Code ▾

FLU Prediction based on Symptoms

Influenza Research Database (IRD) provides a resource for the influenza virus research community that facilitates an understanding of the influenza virus and how it interacts with the host organism, leading to new treatments and preventive actions.

Influenza, commonly known as “the flu”, is an infectious disease caused by an influenza virus. High fever, runny nose, sore throat, headache, coughing, etc. are the most common symptoms. These symptoms typically begin two days after exposure to the virus and most last less than a week. Complications of influenza may include viral pneumonia, secondary bacterial pneumonia, sinus infections, and worsening of previous health problems such as asthma or heart failure. Once detected at an earlier stage based on the symptoms, the flu can be treated by getting plenty of rest, drinking plenty of liquids and, take medications to relieve the fever and muscle aches associated with the flu if necessary.

By applying statistical methods to this database, we would like to identify if a person is infected by FLU or not infected by FLU based on various factors such as symptoms exhibited by the patient.

Loading R-packages

Hide

```
library(readxl)
library("rpart", lib.loc="C:/Program Files/R/R-3.3.3/library")
library("rpart.plot", lib.loc="~/R/win-library/3.3")
library("rattle", lib.loc="~/R/win-library/3.3")
library("RGtk2", lib.loc="~/R/win-library/3.3")
library("randomForest", lib.loc="~/R/win-library/3.3")
library(dplyr)
library(plotrix)
library(car)
library(rpart.plot)
library(rpart)
library(rattle)
library("ggplot2")
library("scales")
library("directlabels")
library("tidyr")
library("RColorBrewer")
library("ROCR")
library(randomForest)
library(class)
```

Loading dataset

Hide

```
FluDB <- read_excel("C:/Users/Supriya Khadake/Desktop/SPRING 2018/ITMD 529/ITMD529_Pro
ject/Data/FluDB.xlsx")
```

Statistical description of the dataset

Hide

```
summary(FluDB)
```

```
 HostIdentifier       Location              Age           Gender       Temperature
MedicalConditions  RunningNose         Cough           Myalgia          Headach
e        ThroatAche        Fever           Fatigue
 Length:1699        Length:1699        Min.   : 1.0   Min.   :0.000   Min.   : 96.2
0   Min.   :0.0000    Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
0   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
 Class :character   Class :character   1st Qu.:19.0   1st Qu.:0.000   1st Qu.: 98.6
0   1st Qu.:0.0000    1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
0   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
 Mode  :character   Mode  :character   Median :20.0   Median :0.000   Median : 99.2
0   Median :0.0000    Median :1.0000   Median :1.0000   Median :1.0000   Median :1.000
0   Median :1.000   Median :1.0000   Median :1.0000
                                       Mean   :28.9   Mean   :0.452   Mean   : 99.5
7   Mean   :0.2831    Mean   :0.5892   Mean   :0.7422   Mean   :0.6157   Mean   :0.524
4   Mean   :0.528   Mean   :0.6268   Mean   :0.5827
                                       3rd Qu.:39.5   3rd Qu.:1.000   3rd Qu.:100.1
0   3rd Qu.:1.0000    3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.000
0   3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000
                                       Max.   :97.0   Max.   :1.000   Max.   :107.2
4   Max.   :1.0000    Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000
0   Max.   :1.000   Max.   :1.0000   Max.   :1.0000
    Vomiting      FluTestStatus
 Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.0000
 Median :0.0000   Median :1.0000
 Mean   :0.4438   Mean   :0.7481
 3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.0000
```

Assigning variables

Hide

```
HostIdentifier = FluDB$HostIdentifier
FluDB$Location = as.factor(FluDB$Location)
FluDB$Age = FluDB$Age
FluDB$Gender = as.factor(FluDB$Gender)
FluDB$Temperature = FluDB$Temperature
FluDB$MedicalConditions = as.factor(FluDB$MedicalConditions)
FluDB$RunningNose = as.factor(FluDB$RunningNose)
FluDB$Cough = as.factor(FluDB$Cough)
FluDB$Myalgia = as.factor(FluDB$Myalgia)
FluDB$Headache = as.factor(FluDB$Headache)
FluDB$ThroatAche = as.factor(FluDB$ThroatAche)
FluDB$Fever = as.factor(FluDB$Fever)
FluDB$Fatigue = as.factor(FluDB$Fatigue)
FluDB$Vomiting = as.factor(FluDB$Vomiting)
FluDB$FluTestStatus = as.factor(FluDB$FluTestStatus)
```

Splitting data into train(70%) for model selection and test(30%) data for evaluation.

Hide

```
set.seed(42)
FluDB=FluDB[sample(nrow(FluDB)),]
select.data= sample (1:nrow(FluDB), 0.7*nrow(FluDB))
train.data= FluDB[select.data,]
test.data= FluDB[-select.data,]
```

To display the number of rows for training and testing data

Hide

```
nrow(test.data)
```

```
[1] 510
```

Hide

```
nrow(train.data)
```

```
[1] 1189
```

Structure of the train data

Hide

```
str(train.data)
```

```
Classes tbl_df, tbl and 'data.frame':   1189 obs. of  15 variables:
 $ HostIdentifier   : chr  "23269979" "F5039" "NIGSP_DOA_00024" "23257879" ...
 $ Location         : Factor w/ 42 levels "Alabama","Alaska",..: 23 37 21 28 21 28 21
37 37 37 ...
 $ Age              : num  18 20 87 19 81 34 60 20 20 20 ...
 $ Gender           : Factor w/ 2 levels "0","1": 2 1 2 2 1 1 1 2 1 2 ...
 $ Temperature      : num  100.4 99.2 101.4 102 101.7 ...
 $ MedicalConditions: Factor w/ 2 levels "0","1": 1 2 2 1 2 1 2 2 1 1 ...
 $ RunningNose      : Factor w/ 2 levels "0","1": 2 1 1 2 2 2 1 1 1 1 ...
 $ Cough            : Factor w/ 2 levels "0","1": 2 1 2 2 2 2 1 1 1 1 ...
 $ Myalgia          : Factor w/ 2 levels "0","1": 2 1 1 2 2 2 1 1 1 1 ...
 $ Headache         : Factor w/ 2 levels "0","1": 2 1 1 2 2 1 1 1 1 1 ...
 $ ThroatAche       : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 1 1 1 ...
 $ Fever            : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 1 1 1 1 ...
 $ Fatigue          : Factor w/ 2 levels "0","1": 2 1 1 1 2 2 1 1 1 1 ...
 $ Vomiting         : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 2 1 1 1 ...
 $ FluTestStatus    : Factor w/ 2 levels "0","1": 2 1 2 2 2 2 2 1 1 1 ...
```

Structure of the test data

Hide

```
str(test.data)
```

```
Classes tbl_df, tbl and 'data.frame':   510 obs. of  15 variables:
 $ HostIdentifier   : chr  "F5005" "F4018C13" "F4031C30" "23301961" ...
 $ Location         : Factor w/ 42 levels "Alabama","Alaska",..: 37 37 37 28 21 37 21
21 21 21 ...
 $ Age              : num  20 20 20 15 59 20 39 1 56 53 ...
 $ Gender           : Factor w/ 2 levels "0","1": 1 2 1 2 2 1 1 1 2 2 ...
 $ Temperature      : num  99.2 99.2 98.6 102 102.6 ...
 $ MedicalConditions: Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 2 2 ...
 $ RunningNose      : Factor w/ 2 levels "0","1": 1 2 1 2 1 2 1 2 2 2 ...
 $ Cough            : Factor w/ 2 levels "0","1": 1 2 1 2 1 2 1 2 2 2 ...
 $ Myalgia          : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 2 2 2 ...
 $ Headache         : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 2 2 2 ...
 $ ThroatAche       : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 2 2 2 ...
 $ Fever            : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 2 2 ...
 $ Fatigue          : Factor w/ 2 levels "0","1": 1 2 2 1 1 2 1 2 2 2 ...
 $ Vomiting         : Factor w/ 2 levels "0","1": 1 1 1 2 2 2 2 2 2 2 ...
 $ FluTestStatus    : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
```

Exploratory Data Analysis

Exploratory Data Analysis(EDA) is a critical step by which team discovered which parameters are most significant in determining the desired outcome. In our case we are trying to predict FLU based on symptoms and its more important to know which symptoms affect the most or if any of the symptom is least important and does not make any difference. With the help of plots for scenario of FluTestStatus as positive it was known that all of the 8 symptoms are relevant

Considering only patients having FLU

Hide

```
FluDB_histogram=filter(FluDB,FluTestStatus==1)
```

Quantitative Variables (Continuous Predictors)

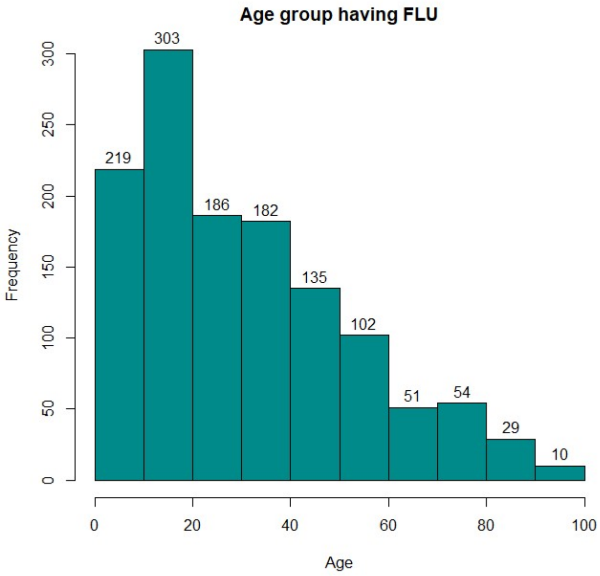
Our current data includes Age and Temperature as Numeric variables. From the plots we observed that people within Age group 10 to 20 and temperature between 99 to 100 has FLU status as positive. Also, we saw that as Age increases the frequency of Flustatus being positive decreases

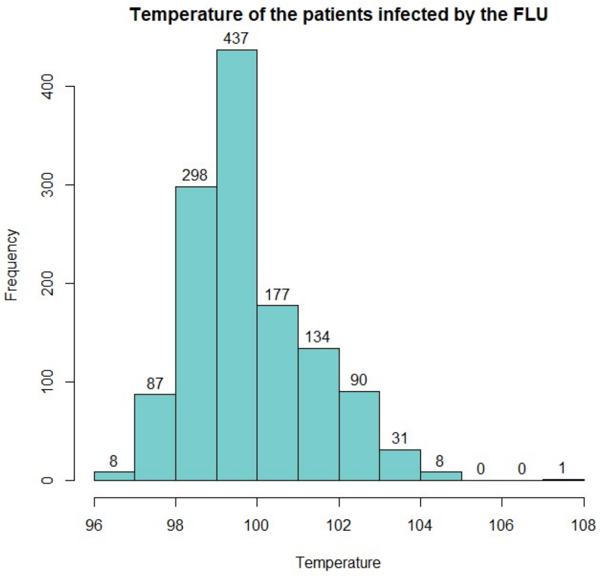
Hide

```
par(mfrow=c(1,2))
hist_age = hist(FluDB_histogram$Age, col = "cyan4", xlab="Age", main = "Age group having FLU" )
text(hist_age$mids, hist_age$counts, labels = hist_age$counts, adj=c(0.5, -0.5))
```

Hide

```
hist_temp = hist(FluDB_histogram$Temperature, col="darkslategray3", xlab="Temperature", main = "Temperature of the patients infected by the FLU")
text(hist_temp$mids, hist_temp$counts, labels = hist_temp$counts, adj=c(0.5, -0.5))
```



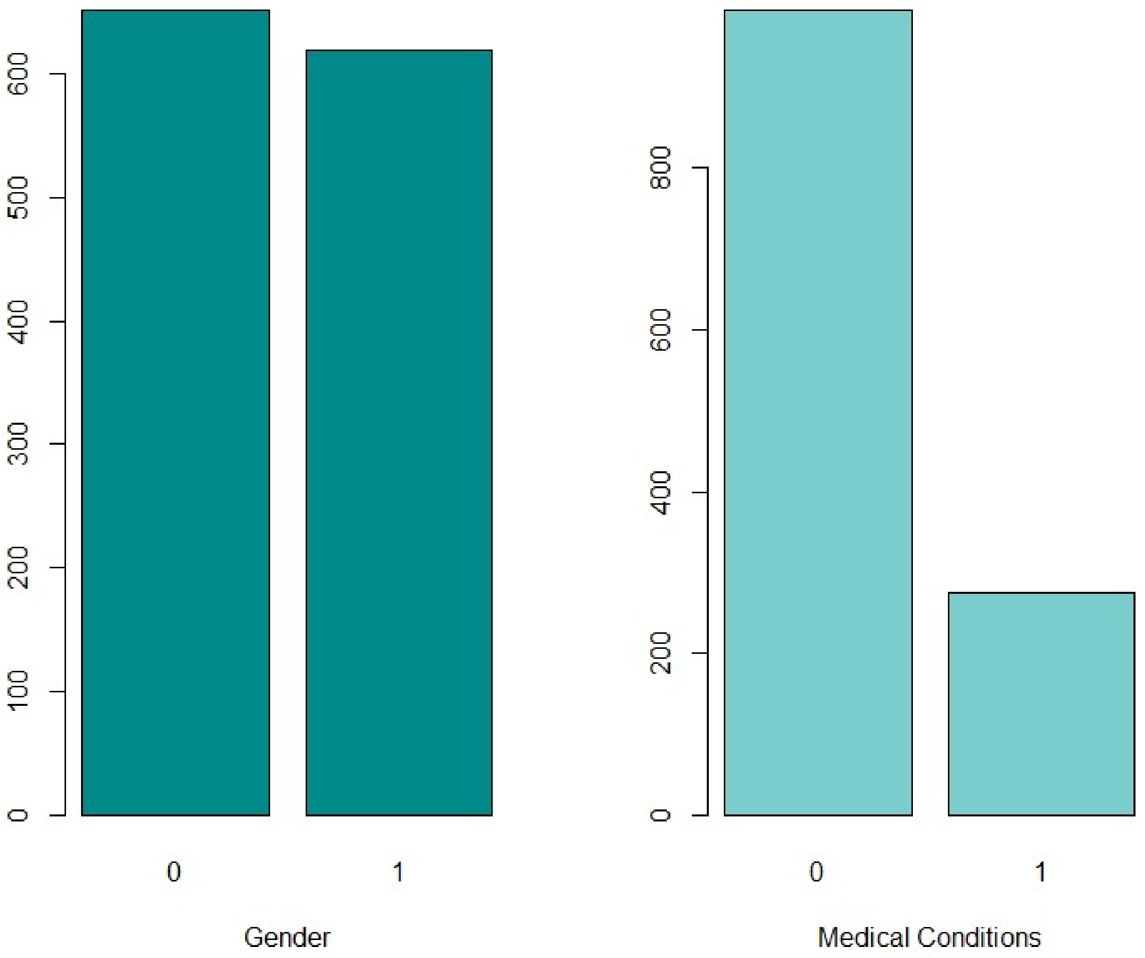


Qualitative Variables (Categorical Predictors)

Histogram for Gender and Medical Conditions with Flu Status as positive: We can see that female are more to have FLU status as positive and medical condition does not affect that much. Also, Female are more susceptable to flu but differ against the range of men marginally

Hide

```
par(mfrow=c(1,2))
plot_gender = plot(FluDB_histogram$Gender,col="cyan4",xlab="Gender")
plot(FluDB_histogram$MedicalConditions,col="darkslategray3",xlab="Medical Conditions")
```



Histogram for all the symptoms with Flu Status as positive:

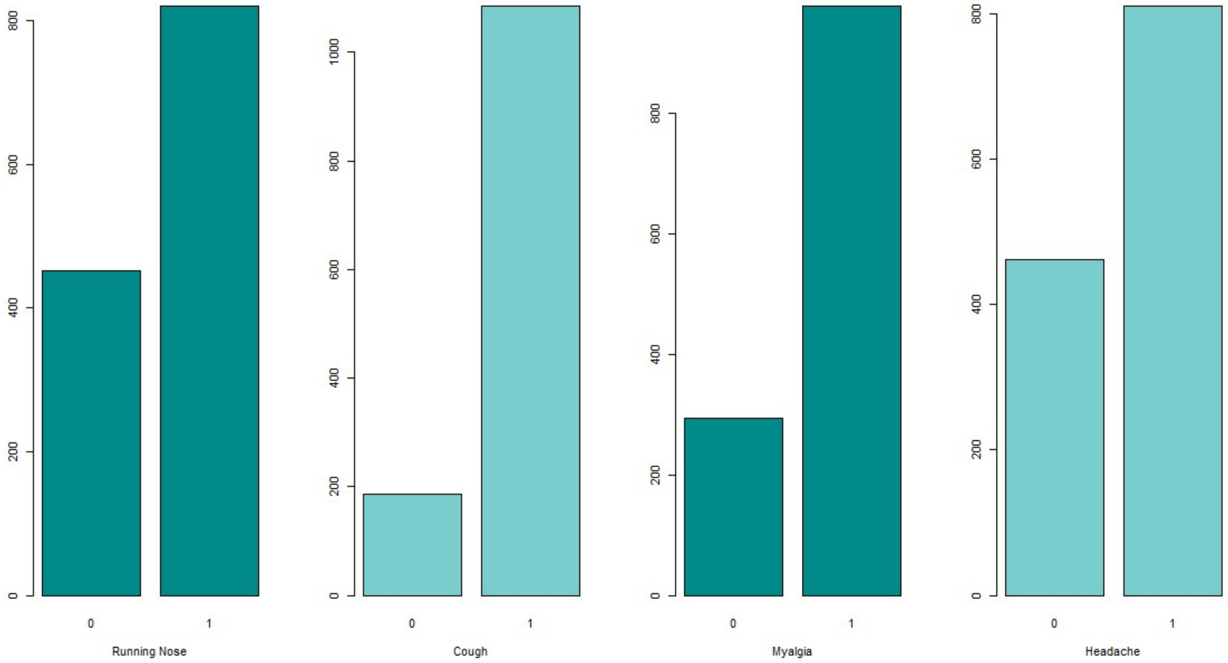
This shows, most of these symptoms can indicate that a person is suffering from flu.

Hide

```
par(mfrow=c(1,4))
plot(FluDB_histogram$RunningNose,col="cyan4",xlab="Running Nose")
plot(FluDB_histogram$Cough,col="darkslategray3",xlab="Cough")
```

Hide

```
plot(FluDB_histogram$Myalgia,col="cyan4",xlab="Myalgia")
plot(FluDB_histogram$Headache,col="darkslategray3",xlab="Headache")
```



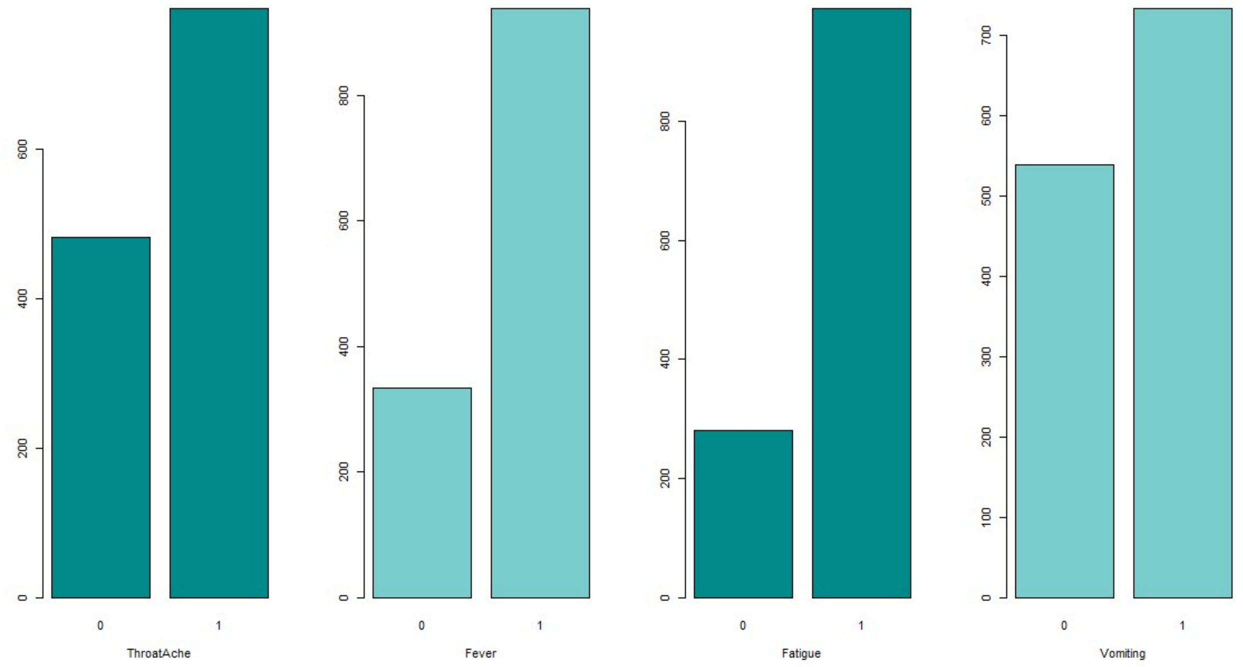
Histogram for location with Flu Status as positive:

Hide

```
par(mfrow=c(1,4))
plot(FluDB_histogram$ThroatAche,col="cyan4",xlab="ThroatAche")
plot(FluDB_histogram$Fever,col="darkslategray3",xlab="Fever")
```

Hide

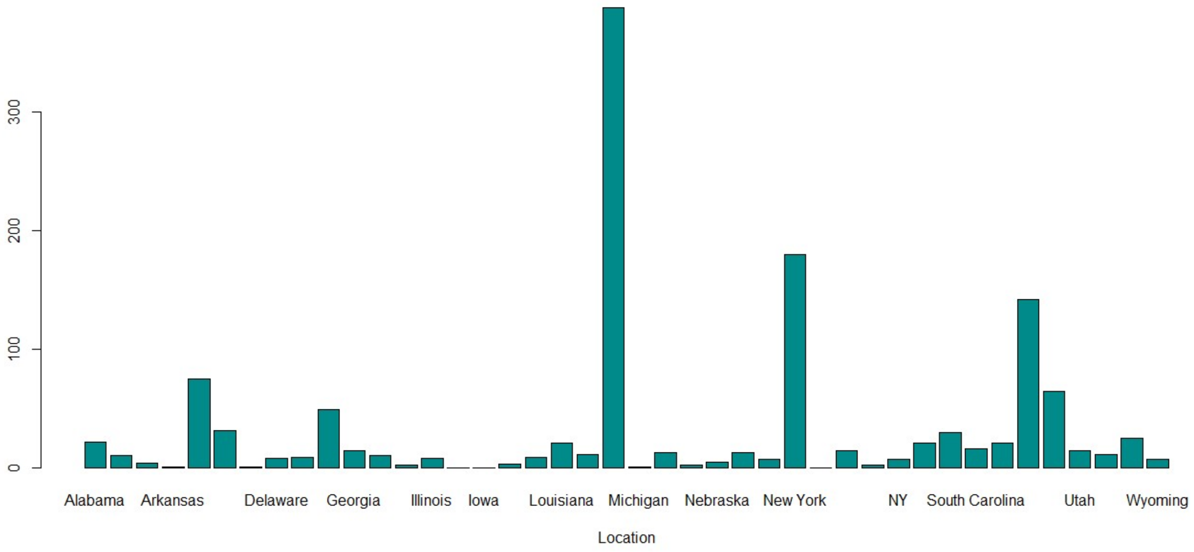
```
plot(FluDB_histogram$Fatigue,col="cyan4",xlab="Fatigue")
plot(FluDB_histogram$Vomiting,col="darkslategray3",xlab="Vomiting")
```



Histogram for location with Flu Status as positive:

Hide

```
par(mfrow=c(1,1))
plot(FluDB_histogram$Location,col="cyan4",xlab="Location")
```



Logistic Regression Model

Building model considering all the variables

Hide

```
log0 = glm(FluTestStatus ~ Location + Age + Temperature + Gender + MedicalConditions + RunningNose + Cough
         + Myalgia + Headache + ThroatAche + Fever + Fatigue + Vomiting, data = train.data, family = "binomial")
```

```
glm.fit: algorithm did not convergeglm.fit: fitted probabilities numerically 0 or 1 occurred
```

Hide

```
summary(log0)
```

```
Call:
glm(formula = FluTestStatus ~ Location + Age + Temperature + 
    Gender + MedicalConditions + RunningNose + Cough + Myalgia + 
    Headache + ThroatAche + Fever + Fatigue + Vomiting, family = "binomial", 
    data = train.data)

Deviance Residuals: 
       Min          1Q      Median          3Q         Max  
-2.384e-04  -2.100e-08   2.100e-08   2.100e-08   2.175e-04  

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                  -1.241e+03  3.655e+05  -0.003    0.997
LocationAlaska               -2.125e+02  1.090e+05  -0.002    0.998
LocationArizona              -2.460e+02  1.595e+05  -0.002    0.999
LocationArkansas             -2.934e+02  3.607e+05  -0.001    0.999
LocationCalifornia            9.404e+01  6.155e+04   0.002    0.999
LocationColorado             -2.814e+01  7.679e+04   0.000    1.000
LocationDelaware             -2.435e+02  1.179e+05  -0.002    0.998
LocationDistrict of Columbia -9.411e+01  1.241e+05  -0.001    0.999
LocationFlorida               1.901e+01  6.753e+04   0.000    1.000
LocationGeorgia              -2.058e+02  9.188e+04  -0.002    0.998
LocationHawaii               -2.178e+02  1.054e+05  -0.002    0.998
LocationIdaho                -2.753e+02  3.613e+05  -0.001    0.999
LocationIllinois             -4.088e+01  1.161e+05   0.000    1.000
LocationIndiana              -1.362e+02  3.599e+05   0.000    1.000
LocationIowa                 -5.617e+01  3.602e+05   0.000    1.000
LocationKansas               -3.914e+01  1.605e+05   0.000    1.000
LocationKentucky             -2.722e+01  1.436e+05   0.000    1.000
LocationLouisiana            -1.955e+01  9.043e+04   0.000    1.000
LocationMaryland             -4.200e+01  1.141e+05   0.000    1.000
LocationMassachusetts         1.031e+02  5.457e+04   0.002    0.998
LocationMichigan             -2.812e+02  3.612e+05  -0.001    0.999
LocationMississippi          -2.232e+02  1.098e+05  -0.002    0.998
LocationMontana              -2.203e+02  3.620e+05  -0.001    1.000
LocationNebraska             -2.161e+02  1.605e+05  -0.001    0.999
LocationNevada               -2.371e+02  1.063e+05  -0.002    0.998
LocationNew Jersey           -2.395e+02  1.558e+05  -0.002    0.999
LocationNew York             -5.058e+01  5.363e+04  -0.001    0.999
LocationNew York,NY          -4.989e+01  3.600e+05   0.000    1.000
LocationNorth Carolina       -1.728e+02  8.633e+04  -0.002    0.998
LocationNY                   -1.337e+02  5.554e+04  -0.002    0.998
LocationOhio                 -2.868e+01  9.425e+04   0.000    1.000
LocationOklahoma             -2.291e+02  7.047e+04  -0.003    0.997
LocationSouth Carolina       -2.110e+02  9.856e+04  -0.002    0.998
LocationSouth Dakota         -4.636e+01  8.764e+04  -0.001    1.000
LocationTennessee            -1.327e+02  5.273e+04  -0.003    0.998
```

```
LocationTexas                4.480e+00  6.089e+04   0.000    1.000
LocationUtah                 4.010e+01  9.393e+04   0.000    1.000
LocationVirginia            -2.153e+02  1.076e+05  -0.002    0.998
LocationWashington          -1.819e+01  7.896e+04   0.000    1.000
LocationWyoming             -2.179e+02  1.526e+05  -0.001    0.999
Age                         -1.088e-01  3.046e+02   0.000    1.000
Temperature                  1.291e+01  3.650e+03   0.004    0.997
Gender1                     -5.025e+00  1.308e+04   0.000    1.000
MedicalConditions1          -8.614e+01  8.993e+03  -0.010    0.992
RunningNose1                 4.612e+00  1.637e+04   0.000    1.000
Cough1                       4.105e+01  1.038e+04   0.004    0.997
Myalgia1                    -3.793e+01  8.148e+03  -0.005    0.996
Headache1                   -3.900e+01  1.073e+04  -0.004    0.997
ThroatAche1                  7.149e-04  8.126e+03   0.000    1.000
Fever1                       3.081e+01  1.636e+04   0.002    0.998
Fatigue1                     2.493e+02  2.203e+04   0.011    0.991
Vomiting1                   -1.069e+00  1.067e+04   0.000    1.000

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1.3666e+03  on 1188  degrees of freedom
Residual deviance: 3.6106e-07  on 1137  degrees of freedom
AIC: 104

Number of Fisher Scoring iterations: 25
```

Without Location

By individual parameter we can see that location is not sugnificant to the response variable Hence we will go ahead and eliminate the same and rebuild model.

Hide

```
log1 = glm(FluTestStatus ~ Age + Temperature + Gender + MedicalConditions + RunningNos
e + Cough
+ Myalgia + Headache + ThroatAche + Fever + Fatigue + Vomiting, data = train.data, fam
ily = "binomial")
```

```
glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Hide

```
summary(log1)
```

```
Call:
glm(formula = FluTestStatus ~ Age + Temperature + Gender + MedicalConditions + 
    RunningNose + Cough + Myalgia + Headache + ThroatAche + Fever + 
    Fatigue + Vomiting, family = "binomial", data = train.data)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.0764  -0.0209   0.0000   0.0000   3.3938  

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -146.46125   35.56317  -4.118 3.82e-05 ***
Age                   0.08284    0.01695   4.887 1.02e-06 ***
Temperature           1.43046    0.35887   3.986 6.72e-05 ***
Gender1               0.99926    0.48582   2.057 0.039701 *  
MedicalConditions1   -1.89013    0.59510  -3.176 0.001492 ** 
RunningNose1         -3.74041    0.80470  -4.648 3.35e-06 ***
Cough1                1.95532    0.70194   2.786 0.005343 ** 
Myalgia1              2.84549    0.73855   3.853 0.000117 ***
Headache1            -2.56710    0.70699  -3.631 0.000282 ***
ThroatAche1          -0.31999    0.63446  -0.504 0.614020    
Fever1                0.90018    0.57809   1.557 0.119432    
Fatigue1             26.58938 1316.28038   0.020 0.983884    
Vomiting1             4.69705    0.62195   7.552 4.28e-14 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1366.61  on 1188  degrees of freedom
Residual deviance:  142.48  on 1176  degrees of freedom
AIC: 168.48

Number of Fisher Scoring iterations: 21
```

Without Location & Eliminating Fatigue

Hide

```
log2 = glm(FluTestStatus ~ Age + Temperature + Gender + MedicalConditions + RunningNose + Cough
+ Myalgia + Headache + ThroatAche + Fever + Vomiting, data = train.data, family = "binomial")
summary(log2)
```

```
Call:
glm(formula = FluTestStatus ~ Age + Temperature + Gender + MedicalConditions + 
    RunningNose + Cough + Myalgia + Headache + ThroatAche + Fever + 
    Vomiting, family = "binomial", data = train.data)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-3.1946  -0.3066   0.1344   0.3380   2.4239  

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -61.69203   15.41344  -4.002 6.27e-05 ***
Age                  0.03316    0.00759   4.369 1.25e-05 ***
Temperature          0.60227    0.15617   3.856 0.000115 ***
Gender1              0.96407    0.22053   4.372 1.23e-05 ***
MedicalConditions1  -1.09093    0.23321  -4.678 2.90e-06 ***
RunningNose1        -0.83635    0.31505  -2.655 0.007938 ** 
Cough1               0.95348    0.31474   3.029 0.002450 ** 
Myalgia1             2.35373    0.34819   6.760 1.38e-11 ***
Headache1           -0.40630    0.33289  -1.221 0.222267    
ThroatAche1          0.14159    0.28089   0.504 0.614201    
Fever1               0.45457    0.27439   1.657 0.097587 .  
Vomiting1            2.68863    0.30970   8.681  < 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: 1366.6  on 1188  degrees of freedom
Residual deviance:  613.6  on 1177  degrees of freedom
AIC: 637.6

Number of Fisher Scoring iterations: 7
```

Without Location & Eliminating ThroatAche

Hide

```
log3 = glm(FluTestStatus ~ Age + Temperature + Gender + MedicalConditions + RunningNos
e + Cough
+ Myalgia + Headache + Fever + Vomiting, data = train.data, family = "binomial")
summary(log3)
```

```
Call:
glm(formula = FluTestStatus ~ Age + Temperature + Gender + MedicalConditions + 
    RunningNose + Cough + Myalgia + Headache + Fever + Vomiting, 
    family = "binomial", data = train.data)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-3.1828  -0.3068   0.1366   0.3379   2.4523  

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -61.560435  15.390018  -4.000 6.33e-05 ***
Age                  0.033364   0.007598   4.391 1.13e-05 ***
Temperature          0.600956   0.155936   3.854 0.000116 ***
Gender1              0.956071   0.219735   4.351 1.36e-05 ***
MedicalConditions1  -1.093518   0.233392  -4.685 2.80e-06 ***
RunningNose1        -0.818098   0.312486  -2.618 0.008844 ** 
Cough1               0.977508   0.311441   3.139 0.001697 ** 
Myalgia1             2.385625   0.343208   6.951 3.63e-12 ***
Headache1           -0.383123   0.329716  -1.162 0.245243    
Fever1               0.471558   0.272177   1.733 0.083177 .  
Vomiting1            2.680690   0.309024   8.675  < 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: 1366.61  on 1188  degrees of freedom
Residual deviance:  613.85  on 1178  degrees of freedom
AIC: 635.85

Number of Fisher Scoring iterations: 7
```

Without Location & Eliminating Headache

We will stop here as we can see that all variables are significant

Hide

```
log4 = glm(FluTestStatus ~ Age + Temperature + Gender + MedicalConditions + RunningNos
e + Cough
+ Myalgia + Fever + Vomiting, data = train.data, family = "binomial")
summary(log4)
```

```

Call:
glm(formula = FluTestStatus ~ Age + Temperature + Gender + MedicalConditions + 
    RunningNose + Cough + Myalgia + Fever + Vomiting, family = "binomial", 
    data = train.data)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-3.2031  -0.3065   0.1343   0.3364   2.3649  

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -60.557030  15.295452  -3.959 7.52e-05 ***
Age                  0.033448   0.007587   4.409 1.04e-05 ***
Temperature          0.590636   0.154965   3.811 0.000138 ***
Gender1              0.958592   0.219570   4.366 1.27e-05 ***
MedicalConditions1  -1.082126   0.233157  -4.641 3.46e-06 ***
RunningNose1        -0.847289   0.311596  -2.719 0.006544 ** 
Cough1               0.928333   0.307554   3.018 0.002541 ** 
Myalgia1             2.150692   0.270987   7.937 2.08e-15 ***
Fever1               0.506056   0.270116   1.873 0.061002 .  
Vomiting1            2.657752   0.309220   8.595  < 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: 1366.61  on 1188  degrees of freedom
Residual deviance:  615.26  on 1179  degrees of freedom
AIC: 635.26

Number of Fisher Scoring iterations: 7
```

Calculating McFadden’s R square value

Hide

```
nullmodel=glm(FluDB$FluTestStatus~1,family="binomial")
1-logLik(log1)/logLik(nullmodel)
```

```
'log Lik.' 0.9257125 (df=13)
```

Hide

```
1-logLik(log2)/logLik(nullmodel)
```

```
'log Lik.' 0.6800712 (df=12)
```

Hide

```
1-logLik(log3)/logLik(nullmodel)
```

```
'log Lik.' 0.6799397 (df=11)
```

Hide

```
1-logLik(log4)/logLik(nullmodel)
```

```
'log Lik.' 0.6792069 (df=10)
```

Model with highest Mc Fadden value is log1 which was the second model build with all variables as significant predictor variables except for location. And it was also the model with lowest AIC. Hence, as per both AIC and Mc Fadden we got model "log1" as the most fitted model ***

Let's check multicollinearity for best model

Hide

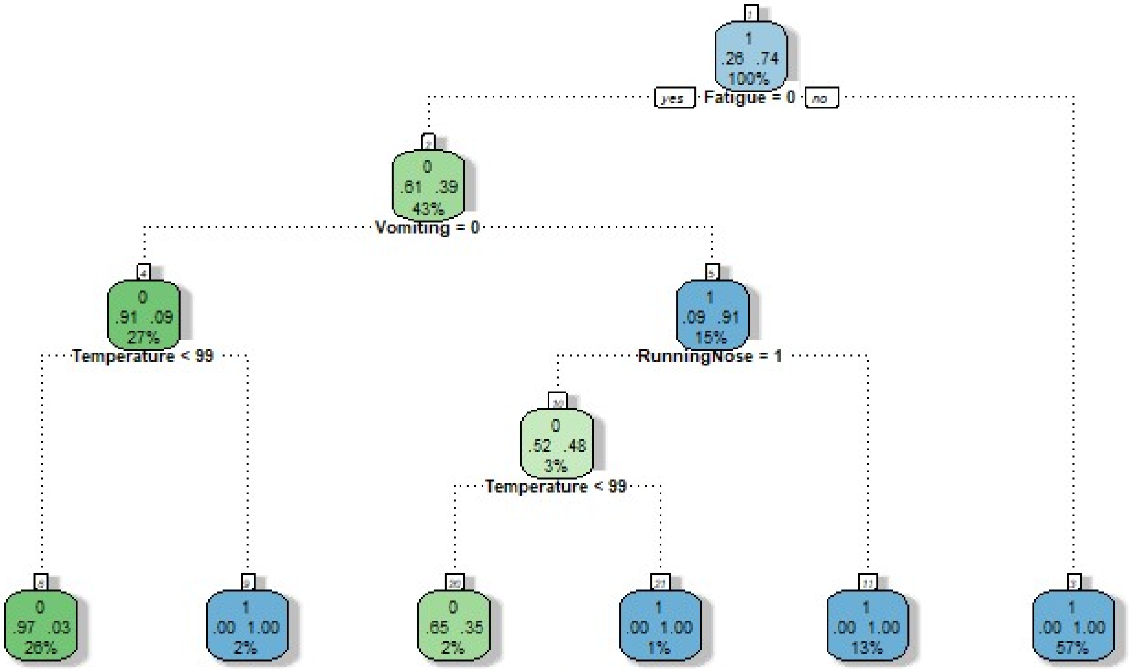
```
vif(log1)
```

```
              Age       Temperature            Gender MedicalConditions       RunningN
ose             Cough           Myalgia          Headache 
         1.548237          1.848781          1.173977          1.592004          3.239
073          2.431516          2.784109          2.321686 
       ThroatAche             Fever           Fatigue          Vomiting 
         1.978134          1.684164          1.000001          1.911019 
```

Decision Tree

Hide

```
model10 = rpart(FluTestStatus ~ Age + Temperature + Gender + MedicalConditions + Runni
ngNose + Cough+ Myalgia + Headache + ThroatAche + Fever + Fatigue + Vomiting, data = t
rain.data, method = "class")
fancyRpartPlot(model10, cex=.58)
```



Rattle 2018-May-26 23:37:17 Supriya Khadake

Constructing confusion matrix and checking accuracy of the model

Hide

```
# Make predictions on the testing set -- Model10
my_prediction10 <- predict(model10, test.data, type = "vector")
# Finish the data.frame() call
my_solution10 <- data.frame(ID = test.data$HostIdentifier, flu10 = my_prediction10)
#Generation of Confusion Matrix
conf10 = table(test.data$FluTestStatus, my_solution10$flu10)
conf10
```

```
      1   2
  0 117   0
  1   9 384
```

Hide

```
acc10 = sum(diag(conf10))/sum(conf10)
acc10
```

```
[1] 0.9823529
```

ROC Curve

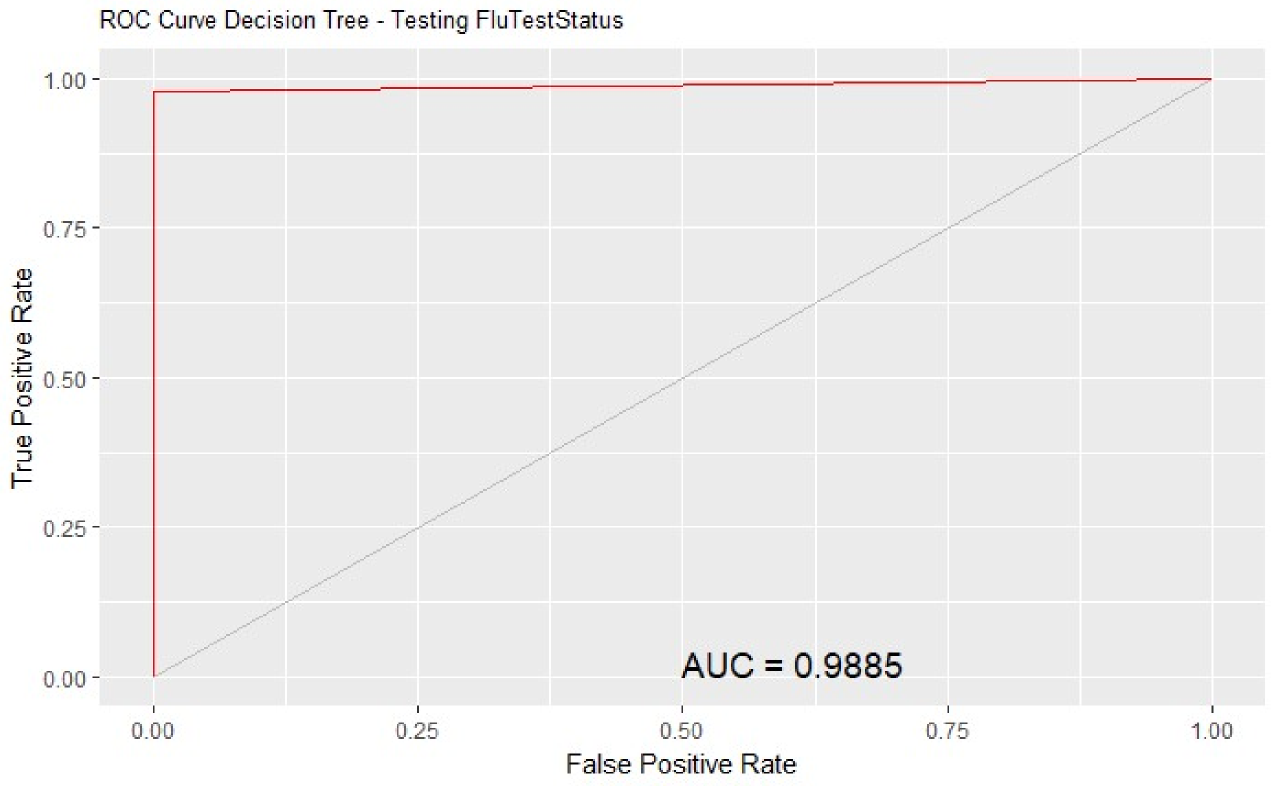
Hide

```
pred <- prediction(my_prediction10, test.data$FluTestStatus)
performance(pred, "auc")
```

```
An object of class "performance"
Slot "x.name":
[1] "None"

Slot "y.name":
[1] "Area under the ROC curve"

Slot "alpha.name":
[1] "none"

Slot "x.values":
list()

Slot "y.values":
[[1]]
[1] 0.9885496


Slot "alpha.values":
list()
```

Hide

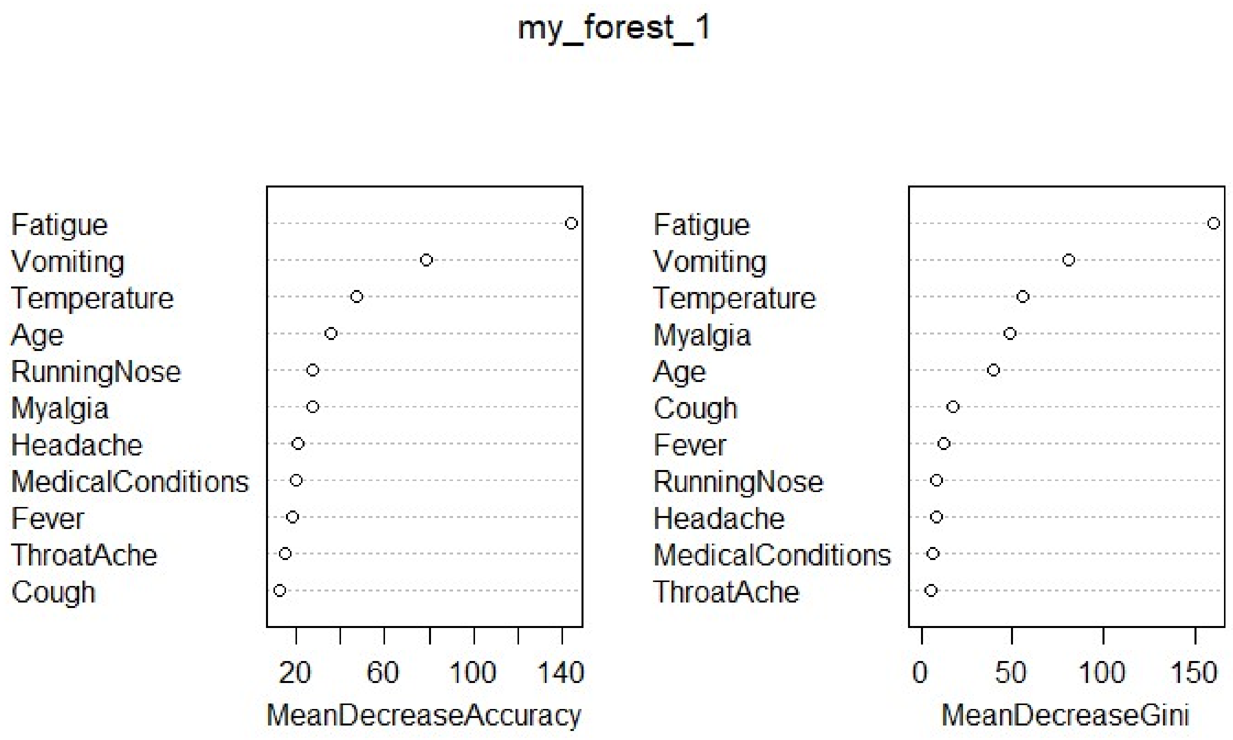
```
pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Decision Tree - Testing FluTestStatus")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                  label=paste("AUC =", round(au, 4)))
print(p)
```



Random Forest

Hide

```
my_forest_1 <- randomForest(FluTestStatus ~ Age + Temperature + MedicalConditions + RunningNose + Cough
+ Myalgia + Headache + ThroatAche + Fever + Fatigue + Vomiting, train.data, ntree=1000, importance=TRUE)
varImpPlot(my_forest_1)
```



Make predictions on the testing set - my_forest_1 without location

Hide

```
my_prediction_1 <- predict(my_forest_1, test.data)
# Make predictions on the testing set -- my_forest_1
my_solution_1 <- data.frame(ID = test.data$HostIdentifier, forest1 = my_prediction_1)
#Generation of Confusion Matrix
conf_1 <- table(test.data$FluTestStatus,my_solution_1$forest1)
conf_1
```

```
      0   1
  0 117   0
  1   2 391
```

Hide

```
acc_1 = sum(diag(conf_1))/sum(conf_1)
acc_1
```

```
[1] 0.9960784
```

KNN Classifier

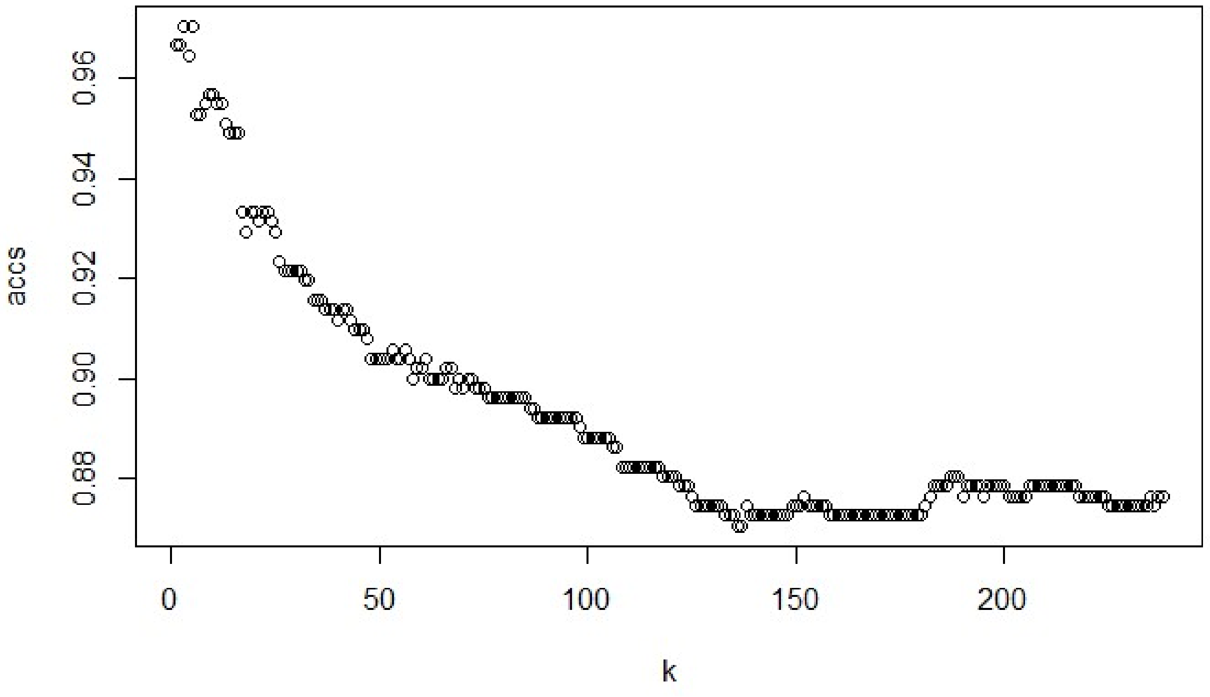
Hide

```
train_label=train.data$FluTestStatus
test_label=test.data$FluTestStatus
# Determing best K-value using accuracy
range=1:round(0.2* nrow(knn_train))
accs= rep(0, length(range))
for(k in range) {knn_pred = knn(knn_train, knn_test, cl=train_label, k=k)
knn_conf <- table(test_label, knn_pred)
knn_conf
accs[k] = sum(diag(knn_conf))/sum(knn_conf)}
accs[k]
```

```
[1] 0.8764706
```

Hide

```
plot(range, accs, xlab="k")
```



Hide

```
which.max(accs)
```

```
[1] 3
```

Hide

```
#Copying train and test data to knn_train and knn_test
knn_train <- train.data
knn_test <- test.data
#Dropping FLUTestStatus column for knn_train and knn_test
knn_train$FluTestStatus <- NULL
knn_test$FluTestStatus <- NULL
# Not Considering Location
knn_train$Location <- NULL
knn_test$Location <- NULL
# Not Considering HostIdentifier
knn_train$HostIdentifier <- NULL
knn_test$HostIdentifier <- NULL
#Normalizing Age
min_age <- min(knn_train$Age)
max_age <- max(knn_train$Age)
knn_train$Age <- (knn_train$Age - min_age)/(max_age - min_age)
knn_test$Age <- (knn_test$Age - min_age)/(max_age - min_age)
#Normalizing Temperature
max_temp <- max(knn_train$Temperature)
min_temp <- min(knn_train$Temperature)
knn_train$Temperature <- (knn_train$Temperature - min_temp)/(max_temp - min_temp)
knn_test$Temperature <- (knn_test$Temperature - min_temp)/(max_temp - min_temp)
```

From the above graph we can see that as value of k increases the Accuracy of the model decreases. The Accuracy of the model can be obtained highest at k = 1 to 5

Hence, checing for k = 10 and k = 3 From the below we proved that as value of k increases the accuracy of the model decreases

Hide

```
knn_pred = knn(knn_train, knn_test, train_label, k=10, prob=TRUE)
knn_conf<- table(test_label, knn_pred)
knn_conf
```

```
          knn_pred
test_label   0   1
         0 109   8
         1  15 378
```

Hide

```
sum(diag(knn_conf))/sum(knn_conf)
```

```
[1] 0.954902
```

Hide

```
knn_pred1 = knn(knn_train, knn_test, train_label, k=3, prob=TRUE)
knn_conf1 <- table(test_label, knn_pred1)
knn_conf1
```

```
          knn_pred1
test_label   0   1
         0 112   5
         1  10 383
```

Hide

```
sum(diag(knn_conf1))/sum(knn_conf1)
```

```
[1] 0.9705882
```

Model Definition and Preparation

With respect to Logistic Regression we found that log1 model was the best model. Considering the variables used in log1 model, Decision Tree was built. We were able to obtain the accuracy of the model using Decision Tree as 98.23% and Random Forest as 99.60%. But before we finalize the model, we will check the quality of prediction that is being performed by our model. We will determine the odds of FluTestStatus and for that let us consider our equation that can be built from the model

Now, Considering p=P(Y=1) as probability of Y which is FluTestStatus. Hence we will set the threshold to 0.5 in order to determine the odds of Y happening a) For odd>1 then pr(Y=1) > Pr(Y=0) -> Pr(Y=1) > 0.5 b) For odd<1 then p=pr(Y=1) < Pr(Y=0) -> Pr(Y=1) < 0.5 c) For odd=1 then Pr(Y=1) = Pr(Y=0) -> Pr (Y=1)=0.5 We can do this by considering one row from the test.data

Hide

```
head(test.data)
```

HostIdentifier	Location	A..	Gen…	Temperature	MedicalConditions	Run
<chr>	<fctr>	<dbl>	<fctr>	<dbl>	<fctr>	<fct
F5005	Tennessee	20	0	99.2	0	0
F4018C13	Tennessee	20	1	99.2	0	1
F4031C30	Tennessee	20	0	98.6	0	0

HostIdentifier <chr>	Location <fctr>	A.. <dbl>	Gen… <fctr>	Temperature <dbl>	MedicalConditions <fctr>	Runni <fctr>
23301961	New York	15	1	102.0	0	1
NIGSP_DOA_00055	Massachusetts	59	1	102.6	1	0
F1042	Tennessee	20	0	99.2	0	1

6 rows | 1-9 of 15 columns

Considering row one, we will substitute these values in the equation obtained to determine the output.

Log(odds) = -146.46 + 0.08 * Age + 1.43 * Temperature + 0.99 * Gender1 -1.89 *MedicalConditions1 -3.74* RunningNose1 + 1.96 * Cough1 + 2.85 * Myalgia1 - 2.57 * Headache1 - 0.32 * Throatache1 + 0.9 * Fever1 + 26.59 * Fatigue1 + 4.70 * Vomiting1.

Log(odds)= -2.104

Hide

```
exp(-2.104)/1+exp(-2.104)
```

```
[1] 0.2439352
```

Since log(odds)=0.24 which is less than 1, The FluTestStatus for this scenario should be 0, which is as per the value in the data set. Hence, we can say that the prediction of our model is appropriate.

Depending on the above models, we have decided the Random Forest performed better in terms of accuracy as it was giving the Highest Accuracy among Logistic Regression, Decision Tree & Random Forest.

Model Implementation

FLUTestStatus = -146.46 + 0.08 * Age + 1.43 * Temperature + 0.99 * Gender1 -1.89 * MedicalConditions1 -3.74 * RunningNose1 + 1.96 * Cough1 + 2.85 * Myalgia1 - 2.57 * Headache1 - 0.32 * Throatache1 + 0.9 * Fever1 + 26.59 * Fatigue1 + 4.70 * Vomiting1