Using Random Forest Classifier to Forecast Infections and Fatalities Due to COVID-19 per Country

Michael Morton   
Computer Science  
Georgia State UniversityAtlanta, Georgia, USA  
mmorton11@student.gsu.edu

*Abstract*—COVID-19 is a worldwide epidemic that is changing the way of life as we know it. By using datasets from the first three weeks of the worldwide outbreak, I was able to implement a random forest classification model. After training the model, I was able to predict the number of confirmed cases and the number of fatalities in a uniform range of dates. My first model used latitude and longitude coordinates and the date (represented as an integer) as input for training and testing. The results of this model were not very consistent. The Week 1 data produced a mean number of confirmed cases of 916 and mean number of fatalities of 39; each of these values had a standard deviation of 5662 and 357, respectively. The Week 2 data produced a mean number of confirmed cases of 1183 and mean number of fatalities of 52; each of these values had a standard deviation of 6847 and 473, respectively. The Week 3 data produced a mean number of confirmed cases of 2133 and mean number of fatalities of 103; each of these values had a standard deviation of 10247 and 801, respectively.

For my second model, I changed the date input to the number of days since the initial date in the test data. These results were more consistent than the first model. The Week 1 data produced a mean number of confirmed cases of 1196 and mean number of fatalities of 52; each of these values had a standard deviation of 6357 and 430, respectively. The Week 2 data produced a mean number of confirmed cases of 1900 and mean number of fatalities of 89; each of these values had a standard deviation of 9104 and 688, respectively. The Week 3 data produced a mean number of confirmed cases of 2133 and mean number of fatalities of 103; each of these values had a standard deviation of 10247 and 801, respectively. Although this model performed better than the first model, it seems that random forest may not be best forecasting method for rapidly spreading epidemics.

Keywords—computer science, data mining, COVID-19, random forest, machine learning

# Introduction

The COVID-19 virus has drastically changed the everyday life of all citizens of the world. The severity and ease of spread of the virus have turned COVID-19 into a worldwide epidemic. The virus is a respiratory illness that can be spread from person to person. It is caused by a novel coronavirus that was first identified in Wuhan, China. COVID-19 has multiple symptoms, but the main symptoms include fever, cough, and shortness of breath. The virus is so dangerous due to the complications that it can cause in the host’s body. These complications can include pneumonia in both lungs, multi-organ failure, and even death [1]. This paper will use random forest in order to predict the number of COVID-19 infections in multiple countries and regions, as well as the number of fatalities caused by these infections.

Random forest is a machine learning algorithm created by Leo Breiman in order to improve the classification of diverse data using random sampling and attribute selection [2]. The random forest classifier algorithm can be found in the Python Scikit library. The algorithm produces a model made of multiple decision trees that each vote on the overall classification of a given set of inputs. Some of the major features associated with the random forest algorithm include: random sampling of data points when building tress, random subsets of features considered when splitting nodes, a sub-sample size that is always the same as the original input size, and samples that are drawn with replacement [3].

# Previous Reseach

Multiple types of machine learning models have been used to predict the outcomes of pandemics and highly infectious diseases. A group of researchers from Virginia produced a Dirichlet process model for classifying and forecasting influenza epidemic curves [4]. A group of scientists from China have already created a prediction model for the spread of COVID-19 in China. In their research, they used population migration data before and after the onset of COVID-19, along with COVID-19epidemiological data, to create an epidemic curve using a Susceptible-Exposed-Infectious-Removed (SEIR) model. The research team also used an artificial intelligence (AI) approach, trained on the 2003 SARS data, to predict the epidemic [5]. While this approach yielded favorable results, it was very complicated, and I wanted to implement a machine learning model that I had not seen used in COVID-19 research yet. Reading various papers on predicting epidemic curves, I found another group of Chinese researchers who were forecasting the African swine fever outbreak. In their research, they used African swine fever outbreak data and the WorldClim database meteorological data. They then selected the CfsSubset Evaluator-Best First feature selection method combined with the random forest algorithms to construct an African swine fever outbreak prediction model [6]. They were able to achieve accuracy higher than any previous models with this implementation. Looking at their research and the associated accuracies of their predictions, I decided to implement the random forest algorithm on COVID-19 epidemiological data.

# Methods

The first step in my research was finding a suitable dataset. While looking on Kaggle, I found three weekly competitions that were focused on forecasting the global infections and fatalities associated with COVID-19. Each competition focused on the data that was available for that current week. After compiling the dataset from each competition, I was left with three folders, each containing three files. The first folder was Covid-19 Global Forecasting Week 1, and it contained a training dataset (Train01), a testing dataset (Test01), and a dataset used to format the results of the test set (Submission01). The second folder was Covid-19 Global Forecasting Week 2. This folder also contained a training dataset (Train02), a testing dataset (Test02), and a results dataset (Submission02). The final folder was Covid-19 Global Forecasting Week 3, and similarly to the other two folders, it contained a training dataset (Train03), a testing dataset (Test03), and a results dataset (Submission03).

Before any tested could be carried out the dataset, I had to do a lot of data preprocessing. In order for the datasets from the different weeks to be comparable with one another, they all had to be put into the same format. The Train01 dataset contains eight columns: Id, Province\_State, County\_Region, Latitude, Longitude, Date, Confirmed\_Cases, and Fatalities. I decided that latitude, longitude, and date would be the best input variables to distinguish a country or region. The only problem with this was that datasets Train02 and Train03 did not contain the latitude and longitude for their regions. This meant I had to go through and add the latitude and longitude coordinates for each entry in each dataset. While preforming this tedious task, I also deleted any entries of countries or regions that were not present in the Train01 dataset. Next, I had to change the format of all the dates in each data set. The dates were originally formatted as a string (##/##/####), but I needed them as integers in order to run random forest classifier that I wanted to use. To accomplish this, I simply removed the “/” in the strings, turning the strings into integers. The final set of data preprocessing was to make sure that there were no null values in the training sets. I noticed that the only place I found null values was in the Province\_State column. This was because not all entries had a province or state associated with them; some were just countries. Since I was using latitude and longitude coordinates to distinguish the location of the entry, the easiest way of removing all the null values was by simply deleting the Province\_State column in the training data sets. I decided to leave the Province\_State column in the testing datasets so I could go back and see with what province or state a result was associated. My datasets were now preprocessed, and I was ready to begin implementing my random forest model.

The random forest model I decided to use is RandomForestClassifier found the in the SciKit Python library. This class allowed me to alter any aspects of the random forest model in order to make them fit my datasets. The three main variables that I focused on when finding that optimal configuration for my data were the max\_depth, the max\_features, and the random\_state. The max\_depth designates the maximum node depth for each tree in the forest. Through trial and error, I found that the optimal value for max\_depth was 200. The next variable I focused on was the max\_features. The max\_features is the number of features that each tree looks at when considering the best splits. The max\_features is determined by the square root of the number of features that are being added to the trees. Since I determined that I would be using four features in two iteration for each dataset, the max\_features would be two for each iteration and four total for each data set. The last variable I considered was the random\_state. The random\_state controls both the randomness of the bootstrapping of the samples used when building trees and the sampling of the features to consider when looking for the best split at each node. By assigning an integer to random\_state, you can make a randomized algorithm deterministic; this means that running the model multiple times with the same dataset will produce the same result. Now that the model was set up, it was time to start running the model on the datasets and begin producing results.

The first step necessary for producing results was to use the random forest classifier to train the number of confirmed cases. For training, the inputs (x) were the latitude and longitude coordinates and the date for each data point in the training datasets. The outputs (y) were the number of confirmed cases associated with that data point. After the model was trained for the number of confirmed cases, I was able to use the testing datasets to produce the predicted number of confirmed cases. The inputs (x) were once again the latitude and longitude coordinates and the date for each data point. The outputs (y) were now the predicted number of confirmed cases associated with that data point. I created a new column named Confirmed\_Cases\_prediction in the results dataset and stored the values there. Next, I had to use the random forest classifier to train the number of fatalities. For this training, the inputs (x) were once again the latitude and longitude coordinates and the date for each data point in the training datasets, but the outputs (y) were the number of fatalities associated with that data point. After the model was trained for the number fatalities, I was able to use the testing datasets to produce the predicted number of deaths. The inputs (x) were once again the latitude and longitude coordinates and the date for each data point. The outputs (y) were now the predicted number of deaths associated with that data point. I created a new column named Death\_prediction in the results dataset and stored the values there. Lastly, I added an Id column to the results dataset. This column showed an Id that corresponded to the entry’s Id in the testing datasets. The process was carried out for each of the three weeks of datasets. I finally used the .describe command to show the statistical representations of each week’s results dataset.

# Results

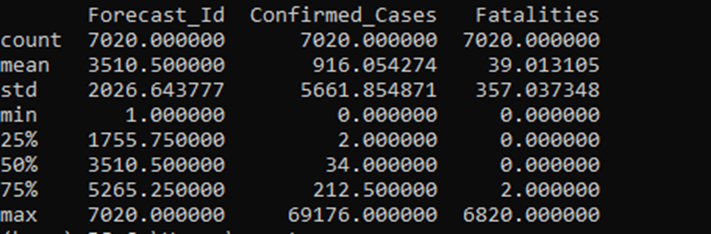


Figure 1: Week 1: Train01 vs Test01



Figure 2: Week 2: Train02 vs Test02



Figure 3: Week 3: Train03 vs Test03

The results produced by this model were not very consistent. Figure 1 shows that the Week 1 data produced a mean number of confirmed cases of 916 and mean number of fatalities of 39; each of these values had a standard deviation of 5662 and 357, respectively. Figure 2 shows that the Week 2 data produced a mean number of confirmed cases of 1183 and mean number of fatalities of 52; each of these values had a standard deviation of 6847 and 473, respectively. Figure 3 shows that the Week 3 data produced a mean number of confirmed cases of 2133 and mean number of fatalities of 103; each of these values had a standard deviation of 10247 and 801, respectively. I was not satisfied with these results, so I decided to create a new model.

# Methods (Model 2)

For my second model, I decided to change the way that the date was represented in the input for both the training and testing datasets. I changed the date from being an integer representation of itself to an integer representing the number of days since the initial date in the test dataset. The new datasets were named train1 and test1 for the Week 1 data, train2 and test2 for the Week 2 data, and train3 and test3 for the Week 3 data. My testing procedures remained the same as the first model. Since the changed date column retained the same column name, there was no need to change how the random forest classifier was implemented.

# Results (Model 2)

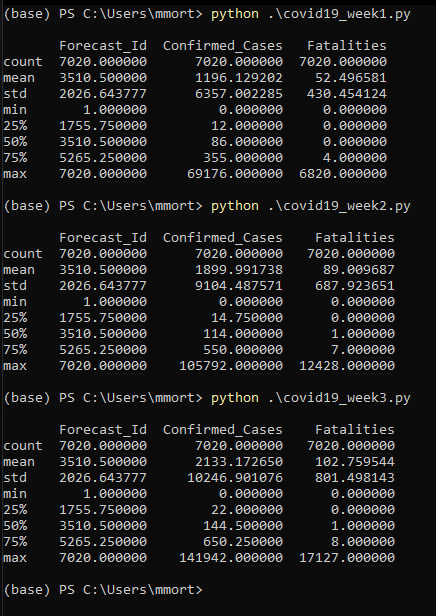


Figure 4: Week 1: Train1 vs Test1

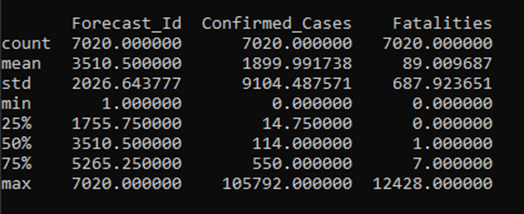


Figure 5: Week 2: Train2 vs Test2



Figure 6: Week 3: Train3 vs Test3

The results produced by the second model were more consistent than the results from the first model. Figure 4 shows that the Week 1 data produced a mean number of confirmed cases of 1196 and mean number of fatalities of 52; each of these values had a standard deviation of 6357 and 430, respectively. Figure 5 shows that the Week 2 data produced a mean number of confirmed cases of 1900 and mean number of fatalities of 89; each of these values had a standard deviation of 9104 and 688, respectively. Figure 6 shows that the Week 3 data produced a mean number of confirmed cases of 2133 and mean number of fatalities of 103; each of these values had a standard deviation of 10247 and 801, respectively.

# Conclusion

While the second model produced more consistent results than the first model, the results are not consistent enough to be considered accurate. Other factors such as country population size, population density, geographic location, and economic resources may have to be considered in order to produce a more realistic and accurate model. It may also be the case that the random forest classification may not be the best suited model to use for an epidemic that is spreading so rapidly. Overall, this research demonstrates how to implement a random forest classification model on multiple datasets in order to compare the consistency of the results. It may not produce the most consistent results, but it will allow others to take other factors into account and possibly produce a more accurate model in the future.

#### 

##### References

1. Asim M, Sathian B, van Teijlingen E, Mekkodathil A, Subramanya SH, Simkhada P. “COVID-19 Pandemic: Public Health Implications in Nepal” . Nepal J Epidemiol. 2020;10(1); 817-820. DOI: 10.3126/nje.v10i1.28269
2. Frederick Livingston. “Implementation of Breiman’s Random Forest Machine Learning Algorithm”. Machine Learning Journal Paper. 2005.
3. Breiman, Leo and Adele Cutler. “Random Forests”. Machine Learning Journal. 2003.
4. Nsoesie et al.: “A Dirichlet process model for classifying and forecasting epidemic curves”. BMC Infectious Diseases 2014 14:12.
5. Yang Z, Zeng Z, Wang K, et al. “Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions”. *J Thorac Dis*. 2020;12(3):165–174. doi:10.21037/jtd.2020.02.64
6. Liang R, Lu Y, Qu X, et al. “Prediction for global African swine fever outbreaks based on a combination of random forest algorithms and meteorological data”. *Transboundary & Emerging Diseases*. 2020;67(2):935-946. doi:10.1111/tbed.13424