- Machine Learning algorithms: Random Forest, KNN and Logistic Regression were used.
- We are concerned with predicting the NoShows=1 more accurately in this study. We want to identify people who schedule an appointment and do not show up at the hospital.

Code:

```
Train.data=read.csv("NoShowData.csv",header=TRUE)
Test.data=read.csv("NoShowHoldoutData.csv",header=TRUE)

summary(Train.data)
```

```
##    PatientId         AppointmentID     Gender                  ScheduledDa
y
##  Min.   :3.922e+04   Min.   :5030230   F:57464   2016-05-06T07:09:53Z:   2
2
##  1st Qu.:4.177e+12   1st Qu.:5640168   M:30958   2016-04-25T17:17:46Z:   1
9
##  Median :3.173e+13   Median :5680472             2016-04-25T17:18:27Z:   1
9
##  Mean   :1.472e+14   Mean   :5675129             2016-05-06T07:09:54Z:   1
9
##  3rd Qu.:9.427e+13   3rd Qu.:5725167             2016-04-25T17:17:23Z:   1
8
##  Max.   :1.000e+15   Max.   :5790481             2016-06-07T16:15:14Z:   1
3
##                                                  (Other)             :8831
2
##               AppointmentDay       Age                 Neighbourhood
##  2016-06-06T00:00:00Z: 3742   Min.   : -1.00   JARDIM CAMBURI : 6195
##  2016-05-16T00:00:00Z: 3709   1st Qu.: 18.00   MARIA ORTIZ    : 4653
##  2016-05-11T00:00:00Z: 3623   Median : 37.00   RESISTÃŠNCIA   : 3565
##  2016-05-09T00:00:00Z: 3601   Mean   : 37.04   JARDIM DA PENHA: 3041
##  2016-05-30T00:00:00Z: 3592   3rd Qu.: 55.00   ITARARÃ‰       : 2830
##  2016-06-08T00:00:00Z: 3561   Max.   :115.00   CENTRO         : 2696
##  (Other)             :66594                    (Other)        :65442
##   Scholarship       Hipertension       Diabetes         Alcoholism
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   :0.09909   Mean   :0.1966   Mean   :0.07165   Mean   :0.02988
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##
##     Handcap        SMS_received    No.show
##  Min.   :0.0000   Min.   :0.0000   No :70653
##  1st Qu.:0.0000   1st Qu.:0.0000   Yes:17769
##  Median :0.0000   Median :0.0000
##  Mean   :0.0224   Mean   :0.3207
##  3rd Qu.:0.0000   3rd Qu.:1.0000
```

```
##  Max.   :4.0000   Max.   :1.0000
##
```

```
summary(Train.data)
```

```
##    PatientId         AppointmentID     Gender                  ScheduledDa
y
##  Min.   :3.922e+04   Min.   :5030230   F:57464   2016-05-06T07:09:53Z:   2
2
##  1st Qu.:4.177e+12   1st Qu.:5640168   M:30958   2016-04-25T17:17:46Z:   1
9
##  Median :3.173e+13   Median :5680472             2016-04-25T17:18:27Z:   1
9
##  Mean   :1.472e+14   Mean   :5675129             2016-05-06T07:09:54Z:   1
9
##  3rd Qu.:9.427e+13   3rd Qu.:5725167             2016-04-25T17:17:23Z:   1
8
##  Max.   :1.000e+15   Max.   :5790481             2016-06-07T16:15:14Z:   1
3
##                                                  (Other)             :8831
2
##               AppointmentDay       Age                 Neighbourhood
##  2016-06-06T00:00:00Z: 3742   Min.   : -1.00   JARDIM CAMBURI : 6195
##  2016-05-16T00:00:00Z: 3709   1st Qu.: 18.00   MARIA ORTIZ    : 4653
##  2016-05-11T00:00:00Z: 3623   Median : 37.00   RESISTÃŠNCIA   : 3565
##  2016-05-09T00:00:00Z: 3601   Mean   : 37.04   JARDIM DA PENHA: 3041
##  2016-05-30T00:00:00Z: 3592   3rd Qu.: 55.00   ITARARÃ‰       : 2830
##  2016-06-08T00:00:00Z: 3561   Max.   :115.00   CENTRO         : 2696
##  (Other)             :66594                    (Other)        :65442
##   Scholarship       Hipertension       Diabetes         Alcoholism
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   :0.09909   Mean   :0.1966   Mean   :0.07165   Mean   :0.02988
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##
##     Handcap         SMS_received    No.show
##  Min.   :0.0000   Min.   :0.0000   No :70653
##  1st Qu.:0.0000   1st Qu.:0.0000   Yes:17769
##  Median :0.0000   Median :0.0000
##  Mean   :0.0224   Mean   :0.3207
##  3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :4.0000   Max.   :1.0000
##
```

DATA PRE-PROCESSING

From the summary, there are multiple modifications that need to be made to the data as shown below:

- Removing Patient ID and Appointment ID as they are not useful
- ScheduledDay contains date and time both, so they are separated first
- Hours are extracted from the new ScheduledTime data to categorize based on hour of the day
- Hours are categorized into morning, noon, evening and night
- Number of days are between the ScheduledDay and AppointmentDay are calculated and added to the dataframe (date_diff)
- AppointmentDay data is converted to weekdays, from Sunday to Saturday
- Age values less than 0 are converted to 0 (Age=0 may imply the appointment is for a child < 1 year old
- Some AppointmentDays are before the ScheduledDays (date_diff < 0). These are considered as error and removed from the dataframe (there are a combined 4 such entries)
- Scholarship, Handcap, Hipertension, Diabetes, Alcoholism and SMS Received are categorical, so they are converted to factors
- Handicap has entries 0,1,2,3,4. 2,3 and 4 probably indicate level of disability in a person. (4 being the highest and 0 being no disability). Entries with 2,3,4 are converted to 1 (signifying handicap=yes)

The code for the above processing is shown below:

Code:

```
#Removing Patient ID and Appointment ID as they will not be useful:
Train.data$PatientId=NULL
Train.data$AppointmentID=NULL


Test.data$PatientId=NULL
Test.data$AppointmentID=NULL

#Scheduled Time and Date need to be separated
ScheduledTime=Train.data$ScheduledDay
ScheduledTime=sub(".*T","",ScheduledTime)
ScheduledTime=sub("Z","",ScheduledTime)
Train.data$ScheduledDay=sub("T.*","",Train.data$ScheduledDay)
Train.data$AppointmentDay=sub("T.*","",Train.data$AppointmentDay)

ScheduledTimetest=Test.data$ScheduledDay
ScheduledTimetest=sub(".*T","",ScheduledTimetest)
ScheduledTimetest=sub("Z","",ScheduledTimetest)
Test.data$ScheduledDay=sub("T.*","",Test.data$ScheduledDay)
Test.data$AppointmentDay=sub("T.*","",Test.data$AppointmentDay)

#Extracting hours from scheduled time data
ScheduledTime=sub(":.*","",ScheduledTime)
ScheduledTimetest=sub(":.*","",ScheduledTimetest)

#Converting to morning, noon, evening and night
ScheduledTime=as.numeric(ScheduledTime)
```

```r
ScheduledTime=ifelse(ScheduledTime >= 5 & ScheduledTime <= 12,"Morning",
                     ifelse(ScheduledTime > 12 & ScheduledTime < 16, "Afterno
on",
                            ifelse(ScheduledTime > 16 & ScheduledTime < 19, "
Evening","Night")))

ScheduledTimetest=as.numeric(ScheduledTimetest)
ScheduledTimetest=ifelse(ScheduledTimetest >= 5 & ScheduledTimetest <= 12,"Mo
rning",
                         ifelse(ScheduledTimetest > 12 & ScheduledTimetest <
16, "Afternoon",
                                ifelse(ScheduledTimetest > 16 & ScheduledTime
test < 19, "Evening","Night")))


#Adding Scheduled Time to dataframe
Train.data=data.frame(Train.data,ScheduledTime)
Test.data=data.frame(Test.data,ScheduledTimetest)
colnames(Test.data)[13]="ScheduledTime"

#Finding number of days between appointment date and scheduled date
Train.data$AppointmentDay=as.Date(Train.data$AppointmentDay,"%Y-%m-%d")
Train.data$ScheduledDay=as.Date(Train.data$ScheduledDay,"%Y-%m-%d")
date_diff=as.numeric(Train.data$AppointmentDay-Train.data$ScheduledDay)
Train.data$ScheduledTime=as.factor(Train.data$ScheduledTime)


#Adding the date difference and days to dataframe
Train.data=data.frame(Train.data,date_diff)

Test.data$AppointmentDay=as.Date(Test.data$AppointmentDay,"%Y-%m-%d")
Test.data$ScheduledDay=as.Date(Test.data$ScheduledDay,"%Y-%m-%d")
date_diff=as.numeric(Test.data$AppointmentDay-Test.data$ScheduledDay)
Test.data$ScheduledTime=as.factor(Test.data$ScheduledTime)


#Adding the date difference and days to dataframe
Test.data=data.frame(Test.data,date_diff)

#Converting appointment day data to weekday
Train.data$AppointmentDay=as.factor(weekdays(Train.data$AppointmentDay))
Test.data$AppointmentDay=as.factor(weekdays(Test.data$AppointmentDay))

summary(Train.data)
```

```
##  Gender     ScheduledDay          AppointmentDay       Age
##  F:57464   Min.   :2015-11-10   Friday   :15264   Min.   : -1.00
##  M:30958   1st Qu.:2016-04-29   Monday   :18173   1st Qu.: 18.00
##            Median :2016-05-10   Saturday :   35   Median : 37.00
##            Mean   :2016-05-08   Thursday :13781   Mean   : 37.04
```

```
##            3rd Qu.:2016-05-20   Tuesday  :20484   3rd Qu.: 55.00
##            Max.   :2016-06-08   Wednesday:20685   Max.   :115.00
##
##          Neighbourhood    Scholarship       Hipertension       Diabetes
##  JARDIM CAMBURI : 6195   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
0
##  MARIA ORTIZ    : 4653   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
0
##  RESISTÃŠNCIA   : 3565   Median :0.00000   Median :0.0000   Median :0.0000
0
##  JARDIM DA PENHA: 3041   Mean   :0.09909   Mean   :0.1966   Mean   :0.0716
5
##  ITARARÃ‰       : 2830   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.0000
0
##  CENTRO         : 2696   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
0
##  (Other)        :65442
##    Alcoholism         Handcap         SMS_received    No.show
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   No :70653
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   Yes:17769
##  Median :0.00000   Median :0.0000   Median :0.0000
##  Mean   :0.02988   Mean   :0.0224   Mean   :0.3207
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :4.0000   Max.   :1.0000
##
##    ScheduledTime     date_diff
##  Afternoon:20928   Min.   : -1.00
##  Evening  : 3428   1st Qu.:  0.00
##  Morning  :59164   Median :  4.00
##  Night    : 4902   Mean   : 10.19
##                    3rd Qu.: 15.00
##                    Max.   :179.00
##
```

```
summary(Test.data)
```

```
##  Gender     ScheduledDay          AppointmentDay      Age
##  F:14376   Min.   :2015-12-07   Friday   :3755   Min.   :  0.00
##  M: 7729   1st Qu.:2016-04-29   Monday   :4542   1st Qu.: 18.00
##            Median :2016-05-10   Saturday :   4   Median : 37.00
##            Mean   :2016-05-09   Thursday :3466   Mean   : 37.28
##            3rd Qu.:2016-05-20   Tuesday  :5156   3rd Qu.: 56.00
##            Max.   :2016-06-08   Wednesday:5182   Max.   :115.00
##
##          Neighbourhood    Scholarship       Hipertension       Diabetes
##  JARDIM CAMBURI : 1522   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
0
##  MARIA ORTIZ    : 1152   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000
0
##  RESISTÃŠNCIA   :  866   Median :0.00000   Median :0.0000   Median :0.0000
```

```
0
##  JARDIM DA PENHA:  836   Mean   :0.09496   Mean   :0.1999   Mean   :0.0727
4
##  ITARARÃ‰       :  684   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.0000
0
##  SANTA MARTHA   :  652   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
0
##  (Other)        :16393
##    Alcoholism         Handcap         SMS_received    No.show
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   No :17555
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000   Yes: 4550
##  Median :0.00000   Median :0.00000   Median :0.0000
##  Mean   :0.03248   Mean   :0.02162   Mean   :0.3221
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :3.00000   Max.   :1.0000
##
##    ScheduledTime     date_diff
##  Afternoon: 5314   Min.   : -6.00
##  Evening  :  821   1st Qu.:  0.00
##  Morning  :14739   Median :  4.00
##  Night    : 1231   Mean   : 10.17
##                    3rd Qu.: 14.00
##                    Max.   :179.00
##

#There is one age value=-1, so that is replaced with 0
#Maximum age is 115, which is realistic
Train.data$Age[Train.data$Age<0]="0"
Test.data$Age[Test.data$Age<0]="0"

#It is seen that there are some appointment dates that were scheduled before
the scheduled day
#These need to be removed
a=which(Train.data$date_diff<0)
Train.data=Train.data[-c(a),]

a=which(Test.data$date_diff<0)
Test.data=Test.data[-c(a),]

#Converting categorical predictors to factors
Train.data$Scholarship=as.factor(Train.data$Scholarship)
Train.data$Hipertension=as.factor(Train.data$Hipertension)
Train.data$Diabetes=as.factor(Train.data$Diabetes)
Train.data$Alcoholism=as.factor(Train.data$Alcoholism)
Train.data$SMS_received=as.factor(Train.data$SMS_received)
Train.data$Handcap=as.factor(Train.data$Handcap)
Train.data$Age=as.numeric(Train.data$Age)

Test.data$Scholarship=as.factor(Test.data$Scholarship)
Test.data$Hipertension=as.factor(Test.data$Hipertension)
```

```
Test.data$Diabetes=as.factor(Test.data$Diabetes)
Test.data$Alcoholism=as.factor(Test.data$Alcoholism)
Test.data$SMS_received=as.factor(Test.data$SMS_received)
Test.data$Handcap=as.factor(Test.data$Handcap)
Test.data$Age=as.numeric(Test.data$Age)

summary(Train.data)
```

```
##  Gender     ScheduledDay          AppointmentDay       Age
##  F:57463   Min.   :2015-11-10   Friday   :15264   Min.   :  0.00
##  M:30956   1st Qu.:2016-04-29   Monday   :18172   1st Qu.: 18.00
##            Median :2016-05-10   Saturday :   35   Median : 37.00
##            Mean   :2016-05-08   Thursday :13781   Mean   : 37.04
##            3rd Qu.:2016-05-20   Tuesday  :20482   3rd Qu.: 55.00
##            Max.   :2016-06-08   Wednesday:20685   Max.   :115.00
##
##          Neighbourhood   Scholarship Hipertension Diabetes  Alcoholism
##  JARDIM CAMBURI : 6195   0:79657     0:71036      0:82084   0:85777
##  MARIA ORTIZ    : 4653   1: 8762     1:17383      1: 6335   1: 2642
##  RESISTÃŠNCIA   : 3564
##  JARDIM DA PENHA: 3041
##  ITARARÃ‰       : 2830
##  CENTRO         : 2696
##  (Other)        :65440
##  Handcap   SMS_received No.show       ScheduledTime     date_diff
##  0:86611   0:60058      No :70653   Afternoon:20927   Min.   :  0.00
##  1: 1651   1:28361      Yes:17766   Evening  : 3428   1st Qu.:  0.00
##  2:  146                            Morning  :59162   Median :  4.00
##  3:    8                            Night    : 4902   Mean   : 10.19
##  4:    3                                              3rd Qu.: 15.00
##                                                       Max.   :179.00
##
```

```
summary(Test.data)
```

```
##  Gender     ScheduledDay          AppointmentDay      Age
##  F:14374   Min.   :2015-12-07   Friday   :3755   Min.   :  0.00
##  M: 7729   1st Qu.:2016-04-29   Monday   :4542   1st Qu.: 18.00
##            Median :2016-05-10   Saturday :   4   Median : 37.00
##            Mean   :2016-05-09   Thursday :3465   Mean   : 37.28
##            3rd Qu.:2016-05-20   Tuesday  :5156   3rd Qu.: 56.00
##            Max.   :2016-06-08   Wednesday:5181   Max.   :115.00
##
##          Neighbourhood   Scholarship Hipertension Diabetes  Alcoholism
##  JARDIM CAMBURI : 1522   0:20004     0:17685      0:20495   0:21385
##  MARIA ORTIZ    : 1152   1: 2099     1: 4418      1: 1608   1:  718
##  RESISTÃŠNCIA   :  866
##  JARDIM DA PENHA:  836
##  ITARARÃ‰       :  684
##  SANTA MARTHA   :  652
```

```
##  (Other)        :16391
##  Handcap   SMS_received No.show       ScheduledTime     date_diff
##  0:21672   0:14982      No :17555   Afternoon: 5312   Min.   :  0.00
##  1:  389   1: 7121      Yes: 4548   Evening  :  821   1st Qu.:  0.00
##  2:   37                            Morning  :14739   Median :  4.00
##  3:    5                            Night    : 1231   Mean   : 10.17
##                                                       3rd Qu.: 14.00
##                                                       Max.   :179.00
##
```

```
#Handicap data has 2,3 and 4 levels which probably belong to 1
Train.data$Handcap=as.character(Train.data$Handcap)
Train.data$Handcap=sub("2","1",Train.data$Handcap)
Train.data$Handcap=sub("3","1",Train.data$Handcap)
Train.data$Handcap=sub("4","1",Train.data$Handcap)
Train.data$Handcap=as.factor(Train.data$Handcap)

Test.data$Handcap=sub("2","1",Test.data$Handcap)
Test.data$Handcap=sub("3","1",Test.data$Handcap)
Test.data$Handcap=as.factor(Test.data$Handcap)
```

Explanation:

In the training data set, we see that there is a large imbalance between the NoShow labels. There are over 70,000 Shows as compared to 17,000 NoShows. Similar trend is observed in test dataset (17,000 Shows compared to 4,500 NoShows). Therefore, the criteria for selecting best model cannot be the Misclassification Error. If Accuracy is taken as the basis for selecting best model, then the model which classifies NoShows=0 very well but NoShows=1 poorly will have better accuracy, due to the class imbalance. Accuracy can be altered by changing the probability threshold.

Therefore, the criteria for evaluating different algorithms will be the AUC parameter for that model. AUC results will determine if the model is able to distinguish between class 0 and 1. AUC =0.5 will mean that model has no class separation capacity. Therefore, higher the AUC value, better the model.

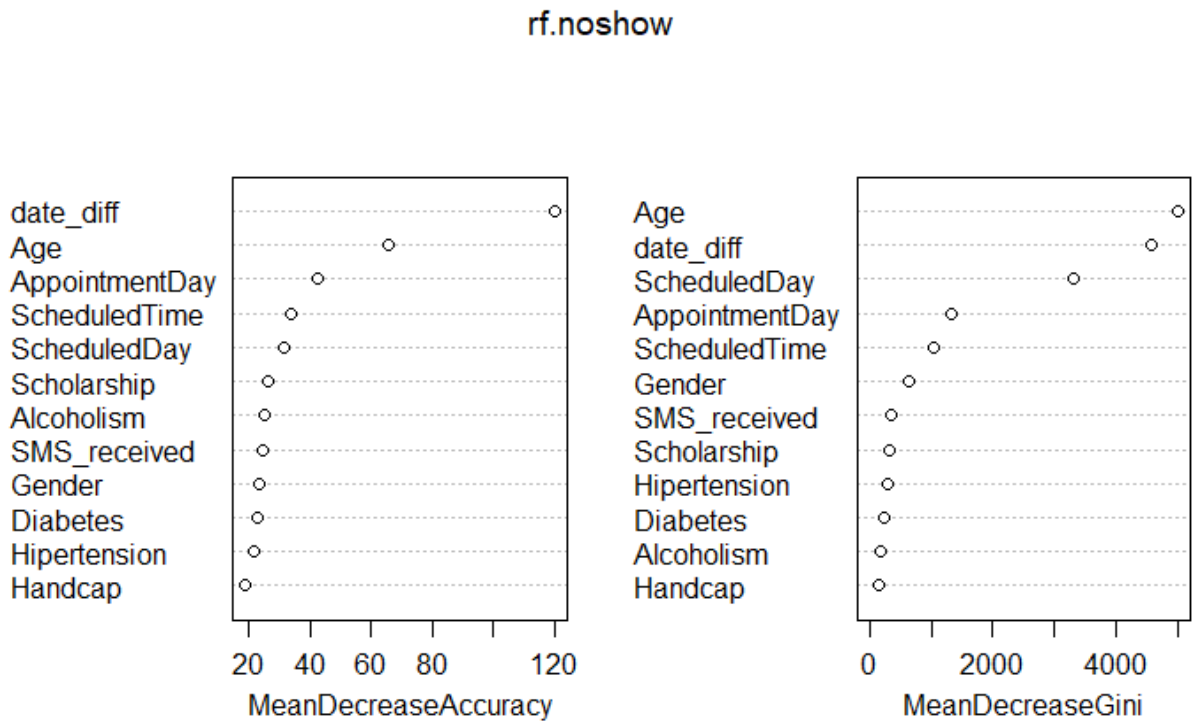
RANDOM FOREST

Code:

```
#Random Forest-----------------------------------------------
library(randomForest)

set.seed(1)
rf.noshow=randomForest(No.show~.-Neighbourhood,data=Train.data,mtry=12,ntree=
300)

rf.noshow

##
## Call:
```

```
##  randomForest(formula = No.show ~ . - Neighbourhood, data = Train.data,
mtry = 13, ntree = 300)
##                Type of random forest: classification
##                      Number of trees: 300
## No. of variables tried at each split: 12
##
##         OOB estimate of  error rate: 22.85%
## Confusion matrix:
##        No  Yes class.error
## No  64325 6328  0.08956449
## Yes 13879 3887  0.78121130

#There is an OOB error of 22.8%
#Only 38% of NoShows=1 are correctly classified by the model

#Using caret library to vary the hyperparameter mtry:

caretGrid=expand.grid(mtry=c(4,8))
trainControl=trainControl(method="cv",number=10,classProbs=TRUE,
                          summaryFunction=twoClassSummary)


set.seed(1)
rf.caret=train(No.show~.-Neighbourhood,data=Train.data, method="rf",
               trControl=trainControl, verbose=FALSE,
               tuneGrid=caretGrid,ntree=300,metric="ROC")

rf.caret

## Random Forest
##
## 88419 samples
##    13 predictor
##     2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 79578, 79578, 79578, 79576, 79577, 79577, ...
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens       Spec
##   4     0.7161812  0.9963201  0.02403522
##   8     0.7153304  0.9404555  0.17083324
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.

rf.caret$results
```

```
##   mtry       ROC      Sens       Spec       ROCSD      SensSD      SpecSD
## 1    4 0.7161812 0.9963201 0.02403522 0.003973311 0.001141984 0.003785104
## 2    8 0.7153304 0.9404555 0.17083324 0.005467282 0.004523992 0.012725178

#We choose the mtry value with higher ROC, as it means that the AUC is more

#Plotting important predictors
set.seed(1)
rf.noshow=randomForest(No.show~.-Neighbourhood,data=Train.data,mtry=4,ntree=3
00,importance=TRUE)
importance(rf.noshow)

##                        No        Yes MeanDecreaseAccuracy MeanDecreaseGini
## Gender          21.036338   9.425231             23.33438         649.2588
## ScheduledDay    30.180121  -1.942249             31.27147        3307.2008
## AppointmentDay  34.015320  19.787759             42.29530        1326.7488
## Age             44.024812  30.401315             65.61886        5007.5717
## Scholarship     14.961754  23.574131             26.11886         315.4318
## Hipertension    19.225156  -5.319980             21.42584         304.6515
## Diabetes        23.399478  -9.820677             22.47497         243.8953
## Alcoholism      22.291716   7.247403             24.75550         173.6159
## Handcap          7.346445  24.416061             18.77658         162.3116
## SMS_received    29.716668 -45.161001             24.41612         360.7752
## ScheduledTime   18.735318  31.313969             33.81032        1053.1382
## date_diff       71.205391  34.945200            119.95464        4572.8201

varImpPlot(rf.noshow)
```



Explanation:

In Random Forest, all the predictors were used for training the model except Neighbourhood. This is because the Random Forest cannot accept predictors with more than 30 factor levels.

Initially, randomForest( ) function of randomForest library was used to build the model. Caret package was used to tune the mtry value. For classification problems the optimal mtry value is generally mtry = $\sqrt{p}$. Here p=12, so mtry =4 was used. Another random value of mtry=8 was added for comparison. From the tuning results, we get mtry = 4 as the optimal value.

Hyperparameter value	**ROC performance**
Mtry=4	**0.71618**
Mtry=8	0.71533

The variable importance plot was also plotted for the predictors. It was found that date_diff (difference between scheduled and appointment day) is the most influential predictor. This is followed by Age, Appointment weekday, Schedule Time and Schedule Day. Factors like Handicap, Hypertension, Diabetes and Gender are not very influential.

K-NEAREST NEIGHBORS

Code:

```
#KNN-----------------------------------------------------------------

trControl=trainControl(method="cv",number=10,classProbs=TRUE,
                       summaryFunction=twoClassSummary)
set.seed(1)
knn.fit=train(No.show~.-Neighbourhood,
              data=Train.data,
              method="knn",
              tuneGrid=expand.grid(k=c(10,50,100)),
              preProcess="scale",
              metric="ROC",
              trControl=trControl)
knn.fit

## k-Nearest Neighbors
##
## 88419 samples
##    13 predictor
##     2 classes: 'No', 'Yes'
##
## Pre-processing: scaled (18)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 79578, 79578, 79578, 79576, 79577, 79577, ...
## Resampling results across tuning parameters:
##
##   k    ROC        Sens       Spec
```

```
##    10  0.6702338  0.9512405  0.119498427
##    50  0.6855965  0.9938431  0.020601803
##   100  0.6842933  0.9983723  0.005459874
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 50.
```

Explanation:

Since we want to determine the ROC performance of all algorithms, caret package was used for KNN as well. The values of K to be tuned were selected as 10, 50 and 100. From the results, K=50 was found to have the best AUC performance of 0.685.

Algorithm	Parameter	ROC Performance
Random Forest	**Mtry=4**	**0.71618**
	Mtry=8	0.71533
KNN	K=10	0.67023
	K=50	0.68555
	K=100	0.68429

The results of KNN and Random Forest have been compared in the table above. It is seen that Random Forest has a better performance on the training data, based on ROC metric.

LOGISTIC REGRESSION

Code:

```
#Logistic Regression---------------------------------
trControl2=trainControl(method="cv",number=10,classProbs=TRUE,
                        summaryFunction=twoClassSummary)
set.seed(1)
cv.logit=train(No.show~.-Neighbourhood,
               data=Train.data,
               method="glm",
               family="binomial",
               metric="ROC",
               trControl=trControl2)
cv.logit
```

```
## Generalized Linear Model
##
## 88419 samples
##    13 predictor
##     2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 79578, 79578, 79578, 79576, 79577, 79577, ...
## Resampling results:
```

```
##
##   ROC        Sens       Spec
##   0.6645297  0.9911398  0.01666112
```

```
summary(cv.logit)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2869  -0.6736  -0.5753  -0.4834   2.2690
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              1.180e+02  1.213e+01   9.724  < 2e-16 ***
## GenderM                 -1.284e-02  1.848e-02  -0.695 0.487270
## ScheduledDay            -7.046e-03  7.164e-04  -9.835  < 2e-16 ***
## AppointmentDayMonday    -4.961e-02  2.790e-02  -1.778 0.075426 .
## AppointmentDaySaturday   3.975e-01  4.072e-01   0.976 0.328921
## AppointmentDayThursday  -1.162e-01  3.019e-02  -3.848 0.000119 ***
## AppointmentDayTuesday   -9.055e-02  2.762e-02  -3.279 0.001042 **
## AppointmentDayWednesday -9.314e-02  2.770e-02  -3.362 0.000773 ***
## Age                     -7.676e-03  4.471e-04 -17.168  < 2e-16 ***
## Scholarship1             2.196e-01  2.774e-02   7.919 2.40e-15 ***
## Hipertension1           -3.061e-02  2.799e-02  -1.094 0.274021
## Diabetes1                1.178e-01  3.886e-02   3.032 0.002428 **
## Alcoholism1              2.566e-01  5.045e-02   5.086 3.66e-07 ***
## Handcap1                 4.949e-02  6.363e-02   0.778 0.436712
## SMS_received1            3.786e-01  1.915e-02  19.773  < 2e-16 ***
## ScheduledTimeEvening     7.042e-02  4.504e-02   1.563 0.117939
## ScheduledTimeMorning    -1.438e-01  2.019e-02  -7.122 1.06e-12 ***
## ScheduledTimeNight       8.879e-02  3.859e-02   2.301 0.021400 *
## date_diff                1.573e-02  8.873e-04  17.733  < 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: 88718  on 88418  degrees of freedom
## Residual deviance: 84943  on 88400  degrees of freedom
## AIC: 84981
##
## Number of Fisher Scoring iterations: 4
```

```
#Removing insignificant factors:
set.seed(1)
cv.logit2=train(No.show~.-Neighbourhood-Gender-Hipertension-Handcap,
                data=Train.data,
```

```
                method="glm",
                family="binomial",
                metric="ROC",
                trControl=trControl2)
cv.logit2
```

```
## Generalized Linear Model
##
## 88419 samples
##    13 predictor
##     2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 79578, 79578, 79578, 79576, 79577, 79577, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.6647101  0.9911681  0.01677367
```

```
summary(cv.logit2)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2907  -0.6736  -0.5752  -0.4839   2.2635
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              1.179e+02  1.213e+01   9.717  < 2e-16 ***
## ScheduledDay            -7.041e-03  7.164e-04  -9.829  < 2e-16 ***
## AppointmentDayMonday    -4.961e-02  2.790e-02  -1.778 0.075404 .
## AppointmentDaySaturday   3.960e-01  4.069e-01   0.973 0.330509
## AppointmentDayThursday  -1.163e-01  3.019e-02  -3.853 0.000117 ***
## AppointmentDayTuesday   -9.050e-02  2.762e-02  -3.277 0.001049 **
## AppointmentDayWednesday -9.312e-02  2.770e-02  -3.362 0.000775 ***
## Age                     -7.824e-03  4.013e-04 -19.497  < 2e-16 ***
## Scholarship1             2.214e-01  2.747e-02   8.062 7.51e-16 ***
## Diabetes1                1.039e-01  3.639e-02   2.855 0.004302 **
## Alcoholism1              2.492e-01  5.002e-02   4.983 6.26e-07 ***
## SMS_received1            3.788e-01  1.913e-02  19.802  < 2e-16 ***
## ScheduledTimeEvening     7.049e-02  4.503e-02   1.566 0.117444
## ScheduledTimeMorning    -1.444e-01  2.018e-02  -7.152 8.58e-13 ***
## ScheduledTimeNight       8.897e-02  3.859e-02   2.306 0.021130 *
## date_diff                1.575e-02  8.870e-04  17.762  < 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: 88718  on 88418  degrees of freedom
## Residual deviance: 84945  on 88403  degrees of freedom
## AIC: 84977
##
## Number of Fisher Scoring iterations: 4
```

```
#Now, examining the effect of under and oversampling data to decrease class i
mbalance
library(ROSE)
```

```
both=ovun.sample(No.show ~ . -Gender-Neighbourhood-Handcap-Hipertension,data=
Train.data, method="both",p=0.5,seed=1,N=88419)$data
```

```
table(both$No.show)
```

```
##
##    No   Yes
## 44167 44252
```

```
set.seed(1)
cv.logit3=train(No.show~.-Neighbourhood-Gender-Hipertension-Handcap,
                data=both,
                method="glm",
                family="binomial",
                metric="ROC",
                trControl=trControl2)
cv.logit3
```

```
## Generalized Linear Model
##
## 88419 samples
##    13 predictor
##     2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 79578, 79576, 79577, 79577, 79576, 79578, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.6688331  0.6849683  0.5719969
```

```
summary(cv.logit3)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -3.1045  -1.0681   0.1979   1.1342   1.6910
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               1.368e+02  1.001e+01  13.662  < 2e-16 ***
## ScheduledDay             -8.080e-03  5.913e-04 -13.665  < 2e-16 ***
## AppointmentDayMonday     -7.503e-02  2.279e-02  -3.293 0.000992 ***
## AppointmentDaySaturday    8.895e-01  3.547e-01   2.507 0.012163 *
## AppointmentDayThursday   -1.303e-01  2.442e-02  -5.337 9.46e-08 ***
## AppointmentDayTuesday    -9.117e-02  2.244e-02  -4.063 4.84e-05 ***
## AppointmentDayWednesday  -1.179e-01  2.242e-02  -5.260 1.44e-07 ***
## Age                      -7.293e-03  3.273e-04 -22.284  < 2e-16 ***
## Scholarship1              2.362e-01  2.270e-02  10.406  < 2e-16 ***
## Diabetes1                 1.980e-01  2.867e-02   6.907 4.93e-12 ***
## Alcoholism1               2.195e-01  4.128e-02   5.318 1.05e-07 ***
## SMS_received1             3.862e-01  1.572e-02  24.567  < 2e-16 ***
## ScheduledTimeEvening      6.685e-02  3.729e-02   1.793 0.073039 .
## ScheduledTimeMorning     -1.588e-01  1.651e-02  -9.618  < 2e-16 ***
## ScheduledTimeNight        1.015e-01  3.190e-02   3.181 0.001469 **
## date_diff                 1.944e-02  7.889e-04  24.642  < 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: 122575  on 88418  degrees of freedom
## Residual deviance: 116395  on 88403  degrees of freedom
## AIC: 116427
##
## Number of Fisher Scoring iterations: 4
```

Explanation:

Logistic regression was first completed with all the predictors except Neighbourhood. In that case, ROC performance of 0.6645297 was obtained. Next, the insignificant predictors (Gender, Hipertension and Handcap) were removed and the model was retrained. The ROC increased slightly to 0.6647101.

Next, to counter the class imbalance, combination of under and oversampling was used using the ROSE library and ovun( ) command. Through this, both the classes of NoShow have nearly equal number of observations. It can be seen that when model is trained with this data, the performance improves slightly to ROC=0.6688331. The performance summary of all the models is represented in the table below:

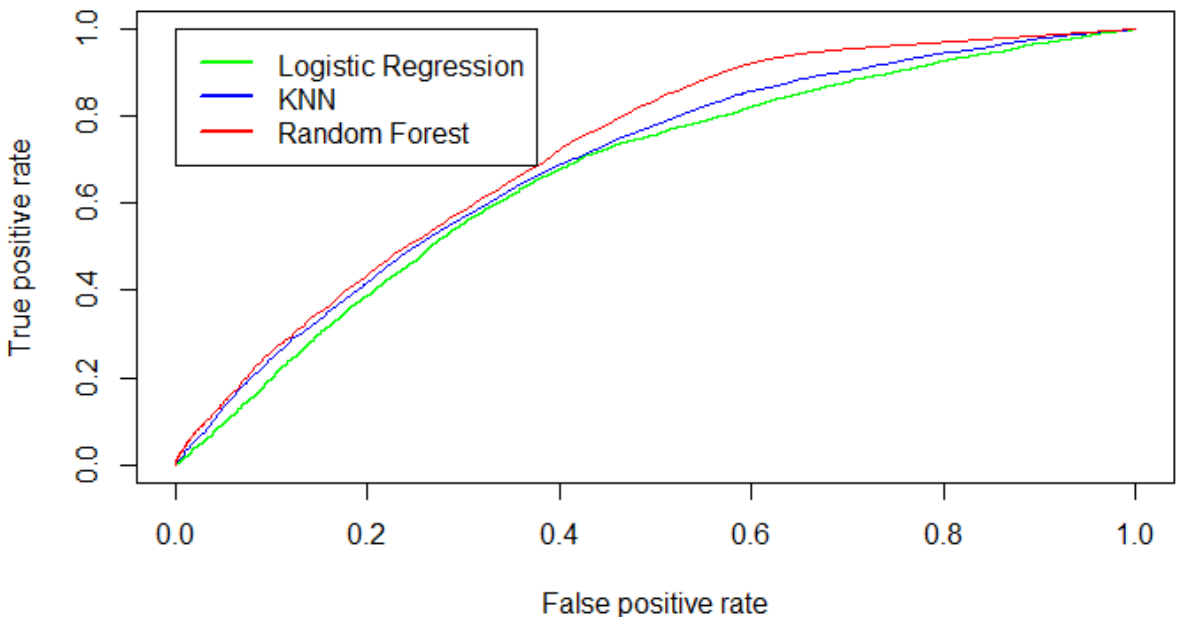
Algorithm	**Parameter**	**ROC Performance**
Random Forest	**Mtry=4**	**0.71618**

	Mtry=8	0.71533
KNN	K=10	0.67023
	K=50	0.68555
	K=100	0.68429
Logistic Regression	All predictors	0.66452
	Significant predictors only	0.66471
	Significant predictors + combination of under and oversampling	0.66883

From the above table, it can be seen that best performance is obtained for Random Forest, followed by KNN and then Logistic Regression.

Code for ROC curves for all algorithms

```
library(pROC)
library(ROCR)

#ROC for Random forest

rf.prob=predict(rf.caret,Test.data,type="prob")
pred.roc.rf=prediction(rf.prob[,2],Test.data$No.show)
roc.rf=performance(pred.roc.rf,"tpr","fpr")


#ROC for KNN

knn.prob=predict(knn.fit,Test.data,type="prob")
pred.roc.knn=prediction(knn.prob[,2],Test.data$No.show)
roc.knn=performance(pred.roc.knn,"tpr","fpr")


#ROC for logistic regression

logit.probability=predict(cv.logit3,Test.data,type="prob")
pred.roc.logit=prediction(logit.probability,Test.data$No.show)
roc.logit=performance(pred.roc.logit,"tpr","fpr")


#Plotting all ROC curves
plot(roc.logit,col="green")
plot(roc.knn,add=TRUE,col="blue")
plot(roc.rf,add=TRUE,col="red")
legend(0,1,legend=c("Logistic Regression","KNN","Random Forest"),lwd=c(2,2,2)
,col=c("green","blue","red"))
```



From the ROC curves also, it can be verified that Random Forest will have the better performance for the given dataset. This is because it has the highest AUC and also, its increase in TPR (true positive rate) will be higher as compared to FPR (false positive rate) when compared with other models.

PREDICTION

We are focusing on predicting NoShows=1 more accurately since we want to determine factors that cause people to not show up. Hence, we want to increase the sensitivity of the model, so that we can predict people who will not show up (NoShow=1) to be predicted by our model as well. We are willing to accept an increase in false positive rate (our model predicts someone will be a NoShow but they actually show up) as long as it means we are able to predict accurately a preson who is going to be a NoShow is actually a NoShow. For this the threshold to classify a person as NoShow will be decreased from the default value of 0.5. This change will decrease the accuracy of the model and also the specificity, but we are not concerned about that, since our focus is on sensitivity.

For evaluating the performance of model on testing data, the threshold to classify a person as a NoShow was reduced until a Sensitivity of 90% was reached.

Prediction using Random Forest

Code:

```
#Default threshold of 0.5
rf.prob=predict(rf.caret,Test.data,type="prob")
rf.predict=rep("No",22103)
rf.predict[rf.prob[,2]>0.5]="Yes"
confusionMatrix(as.factor(rf.predict),Test.data$No.show,positive="Yes")
```

Result:

```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  17504  4463
       Yes    51    85

               Accuracy : 0.7958
                 95% CI : (0.7904, 0.8011)
    No Information Rate : 0.7942
    P-Value [Acc > NIR] : 0.289

                  Kappa : 0.0246

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.018690
            Specificity : 0.997095
         Pos Pred Value : 0.625000
         Neg Pred Value : 0.796832
             Prevalence : 0.205764
         Detection Rate : 0.003846
   Detection Prevalence : 0.006153
      Balanced Accuracy : 0.507892

       'Positive' Class : Yes
```

Code:

```
#Changing threshold to increase sensitivity
rf.prob=predict(rf.caret,Test.data,type="prob")
rf.predict=rep("No",22103)
rf.predict[rf.prob[,2]>0.01]="Yes"

confusionMatrix(as.factor(rf.predict),Test.data$No.show,positive="Yes")
```

```
#Changing threshold to increase sensitivity
rf.prob=predict(rf.caret,Test.data,type="prob")
rf.predict=rep("No",22103)
rf.predict[rf.prob[,2]>0.01]="Yes"
confusionMatrix(as.factor(rf.predict),Test.data$No.show,positive="Yes")

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No   7123   368
##        Yes 10432  4180
```

```
##
##                Accuracy : 0.5114
##                  95% CI : (0.5048, 0.518)
##     No Information Rate : 0.7942
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1785
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9191
##             Specificity : 0.4058
##          Pos Pred Value : 0.2861
##          Neg Pred Value : 0.9509
##              Prevalence : 0.2058
##          Detection Rate : 0.1891
##    Detection Prevalence : 0.6611
##       Balanced Accuracy : 0.6624
##
##        'Positive' Class : Yes
##
```

Algorithm	Threshold	Sensitivity	Specificity	Accuracy
Random Forest	0.01	91.91%	40.58%	51.14%

Prediction using KNN

Code:

```
#Default threshold of 0.5

knn.prob=predict(knn.fit,Test.data,type="prob")

knn.predict=rep("No",22103)
knn.predict[knn.prob[,2]>0.5]="Yes"
confusionMatrix(as.factor(knn.predict),Test.data$No.show,positive="Yes")

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No  17448  4470
##        Yes   107    78
##
##                Accuracy : 0.7929
##                  95% CI : (0.7875, 0.7982)
##     No Information Rate : 0.7942
##     P-Value [Acc > NIR] : 0.6887
##
```

```
##                  Kappa : 0.0172
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.017150
##             Specificity : 0.993905
##          Pos Pred Value : 0.421622
##          Neg Pred Value : 0.796058
##              Prevalence : 0.205764
##          Detection Rate : 0.003529
##    Detection Prevalence : 0.008370
##       Balanced Accuracy : 0.505528
##
##        'Positive' Class : Yes
```

```
#Changing threshold to increase sensitivity
knn.predict=rep("No",22103)
knn.predict[knn.prob[,2]>0.08]="Yes"
table(knn.predict,Test.data$No.show)
```

```
##
## knn.predict    No   Yes
##         No   4605   364
##         Yes 12950  4184
```

```
confusionMatrix(as.factor(knn.predict),Test.data$No.show,positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No   4605   364
##        Yes 12950  4184
##
##                Accuracy : 0.3976
##                  95% CI : (0.3912, 0.4041)
##     No Information Rate : 0.7942
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.09
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9200
##             Specificity : 0.2623
##          Pos Pred Value : 0.2442
##          Neg Pred Value : 0.9267
##              Prevalence : 0.2058
##          Detection Rate : 0.1893
##    Detection Prevalence : 0.7752
##       Balanced Accuracy : 0.5911
```

```
##
##        'Positive' Class : Yes
```

Algorithm	Threshold	Sensitivity	Specificity	Accuracy
Random Forest	0.01	91.91%	40.58%	51.14%
KNN	0.08	92%	26.23%	39.76%

Prediction using Logistic Regression

Code:

```
#Default threshold 0.5
cvlog.prob=predict(cv.logit2,Test.data,type="prob")
cvlog.predict=rep("No",22103)
cvlog.predict[cvlog.prob[,2]>0.5]="Yes"
confusionMatrix(as.factor(cvlog.predict),Test.data$No.show,positive="Yes")

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No  17397  4474
##        Yes   158    74
##
##                Accuracy : 0.7904
##                  95% CI : (0.785, 0.7958)
##     No Information Rate : 0.7942
##     P-Value [Acc > NIR] : 0.9199
##
##                   Kappa : 0.0112
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.016271
##             Specificity : 0.991000
##          Pos Pred Value : 0.318966
##          Neg Pred Value : 0.795437
##              Prevalence : 0.205764
##          Detection Rate : 0.003348
##    Detection Prevalence : 0.010496
##       Balanced Accuracy : 0.503635
##
##        'Positive' Class : Yes
##

#Changing threshold to increase sensitivity
cvlog.predict=rep("No",22103)
cvlog.predict[cvlog.prob[,2]>0.13]="Yes"
confusionMatrix(as.factor(cvlog.predict),Test.data$No.show,positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No   3434   345
##        Yes 14121  4203
##
##                Accuracy : 0.3455
##                  95% CI : (0.3392, 0.3518)
##     No Information Rate : 0.7942
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0564
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9241
##             Specificity : 0.1956
##          Pos Pred Value : 0.2294
##          Neg Pred Value : 0.9087
##              Prevalence : 0.2058
##          Detection Rate : 0.1902
##    Detection Prevalence : 0.8290
##       Balanced Accuracy : 0.5599
##
##        'Positive' Class : Yes
##
```

Algorithm	Threshold	Sensitivity	Specificity	Accuracy
Random Forest	0.01	91.91%	40.58%	51.14%
KNN	0.08	92%	26.23%	39.76%
Logistic Regression	0.13	92.41%	19.56%	34.55%

Prediction using Logistic Regression (with under and oversampling)

Code:

```
#Default threshold 0.5
cvlog.prob=predict(cv.logit3,Test.data,type="prob")
cvlog.predict=rep("No",22103)
cvlog.predict[cvlog.prob[,2]>0.5]="Yes"
confusionMatrix(as.factor(cvlog.predict),Test.data$No.show,positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No  12109  1958
```

```
##        Yes  5446  2590
##
##               Accuracy : 0.665
##                 95% CI : (0.6588, 0.6712)
##    No Information Rate : 0.7942
##    P-Value [Acc > NIR] : 1
##
##                  Kappa : 0.2019
##
##  Mcnemar's Test P-Value : <2e-16
##
##            Sensitivity : 0.5695
##            Specificity : 0.6898
##         Pos Pred Value : 0.3223
##         Neg Pred Value : 0.8608
##             Prevalence : 0.2058
##         Detection Rate : 0.1172
##   Detection Prevalence : 0.3636
##      Balanced Accuracy : 0.6296
##
##       'Positive' Class : Yes
##
```

```
#Changing threshold to increase sensitivity
cvlog.predict=rep("No",22103)
cvlog.predict[cvlog.prob[,2]>0.37]="Yes"
confusionMatrix(as.factor(cvlog.predict),Test.data$No.show,positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    No   Yes
##        No   3973   396
##        Yes 13582  4152
##
##               Accuracy : 0.3676
##                 95% CI : (0.3612, 0.374)
##    No Information Rate : 0.7942
##    P-Value [Acc > NIR] : 1
##
##                  Kappa : 0.0671
##
##  Mcnemar's Test P-Value : <2e-16
##
##            Sensitivity : 0.9129
##            Specificity : 0.2263
##         Pos Pred Value : 0.2341
##         Neg Pred Value : 0.9094
##             Prevalence : 0.2058
##         Detection Rate : 0.1878
```

```
##    Detection Prevalence : 0.8023
##       Balanced Accuracy : 0.5696
##
##        'Positive' Class : Yes
```

Algorithm	Threshold	Sensitivity	Specificity	Accuracy
Random Forest	**0.01**	**91.91%**	**40.58%**	**51.14%**
KNN	0.08	92%	26.23%	39.76%
Logistic Regression	0.13	92.41%	19.56%	34.55%
Logistic Regression with under and oversampling	0.37	91.29%	22.63%	36.76%

Results:

The threshold value was decreased until a model sensitivity of 90% was achieved. This would ensure that our model is able to predict 90% of NoShow=1 accurately. From the above table it is clear that as sensitivity was increased, model specificity and accuracy decreased. 90% sensitivity was achieved in all the models, but the best model would be the one that has relatively high sensitivity, specificity and accuracy.

Thus, on the above criteria, Random Forest performed the best on test data. It has a high sensitivity (91%) and relatively high specificity and accuracy (40% and 51% resp.).

SUMMARY

The current analysis involved determining the influence of several predictors on No shows in medical appointments. No Shows need to be reduced, as they result in wastage of time and resources of the medical center and for other patients who need medical attention. Based on Random Forest and Logistic regression, the following predictors were determined to be relevant:

- Days between scheduled day and the appointment day
- Age
- Appointment weekday
- SMS Received
- Scholarship status
- Diabetes
- Time at which appointment was taken
- Alcoholism

It was found that higher the number of days between the scheduled and appointment date, the higher are the chances of no shows. Thus, people are prone to not showing up if the date allotted to them is not close to their scheduled date. This may be because they found another medical center where they were able to get an earlier date. Therefore, the medical centers should try to reduce the number of days a person has to wait before an appointment. They can probably reschedule the appointment to an earlier date in case someone else cancels their appointment.

Another factor that affected no shows was Age. Older people tend to have fewer NoShows as compared to younger people. This may be because younger people may have more rigorous work schedules (or classes in case the person is a student) that may prevent them from visiting the medical center. The other reason may be that older people have more serious medical conditions that they need to get diagnosed or require medical attention, whereas younger people may not have medical conditions that serious and that is why they do not show up.

It was found that the majority of NoShows were on Fridays and Saturdays. Tuesdays and Wednesdays had the least number of NoShows. The reason for this may be that people had other plans during the weekends, which led them to miss their appointment. Since medical appointments take a long time, people may not be willing to change their weekend plans for the appointment. A way to decrease NoShows on weekends can be to provide incentives (lesser wait time) on the weekends. This way, more people would show up to the appointment during weekends.

Another surprising result was that people who received a reminder SMS were still prone to not showing up. This proves that the SMS strategy does not work. According to the study (Targeted Reminder Phone Calls to Patients at High Risk of No-Show for Primary Care Appointment: A Randomized Trial https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5130951/) people who received a reminder phone call seven days before the appointment from the hospital staff were more prone to showing up for their appointment. Therefore, hospitals can try implementing this strategy.

The next factor that was influential was the Bolsa Familia scholarship status. People enrolled in this welfare program were more likely to not show up for their appointment. The welfare program is for people below the poverty line in Brazil. Thus, a conclusion can be drawn that people belonging to lower economic status increase the probability of NoShows. These people may not have the time or resources to be able to visit the medical center once they are allotted a date. Since the program already provides some medical incentives, another incentive like discounted medical care can be added to the benefits.

People with diabetes are also likely not to show up. This can be resolved by also providing insulin medicine/shots or diabetes-related care at a lower cost if they keep their appointment time. This would incentivize people with diabetes to visit and decrease the NoShows.

The time when the appointment was scheduled also impacts the likelihood of a NoShow. Appointments scheduled during the morning (5 am to 12 pm) have least NoShows, whereas those scheduled after 6 pm (evening and night) have the maximum NoShows. Therefore, the hospital staff must follow up with a phone call with the people who scheduled the appointment after 6 pm. This would reduce the likelihood of a NoShow.

Alcoholics are more prone to missing their appointments. To counter this, medical centers can provide additional resources to the patient, like information about rehabilitation centers, counseling services, etc. to the person. This would help those who want to become sober and decrease the NoShows as well.