Complete dataset analysis

Yoga Ramachandran

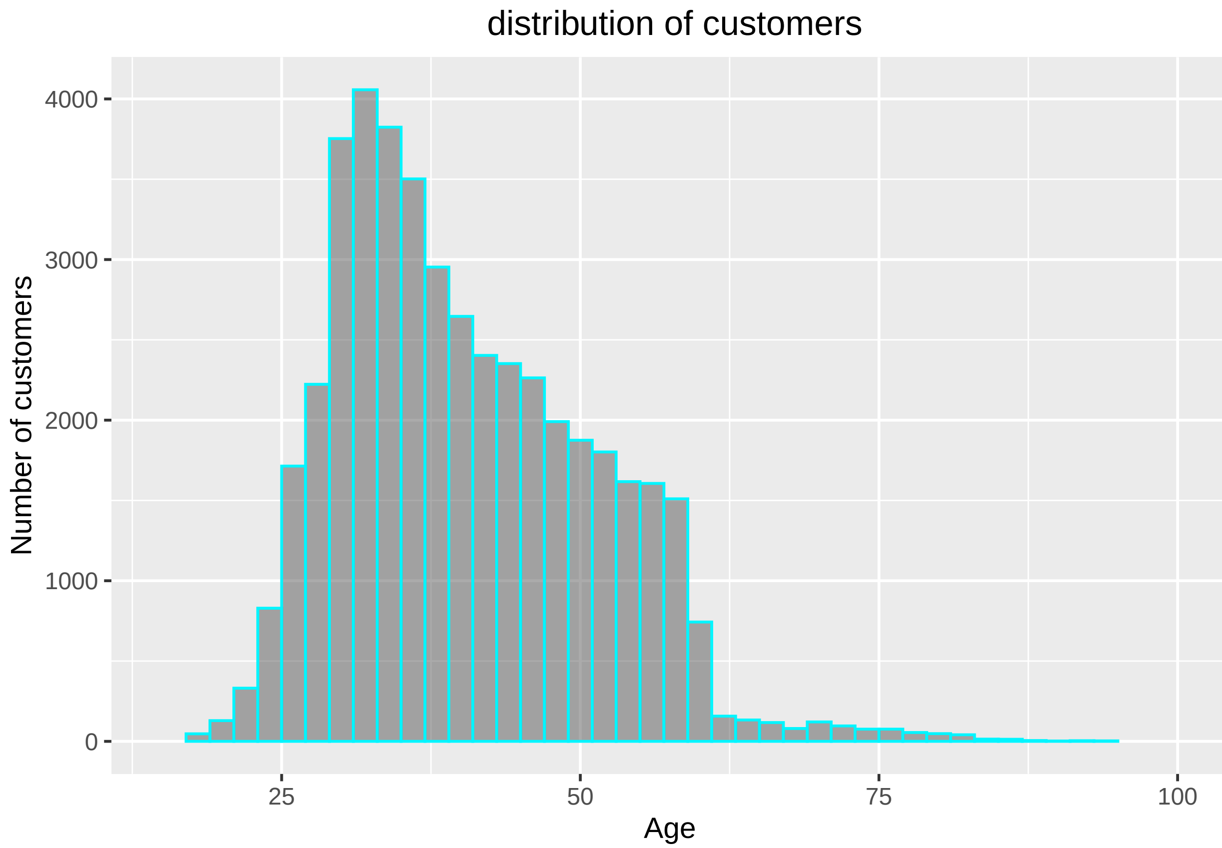
```r
# Rename target variable(y) to termdeposit
bank_data <- bank_data %>% rename(termdeposit=y)
# Creating copy of main data for further analysis
bankfull1<-bank_data
str(bankfull1)
```

```
## 'data.frame':    45211 obs. of  17 variables:
##  $ age        : int  58 44 33 47 33 35 28 42 58 43 ...
##  $ job        : chr  "management" "technician" "entrepreneur" "blue-collar" ...
##  $ marital    : chr  "married" "single" "married" "married" ...
##  $ education  : chr  "tertiary" "secondary" "secondary" "unknown" ...
##  $ default    : chr  "no" "no" "no" "no" ...
##  $ balance    : int  2143 29 2 1506 1 231 447 2 121 593 ...
##  $ housing    : chr  "yes" "yes" "yes" "yes" ...
##  $ loan       : chr  "no" "no" "yes" "no" ...
##  $ contact    : chr  "unknown" "unknown" "unknown" "unknown" ...
##  $ day        : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ month      : chr  "may" "may" "may" "may" ...
##  $ duration   : int  261 151 76 92 198 139 217 380 50 55 ...
##  $ campaign   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays      : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ previous   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome   : chr  "unknown" "unknown" "unknown" "unknown" ...
##  $ termdeposit: chr  "no" "no" "no" "no" ...
```

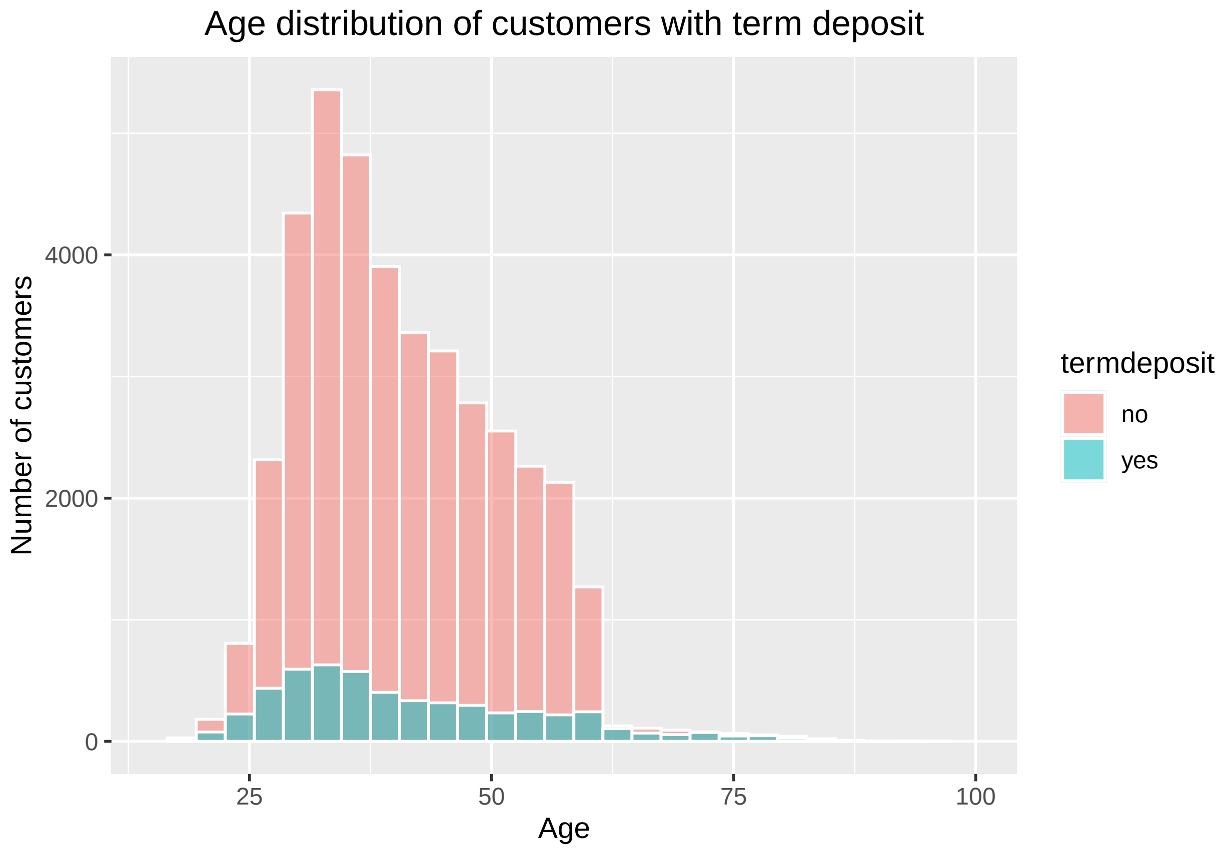
Data Visualisation:

1.Age distribution

```r
#age
plot1<-ggplot(bank_data, aes(x=age))+geom_histogram(color='turquoise1', alpha=0.5,
                                                    position = 'identity',
                                                    binwidth =2)+
  coord_cartesian(xlim=c(15,100))+ylab('Number of customers')+xlab('Age')+ggtitle(' distribution of cus
plot2<-ggplot(bank_data, aes(x=age, fill=termdeposit))+geom_histogram(color='white', alpha=0.5,position
  ylab('Number of customers')+xlab('Age')+ggtitle('Age distribution of customers with term deposit')+the
plot1
```


distribution of customers
Number of customers
4000
3000
2000
1000
0
25
50
75
100
Age

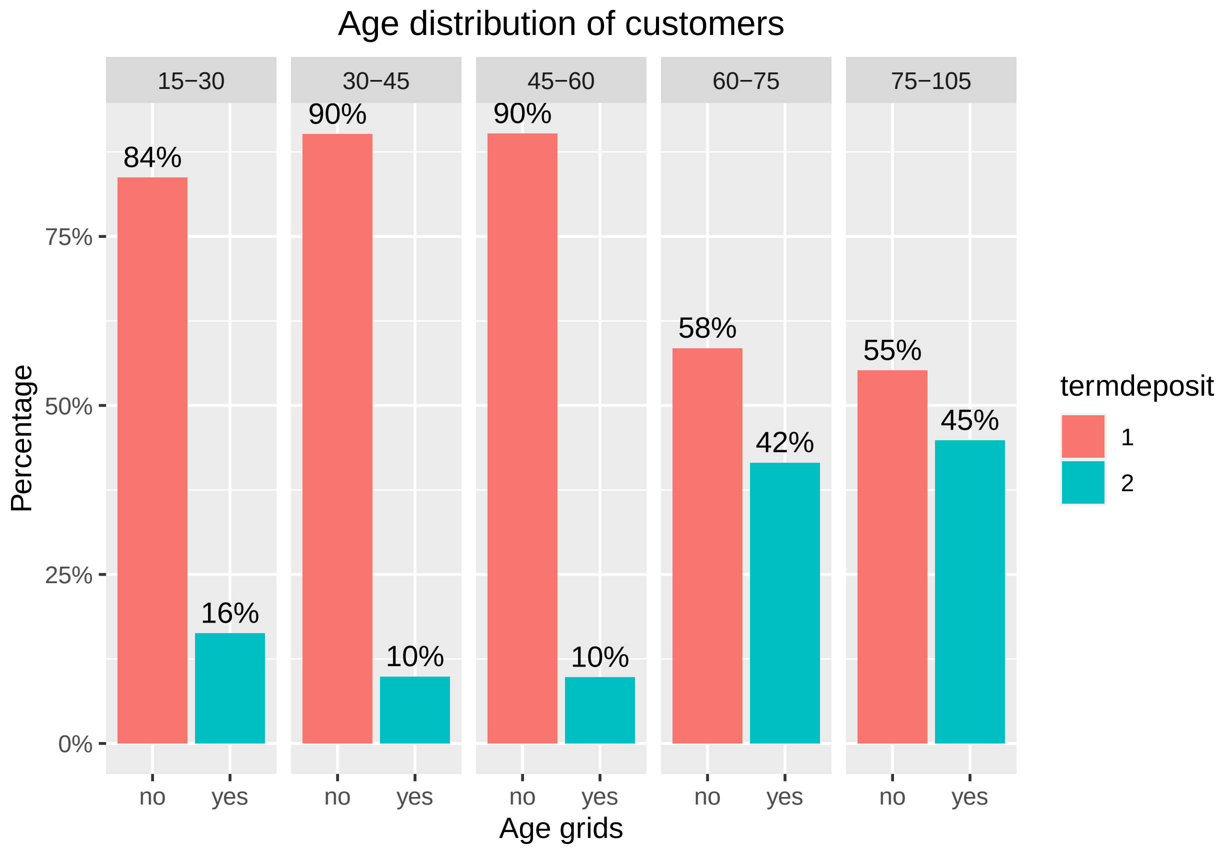
```
plot2
```



In terms of the count, most of the customers are in the age group between 25 to 40. As the number of customers are more in this age group, more customers have subscribed to term deposit.

2. Categorizing age to bins for better understanding

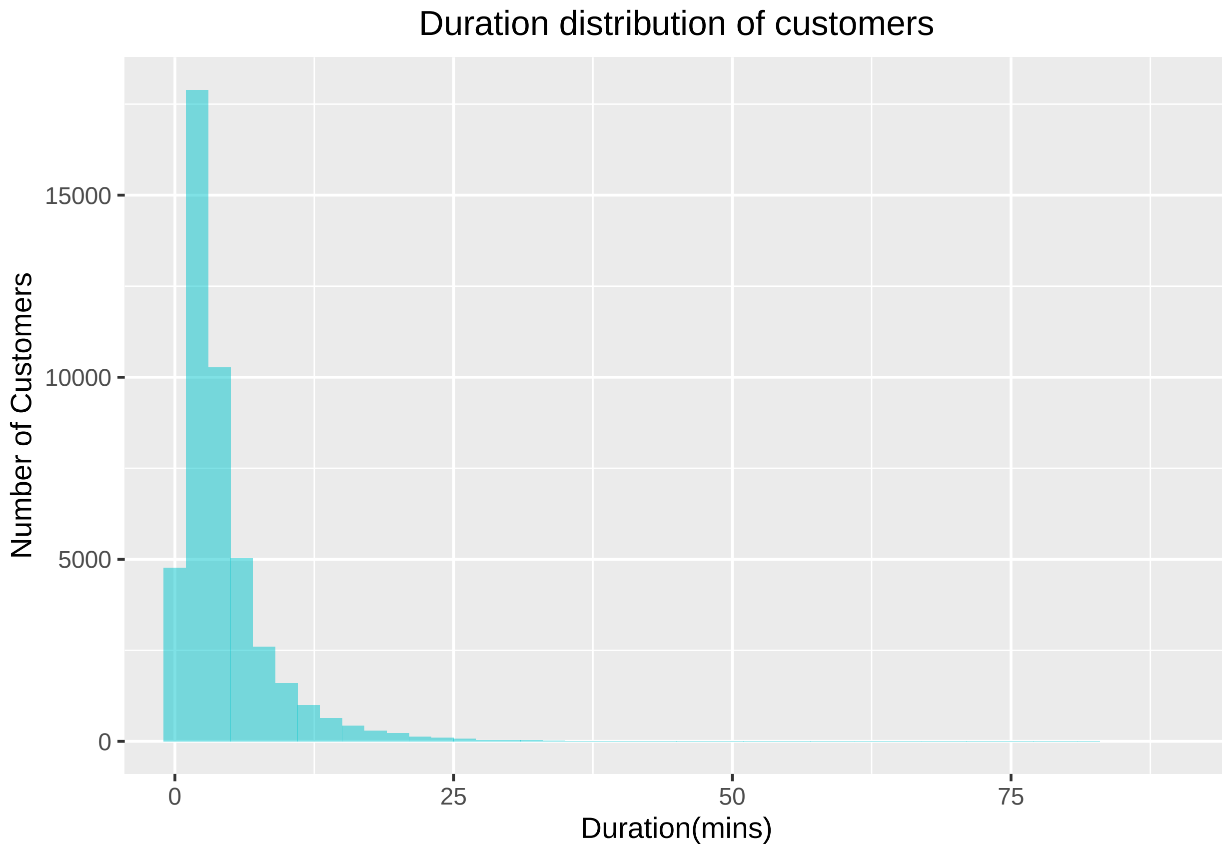
```
agebreaks<-c(15,30,45,60,75,105)
agelabels<-c('15-30','30-45','45-60','60-75','75-105')
bank_data$age_bin<-cut(bank_data$age,breaks = agebreaks, labels = agelabels, include.lowest = T)
#Binning age
ggplot(bank_data, aes(x= termdeposit,  group=age_bin)) +
    geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
    geom_text(aes( label = scales::percent(..prop..,1),
                   y= ..prop.. ), stat= "count", vjust = -.5) +
    labs(y = "Percentage", fill="termdeposit") +xlab('Age grids')+
    facet_grid(~age_bin) +ggtitle('Age distribution of customers')+theme(plot.title = element_text(hjus
    scale_y_continuous(labels = scales::percent)
```



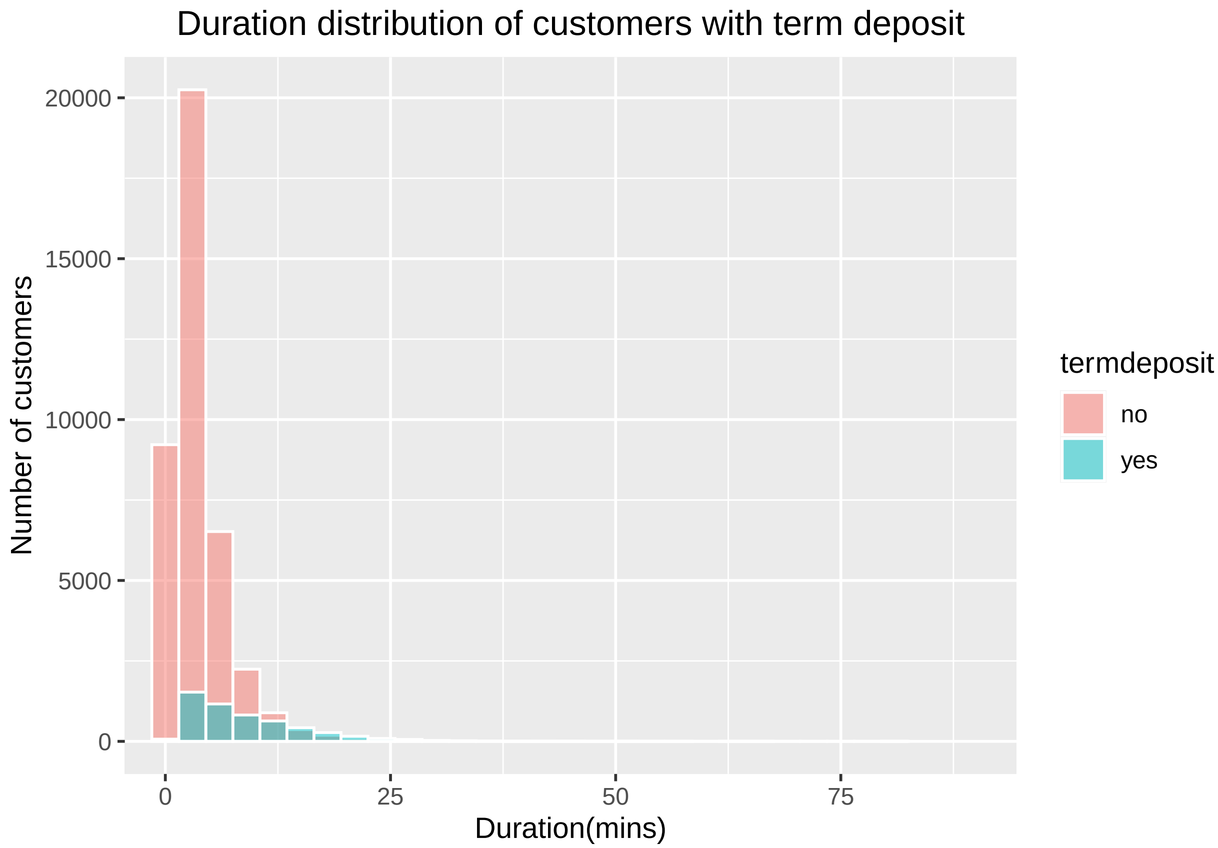
By Segmenting the customers according to age bins, we can see that even though the percentage of customers contacted in the age group 15-60 is more, the customer conversion is higher in the age group of 60 and above

3.Duration distribution

```
plot1<-ggplot(bank_data, aes(x=duration/60))+geom_histogram(fill='turquoise3',alpha=0.5,
                                                position = 'identity',binwidth =2)+
  coord_cartesian(xlim=c(0,90))+xlab('Duration(mins)')+ylab('Number of Customers')+ggtitle('Duration di

plot2<-ggplot(bank_data, aes(x=duration/60, fill=termdeposit))+geom_histogram(color='white', alpha=0.5,p
  ylab('Number of customers')+xlab('Duration(mins)')+ggtitle('Duration distribution of customers with to

plot1
```


Duration distribution of customers
Number of Customers
15000
10000
5000
0
0
25
50
75
Duration(mins)

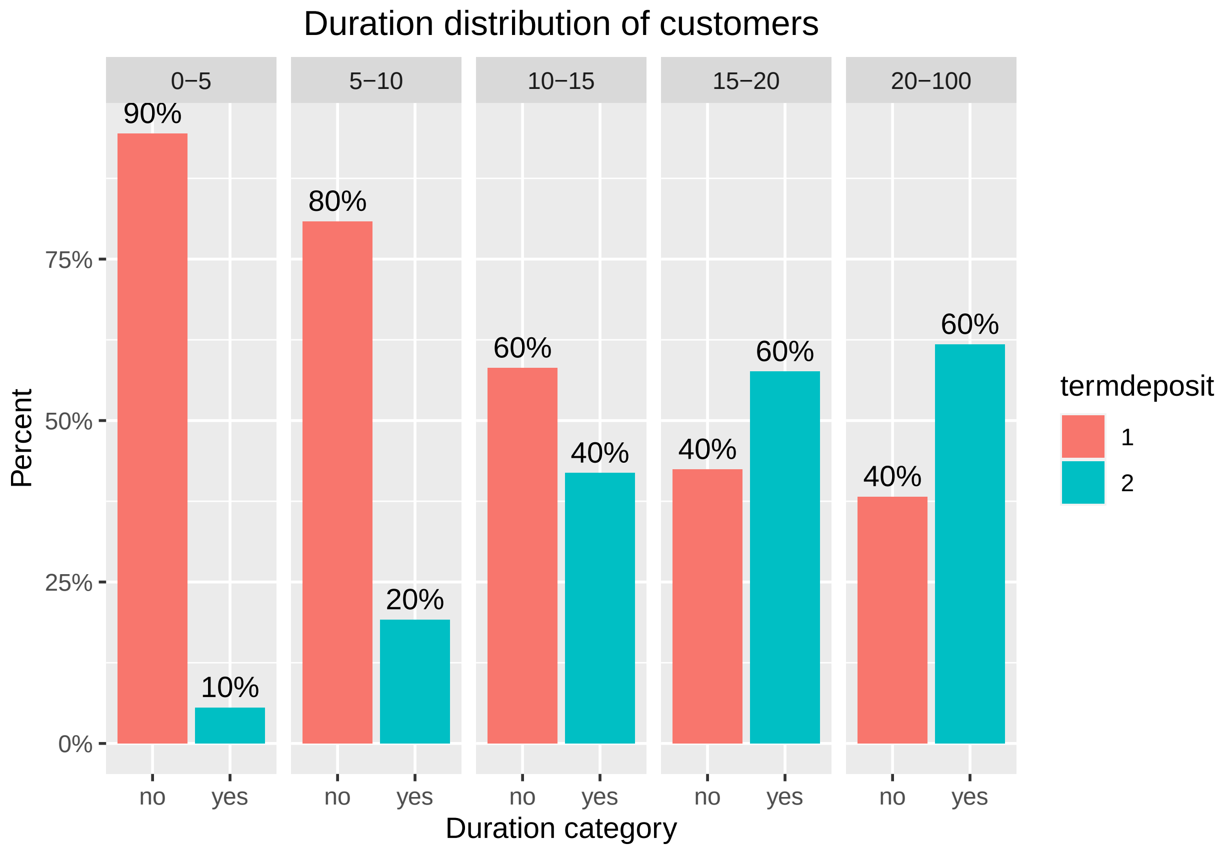
```
plot2
```



The duration(last contact duration) distribution shows a peak around 10 mins. So around 10 minutes people's are more likely to subscribe the term deposit

4.Categorising duration to bins to understand variation among various bins

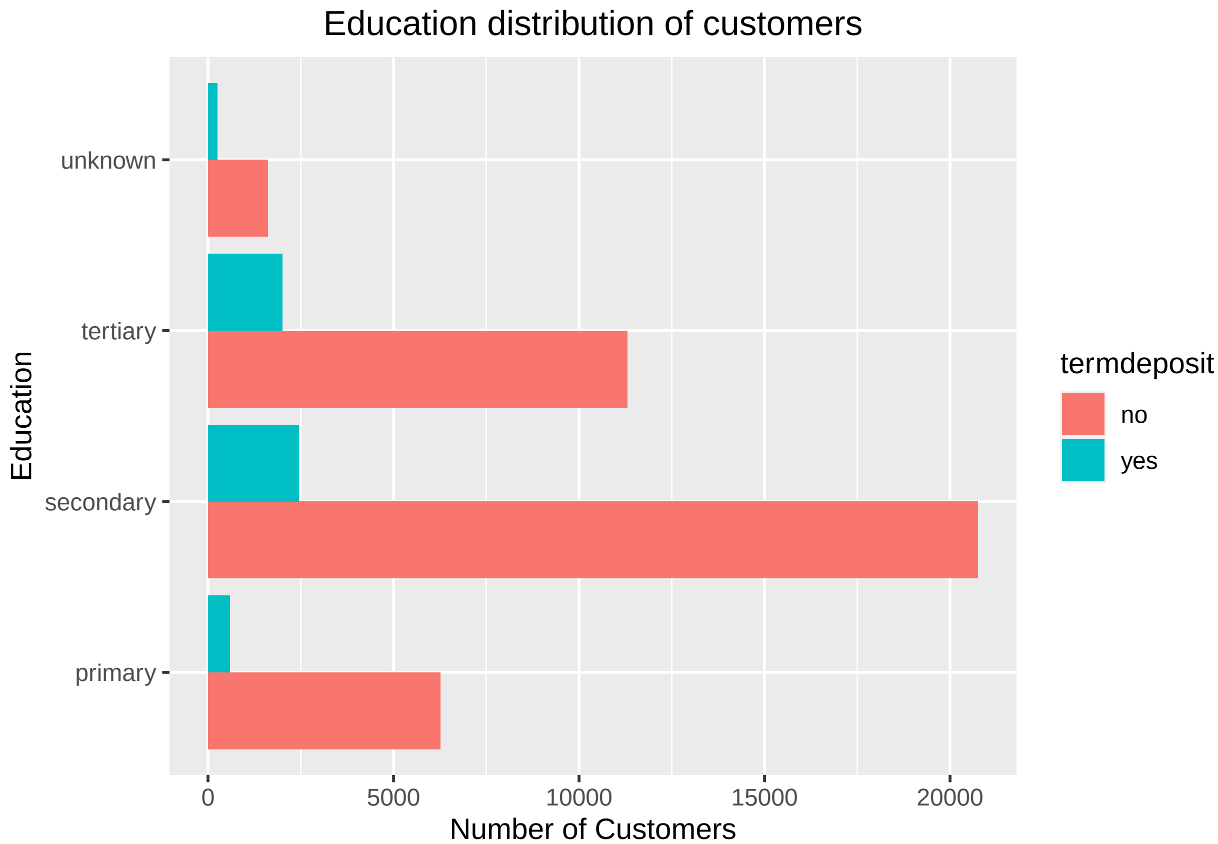
```
durationbreaks<-c(0,5,10,15,20,100)
durationlabels<-c('0-5','5-10','10-15','15-20','20-100')
bank_data$duration_bin<-cut(bank_data$duration/60,breaks = durationbreaks, labels = durationlabels, incl

ggplot(bank_data, aes(x= termdeposit,  group=duration_bin)) +
    geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
    geom_text(aes( label = scales::percent(round(..prop..,1)),
                  y= ..prop.. ), stat= "count", vjust = -.5) +
    labs(y = "Percent", fill="termdeposit") +xlab('Duration category')+ggtitle('Duration distribution o
    facet_grid(~duration_bin) +
    scale_y_continuous(labels = scales::percent)
```



By Segmenting the duration in terms of bins, we find out that as the call duration increases, there is better conversion.

5.Education distribution

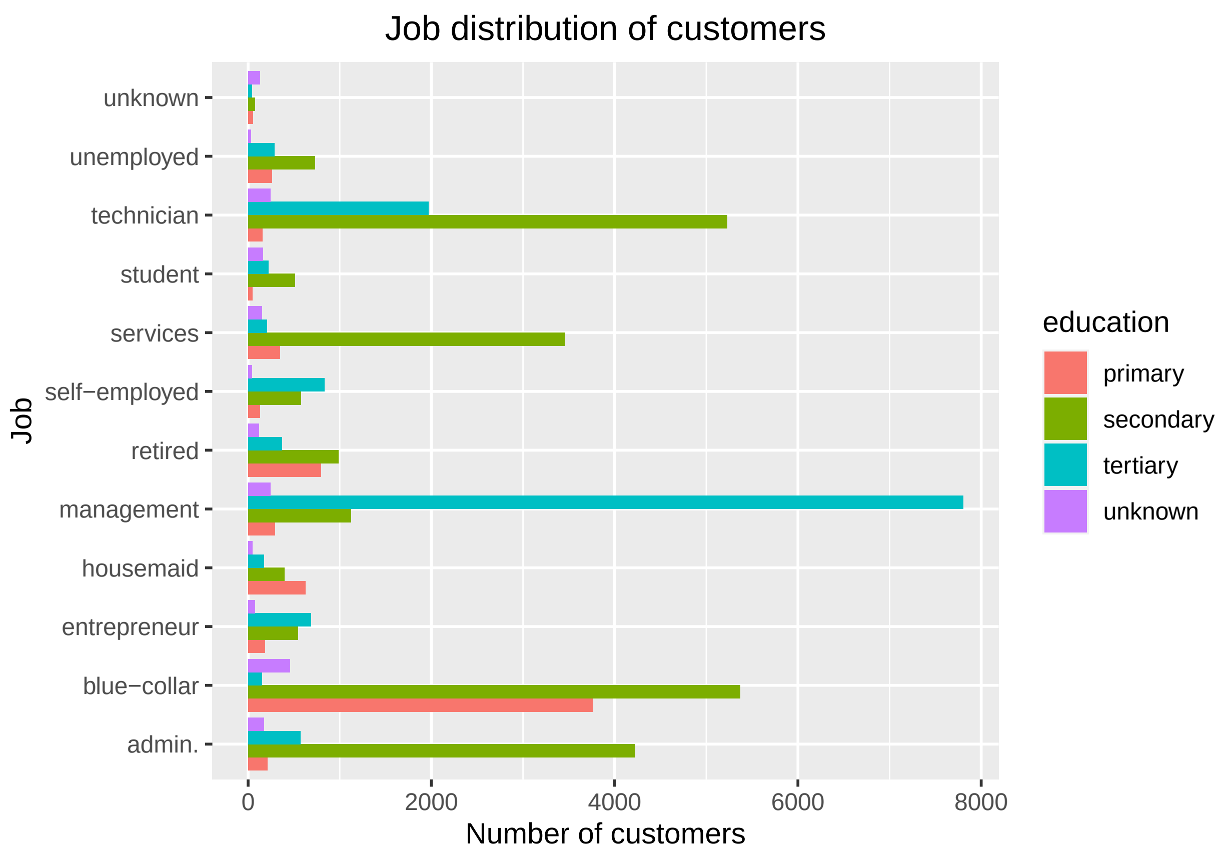
```
ggplot(bank_data, aes(y=education, fill=termdeposit))+geom_bar(position='dodge')+
  xlab('Number of Customers')+ylab('Education')+ggtitle('Education distribution of customers')+theme(pl
```



Considering the education distribution across customers, we find larger number of customers have secondary and tertiary level of education. Similar is the case for conversion.

6.Job distribution

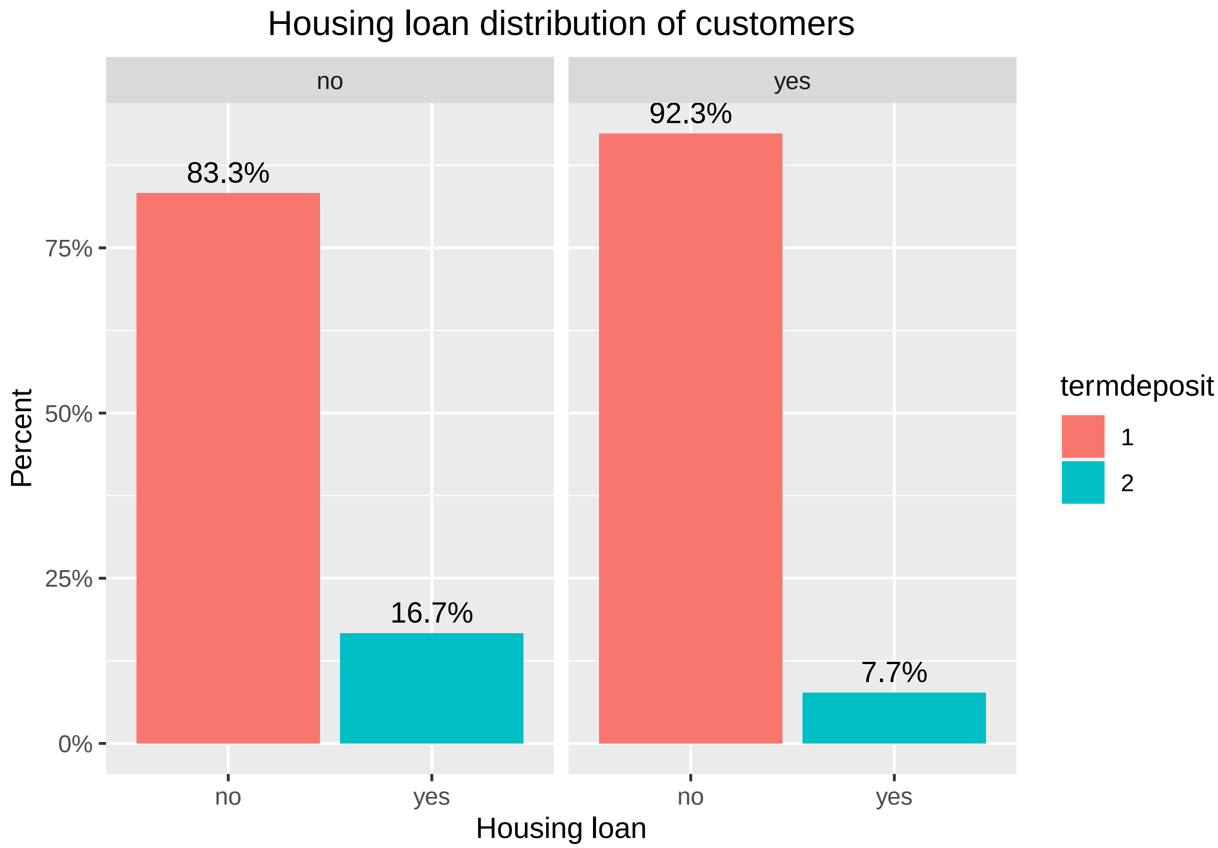
```
ggplot(bank_data, aes(y=job, fill=education))+geom_bar(position='dodge')+xlab('Number of customers')+yla
```



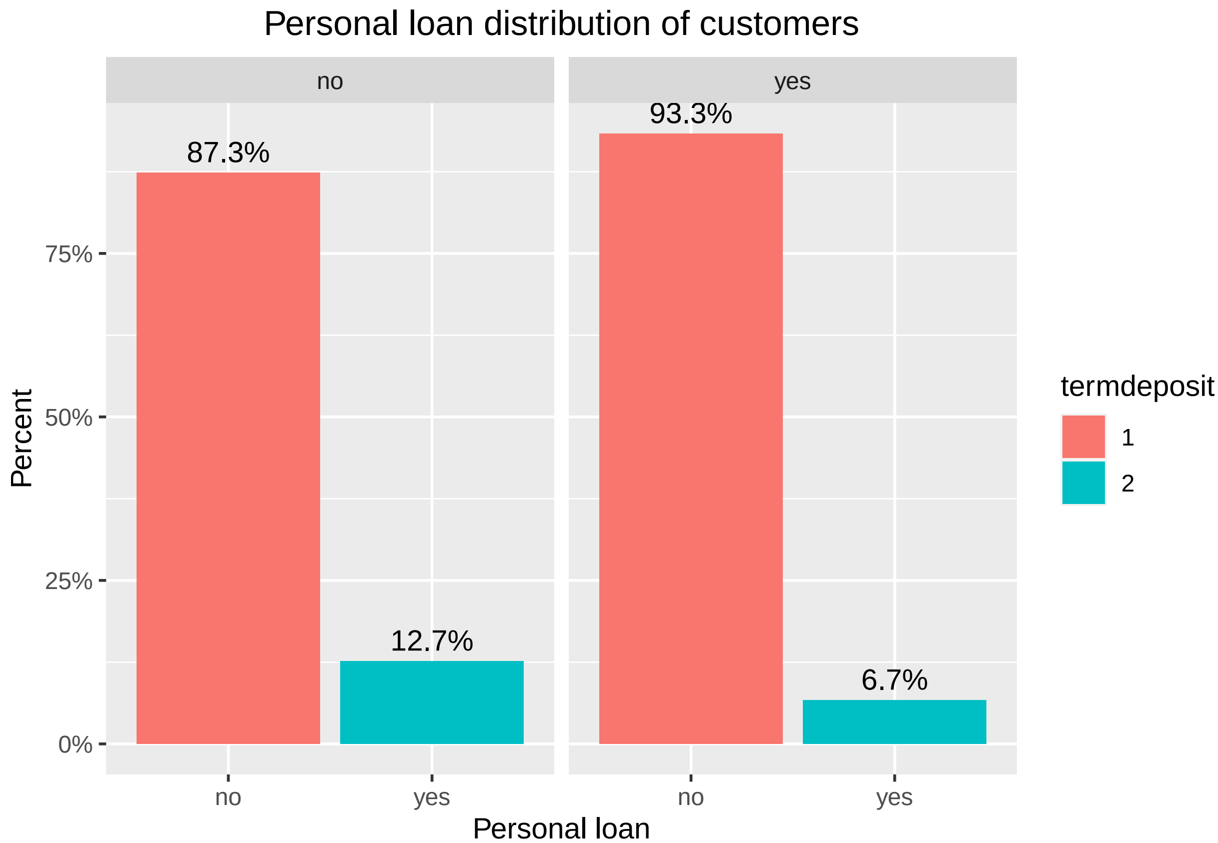
By Digging deep to understand which jobs are associated with secondary and tertiary level of education, we find that larger number of customers contacted are in the category of management, blue collar workers, technicians and admin.

7.Housing and personal loan distribution

```
plot1<-ggplot(bank_data, aes(x= termdeposit,  group=housing)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
  geom_text(aes( label = scales::percent(..prop..),
                 y= ..prop.. ), stat= "count", vjust = -.5) +
  labs(y = "Percent", fill="termdeposit") +xlab('Housing loan')+
  facet_grid(~housing) +ggtitle('Housing loan distribution of customers')+theme(plot.title = element_te
  scale_y_continuous(labels = scales::percent)

plot2<-ggplot(bank_data, aes(x= termdeposit,  group=loan)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
  geom_text(aes( label = scales::percent(..prop..),
                 y= ..prop.. ), stat= "count", vjust = -.5) +
  labs(y = "Percent", fill="termdeposit") +xlab('Personal loan')+
  facet_grid(~loan) +ggtitle('Personal loan distribution of customers')+theme(plot.title = element_text
  scale_y_continuous(labels = scales::percent)
plot1
```


Housing loan distribution of customers
no
yes
83.3%
16.7%
92.3%
7.7%
75%
50%
25%
0%
Percent
no
yes
no
yes
Housing loan
termdeposit
1
2

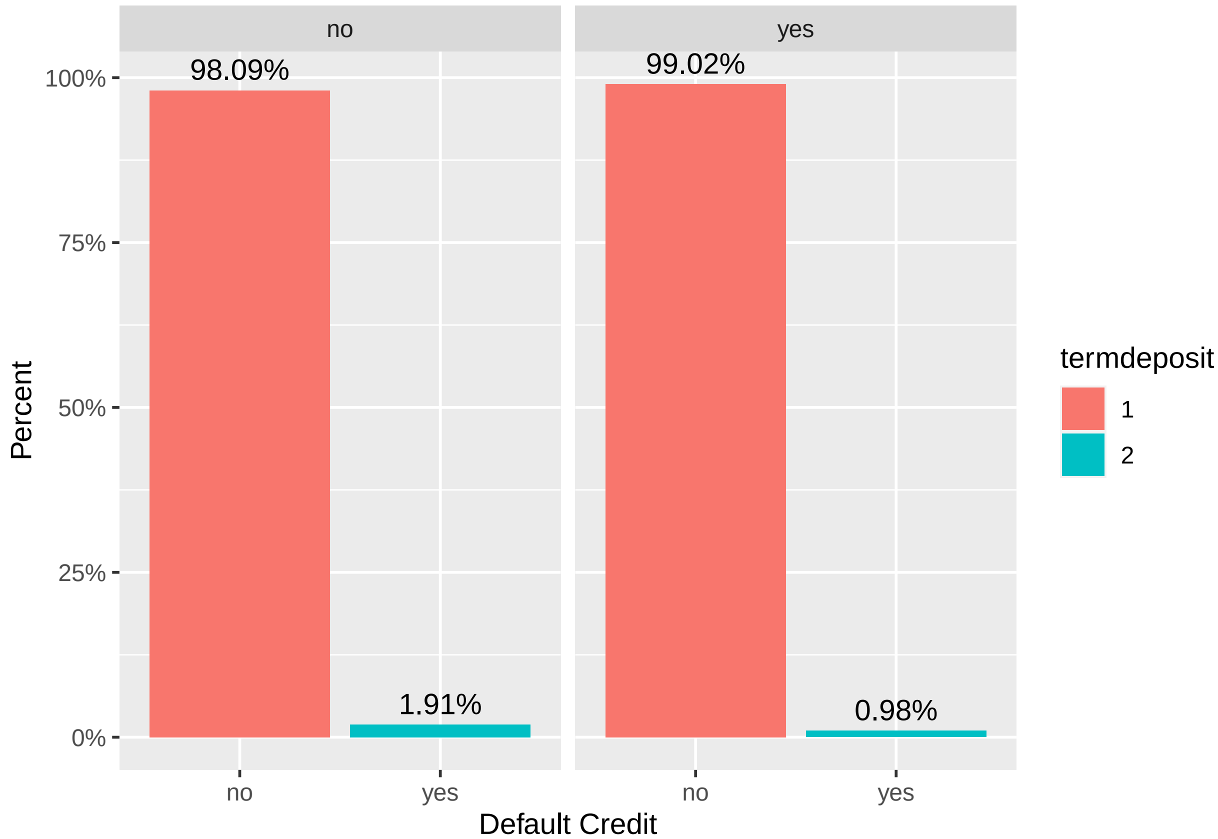
```
plot2
```



Considering the loan distribution of customers, we find that larger percentage of customers who subscribed to term deposit don't have housing or personal loan liability.

8.Distribution of default credit

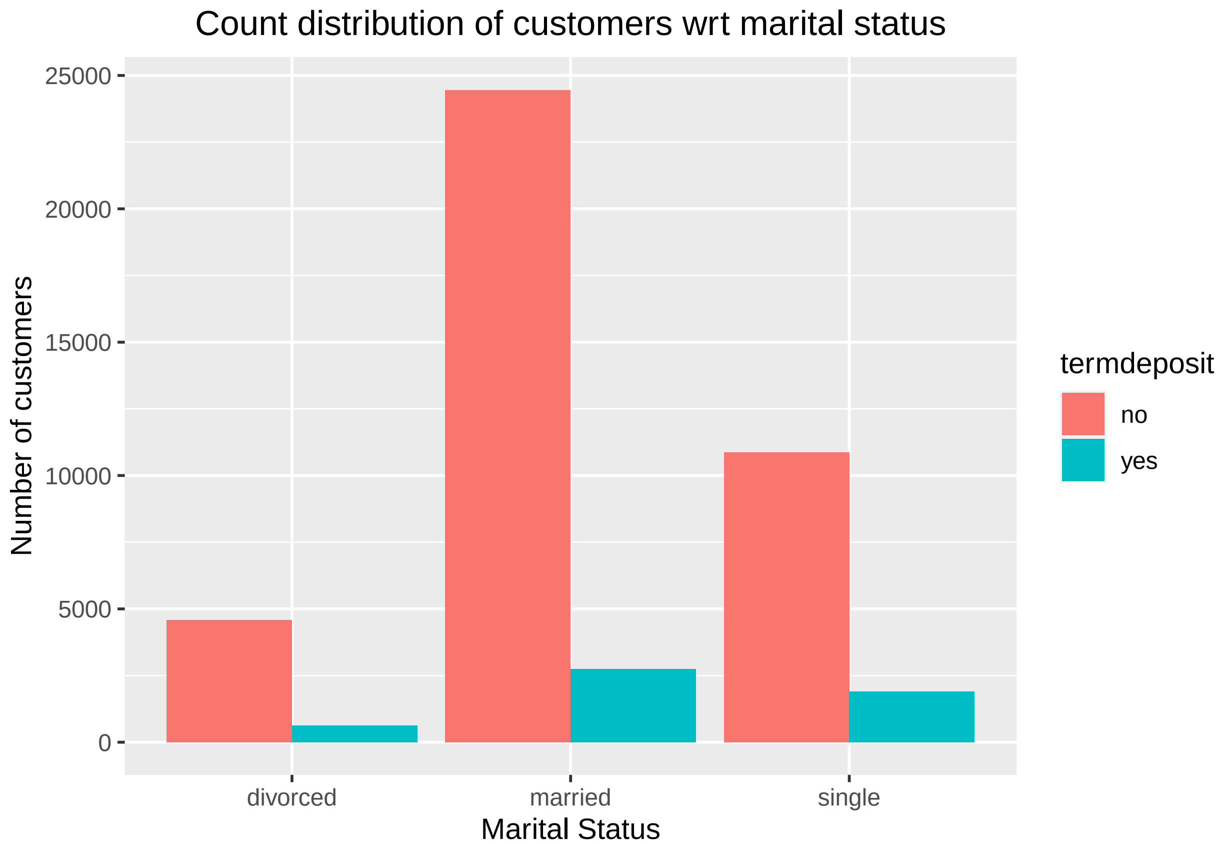
```
ggplot(bank_data, aes(x= default,  group=termdeposit)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
  geom_text(aes( label = scales::percent(..prop..),
                 y= ..prop.. ), stat= "count", vjust = -.5) +
  labs(y = "Percent", fill="termdeposit") +xlab('Default Credit')+
  facet_grid(~termdeposit) +
  scale_y_continuous(labels = scales::percent)
```



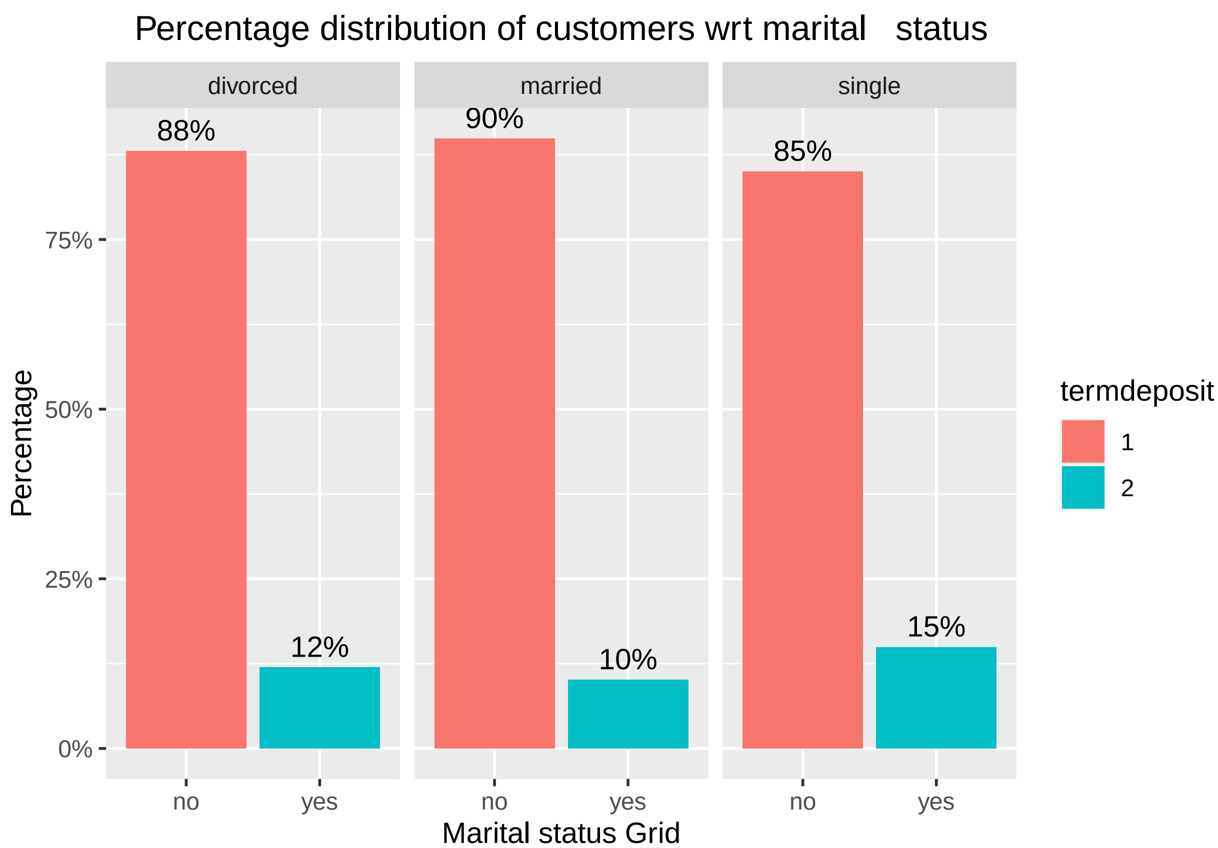
We can see that those who have credit in default have less chance of conversion compared to those to have no credit in default.

9.Distribution of marital status

```
plot1<-ggplot(bank_data, aes(x=marital, fill=termdeposit))+geom_bar(position='dodge')+ylab('Number of cu

plot2<-ggplot(bank_data, aes(x= termdeposit,  group=marital)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
  geom_text(aes( label = scales::percent(..prop..,1),
                 y= ..prop.. ), stat= "count", vjust = -.5) +
  labs(y = "Percentage", fill="termdeposit") +xlab('Marital status Grid')+
  facet_grid(~marital) +ggtitle('Percentage distribution of customers wrt marital   status')+theme(plot

  plot1
```


Count distribution of customers wrt marital status
25000
20000
15000
10000
5000
0
Number of customers
divorced
married
single
Marital Status
termdeposit
no
yes

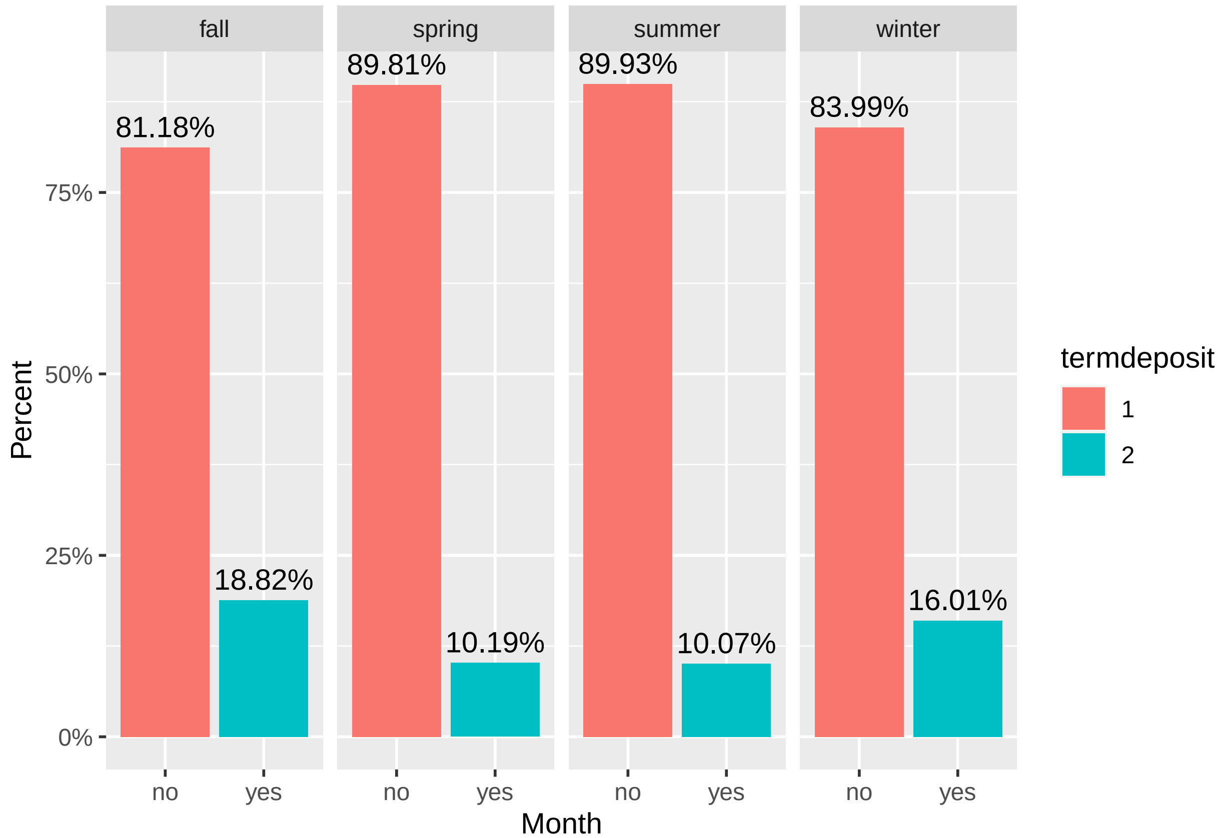
```
plot2
```



The marital distribution across customers shows that large number of customers contacted were married individuals. However, in terms of the percentage conversion, we find that non-married individuals actually subscribed to term deposit more.

10.Categorising months to seasons for understanding how distribution of term deposit in various seasons.

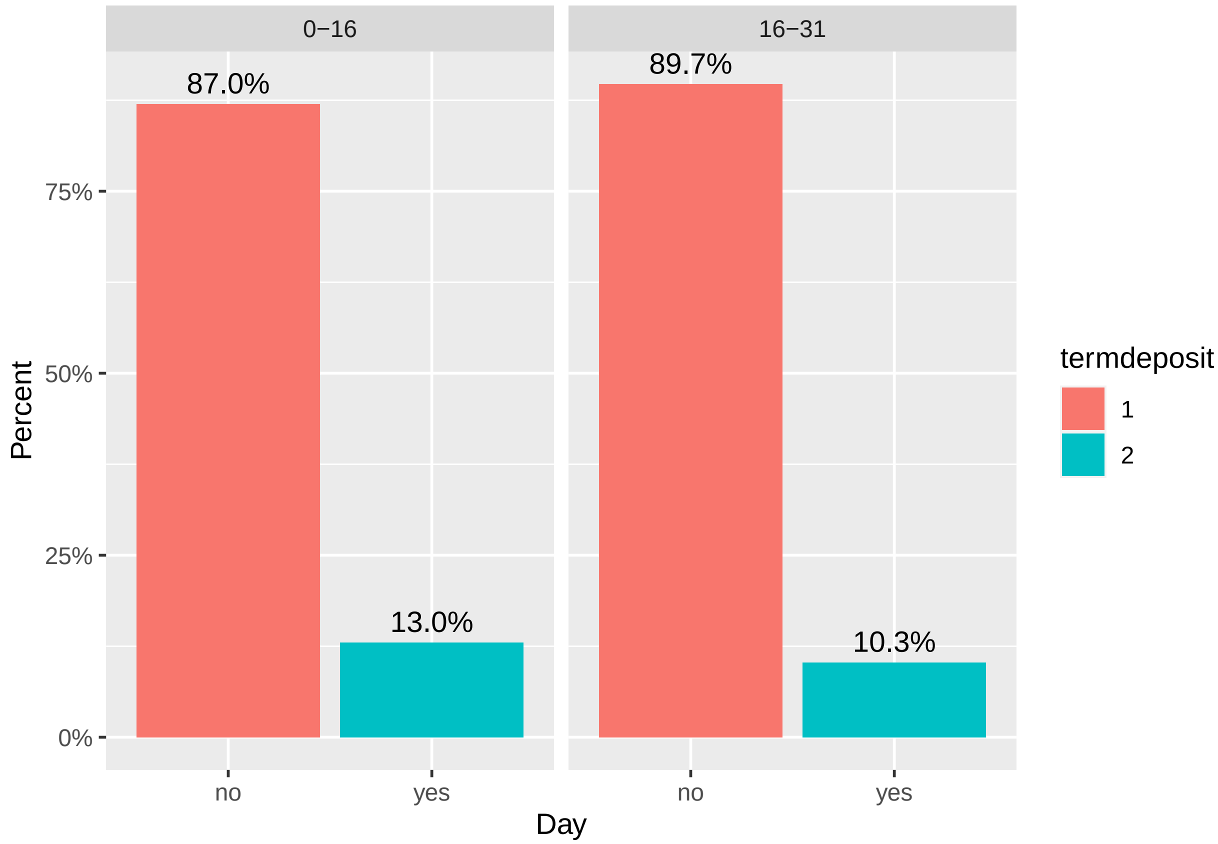
```
bank_data$season<- ifelse(bank_data$month=='jun'| bank_data$month=='jul'|bank_data$month=='aug', 'summe
                      ifelse(bank_data$month=='sep'|bank_data$month=='oct'             |bank_data$month

ggplot(bank_data, aes(x= termdeposit,  group=season)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
  geom_text(aes( label = scales::percent(..prop..),
                 y= ..prop.. ), stat= "count", vjust = -.5) +
  labs(y = "Percent", fill="termdeposit") +xlab('Month')+
  facet_grid(~season) +scale_y_continuous(labels = scales::percent)
```



By Segmenting the customers according to seasons, we find that larger percentage of customers have been contacted in spring and summer compared with winter and fall.

11.Categorising days to first 15 days and next 15 days

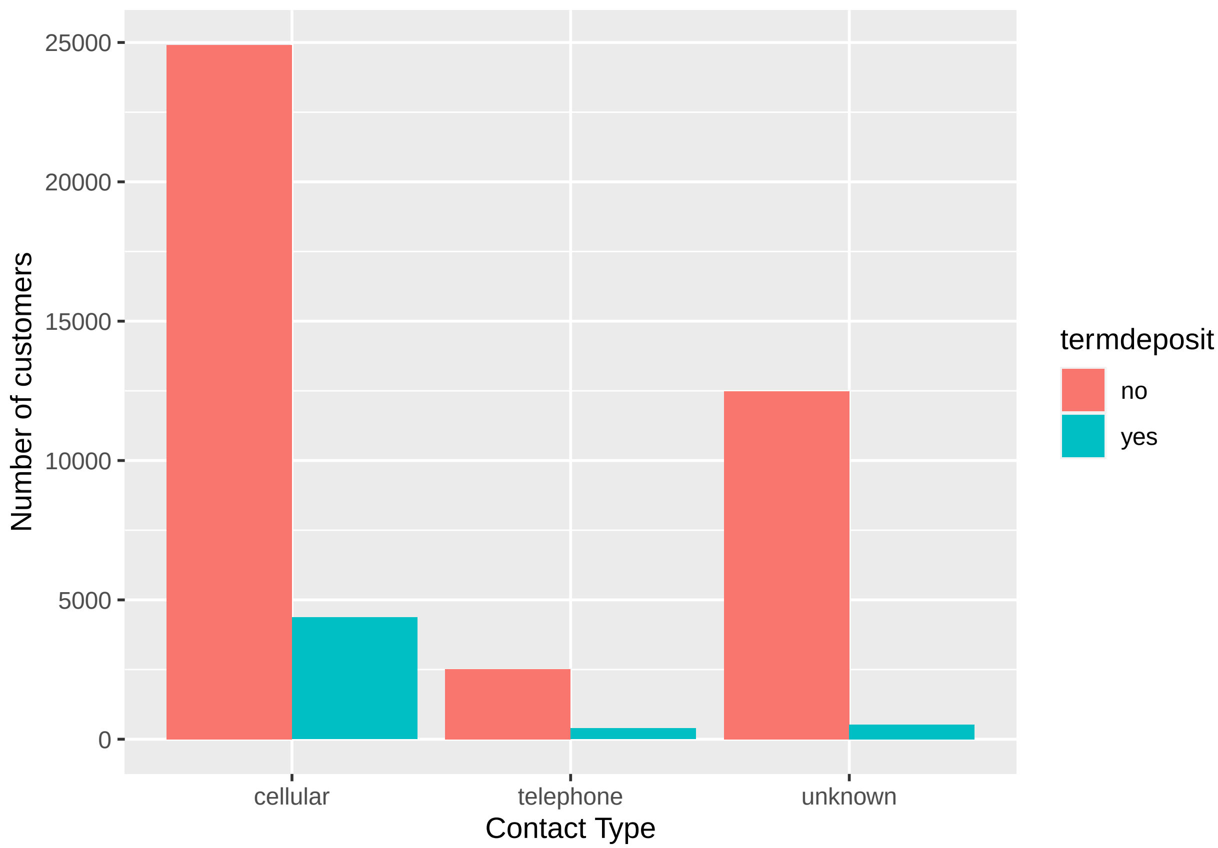
```
daybreaks<-c(0,16,31)
daylabels<-c('0-16','16-31')
bank_data$day_bin<-cut(bank_data$day,breaks = daybreaks, labels = daylabels, include.lowest = T)
ggplot(bank_data, aes(x= termdeposit,  group=day_bin)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
  geom_text(aes( label = scales::percent(..prop..),
                 y= ..prop.. ), stat= "count", vjust = -.5) +
  labs(y = "Percent", fill="termdeposit") +xlab('Day')+
  facet_grid(~day_bin) +scale_y_continuous(labels = scales::percent)
```



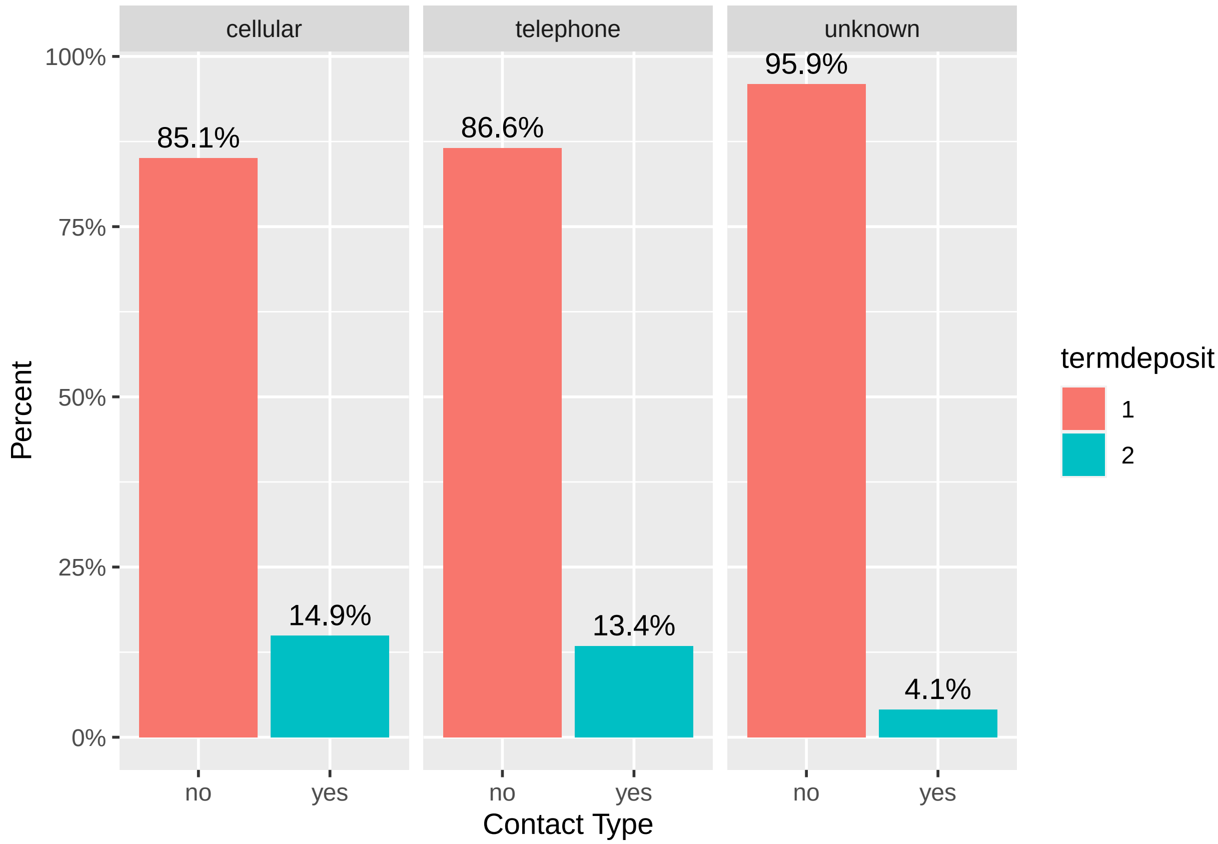
We see higher conversion of customers when contacted in the initial half of the month compared to next half of the month.

12.Distribution of contact type across customers

```
plot1<-ggplot(bank_data, aes(x=contact, fill=termdeposit))+geom_bar(position='dodge')+ylab('Number of cu
plot2<-ggplot(bank_data, aes(x= termdeposit,  group=contact)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat="count") +
  geom_text(aes( label = scales::percent(..prop..),
                 y= ..prop.. ), stat= "count", vjust = -.5) +
  labs(y = "Percent", fill="termdeposit") +xlab('Contact Type')+
  facet_grid(~contact) +
  scale_y_continuous(labels = scales::percent)
plot1
```


25000
20000
15000
10000
5000
0
Number of customers
cellular
telephone
unknown
Contact Type
termdeposit
no
yes

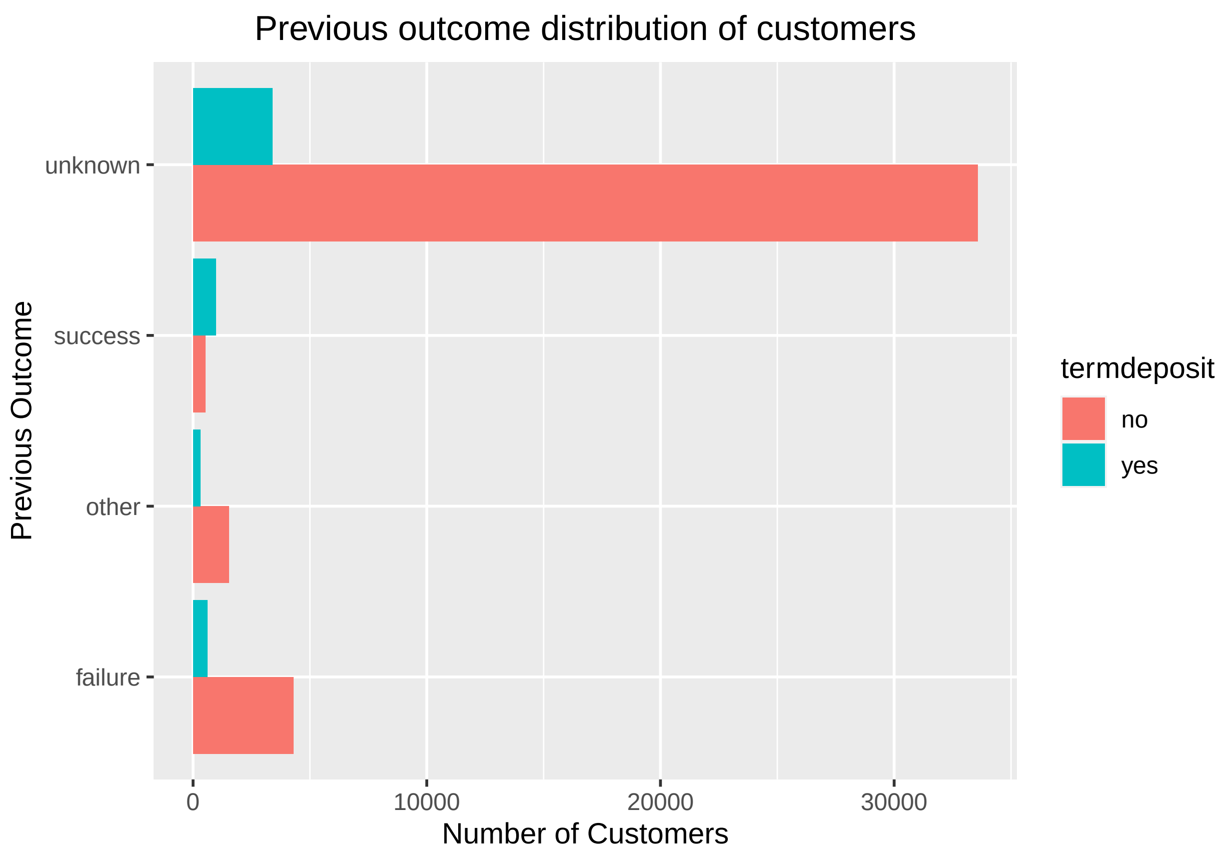
```
plot2
```



We can see that slightly conversion percentage of customers when contacted by cellular rather than telephone.

13. Distribution of previous outcome across customers

```
ggplot(bankfull1, aes(y=poutcome, fill=termdeposit))+geom_bar(position='dodge')+xlab('Number of Custome
```



According to the above graph, a large number of customers have previous outcome unknown which happen to be the newly contacted customers.

The distribution of New customers and previously contacted customers

```
bankfull1 %>% filter(pdays==-1) %>% summarise(count=n())
```

```
##   count
## 1 36954
```

```
bankfull1 %>% filter(pdays!=-1) %>% summarise(count=n())
```

```
##   count
## 1  8257
```

Modelling From our data we find that about 80% of the customers are new customers and the remaining are previously contacted customers. We ran logistic regression model to compare the newly contacted customers and previously contacted customers.

```
# Logistic Regression for newly contacted customers
bankfull1$termdeposit<-ifelse(bankfull1$termdeposit=='no',0,1)
bankfull1$season<- ifelse(bankfull1$month=='jun'| bankfull1$month=='jul'|bankfull1$month=='aug', 'summe
bankfull1=subset(bankfull1, select = -c(month))

bank_pdays<-bankfull1 %>% filter(pdays==-1)
```

```r
bank_pdays=subset(bank_pdays, select = -c(pdays, previous, poutcome, duration))

logit_pdays = glm(termdeposit~., family="binomial", data = bank_pdays)
summary(logit_pdays)
```

```
##
## Call:
## glm(formula = termdeposit ~ ., family = "binomial", data = bank_pdays)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3710  -0.4870  -0.3605  -0.2613   3.3569
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.205e+00  1.604e-01  -7.513 5.79e-14 ***
## age                1.629e-03  2.276e-03   0.716 0.474223
## jobblue-collar    -1.193e-01  7.496e-02  -1.591 0.111605
## jobentrepreneur   -2.055e-01  1.240e-01  -1.657 0.097520 .
## jobhousemaid      -2.640e-01  1.372e-01  -1.924 0.054410 .
## jobmanagement     -1.594e-01  7.726e-02  -2.064 0.039058 *
## jobretired         5.680e-01  9.837e-02   5.774 7.74e-09 ***
## jobself-employed  -1.522e-01  1.147e-01  -1.327 0.184462
## jobservices       -1.459e-01  8.670e-02  -1.682 0.092497 .
## jobstudent         3.951e-01  1.166e-01   3.387 0.000706 ***
## jobtechnician     -1.125e-01  7.205e-02  -1.561 0.118519
## jobunemployed     -4.729e-02  1.152e-01  -0.410 0.681486
## jobunknown        -3.823e-01  2.588e-01  -1.477 0.139650
## maritalmarried    -3.094e-01  5.839e-02  -5.299 1.17e-07 ***
## maritalsingle      9.238e-02  6.661e-02   1.387 0.165465
## educationsecondary 1.125e-01  6.420e-02   1.752 0.079744 .
## educationtertiary  3.254e-01  7.531e-02   4.320 1.56e-05 ***
## educationunknown   2.061e-01  1.059e-01   1.946 0.051672 .
## defaultyes        -1.828e-01  1.549e-01  -1.180 0.238005
## balance            1.778e-05  4.898e-06   3.631 0.000282 ***
## housingyes        -6.411e-01  4.414e-02 -14.526  < 2e-16 ***
## loanyes           -4.178e-01  5.928e-02  -7.048 1.82e-12 ***
## contacttelephone  -1.555e-01  7.416e-02  -2.096 0.036073 *
## contactunknown    -1.141e+00  5.349e-02 -21.327  < 2e-16 ***
## day               -1.149e-02  2.287e-03  -5.025 5.05e-07 ***
## campaign          -7.103e-02  8.978e-03  -7.911 2.55e-15 ***
## seasonspring       2.263e-01  6.405e-02   3.533 0.000411 ***
## seasonsummer      -4.030e-01  6.018e-02  -6.697 2.13e-11 ***
## seasonwinter      -2.036e-01  7.689e-02  -2.647 0.008112 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22628  on 36953  degrees of freedom
## Residual deviance: 20911  on 36925  degrees of freedom
## AIC: 20969
##
```

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

```r
# Logistic Regression for previously contacted customers
bank_no_pdays<-bankfull1 %>% filter(pdays!=-1)

logit_no_pdays = glm(termdeposit~., family="binomial", data = bank_no_pdays)
summary(logit_no_pdays)
```

```
##
## Call:
## glm(formula = termdeposit ~ ., family = "binomial", data = bank_no_pdays)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4436  -0.5121  -0.3238  -0.1773   2.7171
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.834e+00  2.957e-01  -9.581  < 2e-16 ***
## age                 2.595e-03  4.030e-03   0.644 0.519555
## jobblue-collar     -4.648e-01  1.350e-01  -3.444 0.000572 ***
## jobentrepreneur    -8.819e-01  2.728e-01  -3.233 0.001225 **
## jobhousemaid       -2.262e-01  2.542e-01  -0.890 0.373522
## jobmanagement       1.903e-02  1.283e-01   0.148 0.882122
## jobretired          8.051e-02  1.788e-01   0.450 0.652479
## jobself-employed   -3.025e-01  2.029e-01  -1.491 0.135931
## jobservices        -1.822e-01  1.532e-01  -1.189 0.234270
## jobstudent          2.862e-01  1.829e-01   1.564 0.117705
## jobtechnician      -2.257e-01  1.228e-01  -1.837 0.066206 .
## jobunemployed       2.357e-01  2.080e-01   1.133 0.257078
## jobunknown          2.079e-01  4.485e-01   0.464 0.642942
## maritalmarried      1.601e-01  1.135e-01   1.410 0.158475
## maritalsingle       2.402e-01  1.293e-01   1.858 0.063157 .
## educationsecondary  2.285e-01  1.269e-01   1.801 0.071712 .
## educationtertiary   4.059e-01  1.447e-01   2.805 0.005030 **
## educationunknown    3.178e-01  1.959e-01   1.622 0.104798
## defaultyes         -5.536e-01  5.389e-01  -1.027 0.304263
## balance             1.251e-05  9.983e-06   1.253 0.210133
## housingyes         -9.165e-01  7.653e-02 -11.975  < 2e-16 ***
## loanyes            -4.733e-01  1.181e-01  -4.006 6.17e-05 ***
## contacttelephone   -2.812e-01  1.347e-01  -2.088 0.036783 *
## contactunknown     -2.585e-01  3.694e-01  -0.700 0.484156
## day                 1.016e-02  4.082e-03   2.490 0.012783 *
## duration            3.658e-03  1.457e-04  25.117  < 2e-16 ***
## campaign           -1.175e-01  2.611e-02  -4.501 6.76e-06 ***
## pdays               9.007e-04  3.064e-04   2.940 0.003283 **
## previous            9.819e-03  6.301e-03   1.558 0.119144
## poutcomeother       2.717e-01  8.727e-02   3.113 0.001849 **
## poutcomesuccess     2.176e+00  7.957e-02  27.352  < 2e-16 ***
## poutcomeunknown     6.801e-01  9.991e-01   0.681 0.496059
## seasonspring       -3.834e-01  9.098e-02  -4.214 2.51e-05 ***
## seasonsummer        5.303e-01  9.980e-02   5.314 1.07e-07 ***
## seasonwinter       -3.608e-01  1.040e-01  -3.469 0.000523 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8919.8  on 8256  degrees of freedom
## Residual deviance: 5963.2  on 8222  degrees of freedom
## AIC: 6033.2
##
## Number of Fisher Scoring iterations: 5
```

Interpretation: Here we can find that in case of new customers, targeting retired and students can help the bank increase its subscription as both the job groups have possibility of reduced liability compared to other job groups.Similarly new customers have better probability of conversion if contacted in the spring season. In case of the previously contacted customers, singles have a better chance of conversion as due to lack of spouse support, they are likely to save money in term deposit.Similarly previously contacted customers have better success in the summer season. In case of both customer groups, increased education level corresponds to better conversion.

Going Forward we wanted to study the impact of explanatory variables for the complete data set. The initial model we ran didn't show significance for the age variable. Hence we binned the age variable . In case of duration it was having values between 0-5000 secs. We converted it to minutes.We got values between 0-90 minutes.We also found very high z-score for the duration variable.Hence we categorised the duration to different bins to see if the tail drivers the results for the prediction. In the case balance we normalised it due to large scale.

```
# Complete data set
                                                                                                        bankfull
bankfull1$marital<-as.factor(bankfull1$marital)
bankfull1$season<-as.factor(bankfull1$season)
bankfull1$job<-replace(bankfull1$job,bankfull1$job=='self-employed','selfemployed' )
bankfull1$job<-replace(bankfull1$job,bankfull1$job=='blue-collar','bluecollar' )
bankfull1$job<-as.factor(bankfull1$job)
bankfull1$education<-as.factor(bankfull1$education)
bankfull1$poutcome<-as.factor(bankfull1$poutcome)
bankfull1$housing<-ifelse(bankfull1$housing=='no',0,1)
bankfull1$loan<-ifelse(bankfull1$loan=='no',0,1)
bankfull1$default<-ifelse(bankfull1$default=='no',0,1)
bankfull1$balance <- (bankfull1$balance - mean(bankfull1$balance)) / sd(bankfull1$balance)
bankfull1$duration <- bankfull1$duration/60

durationbreaks<-c(0,5,10,15,20,100)
durationlabels<-c('0-5','5-10','10-15','15-20','20-100')
bankfull1$duration_bin<-cut(bankfull1$duration,breaks = durationbreaks,
                            labels = durationlabels, include.lowest = T)

agebreaks<-c(15,30,45,60,75,105)
agelabels<-c('15-30','30-45','45-60','60-75','75-105')
bankfull1$age_bin<-cut(bankfull1$age,breaks = agebreaks, labels = agelabels, include.lowest = T)

bankfull1=subset(bankfull1, select = -c(age,duration))
bankfull1 <- bankfull1 %>% select( -termdeposit, termdeposit)

ind<-sample(2, nrow(bankfull1), replace=T, prob = c(0.7,0.3))
train<-bankfull1[ind==1,]
```

```r
test<-bankfull1[ind==2,]

logit = glm(termdeposit~.
            , family="binomial", data = train)
summary(logit)
```

```
##
## Call:
## glm(formula = termdeposit ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1865  -0.3812  -0.2580  -0.1562   3.1181
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.657e+00  1.804e-01  -9.183  < 2e-16 ***
## jobbluecollar        -3.883e-01  8.389e-02  -4.629 3.68e-06 ***
## jobentrepreneur      -5.575e-01  1.473e-01  -3.785 0.000154 ***
## jobhousemaid         -4.048e-01  1.573e-01  -2.574 0.010066 *
## jobmanagement        -2.471e-01  8.630e-02  -2.863 0.004193 **
## jobretired           -1.565e-01  1.267e-01  -1.236 0.216576
## jobselfemployed      -4.591e-01  1.316e-01  -3.488 0.000486 ***
## jobservices          -3.780e-01  9.874e-02  -3.828 0.000129 ***
## jobstudent            3.595e-01  1.296e-01   2.774 0.005537 **
## jobtechnician        -2.793e-01  8.077e-02  -3.458 0.000544 ***
## jobunemployed        -2.199e-01  1.316e-01  -1.670 0.094841 .
## jobunknown           -7.144e-01  3.057e-01  -2.337 0.019442 *
## maritalmarried       -1.039e-01  6.955e-02  -1.494 0.135133
## maritalsingle         9.060e-02  7.877e-02   1.150 0.250071
## educationsecondary    1.751e-01  7.522e-02   2.328 0.019932 *
## educationtertiary     4.679e-01  8.820e-02   5.305 1.13e-07 ***
## educationunknown      2.257e-01  1.226e-01   1.841 0.065655 .
## default              -2.230e-01  2.009e-01  -1.110 0.267152
## balance               4.178e-02  1.800e-02   2.321 0.020286 *
## housing              -8.081e-01  5.080e-02 -15.909  < 2e-16 ***
## loan                 -5.163e-01  6.936e-02  -7.443 9.81e-14 ***
## contacttelephone     -1.867e-01  8.894e-02  -2.099 0.035834 *
## contactunknown       -1.109e+00  6.828e-02 -16.250  < 2e-16 ***
## day                  -5.533e-03  2.621e-03  -2.111 0.034803 *
## campaign             -9.970e-02  1.203e-02  -8.286  < 2e-16 ***
## pdays                -5.473e-05  3.623e-04  -0.151 0.879953
## previous              3.031e-02  1.122e-02   2.702 0.006894 **
## poutcomeother         2.214e-01  1.058e-01   2.091 0.036498 *
## poutcomesuccess       2.320e+00  9.647e-02  24.048  < 2e-16 ***
## poutcomeunknown      -2.027e-01  1.138e-01  -1.781 0.074838 .
## seasonspring          3.603e-02  6.882e-02   0.523 0.600656
## seasonsummer         -2.511e-01  6.670e-02  -3.765 0.000166 ***
## seasonwinter         -2.757e-01  8.177e-02  -3.372 0.000747 ***
## duration_bin5-10      1.424e+00  4.980e-02  28.600  < 2e-16 ***
## duration_bin10-15     2.906e+00  6.633e-02  43.801  < 2e-16 ***
## duration_bin15-20     3.721e+00  9.940e-02  37.432  < 2e-16 ***
## duration_bin20-100    4.041e+00  1.218e-01  33.187  < 2e-16 ***
```

```
## age_bin30-45       -2.913e-01  6.250e-02  -4.661 3.15e-06 ***
## age_bin45-60       -3.458e-01  7.596e-02  -4.553 5.30e-06 ***
## age_bin60-75        8.096e-01  1.434e-01   5.644 1.66e-08 ***
## age_bin75-105       1.011e+00  2.268e-01   4.460 8.21e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22581  on 31583  degrees of freedom
## Residual deviance: 15534  on 31543  degrees of freedom
## AIC: 15616
##
## Number of Fisher Scoring iterations: 6
```

Interpretation Here we can see that Job student has positive significance compared to Job admin and the rest of the jobs have negative significance with respect to job admin.Similarly if customers have high bank balance, there are high chances they will be converted. Considering negative significance, individuals who are married, have housing/personal loan and those whose is less than 60 have less probability of conversion to name a few.

```r
# ln-likelihood
yActual = test$termdeposit #get the actual value for the choice variable
predTst_logit = predict(logit, test, type="response")
#use the model results in blTrn_basic, to predict the probability of Y=1 for each data poi
lnlike_logit = sum(log(predTst_logit*yActual+(1-predTst_logit)* (1-yActual)))
lnlike_logit
```

```
## [1] -3416.106
```

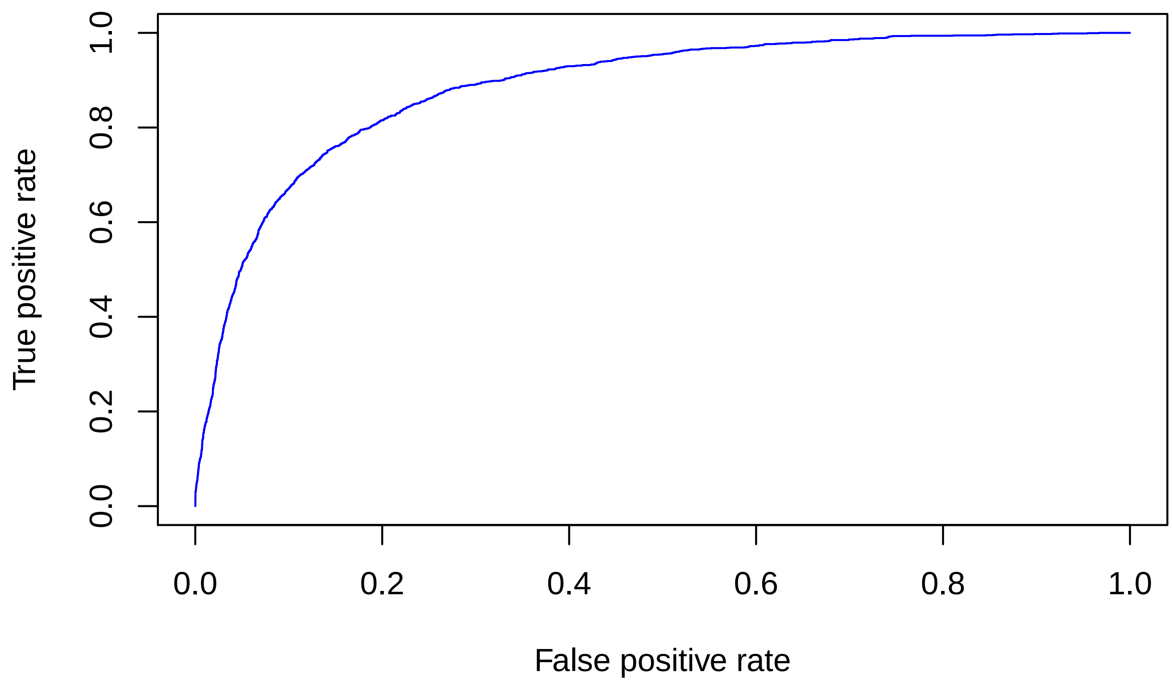
Confusion matrix

```r
# threshold (0.5) for categorizing predicted probabilities
predFac <- ifelse(predTst_logit<0.5, 0, 1)
table(predFac, test$termdeposit)
```

```
##
## predFac     0     1
##       0 11666  1082
##       1   314   565
```

ROC curve

```r
pred_logit <- prediction(as.numeric(predTst_logit), as.numeric(yActual))
perf_logit <- performance(pred_logit,"tpr","fpr")
plot(perf_logit,col='blue')
```



AUC Score

```
perf_auc_logit <- performance(pred_logit,measure="auc")
print(paste("AUC= ", perf_auc_logit@y.values[[1]]))
```

```
## [1] "AUC=  0.884923313800763"
```

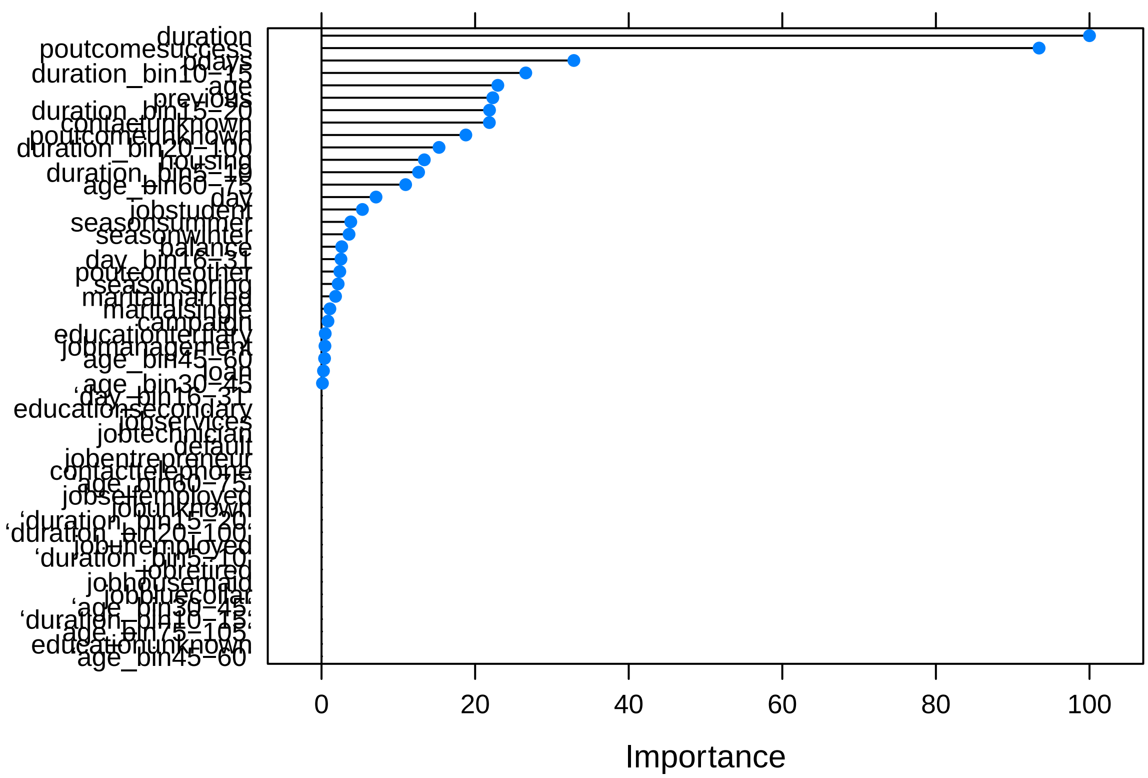
Preparing data for the other machine learning models viz:Decision tree, bagging, XGboost and random forest

```
bankfull2<-bank_data

bankfull2$season<- ifelse(bankfull2$month=='jun'| bankfull2$month=='jul'|bankfull2$month=='aug', 'summe


bankfull2$contact<-as.factor(bankfull2$contact)
bankfull2$marital<-as.factor(bankfull2$marital)
bankfull2$season<-as.factor(bankfull2$season)
bankfull2$job<-replace(bankfull2$job,bankfull2$job=='self-employed','selfemployed' )
bankfull2$job<-replace(bankfull2$job,bankfull2$job=='blue-collar','bluecollar' )
bankfull2$job<-as.factor(bankfull2$job)
bankfull2$education<-as.factor(bankfull2$education)
bankfull2$poutcome<-as.factor(bankfull2$poutcome)


bankfull2$housing<-ifelse(bankfull2$housing=='no',0,1)
```

```r
bankfull2$loan<-ifelse(bankfull2$loan=='no',0,1)
bankfull2$default<-ifelse(bankfull2$default=='no',0,1)
bankfull2$balance <- (bankfull2$balance - mean(bankfull2$balance)) / sd(bankfull2$balance)

bankfull2$termdeposit<-ifelse(bankfull2$termdeposit=='no',0,1)
bankfull2$termdeposit<-as.factor(bankfull2$termdeposit)

bankfull2=subset(bankfull2, select = -c(month))
bankfull2 <- bankfull2 %>% select( -termdeposit, termdeposit)

ind<-sample(2, nrow(bankfull2), replace=T, prob = c(0.7,0.3))
train_tree<-bankfull2[ind==1,]
test_tree<-bankfull2[ind==2,]
```

Decision tree with cross validation

```r
set.seed(123)
cv<-caret::trainControl(method='repeatedcv', number=10, repeats=5, allowParallel=T)

tree_cv<-caret::train(termdeposit~., data=train_tree, method='rpart', trControl=cv,
                      tuneLength=10)
plot(varImp(tree_cv))
```



Here from the plot we see that the duration variable is the most important variable determining the success of the marketing campaign. However duration not being a deterministic variable, the next variable i.e previous outcome success can be used as the next important variable to target the customers.

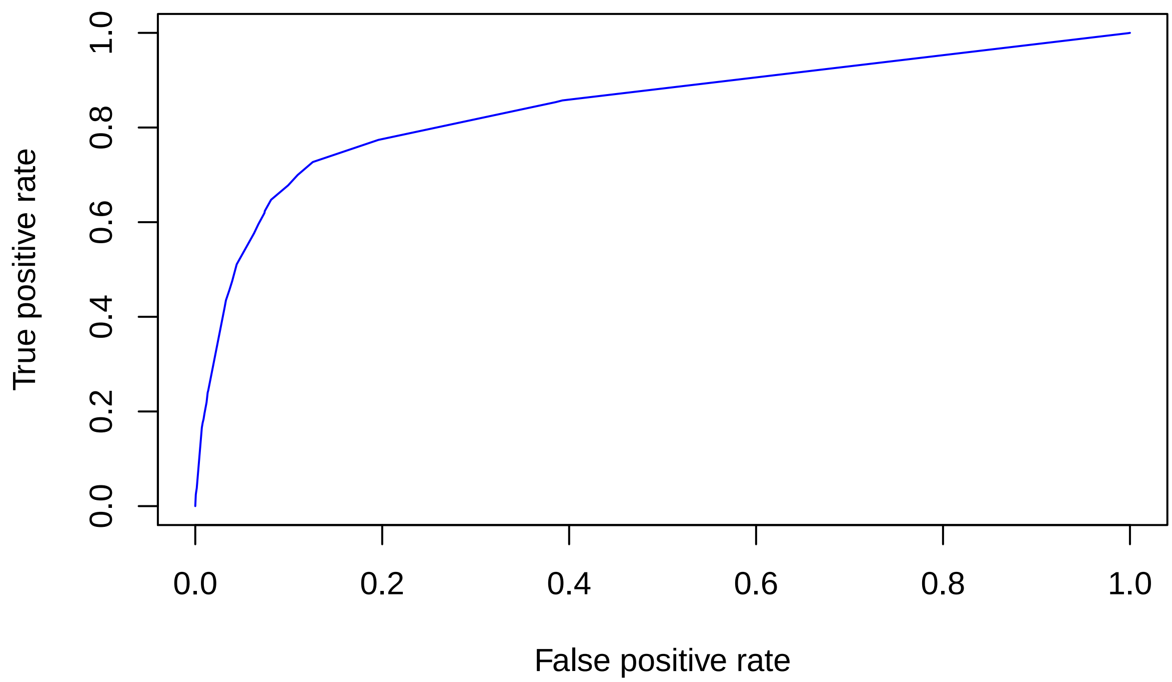
Confusion matrix

```
p_cv<-predict(tree_cv, newdata = test_tree, type='raw')
caret::confusionMatrix(p_cv, test_tree$termdeposit)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11391   872
##          1   432   733
##
##                Accuracy : 0.9029
##                  95% CI : (0.8978, 0.9078)
##     No Information Rate : 0.8805
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4766
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9635
##             Specificity : 0.4567
##          Pos Pred Value : 0.9289
##          Neg Pred Value : 0.6292
##              Prevalence : 0.8805
##          Detection Rate : 0.8483
##    Detection Prevalence : 0.9132
##       Balanced Accuracy : 0.7101
##
##        'Positive' Class : 0
##
```

ROC curve

```
yActual = test_tree$termdeposit #get the actual value for the choice variable
predTst_tree_cv = predict(tree_cv, test_tree, type="prob")

pred_tree_cv <- prediction(as.numeric(predTst_tree_cv[,2]), as.numeric(yActual))
perf_tree_cv <- performance(pred_tree_cv,"tpr","fpr")
plot(perf_tree_cv,col='blue')
```



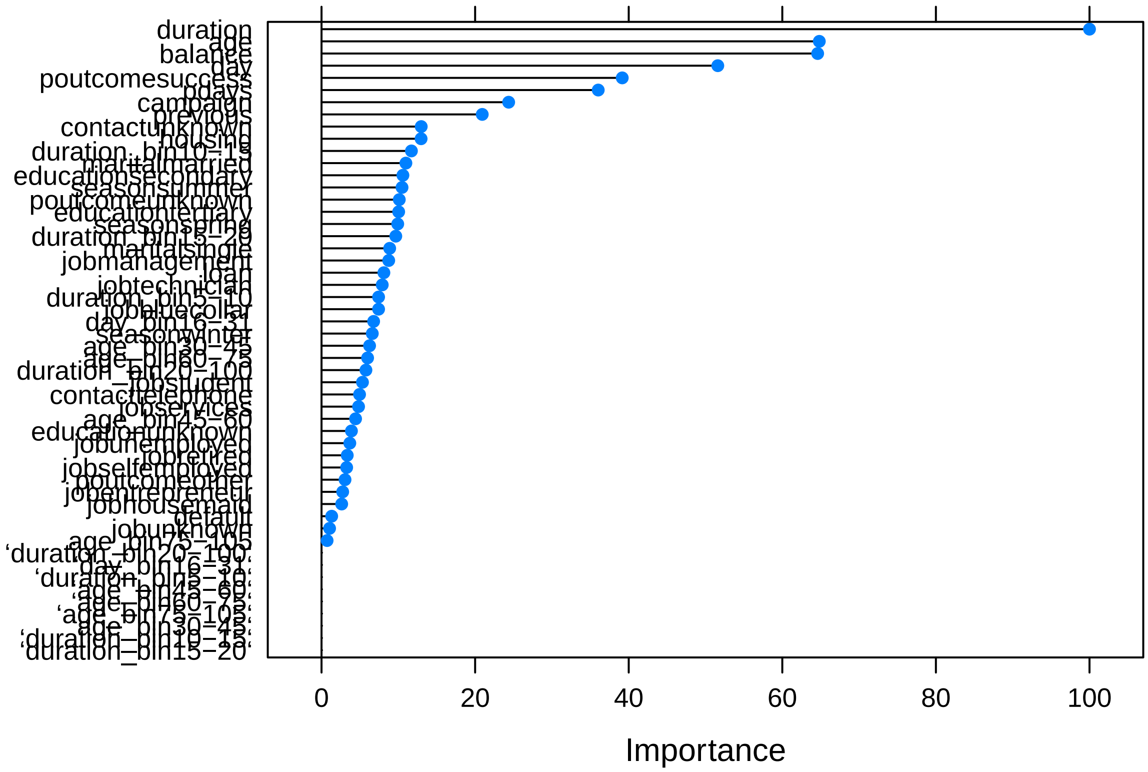
AUC score

```
perf_auc_tree_cv <- performance(pred_tree_cv,measure="auc")
print(paste("AUC= ", perf_auc_tree_cv@y.values[[1]]))
```

```
## [1] "AUC=  0.842736937849901"
```

Bagging with decision tree cross validation

```
set.seed(1234)
bag<-caret::train(termdeposit~., data=train_tree, method='treebag', trControl=cv,importance=T)
plot(varImp(bag))
```



Confusion matrix

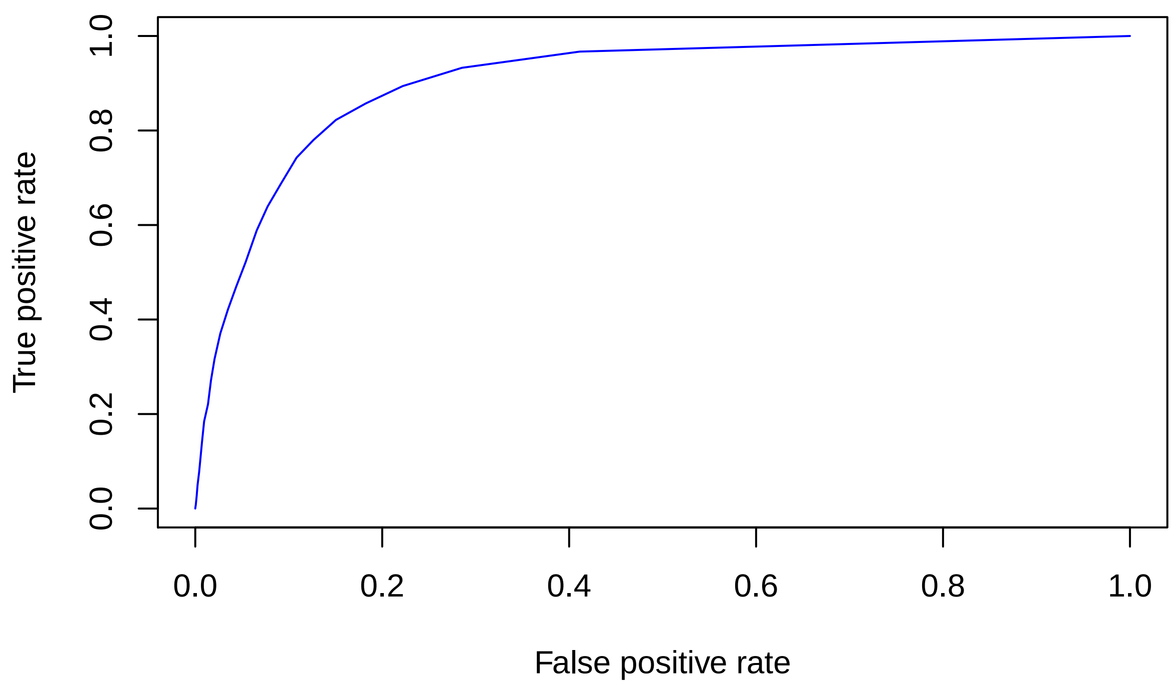
```r
p_bag<-predict(bag, newdata = test_tree, type='raw')
caret::confusionMatrix(p_bag, test_tree$termdeposit)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11410   929
##          1   413   676
##
##                Accuracy : 0.9001
##                  95% CI : (0.8949, 0.9051)
##     No Information Rate : 0.8805
##     P-Value [Acc > NIR] : 4.127e-13
##
##                   Kappa : 0.4486
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9651
##             Specificity : 0.4212
##          Pos Pred Value : 0.9247
##          Neg Pred Value : 0.6208
##              Prevalence : 0.8805
```

```
##          Detection Rate : 0.8497
##    Detection Prevalence : 0.9189
##       Balanced Accuracy : 0.6931
##
##        'Positive' Class : 0
##
```

ROC curve

```r
yActual = test_tree$termdeposit #get the actual value for the choice variable
predTst_tree_bag = predict(bag, test_tree, type="prob")

pred_tree_bag <- prediction(as.numeric(predTst_tree_bag[,2]), as.numeric(yActual))
perf_tree_bag <- performance(pred_tree_bag,"tpr","fpr")
plot(perf_tree_bag,col='blue')
```



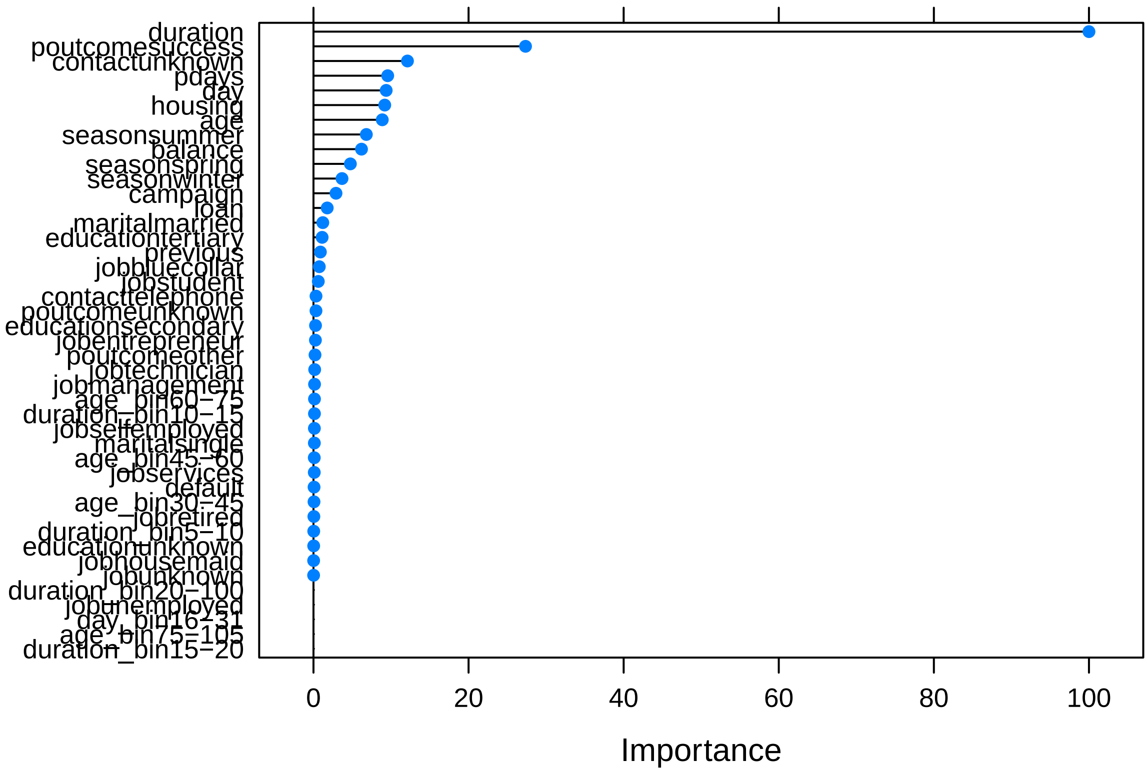
AUC score

```r
perf_auc_tree_bag <- performance(pred_tree_bag,measure="auc")
print(paste("AUC= ", perf_auc_tree_bag@y.values[[1]]))
```

```
## [1] "AUC=  0.904016064574488"
```

Xtreme Gradient boost with cross validation

```r
set.seed(1234)
boost<-caret::train(termdeposit~., data=train_tree, method='xgbTree', trControl=cv,
                    tuneGrid=expand.grid(nrounds=200, max_depth=3, eta=0.2,
                                         gamma=0.01, colsample_bytree=1,
                                         min_child_weight=1,subsample=1))
plot(varImp(boost))
```



Confusion matrix

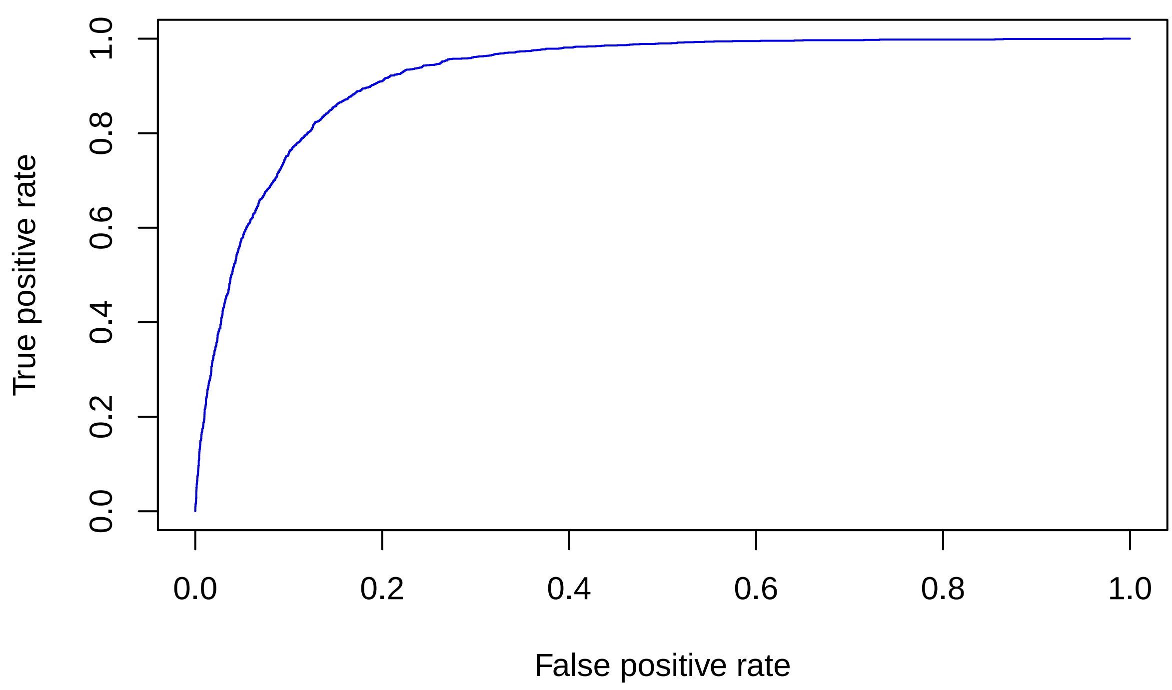
```r
p_boost<-predict(boost, newdata = test_tree, type='raw')
caret::confusionMatrix(p_boost, test_tree$termdeposit)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11482   940
##          1   341   665
##
##                Accuracy : 0.9046
##                  95% CI : (0.8995, 0.9095)
##     No Information Rate : 0.8805
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4596
```

```
##
## Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9712
##             Specificity : 0.4143
##          Pos Pred Value : 0.9243
##          Neg Pred Value : 0.6610
##              Prevalence : 0.8805
##          Detection Rate : 0.8551
##    Detection Prevalence : 0.9251
##       Balanced Accuracy : 0.6927
##
##        'Positive' Class : 0
##
```

ROC curve

```
yActual = test_tree$termdeposit #get the actual value for the choice variable
predTst_tree_boost = predict(boost, test_tree, type="prob")

pred_tree_boost <- prediction(as.numeric(predTst_tree_boost[,2]), as.numeric(yActual))
perf_tree_boost <- performance(pred_tree_boost,"tpr","fpr")
plot(perf_tree_boost,col='blue')
```



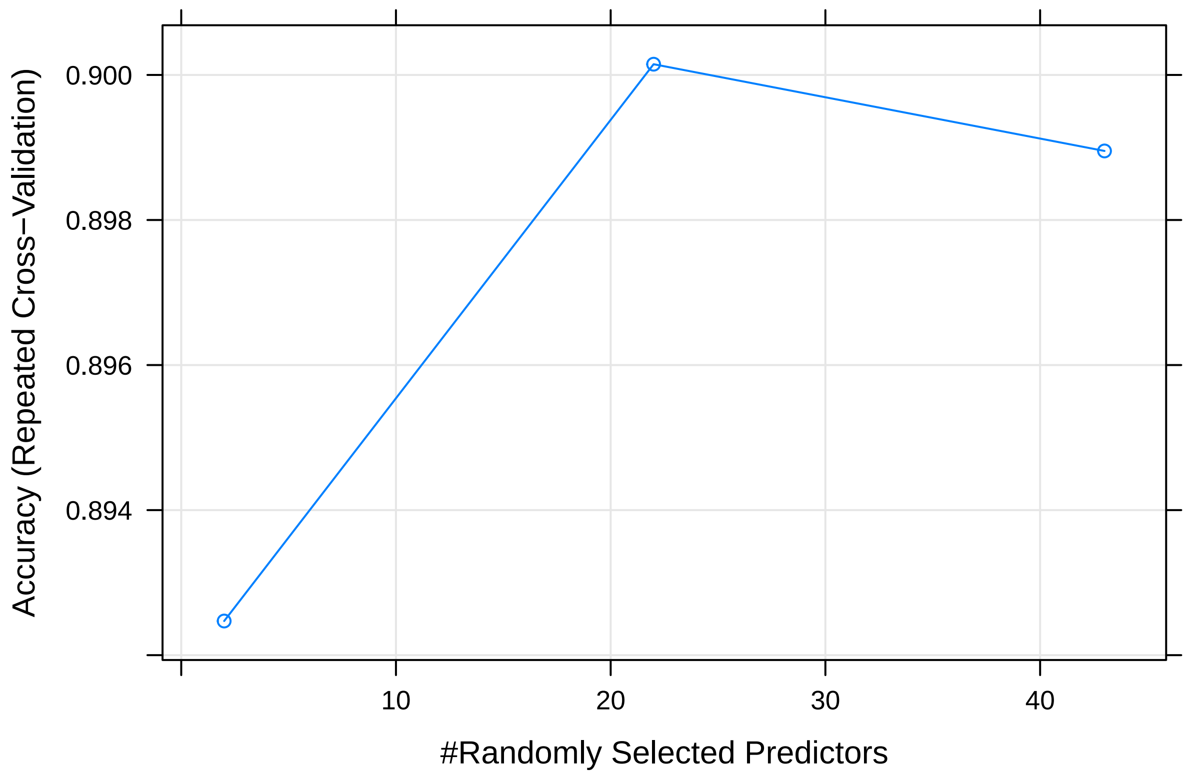
AUC score

```r
perf_auc_tree_boost <- performance(pred_tree_boost,measure="auc")
print(paste("AUC= ", perf_auc_tree_boost@y.values[[1]]))
```

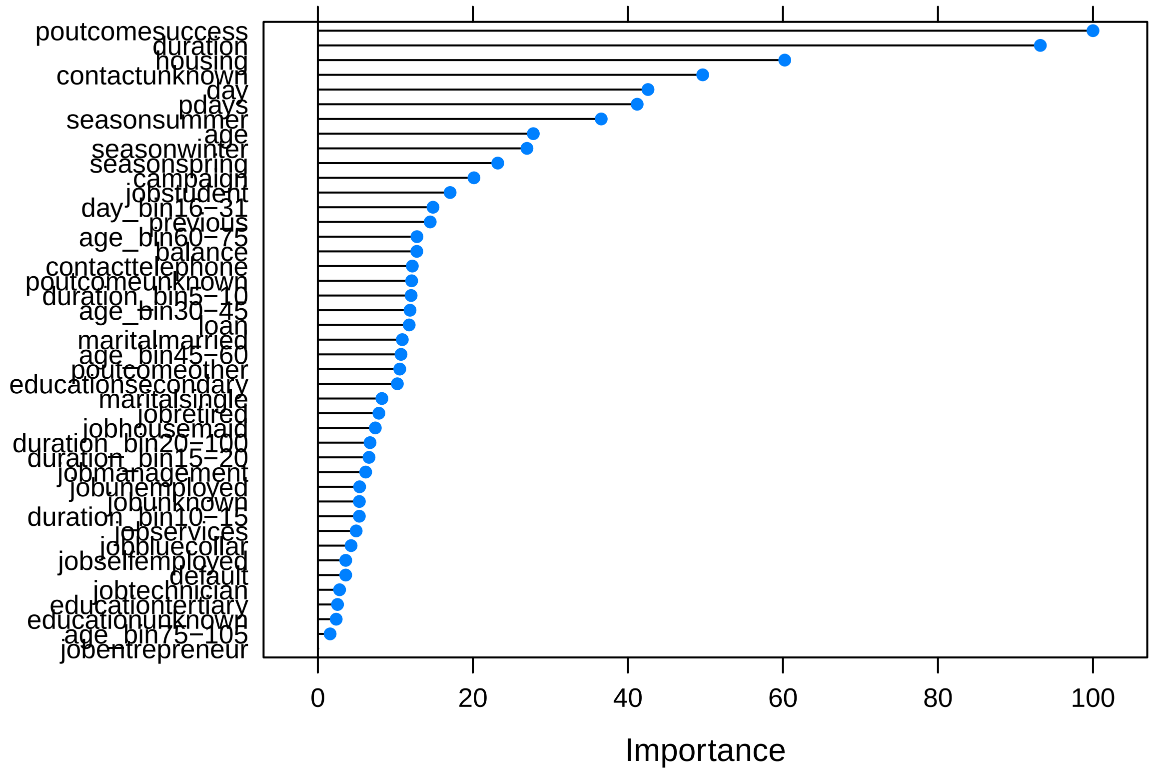
```
## [1] "AUC=  0.924922197427686"
```

Random Forest with cross validation

```r
set.seed(1234)
rf<-caret::train(termdeposit~., data=train_tree, method='rf', trControl=cv,
                 importance=T, ntree=20)
plot(rf)
```




```r
plot(varImp(rf))
```



Confusion matrix

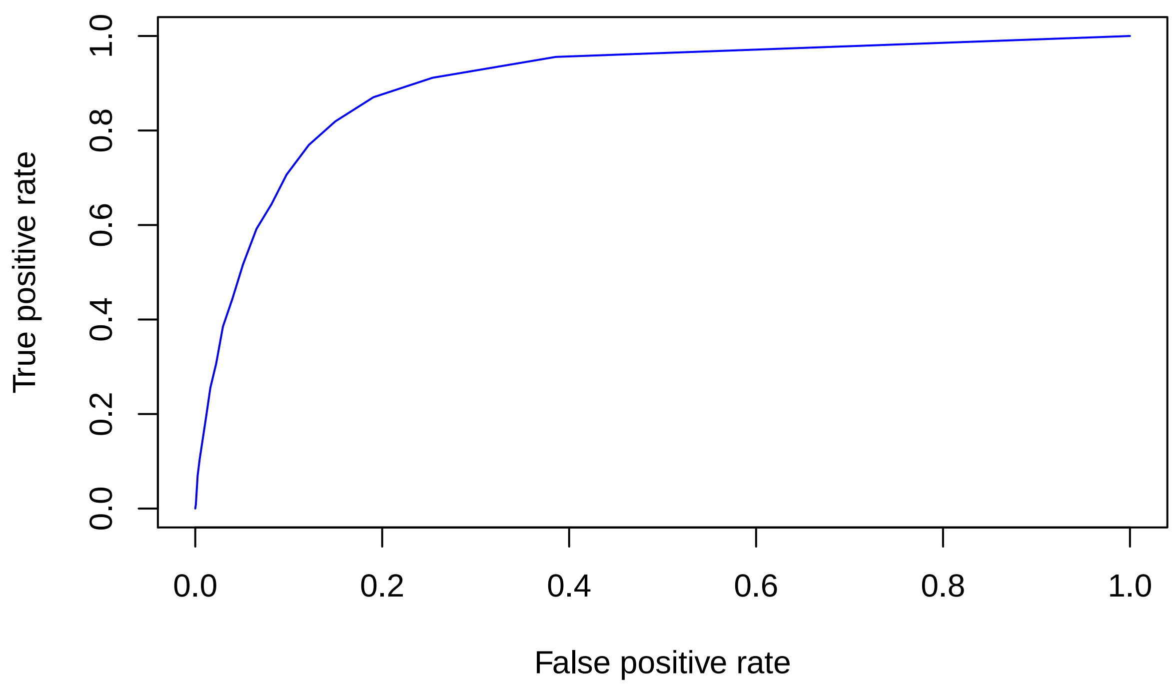
```r
p_rf<-predict(rf, newdata = test_tree, type='raw')
caret::confusionMatrix(p_rf, test_tree$termdeposit)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11438   953
##          1   385   652
##
##                Accuracy : 0.9004
##                  95% CI : (0.8952, 0.9054)
##     No Information Rate : 0.8805
##     P-Value [Acc > NIR] : 1.812e-13
##
##                   Kappa : 0.4411
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9674
##             Specificity : 0.4062
##          Pos Pred Value : 0.9231
##          Neg Pred Value : 0.6287
##              Prevalence : 0.8805
```

```
##          Detection Rate : 0.8518
##    Detection Prevalence : 0.9228
##       Balanced Accuracy : 0.6868
##
##        'Positive' Class : 0
##
```

ROC curve

```
yActual = test_tree$termdeposit
predTst_rf = predict(rf, test_tree, type="prob")

pred_rf <- prediction(as.numeric(predTst_rf[,2]), as.numeric(yActual))
perf_rf <- performance(pred_rf,"tpr","fpr")
plot(perf_rf,col='blue')
```



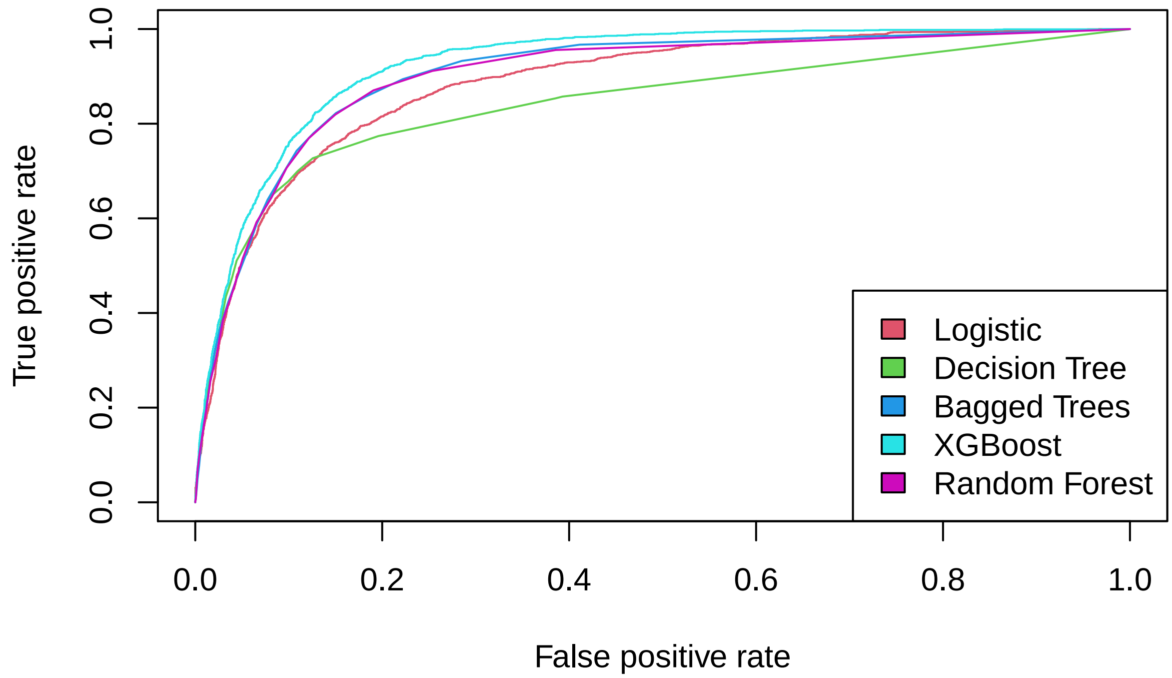
AUC score

```
perf_auc_rf <- performance(pred_rf,measure="auc")
print(paste("AUC= ", perf_auc_rf@y.values[[1]]))
```

```
## [1] "AUC=  0.89970415128862"
```

Combining ROC curves for all the models

```
plot(perf_logit, col=(2))
plot(perf_tree_cv,add=T, col=(3))
plot(perf_tree_bag,add=T, col=(4))
plot(perf_tree_boost,add=T, col=(5))
plot(perf_rf,add=T, col=(6))
legend(x='bottomright', legend=c('Logistic','Decision Tree', 'Bagged Trees', 'XGBoost', 'Random Forest'
```



We get XGboost as our best model. This is further substantiated by the light blue in the ROC curve .

From the table we can see that the difference between people who deposit and who don't is really high, this data set is a highly imbalanced data. To deal with the imbalanced problem, we use SMOTE to make it into a rather balance dataset. SMOTE is a well-known algorithm to deal with imbalanced data. This generates new examples of the minority class and undersamples the majority class examples.

```
#Split into test and train
ind<-sample(2, nrow(bankfull2), replace=T, prob = c(0.7,0.3))
train<-bankfull2[ind==1,]
test<-bankfull2[ind==2,]
table(train$termdeposit)
```

```
##
##     0     1
## 28036  3684
```

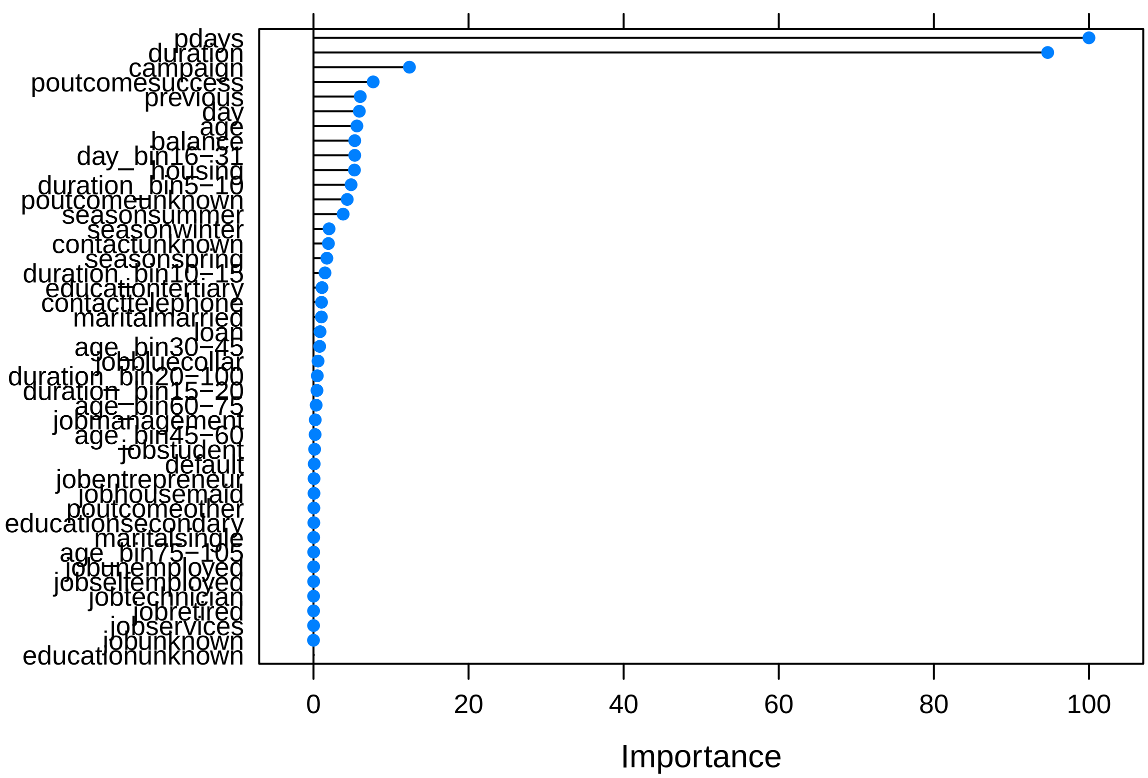
SMOTE technique for imbalanced data

```r
set.seed(1234)
bankfull_smote<-SMOTE(termdeposit~., data=train, perc.over=300, perc.under = 140)
table(bankfull_smote$termdeposit)
```

```
##
##     0     1
## 15472 14736
```

SMOTE technique on Xtreme Gradient boost with cross validation(best model)

```r
set.seed(1234)
boost<-caret::train(termdeposit~., data=bankfull_smote, method='xgbTree', trControl=cv,
                    tuneGrid=expand.grid(nrounds=200, max_depth=3, eta=0.2,
                                         gamma=0.01, colsample_bytree=1,
                                         min_child_weight=1,subsample=1))
plot(varImp(boost))
```



Confusion matrix

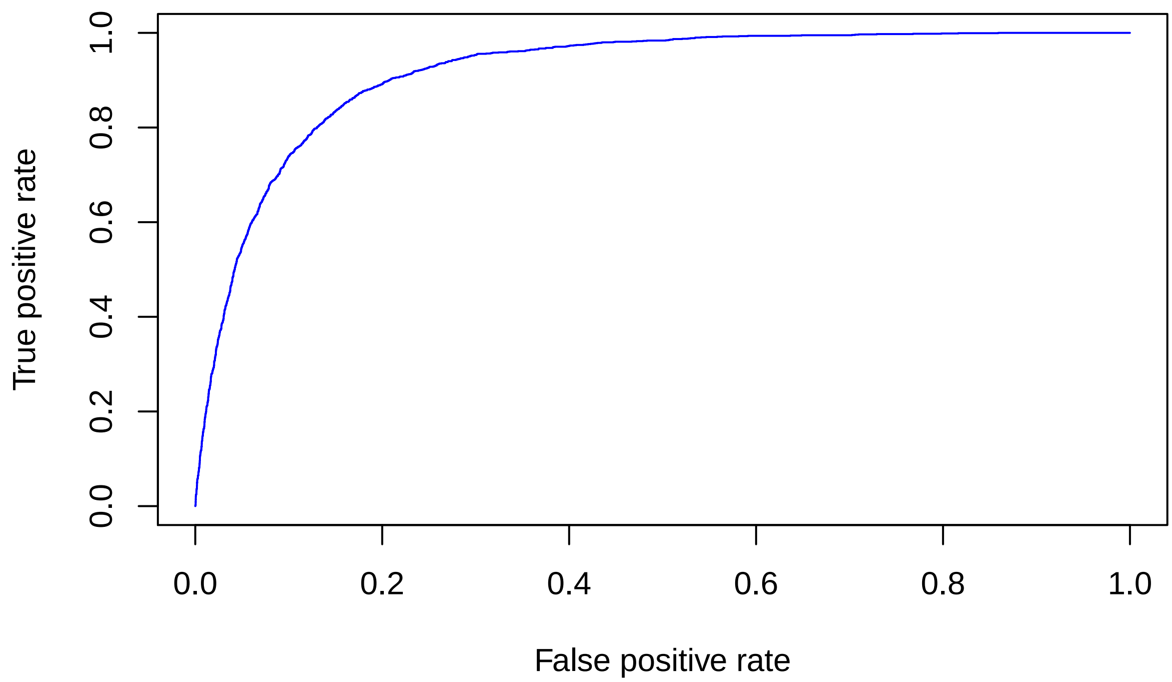
```r
p_boost<-predict(boost, newdata = test, type='raw')
caret::confusionMatrix(p_boost, test$termdeposit)
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction     0     1
##          0 11081   599
##          1   805  1006
##
##                Accuracy : 0.8959
##                  95% CI : (0.8907, 0.901)
##     No Information Rate : 0.881
##     P-Value [Acc > NIR] : 2.816e-08
##
##                   Kappa : 0.5297
##
##  Mcnemar's Test P-Value : 4.474e-08
##
##             Sensitivity : 0.9323
##             Specificity : 0.6268
##          Pos Pred Value : 0.9487
##          Neg Pred Value : 0.5555
##              Prevalence : 0.8810
##          Detection Rate : 0.8214
##    Detection Prevalence : 0.8658
##       Balanced Accuracy : 0.7795
##
##        'Positive' Class : 0
##
```

The confusion matrix results improves with the SMOTE technique . The prediction improves due to the model being trained on a balanced dataset as seen from the confusion matrix and AUC score . ROC curve

```
yActual = test$termdeposit #get the actual value for the choice variable
predTst_tree_boost = predict(boost, test, type="prob")

pred_tree_boost <- prediction(as.numeric(predTst_tree_boost[,2]), as.numeric(yActual))
perf_tree_boost_smote <- performance(pred_tree_boost,"tpr","fpr")
plot(perf_tree_boost_smote,col='blue')
```



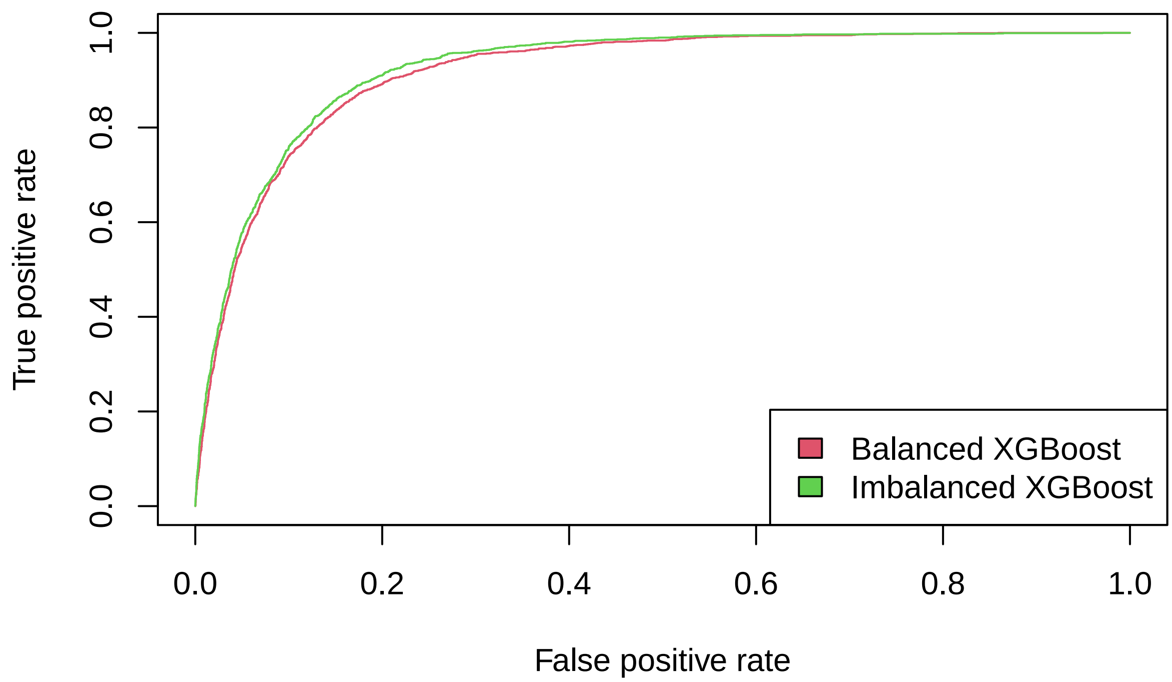
AUC score

```
perf_auc_tree_boost <- performance(pred_tree_boost,measure="auc")
print(paste("AUC= ", perf_auc_tree_boost@y.values[[1]]))
```

```
## [1] "AUC=  0.916933348639667"
```

Combining ROC curves for all the models

```
plot(perf_tree_boost_smote, col=(2))
plot(perf_tree_boost,add=T, col=(3))
legend(x='bottomright', legend=c('Balanced XGBoost','Imbalanced XGBoost'),fill=c(2,3))
```



True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
Balanced XGBoost
Imbalanced XGBoost