Using Random Forest Regression to Forcast Infections and Fatalities Due to COVID-19 in the Southeast United States

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. From using the “US counties COVID 19 dataset” on Kaggle with information about population and area, I created a data for counties in the Southeast United States. Using this dataset along with random forest regression and varying inputs, I was able to predict the number of COVID-19 cases and the number of deaths due to COVID-19 in each county. The accuracies for the predicted number of cases in a county ranged from 94.63% - 95.36% and the accuracies for the predicted number of deaths in a county ranged from 95.25% - 97.81%. This shows that random forest regression is an acceptable metric for predicting the numbers of COVID-19 cases and deaths for a given county. I created a second dataset with data from the CDC’s “United States COVID-19 Cases and Deaths by State over Time” and Google’s COVID-19 Community Mobility Reports. This dataset compared the statewide number of total cases, new cases, total deaths, and new deaths to the metrics tracked by Google’s COVID-19 Community Mobility Reports for each state in the Southeast United States. Using the same testing methods from the first dataset, I created models to predict the total number of COVID-19 cases in a state and the total number of deaths due to COVID-19. The accuracies for the predicted number of total cases in a state ranged from 85.81% - 89.95% and the accuracies for the predicted number of total deaths in a state ranged from 88.31% - 93.24%. This once again shows that random forest regression is an acceptable method for this type of prediction. The lower accuracies in the Southeast state database compared to the Southeast county may show that the accuracies are closely tied to the number of entries in a dataset.

Keywords—computer science, COVID-19, random forest, machine learning, artificial intelligence

# 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 experiment will use random forest to predict the number of COVID-19 infections in multiple counties in states in the United States of America, as well as the number of fatalities caused by these infections.

Random forest is a machine learning algorithm created by Leo Breiman to improve the classification or regression of diverse data using random sampling and attribute selection [2]. The random forest regressor 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 prediction of a given set of inputs. These predictions are then all averaged, and this value is given as the output of the model. This is represented in Figure 1 below. Some of the major features associated with the random forest algorithm include random sampling of data points when building trees, random subsets of features considered when splitting nodes, a sub-sample size that is always the same as the original input size, samples that are drawn with replacement (i.e., bootstrapping), and trees run in parallel and have no interaction with one another [3].

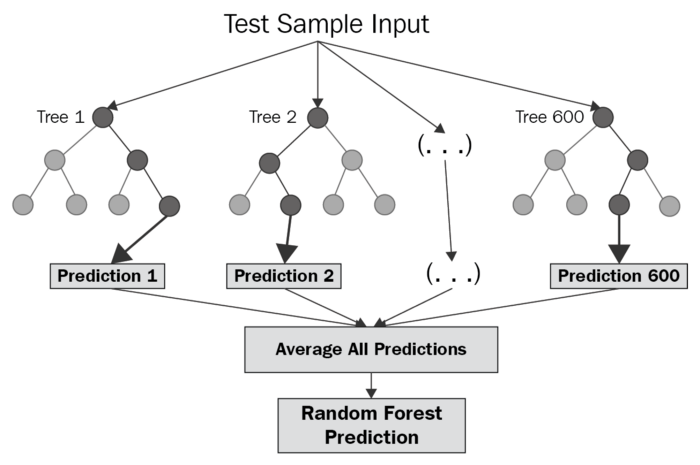


Fig 1. Visual representation of random forest algorithm

# 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.

Last semester, I ran an experiment using the first three weeks of COVID-19 epidemiological data that were available on Kaggle. In that experiment, I ran two different models. The first model used an integer representation of the date that the data was produced. The second model represented the date as the integer number of days since the initial date in the test dataset. The second model performed better than the first model, but neither model was very accurate. I want to build on this experiment, using what I learned previously, in order to create a more accurate model that predicts the number of cases and deaths solely within the United States, rather than using data from countries all over the world.

# Methods

While my previous models were used to make predictions on a worldwide, country by country scale, this model only focuses on cases of COVID-19 within the United States. The first step of my research was finding a suitable dataset. While looking on Kaggle, I found a dataset called “US counties COVID 19 dataset.” This dataset is a .csv file that shows the number of COVID-19 cases and deaths as reported by the New York Times at the state and county level. The original dataset consists of six columns: date, county, state, FIPS, cases, and deaths. The date column contains the date of the reported information in the format of year-month-day (#### - ## - ##). The county and state columns simply contain the name of the county and state where data is collected from. The FIPS column contains a specific code the identifies unique geographic areas. These codes may identify counties or metropolitan areas within counties. Finally, the cases and deaths columns provide the number of COVID-19 cases and deaths for the specific county and date. While this information was a good start, I wanted to add more data that would allow the algorithm to find more similarities and differences in counties, and I also wanted to change how some of the data was represented. After realizing how long it would take to add all of this information for each county in each state, I decided to narrow my research to counties within states in the Southeast United States. For the purposes of this experiment, I defined the Southeast United States as Alabama, Florida, Georgia, Kentucky, Mississippi, North Carolina, South Carolina, Tennessee, and Virginia. These nine states all have the same seasons and, on average, the same weather conditions. From these nine states, my dataset set has a total of 869 counties which create 222,655 data entries.

The next step of my research was preprocessing my newly created dataset. The first step was changing how the date was reported. Since all the data was from the same calendar year, I decided to designate January 1, 2020 as day 1 and then iterated each preceding date following day 1. For example, January 2, 2020 is day 2 and July 1, 2020 is day 182. The next step was adding the population and area, in square miles, to each data entry. This step was extremely time consuming and took almost a week to complete. From the new columns that contain population and area, I was able to create a population density column by dividing the population of the county by the area of the county. The only other change that I made to the dataset was deleting the state and county names from the dataset that was used from actual testing. I determined that the FIPS code was a better identifier to use for distinguishing the county and state associated with each data entry. I kept a copy of the dataset with states and counties to be able to easily lookup FIPS codes. My data was now preprocessed, and I was ready to begin implementing my random forest model.

The random forest model I decided to use is RandomForestRegressor 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 two aspects of the model that I altered were n\_estimators and random\_state. The number of estimators controls the number of trees in the forest and random state control 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 value to random state, the randomized portion of the algorithm becomes deterministic, allowing for results to be reproduced. My test and training sets were created using train\_test\_split. For each initial run, the x variable was assigned to all non-object data sources aside from the desired output, which was assigned to y. I used a training set consisting of 80% of the data while the test set contained the other 20% of the data. The x and y training set were then fit to the RandomForestRegressor model in order to train the model. Using regressor.predict on the x test set I used the model to create a predicted output. This output was assigned to y\_predict and was used later in statistical calculations verses the y test data set, which are the actual values. The next feature that I implemented was regressor.feature\_importances\_, which returns a decimal value of how important each feature was in creating the split for the tress. I would use this importance metric later when deciding which features to use in order to optimize my models. The results of the model were then added to a .csv spreadsheet that contains the testing data, the predicted outcomes, and the actual outcomes. The last step was creating a way to determine how well the model performed. I decided to use a statistical representation of performance. The metrics that I used were the explained variance score, the R squared value, the mean absolute value, and the accuracy. The values show how much the predicted values differ from the actual values. Now that the model was set up, it was time to start running the model on the datasets and begin producing results.

After running my model a few times to either predict the number of COVID-19 cases or the number of deaths caused by the virus, I was able to present my initial findings with my Artificial Intelligence professor and classmates. When my presentation was finished, I was given a link by Dr. Zhang that he thought may be helpful to my research. The link contained Google’s COVID-19 Community Mobility Reports. These reports showed how community mobility has changed day to day compared to a baseline value for each category. The categories in the reports include retail and recreation, grocery and pharmacy, parks, transit stations, workplaces, and residential. I was planning on adding all of this new data to my original data set, but there were just too many missing values for certain categories and certain counties. Although I would not incorporate the data into my original dataset, I decided to create a new dataset that was based on my nine previously defined Southeast states, rather than the counties present in each state. I found a data set from the Centers of Disease Control (CDC) called “United States COVID-19 Cases and Deaths by State over Time.” From this dataset, I was able to obtain the total number of cases, number of new cases, total number of deaths, and number of new deaths for each state in the Southeast. I then took the statewide data from Google’s COVID-19 Community Mobility Reports to create a new dataset that contained the number of statewide cases, new cases, deaths, new deaths, percent change in retail and recreation, percent change in grocery and pharmacy, percent change in parks, percent change in transit stations, percent change in workplaces, and percent change in residential. I then changed the date for each entry to the same days from January 1, 2020 that I used in my previous dataset and added a column that contained the FIPS code for each state. My state-based dataset was now complete and ready for testing. I used the same model as described above for testing this dataset.

##### IV. Results

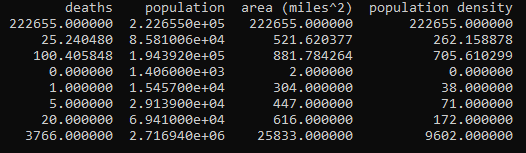
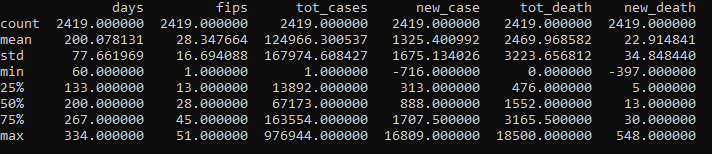


Fig 2: Summary of Southeast Counties Dataset (spilt for size) 

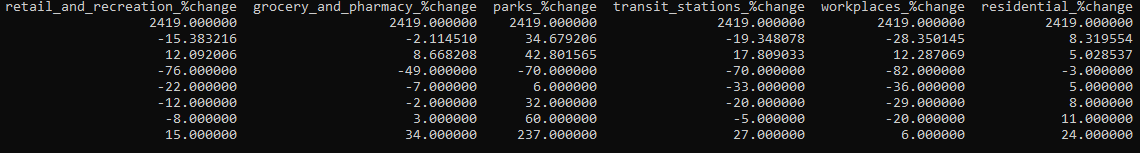


Fig 3: Summary of Southeast States Dataset (spilt for size)

Figures 2 and 3 shows summary of my datasets used in this research project. Figure 2 shows the statistical summation of the categories used in the Southeast counties data set, while Figure 3 shows these summations for the Southeast states dataset.

**Test 1**: Southeast Counties Number of Cases

**Case 1**: X = {days, fips, deaths, population, area, population density} Y ={cases}

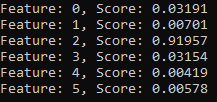


Fig 4: Test 1 Case 1 Importance of Features: Feature 0 = days, 1 = fips, 2 = deaths, 3 = population, 4 = area, 5 = population density

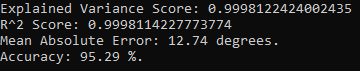


Fig. 5: Test 1 Case 1 Statistical Results

Case 1 was used to get a baseline for the performance of the model for predicting the number of cases of COVID-19 for a given county. The only value removed for the input dataset was cases, which is the output. As seen in Figure 4, feature 2, deaths, was the most important feature in predicting the output. The results in Figure 5 show that the predicted data was very highly associated with actual data and that the predicted data was accurate 95.29% of the time.

**Case 2**: X = {days, fips, population, area, population density} Y ={cases}

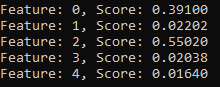


Fig 6: Test 1 Case 2 Importance of Features: Feature 0 = days, 1 = fips, 2 = population, 3 = area, 4 = population density

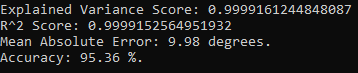


Fig 7: Test 1 Case 2 Statistical Results

Since Case 1 showed how much the category deaths was affect the tree splits, I decided to removed deaths from the input dataset and run the model again. As seen in Figure 6, population followed by date were now the two most important factor for predicting the number of cases of COVID-19 for a given county. This makes perfect sense due to the way that the virus is transmitted. Areas with higher population would generally have a higher number of cases, making population size a good indicator to the total number of cases in a county. The results in Figure 7 show that the predicted data was once again highly associated with the actual data and that the predicted data was accurate 95.36% of the time, a slight increase over Case 1.

**Case 3**: X = {days, fips, deaths, population} Y ={cases}

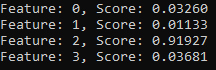


Fig 8: Test 1 Case 3 Importance of Features: Feature 0 = days, 1 = fips, 2 = deaths, 3 = population

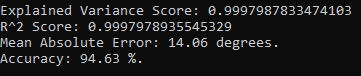


Fig 9: Test 1 Case 3 Statistical Results

From the information gained on the importance of features from Cases 1 and 2, I determined that the number of deaths and the population size were the two most important features. For Case 3 I decided to use only those two features, along with date and fips for identification, to test the model. As seen in Figure 8, it was clear that the number of deaths was still the most important factor when predicting number of cases of COVID-19 for each county. The results in Figure 8 show that, as expected, the predicted data was highly associated with the actual data, but the accuracy of the predicted data actually fell to 94.36%. This may show that only having two categories to base a prediction on is not enough for the model to retain its accuracy. I ran multiple cases after Case 3, but none of those cases showed an increase in accuracy over Case1 or Case 2.

**Test 2**: Southeast Counties Number of Deaths

**Case 1**: X = {days, fips, cases, population, area, population density} Y ={deaths}

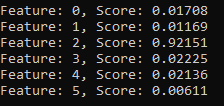


Fig 9: Test 2 Case 1 Importance of Features: Feature 0 = days, 1 = fips, 2 = cases, 3 = population, 4 = area, 5 = population density

|  |
| --- |
| Calculated Accuracy: 95.26% |
| Calculated Rounded Accuracy: 96.54% |

Fig 10: Test 2 Case 1 Statistical Results

Case 1 for Test 2 used the same logical as Case 1 for Test 1, I wanted to create a baseline predicting the number of deaths. It is clear from Figure 9 that the number of cases of COVID-19 was the most important feature when making splits in trees in order to predict the number of deaths from COVID-19 in each county. The results for Test 2 were not as easy to calculate as the results for Test 1. Since there are some counties that do not have any deaths associated with the virus, I could not use the same method for calculating accuracy (you cannot divide by zero). Instead, I did my accuracy calculations within Excel. To get around having to divide by zero, I removed any data where the actual number of deaths were zero. I then calculated the percent error for row of data. From the percent error, I was easily able to calculate accuracy. Taking the average accuracies of each row of data is shown as Calculated Accuracy in Figure 10. I then thought about it and realized that having decimal number for the predicted number of deaths did not make sense; a person is either alive or dead. I then rounded the predicted number of deaths to the nearest integer. Then I carried out the same calculations again to find the accuracy. The accuracy calculated with the rounded predicted number of deaths is shown in Figure 10 as Calculated Rounded Accuracy. The results in Figure 10 show that predicted number of deaths was very highly associated with the actual number of deaths and that the predicted number of deaths was accurate 96.54% of the time.

**Case 2**: X = {days, fips, population, area, population density} Y ={deaths}

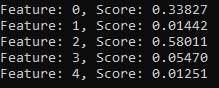


Fig 11: Test 2 Case 2 Importance of Features: Feature 0 = days, 1 = fips, 2 = population, 3 = area, 4 = population density

|  |
| --- |
| Calculated Accuracy: 97.09% |
| Calculated Rounded Accuracy: 97.81% |

Fig 12: Test 2 Case 2 Statistical Results

Since the importance of the feature cases was so much higher than the other features, I decided to remove it and test the model to see what other features are important. As shown in Figure 11, the most important feature is population, and the second most important feature is the date. This makes perfect sense for the reasons listed in Test 1 Case 2. The results in Figure 12 show that predicted number of deaths in Case 2 is more closely associated with the actual number of deaths than those of Case 1. The results also show that the accuracy is 97.81%, which is slightly higher than Case 1.

**Case 3**: X = {days, fips, cases, population} Y ={deaths}

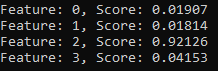


Fig 13: Test 2 Case 3 Importance of Features: Feature 0 = days, 1 = fips, 2 = cases, 3 = population

|  |
| --- |
| Calculated Accuracy: 93.76% |
| Calculated Rounded Accuracy: 95.25% |

Fig 14: Test 2 Case 3 Statistical Results

Following my logic from Test 1 Case 3, I used the two most important factors, cases and population, for Case 3 in Test 2. Figure 13 shows that the number of COVID-19 cases is the most important factor in this iteration of the model. The results in Figure 14 show that the predicted number of deaths is once again highly associated with the number of actual deaths and that the model was able to predict the correct number of deaths 95.25% of the time. This accuracy fell once again, just like Case 3 in Test 1, and I believe this is due to not having enough features to make the best prediction possible.

**Test 3**: Southeast States Number of Total Cases

**Case 1**: X = {days, fips, new\_case, tot\_death, new\_death, retail\_and\_recreation\_%change, grocery\_and\_pharmacy\_ %change, parks\_%change, transit\_stations\_%change, workplaces\_%change, residential\_%change} Y ={tot\_cases}

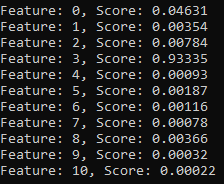


Fig 15: Test 3 Case 1 Important Features: Feature 0 = days, 1 = fips, 2 = new\_case, 3 = tot\_death, 4 = new\_death, 5 = retail\_ and\_recreation\_%change, 6 = grocery\_and\_pharmacy\_ %change, 7 = parks\_%change, 8 = transit\_stations\_%change, 9 = workplaces\_%change, 10 = residential\_%change

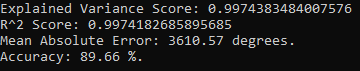


Fig 16: Test 3 Case 1 Statistical Results

Test 3 was my first test using the Southeast sates dataset, so I wanted to create a baseline on how well the dataset worked for predicting the number of COVID-19 cases in a state from all of the other input information. By looking at Figure 15, it is obvious that the total number of deaths due to COVID-19 was the most important feature in predicting the number of total cases. The results in Figure 16 show that the number of predicted cases is highly associated with the number of actual total cases, but this association is slightly lower than those of the previous dataset. The model was able to predict the total number of cases with an accuracy of 89.66%, which is slightly lower than the first dataset. I believe that the differences in accuracies between the two datasets can be explained by the number of entries in each dataset; the Southeast counties dataset has 222,655 entries while the Southeast states dataset only has 2,419 entries.

**Case 2**: X = {days, fips, new\_case, new\_death, retail\_and\_recreation\_%change, grocery\_and\_pharmacy\_ %change, parks\_%change, transit\_stations\_%change, workplaces\_%change, residential\_%change} Y ={tot\_cases}

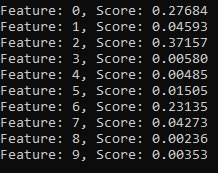


Fig 17: Test 3 Case 2 Important Features: Feature 0 = days, 1 = fips, 2 = new\_case, 3 = new\_death, 4 = retail\_ and\_recreation\_%change, 5 = grocery\_and\_pharmacy\_ %change, 6 = parks\_%change, 7 = transit\_stations\_%change, 8 = workplaces\_%change, 9 = residential\_%change

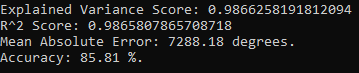


Fig 18: Test 3 Case 2 Statistical Results

From Figure 15 showing how important the number of total deaths were as a feature in Test 3 Case 1, I decided to remove the feature for Case 2 in order to see what other features were the most important. From Figure 17, it is clear that the number of new COVID-19 cases, the date, and the precent change in the number of parks were the most important features, respectively. The results in Figure 18 are all slightly lower than those results for Test 3 Case 1, showing that the total number of deaths is more important were there are less data points. The accuracy of the model dropped to 85.81%, which is the lowest it has been in any case so far.

**Case 3**: X = {days, fips, new\_case, tot\_death, parks\_%change} Y ={tot\_cases}

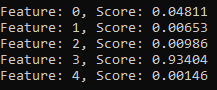


Fig 19: Test 3 Case 3 Important Features: Feature 0 = days, 1 = fips, 2 = new\_case, 3 = tot\_death, 4 = parks\_%change

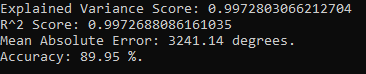


Fig 20: Test 3 Case 3 Statistical Results

The goal of Test 3 Case 3 was to take all the information learned in Cases 1 and 2 to create an optimized model for predicting the total number of COVID-19 cases for a given state. I was able to accomplish this by taking the most important features from the two previous cases and using them as the input (I also included fips for identification purposes). Overall, this tactic was successful. Figure 19 shows that the number of total deaths was the most important feature, which was already expected. The results in Figure 20 show the best statistical results for any test using this dataset. The association between the expected and predicted number of total cases is very high, and the accuracy improved to 89.95%. While this is larger than the other cases using this dataset, it is still lower than the results using the Southeast counties dataset.

**Test 4**: Southeast States Number of Total Deaths

**Case 1**: X = {days, fips, tot\_cases, new\_case, new\_death, retail\_and\_recreation\_%change, grocery\_and\_pharmacy\_ %change, parks\_%change, transit\_stations\_%change, workplaces\_%change, residential\_%change} Y ={tot\_death}

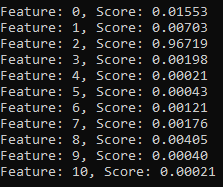


Fig 21: Test 4 Case 1 Important Features: Feature 0 = days, 1 = fips, 2 = tot\_cases, 3 = new\_case, 4 = new\_death, 5 = retail\_ and\_recreation\_%change, 6 = grocery\_and\_pharmacy\_ %change, 7 = parks\_%change, 8 = transit\_stations\_%change, 9 = workplaces\_%change, 10 = residential\_%change

|  |
| --- |
| Calculated Accuracy: 91.98% |
| Calculated Rounded Accuracy: 91.90% |

Fig 22: Test 4 Case 1 Statistical Results

Test 4 was the first testing predicting the number of total deaths using the Southeast states dataset, so, just as the Case 1 in all the other tests, I wanted to create a baseline. I accomplished the same way that I did with the previous tests. It is clear from Figure 21 that the number of total cases is the most important feature when predicting the total number of deaths in a state due to COVID-19. Since this test was dealing with the number of deaths again, I had to follow the same procedure as Test 2 when calculating the accuracies for Test 4. The results in Figure 22 show that predicted number of total deaths is very highly associated with the actual number of total deaths, and the model was able to predict the number of total deaths with an accuracy of 91.90%, which is the highest accuracy that the Southeast states dataset has produced.

**Case 2**: X = {days, fips, new\_case, new\_death, retail\_and\_recreation\_%change, grocery\_and\_pharmacy\_ %change, parks\_%change, transit\_stations\_%change, workplaces\_%change, residential\_%change} Y ={tot\_death}

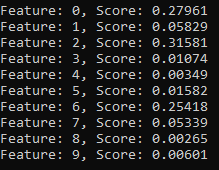


Fig 23: Test 4 Case 2 Important Features: Feature 0 = days, 1 = fips, 2 = new\_case, 3 = new\_death, 4 = retail\_ and\_recreation\_%change, 5 = grocery\_and\_pharmacy\_ %change, 6 = parks\_%change, 7 = transit\_stations\_%change, 8 = workplaces\_%change, 9 = residential\_%change

|  |
| --- |
| Calculated Accuracy: 88.35% |
| Calculated Rounded Accuracy: 88.31% |

Fig 24: Test 4 Case 2 Statistical Results

In order to find the other import features in predicting the total number of deaths, the total number of cases was removed in Test 4 Case 2. Figure 23 shows that the number of new cases is now the most important feature. This is followed by the date and the percent change in parks. Figure 24 shows that both the association between predicted and expected values and the accuracy fell compared to Test 4 Case 1. This could be explained given the size of the dataset and the importance of the total number of cases as a feature. The accuracy only fell a few points to 88.31%, which is higher than it was in Test 3 Case 2.

**Case 3**: X = {days, fips, tot\_cases, new\_case, parks\_%change} Y ={tot\_death}

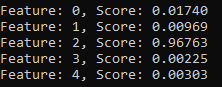


Fig 25: Test 4 Case 3 Important Features: Feature 0 = days, 1 = fips, 2 = tot\_cases, 3 = new\_case, 4 = parks\_%change

|  |
| --- |
| Calculated Accuracy: 93.13% |
| Calculated Rounded Accuracy: 93.24% |

Fig 26: Test 4 Case 3 Statistical Results

After the successful results of optimization in Test 3 Case 3, I wanted to implement the same process in Test 4 Case 3. By removing all input other than date, fips, total cases, new cases, and precent change in parks, the model produced the best output for the Southeast states dataset. As seen in Figure 25, the most important feature is the total number of COVID-19 case in a state; this was expected based on the two previous cases. The statistical results in Figure 26 show the overall, best results produced from the Southeast states dataset. The accuracy increased to 93.24%, which is almost on par with the accuracies obtained from the much larger Southeast counties dataset.

##### V. Conclusion

Both the Southeast counties and Southeast states datasets were able to produce outputs with impressive accuracies. The Southeast counties dataset produced accuracies for the predicted number of cases in a county ranging from 94.63% - 95.36% and accuracies for the predicted number of deaths in a county ranging from 95.25% - 97.81%. While the Southeast states dataset produced accuracies for the predicted number of total cases in a state ranging from 85.81% - 89.95% and accuracies for the predicted number of total deaths in a state ranging from 88.31% - 93.24%. The overall high accuracies produced by both of these datasets show that random forest regression is an acceptable model for predicting the number of cases and deaths for a viral epidemic. The discrepancies between the accuracies may be due to the vast difference in the sizes of the datasets. This seems to point to the conclusion that random forest regression works better on larger datasets with more entries.

Overall, I am very pleased with my results, but there are some changes to the datasets and the test methodology that I would like to implement in the near future. First, I would like to increase the scope of each dataset. For the Southeast counties dataset, I want to increase the dataset to include all of the countries within the United States, and for the Southeast states dataset, I want to increase the dataset to include all of the states within the United States and possibly all the states within North America. Next, I would like to expand the number of input categories for each dataset. For the Southeast states dataset, I want to add population, area, and population density. After that is completed, I want to add temperature data, weather data, elevation data, and demographic statistics to both datasets. Finally, I would like to test these newly updated datasets using my same research methodology and then compare these results by testing the datasets with different machine learning models. I would like to test the datasets with a Support Vector Machine model and a Naïve Bayes model. By having these additional reference points, I will be able to determine a better evaluation for my random forest regression model.

##### 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