Detecting Covid-19 Clinical Spectrum Using Statistics and Machine Learning

STAT 6309

YICHIEN CHOU
ZHERUI LIN
JINGCHEN LIANG

Introduction

Due to the ongoing COVID-19 pandemic in the world, our lifestyle has changed. Staying at home, stockpiling food, and practicing social distancing has become normal methods to prevent the spread of coronavirus. However, there are still many people who are worried about their conditions if they have a fever, sore throat, or muscle pain. Therefore, in this project, we would like to investigate several common features of COVID-19 based on the data from common physical exams. Then we rely on clinical diagnostic data to make a deeper understanding of why these symptoms happen, what body physiological indices actually have changed, and how have they changed. This research probably helps broaden our horizons so that we may have more efficient ways to prevent this disease.

Fundamental Analysis

- **Background**

From the current United States COVID-19 Statistics, over 1,250,000 Americans are infected by the coronavirus.[1] With the search from recent scientists, they have found several common symptoms of infection with novel coronaviruses such as dry cough, temperature, and fever. Consequently, we would apply diagnostic data collected from clinics, where they provide citizens with nationwide testing for COVID-19.

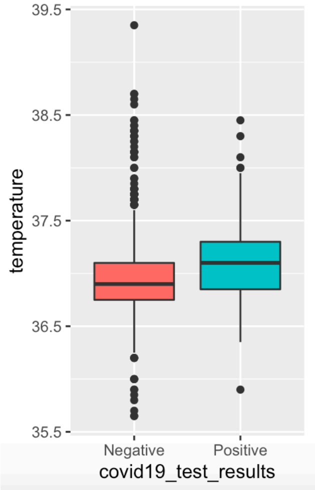
- **Data**

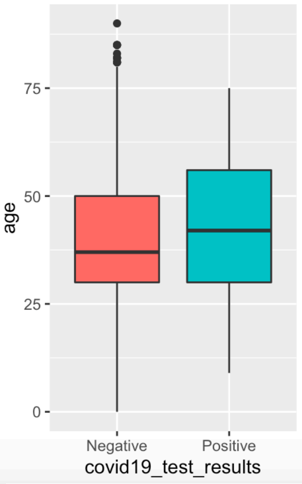
The dataset was sourced from the Carbon Health company website, with three weeks of coronavirus testing. It has a total of 1611 observations and 45 columns. Variables can be generally divided into three specific groups: demography, qualitative physical exams, and quantitative physical exam variables. Below (Table1) there is a preliminary preview of some important variable names.

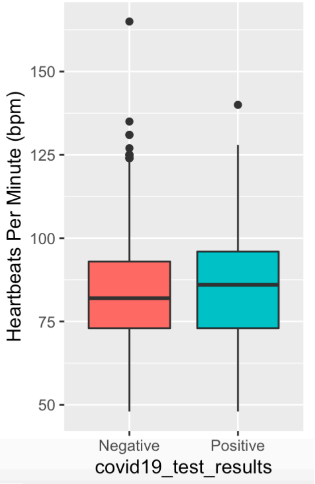
Variables	Description
sys	Systolic blood pressure
dia	Diastolic blood pressure
sats	Oxygen saturation
rr	Respiratory rate measured in breaths per minute
age	Patients age
temperature	Body temperature
cough	Whether patients have cough symptoms. (True & False)
fever	Whether patients have a fever. (True & False)

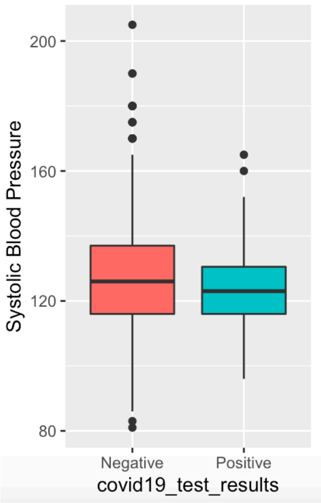
Table 1. Basic descriptions of certain variables in the dataset

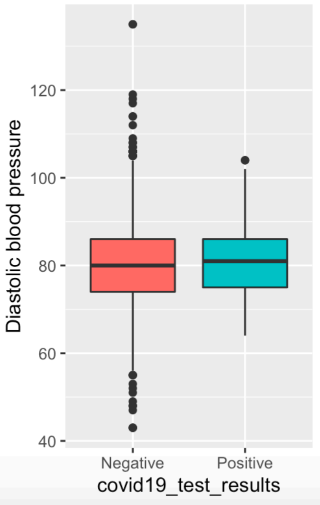
- **Results**

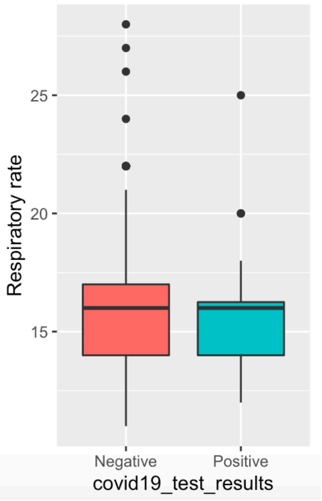

39.5
38.5
37.5
36.5
35.5
temperature
Negative
Positive
covid19_test_results

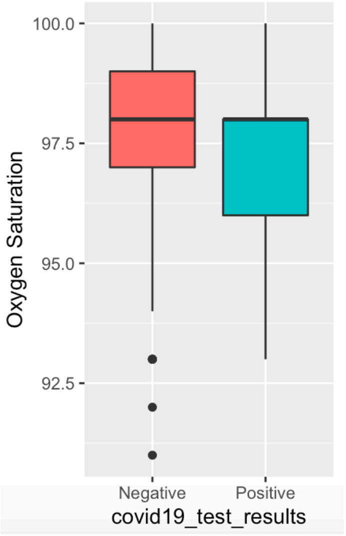

75
50
25
0
age
Negative
Positive
covid19_test_results


150
125
100
75
50
Heartbeats Per Minute (bpm)
Negative
Positive
covid19_test_results


200
160
120
80
Systolic Blood Pressure
Negative
Positive
covid19_test_results


120
100
80
60
40
Diastolic blood pressure
Negative
Positive
covid19_test_results

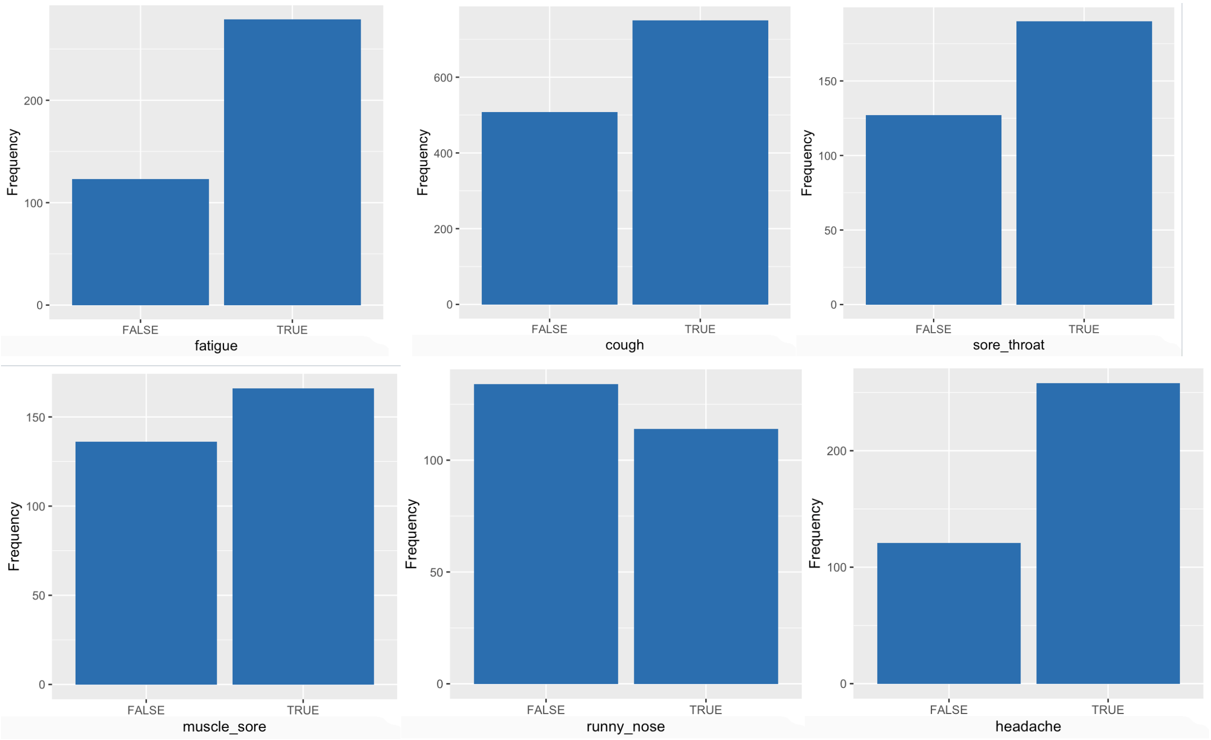

25
20
15
Respiratory rate
Negative
Positive
covid19_test_results


100.0
97.5
95.0
92.5
Oxygen Saturation
Negative
Positive
covid19_test_results

Graph1: Quantitative Variables summary

Comparisons for different kinds of quantitative variables between people who were diagnosed as positive and people who were diagnosed as negative. From the above, several essential findings can be found.

- The body temperature of people diagnosed as positive is significantly higher than that of those diagnosed as people diagnosed as negative.
- More elder people were diagnosed as positive.
- People diagnosed as positive have a higher number of heart beats per minute, leading to increases in heart rate.
- There is no significant evidence to show that these two groups have different systolic blood pressure and diastolic blood pressure, and respiratory rate.
- The median of the oxygen saturation for people diagnosed as positive is roughly the same as the median of the oxygen saturation for people diagnosed as negative. However, people with negative test results have lower overall oxygen saturation.



Graph2: Quantitative Variables summary

The above plots provide a summary of several categorical variables. Here, we only investigate people who were diagnosed as positive.

- More than half of them had fatigue, cough, sore throat, muscle pain, and headache.
- Less than half of them had runny noses.

Therefore, based on the above basic investigations, we have many consistent findings the same as the results from several authorities. According to the WHO, nearly 88% of 55,924 cases reported a fever, followed by 68% of cases that had a dry cough, and 38% of cases that reported fatigue. [2]

However, there are several important points that we need to notice. On the one hand, these data only contain a few positive cases, which may not be a good sample that results could not be generalized to a larger population. On the other hand, there are too many missing values related to some important variables so that we are missing some significant features. Moreover, it is still possible not all COVID-19 cases will get a fever, nor will everyone with a fever test positive for COVID-19. However, it is worth finding a general pattern that cough, fever, temperature, age, and headache are essential indicators to evaluate the symptoms of COVID-19.

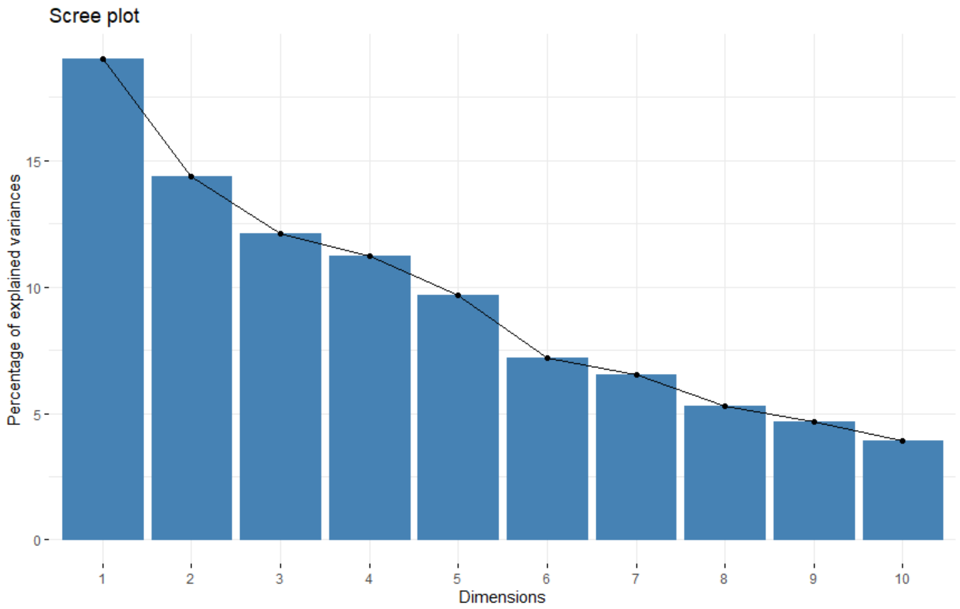
Technical Analysis

- **Data**

The data we used can be found on https://www.kaggle.com/einsteindata4u/covi. The original data has 5644 rows and 111 columns. However, there were many missing values. The first step of the data cleaning was to drop all the variables which contain more than 90% missing values. After that process, 38 columns were left in the data.

Besides the outcome variable, 23 out of 37 the variables are factor variable, and 14 out of 37 are numeric variables. Given the significant amount of missing values, for the numerical variables, we impute them with median values. For the factor variable, we set the missing value to "unknown".

After finishing the data cleaning, we decided to conduct a PCA with the dataset. Based on the result we get from the PCA, 7 principal components explained 80% of the numeric variables. I extract the 7 components and add them back to the dataset.



Graph3: Principal Components Analysis Summary

Before running any machine learning models. We realized that the outcome variable of the data was unbalanced. Only nearly 10% percent of the cases were tested positive.Thus, we decided to use the smote package to resample the data. As a result, the percent of positive cases reach 42.9%.

The last step before running any models is to split the data. I split 80% percent data to the training set and 20% to the test set. Since the PCA was conducted. I subset a training set with all the 7 principal components. Then we have a training set with PCA and a set without PCA. Same as the test set.

- **Methods**

We applied logistic regression and 4 machine learning techniques to build models for predicting.

1. **Logistic Regression**

The first model was built by logistic regression. The response variable was *SARS.Cov.2.exam.result,* whether or not that the patients were confirmed to be infected with COVID-19.

First, we built a model. Then, we checked the assumptions:

i. Multicollinearity

We used the variance inflation factor (VIF) to check if multicollinearity exists between predictors. The values of VIF larger than 4 indicated the multicollinearity.

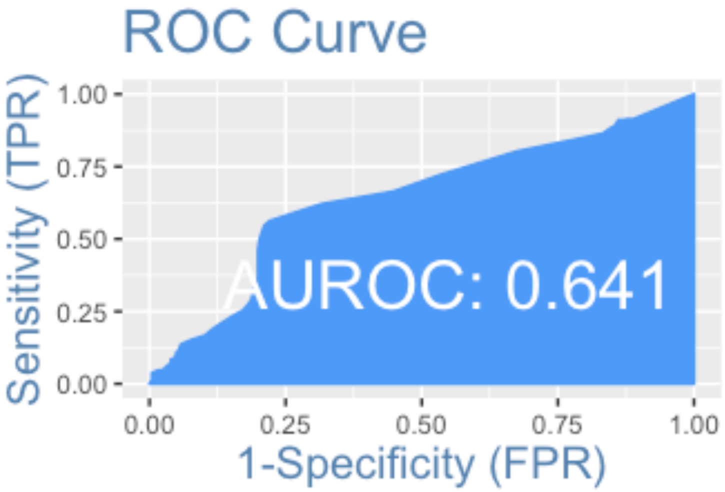
Variable Name	VIF
Hematocrit	285.72
Hemoglobin	199.43
Red.blood.Cells	38.69
Mean.corpuscular.hemoglobin.concentration..MCHC.	38.65
Mean.corpuscular.hemoglobin..MCH.	60.91
Mean.corpuscular.volume..MCV.	90.63

Table 2. Variance Inflation Factor Summary

After removing the variables above (except Hematocrit), the values of VIF were all less than 4.

ii. Independence

In this case, we assumed the assumption satisfied. Next, we wanted to see if the model fitted well.



Graph 4: ROC curve

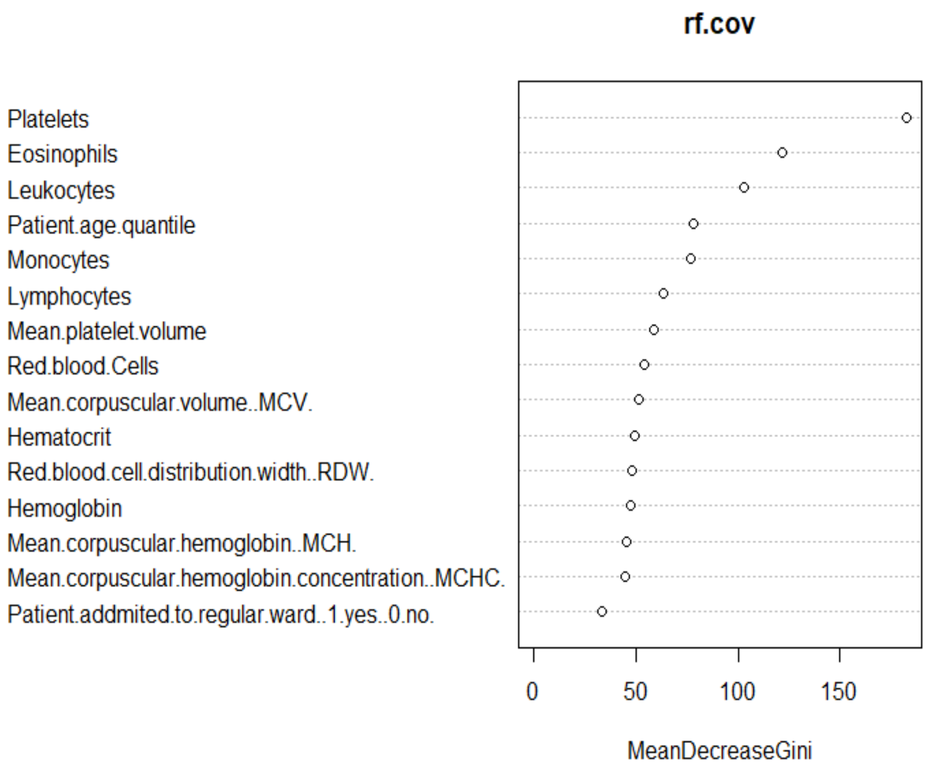
From the ROC curve, we can see it did not fit well. The shape is not ideal. Finally, we find the test accuracy is 43.38%. In sum, the logistic regression model seems not to perform well in predicting in this case.

2. Random Forest

The second model we tested was a random forest model. Based on the results, we build a confusion matrix with test accuracy 86.67%. We run the same test on the data without PCA, the accuracy is 86.026%

Result	Positive	Negative
Positive	441	102
Negative	2	235

Table 3. Confusion Matrix for the random forest



Graph 5: Variable Importance

3. SVM

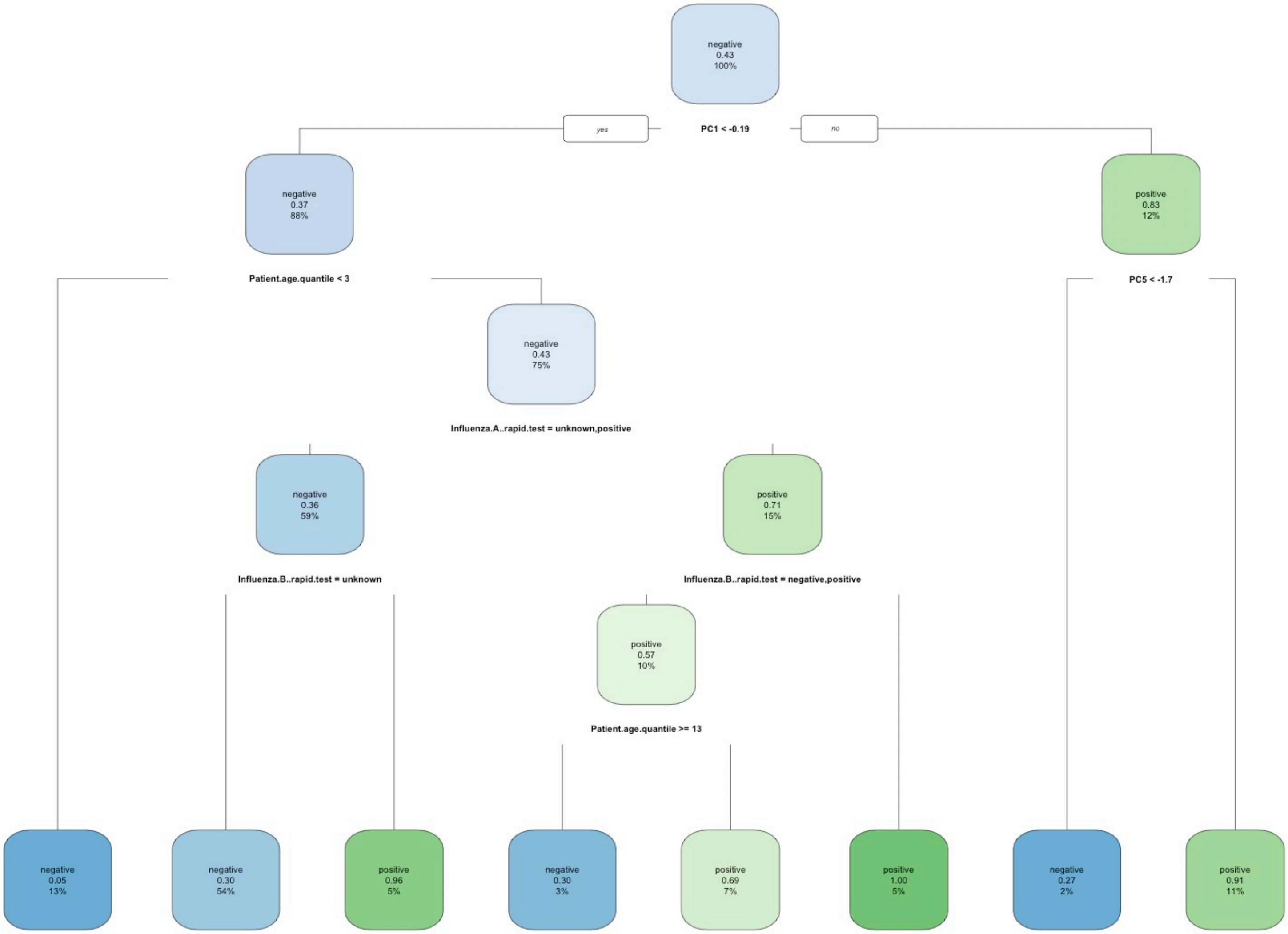
The confusion matrix of SVM is below:

	negative	positive
negative	425	120
positive	6	204

Table 4. Confusion Matrix for SVM

The test accuracy is 83.31%. The result is better than the former two methods.

4. CART



Graph 6: Cart Summary

The confusion matrix of CART is below:

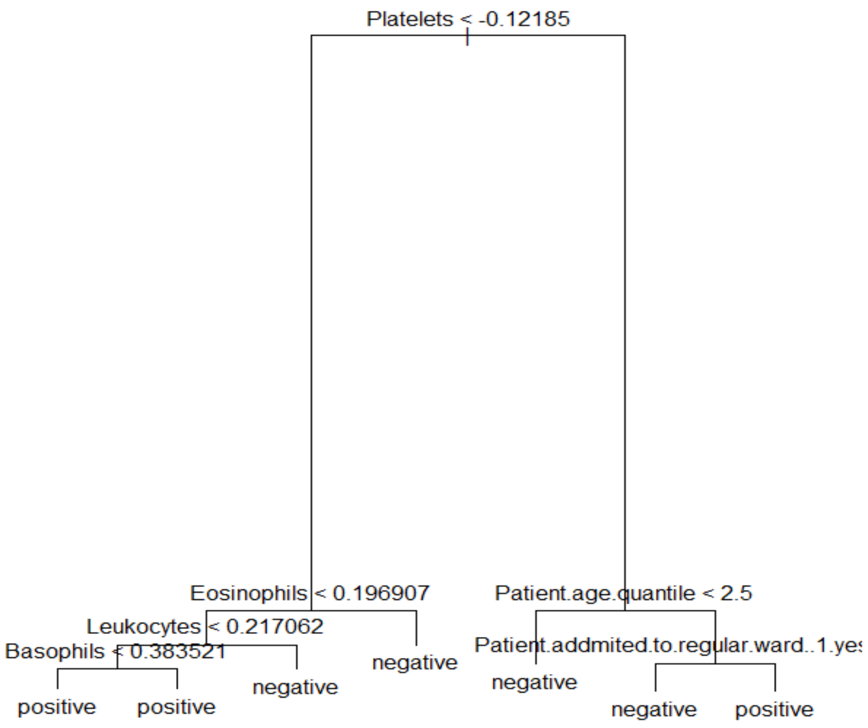
	negative	positive
negative	26	23
positive	405	301

Table 5. Confusion Matrix for CART

The test accuracy is 43.31%. The result is quite similar to the logistic regression model.

5. Decision Tree

The decision tree model has accuracy without PCA 83.974% and 78.590%. The variables that are picked up as nodes are the same as the previous models. Platelets, Eosinophils, leukocytes, and patient age quantile are the top 4 variables to determine if a patient has coronavirus.



Graph 7: Decision Tree Graph

- **Results**

Models	Accuracy (no PCA)	Accuracy (PCA)
Random Forest	0.86667	0.86026
Decision Trees	0.83974	0.78590
SVM	/	0.83311
CART	/	0.4331
Logistic Regression	/	0.4339

Table 6. Test accuracy summary

Thus, we choose a random forest without PCA as our final model. Based on 101 variables from 5644 records kindly provided by Hospital Israelita Albert Einstein, we attempt to predict if a patient was infected by COVID-19 or not. After dealing with missing data, conducting oversampling with SMOTE, and conducting PCA with a lot of models, the best result came from Random Forest, with 86.7% Accuracy and 82% AUC.

Low values for Leukocytes, Platelets, and Eosinophils are a strong indicator of COVID-19 presence. Age is another factor that may influence COVID-19 testing. High values of Monocytes are a strong indicator of COVID-19 presence.

Reference

[1] https://covidusa.net/

[2]https://www.health.com/condition/infectious-diseases/coronavirus/what-temperature-is-considered-a-fever-in-adults

[3]https://www.heart.org/en/coronavirus/coronavirus-covid-19-resources/keeping-a-lid-on-blood-pressure-during-the-coronavirus-crisis

Appendix I

Several important variable explanations are shown below:

Variable Names	Description
Platelets	tiny blood cells that help your body form clots to stop bleeding, thereby initiating a blood clot
Leukocytes	the eosinophil is a type of white blood cell, a specialized cell of the immune system
Eosinophils	a white blood cell containing granules that are readily stained by eosin
Patient.age.quantile	patient age
Monocytes	a type of leukocyte, or white blood cell, a part of the vertebrate innate immune system monocytes also influences the process of adaptive immunity.
Mean.platelet.volume	a machine-calculated measurement of the average size of platelets found in blood and is typically included in blood tests as part of the CBC
Red.blood.Cells	# of red blood cells
Mean.corpuscular.volume..MCV	is a measure of the average volume of a red blood corpuscle.
Hematocrit	volume percentage (vol%) of red blood cells (RBC) in blood,
Lymphocytes	subtypes of a white blood cell in a vertebrate's immune system
Mean corpuscular hemoglobin (MCH)	The average mass of hemoglobin (Hb) per red blood cell (RBC) in a sample of blood
Hemoglobin	iron-containing oxygen-transport metalloprotein in the red blood cells (erythrocytes) of almost all vertebrates

Mean corpuscular hemoglobin concentration? (MCHC)	A measure of the concentration of hemoglobin in a given volume of a packed red blood cell.
Red blood cell distribution width (RDW)	A measure of the range of variation of red blood cell (RBC) volume
Patient admitted to a regular ward (1=yes, 0=no)	Whether the patient received the regular ward
Basophils	a type of white blood cell.

Table 7. variable explanation

Appendix II – R Codes

```
library(VIM)
library(glmnet)
library(DMwR)
library(ggplot2)
library(tidyverse)
library(caTools)
library(OptimalCutpoints)
library(glmnet)
library(caret)
library(tree)
library(randomForest)
library(kernlab)
library(e1071)
library(pROC)
library(rpart)
library(Hmisc)
library(class)
library(corrplot)
library(missForest)


dat1 <- read.csv("Desktop/STAT 6309/Final Project/0407.csv", na.strings = c("",
"NA"))
dat2 <- read.csv("Desktop/STAT 6309/Final Project/0414.csv", na.strings = c("",
"NA"))
dat3 <- read.csv("Desktop/STAT 6309/Final Project/0421.csv", na.strings = c("",
"NA"))
#dat4 <- read_excel("Desktop/Covidclinicaldata/COVID-19 Clinical Data
Repository.xlsx", sheet = 1)

# combine all three datasets
clinical <- rbind(dat1,dat2,dat3)
dim(clinical) #  1611   45

NAcol <- which(colSums(is.na(clinical)) > 0)
missing <- sort(colSums(sapply(clinical[NAcol], is.na)), decreasing = TRUE)

prop.table(table(clinical$covid19_test_results))
```

```
ggplot(data=clinical) +
  geom_bar(mapping = aes(x=factor(covid19_test_results)), colour = "black", fill =
"#4070ad") +
  theme(plot.background = element_blank(),
        panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),
        axis.text.x=element_text(colour="black", size = 10),
        axis.text.y=element_text(colour="black", size = 10),) + xlab("Diagnosis") +
ylab("Frequency")

# clinical1 only positive
clinical1 <- subset(clinical, covid19_test_results=="Positive")
dim(clinical1)

# clinical2 only negative
clinical2 <- subset(clinical, covid19_test_results=="Negative")
dim(clinical2)

hist(clinical1$temperature, breaks=15, col="red")
hist(clinical2$temperature, breaks=15, col="red")
summary(clinical1$temperature)
summary(clinical2$temperature)



NAcol1 <- which(colSums(is.na(clinical1)) > 0)
missing1 <- sort(colSums(sapply(clinical1[NAcol1], is.na)), decreasing = TRUE)


hist(clinical1$temperature, breaks=20, col="red")
ggplot(data=clinical1[!is.na(clinical1$temperature),], aes(x=temperature)) +
  geom_histogram(bins=10,colour = "black", fill = "#4070ad")  + theme(plot.background
= element_blank(),

panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),

axis.text.x=element_text(colour="black", size = 10),

axis.text.y=element_text(colour="black", size = 10),) + xlab("Diagnosis") +
ylab("Frequency")




table(clinical1$covid19_test_results)  ## response variable has three factor levels
clinical$clinical.result <- ifelse(clinical$covid19_test_results == "Negative", 0, 1)
table(clinical$clinical.result) #  1509 negative and 101 positive

# Delete one "Other" observation
other = (clinical$covid19_test_results=="Other")
clinical <- clinical[!other,]
clinical <- clinical[,-5]
```

```
dim(clinical) #  1610   45
# str(clinical)

sum(is.na(clinical))
# 16175 missing values in total

# response variable "clinical.result"
ggplot(clinical, aes(factor(clinical.result))) +
  geom_bar(fill = c("purple","red"))
prop.table(table(clinical$clinical.result))
# Here, only 6.273292% are positive
# This is the unbalanced data - need undersampling and oversampling


dat1 <- read.csv("Desktop/STAT 6309/Final Project/0407.csv", na.strings = c("",
"NA"))
dat2 <- read.csv("Desktop/STAT 6309/Final Project/0414.csv", na.strings = c("",
"NA"))
dat3 <- read.csv("Desktop/STAT 6309/Final Project/0421.csv", na.strings = c("",
"NA"))
#dat4 <- read_excel("Desktop/Covidclinicaldata/COVID-19 Clinical Data
Repository.xlsx", sheet = 1)

# combine all three datasets
clinical <- rbind(dat1,dat2,dat3)
dim(clinical) #  1611   45

NAcol <- which(colSums(is.na(clinical)) > 0)
missing <- sort(colSums(sapply(clinical[NAcol], is.na)), decreasing = TRUE)

prop.table(table(clinical$covid19_test_results))

ggplot(data=clinical) +
  geom_bar(mapping = aes(x=factor(covid19_test_results)), colour = "black", fill =
"#4070ad") +
  theme(plot.background = element_blank(),
        panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),
        axis.text.x=element_text(colour="black", size = 10),
        axis.text.y=element_text(colour="black", size = 10),) + xlab("Diagnosis") +
ylab("Frequency")

# clinical1 only positive
clinical1 <- subset(clinical, covid19_test_results=="Positive")
dim(clinical1)

# clinical2 only negative
clinical2 <- subset(clinical, covid19_test_results=="Negative")
dim(clinical2)

hist(clinical1$temperature, breaks=15, col="red")
hist(clinical2$temperature, breaks=15, col="red")
summary(clinical1$temperature)
summary(clinical2$temperature)
```

```
NAcol1 <- which(colSums(is.na(clinical1)) > 0)
missing1 <- sort(colSums(sapply(clinical1[NAcol1], is.na)), decreasing = TRUE)


hist(clinical1$temperature, breaks=20, col="red")
ggplot(data=clinical1[!is.na(clinical1$temperature),], aes(x=temperature)) +
  geom_histogram(bins=10,colour = "black", fill = "#4070ad")  + theme(plot.background
= element_blank(),

panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),

axis.text.x=element_text(colour="black", size = 10),

axis.text.y=element_text(colour="black", size = 10),) + xlab("Diagnosis") +
ylab("Frequency")




table(clinical1$covid19_test_results)  ## response variable has three factor levels
clinical$clinical.result <- ifelse(clinical$covid19_test_results == "Negative", 0, 1)
table(clinical$clinical.result) #  1509 negative and 101 positive

# Delete one "Other" observation
other = (clinical$covid19_test_results=="Other")
clinical <- clinical[!other,]
clinical <- clinical[,-5]

dim(clinical) #  1610   45
# str(clinical)

sum(is.na(clinical))
# 16175 missing values in total

# response variable "clinical.result"
ggplot(clinical, aes(factor(clinical.result))) +
  geom_bar(fill = c("purple","red"))
prop.table(table(clinical$clinical.result))
# Here, only 6.273292% are positive
# This is the unbalanced data - need undersampling and oversampling

##############
# New graphs #
##############


# combine all three datasets
clinical <- rbind(dat1,dat2,dat3)
dim(clinical) #  1611   45

NAcol <- which(colSums(is.na(clinical)) > 0)
missing <- sort(colSums(sapply(clinical[NAcol], is.na)), decreasing = TRUE)
```

```
prop.table(table(clinical$covid19_test_results))


# clinical1 only positive
clinical.pos <- subset(clinical, covid19_test_results=="Positive")
dim(clinical.pos)

NAcol1 <- which(colSums(is.na(clinical.pos)) > 0)
missing1 <- sort(colSums(sapply(clinical.pos[NAcol1], is.na)), decreasing = TRUE)


# clinical2 only negative
clinical.neg <- subset(clinical, covid19_test_results=="Negative")
dim(clinical.neg)
NAcol2 <- which(colSums(is.na(clinical.neg)) > 0)
missing2 <- sort(colSums(sapply(clinical.neg[NAcol2], is.na)), decreasing = TRUE)


# delete one "other" diagnosis
clinical1 <- subset(clinical, clinical$covid19_test_results != "Other")
dim(clinical1)

#################################
## Compare positive vs negative #
#################################
numericVars <- which(sapply(clinical1, is.numeric)) # index of numeric variables
numericVarNames <- names(numericVars)

###### numeric varaibles
# 1. temperature
#hist(clinical$temperature, breaks=20, col="red")
g1 <- ggplot(data=clinical1[!is.na(clinical1$temperature),],
aes(x=covid19_test_results,y=temperature)) +
  geom_boxplot(fill = "skyblue2")  +
  theme(plot.background = element_blank(),
        panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),
        axis.text.x=element_text(colour="black", size = , angle = 40, hjust=1),
        axis.text.y=element_text(colour="black", size = 6)) + xlab("COVID-19 Test
Results") + ylab("Temperature")
g1

# 2. age
g2 <- ggplot(data=clinical1[!is.na(clinical1$age),],
aes(x=covid19_test_results,y=age)) +
  geom_boxplot(fill = "skyblue2")  +
  theme(plot.background = element_blank(),
        panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),
        axis.text.x=element_text(colour="black", size = , angle = 40, hjust=1),
        axis.text.y=element_text(colour="black", size = 6),) + xlab("COVID-19 Test
Results") + ylab("Age")
g2
```

```
# 3. pulse (Heart rate measured as the number of heartbeats per minute (bpm).)
g3 <- ggplot(data=clinical1[!is.na(clinical1$pulse),],
aes(x=covid19_test_results,y=pulse)) +
  geom_boxplot(fill = "skyblue2")  +
  theme(plot.background = element_blank(),
        panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),
        axis.text.x=element_text(colour="black", size = , angle = 40, hjust=1),
        axis.text.y=element_text(colour="black", size = 6)) + xlab("COVID-19 Test
Results") + ylab("Heart Rate (bpm)")
g3

# 4. sys (Systolic blood pressure measured in units of millimeters of mercury (mmHg))
g4 <- ggplot(data=clinical1[!is.na(clinical1$sys),],
aes(x=covid19_test_results,y=sys)) +
  geom_boxplot(fill = "skyblue2")  +
  theme(plot.background = element_blank(),
        panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),
        axis.text.x=element_text(colour="black", size = , angle = 40, hjust=1),
        axis.text.y=element_text(colour="black", size = 6)) + xlab("COVID-19 Test
Results") + ylab("Systolic Blood Pressure")
g4

# 5. dia (Diastolic blood pressure measured in units of millimeters of mercury
(mmHg))
g5 <- ggplot(data=clinical1[!is.na(clinical1$dia),],
aes(x=covid19_test_results,y=dia)) +
  geom_boxplot(fill = "skyblue2")  +
  theme(plot.background = element_blank(),
        panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),
        axis.text.x=element_text(colour="black", size = , angle = 40, hjust=1),
        axis.text.y=element_text(colour="black", size = 6)) + xlab("COVID-19 Test
Results") + ylab("Diastolic Blood Pressure")
g5

# 6. rr (Respiratory rate measured in breaths per minute)
g6 <- ggplot(data=clinical1[!is.na(clinical1$rr),], aes(x=covid19_test_results,y=rr))
+
  geom_boxplot(fill = "skyblue2")  +
  theme(plot.background = element_blank(),
        panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),
        axis.text.x=element_text(colour="black", size = , angle = 40, hjust=1),
        axis.text.y=element_text(colour="black", size = 6)) + xlab("COVID-19 Test
Results") + ylab("Respiratory Rate")
g6

# 7. sats (Oxygen saturation)
# histgram(clinical$sats)
g7 <- ggplot(data=clinical1[!is.na(clinical1$sats),],
aes(x=covid19_test_results,y=sats)) +
  geom_boxplot(fill = "skyblue2")  +
  theme(plot.background = element_blank(),
        panel.background = element_blank(), axis.line = element_line(size=1, colour =
"black"),
        axis.text.x=element_text(colour="black", size = , angle = 40, hjust=1),
        axis.text.y=element_text(colour="black", size = 6)) + xlab("COVID-19 Test
Results") + ylab("Oxygen Saturation")
g7


tapply(clinical1$sats, clinical1$covid19_test_results, summary)

graph1 <- ggarrange(g1, g2, g3, g4, g5, g6, g7,
                    font.label = list(size = 14, color = "black", face = "bold",
family = NULL),hjust = -1, labels = c("First Visit", "Last Visit"),
                    ncol = 4, nrow = 2)
graph1

####### categorical variables
g8 <- ggplot(data=clinical1[!is.na(clinical1$fatigue),],aes(x=fatigue)) +
  geom_bar(fill = "#4070ad") + theme(plot.background = element_blank(),
                                     panel.background = element_blank(), axis.line =
element_line(size=1, colour = "black"),
                                     axis.text.x=element_text(colour="black", size
= , angle = 40, hjust=1),
                                     axis.text.y=element_text(colour="black", size =
6),) + xlab("Fatigue") + ylab("Frequency")
g8


g9 <-
ggplot(data=clinical1[!is.na(clinical1$rapid_flu_results),],aes(x=rapid_flu_results))
+
  geom_bar(fill = "#4070ad") + theme(plot.background = element_blank(),
                                     panel.background = element_blank(), axis.line =
element_line(size=1, colour = "black"),
                                     axis.text.x=element_text(colour="black", size
= , angle = 40, hjust=1),
                                     axis.text.y=element_text(colour="black", size =
6),) + xlab("Rapid Flu") + ylab("Frequency")
g9


g10 <- ggplot(data=clinical1[!is.na(clinical1$cough),],aes(x=cough)) +
  geom_bar(fill = "#4070ad") + theme(plot.background = element_blank(),
                                     panel.background = element_blank(), axis.line =
element_line(size=1, colour = "black"),
                                     axis.text.x=element_text(colour="black", size
= , angle = 40, hjust=1),
                                     axis.text.y=element_text(colour="black", size =
6),) + xlab("Cough") + ylab("Frequency")
g10

g11 <- ggplot(data=clinical1[!is.na(clinical1$muscle_sore),],aes(x=muscle_sore)) +
  geom_bar(fill = "#4070ad") + theme(plot.background = element_blank(),
                                     panel.background = element_blank(), axis.line =
element_line(size=1, colour = "black"),
                                     axis.text.x=element_text(colour="black", size
= , angle = 40, hjust=1),
                                     axis.text.y=element_text(colour="black", size =
6),) + xlab("Muscle Soreness") + ylab("Frequency")
g11


g12 <- ggplot(data=clinical1[!is.na(clinical1$runny_nose),],aes(x=runny_nose)) +
  geom_bar(fill = "#4070ad") + theme(plot.background = element_blank(),
                                     panel.background = element_blank(), axis.line =
element_line(size=1, colour = "black"),
                                     axis.text.x=element_text(colour="black", size
= , angle = 40, hjust=1),
                                     axis.text.y=element_text(colour="black", size =
6),) + xlab("Runny Nose") + ylab("Frequency")
g12


g13 <- ggplot(data=clinical1[!is.na(clinical1$sob),],aes(x=sob)) +
  geom_bar(fill = "#4070ad") + theme(plot.background = element_blank(),
                                     panel.background = element_blank(), axis.line =
element_line(size=1, colour = "black"),
                                     axis.text.x=element_text(colour="black", size
= , angle = 40, hjust=1),
                                     axis.text.y=element_text(colour="black", size =
6),) + xlab("Shortness of Breath") + ylab("Frequency")
g13


g14 <- ggplot(data=clinical1[!is.na(clinical1$headache),],aes(x=headache)) +
  geom_bar(fill = "#4070ad") + theme(plot.background = element_blank(),
                                     panel.background = element_blank(), axis.line =
element_line(size=1, colour = "black"),
                                     axis.text.x=element_text(colour="black", size
= , angle = 40, hjust=1),
                                     axis.text.y=element_text(colour="black", size =
6),) + xlab("Headache") + ylab("Frequency")
g14


g15 <- ggplot(data=clinical1[!is.na(clinical1$loss_of_taste),],aes(x=loss_of_taste))
+
  geom_bar(fill = "#4070ad") + theme(plot.background = element_blank(),
                                     panel.background = element_blank(), axis.line =
element_line(size=1, colour = "black"),
                                     axis.text.x=element_text(colour="black", size
= , angle = 40, hjust=1),
                                     axis.text.y=element_text(colour="black", size =
6),) + xlab("Loss of Taste") + ylab("Frequency")
g15


################

cov<-read.csv("Desktop/STAT 6309/Final Project/dataset.csv")
```

```
dim(cov)
# The data has 5644 rows and 111 columns.
#check for missing values
# Intend to remove any varibles with more than 5000 missing/Empty values
cov1 <- cov[,-c(40:111)]
# the data after columns 40 does not contained enough information,except Influenza.A
and B rapid test
cov1$Mycoplasma.pneumoniae<-NULL
cov1$Patient.ID<-NULL
cov1$Serum.Glucose<-NULL
cov1$Influenza.A..rapid.test<-cov$Influenza.A..rapid.test
cov1$Influenza.B..rapid.test<-cov$Influenza.B..rapid.test
# Varible contains 5644 missing value
#Left with 38 columns
sapply(cov1, function(x) sum(is.na(x)))
# columns 7-20 has about the  same amount missing value. 5041-5043
# from 20-38, they are all factor varibles
# These varibles are the same categorie, it related to blood.

# impute with mean
set.seed(1)
cov1$Eosinophils <- with(cov1, impute(cov1$Eosinophils, median))
cov1$Monocytes <- with(cov1, impute(cov1$Monocytes, median))
cov1$Mean.platelet.volume <- with(cov1, impute(cov1$Mean.platelet.volume, median))
cov1$Lymphocytes <- with(cov1, impute(cov1$Lymphocytes, median))
cov1$Platelets <- with(cov1, impute(cov1$Platelets, median))
cov1$Leukocytes <- with(cov1, impute(cov1$Leukocytes, median))
cov1$Basophils <- with(cov1, impute(cov1$Basophils, median))
cov1$Red.blood.cell.distribution.width..RDW. <- with(cov1,
impute(cov1$Red.blood.cell.distribution.width..RDW., median))

impute_arg<-
aregImpute(~Hematocrit+Hemoglobin+Red.blood.Cells+Mean.corpuscular.hemoglobin.concent
ration..MCHC.+Mean.corpuscular.hemoglobin..MCH.+Mean.corpuscular.volume..MCV.,data =
cov1,n.impute = 5)
#impute_arg

imputed <-impute.transcan(impute_arg, data=cov1, imputation=1, list.out=TRUE,
pr=FALSE, check=FALSE)
imputed.data <- as.data.frame(do.call(cbind,imputed))
### combine to our orginal data
name <- colnames(imputed.data)
cov2 <- cov1[ , -which(names(cov1) %in% name)]
cov3 <- cbind(cov2,imputed.data)
levels(cov3$Respiratory.Syncytial.Virus)[1] <- "unknown"
levels(cov3$Influenza.A)[1] <- "unknown"
levels(cov3$Influenza.B)[1] <- "unknown"
levels(cov3$Parainfluenza.1)[1] <- "unknown"
levels(cov3$CoronavirusNL63)[1] <- "unknown"
levels(cov3$Rhinovirus.Enterovirus)[1] <- "unknown"
levels(cov3$Coronavirus.HKU1)[1] <- "unknown"
levels(cov3$Parainfluenza.3)[1] <- "unknown"
levels(cov3$Chlamydophila.pneumoniae)[1] <- "unknown"
levels(cov3$Adenovirus)[1] <- "unknown"
levels(cov3$Parainfluenza.4)[1] <- "unknown"
levels(cov3$Coronavirus229E)[1] <- "unknown"
levels(cov3$CoronavirusOC43)[1] <- "unknown"
levels(cov3$Inf.A.H1N1.2009)[1] <- "unknown"
levels(cov3$Bordetella.pertussis)[1] <- "unknown"
levels(cov3$Metapneumovirus)[1] <- "unknown"
levels(cov3$Parainfluenza.2)[1] <- "unknown"
levels(cov3$Influenza.A..rapid.test)[1] <- "unknown"
levels(cov3$Influenza.B..rapid.test)[1] <- "unknown"

# Transfer to factor
cov3$SARS.Cov.2.exam.result<-factor(cov3$SARS.Cov.2.exam.result,levels=c(0,1),labels=
c("negative","positive"))

cov3$Patient.addmited.to.regular.ward..1.yes..0.no.<-
as.factor(cov3$Patient.addmited.to.regular.ward..1.yes..0.no.)
cov3$Patient.addmited.to.semi.intensive.unit..1.yes..0.no.<-
as.factor(cov3$Patient.addmited.to.semi.intensive.unit..1.yes..0.no.)
cov3$Patient.addmited.to.intensive.care.unit..1.yes..0.no.<-
as.factor(cov3$Patient.addmited.to.intensive.care.unit..1.yes..0.no.)
# END OF CLEANING

# EDA
# Respiratory.Syncytial.Virus
ggplot(data = cov3, aes(x = cov3$Respiratory.Syncytial.Virus, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <- cov3 %>% group_by(Respiratory.Syncytial.Virus) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()

ggplot(cov3, aes(x =Respiratory.Syncytial.Virus, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# ALL THE Respiratory.Syncytial.Virus case does not test postive

# Influenza.A
ggplot(data = cov3, aes(x = cov3$Influenza.A, fill = cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Influenza.A) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
```

```
ggplot(cov3, aes(x =Influenza.A, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# ALL THE Influenza.A case does not test postive

# Influenza.B
ggplot(data = cov3, aes(x = cov3$Influenza.B, fill = cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Influenza.B) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Influenza.B, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 3.9% Influenza.B case test postive

# Parainfluenza.1
ggplot(data = cov3, aes(x = cov3$Parainfluenza.1, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Parainfluenza.1) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Parainfluenza.1, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# ALL THE Parainfluenza.1 case does not test postive

# CoronavirusNL63
ggplot(data = cov3, aes(x = cov3$CoronavirusNL63, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(CoronavirusNL63) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =CoronavirusNL63, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 6.7 % CoronavirusNL63 case test postive

# Rhinovirus.Enterovirus
ggplot(data = cov3, aes(x = cov3$Rhinovirus.Enterovirus, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Rhinovirus.Enterovirus) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Rhinovirus.Enterovirus, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 1.6  % Rhinovirus.Enterovirus case test postive

# Coronavirus.HKU1
ggplot(data = cov3, aes(x = cov3$Coronavirus.HKU1, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Coronavirus.HKU1) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Coronavirus.HKU1, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
```

```
# All the Coronavirus.HKU1 case does not test postive

# Parainfluenza.3
ggplot(data = cov3, aes(x = cov3$Parainfluenza.3, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Parainfluenza.3) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Parainfluenza.3, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# All the Parainfluenza.3 case does not test postive

# Chlamydophila.pneumoniae
ggplot(data = cov3, aes(x = cov3$Chlamydophila.pneumoniae, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Chlamydophila.pneumoniae) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Chlamydophila.pneumoniae, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# All the Chlamydophila.pneumoniae case does not test postive

# Adenovirus
ggplot(data = cov3, aes(x = cov3$Adenovirus, fill = cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Adenovirus) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Adenovirus, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# All the Adenovirus case does not test postive

# Parainfluenza.4
ggplot(data = cov3, aes(x = cov3$Parainfluenza.4, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Parainfluenza.4) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Parainfluenza.4, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# All the Parainfluenza.4 case does not test postive

# Coronavirus229E
ggplot(data = cov3, aes(x = cov3$Coronavirus229E, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Coronavirus229E) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Coronavirus229E, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 11.1% Coronavirus229E case test postive

# CoronavirusOC43
ggplot(data = cov3, aes(x = cov3$CoronavirusOC43, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(CoronavirusOC43) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =CoronavirusOC43, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 11.1% CoronavirusOC43 case test postive

# Inf.A.H1N1.2009
ggplot(data = cov3, aes(x = cov3$Inf.A.H1N1.2009, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Inf.A.H1N1.2009) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Inf.A.H1N1.2009, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# All Inf.A.H1N1.2009  case does not test postive

# Bordetella.pertussis
ggplot(data = cov3, aes(x = cov3$Bordetella.pertussis, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Bordetella.pertussis) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Bordetella.pertussis, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# All Bordetella.pertussis case does not test postive

# Metapneumovirus
ggplot(data = cov3, aes(x = cov3$Metapneumovirus, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Metapneumovirus) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Metapneumovirus, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# All Metapneumovirus case does not test postive

# Parainfluenza.2
ggplot(data = cov3, aes(x = cov3$Parainfluenza.2, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Parainfluenza.2) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Parainfluenza.2, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# All Parainfluenza.2 case does not test postive and no case decteted

# Influenza.A..rapid.test
ggplot(data = cov3, aes(x = cov3$Influenza.A..rapid.test, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
```

```
  cov3 %>% group_by(Influenza.A..rapid.test) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Influenza.A..rapid.test, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 8.1 Influenza.A..rapid.test case test postive

# Influenza.B..rapid.test
ggplot(data = cov3, aes(x = cov3$Influenza.B..rapid.test, fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Influenza.B..rapid.test) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Influenza.B..rapid.test, fill = SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 7.8 Influenza.B..rapid.test case test postive

# Patient.addmited.to.regular.ward..1.yes..0.no.
ggplot(data = cov3, aes(x = cov3$Patient.addmited.to.regular.ward..1.yes..0.no., fill
= cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Patient.addmited.to.regular.ward..1.yes..0.no.) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Patient.addmited.to.regular.ward..1.yes..0.no., fill =
SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 45.57 Patient.addmited.to.regular.ward..1.yes..0.no. case test postive
# Patient.addmited.to.semi.intensive.unit..1.yes..0.no.
ggplot(data = cov3, aes(x =
cov3$Patient.addmited.to.semi.intensive.unit..1.yes..0.no., fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Patient.addmited.to.semi.intensive.unit..1.yes..0.no.) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Patient.addmited.to.semi.intensive.unit..1.yes..0.no., fill =
SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 16% Patient.addmited.to.semi.intensive.unit..1.yes..0.no. case test postive

# Patient.addmited.to.intensive.care.unit..1.yes..0.no.
ggplot(data = cov3, aes(x =
cov3$Patient.addmited.to.intensive.care.unit..1.yes..0.no., fill =
cov3$SARS.Cov.2.exam.result)) +
  geom_bar() +
  geom_text(stat = "count", aes(label=..count..),size=4.5,position =
position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette="Paired")

cov3.summary <-
  cov3 %>% group_by(Patient.addmited.to.intensive.care.unit..1.yes..0.no.) %>%
  count(SARS.Cov.2.exam.result) %>%
  mutate(ratio=scales::percent(n/sum(n))) %>%
  ungroup()
ggplot(cov3, aes(x =Patient.addmited.to.intensive.care.unit..1.yes..0.no., fill =
SARS.Cov.2.exam.result)) +
  geom_bar(position = "fill") +
  geom_text(data=cov3.summary, aes(y=n,label=ratio),
            position=position_fill(vjust=0.5)) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Paired")
# 19.5% Patient.addmited.to.intensive.care.unit..1.yes..0.no. case test postive

set.seed(12)
# Use smote for unblance data
cov4<- SMOTE(SARS.Cov.2.exam.result~.,data=cov3)
###PCA
# select all the numeric variable
library(factoextra)
numericVars <- which(sapply(cov4, is.numeric)) # index of numeric variables
all_numVar <- cov4[,numericVars]  # the subset only have numeric variables
pca1 <- prcomp(all_numVar, center=T, scale=T)
```

```
fviz_eig(pca1)
#cumulative percentage of variation
VE <- pca1$sdev^2
PVE <- VE / sum(VE)
round(PVE, 2)
# 7 PCs explained 80% of variation.
# Trainning Test split
pca1$rotation[,1:7]

#Add PCs into data sets
cov4$pc1<-pca1$x[,1]
cov4$pc2<-pca1$x[,2]
cov4$pc3<-pca1$x[,3]
cov4$pc4<-pca1$x[,4]
cov4$pc5<-pca1$x[,5]
cov4$pc6<-pca1$x[,6]
cov4$pc7<-pca1$x[,7]

#split data
set.seed(12)
sample<-sample.split(cov4,SplitRatio = 0.8)
train<-subset(cov4,sample ==TRUE)
trainnopca<-train[,1:38]
trainpca<-train[,-numericVars]
test<-subset(cov4,sample==FALSE)
testnopca<-test[,1:38]
testpca<-test[,-numericVars]

train_x<-as.matrix(trainnopca)[,-2] #without PCA
train_xpca<-as.matrix(trainpca)[,-2] # with PCA
train_y<-as.factor(train$SARS.Cov.2.exam.result)

test_x<-as.matrix(testnopca)[,-2] #without PCA
test_xpca<-as.matrix(testpca)[,-2] #with PCA
test_y<-as.factor(test$SARS.Cov.2.exam.result)

# tree no pca
tree.cov <- tree(SARS.Cov.2.exam.result~., trainnopca)
summary(tree.cov)
plot(tree.cov)
text(tree.cov, pretty=0)
cv.cov<- cv.tree(tree.cov)
plot(cv.cov$size,cv.cov$dev, type='b')
test_prediction <- predict(tree.cov, newdata=testnopca,type="class")
table(test_prediction,test_y)
# (434+221)/780=83.974%
# tree with pca
tree.cov <- tree(SARS.Cov.2.exam.result~., trainpca)
summary(tree.cov)
plot(tree.cov)
text(tree.cov, pretty=0)
cv.cov<- cv.tree(tree.cov)
plot(cv.cov$size,cv.cov$dev, type='b')
test_prediction <- predict(tree.cov, newdata=testpca,type="class")
```

```
table(test_prediction,test_y)
tree.cov$
  # (406+207)=78.590%

  # RF with no pca
  rf.cov<-randomForest(SARS.Cov.2.exam.result~., trainnopca)
pred.rf<-predict(rf.cov, newdata=testnopca)
table(pred.rf,test_y)
# (440+236)/780=86.667%
# RF with PCA
rf.cov<-randomForest(SARS.Cov.2.exam.result~., trainpca)
pred.rf<-predict(rf.cov, newdata=testpca)
table(pred.rf,test_y)
#(439+232)/780=86.026%

# Looks like Pca will not help with model


#####################
##1. logistic
cov.logit <- glm(SARS.Cov.2.exam.result ~. , data = train, family = 'binomial')
summary(cov.logit)

#check collinearity
library(car)
vif(cov.logit)
#Hematocrit, Hemoglobin, Red.blood.Cells, Mean.corpuscular have higher VIF.

#remove ones with lower coefficients and build model again.
cov.logit <- glm(SARS.Cov.2.exam.result ~ . , data = train[,-c(26:30)], family =
'binomial')
summary(cov.logit)

#check collinearity
vif(cov.logit)
#all the VIF were less than 4.

#prediction
predicted <- predict(cov.logit, test, type = "response")

#optimal cutoff
library(InformationValue)
optCutOff <- optimalCutoff(test$SARS.Cov.2.exam.result, predicted)[1]
optCutOff
#0.009903209

#accuracy
SARS.Cov.2.exam.result <- ifelse(test$SARS.Cov.2.exam.result=="negative", 0, 1)
precision(SARS.Cov.2.exam.result, predicted, threshold = optCutOff)
#0.43377

#ROC
plotROC(SARS.Cov.2.exam.result, predicted)

##2. SVM
```

```
library(kernlab)
cov.svm <- ksvm(SARS.Cov.2.exam.result ~. , data = train)
cov.svm.pred.prob <- predict(cov.svm, test, type = "decision")
cov.svm.pred <- predict(cov.svm, test, type = "response")

table.svm <- table(cov.svm.pred, test$SARS.Cov.2.exam.result)
table.svm

accuracy.svm <- (table.svm[1,1]+table.svm[2,2])/sum(table.svm)
accuracy.svm
#0.8331126

##3. Random Forest
library(randomForest) # basic implementation
library(ranger)       # a faster implementation of randomForest
library(caret)        # an aggregator package for performing many machine learning
models
library(h2o)          # an extremely fast java-based platform
library(dplyr)
library(magrittr)

#build model
rf <- randomForest(SARS.Cov.2.exam.result ~. , data = train)
rf
plot(rf)

#find the number to minimize the number of trees
which.min(rf$call)

##4. CART
library(rpart)
library(rpart.plot)

#model
cart <- rpart(SARS.Cov.2.exam.result ~. , data = train, method = "class")
#plot
rpart.plot(cart, extra = 106)
#rule
rpart.rules(cart, cover = TRUE)
#predict
cart.pred <- predict(cart, test, type = "class")
tb.cart <- table(cart.pred, test$SARS.Cov.2.exam.result)
cart.ac <- sum(diag(tb.cart))/ sum(tb.cart)
cart.ac
```