COVID-19 data analysis using R

COVID-19 is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The coronavirus created a pandemic which continues to have a major impact on the health and economies of communities across the globe.

Dataset Description: To answer the italicised question we have worked a real dataset, covid_19_dataset.csv, which includes data on approximately 150,000 people in November and December of the year 2020. The fields of the dataset are the following:

• test_date: The date in which the person received the COVID-19 test.

• cough : binary variable which equals 1 if the person has a cough.

• fever : binary variable which equals 1 if the person has a fever.

•sore_throat: binary variable which equals 1 if the person has a sore throat.

• shortness_of_breath: binary variable which equals 1 if the person has stated that they are having shortness of breath.

• corona_result: variable which equals positive if the test came back positive, negative if the test came back negative, and other if the the result was inconclusive.

• age_60_and_above: binary variable which equals No or Yes. • gender: The dataset includes a self-reported value of male or female.

Install basic packages for analysis

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(tidyverse)
```

Read Dataset:

```
library(readr)
dataset <- read_csv("dataset.csv")

##
## -- Column specification --------------------------------------------------
------
## cols(
##   test_date = col_date(format = ""),
##   cough = col_double(),
##   fever = col_double(),
##   sore_throat = col_double(),
##   shortness_of_breath = col_double(),
##   head_ache = col_double(),
##   corona_result = col_character(),
```

```
##   age_60_and_above = col_character(),
##   gender = col_character()
## )
```

(a) How many people in this dataset tested positive for COVID-19? How many tested negative for COVID-19? Offer a possible explanation for the large difference between these numbers.

```
covid <- dataset %>%
  mutate(corona_result = recode(corona_result,
                                "negative" = "0",
                                "positive" = "1",
                                "other" = "2"))
table(dataset$corona_result)

##
## negative    other positive
##   146810     1294     4553
```

(b) In preparation for our analysis, create a new dataset which removes any observations which satisfy corona_result = other. For the remaining observations, convert corona_result into a numeric variable that equals the number 1 if the person tested positive and 0 otherwise. Finally, remove any observations with missing values for age_60_and_above and gender.

```
covid <- dataset %>%
  mutate(corona_result = recode(corona_result,
                                "negative" = "0",
                                "positive" = "1",
                                "other" = "2")) %>%
  filter(corona_result != "2")
table(covid$corona_result)

##
##      0      1
## 146810   4553
```

Finally, remove any observations with missing values for age_60_and_above and gender.

```
library(tidyverse)
df <- dataset %>%
  filter(corona_result != "other", age_60_and_above != "", gender != "")
df <- df[,-1]
df[] <- lapply(df,factor)
```

Remove missing value

```
newdata <- na.omit(covid)
newdata

## # A tibble: 142,305 x 9
##    test_date  cough fever sore_throat shortness_of_br~ head_ache corona_re
sult
##    <date>     <dbl> <dbl>       <dbl>            <dbl>     <dbl> <chr>
##  1 2020-11-12     0     0           0                0         0 0
```

```
##  2 2020-11-12     0     1           0                 0         0 0
##  3 2020-11-12     0     0           0                 0         0 0
##  4 2020-11-12     0     0           0                 0         0 0
##  5 2020-11-12     0     1           0                 0         0 0
##  6 2020-11-12     1     0           0                 0         0 0
##  7 2020-11-12     1     1           0                 0         0 0
##  8 2020-11-12     0     0           0                 0         0 0
##  9 2020-11-12     0     0           0                 0         0 0
## 10 2020-11-12     1     1           0                 0         0 0
## # ... with 142,295 more rows, and 2 more variables: age_60_and_above <chr>
,
## #   gender <chr>
```

```
head(newdata)
```

```
## # A tibble: 6 x 9
##   test_date  cough fever sore_throat shortness_of_br~ head_ache corona_res
ult
##   <date>     <dbl> <dbl>       <dbl>            <dbl>     <dbl> <chr>
## 1 2020-11-12     0     0           0                0         0 0
## 2 2020-11-12     0     1           0                0         0 0
## 3 2020-11-12     0     0           0                0         0 0
## 4 2020-11-12     0     0           0                0         0 0
## 5 2020-11-12     0     1           0                0         0 0
## 6 2020-11-12     1     0           0                0         0 0
## # ... with 2 more variables: age_60_and_above <chr>, gender <chr>
```

(c) Randomly split the data into a train and test set, with approximately 90% of the data in the train set. Make sure that the train and test set preserve the relative ratio of positive to negative cases Hint: Use the sample.split() function from the caTools library.

```
set.seed(123)
library(caTools)
smp_size <- floor(0.90 * nrow(newdata))
train_ind <- sample(seq_len(nrow(newdata)), size = smp_size)
train <- dataset[train_ind, ]
test <- dataset[-train_ind, ]
train
```

```
## # A tibble: 128,074 x 9
##    test_date  cough fever sore_throat shortness_of_br~ head_ache corona_re
sult
##    <date>     <dbl> <dbl>       <dbl>            <dbl>     <dbl> <chr>
##  1 2020-11-02     0     0           0                0         0 negative
##  2 2020-11-03     0     0           0                0         0 negative
##  3 2020-11-04     0     0           0                0         0 negative
##  4 2020-11-03     0     0           0                0         0 negative
##  5 2020-11-09     0     0           0                0         0 negative
##  6 2020-11-08     0     0           0                0         0 negative
##  7 2020-11-09     1     0           0                0         0 negative
##  8 2020-11-01     0     0           0                0         0 negative
```

```
##  9 2020-11-08     0     0           0                 0         0 negative
## 10 2020-11-05     0     1           1                 0         1 positive
## # ... with 128,064 more rows, and 2 more variables: age_60_and_above <chr>
,
## #   gender <chr>
```

```
test
```

```
## # A tibble: 24,583 x 9
##    test_date  cough fever sore_throat shortness_of_br~ head_ache corona_re
sult
##    <date>     <dbl> <dbl>       <dbl>            <dbl>     <dbl> <chr>
##  1 2020-11-12     0     0           0                0         0 negative
##  2 2020-11-12     0     0           0                0         0 negative
##  3 2020-11-12     0     0           0                0         0 negative
##  4 2020-11-12     0     0           0                0         0 negative
##  5 2020-11-12     0     0           0                0         0 negative
##  6 2020-11-12     0     0           0                0         0 negative
##  7 2020-11-12     1     0           0                0         0 negative
##  8 2020-11-12     0     0           0                0         0 negative
##  9 2020-11-12     0     0           0                0         0 negative
## 10 2020-11-12     0     0           0                0         0 negative
## # ... with 24,573 more rows, and 2 more variables: age_60_and_above <chr>,
## #   gender <chr>
```

Logistic Regression: Build a logistic regression model from the training set using the glm() function to predict whether a person is positive for COVID-19.

```
result<-as.numeric(newdata$corona_result)
head(newdata)
```

```
## # A tibble: 6 x 9
##   test_date  cough fever sore_throat shortness_of_br~ head_ache corona_res
ult
##   <date>     <dbl> <dbl>       <dbl>            <dbl>     <dbl> <chr>
## 1 2020-11-12     0     0           0                0         0 0
## 2 2020-11-12     0     1           0                0         0 0
## 3 2020-11-12     0     0           0                0         0 0
## 4 2020-11-12     0     0           0                0         0 0
## 5 2020-11-12     0     1           0                0         0 0
## 6 2020-11-12     1     0           0                0         0 0
## # ... with 2 more variables: age_60_and_above <chr>, gender <chr>
```

```
model <- glm(result~fever+cough+sore_throat +shortness_of_breath+head_ache ,
poisson(),data=newdata)
```

Report the confusion matrix of your logistic regression model on the train set when the threshold is set to 0.5. Compute the accuracy, true positive rate, and false positive rate for the model.

```
anova(model, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: result
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                142304      29914
## fever                1  2605.42    142303      27309 < 2.2e-16 ***
## cough                1  1395.18    142302      25914 < 2.2e-16 ***
## sore_throat          1   723.68    142301      25190 < 2.2e-16 ***
## shortness_of_breath  1    75.98    142300      25114 < 2.2e-16 ***
## head_ache            1  1774.61    142299      23339 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pred<-ifelse(predict(model,type='response')>0.5,1,0)
table(pred)
```

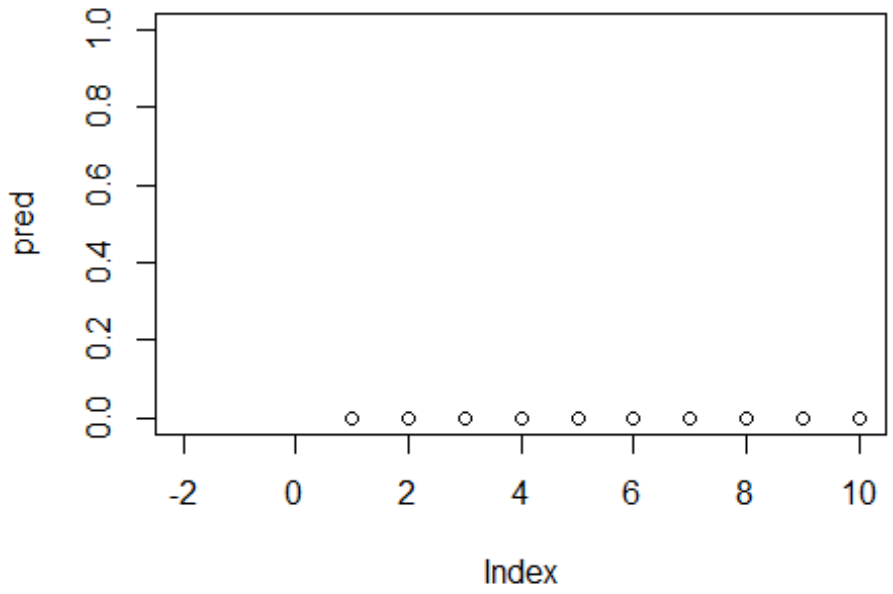
```
## pred
##      0      1
## 141716    589
```

We find 141716 true positive value and 1 false positve value for our model. the logistic regression model is overfitting the data? Yes.modeling error that occurs when a function is too closely fit to a limited set of data points.Besides models the training data too well.This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize

This model would be useful in real life? No.The model is not useful as the class of the dataset is not balanced.So there are imbalaced problem in the output class and thats why our model cannot predict well for new data.So after all we can say that the model would not be useful in real life.

Plot the ROC curve of your logistic regression model on the test set using the ROCR library.

```
## Warning: package 'ROCR' was built under R version 4.0.4
```



Coefficients of your logistic regression model

```
summary(model)
```

```
##
## Call:
## glm(formula = result ~ fever + cough + sore_throat + shortness_of_breath +
##     head_ache, family = poisson(), data = newdata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0587  -0.2063  -0.2063  -0.2063   2.3962
##
## Coefficients:
##                    Estimate Std. Error  z value Pr(>|z|)
## (Intercept)         -3.84969    0.01801 -213.757   <2e-16 ***
## fever                1.05204    0.05090   20.668   <2e-16 ***
## cough                1.09560    0.05223   20.975   <2e-16 ***
## sore_throat          0.51821    0.06161    8.411   <2e-16 ***
## shortness_of_breath  0.19096    0.08967    2.130   0.0332 *
## head_ache            2.53568    0.05413   46.846   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 29914  on 142304  degrees of freedom
## Residual deviance: 23339  on 142299  degrees of freedom
## AIC: 31879
##
## Number of Fisher Scoring iterations: 6
```

Odds of testing positive for COVID-19 First we calculate odds and then we calculted odds ratios with 95% Confidence interval.

```
exp(coef(model))
```

```
##         (Intercept)               fever               cough         sore_t
hroat
##          0.02128642          2.86349727          2.99099096           1.679
01665
## shortness_of_breath           head_ache
##          1.21041045         12.62495653
```

```
exp(cbind(OR = coef(model), confint(model)))
```

```
## Waiting for profiling to be done...
```

```
##                               OR       2.5 %      97.5 %
## (Intercept)          0.02128642  0.02054398  0.02204679
## fever                2.86349727  2.59091224  3.16309301
## cough                2.99099096  2.69917617  3.31245191
## sore_throat          1.67901665  1.48724437  1.89363416
## shortness_of_breath  1.21041045  1.01148691  1.43789955
## head_ache           12.62495653 11.34826141 14.03060807
```

3. Decision Tree:

```
## tibble [142,305 x 7] (S3: tbl_df/tbl/data.frame)
##  $ fever              : Factor w/ 2 levels "0","1": 1 2 1 1 2 1 2 1 1 2 ..
.
##  $ sore_throat        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ shortness_of_breath: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ head_ache          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ corona_result      : Factor w/ 2 levels "negative","positive": 1 1 1 1
1 1 1 1 1 1 ...
##  $ age_60_and_above   : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 1 1 1 1
...
##  $ gender             : Factor w/ 2 levels "female","male": 2 2 1 2 2 2 2
1 2 2 ...
```

```
## Warning: package 'rpart' was built under R version 4.0.4

## Warning: package 'rpart.plot' was built under R version 4.0.4
```

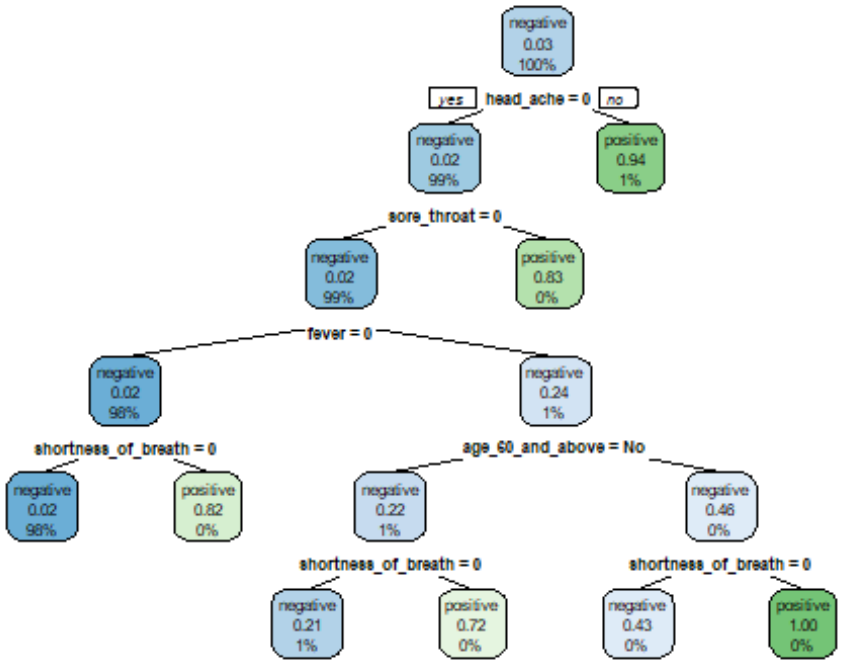


Fig.: Decision Tree Plot

What independent variables does the tree reveal are most important in accurately predicting whether someone has COVID-19? For Finding this answer we calculate variable importance from our decision tree model.

```
vi_tree <- model$variable.importance
vi_tree
```

```
##           head_ache         sore_throat               fever shortness_of_b
reath
##          1811.57086           345.16201           178.03156            79.
67137
##    age_60_and_above
##           18.97304
```

As long as they're small, decision trees are really easy to understand. With depth, the number of terminal nodes rapidly increases. The deeper the tree and the more terminal nodes there are, the more difficult it is to grasp the tree's decision laws. A depth of one indicates the presence of two terminal nodes.

Use 5-fold cross validation:

```
library(caret)

## Warning: package 'caret' was built under R version 4.0.4

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following objects are masked from 'package:Metrics':
##
##     precision, recall

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

```
 model2<- train(
  corona_result ~ .,
  data = newdata,
  method = "rpart",
  trControl = trainControl(method = "cv", number = 5))
```

Concluding Questions:

a) When evaluating models in this assignment, however, we likely found that that the true positive rates in your models was typically quite poor when using a threshold of 0.5. Its due to imbalance output class. We see that maximum class are negative there. That is why TPR is so poor and we cannot use it for future prediction.

b) The model predicted a 0.90 auROC (area under the receiver operating characteristic curve) for the prospective test range, with a 95 percent confidence interval of 0.892–0.905. The potential working points based on predictions from the test set are: 87.30 percent sensitivity and 71.98 percent specificity, or 85.76 percent sensitivity and 79.18 percent specificity. When a COVID-19 diagnosis was compared to sensitivity, the PPV (positive predictive value) was 0.66, with a 95 percent confidence interval of 0.647–0.678. The metrics from all of the ROC curves in this analysis were determined and are available here.

The End