HarvardX PH125.9x Data Science Capstone Project: Noshow

Francis Magombo

20/03/2021

Contents

Downloading the data files:

```
nstmpfile <- tempfile()
download.file("https://raw.githubusercontent.com/FrancisMag/CYO_Capstone_Projet_NoShow/main/KaggleV2-May

data <- read.csv(nstmpfile, stringsAsFactors =TRUE, header=TRUE)
```

1 Introduction

We will use predictive machine learning algorithms to model and predict what factors are important in determining ifr the patient will show up or not. Predictive machine learning algorithms have become more and more useful in providing insights and solutions or predictions towards existing or emerging problems and hence assisting with in planning and making decisions for the future.The methods used in machine learning will essentially depend on the type of data that is available and also the nature of the problem you are trying to solve. The different methods used include classification models, random forest, k-nearest neighbors among other machine learning algorithms that exist. In this projects we will use logistic regression, random forest, k-nearest neighbors as well as ensemble methods to try and find the most accurate prediction method for the factors influencing the patients to miss their appointments(noshows).

2 Dataset and variables

Total number of observations and variables are tabulate as below:

```
dim(data)
```

[1] 110527 14

The“Noshow_KaggleV2-May-2016” data, which we will use in this project can be found on https://www.kaggle.com/wbadry/noshow-appointment-may-2016

This dataset collects information from 110,527 medical appointments in Brazil and is focused on the question of whether or not patients show up for their appointments. A number of characteristics about the patient are included in each row. Some of the factors under consideration are whether the age, gender, neighborhood or receiving a reminder message influence the noshows or whether it is other additional factors like times period between appointment date and scheduled date or whether it is the neighbourhood where the patients live that are key to follow through with the appointments.

The list of variables is as outlined below:

x
PatientId
AppointmentID
Gender
ScheduledDay
AppointmentDay
Age
Neighbourhood
Scholarship
Hipertension
Diabetes
Alcoholism
Handcap
SMS_received
No.show

The variables used in this dataset are: PatientId, AppointmentID, Gender, ScheduledDay, AppointmentDay, Age, Neighbourhood, Scholarship, Hipertension, Diabetes, Alcoholism, Handcap, SMS_received, and No.show. We see that the the rest of the variables are in character class except for the Handcap, Hipertension, Alcoholism, Age, Scholarship,AppointmentID, PatientId and SMS_received.Below we further explore the nature of this data before we plan for its analysis or any modelling that it can allow.

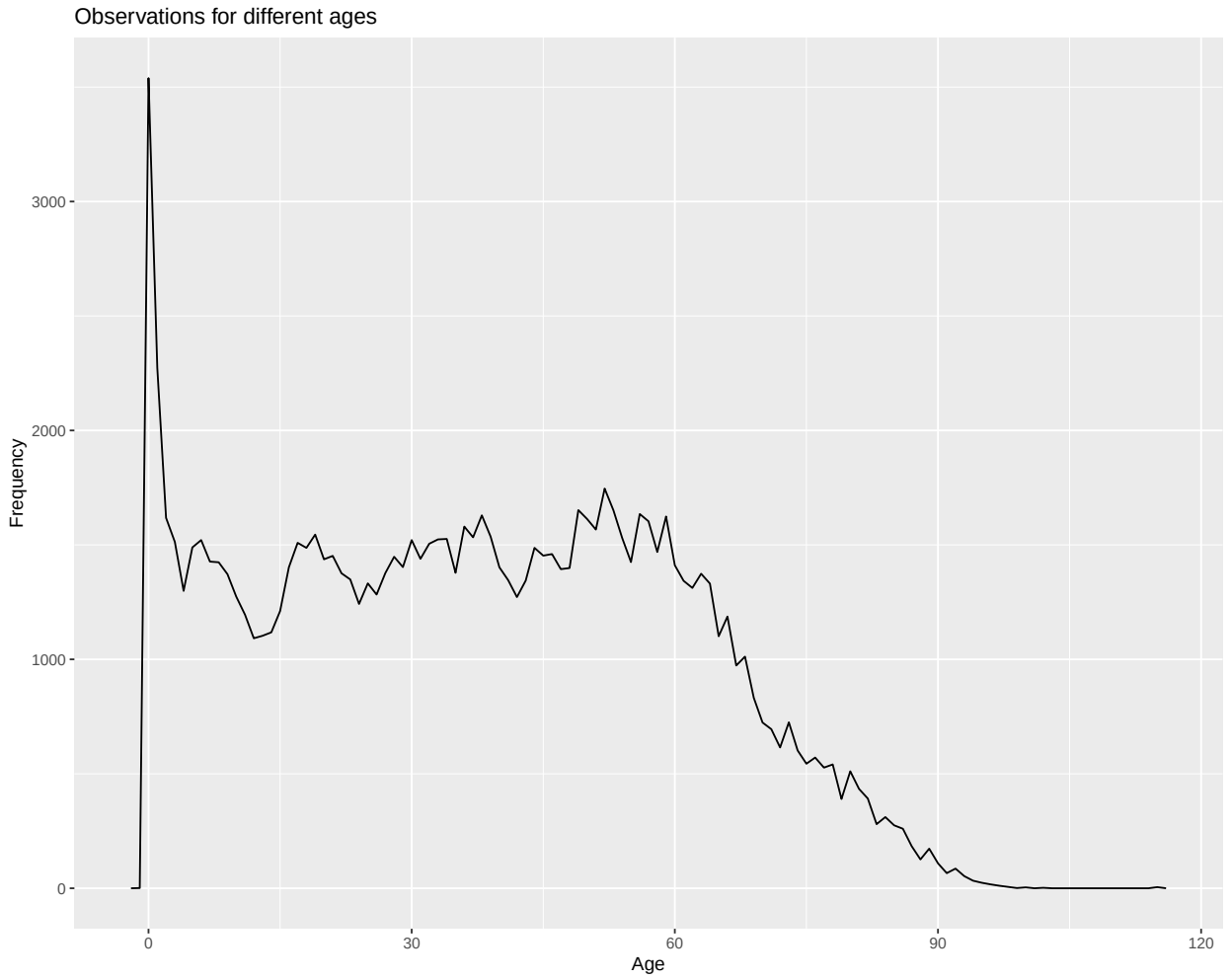
We can see that out of 110,527 patients around 22,319 of them did not show up, which is about 20%.

No.show	n
No	88208
Yes	22319

We disaggregate the number of males and females in order to have a clear picture of the numbers of both groups.

Gender	n
F	71840
M	38687

We also need to know the observations for the different age groups. This is made clear by the graph below.



The data is collected from eighty-one (81) neighbourhoods and contains fourteen (14) variables. The total number of observations is 110,527. There are 71,840 females and 38,687 males in the dataset. The age group is from -1 to 115 years and the mean is 37 years. It is unlikely that one would have an age of less than zero and that there would be seven patients with an age above 100 years. So the analysis will need to exclude those of less than zero age and those above 100 years old. The population is distributed unevenly in the different neighborhoods with one neighborhood having a population of above 5,700 and the next close to 6,000 whilst some neighborhoods have close to zero participants.

3 Goal of the project

The aim of this project is to make come up with a prediction model for the patients that are likely to miss their medical appointment using the variables in the available dataset. We will explore and model using the variables and the different modelling methods to discover which model gives the most accurate prediction for the noshows.

4 Methods and Analysis

Factors to be tested:We will explore the variables available and already described above to see which ones could be included in the models.We will start by considering the demographic factors.

4.1 Data Cleaning

We first note that some of the variables are in character class format and will need to be converted to factor class format to facilitate easy calculations in the models to be used. The variables concerned include:Gender, Neighbourhood, Scholarship, Hipertension, Diabetes, Alcoholism, SMS_received, and No.show.

Change the data type of some columns:

```
data <- mutate_at(data,
                  vars('Gender', 'Neighbourhood',  'Scholarship',
                       'Hipertension',  'Diabetes', 'Alcoholism',
                       'SMS_received', 'No.show'), as.factor)
```

Next we convert the variable Handcap to binary form so that we only assess whether one is handicapped or not.The "Handcap" is described at data description as a binary variable but the values in the column range from 0 to 4 which should not be. Therefore, we will consider any value of "Handcap" bigger than 0 as 1 in this case. We note thet 2,241 (2%) of the patients were handicapped.

We will keep the Handcap variable into a binary form maintaining only two levels fr the factr varable.

	x
0	108286
1	2241

Instead to keeping the "No"or "Yes"in the No.show variable we will convert "Yes" to "1" and "No" to "0" to maintain a similar format being used in the other categorical variables and for easy calculation.

	x
0	88208
1	22319

Likewise we will do the same for the variable "Gender". We note that 71,840 (65.0%) of the patients were female.

	x
0	71840
1	38687

We also will change the variable "Age" from the character class format to numeric class format.

We then check is there are any missing values.It is evident that there are no missing values in the dataset.

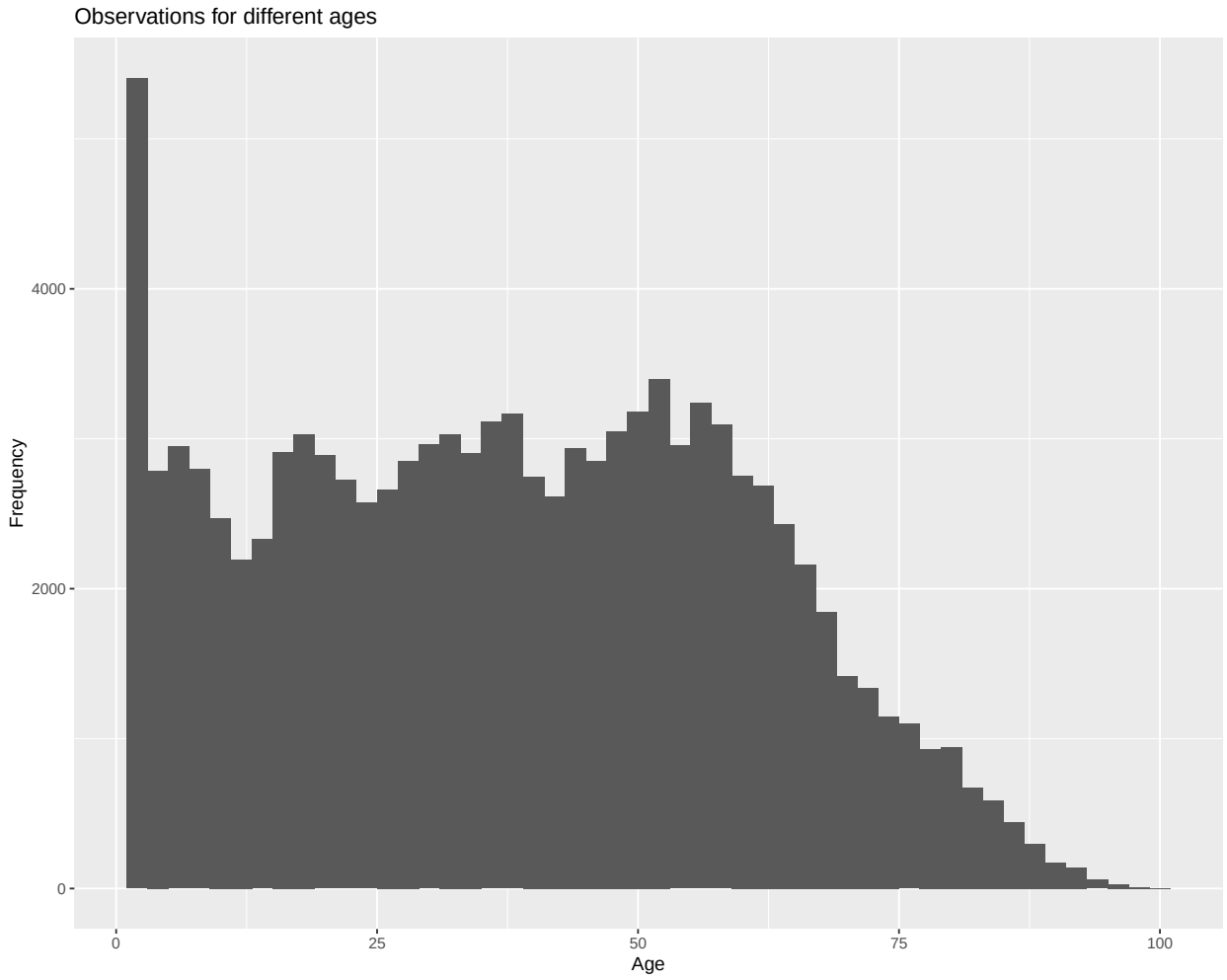
Missing values:

	x
PatientId	0
AppointmentID	0
Gender	0
ScheduledDay	0
AppointmentDay	0
Age	0
Neighbourhood	0
Scholarship	0
Hipertension	0
Diabetes	0
Alcoholism	0
Handcap	0
SMS_received	0
No.show	0

We noted that the ages of the patients ranged from -1 to 115 years. To make more practical calculations we will need to keep the data that has the most appropriate age range which in this case we will consider to be above zero and up to 100 years. We now need to remove the one below zero years of age and above 100 years.

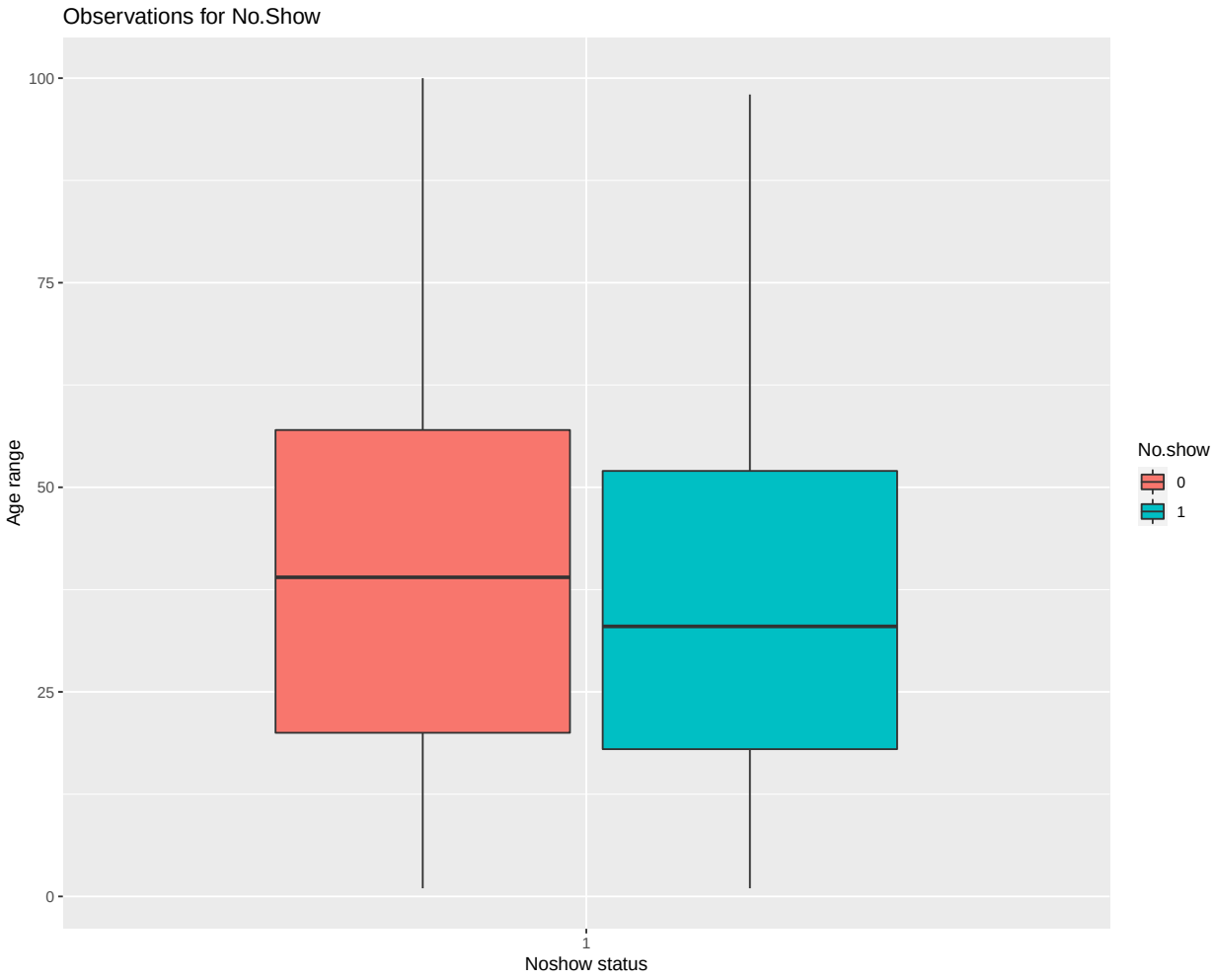
The updated patient age group distribution as per the selected lower limit (0) and upper limit (100 years) is shown in this graph.

```
ggplot(data=data, aes(x=Age)) + geom_histogram(binwidth =2) +
  xlab("Age") +
  ylab("Frequency") +
  ggtitle("Observations for different ages")
```



We see that the age range of the noshows is younger than that of those who actually turned up for their appointments.

Plot Age and No.show status:



4.2 Modeling approach

We select the variables:Age, Gender, Scholarship, Hipertension, Diabetes, Alcoholism, SMS_received, No.show to be used and remove AppointmentDay, ScheduledDay, and PatientId from the first model.

```
data <- data %>%
  dplyr::select('Age', 'Gender', 'Scholarship',  'Hipertension',
                'Diabetes', 'Alcoholism', 'SMS_received', 'No.show')
```

The variable "Neighbourhood" was also removed later on because randomForest could not handle a variable with more than 53 categories and so presented an error message.

We will take a small sample of the dataset for the analysis given the size of the dataset in comparison to the capacity of the laptop computer being used.

```
    set.seed(1998, sample.kind = "Rounding")

    #Create index vector
    idx <- seq_len(nrow(data))

   #Sample from the index vector
```

```
    samp <- sample(idx, 10000)

    #create a dataframe of 10,000 random rows
    data_samp <- data[samp, ]
```

4.3 Data Partitioning (Training and Test data sets)

We will now develop an algorithm using only the training set.The validation(test) subset will be 10 percent of the total data. The test set will be used to evaluate the result from the training set. The 10 percent was chosen arbitrarily as the proportion that is mostly used in other machine learning algorithms. Since we have categorical data we will be reporting on the proportion of the No.shows that will be correctly predicted in the test set using the overall accuracy measure.The overall accuracy is simply defined as the overall proportion that is predicted correctly. Validation set will be 10% of the data.

4.4 Modeling results

4.4.1 Model 1: logistic regression

We start by training logistic regression model which we test afterwards.

Training the logistic regression model:

```
train_glm <- train(No.show ~., method = "glm", data = train_set)

pander(summary(train_glm))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.4	0.06395	-21.89	3.342e-106
Age	-0.006875	0.001413	-4.865	1.144e-06
Gender1	-0.0419	0.05733	-0.7308	0.4649
Scholarship1	0.1723	0.08462	2.036	0.04173
Hipertension1	0.001971	0.08371	0.02354	0.9812
Diabetes1	-0.05456	0.1192	-0.4577	0.6472
Alcoholism1	0.08993	0.1609	0.5587	0.5763
SMS_received1	0.7595	0.05387	14.1	3.846e-45

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	9082 on 8999 degrees of freedom
Residual deviance:	8843 on 8992 degrees of freedom

Making a prediction for the logistic regression model:

```
y_hat_glm <- predict(train_glm, test_set, type = "raw")
```

Evaluating the logistic regression model:

```
confusionMatrix(y_hat_glm, test_set$No.show)$overall[["Accuracy"]]
```

[1] 0.797

We then evaluate the accuracy of the logistic regression model. Which we find to be 79.7%. We note that the variables Age, Scholarship1, SMS_received1 have significant contribution to the patients not showing up at for their appointments in this model. The Age is negatively correlated with not showing up (older patients are more prone to keep their appointments), having scholarship is positively correlated to the patient not showing up and receiving and SMS also is positively correlated to the noshow (the majority of those who received the sms still did not show up for their appointments). The accuracy of the prediction model is 79.7%. we will try to use the significant variables only for the next logistic regression model.

```
train_glm2 <- train(No.show ~ Age + Scholarship + SMS_received, method = "glm", data = train_set)

pander(summary(train_glm2))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.417	0.05693	-24.9	8.304e-137
Age	-0.006856	0.001199	-5.718	1.079e-08
Scholarship1	0.1819	0.08381	2.171	0.02996
SMS_received1	0.761	0.05382	14.14	2.131e-45

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	9082 on 8999 degrees of freedom
Residual deviance:	8844 on 8996 degrees of freedom

```
y_hat_glm2 <- predict(train_glm2, test_set, type = "raw")

confusionMatrix(y_hat_glm2, test_set$No.show)$overall[["Accuracy"]]
```

[1] 0.797

The accuracy of the model prediction does not change if we only include the variables that had been shown to have a significant contribution to the first logistic regression model (Age, Scholarship and SMS_received).It remains 79.7. Therefore we will go ahead and use all the variables initially selected for the first logistic regression model for the subsequent machine learning algorithms.In the next model we will use the kNN method to make the prediction.

4.4.2 Model 2: kNN

Training the kNN model:

```
train_knn <- train(No.show ~ ., method = "knn", data = train_set)

pander(summary(train_knn))
```

	Length	Class	Mode
learn	2	-none-	list
k	1	-none-	numeric
theDots	0	-none-	list
xNames	7	-none-	character
problemType	1	-none-	character
tuneValue	1	data.frame	list
obsLevels	2	-none-	character
param	0	-none-	list

Making the kNN model prediction:

```
y_hat_knn <- predict(train_knn, test_set, type = "raw")
```

Evaluating the accuracy of the model:

```
confusionMatrix(y_hat_knn, test_set$No.show)$overall[["Accuracy"]]
```

[1] 0.79

The accuracy in the kNN model is 79% which is below the one for the logistic regression model 79.7%. We also try applying the randomForest model using the same variables.

4.4.3 Model 3 : randomForest

The first randomForest method using ntree (500, 1000, 1500, 2000) and mtry 3:8 had an accuracy of 79.5%. The second method with ntree (500, 750,1000, 1500, 2000) and mtry 3:7 had an accuracy of 79.6%. However, we run only the the selected, the second, model due to the lengthy time it takes to complete each one of them.

The selected randomForest model is the one below:

```
rf_ranges <- list(ntree = c(500, 1000, 1500, 2000), mtry = 3:7)
rf_tune <- tune(randomForest, No.show ~ ., data =
                  train_set, ranges = rf_ranges)
```

```
rf_tune$best.parameters
```

ntree mtry 2 1000 3

```
rf_best <- rf_tune$best.model
rf_best
```

Call: best.tune(method = randomForest, train.x = No.show ~ ., data = train_set, ranges = rf_ranges) Type of random forest: classification Number of trees: 1000 No. of variables tried at each split: 3

```
    OOB estimate of  error rate: 20.41%
```

Confusion matrix: 0 1 class.error 0 7157 16 0.002230587 1 1821 6 0.996715928

Making the prediction:

```
##predict
y_hat_rf <- predict(rf_best, test_set)
```

Evaluating the accuracy of the randomForest model:

```
confusionMatrix( y_hat_rf, test_set$No.show)$overall[["Accuracy"]]
```

[1] 0.796

Estimating the variable importance:

```
imp <- importance(rf_best)
imp %>% knitr::kable()
```

	MeanDecreaseGini
Age	159.59480
Gender	16.16459
Scholarship	10.69724
Hipertension	13.44392
Diabetes	11.81455
Alcoholism	10.41115
SMS_received	68.87433

The model is 79.6% accurate which is less than the logistic regression model(79.7%) and better than the kNN model(79%).On the basis of the variable importance estimate it is evident tthat the variable that is most influential in the prediction of noshows in this model is the Age followed by SMS_received and Gender.We will now try to see if the ensemble method combining kNN and randomForest will provide us with an improved accuracy.

4.4.4 Model 4: ensembles (random forest and kNN)

Prediction with ensemble method:

```
p_rf <- predict(rf_best, test_set, type = "prob")
 p_rf<- p_rf / rowSums(p_rf)
 p_knn <- predict(train_knn, test_set, type = "prob")
 p <- (p_rf + p_knn)/2
 y_pred <- factor(apply(p, 1, which.max)-1)
```

Evaluating the accuracy:

```
confusionMatrix(y_pred, test_set$No.show)$overall[["Accuracy"]]
```

[1] 0.799

The ensemble model improves the accuracy to 79.9% .

5 Conclusion

The ensemble method, combining kNN and randomForest results in an accuracy of 79.9% which is the best accuracy so far in the methods tried.The logistic regression model and the randomForest had an accuracy of 79.7% while the kNN model had an accuracy of 79%.

We choose the ensemble method as our method for making the predition for the noshows. However, we also note that the models could also have included other variables like Neighbourhood, and the period between appointment date and the scheduling date to improve upon it. This could be explored in future to see how the model improves on the accuracy.Additionally, the number of randomForest models that could be tried are limited by the time it takes to complete the calculations using the computer that I have at my disposal.

6 Appendix- Environment

[1] “Operating System:” _
platform x86_64-w64-mingw32
arch x86_64
os mingw32
system x86_64, mingw32
status
major 4
minor 0.4
year 2021
month 02
day 15
svn rev 80002
language R
version.string R version 4.0.4 (2021-02-15) nickname Lost Library Book