bank)

### observation:
### Cleaned DF is created with 30488 rown and 21 variables.
```

3. Check if there are any duplicate rows in the data. If there are duplicates report how many and remove them. (3 points)

Get the count of distinct rows.

```
count(distinct(bank_cleaned))

## # A tibble: 1 x 1
##       n
##   <int>
## 1 30478

### observation:
### 10 rows are duplicate rows. Remianing 30478 rows are distinct rows in our
data.
```

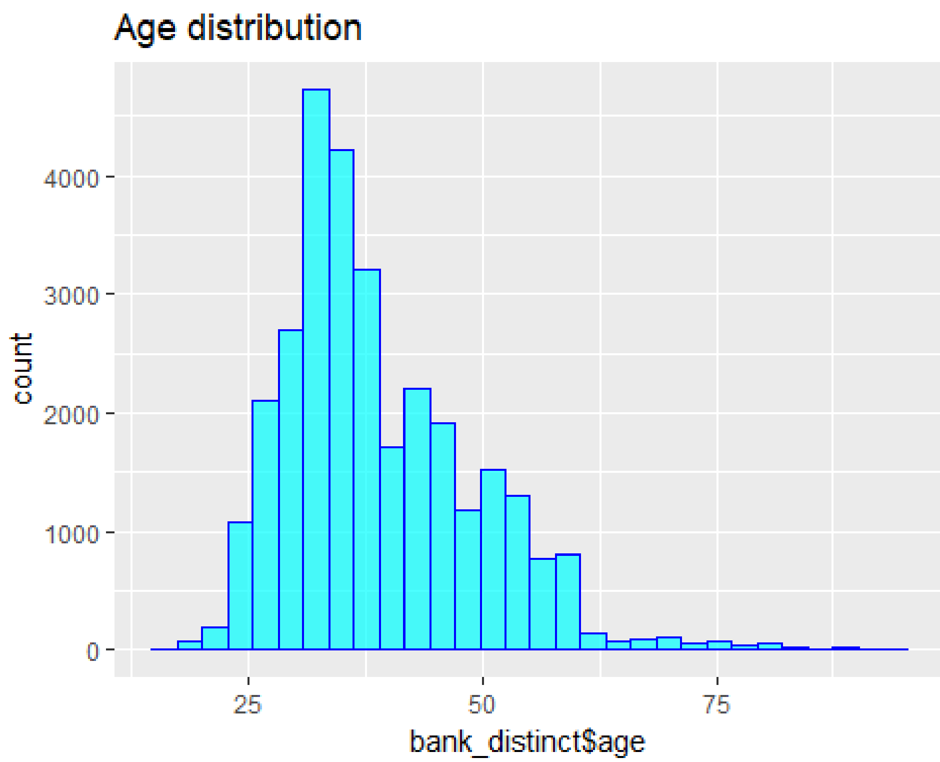
create a new data frame with distinct rows

```
bank_distinct <- distinct(bank_cleaned)

### observation:
### New DF is created with 30478 rows with 21 variables.
```

4. Create at least one derived feature (derived data column) (5 points)

Find "Age" distribution will help us to derive a feture from it.

```
ggplot(data=bank_distinct, aes(bank_distinct$age)) + geom_histogram(bandwidth
= 5, col="blue", fill=rgb(0,1,1,0.7))+ ggtitle("Age distribution")

## Warning: Ignoring unknown parameters: bandwidth

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
### observation:
### We can group age limits into three or four groups and create a new
feature.
```

Creating Age_group as a new feature.

```
for(i in 1 : nrow(bank_distinct)){
  if (bank_distinct$age[i] < 20){
    bank_distinct$age_group[i] = 'Teens'
  } else if (bank_distinct$age[i] < 35 & bank_distinct$age[i] > 19){
    bank_distinct$age_group[i] = 'Adults'
  } else if (bank_distinct$age[i] < 60 & bank_distinct$age[i] > 34){
    bank_distinct$age_group[i] = 'Middle Aged'
  } else if (bank_distinct$age[i] > 59){
    bank_distinct$age_group[i] = 'Senior Citizens'
  }

}
```

Converting Age_group as factor.

```
bank_distinct$age_group<-as.factor(bank_distinct$age_group)

### Convert our response value from yes and no to 1 and 0.
bank_distinct$y<-ifelse(bank_distinct$y =='yes', 1,0)

### Converting our response into factor.
bank_distinct$y<-as.factor(bank_distinct$y)
```

5. Create a frequency table for your data (choose appropriate data attributes for a frequency table) (5 points)

Creating freeqquency table by using “age_group”, “job” and response “y”

```
table(bank_distinct$age_group,bank_distinct$job,bank_distinct$y)
```

```
## , ,  = 0
##
##
##                  admin. blue-collar entrepreneur housemaid management
##   Adults           3364        2023          258       125        562
##   Middle Aged      4109        3184          720       449       1442
##   Senior Citizens    46          15           10        29         21
##   Teens               0           0            0         0          0
##
##                  retired self-employed services student technician
##   Adults               6           380     1214     369       2082
##   Middle Aged        503           567     1381      16       2731
##   Senior Citizens    349            13        3       0         15
##   Teens                0             0        0      22          0
##
##                  unemployed unknown
##   Adults                240       0
##   Middle Aged           365       0
##   Senior Citizens         7       0
##   Teens                   0       0
##
## , ,  = 1
##
##
##                  admin. blue-collar entrepreneur housemaid management
##   Adults            606         196           25        12         81
##   Middle Aged       575         249           72        52        190
##   Senior Citizens    34           7            4        23         15
##   Teens               0           0            0         0          0
##
##                  retired self-employed services student technician
##   Adults               1            57      139     182        309
##   Middle Aged         79            73      119       5        324
##   Senior Citizens    277             2        0       0          8
##   Teens                0             0        0      16          0
##
##                  unemployed unknown
##   Adults                 56       0
##   Middle Aged            67       0
```

```
##   Senior Citizens          3       0
##   Teens                    0       0
```

count the combination of age_group and job

```r
count(bank_distinct,age_group,job)
```

```
## # A tibble: 33 x 3
##    age_group job               n
##    <fct>     <fct>         <int>
##  1 Adults    admin.         3970
##  2 Adults    blue-collar    2219
##  3 Adults    entrepreneur    283
##  4 Adults    housemaid       137
##  5 Adults    management      643
##  6 Adults    retired           7
##  7 Adults    self-employed   437
##  8 Adults    services       1353
##  9 Adults    student         551
## 10 Adults    technician     2391
## # ... with 23 more rows
```

count the combination of age_group and y.

```r
count(bank_distinct,age_group,y)
```

```
## # A tibble: 8 x 3
##   age_group       y         n
##   <fct>           <fct> <int>
## 1 Adults          0     10623
## 2 Adults          1      1664
## 3 Middle Aged     0     15467
## 4 Middle Aged     1      1805
## 5 Senior Citizens 0       508
## 6 Senior Citizens 1       373
## 7 Teens           0        22
## 8 Teens           1        16

### observation.
### Majority of adults and middle aged men are subscribed for term deposit
```

Creating another freequency table by using “age_group”, “loan”, response “y”

```r
table(bank_distinct$age_group,bank_distinct$loan,bank_distinct$y)
```

```
## , ,  = 0
## 
##                  
##                      no unknown   yes
##   Adults           8896       0  1727
##   Middle Aged     13084       0  2383
##   Senior Citizens   437       0    71
##   Teens              20       0     2
```

```
##
## , ,  = 1
##
##
##                        no unknown   yes
##   Adults             1417       0   247
##   Middle Aged        1527       0   278
##   Senior Citizens     317       0    56
##   Teens                12       0     4
```

count the combination of loan and y.

```
count(bank_distinct,loan,y)
```

```
## # A tibble: 4 x 3
##   loan  y         n
##   <fct> <fct> <int>
## 1 no    0     22437
## 2 no    1      3273
## 3 yes   0      4183
## 4 yes   1       585
```

```
### observation:
### If housing is there for a client, then there are lot a chance the client won't subscribe for term deposit.
```

Creating another freequency table by using "age_group", "housing", response "y"

```
table(bank_distinct$age_group,bank_distinct$housing,bank_distinct$y)
```

```
## , ,  = 0
##
##
##                       no unknown  yes
##   Adults            4899       0 5724
##   Middle Aged       7108       0 8359
##   Senior Citizens    233       0  275
##   Teens                6       0   16
##
## , ,  = 1
##
##
##                       no unknown  yes
##   Adults             749       0  915
##   Middle Aged        797       0 1008
##   Senior Citizens    163       0  210
##   Teens                7       0    9
```

count the combination of housing and y.

```
count(bank_distinct,housing,y)
```

```
## # A tibble: 4 x 3
##   housing y         n
##   <fct>   <fct> <int>
## 1 no      0     12246
## 2 no      1      1716
## 3 yes     0     14374
## 4 yes     1      2142

### observation:
### If loan is there for a client, then there are lot a chance the client
won't subscribe for term deposit.
```

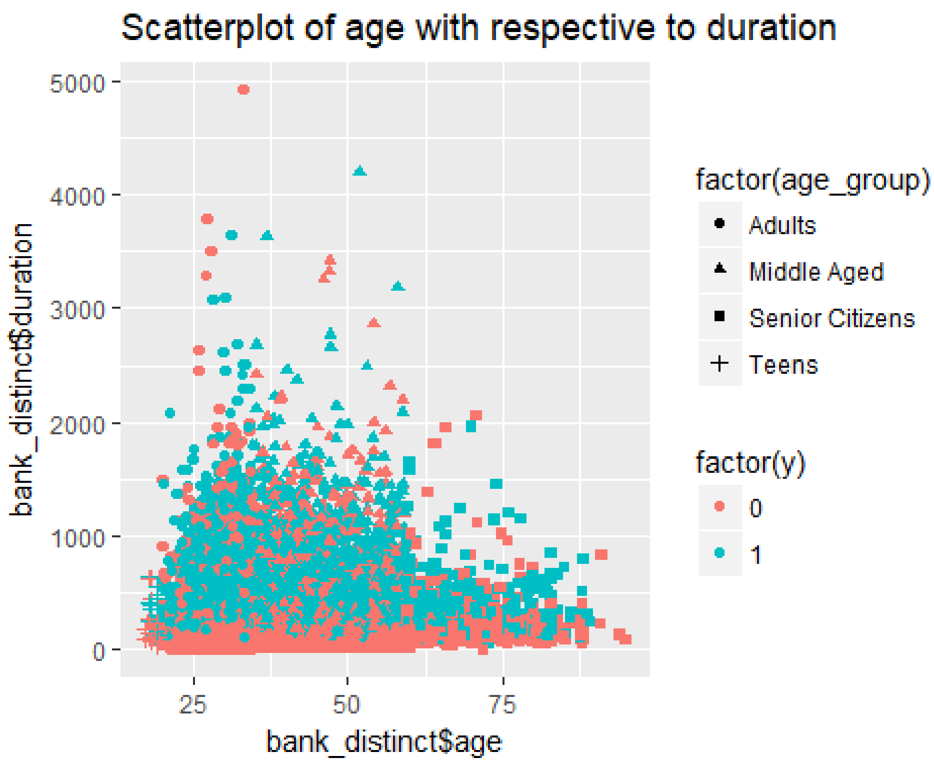
6. Report summary statistics (Mean, median, standard deviation, quartiles and range) for your data (5 points)

```
summary(bank_distinct$duration)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   103.0   181.0   259.5   321.0  4918.0

quantile(bank_distinct$duration)

##   0%  25%  50%  75% 100%
##    0  103  181  321 4918
```

7. Use ggplot2 to plot the following types of graphs with your data. Choose data attributes that are meaningful to plot for each graph. Make inferences about your data based on the graphs (e.g. correlation, shape of distribution etc).

a. Scatter plot (2 points)

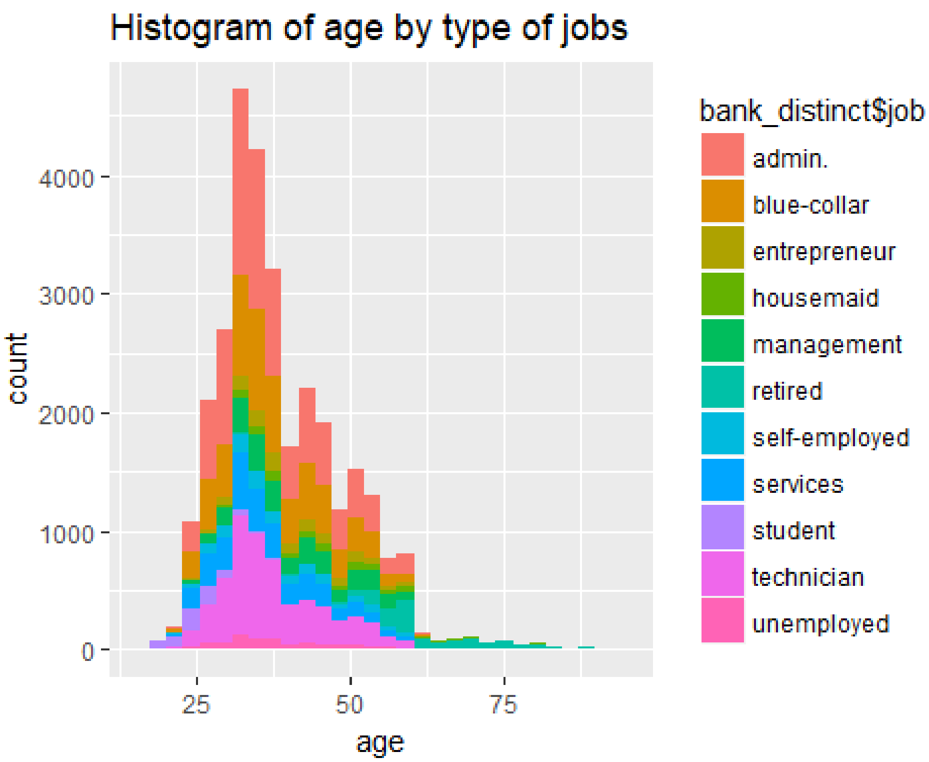
```
ggplot(data=bank_distinct,
aes(bank_distinct$age,bank_distinct$duration,color=factor(y),
shape=factor(age_group))) + geom_point() + geom_jitter() +
ggtitle("Scatterplot of age with respective to duration")
```



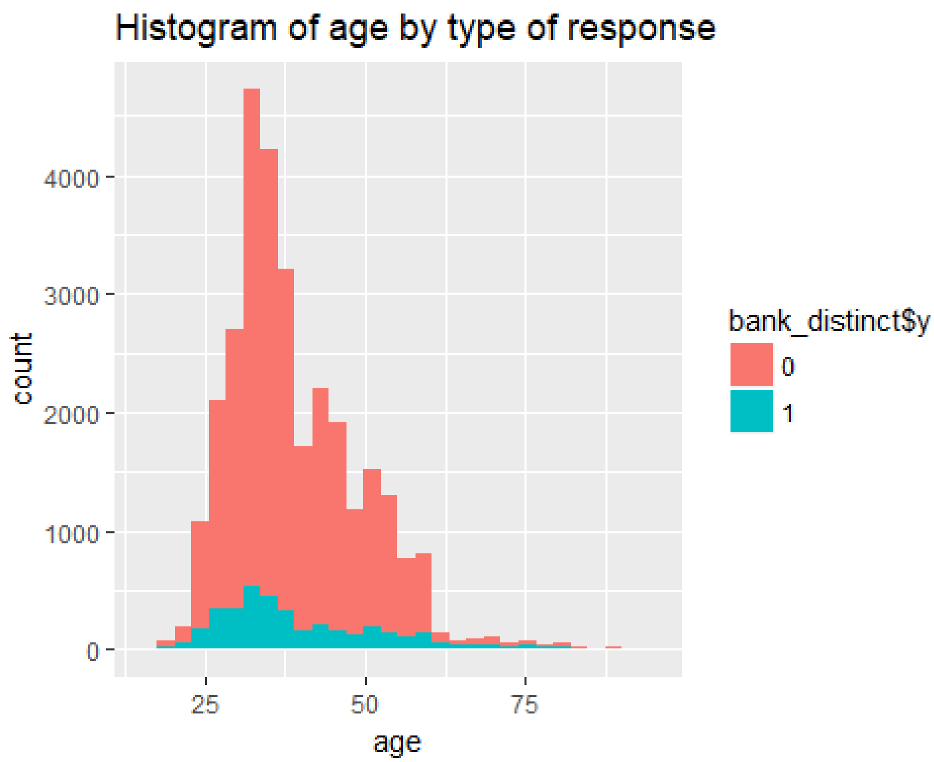

```
### observation:
### Relationship between age and duration is shown above graph as slight bell curve by excluding few outliers.
```

b. Histogram

```
ggplot(data=bank_distinct, aes(age, fill = bank_distinct$job)) +
geom_histogram() + ggtitle("Histogram of age by type of jobs")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



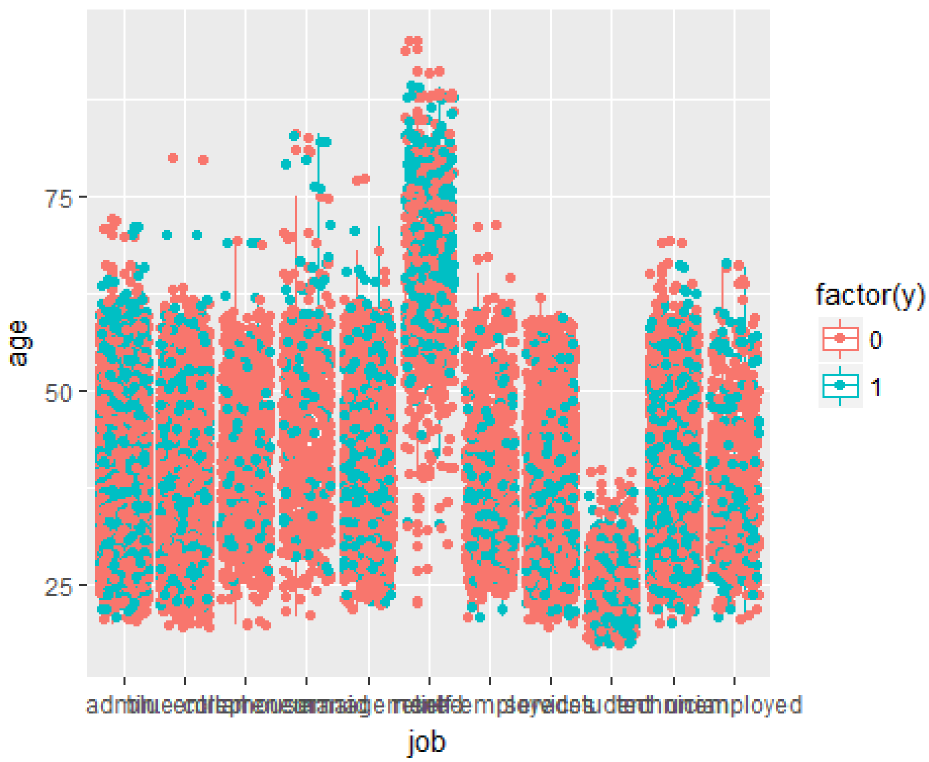

```
### observation:
### The data with respective to age has been distributed as bell curve by
excluding outliers.

ggplot(data=bank_distinct, aes(age, fill = bank_distinct$y)) +
geom_histogram() + ggtitle("Histogram of age by type of response")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
### observation:
### The data with respective to age has been distributed as bell curve by
excluding outliers.
```

c. Box Plot

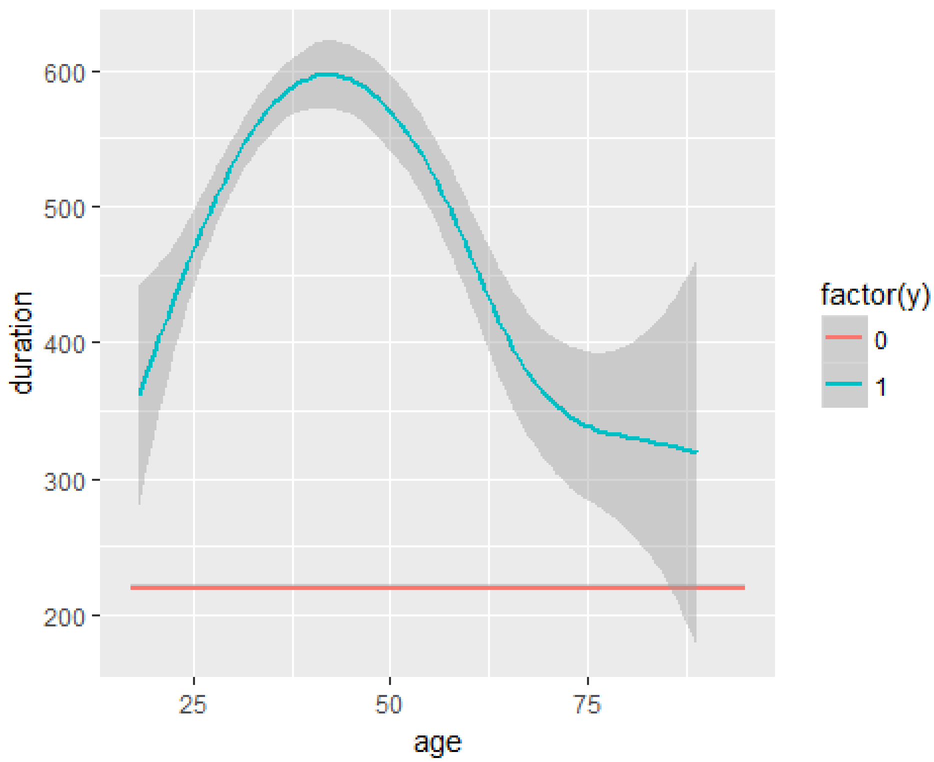
```
ggplot(bank_distinct, aes(job, age, color = factor(y))) + geom_boxplot() +
geom_jitter()
```




```
### observation:
### Student, Retiered and Housemaid are not densely populated jobs among the people in this data.
```

d. Line graph

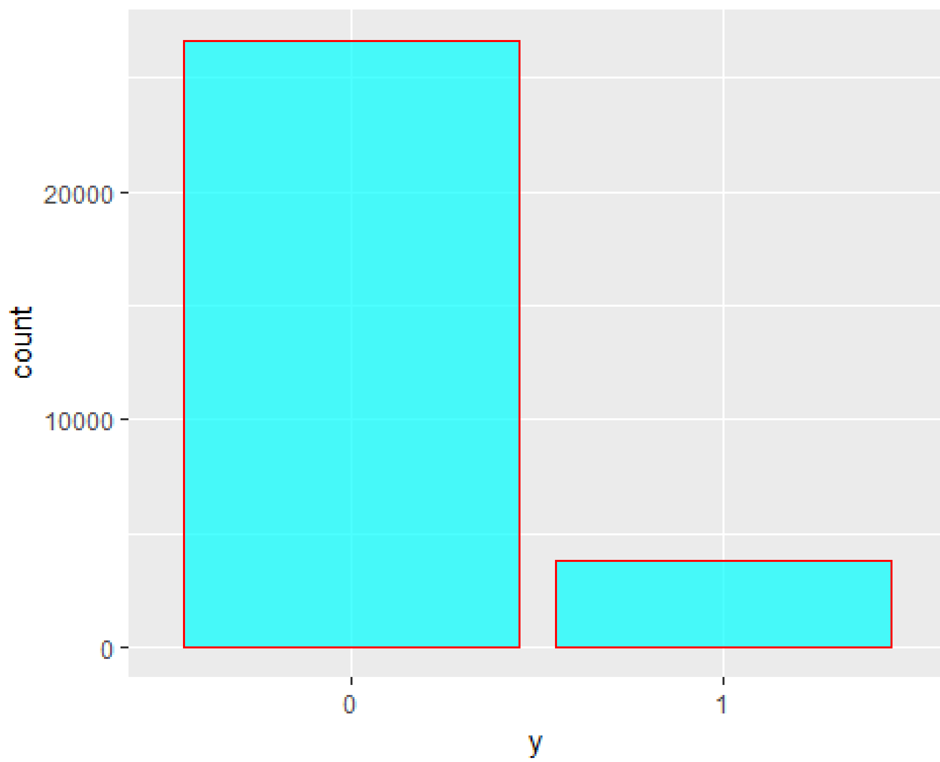
```
ggplot(bank_distinct, aes(age, duration, color = factor(y))) + geom_smooth()

## `geom_smooth()` using method = 'gam'
```




```
### observation:
### The relationship between age and duration with respective to response value as Yes is a bell curve.
### The relationship between age and duration with respective to response value as No is a linear line.
```

8. Check if there is an imbalance in your dataset. Comment on whether there is an imbalance or not. (3 points)

```
load_balance <- summarise(group_by(bank_distinct,y), count = n())
ggplot(data=load_balance, aes(x=y, y=count)) + geom_bar(stat= "identity", color="red", fill=rgb(0,1,1,0.7))
```




```
### observation:
### Yes. there is an imbalance prsesnt in my dataset as per the bar chart.
### So, I tried to split the data set into two seperate by using nonexistent
flag in poutcome variable. But that also didn't do much difference in final
results.

##old_Customer<-subset(bank_distinct, bank_distinct$poutcome !=
"nonexistent")
##new_Customer<-subset(bank_distinct, bank_distinct$poutcome ==
"nonexistent")
```

9. Split your dataset into a training and test dataset choosing the split percentage based on the size of your dataset (2 points)

```
set.seed(12345)
intrain <- createDataPartition(bank_distinct$y,p=0.70,list = FALSE)
train <- bank_distinct[intrain,]
test <- bank_distinct[-intrain,]
table(train$y)

##
##     0     1
## 18634  2701
```

```
table(test$y)

##
##    0    1
## 7986 1157
```

10. Use a k fold cross validation to train the models below. Choose k based on the size of your dataset and the time it would take to fit the model. (2 points)

```
cvctrl <- trainControl(method = "cv", number=10)
```

11. Fit the following models to your training data and predict the class of a meaningful response variable of your choice using the test dataset. Describe the response (dependent) variable and the predictor (independent) variables you have used in your models. If you did not use any predictor variables that were in your dataset, then explain why.

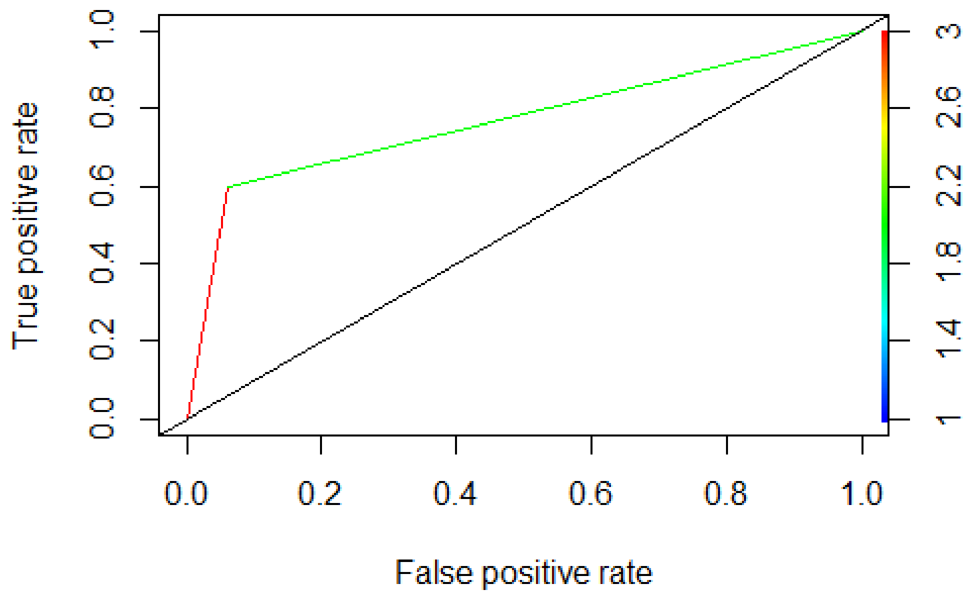
a. Decision tree (5 points)

```
### Fit model
modFit <- train(y ~ ., method='rpart', trControl = cvctrl, data=train)
decisiontreemodel <- modFit$finalModel
print(modFit$finalModel)

## n= 21335
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 21335 2701 0 (0.87340052 0.12659948)
##    2) nr.employed>=5087.65 18286 1351 0 (0.92611834 0.07388166)
##      4) duration< 524.5 16289  503 0 (0.96912027 0.03087973) *
##      5) duration>=524.5 1997  848 0 (0.57536304 0.42463696)
##       10) duration< 834.5 1280  433 0 (0.66171875 0.33828125) *
##       11) duration>=834.5 717  302 1 (0.42119944 0.57880056) *
##    3) nr.employed< 5087.65 3049 1350 0 (0.55723188 0.44276812)
##      6) duration< 158.5 1058  158 0 (0.85066163 0.14933837) *
##      7) duration>=158.5 1991  799 1 (0.40130588 0.59869412) *

### summary(decisiontreemodel)

### Predict model
predictions <- predict(modFit, newdata = test, type='raw')
```

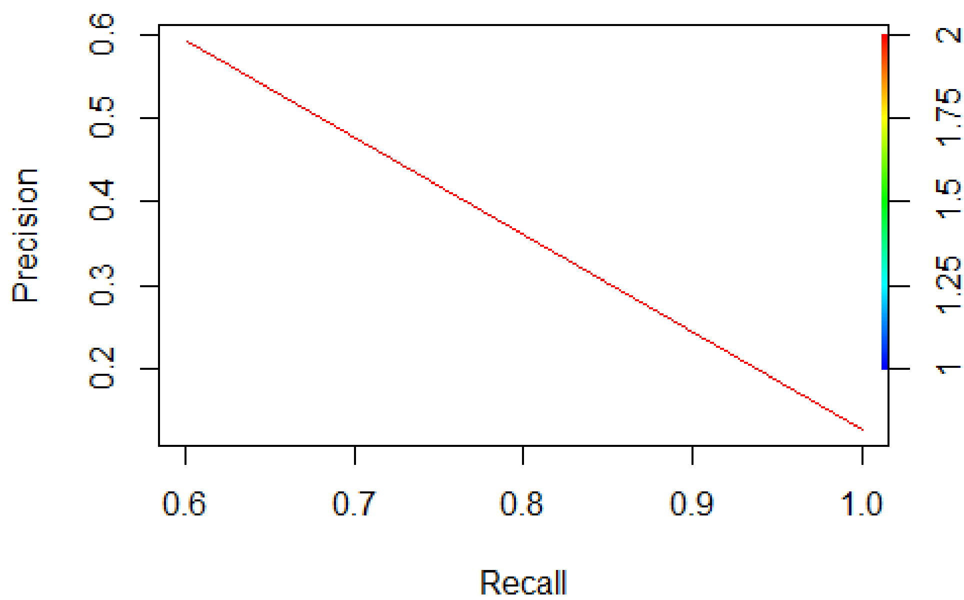
```r
### Check accuracy
confusionMatrix(predictions,test$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7509  462
##          1  477  695
##
##                Accuracy : 0.8973
##                  95% CI : (0.8909, 0.9034)
##     No Information Rate : 0.8735
##     P-Value [Acc > NIR] : 9.749e-13
##
##                   Kappa : 0.538
##  Mcnemar's Test P-Value : 0.6478
##
##             Sensitivity : 0.9403
##             Specificity : 0.6007
##          Pos Pred Value : 0.9420
##          Neg Pred Value : 0.5930
##              Prevalence : 0.8735
##          Detection Rate : 0.8213
##    Detection Prevalence : 0.8718
##       Balanced Accuracy : 0.7705
##
##        'Positive' Class : 0
##
```

```r
ROCRpred <- prediction(as.numeric(predictions), as.numeric(test$y))
### ROC Curve
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
abline(0, 1)
```




```
### AUC
auc_ROCR <- performance(ROCRpred, measure = "auc")
auc_ROCR <- auc_ROCR@y.values[[1]]
print(auc_ROCR)

## [1] 0.770481

### Precision/Recall
RPperf <- performance(ROCRpred, "prec", "rec");
plot(RPperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```




```
### Precision/Recall
print(RPperf)

## An object of class "performance"
## Slot "x.name":
## [1] "Recall"
##
## Slot "y.name":
## [1] "Precision"
##
## Slot "alpha.name":
## [1] "Cutoff"
##
## Slot "x.values":
## [[1]]
## [1] 0.0000000 0.6006914 1.0000000
##
##
## Slot "y.values":
## [[1]]
## [1]       NaN 0.5930034 0.1265449
##
##
## Slot "alpha.values":
## [[1]]
## [1] Inf   2   1
```

b. Logistic regression (5 points)

```
#Fit model
modFit <- train(y
~age+duration+pdays+poutcome+day_of_week+month+contact+emp.var.rate+cons.pric
e.idx+cons.conf.idx+nr.employed, method='glm', trControl = cvctrl,
data=train, family=binomial(link='logit'))
logitmodel <- modFit$finalModel
summary(logitmodel)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.7997  -0.3319  -0.1995  -0.1442   3.0897
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -3.672e+02  4.038e+01  -9.095  < 2e-16 ***
## age                  -7.555e-04  2.167e-03  -0.349 0.727429
## duration              4.517e-03  9.938e-05  45.453  < 2e-16 ***
## pdays                -9.286e-04  2.582e-04  -3.597 0.000322 ***
## poutcomenonexistent   5.599e-01  8.237e-02   6.797 1.07e-11 ***
## poutcomesuccess       1.044e+00  2.592e-01   4.028 5.62e-05 ***
## day_of_weekmon       -1.110e-01  8.755e-02  -1.268 0.204961
## day_of_weekthu        1.515e-01  8.440e-02   1.795 0.072701 .
## day_of_weektue        1.955e-01  8.692e-02   2.249 0.024521 *
## day_of_weekwed        2.691e-01  8.639e-02   3.115 0.001841 **
## monthaug              1.064e+00  1.526e-01   6.974 3.07e-12 ***
## monthdec              6.573e-01  2.572e-01   2.556 0.010590 *
## monthjul              4.721e-02  1.260e-01   0.375 0.707979
## monthjun             -8.198e-01  1.621e-01  -5.057 4.26e-07 ***
## monthmar              2.330e+00  1.729e-01  13.479  < 2e-16 ***
## monthmay             -3.521e-01  1.045e-01  -3.371 0.000750 ***
## monthnov             -1.367e-01  1.224e-01  -1.117 0.264056
## monthoct              6.052e-01  1.596e-01   3.792 0.000150 ***
## monthsep              8.402e-01  1.997e-01   4.207 2.59e-05 ***
## contacttelephone     -8.129e-01  1.010e-01  -8.051 8.22e-16 ***
## emp.var.rate         -2.075e+00  1.814e-01 -11.440  < 2e-16 ***
## cons.price.idx        3.028e+00  2.908e-01  10.416  < 2e-16 ***
## cons.conf.idx         3.900e-02  7.171e-03   5.439 5.37e-08 ***
## nr.employed           1.587e-02  2.632e-03   6.031 1.63e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16209.1  on 21334  degrees of freedom
## Residual deviance:  9797.9  on 21311  degrees of freedom
```

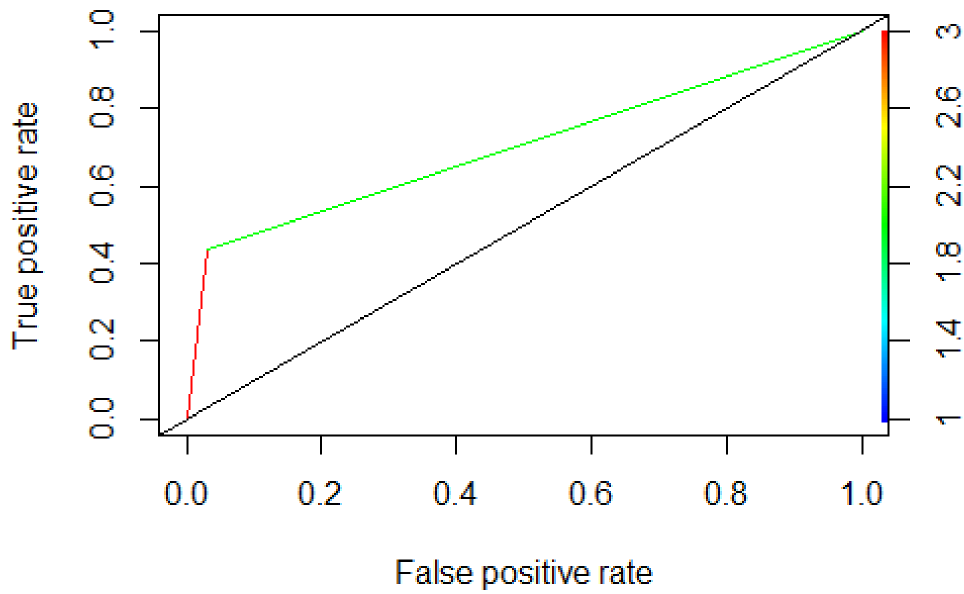
```
## AIC: 9845.9
##
## Number of Fisher Scoring iterations: 6
```

I used initally all the predictors to train my mdel. Then I found the below predictors are only having impact over my response through coefficient section Pr value. So, I modified my model by using following predictors. age+duration+pdays+poutcome+day_of_week+month+contact+emp.var.rate+cons.price.idx+cons.conf.idx+nr.employed

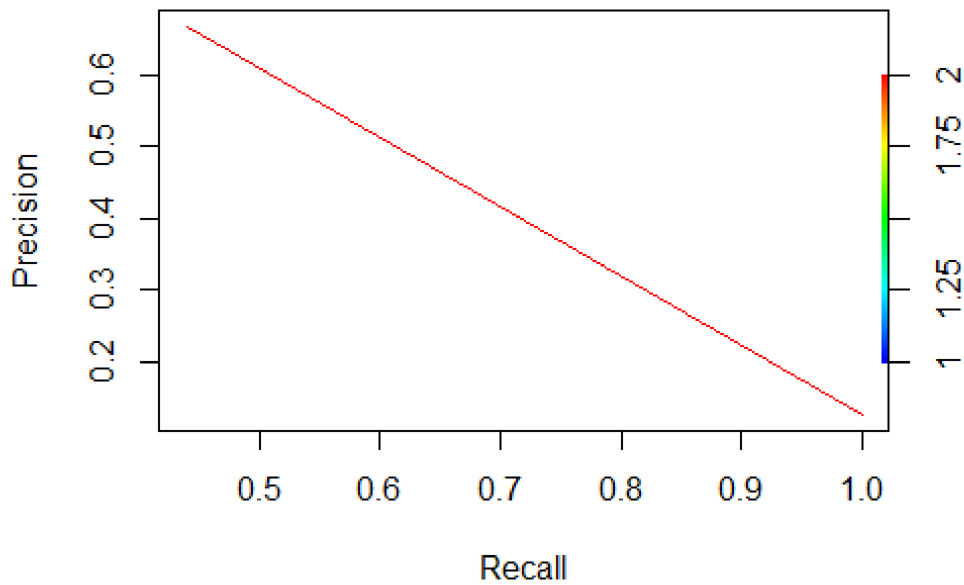
```
#Predict model
predictions <- predict(modFit, newdata = test, type="raw", na.action=na.pass)

#Check accuracy
confusionMatrix(predictions,test$y )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7732  649
##          1  254  508
##
##                Accuracy : 0.9012
##                  95% CI : (0.8949, 0.9073)
##     No Information Rate : 0.8735
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4769
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9682
##             Specificity : 0.4391
##          Pos Pred Value : 0.9226
##          Neg Pred Value : 0.6667
##              Prevalence : 0.8735
##          Detection Rate : 0.8457
##    Detection Prevalence : 0.9167
##       Balanced Accuracy : 0.7036
##
##        'Positive' Class : 0
##
```

```
ROCRpred <- prediction(as.numeric(predictions), as.numeric(test$y))
### ROC Curve
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
abline(0, 1)
```




```
### AUC
auc_ROCR <- performance(ROCRpred, measure = "auc")
auc_ROCR <- auc_ROCR@y.values[[1]]
print(auc_ROCR)

## [1] 0.7036304

### Precision/Recall
RPperf <- performance(ROCRpred, "prec", "rec");
plot(RPperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```




```
### Precision/Recall
print(RPperf)

## An object of class "performance"
## Slot "x.name":
## [1] "Recall"
##
## Slot "y.name":
## [1] "Precision"
##
## Slot "alpha.name":
## [1] "Cutoff"
##
## Slot "x.values":
## [[1]]
## [1] 0.0000000 0.4390666 1.0000000
##
##
## Slot "y.values":
## [[1]]
## [1]       NaN 0.6666667 0.1265449
##
##
## Slot "alpha.values":
## [[1]]
## [1] Inf   2   1
```

c. Linear Discriminant Analysis (5 points)

```
### Fit model
modFit <- train(y
~age_group+duration+pdays+poutcome+day_of_week+month+contact+emp.var.rate+con
s.price.idx+cons.conf.idx+nr.employed, method='lda',data=train,trControl =
cvctrl)
ldamodel <- modFit$finalModel
ldamodel
```

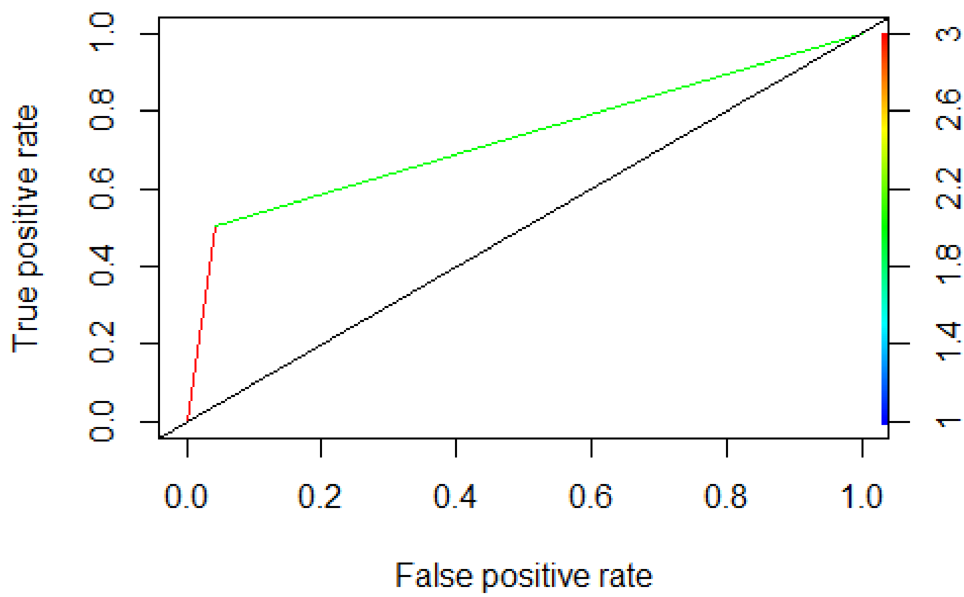
```
## Call:
## lda(x, grouping = y)
##
## Prior probabilities of groups:
##         0         1
## 0.8734005 0.1265995
##
## Group means:
##   age_groupMiddle Aged age_groupSenior Citizens age_groupTeens duration
## 0            0.5797467               0.01964152   0.0008586455 220.7441
## 1            0.4557571               0.09848204   0.0040725657 526.9841
##      pdays poutcomenonexistent poutcomesuccess day_of_weekmon
## 0 981.5758           0.8736718      0.01524096      0.2118171
## 1 785.3384           0.6675305      0.19918549      0.1795631
##   day_of_weekthu day_of_weektue day_of_weekwed  monthaug    monthdec
## 0      0.2067189      0.1899216      0.2000644 0.1564345 0.002790598
## 1      0.2302851      0.2065902      0.2095520 0.1384672 0.019992595
##    monthjul  monthjun   monthmar  monthmay   monthnov   monthoct
## 0 0.1691532 0.1194591 0.00933777 0.3416872 0.11650746 0.01287968
## 1 0.1299519 0.1147723 0.06256942 0.1884487 0.09885228 0.07256572
##     monthsep contacttelephone emp.var.rate cons.price.idx cons.conf.idx
## 0 0.01014275        0.3556402    0.1073092       93.54823     -40.71796
## 1 0.05997779        0.1469826   -1.3654202       93.32180     -39.79944
##   nr.employed
## 0    5170.984
## 1    5088.888
##
## Coefficients of linear discriminants:
##                                    LD1
## age_groupMiddle Aged     -0.0779512220
## age_groupSenior Citizens  0.2324223346
## age_groupTeens            0.1438754467
## duration                  0.0029432742
## pdays                    -0.0009911046
## poutcomenonexistent       0.3432677496
## poutcomesuccess           1.1502173611
## day_of_weekmon           -0.0813893303
## day_of_weekthu            0.0387303064
## day_of_weektue            0.0563580408
## day_of_weekwed            0.0782603711
## monthaug                  0.8514626476
```

```
## monthdec                     0.8348747593
## monthjul                     0.1727803957
## monthjun                    -0.6074612716
## monthmar                     2.0025758927
## monthmay                    -0.1252279229
## monthnov                     0.1115791800
## monthoct                     0.5018958305
## monthsep                     0.6055062898
## contacttelephone            -0.4008395974
## emp.var.rate                -1.3234272172
## cons.price.idx               2.1363544111
## cons.conf.idx                0.0442832737
## nr.employed                  0.0098839029
```

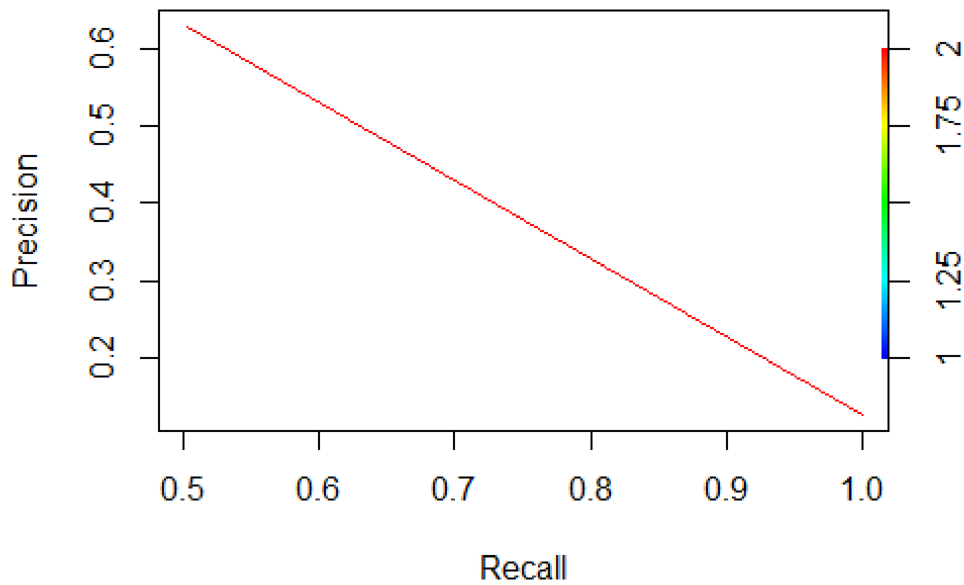
```
### I used initally all the predictors to train my mdel. Then I found the
below predictors are only having impact over my response through coefficient
section Pr value. So, I modified my model by using following predictors.
age_group+duration+pdays+poutcome+day_of_week+month+contact+emp.var.rate+cons
.price.idx+cons.conf.idx

### Predict model
predictions <- predict(modFit, newdata = test)

### Check accuracy
confusionMatrix(predictions,test$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7642  576
##          1  344  581
##
##                Accuracy : 0.8994
##                  95% CI : (0.893, 0.9055)
##     No Information Rate : 0.8735
##     P-Value [Acc > NIR] : 8.323e-15
##
##                   Kappa : 0.5021
##  Mcnemar's Test P-Value : 2.620e-14
##
##             Sensitivity : 0.9569
##             Specificity : 0.5022
##          Pos Pred Value : 0.9299
##          Neg Pred Value : 0.6281
##              Prevalence : 0.8735
##          Detection Rate : 0.8358
##    Detection Prevalence : 0.8988
##       Balanced Accuracy : 0.7295
##
```

```
##        'Positive' Class : 0
##

library(ROCR)
ROCRpred <- prediction(as.numeric(predictions), as.numeric(test$y))
### ROC Curve
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
abline(0, 1)
```




```
### AUC
auc_ROCR <- performance(ROCRpred, measure = "auc")
auc_ROCR <- auc_ROCR@y.values[[1]]
print(auc_ROCR)

## [1] 0.7295427

### Precision/Recall
RPperf <- performance(ROCRpred, "prec", "rec");
plot(RPperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```




```
### Precision/Recall
print(RPperf)

## An object of class "performance"
## Slot "x.name":
## [1] "Recall"
##
## Slot "y.name":
## [1] "Precision"
##
## Slot "alpha.name":
## [1] "Cutoff"
##
## Slot "x.values":
## [[1]]
## [1] 0.0000000 0.5021608 1.0000000
##
##
## Slot "y.values":
## [[1]]
## [1]       NaN 0.6281081 0.1265449
##
##
## Slot "alpha.values":
## [[1]]
## [1] Inf   2   1
```

12. For the logistic regression model did you find any predictor variables to be not significant? Based on what metric did you decide they were not significant? Report that metric. If they were all significant then, similarly, indicate the metric you used to make this decision and report the metric. (3 points)

Since age is not significant coefficient based on Pr value under coefficient section. I tried removing age and keep other significant predictors.

Similarly, I removed education, loan, housing, job also based on Pr value from coefficient section.

```
### Fit model
modFit <- train(y
~duration+pdays+poutcome+day_of_week+month+contact+emp.var.rate+cons.price.id
x+cons.conf.idx+nr.employed, method='glm', trControl = cvctrl, data=train,
family=binomial(link='logit'))
logitmodel <- modFit$finalModel
summary(logitmodel)

##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.7989  -0.3320  -0.1994  -0.1444   3.0914
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -3.666e+02  4.034e+01  -9.089  < 2e-16 ***
## duration              4.517e-03  9.938e-05  45.451  < 2e-16 ***
## pdays                -9.310e-04  2.581e-04  -3.607 0.000309 ***
## poutcomenonexistent   5.604e-01  8.236e-02   6.805 1.01e-11 ***
## poutcomesuccess       1.042e+00  2.591e-01   4.022 5.78e-05 ***
## day_of_weekmon       -1.113e-01  8.754e-02  -1.272 0.203480
## day_of_weekthu        1.517e-01  8.440e-02   1.797 0.072305 .
## day_of_weektue        1.950e-01  8.691e-02   2.244 0.024820 *
## day_of_weekwed        2.691e-01  8.639e-02   3.115 0.001839 **
## monthaug              1.064e+00  1.526e-01   6.972 3.12e-12 ***
## monthdec              6.539e-01  2.570e-01   2.544 0.010957 *
## monthjul              4.959e-02  1.259e-01   0.394 0.693624
## monthjun             -8.151e-01  1.616e-01  -5.045 4.52e-07 ***
## monthmar              2.329e+00  1.728e-01  13.475  < 2e-16 ***
## monthmay             -3.504e-01  1.044e-01  -3.358 0.000785 ***
## monthnov             -1.363e-01  1.224e-01  -1.113 0.265583
## monthoct              6.044e-01  1.596e-01   3.787 0.000153 ***
```

```
## monthsep             8.396e-01  1.997e-01   4.204 2.62e-05 ***
## contacttelephone    -8.121e-01  1.009e-01  -8.046 8.56e-16 ***
## emp.var.rate        -2.072e+00  1.812e-01 -11.437  < 2e-16 ***
## cons.price.idx       3.023e+00  2.904e-01  10.412  < 2e-16 ***
## cons.conf.idx        3.877e-02  7.140e-03   5.430 5.62e-08 ***
## nr.employed          1.584e-02  2.630e-03   6.022 1.73e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16209  on 21334  degrees of freedom
## Residual deviance:  9798  on 21312  degrees of freedom
## AIC: 9844
##
## Number of Fisher Scoring iterations: 6
```

```
### Predict model
predictions <- predict(modFit, newdata = test, type="raw", na.action=na.pass)
```

```
### Check accuracy
confusionMatrix(predictions,test$y )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7734  649
##          1  252  508
##
##                Accuracy : 0.9015
##                  95% CI : (0.8952, 0.9075)
##     No Information Rate : 0.8735
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4776
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9684
##             Specificity : 0.4391
##          Pos Pred Value : 0.9226
##          Neg Pred Value : 0.6684
##              Prevalence : 0.8735
##          Detection Rate : 0.8459
##    Detection Prevalence : 0.9169
##       Balanced Accuracy : 0.7038
##
##        'Positive' Class : 0
##
```

13. Use the following transformation/interaction of predictors in each of the above models and build additional models. Comment on whether any interactions were statistically significant. (5 points)

a. A log transformation

```
bank_distinct_log <- bank_distinct
bank_distinct_log$lnage <- log(bank_distinct$age)
summary(bank_distinct_log)
```

```
##       age                 job            marital
##  Min.   :17.00   admin.     :8734   divorced: 3552
##  1st Qu.:31.00   blue-collar:5674   married :17487
##  Median :37.00   technician :5469   single  : 9439
##  Mean   :39.03   services   :2856   unknown :    0
##  3rd Qu.:45.00   management :2311
##  Max.   :95.00   retired    :1215
##                  (Other)    :4219
##                education        default         housing
##  university.degree  :10408   no     :30475   no     :13962
##  high.school        : 7697   unknown:    0   unknown:    0
##  professional.course: 4318   yes    :    3   yes    :16516
##  basic.9y           : 4276
##  basic.4y           : 2380
##  basic.6y           : 1388
##  (Other)            :   11
##       loan            contact          month       day_of_week
##  no     :25710   cellular :20435   may    :9731   fri:5733
##  unknown:    0   telephone:10043   jul    :5077   mon:6278
##  yes    : 4768                     aug    :4672   thu:6391
##                                    jun    :3614   tue:5951
##                                    nov    :3495   wed:6125
##                                    apr    :2114
##                                    (Other):1775
##     duration         campaign          pdays          previous
##  Min.   :   0.0   Min.   : 1.000   Min.   :  0.0   Min.   :0.0000
##  1st Qu.: 103.0   1st Qu.: 1.000   1st Qu.:999.0   1st Qu.:0.0000
##  Median : 181.0   Median : 2.000   Median :999.0   Median :0.0000
##  Mean   : 259.5   Mean   : 2.522   Mean   :956.3   Mean   :0.1943
##  3rd Qu.: 321.0   3rd Qu.: 3.000   3rd Qu.:999.0   3rd Qu.:0.0000
##  Max.   :4918.0   Max.   :43.000   Max.   :999.0   Max.   :7.0000
##
##         poutcome      emp.var.rate      cons.price.idx  cons.conf.idx
##  failure    : 3461   Min.   :-3.40000   Min.   :92.20   Min.   :-50.8
##  nonexistent:25826   1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7
##  success    : 1191   Median : 1.10000   Median :93.44   Median :-41.8
##                      Mean   :-0.07143   Mean   :93.52   Mean   :-40.6
##                      3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4
##                      Max.   : 1.40000   Max.   :94.77   Max.   :-26.9
```

```
##
##    euribor3m      nr.employed   y                   age_group
##  Min.   :0.634   Min.   :4964   0:26620   Adults         :12287
##  1st Qu.:1.313   1st Qu.:5099   1: 3858   Middle Aged    :17272
##  Median :4.856   Median :5191             Senior Citizens:  881
##  Mean   :3.460   Mean   :5161             Teens          :   38
##  3rd Qu.:4.961   3rd Qu.:5228
##  Max.   :5.045   Max.   :5228
##
##      lnage
##  Min.   :2.833
##  1st Qu.:3.434
##  Median :3.611
##  Mean   :3.632
##  3rd Qu.:3.807
##  Max.   :4.554
##

### Yes. I am seeing slight increase in Sensitivity and Accuracy values of
LDA, Logistic models.

set.seed(12345)
intrainlog <- createDataPartition(bank_distinct_log$y,p=0.70,list = FALSE)
trainlog <- bank_distinct_log[intrainlog,]
testlog <- bank_distinct_log[-intrainlog,]
#table(trainlog$y)
#table(testlog$y)
cvctrl <- trainControl(method = "cv", number=10)
### a.  Decision tree (5 points)

modFitlog <- train(y ~.-age, method='rpart', trControl = cvctrl,
data=trainlog)
decisiontreemodellog <- modFitlog$finalModel
print(modFitlog$finalModel)

## n= 21335
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 21335 2701 0 (0.87340052 0.12659948)
##    2) nr.employed>=5087.65 18286 1351 0 (0.92611834 0.07388166)
##      4) duration< 524.5 16289  503 0 (0.96912027 0.03087973) *
##      5) duration>=524.5 1997  848 0 (0.57536304 0.42463696)
##       10) duration< 834.5 1280  433 0 (0.66171875 0.33828125) *
##       11) duration>=834.5 717  302 1 (0.42119944 0.57880056) *
##    3) nr.employed< 5087.65 3049 1350 0 (0.55723188 0.44276812)
##      6) duration< 158.5 1058  158 0 (0.85066163 0.14933837) *
##      7) duration>=158.5 1991  799 1 (0.40130588 0.59869412) *

predictionslog <- predict(modFitlog, newdata = testlog, type='raw')
```

```r
confusionMatrix(predictionslog,testlog$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7509  462
##          1  477  695
##
##                Accuracy : 0.8973
##                  95% CI : (0.8909, 0.9034)
##     No Information Rate : 0.8735
##     P-Value [Acc > NIR] : 9.749e-13
##
##                   Kappa : 0.538
##  Mcnemar's Test P-Value : 0.6478
##
##             Sensitivity : 0.9403
##             Specificity : 0.6007
##          Pos Pred Value : 0.9420
##          Neg Pred Value : 0.5930
##              Prevalence : 0.8735
##          Detection Rate : 0.8213
##    Detection Prevalence : 0.8718
##       Balanced Accuracy : 0.7705
##
##        'Positive' Class : 0
##
```

b. Logistic regression (5 points)

```r
modFitlog <- train(y
~lnage+duration+pdays+poutcome+day_of_week+month+contact+emp.var.rate+cons.pr
ice.idx+cons.conf.idx, method='glm', trControl = cvctrl, data=trainlog,
family=binomial(link='logit'))
logitmodellog <- modFitlog$finalModel
summary(logitmodellog)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.8351  -0.3356  -0.1993  -0.1443   3.0162
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.275e+02  6.791e+00 -18.777  < 2e-16 ***
## lnage                -9.206e-02  9.174e-02  -1.003 0.315654
## duration              4.516e-03  9.938e-05  45.447  < 2e-16 ***
```

```
## pdays                -9.735e-04  2.556e-04  -3.808 0.000140 ***
## poutcomenonexistent  5.733e-01  8.203e-02   6.988 2.78e-12 ***
## poutcomesuccess      9.835e-01  2.564e-01   3.836 0.000125 ***
## day_of_weekmon      -1.243e-01  8.730e-02  -1.424 0.154401
## day_of_weekthu       1.386e-01  8.411e-02   1.648 0.099389 .
## day_of_weektue       1.805e-01  8.667e-02   2.083 0.037258 *
## day_of_weekwed       2.492e-01  8.613e-02   2.894 0.003809 **
## monthaug             6.226e-01  1.340e-01   4.648 3.35e-06 ***
## monthdec             3.582e-01  2.514e-01   1.425 0.154209
## monthjul             1.524e-01  1.238e-01   1.232 0.218093
## monthjun            -1.620e-01  1.182e-01  -1.371 0.170433
## monthmar             1.756e+00  1.461e-01  12.024  < 2e-16 ***
## monthmay            -5.814e-01  9.714e-02  -5.986 2.16e-09 ***
## monthnov            -1.496e-01  1.224e-01  -1.222 0.221690
## monthoct             3.157e-01  1.553e-01   2.032 0.042113 *
## monthsep             1.739e-01  1.674e-01   1.039 0.298759
## contacttelephone    -5.851e-01  8.984e-02  -6.513 7.39e-11 ***
## emp.var.rate        -9.989e-01  3.075e-02 -32.481  < 2e-16 ***
## cons.price.idx       1.339e+00  7.285e-02  18.377  < 2e-16 ***
## cons.conf.idx        2.314e-02  6.585e-03   3.514 0.000441 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16209.1  on 21334  degrees of freedom
## Residual deviance:  9833.2  on 21312  degrees of freedom
## AIC: 9879.2
##
## Number of Fisher Scoring iterations: 6
```

```
predictionslog <- predict(modFitlog, newdata = testlog, type="raw",
na.action=na.pass)
```

```
confusionMatrix(predictionslog,testlog$y )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7734  655
##          1  252  502
##
##                Accuracy : 0.9008
##                  95% CI : (0.8945, 0.9069)
##     No Information Rate : 0.8735
##     P-Value [Acc > NIR] : 2.497e-16
##
##                   Kappa : 0.4727
##  Mcnemar's Test P-Value : < 2.2e-16
```

```
##
##             Sensitivity : 0.9684
##             Specificity : 0.4339
##          Pos Pred Value : 0.9219
##          Neg Pred Value : 0.6658
##              Prevalence : 0.8735
##          Detection Rate : 0.8459
##    Detection Prevalence : 0.9175
##       Balanced Accuracy : 0.7012
##
##        'Positive' Class : 0
##
```

c. Linear Discriminant Analysis (5 points)

```
modFitlog <- train(y
~lnage+duration+pdays+poutcome+day_of_week+month+contact+emp.var.rate+cons.pr
ice.idx+cons.conf.idx, method='lda',data=trainlog,trControl = cvctrl)
ldamodellog <- modFitlog$finalModel
ldamodellog
```

```
## Call:
## lda(x, grouping = y)
##
## Prior probabilities of groups:
##         0         1
## 0.8734005 0.1265995
##
## Group means:
##      lnage duration    pdays poutcomenonexistent poutcomesuccess
## 0 3.628832 220.7441 981.5758           0.8736718      0.01524096
## 1 3.645820 526.9841 785.3384           0.6675305      0.19918549
##   day_of_weekmon day_of_weekthu day_of_weektue day_of_weekwed  monthaug
## 0      0.2118171      0.2067189      0.1899216      0.2000644 0.1564345
## 1      0.1795631      0.2302851      0.2065902      0.2095520 0.1384672
##      monthdec  monthjul  monthjun   monthmar  monthmay   monthnov
## 0 0.002790598 0.1691532 0.1194591 0.00933777 0.3416872 0.11650746
## 1 0.019992595 0.1299519 0.1147723 0.06256942 0.1884487 0.09885228
##     monthoct   monthsep contacttelephone emp.var.rate cons.price.idx
## 0 0.01287968 0.01014275        0.3556402    0.1073092       93.54823
## 1 0.07256572 0.05997779        0.1469826   -1.3654202       93.32180
##   cons.conf.idx
## 0     -40.71796
## 1     -39.79944
##
## Coefficients of linear discriminants:
##                                LD1
## lnage                 -0.024653247
## duration               0.002953409
## pdays                 -0.001024927
## poutcomenonexistent    0.353556676
```

```
## poutcomesuccess      1.110628254
## day_of_weekmon      -0.090367073
## day_of_weekthu       0.034659061
## day_of_weektue       0.047491482
## day_of_weekwed       0.073959980
## monthaug             0.466241234
## monthdec             0.621197731
## monthjul             0.186299241
## monthjun            -0.180578882
## monthmar             1.668480822
## monthmay            -0.281710958
## monthnov             0.074572656
## monthoct             0.335458390
## monthsep             0.136146376
## contacttelephone    -0.308217849
## emp.var.rate        -0.665964585
## cons.price.idx       1.147528731
## cons.conf.idx        0.040255088
```

```r
predictionslog <- predict(modFitlog, newdata = testlog)
```

```r
confusionMatrix(predictionslog,testlog$y)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    0    1
##          0 7648  575
##          1  338  582
##                                           
##                Accuracy : 0.9001          
##                  95% CI : (0.8938, 0.9062)
##     No Information Rate : 0.8735          
##     P-Value [Acc > NIR] : 1.292e-15       
##                                           
##                   Kappa : 0.5049          
##  Mcnemar's Test P-Value : 5.698e-15       
##                                           
##             Sensitivity : 0.9577          
##             Specificity : 0.5030          
##          Pos Pred Value : 0.9301          
##          Neg Pred Value : 0.6326          
##              Prevalence : 0.8735          
##          Detection Rate : 0.8365          
##    Detection Prevalence : 0.8994          
##       Balanced Accuracy : 0.7304          
##                                           
##        'Positive' Class : 0               
## 
```

b. An interaction of two variables

```
### Tried few combinations of predictors. Ex:poutcome:duration. But it didn't
give any impact in results.
```

a. Decission Tree Model

```
bank_distinct_interaction <- bank_distinct_log

### Yes. I am seeing slight increase in Sensitivity and Accuracy values of
LDA, Logistic models.

set.seed(12345)
intraininteraction <-
createDataPartition(bank_distinct_interaction$y,p=0.70,list = FALSE)
traininteraction <- bank_distinct_interaction[intraininteraction,]
testinteraction <- bank_distinct_interaction[-intraininteraction,]
#table(trainlog$y)
#table(testlog$y)
cvctrl <- trainControl(method = "cv", number=10)
### a.  Decision tree (5 points)

modFitinteraction <- train(y ~.-age, method='rpart', trControl = cvctrl,
data=traininteraction)
decisiontreemodelinteraction <- modFitinteraction$finalModel
print(modFitinteraction$finalModel)

## n= 21335
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 21335 2701 0 (0.87340052 0.12659948)
##    2) nr.employed>=5087.65 18286 1351 0 (0.92611834 0.07388166)
##      4) duration< 524.5 16289  503 0 (0.96912027 0.03087973) *
##      5) duration>=524.5 1997  848 0 (0.57536304 0.42463696)
##       10) duration< 834.5 1280  433 0 (0.66171875 0.33828125) *
##       11) duration>=834.5 717  302 1 (0.42119944 0.57880056) *
##    3) nr.employed< 5087.65 3049 1350 0 (0.55723188 0.44276812)
##      6) duration< 158.5 1058  158 0 (0.85066163 0.14933837) *
##      7) duration>=158.5 1991  799 1 (0.40130588 0.59869412) *

predictionsinteraction <- predict(modFitinteraction, newdata =
testinteraction, type='raw')

confusionMatrix(predictionsinteraction,testinteraction$y)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7509  462
##          1  477  695
```

```
##
##                Accuracy : 0.8973
##                  95% CI : (0.8909, 0.9034)
##     No Information Rate : 0.8735
##     P-Value [Acc > NIR] : 9.749e-13
##
##                   Kappa : 0.538
##  Mcnemar's Test P-Value : 0.6478
##
##             Sensitivity : 0.9403
##             Specificity : 0.6007
##          Pos Pred Value : 0.9420
##          Neg Pred Value : 0.5930
##              Prevalence : 0.8735
##          Detection Rate : 0.8213
##    Detection Prevalence : 0.8718
##       Balanced Accuracy : 0.7705
##
##        'Positive' Class : 0
##
```

b. Logistic regression (5 points)

```
modFitinteraction <- train(y
~lnage+duration+pdays+poutcome+day_of_week+month+contact+emp.var.rate+cons.pr
ice.idx+cons.conf.idx+poutcome:duration, method='glm', trControl = cvctrl,
data=traininteraction, family=binomial(link='logit'))
logitmodelinteraction <- modFitinteraction$finalModel
summary(logitmodelinteraction)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.8925  -0.3357  -0.1979  -0.1426   3.0224
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.274e+02  6.782e+00 -18.780  < 2e-16 ***
## lnage                        -8.746e-02  9.166e-02  -0.954 0.339961
## duration                      4.189e-03  2.672e-04  15.678  < 2e-16 ***
## pdays                        -9.537e-04  2.543e-04  -3.750 0.000177 ***
## poutcomenonexistent           4.353e-01  1.273e-01   3.420 0.000625 ***
## poutcomesuccess               1.085e+00  2.980e-01   3.640 0.000273 ***
## day_of_weekmon               -1.256e-01  8.725e-02  -1.440 0.150002
## day_of_weekthu                1.367e-01  8.410e-02   1.625 0.104100
## day_of_weektue                1.786e-01  8.667e-02   2.061 0.039343 *
## day_of_weekwed                2.476e-01  8.613e-02   2.875 0.004035 **
## monthaug                      6.271e-01  1.338e-01   4.687 2.77e-06 ***
```

```
## monthdec                          3.636e-01  2.507e-01   1.450 0.147001    
## monthjul                          1.511e-01  1.237e-01   1.221 0.221970    
## monthjun                         -1.635e-01  1.181e-01  -1.385 0.166111    
## monthmar                          1.753e+00  1.460e-01  12.008  < 2e-16 ***
## monthmay                         -5.782e-01  9.692e-02  -5.966 2.44e-09 ***
## monthnov                         -1.469e-01  1.223e-01  -1.202 0.229477    
## monthoct                          3.189e-01  1.551e-01   2.057 0.039722 *  
## monthsep                          1.818e-01  1.670e-01   1.089 0.276168    
## contacttelephone                 -5.831e-01  8.981e-02  -6.492 8.46e-11 ***
## emp.var.rate                     -1.002e+00  3.084e-02 -32.498  < 2e-16 ***
## cons.price.idx                    1.337e+00  7.272e-02  18.389  < 2e-16 ***
## cons.conf.idx                     2.180e-02  6.610e-03   3.299 0.000972 ***
## `duration:poutcomenonexistent`   4.013e-04  2.853e-04   1.407 0.159496    
## `duration:poutcomesuccess`      -4.181e-04  5.844e-04  -0.715 0.474384    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 16209.1  on 21334  degrees of freedom
## Residual deviance:  9829.3  on 21310  degrees of freedom
## AIC: 9879.3
## 
## Number of Fisher Scoring iterations: 6
```

```
predictionsinteraction <- predict(modFitinteraction, newdata =
testinteraction, type="raw", na.action=na.pass)
```

```
confusionMatrix(predictionsinteraction,testinteraction$y )
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    0    1
##          0 7729  652
##          1  257  505
##                                           
##                Accuracy : 0.9006          
##                  95% CI : (0.8943, 0.9066)
##     No Information Rate : 0.8735          
##     P-Value [Acc > NIR] : 4.34e-16        
##                                           
##                   Kappa : 0.4734          
##  Mcnemar's Test P-Value : < 2.2e-16       
##                                           
##             Sensitivity : 0.9678          
##             Specificity : 0.4365          
##          Pos Pred Value : 0.9222          
##          Neg Pred Value : 0.6627          
##              Prevalence : 0.8735          
```

```
##          Detection Rate : 0.8453
##    Detection Prevalence : 0.9167
##       Balanced Accuracy : 0.7021
##
##        'Positive' Class : 0
##
```

c. Linear Discriminant Analysis (5 points)

```
modFitinteraction <- train(y
~lnage+duration+pdays+poutcome+day_of_week+month+contact+emp.var.rate+cons.pr
ice.idx+cons.conf.idx+poutcome:duration,
method='lda',data=traininteraction,trControl = cvctrl)
ldamodellog <- modFitinteraction$finalModel
ldamodellog
```

```
## Call:
## lda(x, grouping = y)
##
## Prior probabilities of groups:
##         0         1
## 0.8734005 0.1265995
##
## Group means:
##      lnage duration    pdays poutcomenonexistent poutcomesuccess
## 0 3.628832 220.7441 981.5758           0.8736718      0.01524096
## 1 3.645820 526.9841 785.3384           0.6675305      0.19918549
##   day_of_weekmon day_of_weekthu day_of_weektue day_of_weekwed  monthaug
## 0      0.2118171      0.2067189      0.1899216      0.2000644 0.1564345
## 1      0.1795631      0.2302851      0.2065902      0.2095520 0.1384672
##      monthdec  monthjul  monthjun   monthmar  monthmay   monthnov
## 0 0.002790598 0.1691532 0.1194591 0.00933777 0.3416872 0.11650746
## 1 0.019992595 0.1299519 0.1147723 0.06256942 0.1884487 0.09885228
##     monthoct   monthsep contacttelephone emp.var.rate cons.price.idx
## 0 0.01287968 0.01014275        0.3556402    0.1073092       93.54823
## 1 0.07256572 0.05997779        0.1469826   -1.3654202       93.32180
##   cons.conf.idx duration:poutcomenonexistent duration:poutcomesuccess
## 0     -40.71796                     193.4575                 3.580605
## 1     -39.79944                     394.7231                70.594224
##
## Coefficients of linear discriminants:
##                                          LD1
## lnage                          -0.0254413124
## duration                        0.0033031789
## pdays                          -0.0010085274
## poutcomenonexistent             0.4480467357
## poutcomesuccess                 1.2784963916
## day_of_weekmon                 -0.0892571961
## day_of_weekthu                  0.0358048950
## day_of_weektue                  0.0491356133
## day_of_weekwed                  0.0743583979
```

```
## monthaug                        0.4699507159
## monthdec                        0.6300912255
## monthjul                        0.1908748983
## monthjun                       -0.1776289029
## monthmar                        1.6726502240
## monthmay                       -0.2773810410
## monthnov                        0.0793585983
## monthoct                        0.3369387732
## monthsep                        0.1366316111
## contacttelephone               -0.3081707038
## emp.var.rate                   -0.6659894103
## cons.price.idx                  1.1477653431
## cons.conf.idx                   0.0401478834
## duration:poutcomenonexistent -0.0003795441
## duration:poutcomesuccess     -0.0005556399
```

```
predictionsinteraction <- predict(modFitinteraction, newdata =
testinteraction)
```

```
confusionMatrix(predictionsinteraction,testinteraction$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7644  575
##          1  342  582
##
##                Accuracy : 0.8997
##                  95% CI : (0.8934, 0.9058)
##     No Information Rate : 0.8735
##     P-Value [Acc > NIR] : 3.773e-15
##
##                   Kappa : 0.5036
##  Mcnemar's Test P-Value : 1.840e-14
##
##             Sensitivity : 0.9572
##             Specificity : 0.5030
##          Pos Pred Value : 0.9300
##          Neg Pred Value : 0.6299
##              Prevalence : 0.8735
##          Detection Rate : 0.8360
##    Detection Prevalence : 0.8989
##       Balanced Accuracy : 0.7301
##
##        'Positive' Class : 0
##
```

14. Comment on the pros and cons of each of the models above. Based on your data and your exploratory data analysis do you feel that one model might fit better than another? If so why? (3 points)

Though accuracy statistics shows that Logistic regression (0.9008) model is better than other two models, Specificity value is coming as 0.4339 which is lower than other two models.

In the other hand Decision tree model is giving specificity stat as 0.6007.

At the same time, LDA is giving high Sensitivity value as 0.957.

When we compare AUC value of ROC, we are able to see that Decision tree value is far better than other two models. 0.770481.

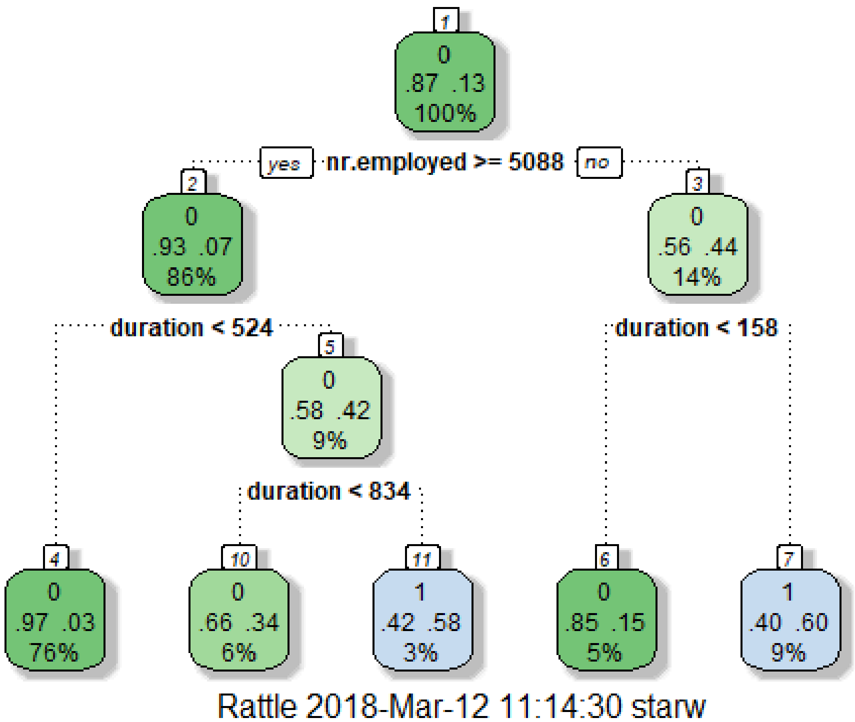
Meanwhile, Recall and Precision values are comparatively higher in Decision tree and LDA in order. 0.6006914, 0.5930034.

Based on the above observation, I believe that Decision Tree and LDA models are serving the purpose of predicting which customer will opt next midterm deposit (True Positive) more accurately than Logistic model in this case.

Because, even though Logistic model has higher accuracy, it's specificity and Recall values are lower than other models.

15. Plot the fitted decision tree. What attribute was used for the first split? (3 points)

```
library(rattle)

## Rattle: A free graphical interface for data science with R.
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

library(rpart.plot)

## Loading required package: rpart

library(RColorBrewer)
fancyRpartPlot(decisiontreemodel)
```




```
### observation:
### nr.employed is the attribute used for first split.
```

16. Report the following accuracy measures for the each model you fit above

a. Confusion matrix (2 points)

Decision Tree

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7509  462
##          1  477  695
```

Logistic Regression

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7734  655
##          1  252  502
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7648  575
##          1  338  582
```

LDA

b. Accuracy (2 points)

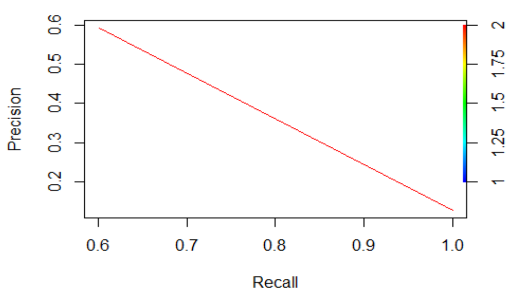
Decision Tree	Logistic Regression	LDA
Accuracy : 0.8973	Accuracy : 0.9008	Accuracy : 0.9001

c. Sensitivity/Specificity (2 points)

Decision Tree	Logistic Regression	LDA
Sensitivity : 0.9403	Sensitivity : 0.9684	Sensitivity : 0.9577
Specificity : 0.6007	Specificity : 0.4339	Specificity : 0.5030

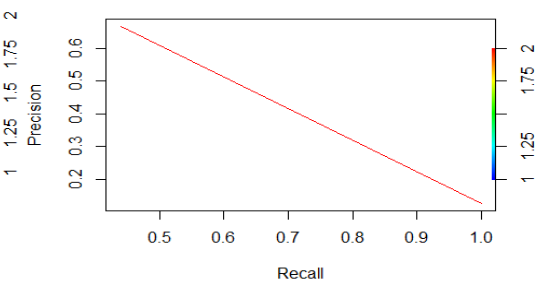
d. Precision/Recall (2 points)

Decision Tree

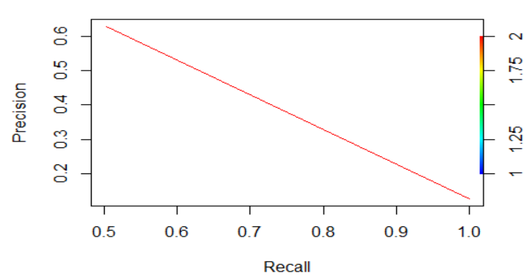


Recall → 0.6006914
Precision→0.5930034

Logistic Regression



Recall → 0.4390666
Precision→0.6666667

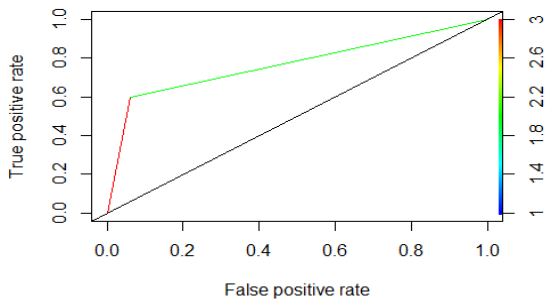


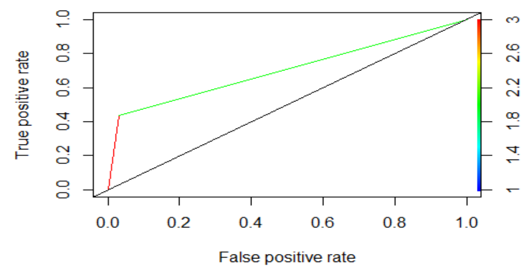
LDA

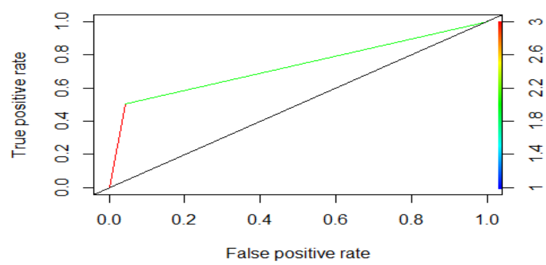
Recall → 0.5021608
Precision→0.6281081

e. ROC curve (2 points)

Decision Tree Logistic Regression







f. AUC (2 points)

Decision Tree → 0.770481

Logistic Regression → 0.7036304

LDA → 0.7295427

17. Comment on which accuracy measure is the appropriate one to use for your dataset. Based on the accuracy measure you picked which model gives you the best results? (2 points)

Though accuracy statistics shows that Logistic regression (0.9008) model is better than other two models, Specificity value is coming as 0.4339 which is lower than other two models.

In the other hand Decision tree model is giving specificity stat as 0.6007.

At the same time, LDA is giving high Sensitivity value as 0.957.

When we compare AUC value of ROC, we are able to see that Decision tree value is far better than other two models. 0.770481.

Meanwhile, Recall and Precision values are comparatively higher in Decision tree and LDA in order. 0.6006914, 0.5930034.

Based on the above observation, I believe that Decision Tree and LDA models are serving the purpose of predicting which customer will opt next midterm deposit (True Positive) more accurately than Logistic model in this case.

Because, even though Logistic model has higher accuracy, it's specificity and Recall values are lower than other models.