```
cov, smooth, var

> library(rpart)
> library(xgboost)

Attaching package: ‘xgboost’

The following object is masked from ‘package:plotly’:

    slice

The following object is masked from ‘package:dplyr’:

    slice

> getwd()
[1] "C:/Users/Asus/Documents"
> setwd("D:/Work/Gre/UTD/Courses/Elearning/Vcode/Marketin
g_Analytics")
> ## Reading the dataset
> bank_data<-read.csv("Bank Marketing dataset.csv")
> ########################### DATA EXPLORATION ##########
#################
> ## head of dataset
> head(bank_data)
  X age       job marital   education default housing loa
n   contact month day_of_week duration
1 1  56 housemaid married    basic.4y      no      no   n
o telephone   may         mon      261
2 2  57  services married high.school unknown      no   n
o telephone   may         mon      149
3 3  37  services married high.school      no     yes   n
o telephone   may         mon      226
4 4  40    admin. married    basic.6y      no      no   n
o telephone   may         mon      151
5 5  56  services married high.school      no      no  ye
s telephone   may         mon      307
6 6  45  services married    basic.9y unknown      no   n
o telephone   may         mon      198
  campaign pdays previous    poutcome emp.var.rate cons.p
rice.idx cons.conf.idx euribor3m
1        1   999        0 nonexistent          1.1
93.994         -36.4     4.857
2        1   999        0 nonexistent          1.1
93.994         -36.4     4.857
3        1   999        0 nonexistent          1.1
93.994         -36.4     4.857
4        1   999        0 nonexistent          1.1
93.994         -36.4     4.857
5        1   999        0 nonexistent          1.1
93.994         -36.4     4.857
6        1   999        0 nonexistent          1.1
93.994         -36.4     4.857
```

```
  nr.employed  y
1        5191 no
2        5191 no
3        5191 no
4        5191 no
5        5191 no
6        5191 no
> # refer to the meta data description
> bank_data <- subset(bank_data, select = -duration)
> ## string type of data
> str(bank_data)
'data.frame':	41188 obs. of  21 variables:
 $ X             : int  1 2 3 4 5 6 7 8 9 10 ...
 $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
 $ job           : chr  "housemaid" "services" "services"
"admin." ...
 $ marital       : chr  "married" "married" "married" "ma
rried" ...
 $ education     : chr  "basic.4y" "high.school" "high.sc
hool" "basic.6y" ...
 $ default       : chr  "no" "unknown" "no" "no" ...
 $ housing       : chr  "no" "no" "yes" "no" ...
 $ loan          : chr  "no" "no" "no" "no" ...
 $ contact       : chr  "telephone" "telephone" "telephon
e" "telephone" ...
 $ month         : chr  "may" "may" "may" "may" ...
 $ day_of_week   : chr  "mon" "mon" "mon" "mon" ...
 $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays         : int  999 999 999 999 999 999 999 999 9
99 999 ...
 $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome      : chr  "nonexistent" "nonexistent" "none
xistent" "nonexistent" ...
 $ 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             : chr  "no" "no" "no" "no" ...
> ## missing data
> colSums(is.na(bank_data)) %>% show()
             X            age            job         marit
al      education        default
             0              0              0
0              0              0
       housing           loan        contact           mon
th    day_of_week       campaign
             0              0              0
0           2059              0
         pdays       previous       poutcome   emp.var.ra
te cons.price.idx  cons.conf.idx
```

```
             0               0               0
0               0               0
     euribor3m    nr.employed              y
          4530               0               0
> ############################ DATA PRE-PROCESSING ######
#####################
> names(bank_data)
 [1] "X"              "age"            "job"            "
marital"         "education"
 [6] "default"        "housing"        "loan"           "
contact"         "month"
[11] "day_of_week"    "campaign"       "pdays"          "
previous"        "poutcome"
[16] "emp.var.rate"   "cons.price.idx" "cons.conf.idx"  "
euribor3m"       "nr.employed"
[21] "y"
> sum(is.na(bank_data$euribor3m))
[1] 4530
> # treating missing values in variable - euribor3m
> bank_data$euribor3m[is.na(bank_data$euribor3m)]<-mean(b
ank_data$euribor3m,na.rm=TRUE)
> sum(is.na(bank_data$euribor3m))
[1] 0
> # treating missing values in variable - day_of_week
> sum(is.na(bank_data$day_of_week))
[1] 2059
> bank_data$day_of_week[is.na(bank_data$day_of_week)]<-mo
de(bank_data$day_of_week)
> #Checking missing values
> sum(is.na(bank_data$day_of_week))
[1] 0
> ############################ EXPLORATORY DATA ANALYSIS
#########################
> ## Dimension of dataset
> dim(bank_data)
[1] 41188    21
> # checking % of target variable
> table(bank_data$y)/nrow(bank_data)*100

      no      yes
88.73458 11.26542
> ## summary of all columns
> summary(bank_data)
       X              age            job                 mar
ital          education
 Min.   :    1   Min.   :17.00   Length:41188       Lengt
h:41188        Length:41188
 1st Qu.:10298   1st Qu.:32.00   Class :character   Class
:character   Class :character
 Median :20595   Median :38.00   Mode  :character   Mode
:character   Mode  :character
 Mean   :20595   Mean   :40.02
 3rd Qu.:30891   3rd Qu.:47.00
```

```
 Max.   :41188   Max.   :98.00  
   default             housing              loan          
contact             month          
 Length:41188       Length:41188       Length:41188      
Length:41188       Length:41188      
 Class :character   Class :character   Class :character  
Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character  
Mode  :character   Mode  :character  
                                                          
                                                          
                                                          
 day_of_week           campaign          pdays          p
revious       poutcome        
 Length:41188       Min.   : 1.000   Min.   :  0.0   Min.   
:0.000   Length:41188      
 Class :character   1st Qu.: 1.000   1st Qu.:999.0   1st 
Qu.:0.000   Class :character  
 Mode  :character   Median : 2.000   Median :999.0   Medi
an :0.000   Mode  :character  
                    Mean   : 2.568   Mean   :962.5   Mean   
:0.173                     
                    3rd Qu.: 3.000   3rd Qu.:999.0   3rd 
Qu.:0.000                     
                    Max.   :56.000   Max.   :999.0   Max.   
:7.000                     
  emp.var.rate      cons.price.idx  cons.conf.idx     eur
ibor3m      nr.employed  
 Min.   :-3.40000   Min.   :92.20   Min.   :-50.8   Min.   
:0.634   Min.   :4964  
 1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7   1st Q
u.:1.405   1st Qu.:5099  
 Median : 1.10000   Median :93.75   Median :-41.8   Media
n :4.856   Median :5191  
 Mean   : 0.08189   Mean   :93.58   Mean   :-40.5   Mean   
:3.620   Mean   :5167  
 3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4   3rd Q
u.:4.961   3rd Qu.:5228  
 Max.   : 1.40000   Max.   :94.77   Max.   :-26.9   Max.   
:5.045   Max.   :5228  
      y            
 Length:41188      
 Class :character  
 Mode  :character  
                   
                   
                   
> bp <- barplot(table(bank_data$y),
+               beside=TRUE,
+               ylim=c(0, max(table(bank_data$y)) + 3452)
,
+               main="Term Deposit(yes/no) Distribution",
+               col = c("#eb8060", "#b9e38d"), border=0)
```

```
> text(bp, table(bank_data$y) + 1200, table(bank_data$y),
font=2, col="black")
> head(bank_data)
  X age       job marital   education default housing loa
n   contact month day_of_week campaign
1 1  56 housemaid married    basic.4y      no      no   n
o telephone   may         mon        1
2 2  57  services married high.school unknown      no   n
o telephone   may         mon        1
3 3  37  services married high.school      no     yes   n
o telephone   may         mon        1
4 4  40    admin. married    basic.6y      no      no   n
o telephone   may         mon        1
5 5  56  services married high.school      no      no  ye
s telephone   may         mon        1
6 6  45  services married    basic.9y unknown      no   n
o telephone   may         mon        1
  pdays previous    poutcome emp.var.rate cons.price.idx
cons.conf.idx euribor3m nr.employed  y
1   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
2   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
3   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
4   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
5   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
6   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
> ## Density plot for age column
> # Create a histogram
> hist(bank_data$age,
+      freq = TRUE,
+      xlab = "Age",
+      main = "Distribution of Age",
+      col = 'royal blue')
> ## Distribution of Term deposit across the age
> ggplot(bank_data, aes(x = age, fill = y)) +
+   geom_histogram(position = "identity", alpha = 0.4) +
+   labs(title = "Age and Term Deposit") +
+   theme(plot.title = element_text(hjust = 0.5))+guides(
fill=guide_legend(title="Term Deposit"))
`stat_bin()` using `bins = 30`. Pick better value with `b
inwidth`.
> ## Distribution of customer marital status by Term Depo
sit
> mar_counts <- bank_data %>%
+   count(Marital = factor(marital), Term_Deposit = facto
r(y)) %>%
+   mutate(pct = prop.table(n))
> mar_counts$pct<-round(mar_counts$pct,digits = 3)
```

```
> ggplot(mar_counts,aes(x = reorder(Marital,-pct), y = pc
t, fill = Term_Deposit, label = scales::percent(pct))) +
+   geom_col(position = 'dodge') +
+   geom_text(position = position_dodge(width = .9),    #
move to center of bars
+             vjust = -0.5,    # nudge above top of bar
+             size = 3) +
+   scale_y_continuous(labels = scales::percent) + theme(
axis.title.x=element_blank(),axis.text.x = element_text(a
ngle = 0)) + ggtitle("Marital Status v/s Term Deposit") +
ylab("% of Records") + theme(plot.title = element_text(hj
ust = 0.5)) + guides(fill=guide_legend(title="Term Deposi
t"))
> # Statistical test between marital status variable and
Term Deposit target variable
> chisq.test(bank_data$marital, bank_data$y, correct=FALS
E)

	Pearson's Chi-squared test

data:  bank_data$marital and bank_data$y
X-squared = 122.66, df = 3, p-value < 2.2e-16

> ## checking any relation in job and the term deposit
> job_counts<-as.data.frame(table(bank_data$job, bank_dat
a$y))
> job_counts<-job_counts %>%
+   pivot_wider(names_from=Var2, values_from=Freq)
> job_counts<-as.data.frame(job_counts)
> names(job_counts)<-c("Job Title","Term Deposit No","Ter
m Deposit Yes")
> job_counts$TD_No_Per<-round((job_counts$`Term Deposit N
o`/sum(job_counts$`Term Deposit No`))*100,2)
> job_counts$TD_Yes_Per<-round((job_counts$`Term Deposit
Yes`/sum(job_counts$`Term Deposit Yes`))*100,2)
> job_counts
       Job Title Term Deposit No Term Deposit Yes TD_No_P
er TD_Yes_Per
1         admin.            9070             1352     24.
82      29.14
2    blue-collar            8616              638     23.
57      13.75
3   entrepreneur            1332              124      3.
64       2.67
4      housemaid             954              106      2.
61       2.28
5     management            2596              328      7.
10       7.07
6        retired            1286              434      3.
52       9.35
7  self-employed            1272              149      3.
48       3.21
```

```
8       services              3646              323      9.
98        6.96
9         student              600              275      1.
64        5.93
10     technician             6013              730     16.
45       15.73
11     unemployed              870              144      2.
38        3.10
12        unknown              293               37      0.
80        0.80
> ## Distribution of Job variable
> library(dplyr)
> JB_counts <- bank_data %>%
+   count(Job = factor(job)) %>%
+   mutate(pct = prop.table(n))
> JB_counts$pct<-round(JB_counts$pct,digits = 3)
> ggplot(JB_counts,aes(x = reorder(Job,-pct), y = pct, fi
ll = Job, label = scales::percent(pct))) +
+   geom_col(position = 'dodge') +
+   geom_text(position = position_dodge(width = .9),    #
move to center of bars
+             vjust = -0.5,    # nudge above top of bar
+             size = 3) +
+   scale_y_continuous(labels = scales::percent) + theme(
axis.title.x=element_blank(),axis.text.x = element_text(a
ngle = 90),legend.position="none") + ggtitle("Distributio
n of Job variable") + ylab("% of Records") + theme(plot.t
itle = element_text(hjust = 0.5))
> # Statistical test between Job variable and Term Deposi
t target variable
> chisq.test(bank_data$job, bank_data$y, correct=FALSE)

      Pearson's Chi-squared test

data:  bank_data$job and bank_data$y
X-squared = 961.24, df = 11, p-value < 2.2e-16

> ## Distribution of education variable
> ed_counts <- bank_data %>%
+   count(Education = factor(education)) %>%
+   mutate(pct = prop.table(n))
> ed_counts$pct<-round(ed_counts$pct,digits = 3)
> ggplot(ed_counts,aes(x = reorder(Education,-pct), y = p
ct, fill = Education, label = scales::percent(pct))) +
+   geom_col(position = 'dodge') +
+   geom_text(position = position_dodge(width = .9),    #
move to center of bars
+             vjust = -0.5,    # nudge above top of bar
+             size = 3) +
+   scale_y_continuous(labels = scales::percent) + theme(
axis.title.x=element_blank(),axis.text.x = element_text(a
ngle = 90),legend.position="none") + ggtitle("Distributio
```

```
n of Education variable") + ylab("% of Records") + theme(
plot.title = element_text(hjust = 0.5))
> # Statistical test between Education variable and Term
Deposit target variable
> chisq.test(bank_data$education, bank_data$y, correct=FA
LSE)

	Pearson's Chi-squared test

data:  bank_data$education and bank_data$y
X-squared = 193.11, df = 7, p-value < 2.2e-16

Warning message:
In chisq.test(bank_data$education, bank_data$y, correct =
FALSE) :
  Chi-squared approximation may be incorrect
> # Distribution of education variable by term deposit
> edu_counts <- bank_data %>%
+   count(Education = factor(education), Term_Deposit = f
actor(y)) %>%
+   mutate(pct = prop.table(n))
> edu_counts$pct<-round(edu_counts$pct,digits = 3)
> ggplot(edu_counts,aes(x = reorder(Education,-pct), y =
pct, fill = Term_Deposit, label = scales::percent(pct)))
+
+   geom_col(position = 'dodge') +
+   geom_text(position = position_dodge(width = .9),    #
move to center of bars
+             vjust = -0.5,    # nudge above top of bar
+             size = 3) +
+   scale_y_continuous(labels = scales::percent) + theme(
axis.title.x=element_blank(),axis.text.x = element_text(a
ngle = 90),legend.position="none") + ggtitle("Education v
/s Term Deposit") + ylab("% of Records") + theme(plot.tit
le = element_text(hjust = 0.5)) + guides(fill=guide_legen
d(title="Term Deposit"))
> ## Distribution of housing variable
> hou_counts <- bank_data %>%
+   count(Housing = factor(housing)) %>%
+   mutate(pct = prop.table(n))
> hou_counts$pct<-round(hou_counts$pct,digits = 3)
> ggplot(hou_counts,aes(x = reorder(Housing,-pct), y = pc
t, fill = Housing, label = scales::percent(pct))) +
+   geom_col(position = 'dodge') +
+   geom_text(position = position_dodge(width = .9),    #
move to center of bars
+             vjust = -0.5,    # nudge above top of bar
+             size = 3) +
+   scale_y_continuous(labels = scales::percent) + theme(
axis.title.x=element_blank(),axis.text.x = element_text(a
ngle = 0),legend.position="none") + ggtitle("Distribution
of Housing variable") + ylab("% of Records") + theme(plot
.title = element_text(hjust = 0.5))
```

```
> ## checking any relation in housing and the term deposi
t
> hou_counts1 <- bank_data %>%
+   count(Housing = factor(housing), Term_Deposit = facto
r(y)) %>%
+   mutate(pct = prop.table(n))
> hou_counts1$pct<-round(hou_counts1$pct,digits = 3)
> ggplot(hou_counts1,aes(x = reorder(Housing,-pct), y = p
ct, fill = Term_Deposit, label = scales::percent(pct))) +
+   geom_col(position = 'dodge') +
+   geom_text(position = position_dodge(width = .9),    #
move to center of bars
+             vjust = -0.5,    # nudge above top of bar
+             size = 3) +
+   scale_y_continuous(labels = scales::percent) + theme(
axis.title.x=element_blank(),axis.text.x = element_text(a
ngle = 0),legend.position="none") + ggtitle("Housing v/s
Term Deposit") + ylab("% of Records") + theme(plot.title
= element_text(hjust = 0.5)) + guides(fill=guide_legend(t
itle="Term Deposit"))
> ## Distribution of Loan variable
> ln_counts <- bank_data %>%
+   count(Loan = factor(loan)) %>%
+   mutate(pct = prop.table(n))
> ln_counts$pct<-round(ln_counts$pct,digits = 3)
> ggplot(ln_counts,aes(x = reorder(Loan,-pct), y = pct, f
ill = Loan, label = scales::percent(pct))) +
+   geom_col(position = 'dodge') +
+   geom_text(position = position_dodge(width = .9),    #
move to center of bars
+             vjust = -0.5,    # nudge above top of bar
+             size = 3) +
+   scale_y_continuous(labels = scales::percent) + theme(
axis.title.x=element_blank(),axis.text.x = element_text(a
ngle = 0),legend.position="none") + ggtitle("Distribution
of Loan variable") + ylab("% of Records") + theme(plot.ti
tle = element_text(hjust = 0.5))
> ## checking any relation in loan and the term deposit
> loan_counts <- bank_data %>%
+   count(Loan = factor(loan), Term_Deposit = factor(y))
%>%
+   mutate(pct = prop.table(n))
> loan_counts$pct<-round(loan_counts$pct,digits = 3)
> ggplot(loan_counts,aes(x = reorder(Loan,-pct), y = pct,
fill = Term_Deposit, label = scales::percent(pct))) +
+   geom_col(position = 'dodge') +
+   geom_text(position = position_dodge(width = .9),    #
move to center of bars
+             vjust = -0.5,    # nudge above top of bar
+             size = 3) +
+   scale_y_continuous(labels = scales::percent) + theme(
axis.title.x=element_blank(),axis.text.x = element_text(a
ngle = 0),legend.position="none") + ggtitle("Loan v/s Ter
```

```
m Deposit") + ylab("% of Records") + theme(plot.title = e
lement_text(hjust = 0.5)) + guides(fill=guide_legend(titl
e="Term Deposit"))
> ## checking any relation in month and the term deposit
> mon_counts <- bank_data %>%
+   count(Month = factor(month), Term_Deposit = factor(y)
) %>%
+   mutate(pct = prop.table(n))
> mon_counts$pct<-round(mon_counts$pct,digits = 3)
> ggplot(mon_counts,aes(x = reorder(Month,-pct), y = pct,
fill = Term_Deposit, label = scales::percent(pct))) +
+   geom_col(position = 'dodge') +
+   geom_text(position = position_dodge(width = .9),    #
move to center of bars
+             vjust = -0.5,    # nudge above top of bar
+             size = 3) +
+   scale_y_continuous(labels = scales::percent) + theme(
axis.title.x=element_blank(),axis.text.x = element_text(a
ngle = 0),legend.position="none") + ggtitle("Month v/s Te
rm Deposit") + ylab("% of Records") + theme(plot.title =
element_text(hjust = 0.5)) + guides(fill=guide_legend(tit
le="Term Deposit"))
> mon_cont_y_counts<-as.data.frame(table(bank_data$month,
bank_data$contact, bank_data$y))
> names(mon_cont_y_counts)<-c("Month","Contact","TermDepo
sitYesNo","Freq")
> ggplot(mon_cont_y_counts, aes(x = Month, y = Freq))+
+   geom_bar(
+     aes(fill = TermDepositYesNo), stat = "identity", co
lor = "white",
+     position = position_dodge(0.9)
+   )+facet_wrap(~Contact)+guides(fill=guide_legend(title
="Contact"))
> ##############################  FACTOR DATA ############
#####################
> factor_cols <- c("job", "marital", "education", "defaul
t","housing","loan","contact","month","day_of_week","pout
come","y")
> bank_data[,factor_cols] <- lapply(bank_data[,factor_col
s], factor)
> #bank_data[,factor_cols] <- lapply(bank_data[,factor_co
ls], as.numeric)
> str(bank_data)
'data.frame':    41188 obs. of  21 variables:
 $ X             : int  1 2 3 4 5 6 7 8 9 10 ...
 $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
 $ job           : Factor w/ 12 levels "admin.","blue-col
lar",..: 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.6
y",..: 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","telepho
ne": 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/ 6 levels "character","fri",.
.: 3 3 3 3 3 3 3 3 3 3 ...
 $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays         : int  999 999 999 999 999 999 999 999 9
99 999 ...
 $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome      : Factor w/ 3 levels "failure","nonexist
ent",..: 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 ...
> head(bank_data)
  X age       job marital   education default housing loa
n   contact month day_of_week campaign
1 1  56 housemaid married    basic.4y      no      no   n
o telephone   may         mon        1
2 2  57  services married high.school unknown      no   n
o telephone   may         mon        1
3 3  37  services married high.school      no     yes   n
o telephone   may         mon        1
4 4  40    admin. married    basic.6y      no      no   n
o telephone   may         mon        1
5 5  56  services married high.school      no      no  ye
s telephone   may         mon        1
6 6  45  services married    basic.9y unknown      no   n
o telephone   may         mon        1
  pdays previous    poutcome emp.var.rate cons.price.idx
cons.conf.idx euribor3m nr.employed  y
1   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
2   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
3   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
4   999        0 nonexistent          1.1         93.994
-36.4     4.857        5191 no
```

```
5   999         0 nonexistent          1.1         93.994
-36.4     4.857         5191 no
6   999         0 nonexistent          1.1         93.994
-36.4     4.857         5191 no
> # Count the number of samples in each class
> table(bank_data$y)

   no   yes 
36548  4640 
> # Use ROSE to oversample the minority class
> bank_data<- ROSE(y ~ ., data = bank_data)$data
> # Count the number of samples in each class after overs
ampling
> table(bank_data$y)

   no   yes 
20627 20561 
> # Plotting dependent variable distribution in data afte
r class balance treatment
> bp <- barplot(table(bank_data$y),
+                 beside=TRUE,
+                 ylim=c(0, max(table(bank_data$y)) + 3452)
,
+                 main="Term Deposit(yes/no) Distribution",
+                 col = c("#eb8060", "#b9e38d"),
+                 border=0)
> text(bp, table(bank_data$y) + 1200, table(bank_data$y),
font=2, col="black")
> # Correlation matrix
> corr_data<-round(cor(bank_data[sapply(bank_data, is.num
eric)]),2)
> corr_data
                   X   age campaign pdays previous emp.va
r.rate cons.price.idx cons.conf.idx
X               1.00  0.04    -0.13 -0.32     0.38
-0.73          -0.48         -0.01
age             0.04  1.00     0.00 -0.05     0.04
-0.05          -0.02          0.11
campaign       -0.13  0.00     1.00  0.08    -0.09
0.17           0.11          -0.03
pdays          -0.32 -0.05     0.08  1.00    -0.58
0.28           0.03          -0.12
previous        0.38  0.04    -0.09 -0.58     1.00
-0.32          -0.05          0.06
emp.var.rate   -0.73 -0.05     0.17  0.28    -0.32
1.00           0.59          -0.05
cons.price.idx -0.48 -0.02     0.11  0.03    -0.05
0.59           1.00          -0.13
cons.conf.idx  -0.01  0.11    -0.03 -0.12     0.06
-0.05          -0.13          1.00
euribor3m      -0.71 -0.03     0.15  0.30    -0.35
0.76           0.45          0.03
```

```
nr.employed    -0.71 -0.06     0.16  0.40    -0.44
0.74           0.29        -0.07
               euribor3m nr.employed
X                  -0.71       -0.71
age                -0.03       -0.06
campaign            0.15        0.16
pdays               0.30        0.40
previous           -0.35       -0.44
emp.var.rate        0.76        0.74
cons.price.idx      0.45        0.29
cons.conf.idx       0.03       -0.07
euribor3m           1.00        0.75
nr.employed         0.75        1.00
> # plotting corr matrix
> melted_corr_data <- melt(corr_data)
> ggplot(data = melted_corr_data, aes(x=Var1, y=Var2, fil
l=value)) +
+   geom_tile() +
+   geom_text(aes(Var2, Var1, label = value), size = 5) +
+   scale_fill_gradient2(low = "blue", high = "red",
+                         limit = c(-1,1), name="Correlati
on") +
+   theme(axis.title.x = element_blank(),
+         axis.text.x = element_text(angle = 90),
+         axis.title.y = element_blank(),
+         panel.background = element_blank())
> ############################ DATA MODELING (CLASSIFICAT
ION) ####################
> library(lattice)
> library(ggplot2)
> library(caret)
> library(rlang)

Attaching package: ‘rlang’

The following object is masked from ‘package:wrapr’:

    :=

The following objects are masked from ‘package:purrr’:

    %@%, flatten, flatten_chr, flatten_dbl, flatten_int,
flatten_lgl, flatten_raw,
    invoke, splice

> library(Rcpp)
> # Splitting the data into train and test
> index <- createDataPartition(bank_data$y, p = .70, list
= FALSE)
> train <- bank_data[index, ]
> test <- bank_data[-index, ]
> dim(train)
[1] 28832    21
```

```
> #Checking dimentions
> dim(train)
[1] 28832    21
> dim(test)
[1] 12356    21
> # Check distrn of target var
> table(train$y)

   no   yes 
14439 14393 
> table(test$y)

  no  yes 
6188 6168 
> # Training the model
> logistic_model <- glm(y ~ ., family = binomial(), train
)
> # Checking the model
> summary(logistic_model)

Call:
glm(formula = y ~ ., family = binomial(), data = train)

Coefficients: (1 not defined because of singularities)
                               Estimate Std. Error z valu
e Pr(>|z|)
(Intercept)                   9.869e+00  3.641e+00   2.71
0 0.006722 **
X                             1.274e-05  2.106e-06   6.05
2 1.43e-09 ***
age                          -2.521e-03  1.391e-03  -1.81
2 0.070004 .
jobblue-collar               -1.416e-01  5.202e-02  -2.72
2 0.006491 **
jobentrepreneur              -9.390e-02  8.059e-02  -1.16
5 0.243950
jobhousemaid                 -1.681e-01  9.798e-02  -1.71
6 0.086227 .
jobmanagement                -4.607e-02  5.935e-02  -0.77
6 0.437598
jobretired                    3.653e-01  7.655e-02   4.77
2 1.82e-06 ***
jobself-employed             -1.209e-01  7.995e-02  -1.51
3 0.130329
jobservices                  -4.706e-02  5.645e-02  -0.83
4 0.404510
jobstudent                    3.049e-01  9.449e-02   3.22
7 0.001252 **
jobtechnician                -4.111e-02  4.836e-02  -0.85
0 0.395289
jobunemployed                -2.428e-02  9.132e-02  -0.26
6 0.790375
```

```
jobunknown                     2.359e-01  1.702e-01   1.38
6 0.165675
maritalmarried                 5.520e-02  4.578e-02   1.20
6 0.227970
maritalsingle                  1.528e-01  5.153e-02   2.96
6 0.003014 **
maritalunknown                 5.125e-01  2.956e-01   1.73
4 0.082989 .
educationbasic.6y              1.719e-01  7.774e-02   2.21
1 0.027049 *
educationbasic.9y             -8.271e-02  6.200e-02  -1.33
4 0.182169
educationhigh.school          -1.192e-02  6.261e-02  -0.19
0 0.848939
educationilliterate            8.631e-01  5.540e-01   1.55
8 0.119266
educationprofessional.course -3.690e-02  6.951e-02  -0.53
1 0.595535
educationuniversity.degree     2.295e-02  6.350e-02   0.36
1 0.717856
educationunknown               4.220e-02  8.488e-02   0.49
7 0.619089
defaultunknown                -1.983e-01  3.997e-02  -4.96
0 7.05e-07 ***
defaultyes                    -8.810e+00  7.246e+01  -0.12
2 0.903228
housingunknown                -2.047e-01  9.348e-02  -2.19
0 0.028518 *
housingyes                    -2.628e-02  2.823e-02  -0.93
1 0.351869
loanunknown                           NA         NA      N
A       NA
loanyes                       -1.593e-02  3.865e-02  -0.41
2 0.680157
contacttelephone              -3.589e-01  4.706e-02  -7.62
6 2.41e-14 ***
monthaug                      -2.120e-01  7.693e-02  -2.75
6 0.005846 **
monthdec                       7.309e-01  2.110e-01   3.46
4 0.000532 ***
monthjul                       6.865e-02  6.629e-02   1.03
5 0.300441
monthjun                       5.937e-02  6.778e-02   0.87
6 0.381111
monthmar                       8.654e-01  1.076e-01   8.04
6 8.58e-16 ***
monthmay                      -7.064e-01  5.469e-02 -12.91
6  < 2e-16 ***
monthnov                      -5.256e-01  6.974e-02  -7.53
7 4.82e-14 ***
monthoct                       5.689e-01  1.106e-01   5.14
2 2.72e-07 ***
```

```
monthsep                             -7.661e-02  1.197e-01  -0.64
0 0.522129
day_of_weekfri                        4.975e-02  6.960e-02   0.71
5 0.474755
day_of_weekmon                       -1.338e-01  6.941e-02  -1.92
8 0.053839 .
day_of_weekthu                        2.358e-02  6.875e-02   0.34
3 0.731646
day_of_weektue                        4.086e-02  6.943e-02   0.58
9 0.556126
day_of_weekwed                        1.345e-01  6.905e-02   1.94
7 0.051475 .
campaign                             -4.410e-02  5.673e-03  -7.77
4 7.61e-15 ***
pdays                                -2.365e-04  8.556e-05  -2.76
5 0.005699 **
previous                              1.061e-01  3.533e-02   3.00
3 0.002671 **
poutcomenonexistent                   4.803e-01  6.153e-02   7.80
5 5.95e-15 ***
poutcomesuccess                       1.641e+00  1.171e-01  14.01
4  < 2e-16 ***
emp.var.rate                         -1.439e-01  1.568e-02  -9.18
0  < 2e-16 ***
cons.price.idx                        1.198e-01  3.218e-02   3.72
2 0.000197 ***
cons.conf.idx                         1.612e-02  3.386e-03   4.76
0 1.94e-06 ***
euribor3m                            -6.165e-02  1.318e-02  -4.67
7 2.92e-06 ***
nr.employed                          -3.972e-03  2.967e-04 -13.38
8  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 39970  on 28831  degrees of freedom
Residual deviance: 31290  on 28778  degrees of freedom
AIC: 31398

Number of Fisher Scoring iterations: 8

> # Predicting in the test dataset
> pred_prob <- predict(logistic_model, test, type = "resp
onse")
> # Converting from probability to actual output
> test$pred_class <- ifelse(pred_prob >= 0.5, "yes", "no"
)
> test$pred_class <- as.factor(test$pred_class)
> # Generating the classification table
> ctab_test <- table(test$y, test$pred_class)
```

```
> ctab_test

        no  yes
  no  5100 1088
  yes 2114 4054
> #ROC
> roc <- roc(train$y, logistic_model$fitted.values)
Setting levels: control = no, case = yes
Setting direction: controls < cases
> auc(roc)
Area under the curve: 0.7918
> ## Accuracy in Test dataset
> # Accuracy = (TP + TN)/(TN + FP + FN + TP)
> accuracy_test <- sum(diag(ctab_test))/sum(ctab_test)
> accuracy_test
[1] 0.7408546
> #Precision = TP/FP + TP (Precision indicates how often
does your predicted TRUE values are actually TRUE.)
> # Precision in Test dataset
> Precision <- (ctab_test[2, 2]/sum(ctab_test[, 2]))
> Precision
[1] 0.7884092
> # Recall Or TPR = TP/(FN + TP) (Recall or TPR indicates
how often does our model predicts actual TRUE from the ov
erall TRUE events.)
> # Recall in Train dataset
> Recall <- (ctab_test[2, 2]/sum(ctab_test[2, ]))
> Recall
[1] 0.6572633
> # F1 score (F-Score is a harmonic mean of recall and pr
ecision. The score value lies between 0 and 1. The value
of 1 represents perfect precision & recall. The value 0 r
epresents the worst case.)
> F_Score <- (2 * Precision * Recall / (Precision + Recal
l))
> F_Score
[1] 0.7168877
> # Formatting results
> metric_eval <- data.frame(matrix(ncol = 6, nrow = 0))
> x <- c("Model_Name", "Accuracy", "Precision","Recall",
"F1_score", "AUC")
> colnames(metric_eval) <- x
> library(caret)
> lgr_val <- c("Logistic Regression",accuracy_test,Precis
ion,Recall,F_Score,auc(roc))
> metric_eval <- rbind(metric_eval,lgr_val)
> names(metric_eval)<-x
> ## making null for predicted column created in test dat
a
> test$pred_class<-NULL
> library(caTools)
> library(knitr)
> set.seed(123)
```

```
> library(rpart)
> classifier <- rpart(formula = y ~ .,
+                     data = train)
> # rpart.plot(classifier)
> # Predicting the Test set results
> names(test)
 [1] "X"              "age"            "job"            "
marital"        "education"
 [6] "default"        "housing"        "loan"           "
contact"        "month"
[11] "day_of_week"    "campaign"       "pdays"          "
previous"       "poutcome"
[16] "emp.var.rate"   "cons.price.idx" "cons.conf.idx"  "
euribor3m"      "nr.employed"
[21] "y"
> str(test)
'data.frame':	12356 obs. of  21 variables:
 $ X             : num  24940 17662 32863 37848 17959 ...
 $ age           : num  42 52.3 43.1 26.9 50.2 ...
 $ job           : Factor w/ 12 levels "admin.","blue-col
lar",..: 8 2 2 12 1 1 2 2 1 8 ...
 $ marital       : Factor w/ 4 levels "divorced","married
",..: 2 3 2 2 2 1 2 2 2 3 ...
 $ education     : Factor w/ 8 levels "basic.4y","basic.6
y",..: 7 3 3 6 7 4 1 1 7 3 ...
 $ default       : Factor w/ 3 levels "no","unknown",..:
1 1 1 1 1 1 2 2 1 1 ...
 $ housing       : Factor w/ 3 levels "no","unknown",..:
1 3 2 3 3 1 1 1 3 3 ...
 $ loan          : Factor w/ 3 levels "no","unknown",..:
1 1 2 1 1 1 1 3 1 3 ...
 $ contact       : Factor w/ 2 levels "cellular","telepho
ne": 1 1 1 1 1 1 1 2 1 1 ...
 $ month         : Factor w/ 10 levels "apr","aug","dec",
..: 2 4 1 7 2 8 7 7 1 8 ...
 $ day_of_week   : Factor w/ 6 levels "character","fri",.
.: 5 6 2 1 4 2 2 2 6 6 ...
 $ campaign      : num  1.9515 18.2955 4.0056 1.012 0.059
4 ...
 $ pdays         : num  954 870 1026 998 983 ...
 $ previous      : num  0.1729 0.5507 -0.2379 -0.3189 -0.
0716 ...
 $ poutcome      : Factor w/ 3 levels "failure","nonexist
ent",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ emp.var.rate  : num  1.65 1.3 -2.33 -3.29 2.13 ...
 $ cons.price.idx: num  93.4 94.3 92.8 93.1 93.9 ...
 $ cons.conf.idx : num  -33.2 -45 -47.1 -43.9 -36.4 ...
 $ euribor3m     : num  5.45 4.95 1.71 2.56 2.67 ...
 $ nr.employed   : num  5220 5257 5102 5133 5218 ...
 $ y             : Factor w/ 2 levels "no","yes": 1 1 1 1
1 1 1 1 1 1 ...
> y_pred <- predict(classifier,
+                   newdata = test,
```

```
+                   type = 'prob')[,2]
> library(pROC)
> tree.roc <- roc(test$y, y_pred)
Setting levels: control = no, case = yes
Setting direction: controls < cases
> dt_auc<-tree.roc$auc[1]
> ## for confusion matrix evaluation
> y_pred = predict(classifier,
+                   newdata = test,
+                   type = 'class')
> # Making the Confusion Matrix
> library(caret)
> cm<-confusionMatrix(as.factor(y_pred), test$y, mode = "
everything", positive="yes")
> cm
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  5340 1155
       yes  848 5013

               Accuracy : 0.8379
                 95% CI : (0.8313, 0.8444)
    No Information Rate : 0.5008
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6758

 Mcnemar's Test P-Value : 8.073e-12

            Sensitivity : 0.8127
            Specificity : 0.8630
         Pos Pred Value : 0.8553
         Neg Pred Value : 0.8222
              Precision : 0.8553
                 Recall : 0.8127
                     F1 : 0.8335
             Prevalence : 0.4992
         Detection Rate : 0.4057
   Detection Prevalence : 0.4743
      Balanced Accuracy : 0.8379

       'Positive' Class : yes

> # Adding results in formatted matrix
> dt_val <- c("Decision Tree",
+              cm$overall[1],
+              cm$byClass[5],
+              cm$byClass[6],
+              cm$byClass[7],dt_auc)
> metric_eval <- rbind(metric_eval,dt_val)
> names(metric_eval)<-x
```

```
> ###############################    RANDOM FOREST   #####
#########################
> install.packages("randomForest")
WARNING: Rtools is required to build R packages but is no
t currently installed. Please download and install the ap
propriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Asus/AppData/Local/R/wi
n-library/4.3'
(as 'lib' is unspecified)
trying URL 'http://cran.rstudio.com/bin/windows/contrib/4
.3/randomForest_4.7-1.1.zip'
Content type 'application/zip' length 222105 bytes (216 K
B)
downloaded 216 KB

package 'randomForest' successfully unpacked and MD5 sums
checked
Warning in install.packages :
  cannot remove prior installation of package 'randomFore
st'
Warning in install.packages :
  problem copying C:\Users\Asus\AppData\Local\R\win-libra
ry\4.3\00LOCK\randomForest\libs\x64\randomForest.dll to C
:\Users\Asus\AppData\Local\R\win-library\4.3\randomForest
\libs\x64\randomForest.dll: Permission denied
Warning in install.packages :
  restored 'randomForest'

The downloaded binary packages are in
      C:\Users\Asus\AppData\Local\Temp\Rtmpqs94wc\downloa
ded_packages
> library(randomForest)
randomForest 4.7-1.1
Type rfNews() to see new features/changes/bug fixes.

Attaching package: 'randomForest'

The following object is masked from 'package:gridExtra':

    combine

The following object is masked from 'package:dplyr':

    combine

The following object is masked from 'package:ggplot2':

    margin

> library(knitr)
> library(randomForest)
```

```
> # Random Forest for classification
> classifier_RF = randomForest(x = train[-21],
+                              y = train$y,
+                              ntree = 500)
> classifier_RF

Call:
 randomForest(x = train[-21], y = train$y, ntree = 500) 
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 10.73%
Confusion matrix:
       no   yes class.error
no  12906  1533   0.1061708
yes  1562 12831   0.1085250
> # Predicting the Test set results
> y_pred_rf = predict(classifier_RF, newdata = test[-21])
> # Plot the error vs The number of trees graph
> plot(classifier_RF)
> # Variable importance plot
> varImpPlot(classifier_RF)
> # confusion matrix
> cm<-confusionMatrix(y_pred_rf, test$y, mode = "everythi
ng", positive="yes")
> cm
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  5492  675
       yes  696 5493
                                          
               Accuracy : 0.889           
                 95% CI : (0.8834, 0.8945)
    No Information Rate : 0.5008          
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.7781          
                                          
 Mcnemar's Test P-Value : 0.5891          
                                          
            Sensitivity : 0.8906          
            Specificity : 0.8875          
         Pos Pred Value : 0.8875          
         Neg Pred Value : 0.8905          
              Precision : 0.8875          
                 Recall : 0.8906          
                     F1 : 0.8891          
             Prevalence : 0.4992          
         Detection Rate : 0.4446          
   Detection Prevalence : 0.5009          
```

```
      Balanced Accuracy : 0.8890

       'Positive' Class : yes

> # ROC
> require(pROC)
> rf.roc<-roc(train$y,classifier_RF$votes[,2])
Setting levels: control = no, case = yes
Setting direction: controls < cases
> plot(rf.roc)
> # AUC
> rf_auc<-auc(rf.roc)[1]
> rf_val <- c("Random Forest",cm$overall[1],cm$byClass[5]
,cm$byClass[6],cm$byClass[7],rf_auc)
> metric_eval <- rbind(metric_eval,rf_val)
> colnames(metric_eval) <- x
> # Adding results in formatted matrix
> metric_eval$Accuracy<-round(as.numeric(metric_eval$Accu
racy),digits = 4)
> metric_eval$Precision<-round(as.numeric(metric_eval$Pre
cision),digits = 4)
> metric_eval$Recall<-round(as.numeric(metric_eval$Recall
),digits = 4)
> metric_eval$F1_score<-round(as.numeric(metric_eval$F1_s
core),digits = 4)
> metric_eval$AUC<-round(as.numeric(metric_eval$AUC),digi
ts = 4)
> metric_eval
           Model_Name Accuracy Precision Recall F1_score
AUC
1 Logistic Regression   0.7409    0.7884 0.6573   0.7169
0.7918
2       Decision Tree   0.8379    0.8553 0.8127   0.8335
0.8636
3       Random Forest   0.8890    0.8875 0.8906   0.8891
0.9562
> head(train[,21])
[1] no no no no no no
Levels: no yes
> X_train = data.matrix(train[,-21])                  # i
ndependent variables for train
> y_train = train[,21]                                # d
ependent variables for train
> X_test = data.matrix(test[,-21])                    # i
ndependent variables for test
> y_test = test[,21]                                   #
dependent variables for test
> # convert the train and test data into xgboost matrix t
ype.
> xgboost_train = xgb.DMatrix(data=X_train, label=y_train
)
> xgboost_test = xgb.DMatrix(data=X_test, label=y_test)
> # train a model using our training data
```

```
> model <- xgboost(data = xgboost_train,
# the data
+                    max.depth=3,
# max depth
+                    nrounds=50)
# max number of boosting iterations
[1]   train-rmse:0.823059
[2]   train-rmse:0.628487
[3]   train-rmse:0.504846
[4]   train-rmse:0.426756
[5]   train-rmse:0.381586
[6]   train-rmse:0.353500
[7]   train-rmse:0.337596
[8]   train-rmse:0.328506
[9]   train-rmse:0.322962
[10]  train-rmse:0.319107
[11]  train-rmse:0.317086
[12]  train-rmse:0.315439
[13]  train-rmse:0.312675
[14]  train-rmse:0.311518
[15]  train-rmse:0.310669
[16]  train-rmse:0.309970
[17]  train-rmse:0.309127
[18]  train-rmse:0.308530
[19]  train-rmse:0.307450
[20]  train-rmse:0.307012
[21]  train-rmse:0.306452
[22]  train-rmse:0.305276
[23]  train-rmse:0.304790
[24]  train-rmse:0.304363
[25]  train-rmse:0.303636
[26]  train-rmse:0.303277
[27]  train-rmse:0.302560
[28]  train-rmse:0.302270
[29]  train-rmse:0.301510
[30]  train-rmse:0.301212
[31]  train-rmse:0.300568
[32]  train-rmse:0.299573
[33]  train-rmse:0.299228
[34]  train-rmse:0.298836
[35]  train-rmse:0.298129
[36]  train-rmse:0.297282
[37]  train-rmse:0.297008
[38]  train-rmse:0.296870
[39]  train-rmse:0.296453
[40]  train-rmse:0.296265
[41]  train-rmse:0.296093
[42]  train-rmse:0.295908
[43]  train-rmse:0.295525
[44]  train-rmse:0.295035
[45]  train-rmse:0.294670
[46]  train-rmse:0.294373
[47]  train-rmse:0.294160
```

```
[48]  train-rmse:0.294079
[49]  train-rmse:0.293949
[50]  train-rmse:0.293720
> summary(model)
               Length Class              Mode
handle             1  xgb.Booster.handle externalptr
raw            61973  -none-             raw
niter              1  -none-             numeric
evaluation_log     2  data.table         list
call              14  -none-             call
params             2  -none-             list
callbacks          2  -none-             list
feature_names     20  -none-             character
nfeatures          1  -none-             numeric
> # Predicting
> pred_test = predict(model, xgboost_test)
> pred_y = as.factor((levels(y_test))[round(pred_test)])
> print(pred_y)
   [1] no  yes no  no  no  no  yes no  no  no  no  no  no
no  no  no  no  no  no  yes no  yes
  [23] no  no  no  no  no  yes no  no  no  no  yes yes no
no  no  no  no  no  yes no  no  yes
  [45] no  no  no  no  no  no  no  no  no  no  no  no  no
no  yes no  no  no  no  no  yes no
  [67] no  no  no  yes no  no  no  no  no  no  no  no  no
no  no  no  no  no  no  no  no  no
  [89] no  no  no  no  no  no  no  no  no  no  no  no  no
no  no  no  no  no  no  no  yes no
 [111] no  no  no  no  no  no  no  no  no  no  no  no  no
no  no  yes yes no  no  no  no  no
 [133] no  no  no  no  no  no  no  no  no  no  no  no  no
no  no  no  no  no  no  no  no  no
 [155] no  no  no  no  no  no  no  no  no  no  no  no  no
no  no  no  no  no  no  no  no  yes
 [177] no  no  no  no  no  no  no  no  no  no  no  no  no
no  no  no  no  no  yes no  yes no
 [199] no  no  yes yes no  no  yes no  no  no  yes no  no
yes no  no  no  no  no  no  no  no
 [221] no  no  no  no  no  no  no  no  no  no  yes no  no
no  no  yes no  no  no  no  no  no
 [243] no  no  yes no  no  no  no  no  yes no  no  no  no
no  no  no  no  no  no  no  no  no
 [265] no  no  no  no  no  yes no  no  no  no  no  no  no
no  no  no  no  no  no  no  no  no
 [287] no  no  no  no  no  yes no  no  no  no  no  no  no
no  no  no  no  no  no  no  no  no
 [309] no  no  yes no  yes yes no  no  no  no  no  no  no
yes no  no  no  no  no  no  no  yes
 [331] no  no  no  no  no  no  no  no  no  no  no  no  no
no  no  no  no  no  no  no  no  no
 [353] no  no  no  no  no  no  no  no  no  no  no  no  no
no  yes no  no  yes no  no  no  no
```

```
 [375] no  no  no  yes no  no  no  no  no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [397] no  no  no  no  no  no  no  no  no  no  no  yes no 
no  no  no  no  no  no  no  no  no 
 [419] no  no  no  no  no  no  no  no  no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [441] no  no  no  no  no  no  no  no  no  no  no  no  ye
s no  no  no  no  no  no  no  yes no 
 [463] yes no  no  no  no  no  no  no  no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [485] no  no  no  no  yes no  no  no  no  no  no  yes no 
no  no  no  no  no  no  no  no  no 
 [507] no  no  no  no  no  no  yes no  no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [529] no  no  no  no  no  no  no  no  no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [551] no  no  no  no  no  no  yes no  no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [573] no  no  no  no  no  no  yes no  no  no  no  yes no 
no  no  no  no  no  no  no  no  no 
 [595] no  no  no  no  no  no  no  no  no  no  no  no  no 
yes no  no  no  no  no  no  no  no 
 [617] no  no  no  no  no  no  yes no  no  yes no  no  no 
no  no  no  no  no  no  yes yes no 
 [639] no  no  no  no  no  no  no  no  no  no  no  no  no 
no  no  no  no  yes no  no  no  no 
 [661] no  no  no  yes no  no  no  no  no  no  no  no  ye
s no  no  no  no  yes no  no  yes no 
 [683] no  no  no  no  no  no  no  no  no  no  no  no  no 
no  no  no  no  no  no  yes no  no 
 [705] no  yes no  no  yes no  no  yes no  yes no  no  no 
no  no  yes no  no  no  no  no  no 
 [727] no  no  no  no  no  no  no  no  no  no  yes no  no 
no  no  no  no  no  no  no  no  no 
 [749] no  yes no  no  no  yes no  no  no  no  yes no  no 
no  no  no  yes no  no  no  no  no 
 [771] no  no  no  no  no  no  no  no  no  no  no  no  no 
no  no  no  no  yes no  no  no  no 
 [793] no  no  no  no  no  no  no  no  no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [815] no  no  no  no  no  no  yes yes no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [837] yes no  no  no  no  no  no  no  no  no  yes no  no 
no  no  no  no  no  no  no  yes yes
 [859] no  no  no  no  no  no  no  no  no  no  no  no  no 
no  no  no  yes no  no  no  no  no 
 [881] no  no  no  yes no  no  no  yes no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [903] no  no  no  no  no  no  no  no  no  no  yes no  no 
no  no  no  no  no  no  no  no  no 
 [925] no  no  no  no  no  no  no  no  no  no  no  no  no 
no  no  no  no  no  no  no  no  no 
```

```
 [947] no  no  no  no  no  no  no  no  yes no  no  no  no 
no  no  no  no  no  no  no  no  no 
 [969] no  no  no  no  no  no  no  yes no  no  no  no  ye
s no  no  no  no  no  no  no  no  no 
 [991] no  no  no  no  no  no  no  no  no  no 
 [ reached getOption("max.print") -- omitted 11356 entrie
s ]
Levels: no yes
> #Confusion matrix
> conf_mat = confusionMatrix(y_test, pred_y)
> print(conf_mat)
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  5637  551
       yes  977 5191
                                          
               Accuracy : 0.8763          
                 95% CI : (0.8704, 0.8821)
    No Information Rate : 0.5353          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.7526          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.8523          
            Specificity : 0.9040          
         Pos Pred Value : 0.9110          
         Neg Pred Value : 0.8416          
             Prevalence : 0.5353          
         Detection Rate : 0.4562          
   Detection Prevalence : 0.5008          
      Balanced Accuracy : 0.8782          
                                          
       'Positive' Class : no              
                                          
> #ROC
> roc_test <- roc(test$y,round(pred_test),  algorithm = 2
)
Setting levels: control = no, case = yes
Setting direction: controls < cases
> plot(roc_test )
> #AUC
> Xgb_auc = auc(roc_test )
> # Adding results in formatted matrix
> xgb_val <- c("Xgboost",conf_mat$overall[1],conf_mat$byC
lass[5],conf_mat$byClass[6],conf_mat$byClass[7],Xgb_auc)
> metric_eval <- rbind(metric_eval,xgb_val)
> colnames(metric_eval) <- x
> metric_eval$Accuracy<-round(as.numeric(metric_eval$Accu
racy),digits = 4)
```

```
> metric_eval$Precision<-round(as.numeric(metric_eval$Pre
cision),digits = 4)
> metric_eval$Recall<-round(as.numeric(metric_eval$Recall
),digits = 4)
> metric_eval$F1_score<-round(as.numeric(metric_eval$F1_s
core),digits = 4)
> metric_eval$AUC<-round(as.numeric(metric_eval$AUC),digi
ts = 4)
> metric_eval
           Model_Name Accuracy Precision Recall F1_score
AUC
1 Logistic Regression   0.7409    0.7884 0.6573   0.7169
0.7918
2       Decision Tree   0.8379    0.8553 0.8127   0.8335
0.8636
3       Random Forest   0.8890    0.8875 0.8906   0.8891
0.9562
4             Xgboost   0.8763    0.9110 0.8523   0.8806
0.8763
>
>
>
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
> ############################### END ###################
#######################
```