Bank Marketing Analytics

Season segment - Summer + Fall

Yoga Ramachandran

As there is a difference in significance across the different seasons. Therefore we wanted to study which months to focus and other significant factors in different seasons. Accordingly we grouped the customers in to two groups namely winter + spring together and summer + fall together. Here is the code for the summer and fall.We tried interaction between different explanatory variables like education-Job, Job-Balance, Contact-duration, poutcome success-pdays etc.However, we didn't include them in our final modeling as these interactions didn't show any significant results. Same is the case for the complete dataset.

```
bank_data<-read.csv('~/Documents/SantaClaraUniversity/Marketing_anlaytics/Project/bank/bank-full.csv')
# Rename y to termdeposit
bank_data <- bank_data %>% rename(termdeposit=y)
# Grouping data into seasons
Season3<-bank_data %>% filter(month=='jun'|month=='jul'
                              |month=='aug'|month=='sep' |
                                month=='oct' |month=='nov')
Season3$termdeposit<-ifelse(Season3$termdeposit=='no',0,1)
Season3$contact<-as.factor(Season3$contact)
Season3$marital<-as.factor(Season3$marital)
Season3$job<-replace(Season3$job,Season3$job=='self-employed','selfemployed' )
Season3$job<-replace(Season3$job,Season3$job=='blue-collar','bluecollar' )
Season3$job<-as.factor(Season3$job)
Season3$education<-as.factor(Season3$education)
Season3$poutcome<-as.factor(Season3$poutcome)
Season3$housing<-ifelse(Season3$housing=='no',0,1)
Season3$loan<-ifelse(Season3$loan=='no',0,1)
Season3$default<-ifelse(Season3$default=='no',0,1)
Season3$balance <- (Season3$balance - mean(Season3$balance)) / sd(Season3$balance)

#Categorising duration
durationbreaks<-c(0,5,10,15,20,100)
durationlabels<-c('0-5','5-10','10-15','15-20','20-100')
Season3$duration_bin<-cut(Season3$duration/60,breaks = durationbreaks,
                          labels = durationlabels, include.lowest = T)

#Categorising duration
agebreaks<-c(15,30,45,60,75,105)
agelabels<-c('15-30','30-45','45-60','60-75','75-105')
Season3$age_bin<-cut(Season3$age,breaks = agebreaks, labels = agelabels, include.lowest = T)
# drop original duration,age
Season3=subset(Season3, select = -c(duration,age))
Season3 <- Season3 %>% select( -termdeposit, termdeposit)

#Splitting data into test and train
```

```r
ind<-sample(2, nrow(Season3), replace=T, prob = c(0.7,0.3))
train_season<-Season3[ind==1,]
test_season<-Season3[ind==2,]

#Logistic model
logit_season = glm(termdeposit~.
            , family="binomial", data = train_season)
summary(logit_season)
```

```
##
## Call:
## glm(formula = termdeposit ~ ., family = "binomial", data = train_season)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8452  -0.3245  -0.2240  -0.1621   3.3776
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.1879552  0.2574237  -8.499  < 2e-16 ***
## jobbluecollar      -0.0420290  0.1265064  -0.332 0.739717
## jobentrepreneur    -0.0323210  0.1976551  -0.164 0.870107
## jobhousemaid       -0.5258419  0.2125216  -2.474 0.013350 *
## jobmanagement      -0.1222080  0.1241672  -0.984 0.325007
## jobretired         -0.1869698  0.1766467  -1.058 0.289855
## jobselfemployed    -0.0925481  0.1806457  -0.512 0.608428
## jobservices        -0.1165606  0.1461029  -0.798 0.424988
## jobstudent          0.4361067  0.1947426   2.239 0.025130 *
## jobtechnician      -0.1372957  0.1181289  -1.162 0.245133
## jobunemployed      -0.1377588  0.1989750  -0.692 0.488722
## jobunknown         -0.8812386  0.3990208  -2.209 0.027209 *
## maritalmarried     -0.3545619  0.0946881  -3.745 0.000181 ***
## maritalsingle      -0.0665172  0.1105316  -0.602 0.547311
## educationsecondary  0.2787798  0.1104688   2.524 0.011616 *
## educationtertiary   0.3900343  0.1273795   3.062 0.002199 **
## educationunknown    0.2975410  0.1771680   1.679 0.093068 .
## default            -0.4853343  0.2905422  -1.670 0.094832 .
## balance             0.0134188  0.0297528   0.451 0.651984
## housing            -0.3229057  0.0704137  -4.586 4.52e-06 ***
## loan               -0.4045621  0.0963262  -4.200 2.67e-05 ***
## contacttelephone   -0.0148324  0.1196973  -0.124 0.901382
## contactunknown     -2.1318912  0.1397975 -15.250  < 2e-16 ***
## day                -0.0122730  0.0041775  -2.938 0.003305 **
## monthjul           -0.2311998  0.0931766  -2.481 0.013090 *
## monthjun            1.2733033  0.1234530  10.314  < 2e-16 ***
## monthnov           -0.2640590  0.1046877  -2.522 0.011657 *
## monthoct            1.3342468  0.1356062   9.839  < 2e-16 ***
## monthsep            1.4097631  0.1471264   9.582  < 2e-16 ***
## campaign           -0.0684472  0.0145476  -4.705 2.54e-06 ***
## pdays               0.0019636  0.0004869   4.033 5.50e-05 ***
## previous            0.0232832  0.0175314   1.328 0.184149
## poutcomeother       0.1966401  0.1698551   1.158 0.246989
## poutcomesuccess     2.1670653  0.1426595  15.190  < 2e-16 ***
```

```
## poutcomeunknown    -0.2901875  0.1565126  -1.854 0.063727 .
## duration_bin5-10    1.6359033  0.0729512  22.425  < 2e-16 ***
## duration_bin10-15   3.1837545  0.0938540  33.922  < 2e-16 ***
## duration_bin15-20   3.9309400  0.1343609  29.257  < 2e-16 ***
## duration_bin20-100  4.2466607  0.1737833  24.437  < 2e-16 ***
## age_bin30-45       -0.3473009  0.0957333  -3.628 0.000286 ***
## age_bin45-60       -0.3169354  0.1123529  -2.821 0.004789 **
## age_bin60-75        0.4448058  0.2037961   2.183 0.029065 *
## age_bin75-105       0.3016790  0.3289745   0.917 0.359128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12159  on 16714  degrees of freedom
## Residual deviance:  7693  on 16672  degrees of freedom
## AIC: 7779
##
## Number of Fisher Scoring iterations: 6
```

In the logistic model for Summer and fall, We see that as the frequency of contact in the current campaign increases, the probability of people accepting term deposit decreases as most people would consider frequent calling as a nuisance. Compared to the people in the age group 15-30, people in age group 30-60 are less likely to subscribe to term deposit and people in age group 60 and above are more likely to subscribe to term deposits as there are high chances that as the age progresses beyond 60, the liability decreases.

```
# ln-likelihood
yActual = test_season$termdeposit #get the actual value for the choice variable
predTst_logit = predict(logit_season, test_season, 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] -1796.698
```

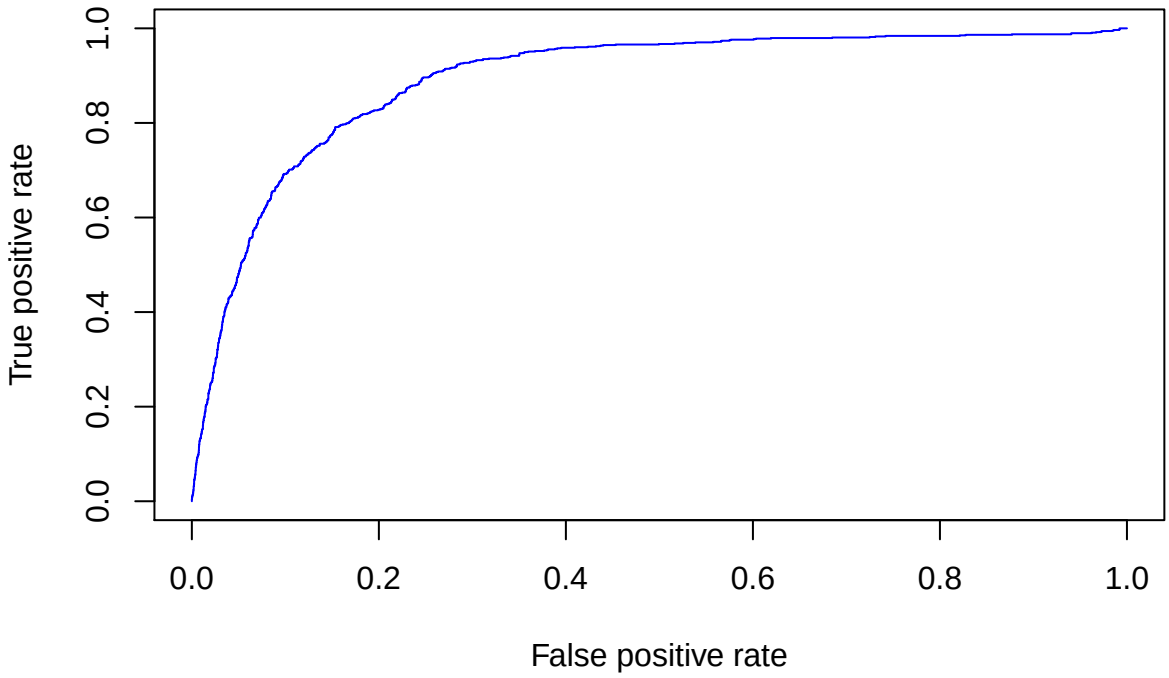
Confusion matrix

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

```
##
## predFac    0    1
##       0 6004  585
##       1  174  292
```

ROC curve

```
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_logit1 <- performance(pred_logit,measure="auc")
print(paste("AUC= ", perf_logit1@y.values[[1]]))
```

```
## [1] "AUC=  0.891953571967782"
```

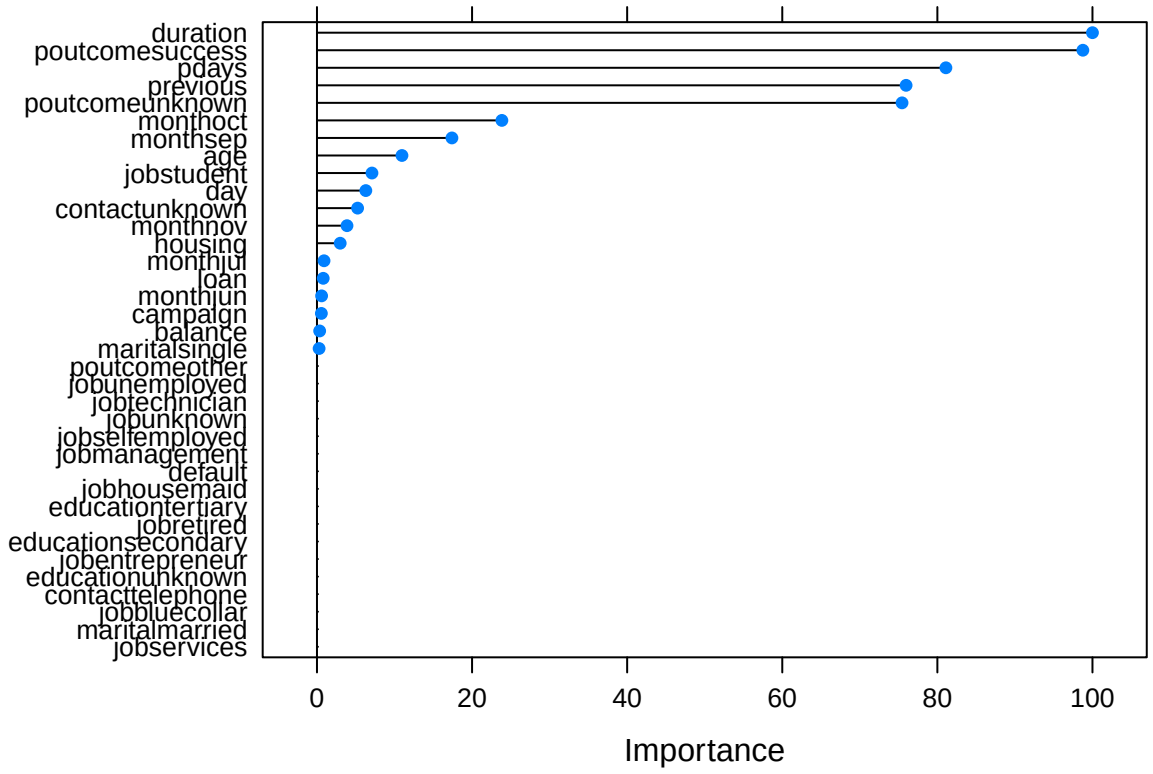
Preparing data for the other machine learning models viz:Decision tree, bagging, XGboost and random forest

```
bankfull2<-bank_data

# Grouping data into seasons
bankfull2<-bankfull2 %>% filter(month=='jun'|month=='jul'
                                |month=='aug'|month=='sep' |
                                  month=='oct' |month=='nov')
bankfull2$contact<-as.factor(bankfull2$contact)
bankfull2$marital<-as.factor(bankfull2$marital)
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)
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 <- 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,]
```

Tree with cross-validation

```
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))
```



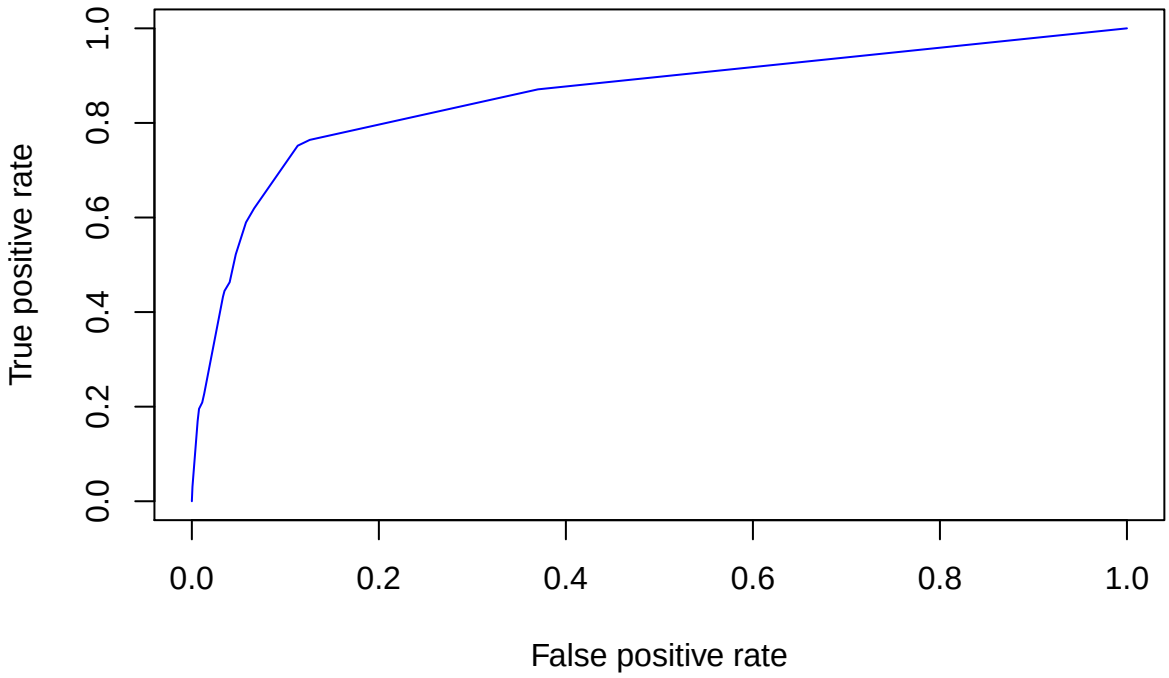
Confusion matrix

```
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 5882  437
##          1  248  377
##
##                Accuracy : 0.9014
##                  95% CI : (0.8941, 0.9083)
##     No Information Rate : 0.8828
##     P-Value [Acc > NIR] : 4.639e-07
##
##                   Kappa : 0.47
##
##  Mcnemar's Test P-Value : 6.814e-13
##
##             Sensitivity : 0.9595
##             Specificity : 0.4631
##          Pos Pred Value : 0.9308
##          Neg Pred Value : 0.6032
##              Prevalence : 0.8828
##          Detection Rate : 0.8471
##    Detection Prevalence : 0.9100
##       Balanced Accuracy : 0.7113
##
##        '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')
```



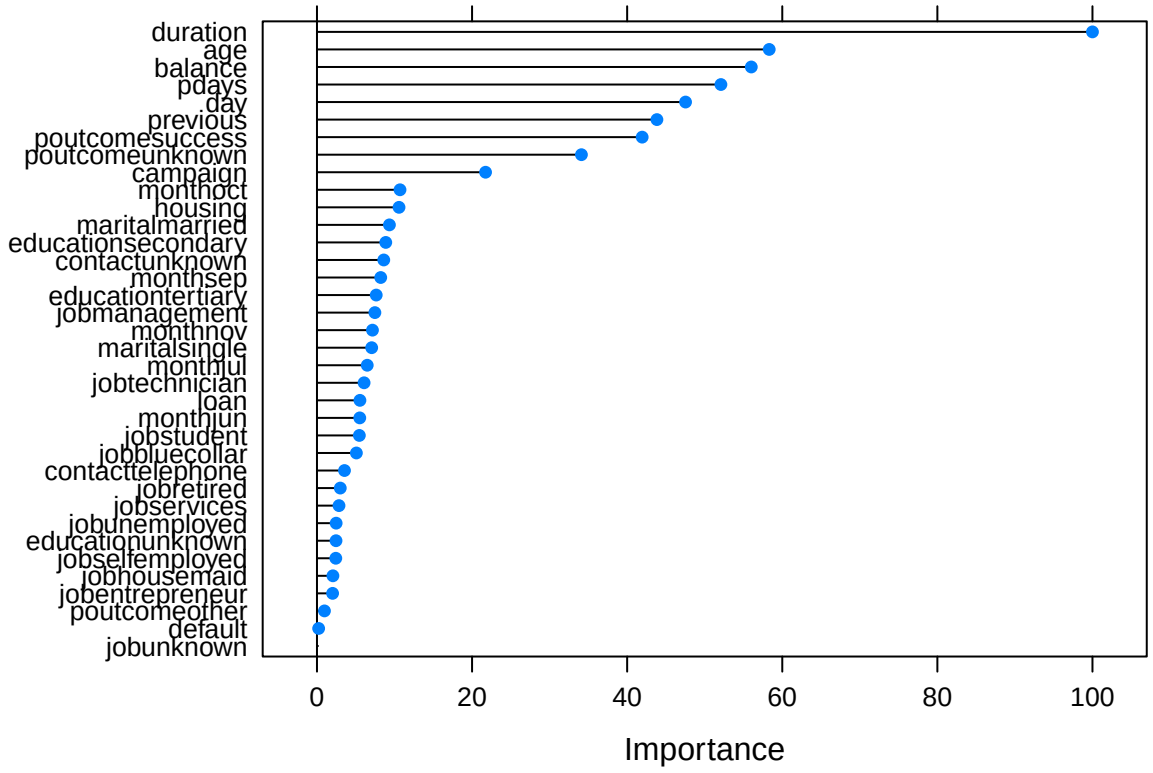
AUC score

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

```
## [1] "AUC=  0.856860868728732"
```

Bagging with decision tree

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



Confusion matrix

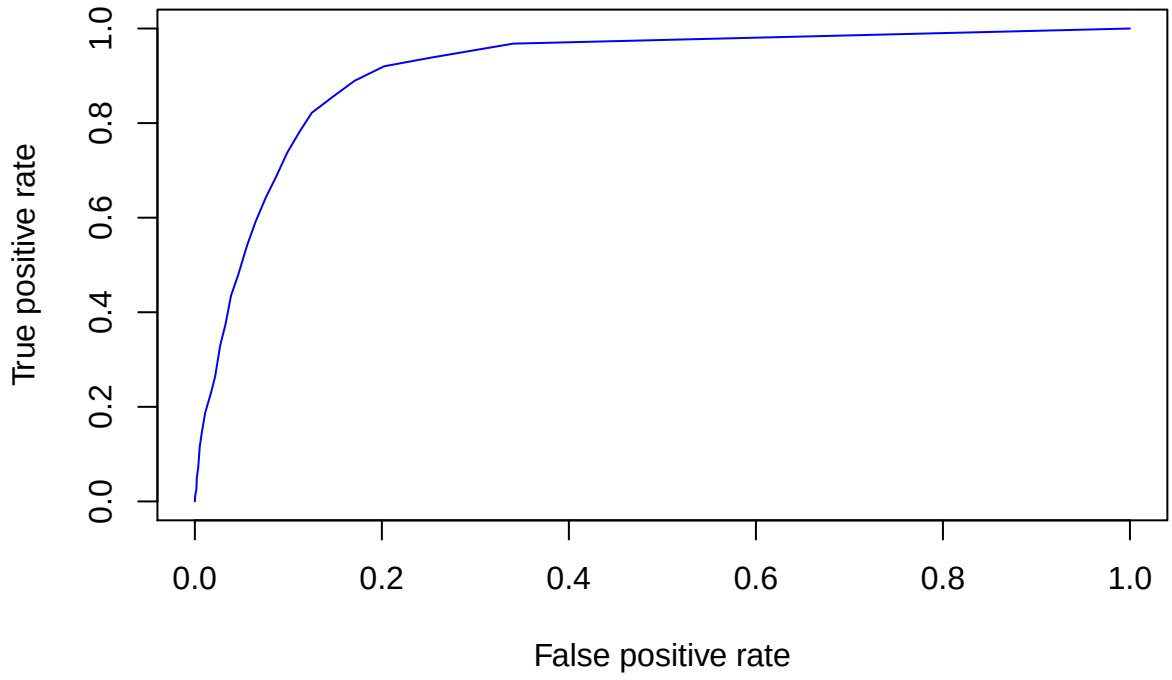
```
p_bag<-predict(bag, newdata = test_tree, type='raw')
caret::confusionMatrix(p_bag, test_tree$termdeposit)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5847  425
##          1  283  389
##
##                Accuracy : 0.898
##                  95% CI : (0.8907, 0.9051)
##     No Information Rate : 0.8828
##     P-Value [Acc > NIR] : 3.049e-05
##
##                   Kappa : 0.467
##
##  Mcnemar's Test P-Value : 1.164e-07
##
##             Sensitivity : 0.9538
##             Specificity : 0.4779
##          Pos Pred Value : 0.9322
##          Neg Pred Value : 0.5789
##              Prevalence : 0.8828
```

```
##          Detection Rate : 0.8420
##    Detection Prevalence : 0.9032
##       Balanced Accuracy : 0.7159
##
##        'Positive' Class : 0
##
```

ROC curve

```
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')
```



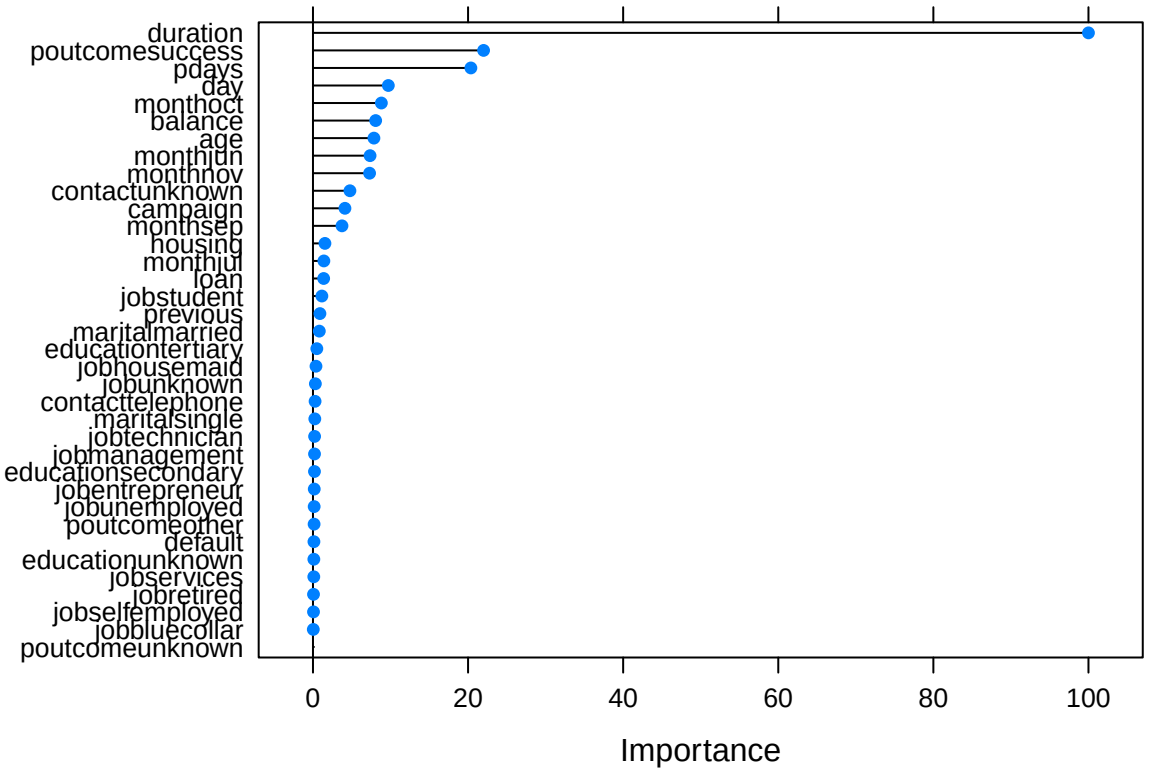
AUC score

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

```
## [1] "AUC=  0.91325488694983"
```

Xtreme Gradient boost and cross validation

```
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

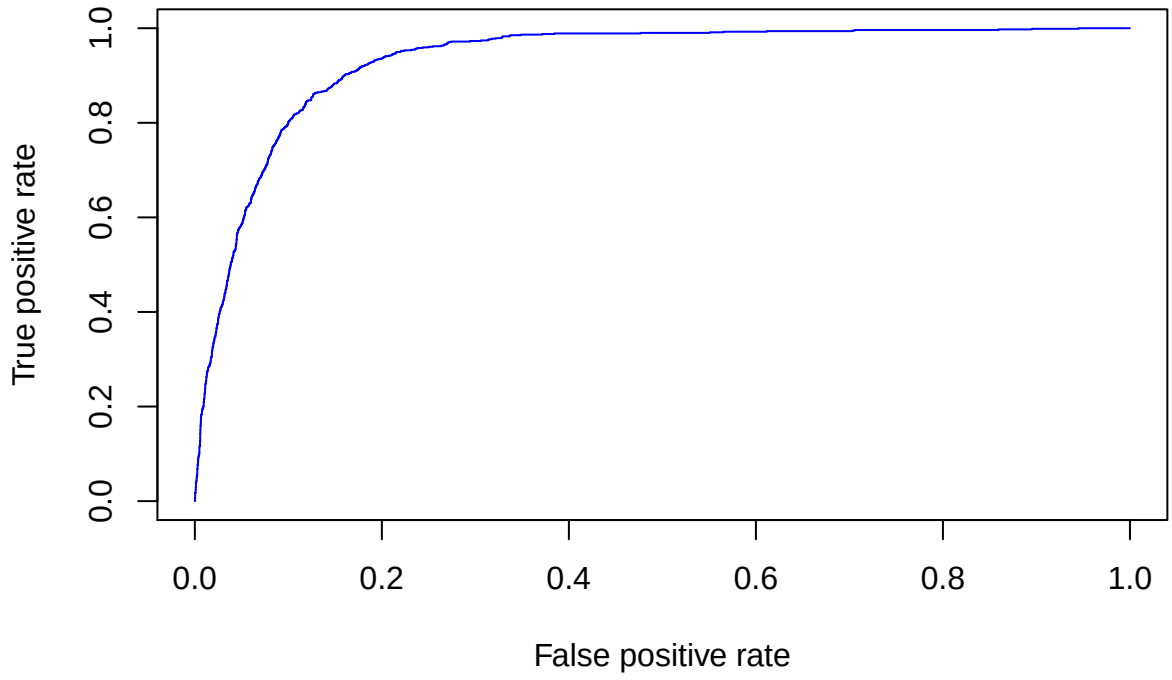
```
p_boost<-predict(boost, newdata = test_tree, type='raw')
caret::confusionMatrix(p_boost, test_tree$termdeposit)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5903  418
##          1  227  396
##
##                Accuracy : 0.9071
##                  95% CI : (0.9, 0.9138)
##     No Information Rate : 0.8828
##     P-Value [Acc > NIR] : 4.344e-11
##
##                   Kappa : 0.5004
```

```
##
##  Mcnemar's Test P-Value : 7.362e-14
##
##             Sensitivity : 0.9630
##             Specificity : 0.4865
##          Pos Pred Value : 0.9339
##          Neg Pred Value : 0.6356
##              Prevalence : 0.8828
##          Detection Rate : 0.8501
##    Detection Prevalence : 0.9103
##       Balanced Accuracy : 0.7247
##
##        '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')
```



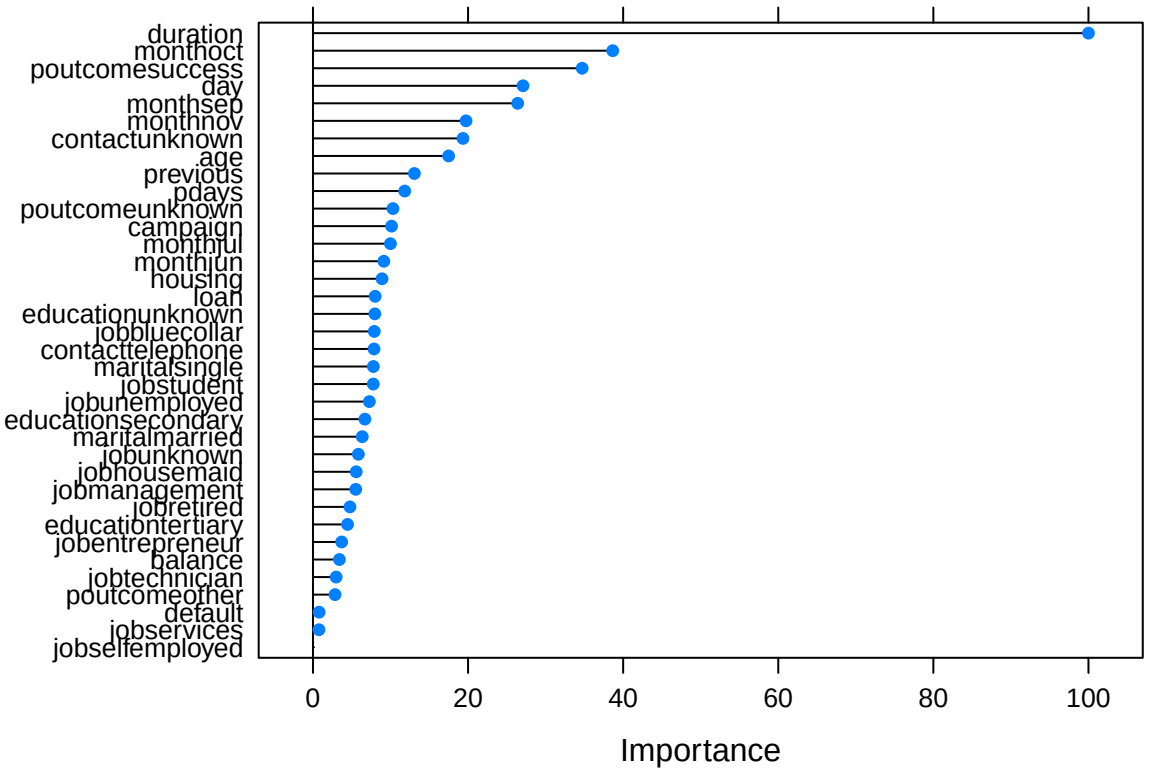
AUC score

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

```
## [1] "AUC=  0.932226112364768"
```

Random Forest with cross validation

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



Confusion matrix

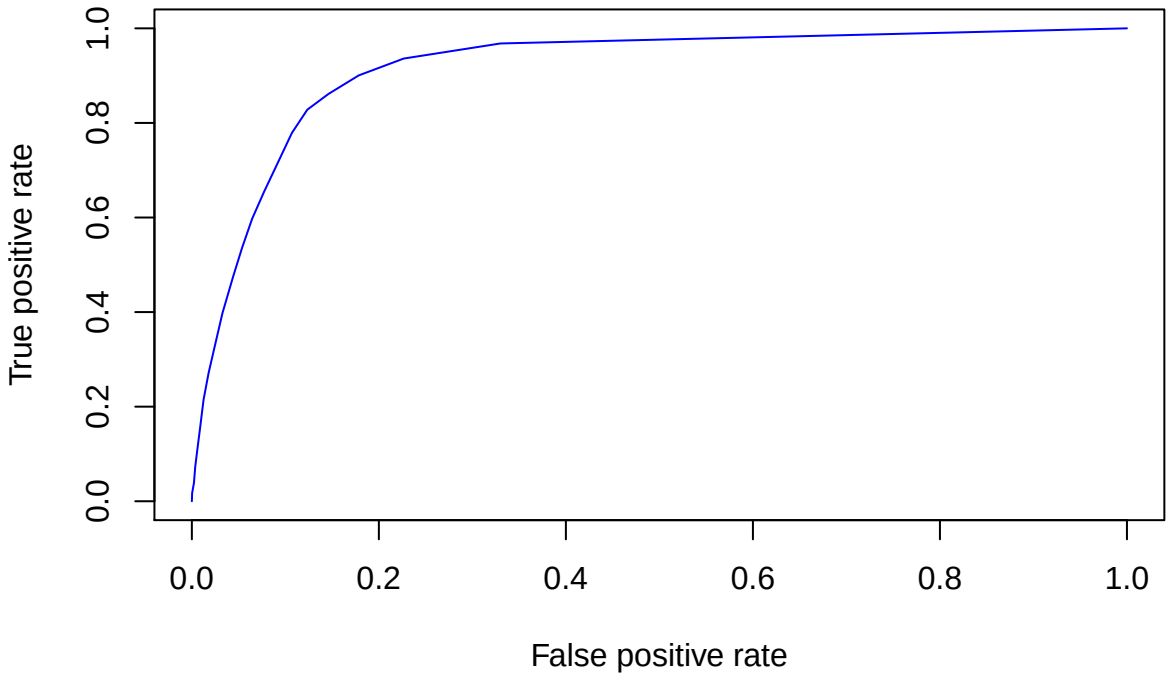
```
p_rf<-predict(rf, newdata = test_tree, type='raw')
caret::confusionMatrix(p_rf, test_tree$termdeposit)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5840  416
##          1  290  398
##
```

```
##                Accuracy : 0.8983
##                  95% CI : (0.891, 0.9053)
##     No Information Rate : 0.8828
##     P-Value [Acc > NIR] : 2.188e-05
##
##                   Kappa : 0.4734
##
##  Mcnemar's Test P-Value : 2.546e-06
##
##             Sensitivity : 0.9527
##             Specificity : 0.4889
##          Pos Pred Value : 0.9335
##          Neg Pred Value : 0.5785
##              Prevalence : 0.8828
##          Detection Rate : 0.8410
##    Detection Prevalence : 0.9009
##       Balanced Accuracy : 0.7208
##
##        'Positive' Class : 0
##
```

ROC curve

```
yActual = test_tree$termdeposit #get the actual value for the choice variable
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')
```



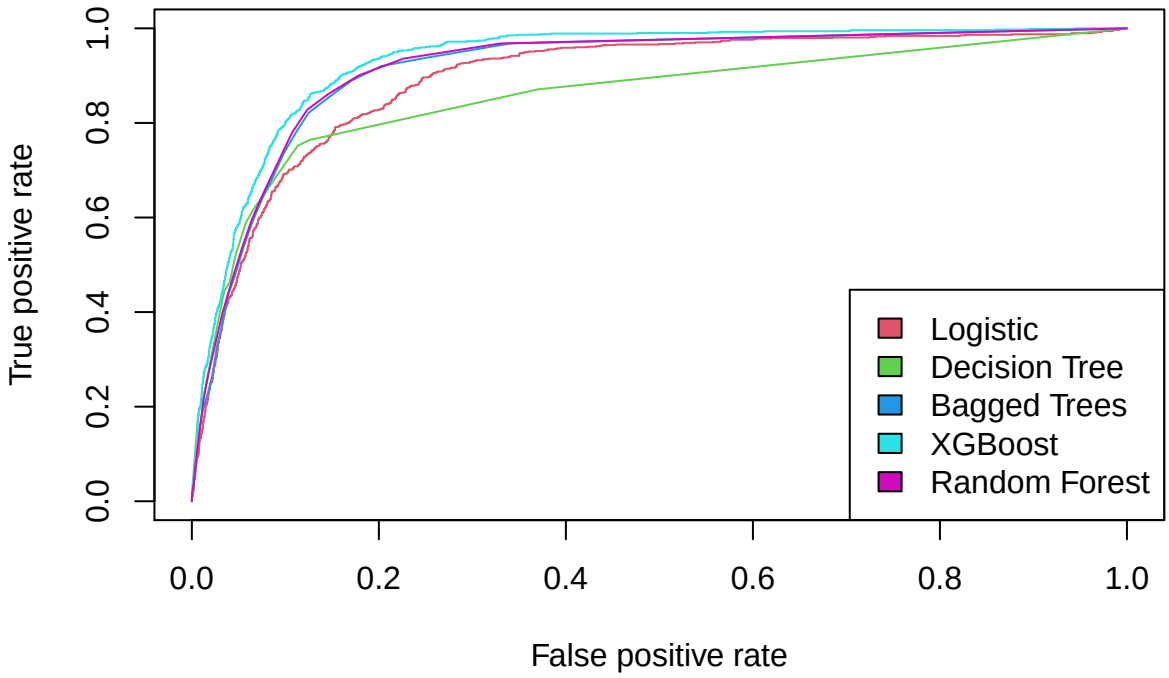
AUC score

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

```
## [1] "AUC=  0.915711688998802"
```

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 can see that the light blue line corresponding to the XGboost classifier performs the best out of all the classifiers presented here for summer and fall as it repetitively strengthens the model with weak predictions and makes it better. This is further substantiated by the high AUC score of 0.93

Target customers based on the following to increase the efficiency of the marketing campaign: We see from the previous results, customers have higher probability of subscribing to deposit in the months of September, October, December and March as around September the schools are likely to open and individuals may have increased spending in the months of July and August.Once it's September and October, again people would want to save their money in term deposit. Similarly, individuals often don't have enough money at the start of the year due to cumulative spending in the past months. With regards to December, individuals are likely to subscribe to term deposits in order to reduce the increased spending during holidays.

Similarly target the students majorly in the age group of 18-30 as most of them don't have any personal or housing loan. Also as we don't have any information about student loan, we assume they have higher chances of opting to term deposit compared to others in that age group.

Another factor to target are those individuals having secondary and tertiary education as higher the level of education, better would be their understanding of handling finances.

Also, if customers had term deposits before, they were more likely to make a deposit again. Usually, people won't try new things easily, but we may repeat an action if we have done it before. Thus, if customers already made a deposit before, they know better and are more familiar with the bank service, and this make them less likely to reject the campaigns.