COVID survival

Maris Sekar

11/11/2020

Executive Summary

The COVID-19 virus has taken on the world by storm since its discovery late December of 2019. 1.28 million deaths have been reported worldwide at the time of the writing of this report and this number is seen to be climbing almost one year into its discovery. It would be helpful to see the various factors influencing COVID related deaths using a ML model. In this project we are going to predict the COVID death based on patient features such as age, gender, and medical conditions.

After reviewing a few data sources, the Mexico Government COVID data set seems to be the most complete data set found on Kaggle.com. I tried three classification models - Decision Tree, KNN and Random Forest and finally an Ensemble to see if any improvements can be generated. The Decision Tree model appears to be the most accurate out of the three for this data set. Due to the high prevalence of survivors the specificity of the models are low. Overall accuracy was about 94.8% and balanced accuracy of the model was about 67%.

Link to dataset: https://www.kaggle.com/tanmoyx/covid19-patient-precondition-dataset?select=covid.csv

Methods/Analysis

Lets load the required libraries.

```
# Required libraries
if (!require("readr")) install.packages("readr")
if (!require("rpart.plot")) install.packages("rpart.plot")
if (!require("dplyr")) install.packages("dplyr")
if (!require("caret")) install.packages("caret")
if (!require("randomForest")) install.packages("randomForest")
library(readr)
library(caret)
library(dplyr)
library(rpart.plot)
library(randomForest)
```

Lets load the COVID patient characteristics from my github.

```
urlfile="https://raw.githubusercontent.com/cube27/harvardx-r-exercises/master/covid/data/covid.csv"
covid <- read_csv(url(urlfile))
```

We see that there is date_died field that records the date of death. Lets create a target variable, “died”, that keeps track of the death. A one indicates that the patient expired and zero means they are alive.

```
# Add field to data set
covid <- covid %>% mutate(died=ifelse(covid$date_died == "9999-99-99", 0, 1))
head(covid$date_died)
```

```
## [1] "9999-99-99" "9999-99-99" "9999-99-99" "9999-99-99" "22-04-2020"
## [6] "29-04-2020"
```

```
head(covid$died)
```

```
## [1] 0 0 0 0 1 1
```

The class needs to be a factor for classification models. Change type of the died field to factor data type.

```
covid$died <- as.factor(covid$died)
class(covid$died)
```

```
## [1] "factor"
```

Lets review the distribution of the COVID death population.

```
table(covid$died)
```

```
##
##      0      1
## 530426  36176
```

We will split the data to training and validation datasets. A 90/10 split is performed because we have a considerably large dataset.

```
# Split the data into 90% training data and 10% test data for validation.
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(y = covid$died, times = 1,
                                  p = 0.1, list = FALSE)
train_set <- covid[-test_index,]
test_set <- covid[test_index,]
```

Exploratory data analysis

Lets review the columns in detail:

```
names(train_set)
```

```
##  [1] "id"                "sex"               "patient_type"
##  [4] "entry_date"        "date_symptoms"     "date_died"
##  [7] "intubed"           "pneumonia"         "age"
## [10] "pregnancy"         "diabetes"          "copd"
## [13] "asthma"            "inmsupr"           "hypertension"
## [16] "other_disease"     "cardiovascular"    "obesity"
## [19] "renal_chronic"     "tobacco"           "contact_other_covid"
## [22] "covid_res"         "icu"               "died"
```

Dimensions of the training data:

```
dim(train_set)
```

```
## [1] 509941     24
```

Get a subset of features in the data set. Only the numeric columns are selected as features. 19 features are selected.

```
features <- c("sex", "patient_type", "intubed", "pneumonia", "age", "pregnancy", "diabetes", "copd", "as
```

We are going to to do some preprocessing. We select only 10000 random samples from the training set to train the classification models. We will also select 1000 random samples from the validation dataset to do our actual validation.

```
set.seed(1, sample.kind = "Rounding")
# Sample 10000 observations from the training data.
index <- sample(nrow(train_set), 10000)
x <- train_set[index, features]
y <- train_set$died[index]

# Sample 1000 observations from the validation data
index <- sample(nrow(test_set), 1000)
x_test <- test_set[index, features]
y_test <- test_set$died[index]
```

Lets analyze the relationship between the features.

```
# Find correlation matrix of features
cor(x)
```

```
##                             sex patient_type      intubed    pneumonia
## sex                1.0000000000  0.099185658 -0.099002701 -0.092596427
## patient_type       0.0991856578  1.000000000 -0.999072224 -0.665973979
## intubed           -0.0990027006 -0.999072224  1.000000000  0.665648831
## pneumonia         -0.0925964267 -0.665973979  0.665648831  1.000000000
## age                0.0221511230  0.329994435 -0.329736170 -0.298542042
## pregnancy          0.9949047987  0.098436840 -0.098260804 -0.092596709
## diabetes           0.0064920396  0.017761598 -0.017818878 -0.002837410
## copd              -0.0006446835  0.018378315 -0.018472944 -0.015832870
## asthma            -0.0025110981  0.024257609 -0.024351413 -0.012887197
## inmsupr            0.0010038742  0.023887596 -0.023975949 -0.010877454
## hypertension      -0.0058825130 -0.001157783  0.001040707  0.002269500
## other_disease      0.0116148491  0.044097487 -0.044201989 -0.040252038
## cardiovascular    -0.0042408157  0.030692997 -0.030776584 -0.017228781
## obesity           -0.0027564081  0.015841698 -0.015914385 -0.006509731
## renal_chronic     -0.0012744292  0.030874647 -0.030960166 -0.019131251
## tobacco           -0.0044220804  0.021682573 -0.021767000 -0.014976937
## contact_other_covid  0.0091235676  0.219225744 -0.218994349 -0.106134842
## covid_res         -0.0478410380 -0.128328628  0.127602864  0.140814044
## icu               -0.0990742661 -0.999072224  0.999994317  0.665537091
##                            age    pregnancy     diabetes          copd
```

```
## sex                  0.022151123  0.994904799  0.006492040 -0.0006446835
## patient_type         0.329994435  0.098436840  0.017761598  0.0183783149
## intubed             -0.329736170 -0.098260804 -0.017818878 -0.0184729437
## pneumonia           -0.298542042 -0.092596709 -0.002837410 -0.0158328704
## age                  1.000000000  0.022514394 -0.005456498  0.0124267146
## pregnancy            0.022514394  1.000000000  0.015794383  0.0097487827
## diabetes            -0.005456498  0.015794383  1.000000000  0.7805042177
## copd                 0.012426715  0.009748783  0.780504218  1.0000000000
## asthma               0.015957256  0.007566341  0.811148656  0.9085426869
## inmsupr              0.011018440  0.010399557  0.782796089  0.8483317857
## hypertension        -0.009319798  0.004708675  0.820875848  0.9184069171
## other_disease        0.009479119  0.019432958  0.679452421  0.7137520892
## cardiovascular       0.008811990  0.005600927  0.841221083  0.8535510056
## obesity              0.006518358  0.008021717  0.804640387  0.8356939937
## renal_chronic        0.013045729  0.008288005  0.818551674  0.8594164000
## tobacco              0.011884261  0.005609016  0.798749486  0.8942287103
## contact_other_covid  0.096390990  0.009250661  0.008483167  0.0111322867
## covid_res           -0.107648741 -0.048818545  0.010449351  0.0073178518
## icu                 -0.329646491 -0.098332361 -0.017783127 -0.0183820126
##                           asthma      inmsupr hypertension other_disease
## sex                 -0.002511098  0.001003874 -0.005882513   0.011614849
## patient_type         0.024257609  0.023887596 -0.001157783   0.044097487
## intubed             -0.024351413 -0.023975949  0.001040707  -0.044201989
## pneumonia           -0.012887197 -0.010877454  0.002269500  -0.040252038
## age                  0.015957256  0.011018440 -0.009319798   0.009479119
## pregnancy            0.007566341  0.010399557  0.004708675   0.019432958
## diabetes             0.811148656  0.782796089  0.820875848   0.679452421
## copd                 0.908542687  0.848331786  0.918406917   0.713752089
## asthma               1.000000000  0.905030021  0.890354155   0.714671215
## inmsupr              0.905030021  1.000000000  0.830946548   0.775861807
## hypertension         0.890354155  0.830946548  1.000000000   0.723709273
## other_disease        0.714671215  0.775861807  0.723709273   1.000000000
## cardiovascular       0.885029882  0.853113570  0.865637621   0.762804993
## obesity              0.842252039  0.786519305  0.880440308   0.661338221
## renal_chronic        0.861348957  0.830288315  0.871122729   0.720377042
## tobacco              0.896834661  0.837199134  0.907143495   0.704110010
## contact_other_covid  0.016869122  0.009491812  0.006880522  -0.005594743
## covid_res            0.008259966  0.010733161  0.016333819   0.005889789
## icu                 -0.024263221 -0.023894397  0.001132754  -0.044133634
##                     cardiovascular      obesity renal_chronic      tobacco
## sex                   -0.004240816 -0.002756408  -0.001274429 -0.004422080
## patient_type           0.030692997  0.015841698   0.030874647  0.021682573
## intubed               -0.030776584 -0.015914385  -0.030960166 -0.021767000
## pneumonia             -0.017228781 -0.006509731  -0.019131251 -0.014976937
## age                    0.008811990  0.006518358   0.013045729  0.011884261
## pregnancy              0.005600927  0.008021717   0.008288005  0.005609016
## diabetes               0.841221083  0.804640387   0.818551674  0.798749486
## copd                   0.853551006  0.835693994   0.859416400  0.894228710
## asthma                 0.885029882  0.842252039   0.861348957  0.896834661
## inmsupr                0.853113570  0.786519305   0.830288315  0.837199134
## hypertension           0.865637621  0.880440308   0.871122729  0.907143495
## other_disease          0.762804993  0.661338221   0.720377042  0.704110010
## cardiovascular         1.000000000  0.818634316   0.891402360  0.871695301
## obesity                0.818634316  1.000000000   0.855285968  0.860190466
```

```
## renal_chronic          0.891402360  0.855285968   1.000000000  0.875853117
## tobacco                0.871695301  0.860190466   0.875853117  1.000000000
## contact_other_covid    0.017360205  0.016852356   0.013763591  0.015456762
## covid_res              0.015225995  0.019419484   0.003745176  0.005914390
## icu                   -0.030691285 -0.015822296  -0.030918648 -0.021679737
##                     contact_other_covid    covid_res          icu
## sex                         0.009123568 -0.047841038 -0.099074266
## patient_type                0.219225744 -0.128328628 -0.999072224
## intubed                    -0.218994349  0.127602864  0.999994317
## pneumonia                  -0.106134842  0.140814044  0.665537091
## age                         0.096390990 -0.107648741 -0.329646491
## pregnancy                   0.009250661 -0.048818545 -0.098332361
## diabetes                    0.008483167  0.010449351 -0.017783127
## copd                        0.011132287  0.007317852 -0.018382013
## asthma                      0.016869122  0.008259966 -0.024263221
## inmsupr                     0.009491812  0.010733161 -0.023894397
## hypertension                0.006880522  0.016333819  0.001132754
## other_disease              -0.005594743  0.005889789 -0.044133634
## cardiovascular              0.017360205  0.015225995 -0.030691285
## obesity                     0.016852356  0.019419484 -0.015822296
## renal_chronic               0.013763591  0.003745176 -0.030918648
## tobacco                     0.015456762  0.005914390 -0.021679737
## contact_other_covid         1.000000000 -0.073802527 -0.219032769
## covid_res                  -0.073802527  1.000000000  0.127614430
## icu                        -0.219032769  0.127614430  1.000000000
```

There are some interesting relationships highlighted above: - Gender is highly correlated with pregnancy. Since females are most likely to be pregnant. This makes sense. - Patient type is highly correlated with intubated, icu and pneumonia medical conditions. I am not quite sure what patient type is but it appears to be a category assigned to patients based on their medical condition which we can see in the correlation matrix. - Patients who had pneumonia and were in icu were most likely intubated. - Younger patients were most likely intubated and had pneumonia due to the inverse relationship between age and these medical conditions. - Diabetic and obese patients were more likely to have other medical conditions as well.

Next, lets ignore fields that are near zero to avoid noise.

```
nzv <- nearZeroVar(x)

# Strip the near zero columns.
col_index <- setdiff(1:ncol(x), nzv)
length(col_index)
```

```
## [1] 13
```

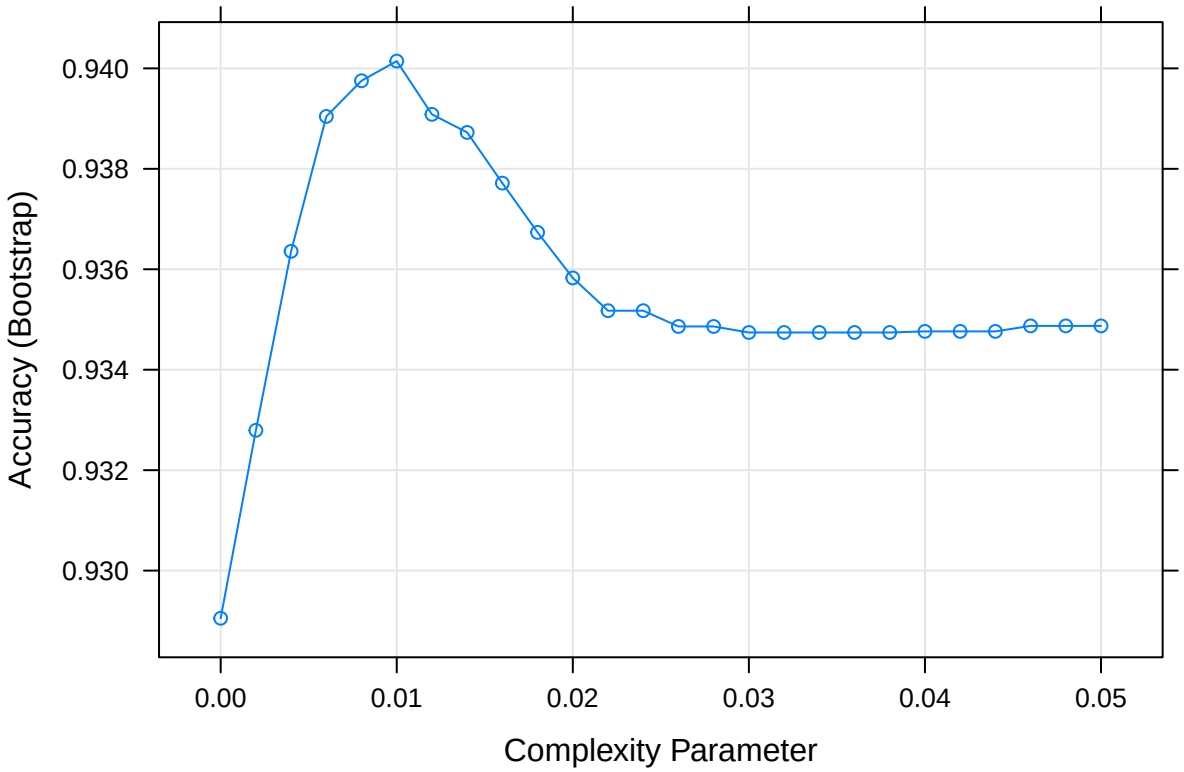
There are 13 fields after this. 6 features were removed.

Modeling/Analysis

Lets train a Decision Tree Model and plot to see which complexity parameter gives the most accuracy.

```
set.seed(1, sample.kind = "Rounding")
train_rpart <- train(x[, col_index], y,
```

```
                      method = "rpart",
                      tuneGrid = data.frame(cp = seq(0, 0.05, 0.002)))
# Plot the Accuracy vs. complexity paratemeter, cp
plot_rpart <- plot(train_rpart)
plot_rpart
```



We can see cp = 0.01 gives the most accuracy.

Get confusion matrix for the model and get the balanced accuracy.

```
rpart_cm <- confusionMatrix(predict(train_rpart, x_test), y_test)
rpart_cm$byClass["Balanced Accuracy"]
```

```
## Balanced Accuracy
##         0.6742813
```

```
accuracy_results <- tibble(Model = "Decision Tree Model",
                           `Overall Accuracy` = rpart_cm$overall["Accuracy"],
                           `Balanced Accuracy` = rpart_cm$byClass["Balanced Accuracy"])
```

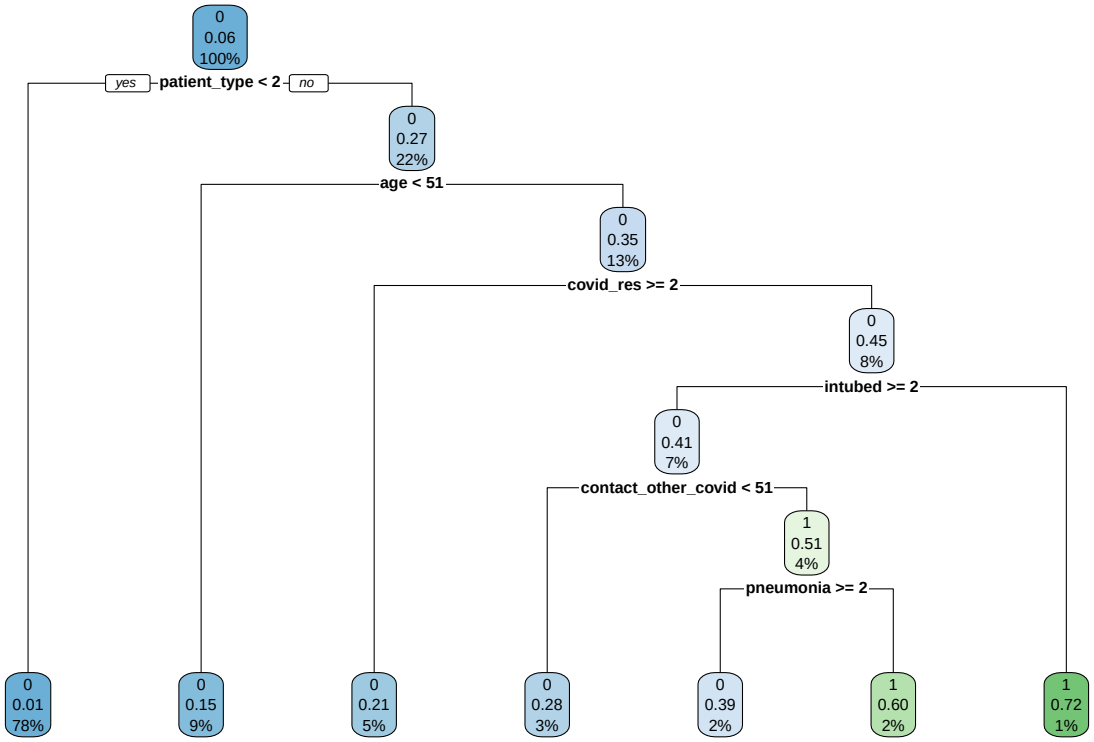
Please note that the the regular accuracy is 94.8% as shown below. The above is the balanced accuracy taking specificity and sensitivity into account.

```
rpart_cm$overall["Accuracy"]
```

```
## Accuracy
##    0.948
```

Access the final model and plot it.

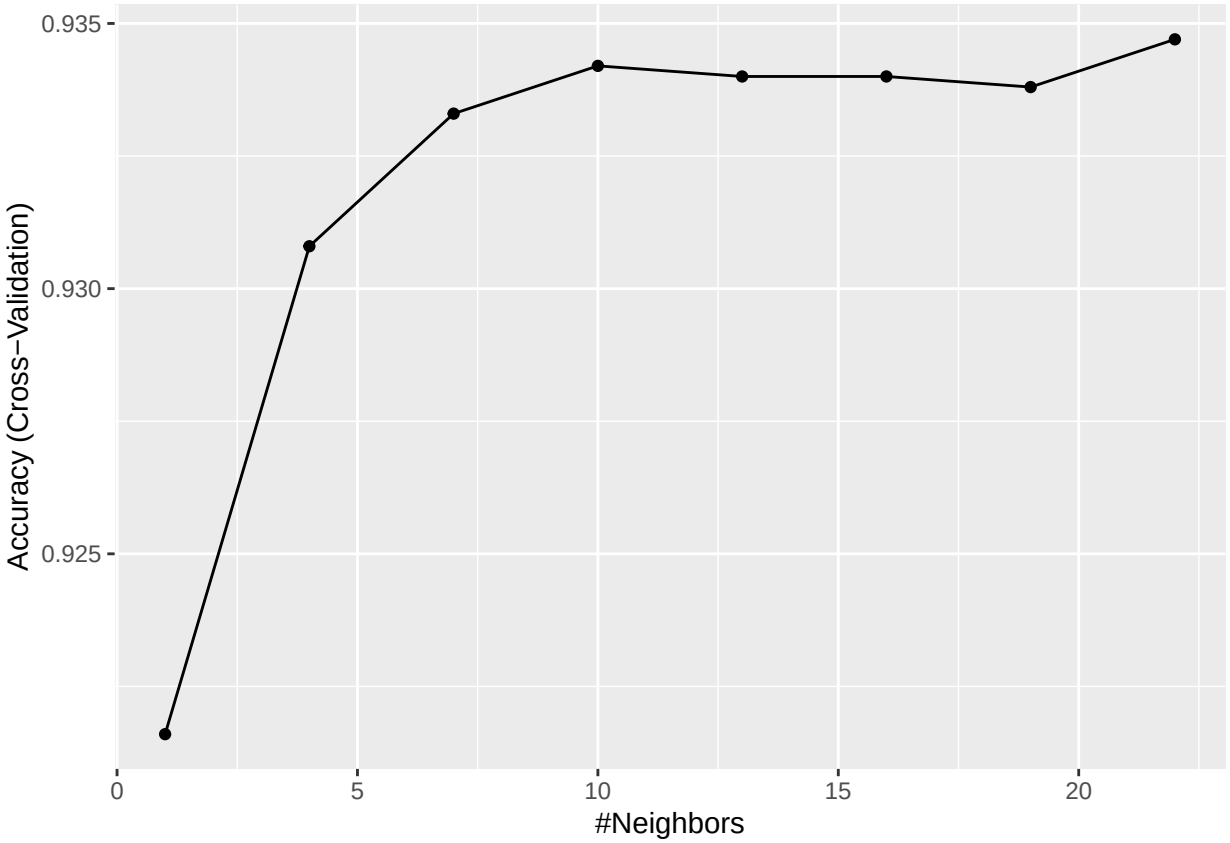
```
rpart.plot(train_rpart$finalModel)
```



Looks like the features patient_type, age, covid_res, intubed, contact_other_covid and pneumonia captures almost all the COVID situations seen in the dataset. We can see patients older than 51 are more likely to expire provided they are intubed and have pneumonia. This shows the small percentage of COVID deaths on the bottom right side of the decision tree.

Lets now turn to the KNN Model. We will train a KNN model and plot it to find the no. of neighbors producing the most accuracy.

```
set.seed(1, sample.kind = "Rounding")
control <- trainControl(method = "cv", number = 10, p = .9)
train_knn <- train(x[,col_index], y,
                   method = "knn",
                   tuneGrid = data.frame(k = seq(1, 24, 3)),
                   trControl = control)
# Plot the KNN model Accuracy vs. No. of Neighbors
ggplot(train_knn)
```



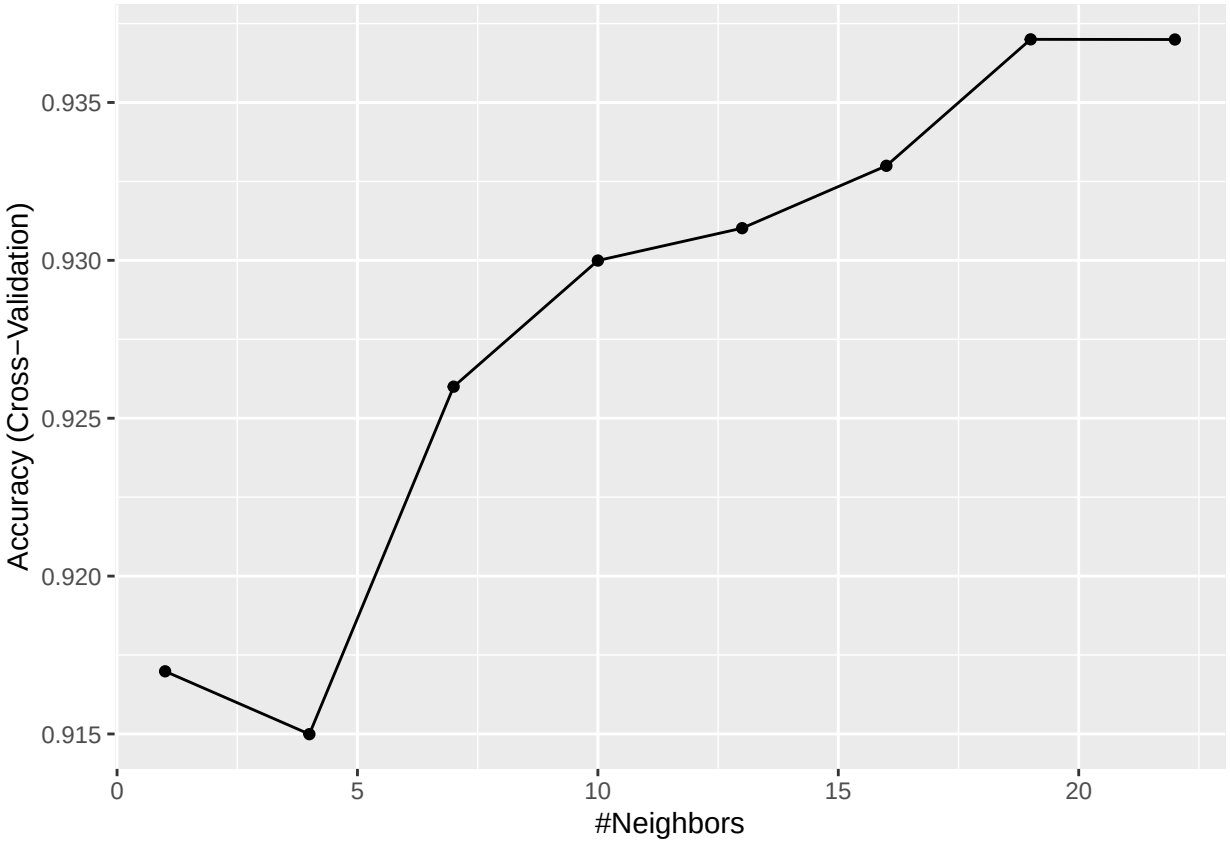
The bestTune field indicates:

```
train_knn$bestTune[1]
```

```
##    k
## 8 22
```

Now lets repeat with 5-fold cross validation KNN model. Each time it will sample 1000 samples.

```
# sample size
n <- 1000
# no. of folds
b <- 5
set.seed(1)
index <- sample(nrow(x), n)
set.seed(1, sample.kind = "Rounding")
control <- trainControl(method = "cv", number = b, p = .9)
train_knn <- train(x[index ,col_index], y[index],
                   method = "knn",
                   tuneGrid = data.frame(k = seq(1, 24, 3)),
                   trControl = control)
# Plot the KNN model Accuracy vs. No. of Neighbors
ggplot(train_knn)
```



Now the bestTune field indicates:

```
train_knn$bestTune[1]
```

```
##    k
## 7 19
```

Lets fit the knn model based on this tuned k value. We will make predictions on the validation data set and create a confusion matrix.

```
fit_knn <- knn3(x[ ,col_index], y,  k = train_knn$bestTune[1])

y_hat_knn <- predict(fit_knn,
                     x_test[, col_index],
                     type="class")
knn_cm <- confusionMatrix(y_hat_knn, factor(y_test))
knn_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 928  51
##          1   5  16
##
```

```
##                Accuracy : 0.944
##                  95% CI : (0.9279, 0.9574)
##     No Information Rate : 0.933
##     P-Value [Acc > NIR] : 0.08952
##
##                   Kappa : 0.3426
##
##  Mcnemar's Test P-Value : 1.817e-09
##
##             Sensitivity : 0.9946
##             Specificity : 0.2388
##          Pos Pred Value : 0.9479
##          Neg Pred Value : 0.7619
##              Prevalence : 0.9330
##          Detection Rate : 0.9280
##    Detection Prevalence : 0.9790
##       Balanced Accuracy : 0.6167
##
##        'Positive' Class : 0
##
```

Again, much lower specificity but high sensitivity. This is due to the high prevalence in the positive class, 0, with a prevalance of 0.933. Accuracy is 94.4% and balanced accuracy is 61.7%. This is lower than the Decision Tree model.

Record the balanced accuracy in our results table.

```
accuracy_results <- bind_rows(accuracy_results,
                          data_frame(Model = "KNN Model",
                                     `Overall Accuracy` = knn_cm$overall["Accuracy"],
                                     `Balanced Accuracy` = knn_cm$byClass["Balanced Accuracy"] ))
```

Lets check prevalence in data to confirm the survival rate.

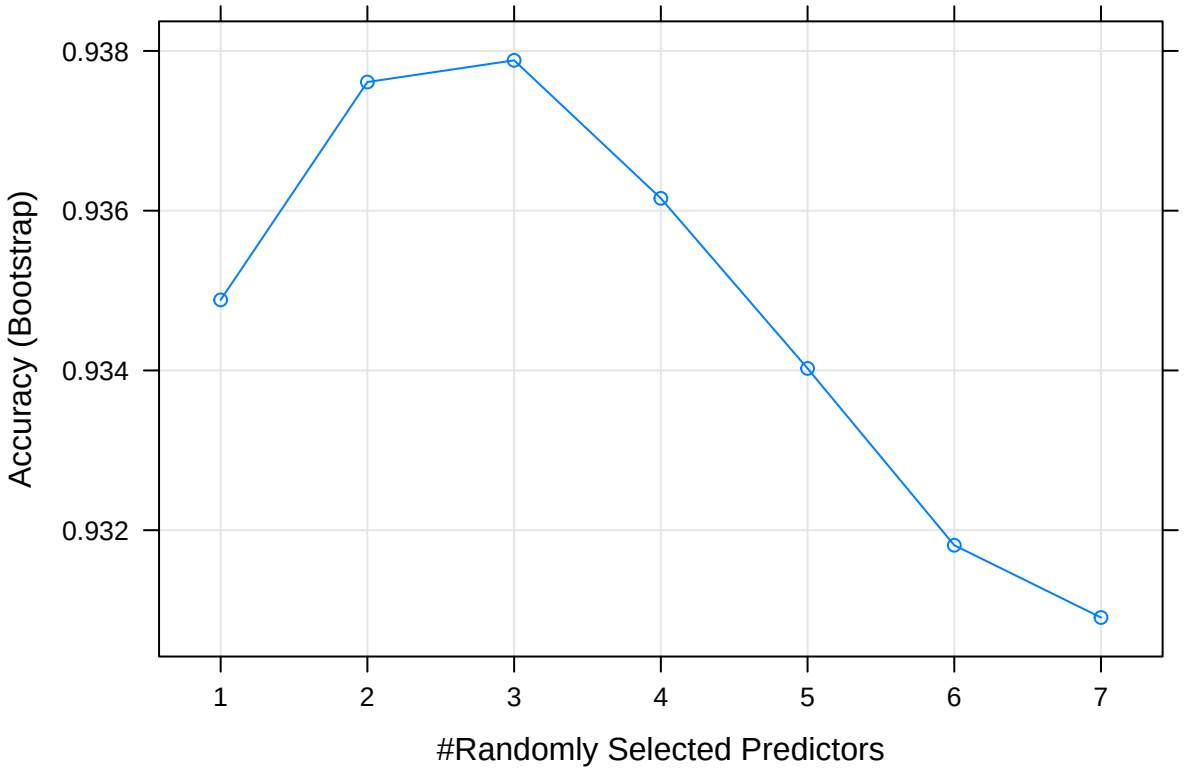
```
mean(covid$died == "0")
```

```
## [1] 0.9361527
```

This confirms the prevalence we saw earlier above.

Now we will train a Random Forest classifier. We choose to use 100 trees. We will also find the optimal mtry number - number of variables available for splitting at each tree node.

```
set.seed(1, sample.kind = "Rounding")
train_rf <- train(x[, col_index], y, method = "rf",
                  ntree = 100,
                  tuneGrid = data.frame(mtry = seq(1:7)))
# Plot Random Forest Accuracy vs. No. of Randomly selected predictors
plot(train_rf)
```



The optimal mtry value appears to be 3.

Lets look at the confusion matrix after testing the predicted values with the validation data set.

```
rf_cm <- confusionMatrix(predict(train_rf, x_test), y_test)
rf_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 926  51
##          1   7  16
##
##                Accuracy : 0.942
##                  95% CI : (0.9257, 0.9557)
##     No Information Rate : 0.933
##     P-Value [Acc > NIR] : 0.1404
##
##                   Kappa : 0.3327
##
##  Mcnemar's Test P-Value : 1.641e-08
##
##             Sensitivity : 0.9925
##             Specificity : 0.2388
##          Pos Pred Value : 0.9478
##          Neg Pred Value : 0.6957
```

```
##              Prevalence : 0.9330
##          Detection Rate : 0.9260
##    Detection Prevalence : 0.9770
##       Balanced Accuracy : 0.6157
##
##        'Positive' Class : 0
##
```

Again, similar results to the KNN model. Decision Tree model balanced accuracy is still the highest we have.

We will record the balanced accuracy in our results table for comparison.

```
accuracy_results <- bind_rows(accuracy_results,
                              data_frame(Model = "Random Forest",
                                         `Overall Accuracy` = rf_cm$overall["Accuracy"],
                                         `Balanced Accuracy` = rf_cm$byClass["Balanced Accuracy"] ))
```

Find features importance for the random forest model.

```
imp <- varImp(train_rf)
imp
```

```
## rf variable importance
##
##                      Overall
## age                 100.0000
## intubed              48.5263
## pneumonia            46.2546
## icu                  43.1339
## patient_type         37.8595
## covid_res            25.1438
## contact_other_covid  23.8384
## diabetes              7.5977
## hypertension          5.8880
## obesity               5.3919
## tobacco               2.2945
## sex                   0.3699
## pregnancy             0.0000
```

We can see how age, intubed, pneumonia, icu and patient_type take the top 5 places for feature importance. This makes sense based on the statistics that is publicly available.

Lets combine with the tree terms from the Decision Tree model.

```
tree_terms <- as.character(unique(train_rpart$finalModel$frame$var[!(train_rpart$finalModel$frame$var =
tree_terms
```

```
## [1] "patient_type"        "age"                 "covid_res"
## [4] "intubed"             "contact_other_covid" "pneumonia"
```

We can see that almost all the terms we saw in the Decision Tree model agrees with the feature importance from the random forest model except for icu.

Lets rank the combined features based on their importance.

```r
tibble(term = rownames(imp$importance), importance = imp$importance$Overall) %>%
  mutate(rank = rank(-importance)) %>%
  arrange(desc(importance)) %>%
  filter(term %in% tree_terms)
```

```
## # A tibble: 6 x 3
##   term                importance  rank
##   <chr>                    <dbl> <dbl>
## 1 age                      100       1
## 2 intubed                   48.5     2
## 3 pneumonia                 46.3     3
## 4 patient_type              37.9     5
## 5 covid_res                 25.1     6
## 6 contact_other_covid       23.8     7
```

Age is the main factor contributing to a COVID death followed by the medical conditions of being intubated and having pneumonia. This is very interesting and supports the views of the statistics we have publicly available.

Lets build an ensemble with Decision Tree, KNN and Random Forest models to see if we can improve our accuracy.

```r
# Combine all the three model predictions
pred_rpart <- predict(train_rpart, x_test)
pred_knn <- predict(train_knn, x_test)
pred_rf <- predict(train_rf, x_test)
pred <- cbind(rpart=as.numeric(pred_rpart)-1, knn=as.numeric(pred_knn)-1, rf=as.numeric(pred_rf)-1)
dim(pred)
```

```
## [1] 1000    3
```

Calculate average accuracy for each model. Note: this is not the balanced accuracy!

```r
accuracy <- colMeans(pred == y_test)
accuracy
```

```
## rpart   knn    rf
## 0.948 0.931 0.943
```

```r
mean(accuracy)
```

```
## [1] 0.9406667
```

The Ensemble model accuracy is calculated below taking the best accuracy for each observation.

```r
means_died <- rowMeans(pred == 1)
y_hat <- ifelse(means_died > 0.3, "1", "0")
mean(y_hat == y_test)
```

```
## [1] 0.942
```

Results

Here are the results of our models:

```
accuracy_results
```

```
## # A tibble: 3 x 3
##   Model               `Overall Accuracy` `Balanced Accuracy`
##   <chr>                            <dbl>               <dbl>
## 1 Decision Tree Model              0.948               0.674
## 2 KNN Model                        0.944               0.617
## 3 Random Forest                    0.942               0.616
```

The overall accuracy is very similar for all models with the Decision Tree model being slightly higher at 94.8%. The balanced accuracy for the Decision Tree model is comparatively the highest at 67.4% after taking into account the sensitivity and specificity. This lower balanced accuracy is due to the prevalance in the data set caused by survivors - a high prevlance means much lower specificity.

Conclusion

The results indicated a lot of interesting insights such as: 1) What are the most important features that contribute to a COVID related death which in this case were - age, intubation, pneumonia medical condition (Top 3 importance). 2) How the features were related to each other confirming the validity of the dataset such as gender influencing pregnant feature. ICU patients were more likely to be intubated and had pneumonia medical condition.

Overall, I enjoyed this project and the insights it had to offer. I am thankful for being able to take this course and keeping my evenings busy with something like this!