Covid Data Analysis

## Load the data

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(ROCR)  
library(rpart)  
library(rpart.plot)  
  
  
data <- read.csv("cov.csv")  
glimpse(data) # take a look on the data

## Rows: 152,657  
## Columns: 9  
## $ test\_date <chr> "2020-11-12", "2020-11-12", "2020-11-12", "2020-11~  
## $ cough <int> 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1,~  
## $ fever <int> 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~  
## $ sore\_throat <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ shortness\_of\_breath <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ head\_ache <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ corona\_result <chr> "negative", "negative", "negative", "negative", "n~  
## $ age\_60\_and\_above <chr> "No", "No", "Yes", "No", "No", "No", "No", "No", "~  
## $ gender <chr> "male", "male", "female", "male", "male", "male", ~

# Initial Questions

Number of people tested positive and negative for covid-19 in this dataset:

data %>%  
 group\_by(corona\_result) %>%  
 summarise(Freq = n())

## # A tibble: 3 x 2  
## corona\_result Freq  
## <chr> <int>  
## 1 negative 146810  
## 2 other 1294  
## 3 positive 4553

Number of tested positive = 4553 and negative = 146810.

To remove “other” from corona\_result and change positive and negative into 0 and 1 respectively:

# Remove other from the result column  
data1 <- subset(data, corona\_result != "other")  
  
# make positive = 1 and negative = 0  
  
data1 <- mutate(data1, res = recode(corona\_result,  
 "negative" = 0,  
 "positive" = 1))  
  
# Remove observations of missing values with observations for age\_60\_and\_above and gender  
  
summary(is.na(data1[,8:9]))

## age\_60\_and\_above gender   
## Mode :logical Mode :logical   
## FALSE:151363 FALSE:151363

There are no NA’s in these two columns. So missing values can occur with blank spaces. So we shall need to remove those blank spaces to make the data prepared

data1 <- subset(data1, age\_60\_and\_above != "" | gender != "")

To split the dataset into 90:10 ratio, we can execute the following code:

set.seed(111)  
library(caTools)  
sample <- floor(0.70 \* nrow(data1))  
train\_indx <- sample(seq\_len(nrow(data1)), size = sample)  
train <- data1[train\_indx, ]  
test <- data1[-train\_indx, ]

# Logistic Regression

Fit a logistic model treating the res as dependent variable and the symptoms as independent variable:

m1 <- glm(as.factor(res) ~ cough + fever + sore\_throat + shortness\_of\_breath + head\_ache + age\_60\_and\_above + gender, family = "binomial", train)

Construct a confusion matrix with the threshold value 0.5 in test set.

pred\_test <- predict(m1, newdata = test, type = "response")  
cm <- table(test$res, as.numeric(pred\_test >= 0.5))  
cm

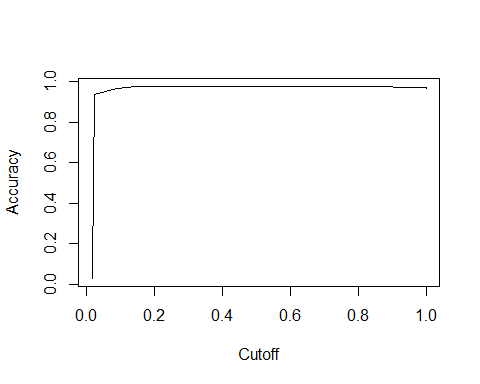
##   
## 0 1  
## 0 43934 43  
## 1 953 402

confusionMatrix(cm)

## Confusion Matrix and Statistics  
##   
##   
## 0 1  
## 0 43934 43  
## 1 953 402  
##   
## Accuracy : 0.978   
## 95% CI : (0.9766, 0.9794)  
## No Information Rate : 0.9902   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.4384   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.9788   
## Specificity : 0.9034   
## Pos Pred Value : 0.9990   
## Neg Pred Value : 0.2967   
## Prevalence : 0.9902   
## Detection Rate : 0.9692   
## Detection Prevalence : 0.9701   
## Balanced Accuracy : 0.9411   
##   
## 'Positive' Class : 0   
##

ROC curve of the logistic regression on test set:

pred <- prediction(pred\_test, test$res)  
perf <- performance(pred, "acc")  
plot(perf)



auc = performance(pred, measure = "auc")  
print(auc@y.values)

## [[1]]  
## [1] 0.7443084

The coefficients are:

coef(m1)

## (Intercept) cough fever sore\_throat   
## -3.90795769 1.75869582 1.98736171 4.36224997   
## shortness\_of\_breath head\_ache age\_60\_and\_aboveNo age\_60\_and\_aboveYes   
## 3.23395142 5.83218051 -0.21352482 -0.25760411   
## genderfemale gendermale   
## 0.07828413 0.16680608

exp(cbind(Odds\_Ratio = coef(m1), confint(m1)))

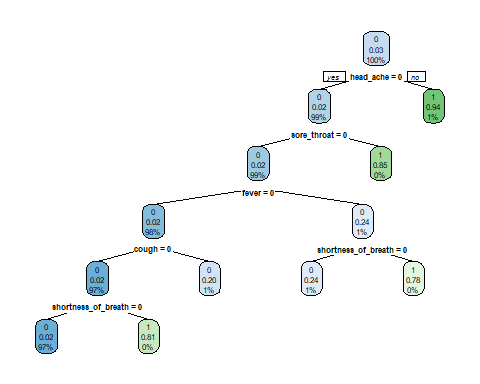
## Waiting for profiling to be done...

## Odds\_Ratio 2.5 % 97.5 %  
## (Intercept) 0.02008147 3.127809e-03 0.06939364  
## cough 5.80486185 4.970603e+00 6.75969632  
## fever 7.29625869 6.203946e+00 8.55680246  
## sore\_throat 78.43340895 5.393383e+01 115.98854209  
## shortness\_of\_breath 25.37974525 1.361695e+01 48.56363307  
## head\_ache 341.10164577 2.524724e+02 471.86044297  
## age\_60\_and\_aboveNo 0.80773212 6.841976e-01 0.96037548  
## age\_60\_and\_aboveYes 0.77290116 6.363066e-01 0.94278791  
## genderfemale 1.08142988 3.166759e-01 6.90578682  
## gendermale 1.18152512 3.460453e-01 7.54437703

# Decision Tree

Train and generate decision tree using rpart and rpart.plot:

m2 <- rpart(res ~ cough + fever + sore\_throat + shortness\_of\_breath + head\_ache + age\_60\_and\_above + gender, train, method = "class",  
 control = rpart.control(cp = 0.001))  
rpart.plot(m2, fallen.leaves = F)



m2$variable.importance

## head\_ache sore\_throat fever cough   
## 1322.97592 255.81934 132.82024 84.49778   
## shortness\_of\_breath   
## 35.86443

We execute the code to find the answers:

pred\_test <- (predict(m2, newdata = test, type = "class"))  
cm <- table(pred\_test, test$res)  
confusionMatrix(cm)

## Confusion Matrix and Statistics  
##   
##   
## pred\_test 0 1  
## 0 43936 950  
## 1 41 405  
##   
## Accuracy : 0.9781   
## 95% CI : (0.9768, 0.9795)  
## No Information Rate : 0.9701   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.4415   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9991   
## Specificity : 0.2989   
## Pos Pred Value : 0.9788   
## Neg Pred Value : 0.9081   
## Prevalence : 0.9701   
## Detection Rate : 0.9692   
## Detection Prevalence : 0.9902   
## Balanced Accuracy : 0.6490   
##   
## 'Positive' Class : 0   
##