Project Title: Using Patient Data to Predict COVID-19 Contraction, Recovery and Mortality

COVID-19 Model Report

Author: Robert Ferdinand

Email: UOFLOUISIANA@GMAIL.COM

Project Title: Using Patient Data to Predict COVID-19 Contraction, Recovery and Mortality

Table of Contents

Project Title: Using Patient Data to Predict COVID-19 Contraction, Recovery and Mortality

Section 1: Introduction of Problem

Medical data on patients infected with the COVID-19 virus is collected from a hospital at an undisclosed location. The data consists of 23 features. Among the features that may appear to attribute to susceptibility to COVID-19 infection include pneumonia, age, diabetes, asthma, chronic obstructive pulmonary disease (COPD), asthma, hypertension, cardiovascular disorders, obseity and so on. Additionally the data on each patient has a target label 1 if the patient was infected by COVID-19, a target label 2 if the patient recovered from the COVID-19 and a target label 3 if the patient died from the COVID-19 infection. We can use this labeled data and classification techniques from Machine Learning (ML) and Deep Learning (DL) to attempt to predict the condition of a patient (1, 2 or 3) given the features from the data. Depending on what label the patient may be predicted as receiving, corresponding medical care will need to be planned by a medical facility equipped with handling COVID-19 outbreaks.

To be able to make such a prediction, one would need to analyze the data, to be able to “learn” from it. This can be done using Machine Learning (ML) and Deep Learning (DL) which are both Data Science tools. The PYTHON programming language and all of its excellent packages and modules will be fully utilized to accomplish the task at hand.

Both Machine Learning (ML) and Deep Learning (DL) will be carried out in an Apache Spark Machine Learning (ML) framework which facilitates parallel computation on the IBM Watson Cloud.

Section 2: The Data

The patient data is acquired from the open-source website Kaggle.com. This website has several sets of highly interesting and accurate data sets that have important real-world, societal applications. The patient data in particular contains information on features and labels of several patients obtained from a medical facility in an undisclosed location over a period of time during the COVID-19 pandemic. The features include parameters such as pneumonia, age, diabetes, asthma, chronic obstructive pulmonary disease (COPD), asthma, hypertension, cardiovascular disorders, obseity and so on. There are a total of 23 features in all. The data also contains a label or target column which indicates whether the patient contracted COVID-19 (1), recovered from COVID-19 (2) or died as a result of the infection (3).

The data and its features as well as the corresponding labels point in the direction of a multi-classification, supervised or labeled Machine Learning (ML) and/or Deep Learning (DL) project.

The data will need to be cleaned of any missing values and then removed of any duplicate records or rows. Also the data will need to be normalized so that Machine Learning (ML) and/or Deep Learning (DL) algorithms can work efficiently with it. Finally we will need to make sure that all of the data makes sense. For example the age of a patient would need to be an integer and so on.

We note here that the data is in a comma-separated version (CSV) file that will need to be read into a technology platform that can be ultimately used for ML and DL. The platform we will use for computation is the IBM Watson Cloud.

Section 3: The Methodology

In this section we use the data set described in the section above, from the Kaggle.com website. Then we proceed with learning from this data set as follows:

Step I: Data Cleaning

(a) Import data set from Kaggle and read the .CSV file as a pandas Data Frame.

(b) Strip column names of any extra white spaces to be able to correctly identify the names of the data features and labels.

(c) Drop feature columns including date of admission to medical facility, date of death, date of start of COVID-19 symptoms. Also drop patient ID feature column. This, since these features do not affect and hence will not be included in our machine learning and deep learning studies.

(d) Ensure that all columns are of data types that makes sense. For example the age of a patient should be an integer and so on and the feature as well as label codes are all integers.

(e) Using the PYTHON *value_counts()* function, to obtain the frequency distribution for each feature and label columns in the pandas Data Frame.

(f) From part (e) above, we first drop the feature columns that have a significant (50% or above) percentage of missing or insignificant data.

(g) From part (e) above, we clean column feature as well as labeled data by deleting corresponding rows or records that have missing values.

(h) As a results of (f) and (g) above, our data is reduced to about 560,000 rows with 14 feature columns and one label column. Even after cleaning the data, we have a very large sized data set and would need substantial computing resources for its processing.

Step II: Data Exploration

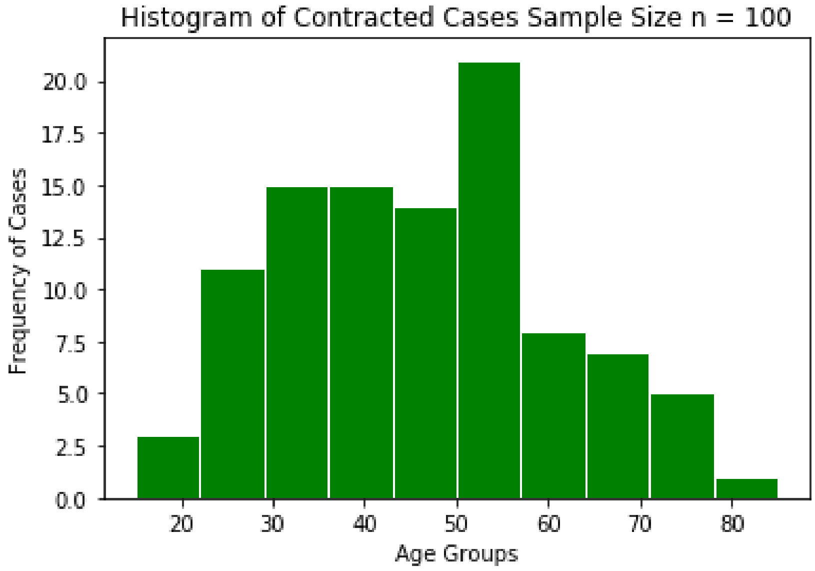
(a) Using the PYTHON SCIPY.STATS package we calculate the following statistics for each feature column. Note these statistics will NOT be computed for the target or label column: *minimum, maximum, mean (average), standard deviation, skewness and kurtosis*. A couple of points to be noted here are as follows:

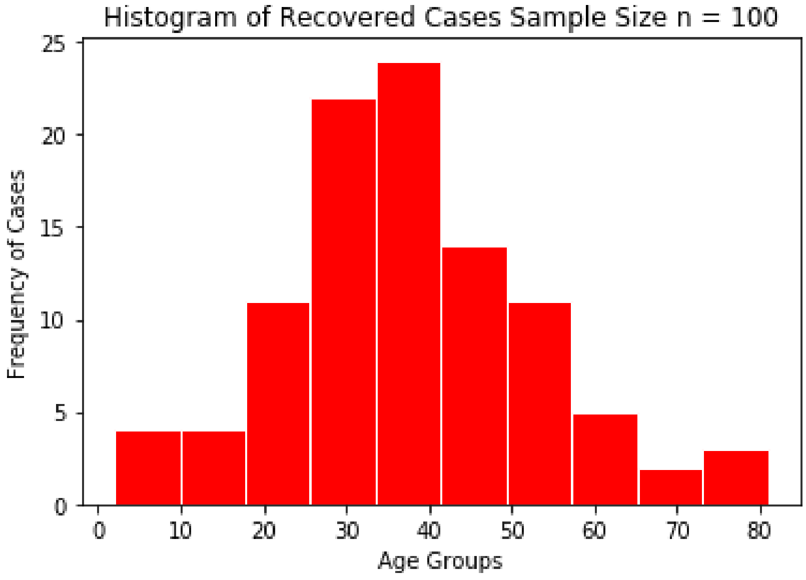
  i. All feature data are classification-based (for example in the pneumonia feature column, 0 would indicate that a patient does not have pneumonia while 1 would indicate a patient does have pneumonia). Hence we don't have any outliers in the feature columns.

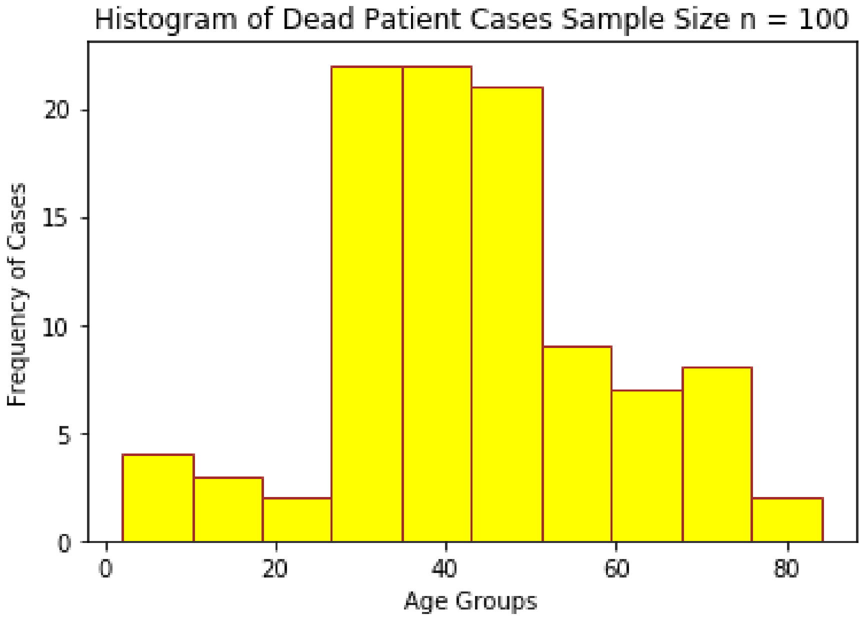
  ii. We get kurtosis statistical measure values greater than 10 for patient features such as pneumonia, COPD, chrnoic renal conditions, cardiovascular disorders and so on. This may indicate that these medical conditions may be highly correlated or causal to a patient contracting COVID-19.

(b) Using PYTHON's MATPLOTLIB plotting package and a random sample of n = 100 data points we create three histograms between patient age-groups having length 10 and their respective frequencies for the patients that contracted COVID-19, that recovered from COVID-19 and that succummed to COVID-19, respectively. The graphs are provided below. From the graphs, it may be observed that the highest numbers of patients with COVID-19 conditions come from 20

– 50 age range. This may suggest that COVID-19 is not prevalent in high numbers in younger children and seniors.

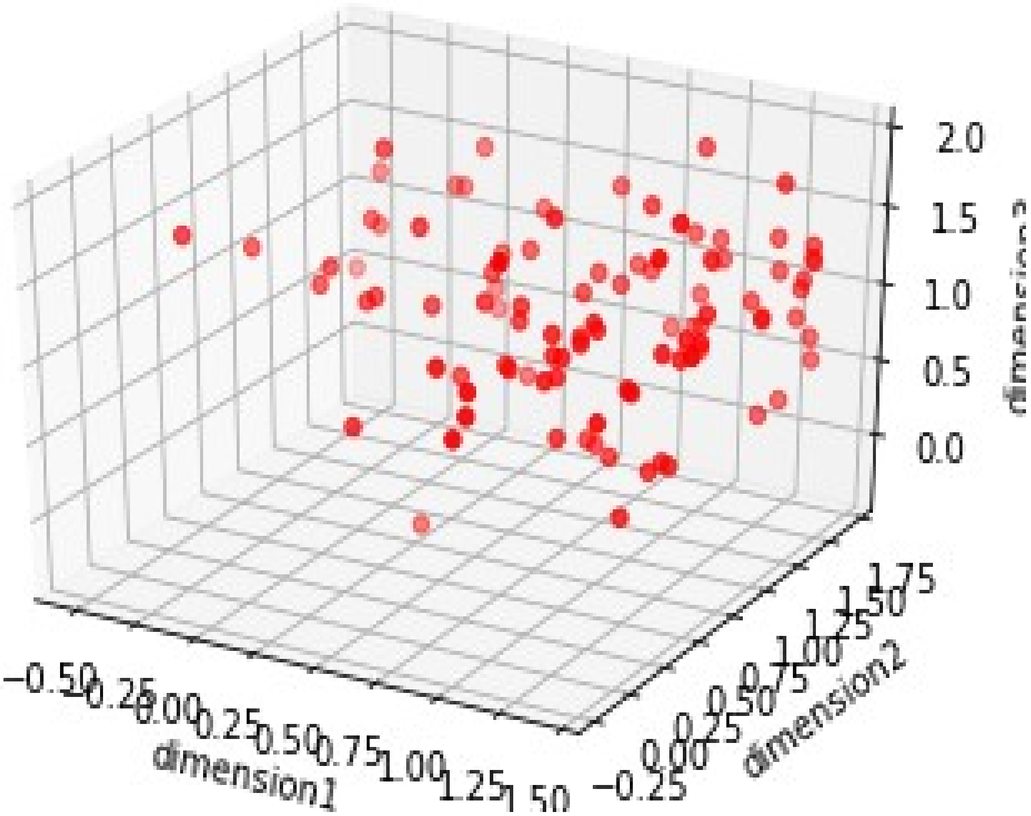






(c) Principal Component Analysis (PCA): Principal Component Analysis (PCA) or dimensionality reduction is performed on the feature columns of the data set. This reduces the dimension of the data set to k = 3. This is followed by a 3-D graph of the data set to detect any clustering or correlation among the data. The PCA algorithm is utilized. The three dimensional graph is provided below. From the graph, no evidence of major clustering of data may be observed:

Principal Component Analysis (PCA) Graph (k = 3)



Step III: Data Extraction, Transformation and Loading (ETL)

(a) First we note that none of the feature columns contains data that is textual. Hence the need for one-hot encoding the respective columns using the popular ONE-HOT ENCODING method is not necessary in this case. However, the usefulness of one-hot encoding does need to be mentioned for any Data Science project. Having stated that, ONE-HOT ENCODING will be used in the Keras Deep Learning Neural Network Models in Step IV below.

(b) Next, the feature columns named 'age' has numeric data ranging from age of 0 years to 85 years. Dividing the data by its maximum value, we bring the data to fit into the [0, 1] interval. This allows the machine learning and deep learning to perform with greater efficiency since it does not have to deal with high variation in the 'age' feature column.

(c) From above, our data should now be available and ready for the analysis that follows. One cannot emphasize the important of ETL any more, since it is gets the data ready for the modeling by the machine learning and deep learning techniques.

Step IV: Modeling

This is the most FUN part of the project. We apply the Machine Learning (ML) and Keras Deep Learning (DL) algorithms of Data Science to help us learn from the data set. Here we proceed as follows:

(a) Apache PySpark Machine Learning (ML) is installed on the IBM Watson Cloud. This takes a few minutes to install. The advantages of using a Spark Framework is that Apache Spark ML uses a parallel computing framework in its Machine and Deep Learning algorithms. This helps greatly with computing efficiency.

(b) The data set from Step III above is converted into an Apache Spark Data Frame. This to enable us to apply the available Apache Spark's Machine Learning (ML) and Keras Deep Learning (DL) algorithms to our model. Again converting the Pandas Data Frame to an Apache Spark Data Frame take a few minutes on the IBM Watson Cloud.

(c) The data set is randomly split into Training and Testing sets, in a 80% to 20% ratio. These Training and Testing sets will be used in the Machine Learning and Deep Learning methods that follow next.

(d) Machine Learning Model-1: Here we use a *Decision Tree Multi-Classifier* Machine Learning algorithm for the supervised or labeled problem. The pipeline method that comprises of a VectorAssembler and a Decision Tree Classifier is used. The model for prediction is created on the Training Data Set. A MultiClassClassificationEvaluator is used to measure the accuracy of the prediction of the model on the Testing Data Set. The metric used is ACCURACY that gives us the error of fitting the data to the Testing Data Set.

(e) Machine Learning Model-2: In this case we use a *Random Forest Classifier* Machine Learning algorithm for the supervised or labeled problem. The Random Forest Classifier method uses the Bootstrapping Algorithm a.k.a. BAGGING. In essence, a Random Forest Classifier uses a "forest" of several Decision Trees in its algorithm. The pipeline method that comprises of a VectorAssembler and a Random Forest Classifier is used. The model for prediction is created on the Training Data Set. A MultiClassClassificationEvaluator is used to measure the accuracy of the prediction of the model on the Testing Data Set. The metric used is ACCURACY that gives us the error of fitting the data on the Testing Data Set.

(f) Deep Learning Model-1: A Keras *Deep Feed Forward Neural Network (NN)* having *Tensor Flow* backend is used. A Keras *Sequential model* is used with one dense input layer having the RELU activation function, one dropout layer and output layer with function SOFTMAX. The optimizer is CROSSENTROPY. To be able to implement this we need to use the TO_CATEGORICAL function from Keras TensorFlow to convert the label column into three binary columns. This is akin to the ONE-HOT ENCODING process. The optimizer used is the successive gradient descent (SGD) and the measure metric is ACCURACY. One may note that the ADAM optimizer and SIGMOID output functions are also valid options that can be used here. Other accuracy measures that can be used are F-1 and SCORE. The prediction model is created on the Training Data Set and then evaluated on the Testing Data set to give us the accuracy score of the pediction created by the model.

(g) Deep Learning Model-2: A Keras *Deep Feed Forward Neural Network (NN)* having *Tensor Flow* backend is used. A Keras *Non-Sequential model* is used with dense input layers having the RELU activation functions and an output layer having the SIGMOID function. The optimizer used is CROSSENTROPY. To be able to implement this we need to use the TO_CATEGORICAL function from Keras TensorFlow to convert the label column into three binary columns. This is akin to the ONE-HOT ENCODING process. The optimizer used is the successive gradient descent (SGD) and the measure metric is ACCURACY. One may note that the ADAM optimizer and SIGMOID output functions are also valid options that can be used here. Other accuracy measures that can be used are F-1 and SCORE. The prediction model is created on the Training Data Set and then evaluated on the Testing Data set to give us the accuracy score of the pediction created by the model.

Project Title: Using Patient Data to Predict COVID-19 Contraction, Recovery and Mortality

Section 4: The Results

In this section we discuss results from Section 3 above. The analytical and graphical results can be found in the IPYTHON notebook on Github. URL: https://github.com/DataScienceEsq/PROJECT-3.

(a) Statistical and graphical analysis of the data were discussed earlier in Section 3, Step II.

(b) Machine Learning Model-1 gives us about a 60% accuracy score. Hence the data fits the model only fairly well.

(c) Machine Learning Model-2 also give us about a 60% accuracy. There is insignificant improvement here compared to the Machine Learning Model-1.

(d) Sequential Deep Learning Model-1 presents a Loss Score of 0.9 and an Accuracy Score of about 0.65 or 65% which is again only a fair accuracy value.

(e) Non-sequential Deep Learning Model-2 presents a Loss Score of about 0.8 and Accuary Score of about 0.7 or 70%. It is an improvement from the other models but the accuracy score is not exceptional here either.

Section 5: Discussion and Conclusion

Pandas DataFrames, Apache Spark Machine Learning and Keras Deep Learning Neural Networks were used extensively in this project. The implementation was carried out on a PYTHON notebook on IBM Watson Studio platform which proves to be an outstanding resource.

On the whole, fitting the model to the data having close to half a million data points only gave us fair results using the Apache Spark Machine Learning or the Deep Learning Algorithms. In the Deep Learning Algorithms, the Non-Sequential Model was slightly more accurate than the Sequential Model. From the results it may appear that the patient features of medical conditions may not have a very significant causal effect as regards the labels for the COVID-19 virus.

What may be more helpful would be data on patient subjects who contacted COVID-19 and did not contact COVID-19 and feature information on the subjects. The features could be the same features as in this project or additional features could be added or dropped. This data set may give a better prediction on what patient feature conditions can be more causal towards a patient contracting the COVID-19 virus.

In conclusion we have used Machine Learning (ML) and Deep Learning (DL) techniques, extensively to study patient data with several features and its causal affect on the patient contracting, recovering and dying from the COVID-19 virus. The results or the modeling have been only fairly accurate with maximum accuracy achieved of about 70%.

Acknowledgments: The author would like to acknowledge the FREE computing resources provided by the IBM Watson Cloud platform.