# Identifying the Strongest Indicators of COVID-19


## Abstract:

Covid-19 is among top five leading causes of death worldwide and accounts for thousands of deaths each day. In addition to loss of life, Covid-19 is costly to the healthcare system and the economy. The United States spent approximately over $4 trillion to combat covid-19. For this reason, it is imperative that providers know how to accurately and efficiently screen for Covid-19 so that individuals are treated appropriately based on certain symptoms they experience. In this paper, we showed the trends of Covid-19 cases, deaths, and used machine learning algorithms such as Logistic Regression and Random Forest to identify strong indicators of Covid-19. We assessed the performance based on Kappa metric and came with 91.4% for Logistic Regression model and 93.8% for Random Forest model.


## Introduction:

As a Clinical Analyst at the Health 360 Urgent and Primary Care, there is an influx of Covid-19 patients. We keep track of these patients and build dashboards that highlight trends of Covid-19 cases. Therefore, the motivation of this project is to first visualize the Covid-19 trend and then analyze what are the strongest indicators of Covid-19. The original data of patients and dashboards created during internship cannot be disclosed due to privacy issues. Hence, we will use datasets from Kaggle that highlight the trend of Covid-19 cases and identify strong indicators of Covid-19 patients using couple of machine learning algorithms.

# Data:

### Dataset 1:

day_wise.csv dataset consists of 188 observations and 12 variables. The only variables that were used include Date, New Cases, and New Deaths. These variables were used to create visualizations in Tableau.

### About Dataset 1:

- The dates in this dataset range from January 2020 to July 2020.

- The Covid-19 cases in this dataset are global.

- There are no missing values present.

### Dataset 2:

```
> str(data)
'data.frame':   5434 obs. of  19 variables:
 $ Breathing.Problem                     : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ Fever                                 : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ Dry.Cough                             : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ Sore.throat                           : Factor w/ 2 levels "No","Yes": 2 2 2 1 2 1 1 1 1 1 ...
 $ Running.Nose                          : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 1 1 2 2 1 ...
 $ Asthma                                : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 1 2 1 2 ...
 $ Chronic.Lung.Disease                  : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 1 2 1 2 1 ...
 $ Headache                              : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 1 1 1 1 1 ...
 $ Heart.Disease                         : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 2 2 1 1 1 ...
 $ Diabetes                              : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 1 2 2 2 2 ...
 $ Hyper.Tension                         : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 2 2 2 1 2 ...
 $ Fatigue                               : Factor w/ 2 levels "No","Yes": 2 2 2 1 1 1 2 1 2 2 ...
 $ Gastrointestinal                      : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 1 2 2 1 1 ...
 $ Abroad.travel                         : Factor w/ 2 levels "No","Yes": 1 1 2 2 1 1 1 2 2 1 ...
 $ Contact.with.COVID.Patient            : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 1 1 1 2 1 ...
 $ Attended.Large.Gathering              : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 1 2 1 2 1 ...
 $ Visited.Public.Exposed.Places         : Factor w/ 2 levels "No","Yes": 2 2 1 2 2 1 2 2 1 2 ...
 $ Family.working.in.Public.Exposed.Places: Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 2 1 1 1 ...
 $ COVID.19                              : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
```

Covid Dataset.csv dataset consists of 21 variables and 5434 observations. The dataset includes two more variables *Wearing. Masks* and *Sanitization.from. Market* which are not included above because they were removed since all the values were null.

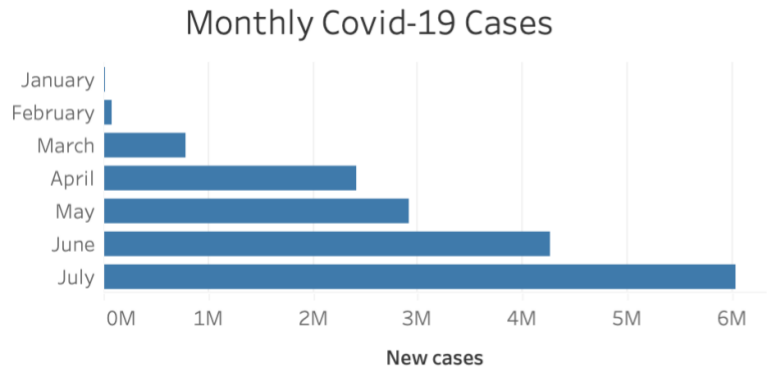
**About Dataset 2:**

We are dealing with imbalanced class data with 4383 observations being Covid Positive and 1051 observations being Covid Negative. There are couple of ways to handle class imbalance problem as listed below:

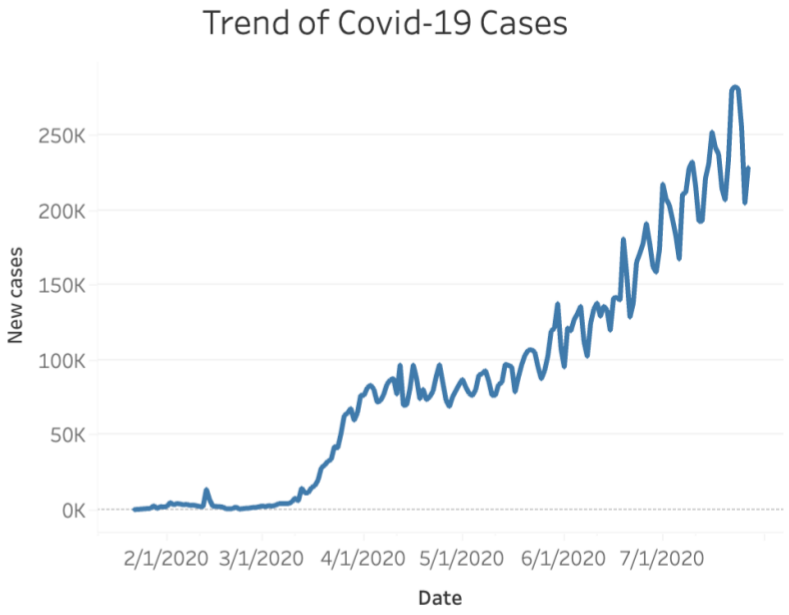
1. Apply a sampling technique: Oversampling, Under sampling, Combination of Oversampling and Under sampling, or SMOTE/ROSE.
2. Use another metric besides accuracy and keep the dataset same.

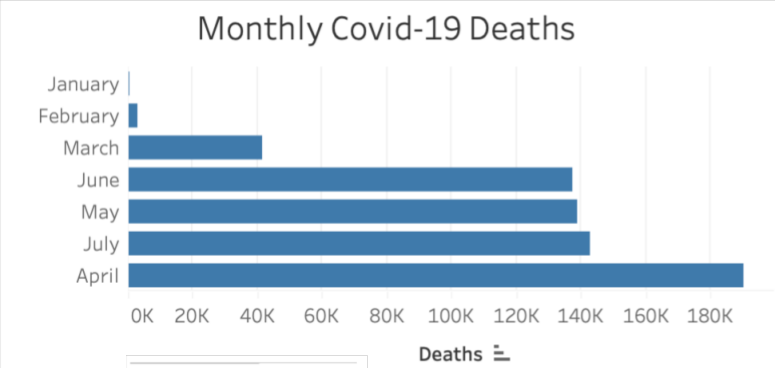
Ideally, we would like both the classes (Covid-19 Positive and Covid-19 Negative) to be the equal, that way when we train our model, it will not be biased toward once class. However, in real-world scenarios it may be hard to attain more data or get rid of data as it will impact the model's accuracy. We could also use SMOTE/ROSE to generate synthetic data, but data generated will have a lot of noise and instances created may be along the same direction, complicating the decision making of machine learning algorithms. Hence, we will use another metric – Cohen's Kappa, when evaluating the performance of our machine learning models. Kappa metric is more useful when it comes to dealing with class imbalance problems. Kappa will always be less than or equal to 1. The closer the Kappa value is to 1, the better the model is.

# Figures and Tables:

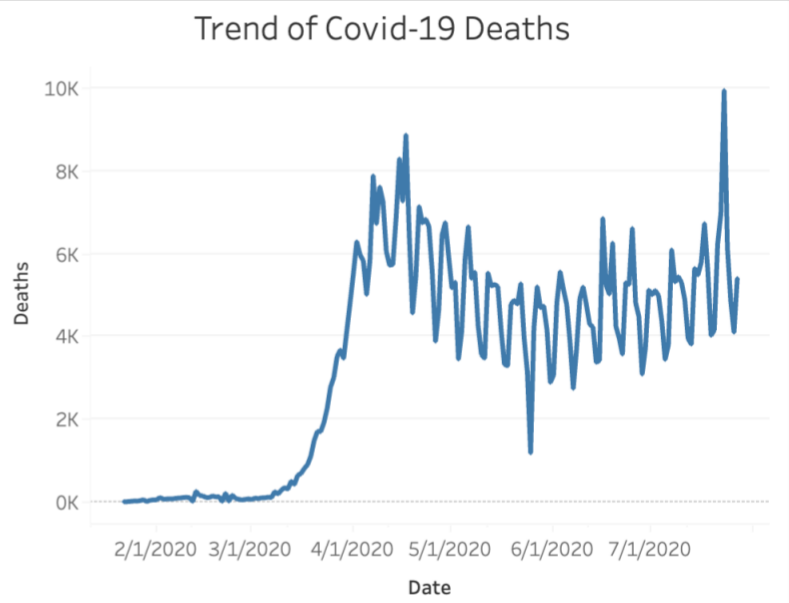



| | |
|---|---|
| January | 9,372 |
| February | 75,379 |
| March | 786,064 |
| April | 2,412,383 |
| May | 2,921,042 |
| June | 4,265,801 |
| July | 6,030,911 |




| | |
|---|---|
| January | 196 |
| February | 2,723 |
| March | 41,542 |
| April | 190,226 |
| May | 138,902 |
| June | 137,604 |
| July | 142,826 |




Note: Cases and deaths in January maybe significantly low compared to other months since data of Covid-19 was just started to be collected.

Due to the huge impact of Covid-19 on many lives, we will use Dataset 2 to make predictions whether a patient has Covid-19 using machine learning and identify strong indicators of Covid-19.

# Method 1: Logistic Regression

Logistic Regression is a machine learning algorithm that is used when the target variable is binary. Logistic regression follows the following equation below

$$p(X) = e^{\beta_0} + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p \,/\, (1 + e^{\beta_0} + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p)$$

where:

- $\mathbf{X_j}$: The $j^{th}$ predictor variable
- $\boldsymbol{\beta_j}$: The coefficient estimate for the $j^{th}$ predictor variable

p(X) represents the probability of X with an output of 0 or 1. In this situation, p(X) represents the probability of a person being a Covid-19 patient or not.

In R, logistic model is created as shown in the steps below.

```
> fit.logistic <- train(COVID.19~ ., data = train, method = "glm",
+                       family='binomial' ,trControl = fitControl,
+                       metric = 'Kappa')
```

## Results:

```
> summary(fit.logistic)

Call:
NULL

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.71186   0.00000   0.00000   0.00182   1.79094

Coefficients:
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                            -10.72695    0.80320 -13.355  < 2e-16 ***
Breathing.ProblemYes                     2.86158    0.27897  10.258  < 2e-16 ***
FeverYes                                 4.98330    0.47189  10.560  < 2e-16 ***
Dry.CoughYes                             4.07901    0.36297  11.238  < 2e-16 ***
Sore.throatYes                           3.95341    0.35012  11.292  < 2e-16 ***
Running.NoseYes                         -1.57557    0.26301  -5.991 2.09e-09 ***
AsthmaYes                               -0.15877    0.25571  -0.621  0.53467
Chronic.Lung.DiseaseYes                  0.08469    0.24010   0.353  0.72429
HeadacheYes                             -0.43536    0.23502  -1.852  0.06396 .
Heart.DiseaseYes                        -0.35623    0.24498  -1.454  0.14590
DiabetesYes                              0.47100    0.22931   2.054  0.03998 *
Hyper.TensionYes                        -0.54519    0.24546  -2.221  0.02634 *
FatigueYes                              -0.19141    0.26533  -0.721  0.47066
GastrointestinalYes                     -0.05889    0.26334  -0.224  0.82305
Abroad.travelYes                        21.93558  477.00856   0.046  0.96332
Contact.with.COVID.PatientYes            1.82255    0.26689   6.829 8.56e-12 ***
Attended.Large.GatheringYes              9.83599    0.86939  11.314  < 2e-16 ***
Visited.Public.Exposed.PlacesYes        -0.76179    0.24489  -3.111  0.00187 **
Family.working.in.Public.Exposed.PlacesYes  1.34012    0.27336   4.902 9.46e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3737.71  on 3804  degrees of freedom
Residual deviance:  603.99  on 3786  degrees of freedom
AIC: 641.99
```

## Training the Model:

Training data is used to create a model, and then test data is used to check the performance of the model. 70 % of our data was put into the train set, and 30% was retained for testing. Note: k-fold cross validation was also applied with k = 10.

The logistic model created from the train set was:

*Covid* = -10.73 + 2.86 * *Breathing Problem* + 4.98 * *Fever* + 4.10 * *Dry Cough* + 3.95 * *Sore Throat* -1.58 * *Running Nose* -0.16 * *Asthma* + 0.085 * *Chronic Lung Disease* - 0.44 * *Headache* - 0.36 * *Heart Disease* + 0.47 * *Diabetes* - 0.55 * *Hyper Tension* - 0.19 * *Fatigue* - 0.059 * *Gastrointestinal* + 21.94 * *Abroad Travel* + 1.82 * *Contact with Covid Patient* + 9.84 * *Attended Large Gathering* - 0.76 * *Visited Public Exposed Places* + 1.34 * *Family working in Public Exposed Places*

## Testing the Model:

To test the model, we ran the reserved test observations through the model to predict whether or not the person was a Covid-19 patient. The output for each observation less than 0.5 is turned to 0 (*Covid Negative, not a Covid Patient*) and every output greater than or equal to 0.5 is turned to 1 (*Covid Positive, Covid Patient*). We then compared each predicted result with the actual results.

```
> pred <- predict(fit.logistic, newdata=test)
> confusionMatrix(table(pred,test[,"COVID.19"]), positive="Yes")
Confusion Matrix and Statistics


pred    No  Yes
  No   294   23
  Yes   21 1291

               Accuracy : 0.973
                 95% CI : (0.9639, 0.9803)
    No Information Rate : 0.8066
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9136

 Mcnemar's Test P-Value : 0.8802

            Sensitivity : 0.9825
            Specificity : 0.9333
         Pos Pred Value : 0.9840
         Neg Pred Value : 0.9274
             Prevalence : 0.8066
         Detection Rate : 0.7925
   Detection Prevalence : 0.8054
      Balanced Accuracy : 0.9579

       'Positive' Class : Yes
```

## Discussion and Conclusion:

As we can see from metrics above, the model is performing really well since the Kappa value is close to one. Also, we were able to predict Covid patients (*Covid Positive)* correctly 98.25% of the time. This is partly expected since we have more Covid Positive patients than Covid Negative patients.

After fitting the logistic regression model, we found that predictors such as having breathing problems, fever, dry cough, sore throat, running nose, diabetes, hypertension, contact with Covid patients, attending large gatherings, visiting exposed places, or having family members that work in exposed places are strong indicators of a patient being likely to have Covid-19.

We have also tried to oversample the minority class and under sample the majority class during the training set but did not notice any significant changes in the performance.

# Method 2: Random Forest

Random Forest is an ensemble machine learning algorithm that develops by aggregating multiple decision trees to construct a prediction model. Prediction for classification tasks is based on the majority vote of the multiple aggregated decision trees whereas prediction for regression tasks is based on the average of multiple aggregated decision trees.

In R, Random Forest model is created using "rf" method as shown in the steps below.

```
> fit.rf <- train(COVID.19~ ., data = train, method = "rf", # Random Forest
+                 trControl = fitControl,
+                 tuneLength = 18, # number of predictors
+                 metric = 'Kappa')
```

## Results:

```
> fit.rf
Random Forest

3805 samples
  18 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 3424, 3424, 3425, 3425, 3424, 3425, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.9789743  0.9319261
   3    0.9834411  0.9472818
   4    0.9834411  0.9472818
   5    0.9839674  0.9490501
   6    0.9834411  0.9472818
   7    0.9839674  0.9490501
   8    0.9839674  0.9490501
   9    0.9839674  0.9490501
  10    0.9839674  0.9490501
  11    0.9834411  0.9472818
  12    0.9837042  0.9481706
  13    0.9839674  0.9490501
  14    0.9839674  0.9490501
  15    0.9839674  0.9490501
  16    0.9839674  0.9490501
  17    0.9839674  0.9490501
  18    0.9839674  0.9490501

Kappa was used to select the optimal model using the largest value.
The final value used for the model was mtry = 5.
```

## Training the Model:

Training was the same as for Logistic Regression - 70 % of our data was used for the train set, and 30% was retained for the test data with 10-fold cross validation

Note:

- New parameter tuneLength = 18 was added, to try 18 different mtry values.
- mtry indicates the number of variables that are randomly sampled as candidates at each split.

After running fitting random forest model on the training data, model with mtry=5 was built.

## Testing the Model:

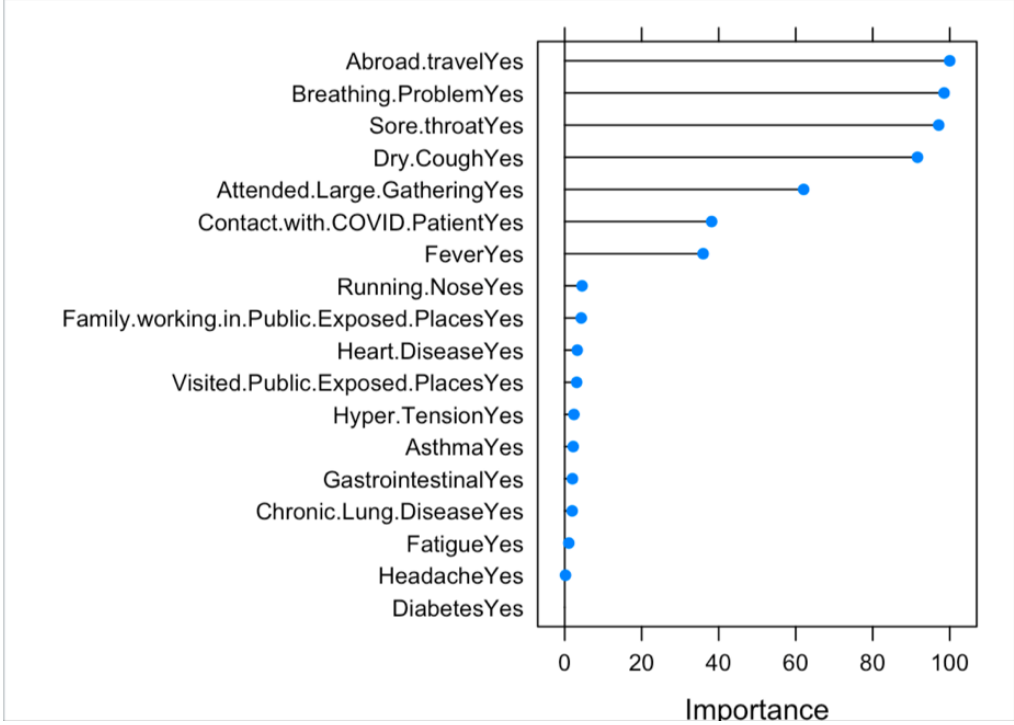
To test the model, we ran the reserved test observations through the model to predict whether or not the person was a Covid patient. The predictors were passed through multiple aggregated decision trees. The highest voted predicted class will be considered as the final prediction.

```
> predRf <- predict(fit.rf,test)
> confusionMatrix(table(test[,"COVID.19"],predRf), positive="Yes")
Confusion Matrix and Statistics

     predRf
        No  Yes
  No   307    8
  Yes   24 1290

               Accuracy : 0.9804
                 95% CI : (0.9724, 0.9865)
    No Information Rate : 0.7968
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9382

 Mcnemar's Test P-Value : 0.00801

            Sensitivity : 0.9938
            Specificity : 0.9275
         Pos Pred Value : 0.9817
         Neg Pred Value : 0.9746
             Prevalence : 0.7968
         Detection Rate : 0.7919
   Detection Prevalence : 0.8066
      Balanced Accuracy : 0.9607

       'Positive' Class : Yes
```

## Discussion and Conclusion:

As we can see from the metrics above, Kappa value slightly improved whilst the specificity value slightly decreased. Meaning that model's ability to predict Covid Negative patients went down. Overall, the model is performance is still good but Logistic Regression Model may be preferred due to its less training time.

We found that traveling abroad, having breathing problem, sore throat, dry cough, fever, and attending large gathers or having contact with Covid patients are the most important features in determining where a patient is going to be Covid Positive or not.



# Summary

Dataset 1 highlights the trends of Covid-19 cases and the number of deaths related to Covid-19. At Health 360 Urgent and Primary Care, Covid-19 test results and symptoms of patients are collected then stored in the eClinicalWorks portal. Also, Covid-19 cases are displayed in a similar way as shown on page 4. From the symptoms recorded in eClinicalWorks portal, we analyze the data and try to identify the most significant indicators of why a person may have a certain disease (Covid-19 in this case). This process was mimicked using Dataset 2 where we were trying to predict Covid-19 presence and identify strong indicators using Machine Learning algorithms like Logistic Regression and Random Forests. Both the models resulted Kappa over 90%. Also, the significant indicators that were found align with the patients in the urgent care.

# References

"Covid Facts." *CDC*, 2 August 2022, https://deathmeters.info/# Accessed 2 August 2022.

hemanthhari. (May 2020). Covid Dataset.csv. Retrieved [Date Retrieved] from https://www.kaggle.com/datasets/hemanthhari/symptoms-and-covid-presence

devakumar. (2020). day_wise.csv. Retrieved [Date Retrieved] from https://www.kaggle.com/datasets/imdevskp/corona-virus-report

"Kappa." 2 August 2022 http://www.pmean.com/definitions/kappa.htm Accessed 2 August 2022.

"Machine Learning Evaluation Metrics in R" 2 August 2022 https://machinelearningmastery.com/machine-learning-evaluation-metrics-in-r/ Accessed 2 August 2022.

# Code

```
library(grid)
library(gridExtra)
library(ggplot2)
library(caret)
library(ROSE)
library(pscl)
#library(InformationValue)
library(Rcpp)
library(tidyverse)
library(MASS)

library(pROC)

#Load Data
data <- read.csv("~/Desktop/UH LIFE/YEAR 4 UH GRAD/Summer 2022/MATH 6315
Internship/Assignment 1/Covid Dataset.csv")
head(data)


#Change to Categorical Variables
cols <- c(colnames(data))
data[cols] <- lapply(data[cols], as.factor)
data$Wearing.Masks <- NULL
data$Sanitization.from.Market <- NULL

# Check Class
barplot(prop.table(table(data$COVID.19)),
        col = rainbow(2),
        ylim = c(0,1),
        main = "Class Distribution") #80% Yes and 20% NO

# EDA
indpendentVars <- data[,-18]
dependent <- data$COVID.19

cnt=0
plot_lst <- vector("list", length = length(indpendentVars))
for (columnName in colnames(indpendentVars)){
  cnt = cnt+1
  plot_lst[[cnt]] <- local({
    cnt <- cnt
    inp <- ggplot(data=indpendentVars, aes(x = unlist(indpendentVars[,cnt]),
fill=dependent)) + xlab(columnName) + ggtitle(columnName) +
geom_bar(stat="count",

position=position_dodge(width = 0.5)) + theme_bw() + theme(axis.title.x =
element_text(size = 15), plot.title = element_text(size = 18)) +
labs(fill="COVID-19")
    plot_lst[[cnt]] <- inp
  })
}
```

```
set_plot_dimensions <- function(width_choice, height_choice) {
  options(repr.plot.width=width_choice, repr.plot.height=height_choice)
}
set_plot_dimensions(20, 10)
factor_plots <- marrangeGrob(plot_lst, nrow = 2, ncol = 3)
factor_plots

# Split the Data into Train and Test

index <- createDataPartition(data$COVID.19, p=0.7,list = FALSE)
train <- data[index,]
test <- data[-index,]

fitControl <- trainControl(method = 'cv', number = 10,
                           savePredictions = 'final')#, repeats = 3)#,
                          #sampling = 'under')


#Logistic Regression
set.seed(123)
fit.logistic <- train(COVID.19~ ., data = train, method = "glm",
                      family='binomial' ,trControl = fitControl,
                      metric = 'Kappa')

exp(coef(fit.logistic$finalModel)) # gives you intercepts

summary(fit.logistic)
pred <- predict(fit.logistic, newdata=test)
confusionMatrix(table(pred,test[,"COVID.19"]), positive="Yes")#, mode =
"prec_recall")

#plot(caret::varImp(fit.logistic))

#Random Forest
set.seed(123)
fit.rf <- train(COVID.19~ ., data = train, method = "rf", # Random Forest
                trControl = fitControl,
                tuneLength = 18, # number of predictors
                metric = 'Kappa')

predRf <- predict(fit.rf,test)
confusionMatrix(table(test[,"COVID.19"],predRf), positive="Yes")

plot(caret::varImp(fit.rf))
```