Covid-19 Death Investigation

wliu16

2022-12-14

```
setwd("~/Desktop/R/64060-003/Final_Exam")  #set working directory
Covid <- read.csv("Covid Data.csv")  #load the data
summary(Covid)
```

```
##       SEX              AGE         CLASIFFICATION_FINAL  PATIENT_TYPE
##  Min.   :1.000   Min.   :  0.00   Min.   :1.000        Min.   :1.000
##  1st Qu.:1.000   1st Qu.: 30.00   1st Qu.:3.000        1st Qu.:1.000
##  Median :1.000   Median : 40.00   Median :6.000        Median :1.000
##  Mean   :1.499   Mean   : 41.79   Mean   :5.306        Mean   :1.191
##  3rd Qu.:2.000   3rd Qu.: 53.00   3rd Qu.:7.000        3rd Qu.:1.000
##  Max.   :2.000   Max.   :121.00   Max.   :7.000        Max.   :2.000
##    PNEUMONIA         PREGNANT        DIABETES           COPD
##  Min.   : 1.000   Min.   : 1.00   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 2.000   1st Qu.: 2.00   1st Qu.: 2.000   1st Qu.: 2.000
##  Median : 2.000   Median :97.00   Median : 2.000   Median : 2.000
##  Mean   : 3.347   Mean   :49.77   Mean   : 2.186   Mean   : 2.261
##  3rd Qu.: 2.000   3rd Qu.:97.00   3rd Qu.: 2.000   3rd Qu.: 2.000
##  Max.   :99.000   Max.   :98.00   Max.   :98.000   Max.   :98.000
##      ASTHMA          INMSUPR        HIPERTENSION    CARDIOVASCULAR
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000
##  Median : 2.000   Median : 2.000   Median : 2.000   Median : 2.000
##  Mean   : 2.243   Mean   : 2.298   Mean   : 2.129   Mean   : 2.262
##  3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.: 2.000
##  Max.   :98.000   Max.   :98.000   Max.   :98.000   Max.   :98.000
##  RENAL_CHRONIC    OTHER_DISEASE       OBESITY          TOBACCO
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000
##  Median : 2.000   Median : 2.000   Median : 2.000   Median : 2.000
##  Mean   : 2.257   Mean   : 2.435   Mean   : 2.125   Mean   : 2.214
##  3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.: 2.000
##  Max.   :98.000   Max.   :98.000   Max.   :98.000   Max.   :98.000
##      USMER        MEDICAL_UNIT       INTUBED           ICU
##  Min.   :1.000   Min.   : 1.000   Min.   : 1.00   Min.   : 1.00
##  1st Qu.:1.000   1st Qu.: 4.000   1st Qu.:97.00   1st Qu.:97.00
##  Median :2.000   Median :12.000   Median :97.00   Median :97.00
##  Mean   :1.632   Mean   : 8.981   Mean   :79.52   Mean   :79.55
##  3rd Qu.:2.000   3rd Qu.:12.000   3rd Qu.:97.00   3rd Qu.:97.00
##  Max.   :2.000   Max.   :13.000   Max.   :99.00   Max.   :99.00
##   DATE_DIED
##  Length:1048575
##  Class :character
##  Mode  :character
```

```
##
##
##
```

Data Preparation

```
#Since all the boolean in the dataset uses 1("Yes") and 2("No"), convert rest of the columns

#Convert non-Death cases to NA
Covid$DATE_DIED <- as.Date(Covid$DATE_DIED)
#Convert values of 97, 98 and 99 to NAs in all binary columns
Covid <- Covid %>% na_if(97) %>% na_if(98) %>% na_if(99)

#Convert classification from 1-7 to binary 1("Yes") and 2("No")
Covid <- Covid%>%
  mutate(CLASSIFICATION = ifelse(CLASIFFICATION_FINAL<=3, 1 , 2))%>%
  filter(CLASSIFICATION == 1) #keep only covid-positive cases

#Convert death from date to binary 1("Yes") and 2("No")
Covid <- Covid%>%
  mutate(DEATH = ifelse(is.na(DATE_DIED), 2, 1))

#391979 obs of 23 variables
```

Data Exploration

```
xtabs(~DEATH, data = Covid)
```

```
## DEATH
##      1      2
##  54236 337743
```

```
xtabs(~DEATH+SEX, data = Covid) #death and covid distribution by sex
```

```
##      SEX
## DEATH      1      2
##     1  18959  35277
##     2 163531 174212
```

```
xtabs(~DEATH+PREGNANT, data = Covid) #death and covid distribution by PREGNANCY
```

```
##      PREGNANT
## DEATH      1      2
##     1     65  18853
##     2   2689 159500
```

Comment

Early observations for death and covid cases:
1) Male (sex =2) death rate is higher than female whether is covid positive or not
2) Since prenant women is only a small portion of female (1.55%) and our study is not focused on pregnant women, let's delete pregnant variable for the study 3) We can't impute the 345 NA records for age. Let's remove it since it is a small portion.
4) Let's also remove USMER, MEDICAL_UNIT. Not important where patients receive care
5) INTUBED and ICU still have high ratio of NAs. Assumne only severe cases need those procedures, it's ok to replace those NA with median which is 2("no")
6) The rest of NAs usually count than less than 1% of column data so let's impute the NAs by median number

Data Preparation

```
Covid <- Covid[-c(3, 6, 17, 18, 21, 22)] #delete pregnant, USMER, MEDICAL_UNIT etc
#391979 obs. of  17 variables
Covid <- Covid%>%
  filter(!is.na(AGE))  #391853 obs. of  17 variables

#impute missing values with mean
Covid[, c(4:16)] <- Covid[, c(4:16)]%>%
    mutate_if(is.numeric, function(x) ifelse(is.na(x), median(x, na.rm = T), x))

#Change data attribute from character to factor, the data is coded as 1 as no and 2 as yes
Covid$DEATH <- as.factor(Covid$DEATH)
```

Data partition

```
#Partition the given training data into 70% training data and 30% testing data
set.seed(100)
index_train <- createDataPartition(Covid$DEATH, p=0.7, list= F)
Covid_train <- Covid[index_train, ]
Covid_test <- Covid[-index_train, ]
```

Run logistic regression model

```
set.seed(1)
log_model <- glm(DEATH~., data = Covid_train, family = 'binomial')
log_model
```

```
##
## Call:  glm(formula = DEATH ~ ., family = "binomial", data = Covid_train)
##
## Coefficients:
##    (Intercept)             SEX             AGE    PATIENT_TYPE       PNEUMONIA
##       -0.70032        -0.41051        -0.05244        -1.98410         1.19472
##       DIABETES            COPD          ASTHMA         INMSUPR    HIPERTENSION
##        0.29444         0.13309        -0.07068         0.29018         0.12659
## CARDIOVASCULAR   RENAL_CHRONIC   OTHER_DISEASE         OBESITY         TOBACCO
##       -0.04596         0.73844         0.30944         0.23432        -0.13487
##        INTUBED             ICU
##        2.50672        -0.54479
##
## Degrees of Freedom: 274297 Total (i.e. Null);  274281 Residual
## Null Deviance:       220400
## Residual Deviance: 109100    AIC: 109100
```

Comment

Factors will increase the chance of death:
SEX(male), AGE (high), PATIENT TYPE (hospitalized), PNEUMONIA (positive), DIABETES (positive), COPD (negative), ASTHMA (negative), INMSUPR (positive), HIPERTENSION (positive), CARDIOVASCULAR (negative), RENAL_CHRONIC (positive), OTHER_DISEASE (positive), OBESITY (positive), TOBACCO(negative), INTUBED (positive), ICU(negative)

Run knn model

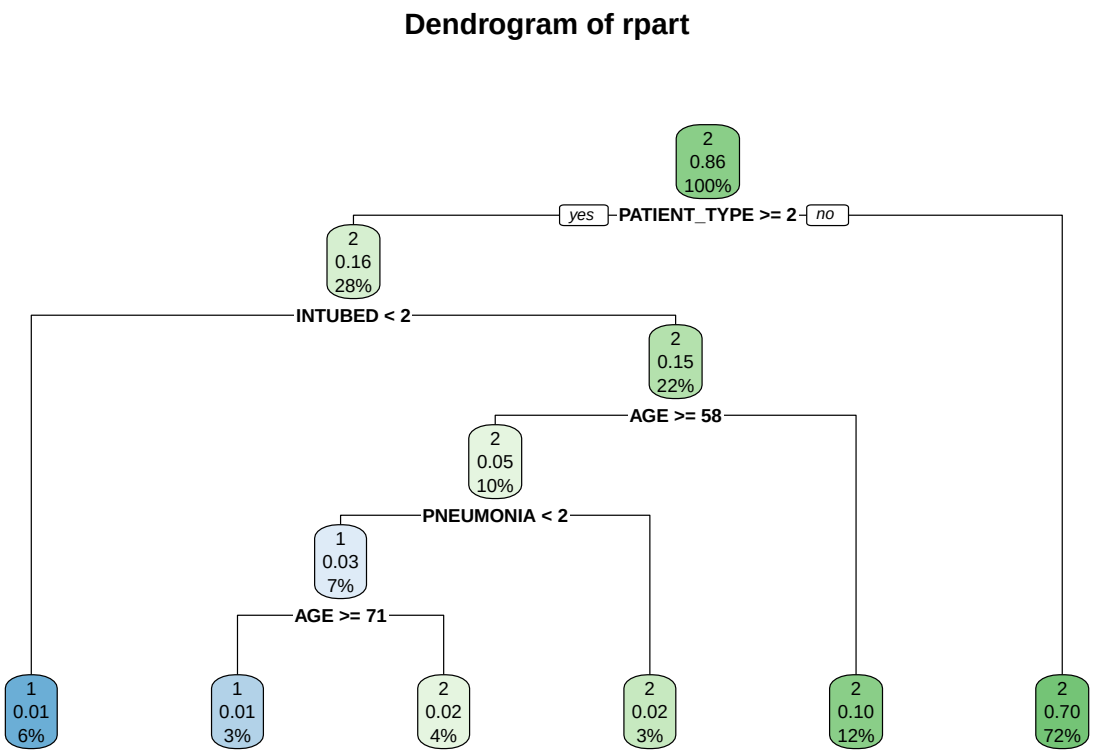
Comment

KNN model failed since there are too many ties and KNN can't deal with ties. This means that there are many similar data points which have the same distance.

Run NB model

Run Decision Tree

```
set.seed(4)
#agnes or hclust object does not work with later prediction
dt_model <- rpart(DEATH~., data = Covid_train, method = "class") #class for binary
rpart.plot(dt_model, extra = 110, main = "Dendrogram of rpart")
```

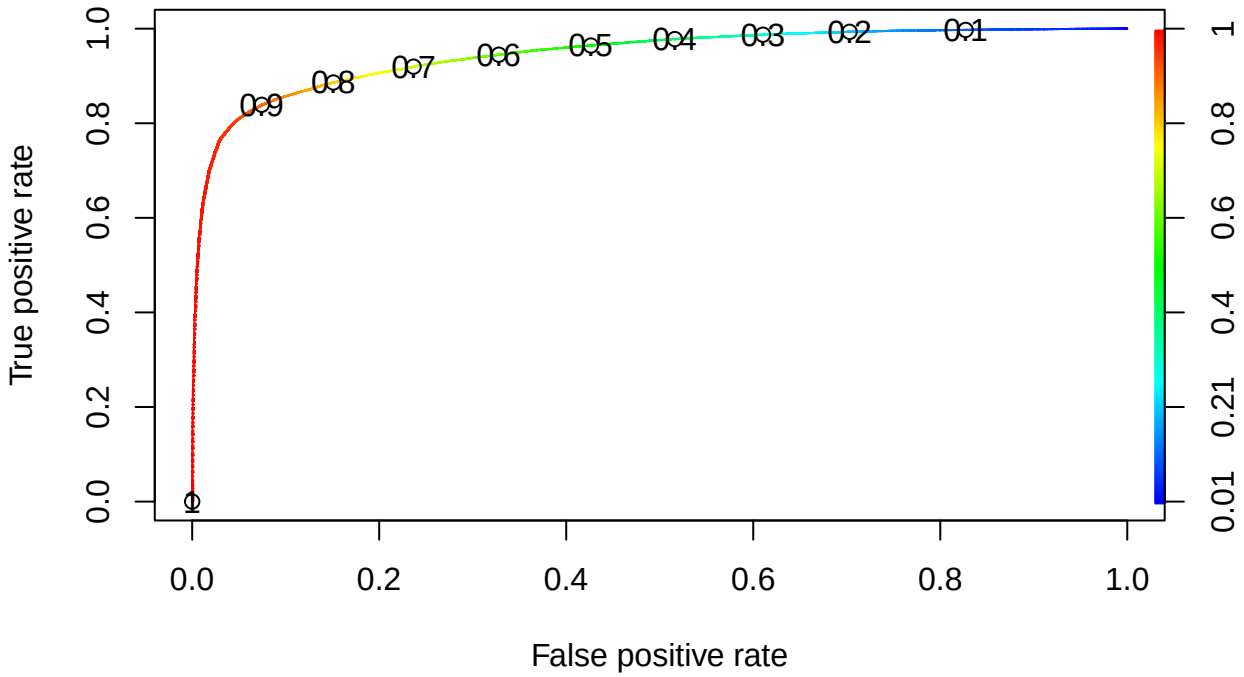


Model Testing

```
#Test the logistic regression model and return in probability
log_test_prob <- predict(log_model, Covid_test, type = "response")

#Test the knn model
#knn_test_prob <- predict(knn_model, Covid_test, type = "prob")

#Test the nb model
nb_test_prob <- predict(nb_model, Covid_test, type = "raw")

#Test the dt model- (predict does not apply to "hclust" or "agnes" object)
dt_test_prob <- predict(dt_model, Covid_test, type = "prob")
```

Model Comparison: Thresholding, best cutoff point, confusion table and ROC

```
#logistic regression
pred_log_test <- prediction(log_test_prob, Covid_test$DEATH)#create prediction obj

#TPR FPR plot
roc_perf_log_test <- performance(pred_log_test, measure = "tpr", x.measure = "fpr")
plot(roc_perf_log_test,colorize=TRUE,print.cutoffs.at=seq(0.1,by=0.1))
```




```
#TPR/FPR cutoff graph<br>
```

```
#Logistic regression AUC value
auc.perf = performance(pred_log_test, measure = "auc")
auc.perf@y.values
```

```
## [[1]]
## [1] 0.9453144
```

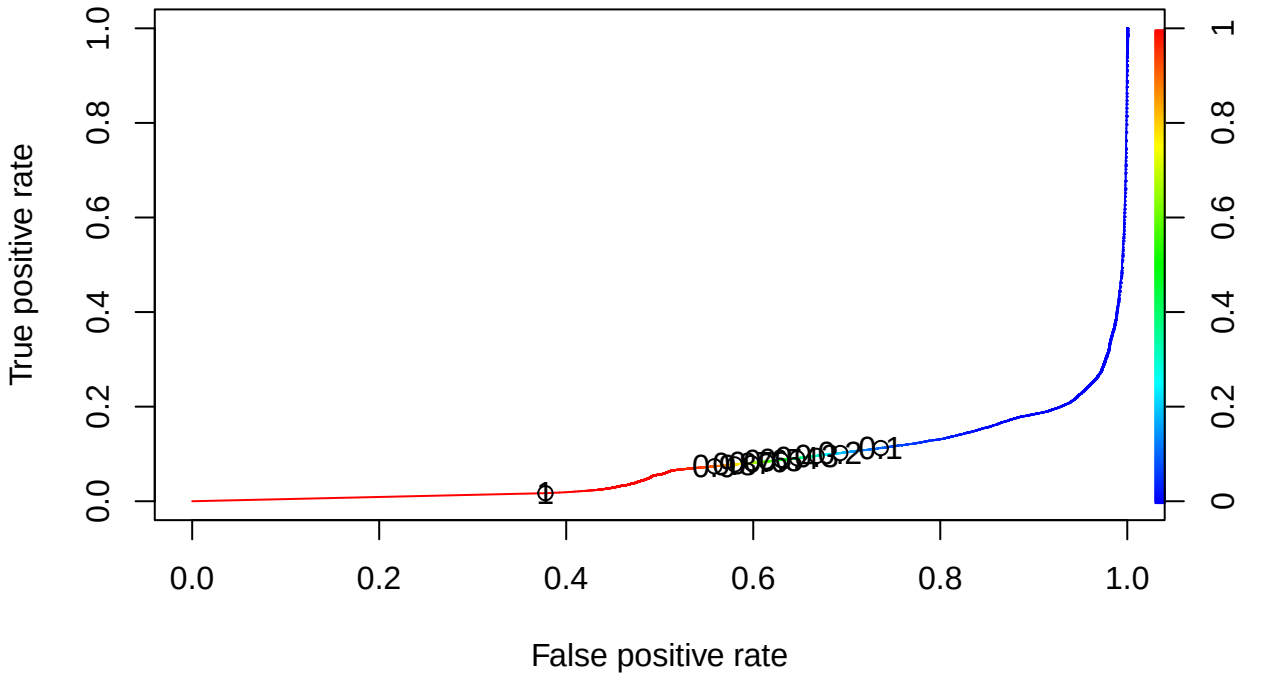
```
#Confusion table
confusionMatrix(as.factor(ifelse(log_test_prob>0.1, "1", "2")), Covid_test$DEATH, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      1      2
##          1  13442 101037
##          2   2812    264
##
##                Accuracy : 0.1166
##                  95% CI : (0.1148, 0.1184)
##     No Information Rate : 0.8617
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.0482
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.826996
##             Specificity : 0.002606
##          Pos Pred Value : 0.117419
##          Neg Pred Value : 0.085826
##              Prevalence : 0.138267
##          Detection Rate : 0.114346
```

```
##    Detection Prevalence : 0.973834
##       Balanced Accuracy : 0.414801
##
##        'Positive' Class : 1
##
```

Logistic Regression Metric
True Positive (TP) = 13442
True Negative (TN) = 264
False Positive (FP) = 101037
False Negative (FN) = 2812
Miscalculations = 103849
Accuracy = 11.66%
Sensitivity = 82.70%
Specificity = 0.26%

```
#Naive Bayes
pred_nb_test <- prediction(nb_test_prob[,1], Covid_test$DEATH)
roc_perf_nb_test <- performance(pred_nb_test, measure = "tpr", x.measure = "fpr")
plot(roc_perf_nb_test,colorize=TRUE,print.cutoffs.at=seq(0.1,by=0.1))
```




```
#Calculate ROC value for binary classifier
roc.curve(Covid_test$DEATH, nb_test_prob[,1], plotit= F)
```

```
## Area under the curve (AUC): 0.921
```

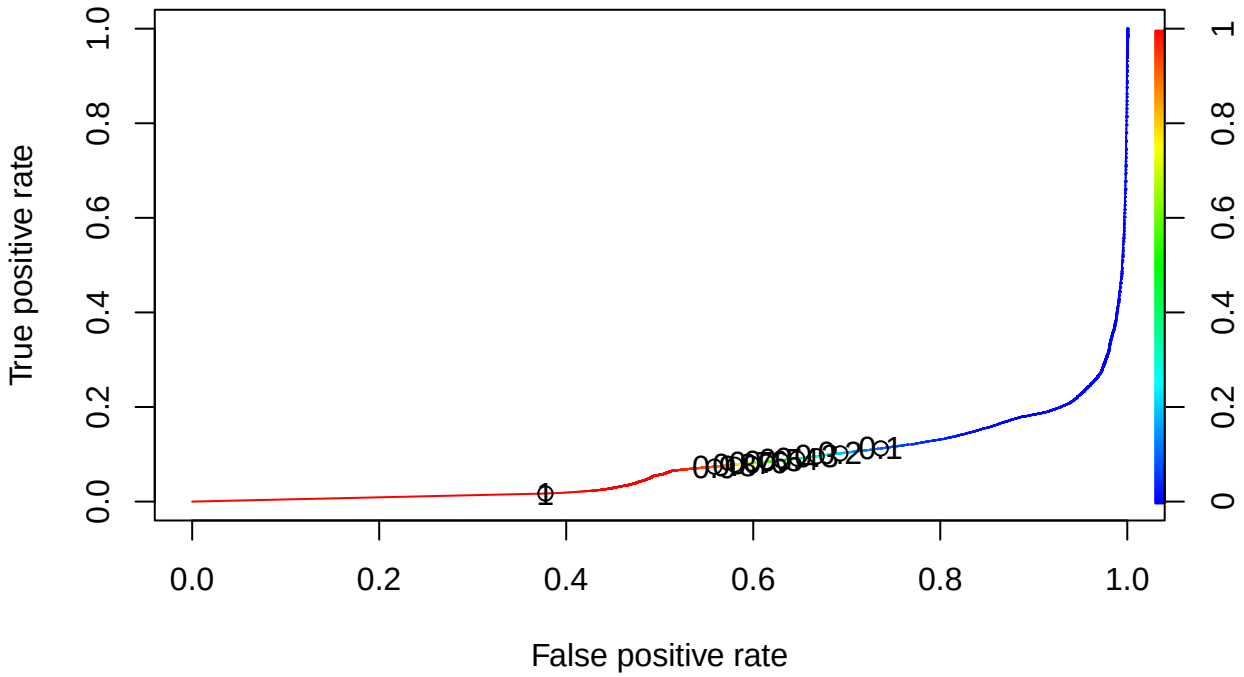
```
confusionMatrix(as.factor(ifelse(nb_test_prob[,1]>0.01, "1", "2")), Covid_test$DEATH, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 13857 15964
```

```
##          2  2397 85337
##
##                Accuracy : 0.8438
##                  95% CI : (0.8417, 0.8459)
##     No Information Rate : 0.8617
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.5146
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8525
##             Specificity : 0.8424
##          Pos Pred Value : 0.4647
##          Neg Pred Value : 0.9727
##              Prevalence : 0.1383
##          Detection Rate : 0.1179
##    Detection Prevalence : 0.2537
##       Balanced Accuracy : 0.8475
##
##        'Positive' Class : 1
##
```

Naive Bayes Metric
True Positive (TP) = 13857
True Negative (TN) = 85337
False Positive (FP) = 15964
False Negative (FN) = 2397
Miscalculations = 18361
Accuracy = 84.38%
Sensitivity = 85.25%
Specificity = 84.24%

```
#decision tree (dt): create prediction object for ROCR evaluation
pred_dt_test <- prediction(dt_test_prob[,1], Covid_test$DEATH)
roc_perf_dt_test <- performance(pred_dt_test, measure = "tpr", x.measure = "fpr")
plot(roc_perf_nb_test,colorize=TRUE,print.cutoffs.at=seq(0.1,by=0.1))
```




```
#Calculate ROC value for binary classifier
roc.curve(Covid_test$DEATH, dt_test_prob[,1], plotit= F)
```

```
## Area under the curve (AUC): 0.906
```

```
confusionMatrix(as.factor(ifelse(dt_test_prob[,1]>0.2, "1", "2")), Covid_test$DEATH, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 14752 18717
##          2  1502 82584
##
##                Accuracy : 0.828
##                  95% CI : (0.8258, 0.8302)
##     No Information Rate : 0.8617
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.5004
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9076
##             Specificity : 0.8152
##          Pos Pred Value : 0.4408
##          Neg Pred Value : 0.9821
##              Prevalence : 0.1383
##          Detection Rate : 0.1255
##    Detection Prevalence : 0.2847
##       Balanced Accuracy : 0.8614
##
##        'Positive' Class : 1
```

##

Decision Tree Metric
True Positive (TP) = 14752
True Negative (TN) = 82584
False Positive (FP) = 18717
False Negative (FN) = 1502
Miscalculations = 20219
Accuracy = 82.8%
Sensitivity = 90.76% Specificity = 81.52%

Conclusion
The model aims to reduce false negatives and tolerates more on false positives. It will cost more to miss a covid positive patient than to mis-classify a negative one.
The next step is to discover the cost for false positive and false negative patient to adjust the model to save the total cost.