Data-driven Modeling and Prediction of COVID-19 Infection

Jeffrey Kim, Nicholas Liu

Questions

- Can we predict the total number of cases and deaths from COVID-19?
- Can we predict the ratio of deaths to cases (death rate)?
- Which features best predict the total number of cases?
- Which features best predict the total number of deaths?

Data Cleaning

Counties dataset:

The boxplots of each feature identified a few features with 0 variance. For example, the 'federal guidelines' feature had the same value for every county. Such features have little predictive power so these features were removed.

The missing data ranking revealed that some of the counties and features in the dataset have very few recorded values. For example, the 3 year mortality for persons aged 5 to 15 feature has values for less than 200 of approximately 3200 counties. Furthermore, some counties have very few values for each feature; in an extreme example, the South Boston county with county FIPS code 51780 has 0 recorded values beyond its name and FIPS. We removed counties with less than 40 values and features with less than 2000 values because having an extreme number of missing values could skew our analysis.

Normalization: Some features have values that are much larger than other features. For example, values for population will be much higher than values for median age. To prevent these larger values from dominating predictions, we normalize and demean each quantitative feature, leaving categorical features the same.

Feature Selection: The correlation heatmap showed that a lot of features were highly correlated and thus potentially redundant. This means that the data could very well be represented in a lower dimension space. We decided to use sklearn's recursive feature elimination library in order to choose a subset of features.

One-hot encoding: The rural-urban continuum code is a classification scheme that distinguishes metropolitan on a scale from 1 to 9. A problem we have is that the difference between a label of 1 (Metro county of population > 1,000,000) and 2 (Metro county of population > 250,000 and < 1,000,00) is not the same as the difference between a label of 2 and 3 (Metro county of population < 250,000). Because this is a categorical feature, we modified it into a one-hot encoding of 9 binary features.

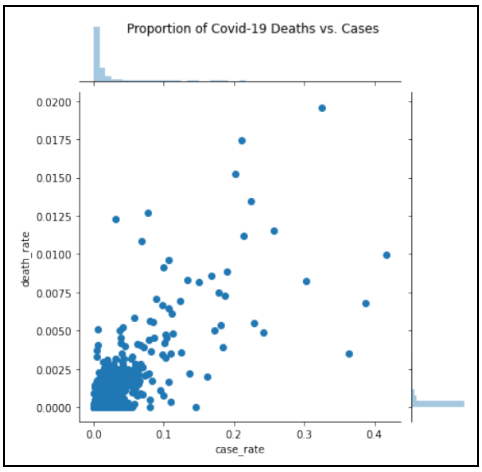

Figure 1: Distribution of deaths/population vs cases/population

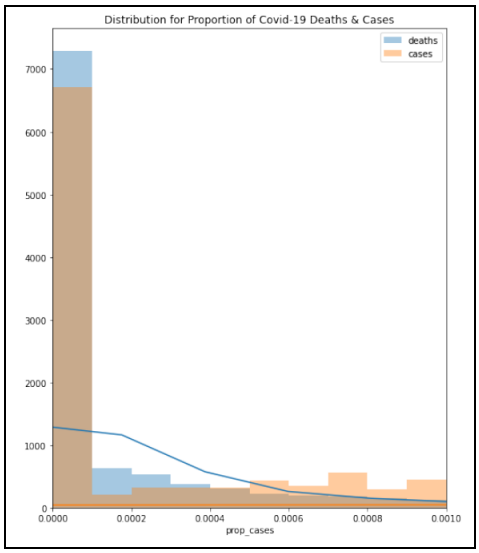

Figure 2: Distribution of deaths/population and cases/population

Data Exploration

Deaths and cases dataset:

Four of the counties in both the deaths and cases dataset were missing values for their FIPS code. Because we used this code to join our tables, we removed the four counties with missing codes.

We also created a scatter plot and distribution graph for the number of deaths vs. cases due to COVID-19.

Counties dataset:

Box Plots: We created boxplots of each feature in the dataset to roughly visually approximate the spread of each feature and identify the existence of outliers. Some features had very low variance while others had very notable outliers. However, in many of the boxplots, we believe these outliers to be the result of differences in the counties themselves rather than data collection, so they were left in the dataset.

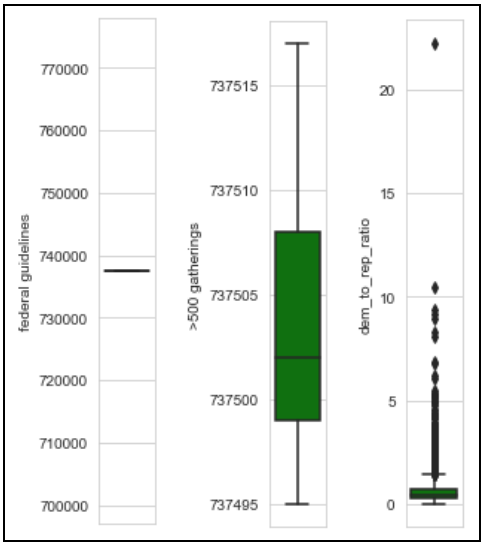

Figure 3: Box plots of selected features

Scatterplots and covariance matrix heatmap: We created scatterplots of features that were likely to be related in order to show the correlation between two features. For example, the number of men aged 10-14 in a particular county is likely to be very similar to the number of women aged 10-14. Scatterplots showing a very linear relationship between such features indicates correlation. We then created a correlation heatmap to quickly identify correlated variables and quantify the degree of correlation. Highly correlated features had darker shades of blue.

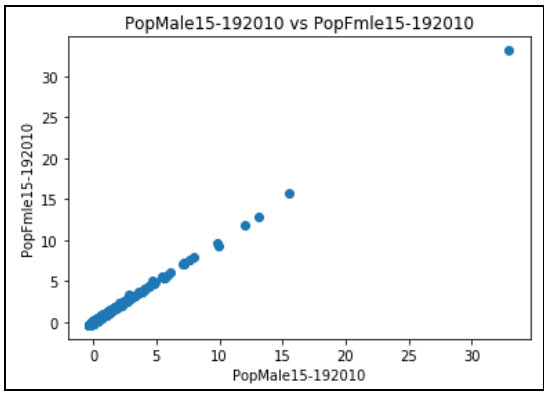

Figure 4: Scatterplot

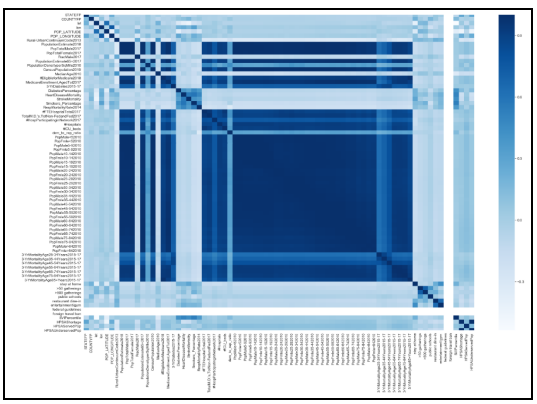

Figure 5: Covariance matrix heatmap

Missing Data Ranking: We also counted the number of missing values in each row and column of the data to determine how complete our dataset was. By sorting these values from greatest to least, we found certain counties and features that had many many missing values.

```
Number of nans in each column:

3-YrMortalityAge1-4Years2015-17      3179
mortality2015-17Estimated            3149
3-YrMortalityAge5-14Years2015-17     3149
3-YrMortalityAge<1Year2015-17        2772
3-YrMortalityAge15-24Years2015-17    2610
3-YrMortalityAge25-34Years2015-17    2270
3-YrMortalityAge35-44Years2015-17    1916
3-YrDiabetes2015-17                  1744
HPSAUnderservedPop                   1141
HPSAServedPop                        1141
dtype: int64
```

Figure 6: Null values per feature

```
Number of nans in each county:

CountyName
Kansas City                          78
Dade                                 78
Yellowstone Nat Park                 78
South Boston City                    78
New York City                        78
Clifton Forge City                   76
Aguijan                              75
Prince of Wales-Outer Ketchikan      75
Rota                                 75
Skagway-Hoonah-Angoon                75
dtype: int64
```

Figure 7: Null values per county

Methodology

First, we performed data cleaning and normalization of each of our data tables and then merged them together using each county's FIPS code as an index.

We then created two columns corresponding to the total number of cases and total number of deaths both divided by the total population of the county. We then grouped these proportions into one of five different classes that represents how high of risk a county is. Then, we split our data into training and validation data, training our model on the training data and using the validation to assess its accuracy.

Using the original county data, we used the sklearn recursive feature elimination package to identify the most predictive subset of features.

We then trained two more models with only 30 out of the original 86 features, to prove that certain features were more predictive than others.

Using the recursive feature eliminator, we were able to determine which features had the most predictive power for each of the different scenarios.

Results

For the cases predictor, the training and testing accuracies are quite low between 40%-45%. Although this seems to be

quite poor initially, when we plot the predicted values against the actual values with the percent occurrence on the heatmaps (Figure 8), we can see that our predictions are generally pretty close. The lighter a square is, the more that that (prediction, label) pair shows up in our predictions. We can see that along and close to the top-left to bottom-right diagonal, the squares are generally lighter in color. This shows that although our model is perfect at predicting classes, it's usually not far off.

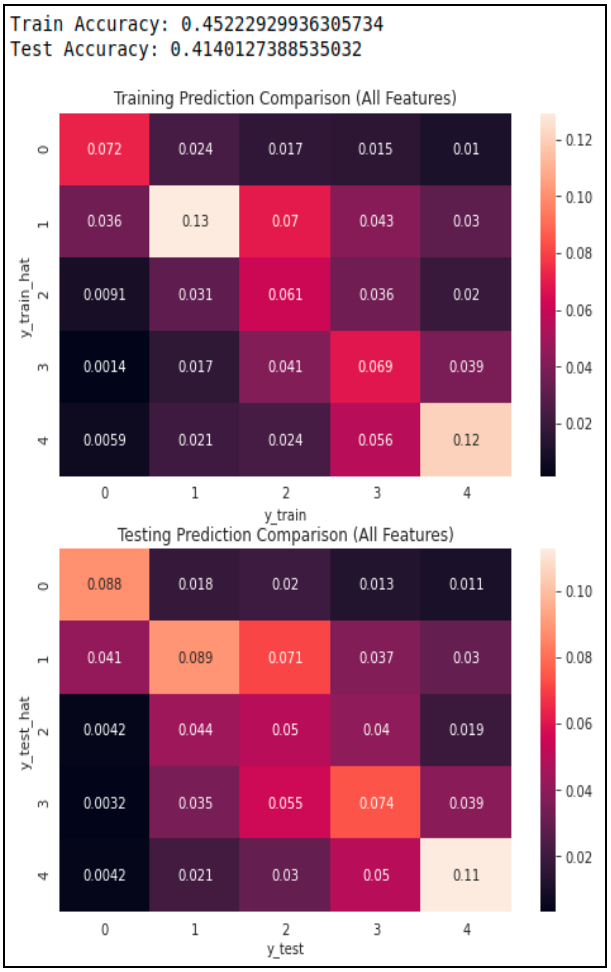

Figure 8: Prediction visualizer for Cases/Population

Our death/population predictor, ended up predicting over 80% of counties to be in the 0 class. This is due to the fact that over 60% of the training data is of the 0 class, and the other 40% is split between the other 4 classes. The predictor is able to achieve higher accuracy by predicting 0 more frequently, but this result isn't what we desire, because it shows that the predictor isn't making its decisions based on features but by exploiting the distribution of training points.

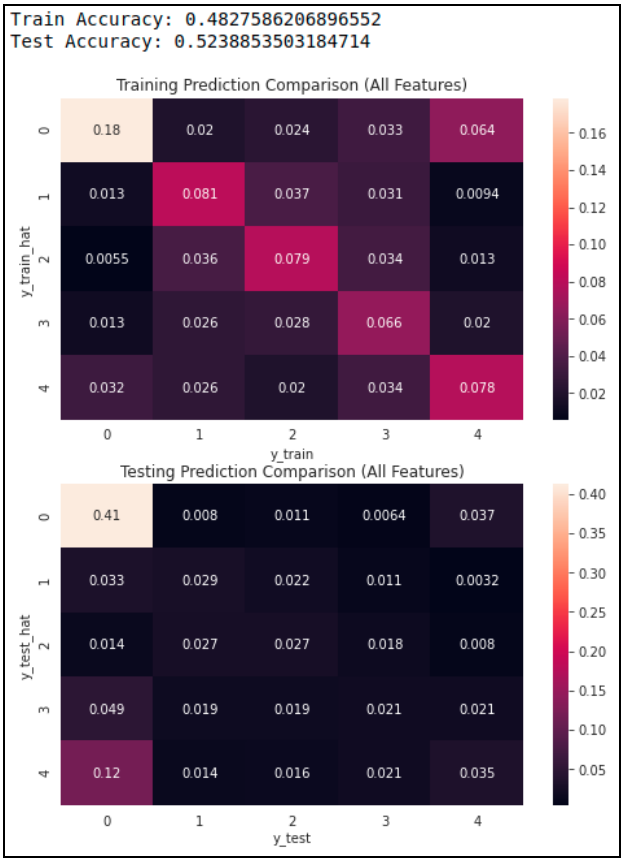

Figure 9: Prediction visualizer for Deaths/Cases

When training our death/cases predictor, we decided to drop 80% of the training points with label 0 so that there was a more even distribution within the training set. This resulted in a higher testing accuracy than training accuracy. Although strange, this is reasonable because the training set is not representative of the real world. We see that in testing, the model correctly predicts the class 0, 41% of the time, which is significantly larger than the 18% in training. Because the distribution of the training and testing sets are different and the testing accuracy is significantly greater, this indicates that our model is indeed using the features to predict labels, rather than exploiting the distribution of the training set.

Using recursive feature elimination, we found that the top three predictive features for a county's COVID-19 cases were population, total number of M.D.'s, number of ICU beds, HPSA underserved population, and population density. Though we expected the number of cases to correlate with the population and population density of each county, it was interesting that metrics about county hospitals also strongly predicted the number of cases. Perhaps a shortage of ICU beds and medical staff could explain an increased number of cases.

1	PopMale45-542010
2	PopMale65-742010
3	PopFmle<52010
4	PopFmle10-142010
5	PopFmle30-342010
6	PopTotalMale2017
7	3-YrMortalityAge45-54Years2015-17
8	3-YrMortalityAge65-74Years2015-17
9	PopMale60-642010
10	PopFmle>842010
11	PopMale<52010
12	PopFmle35-442010
13	PopMale>842010
14	PopFmle60-642010
15	rucc9
16	3-YrMortalityAge75-84Years2015-17
17	Population
18	PopulationDensityperSqMile2010
19	PopFmle75-842010
20	PopMale10-142010

Figure 10: Best features for Cases/Population

Using recursive feature elimination, we found that the two most predictive features for a county's deaths/population were population and 3-year mortality.

1	PopFmle5-92010
2	case_rate
3	#EligibleforMedicare2018
4	PopFmle10-142010
5	PopMale30-342010
6	#ICU_beds
7	PopFmle30-342010
8	PopFmle15-192010
9	PopFmle>842010
10	PopFmle45-542010
11	TotalM.D.'s,TotNon-FedandFed2017
12	#FTEHospitalTotal2017
13	PopMale10-142010
14	PopMale65-742010
15	PopMale>842010
16	PopFmle25-292010
17	PopMale5-92010
18	3-YrMortalityAge55-64Years2015-17
19	PopMale25-292010
20	PopMale45-542010

Figure 11: Best features for Deaths/Cases

7 Questions

Most interesting features:

The number of ICU beds proved to be a strong predictor for the number of cases of COVID-19. Though we are unsure of the reason for the correlation, a shortage of ICU beds and lack of patient hospitalization could result in an increased number of cases.

Surprisingly ineffective feature:

We initially thought that features like population density per square mile would be strong predictors for the number of deaths and cases, expecting the virus to more easily infect large, dense populations.

The SVI percentile was surprisingly ineffective. Looking into the feature more, the Social Vulnerability Index rates a community's ability to respond to hazardous events, including infectious diseases and pandemics. We would expect communities with higher SVI percentiles to have a lower number of cases and deaths. However, the index also incorporates a community's ability to respond to hazardous events like natural disasters, which may influence the score.

Challenges:

We struggled to incorporate the rural-urban continuum code feature. As a feature that describes the degree of urbanization in a county, it could be very important in predicting the spread of COVID-19. However, we struggled to incorporate this categorical variable into our model, which requires quantitative variables. Instead, we transformed the feature into 9 binary features indicating a county's rural-urban continuum code.

Using linear regression to predict the total number of cases and deaths in a given county resulted in models with fairly low scores. Instead, we chose to classify each county's cases and deaths into one of five ranges, shifting our model to a classification problem. We solved this classification problem using logistic regression.

Limitations and assumptions:

Addressing null values was difficult, as we had no way to accurately fill in missing values with additional data. Instead, we noticed that rows in the dataset were grouped geographically by state and decided to fill in null values with preceding and succeeding values. Here, we assume that counties in each state will have similar values for each feature. Though this assumption may not necessarily be correct, we determined this method of addressing null values would be more accurate than using either the mean or median, which would consider all counties across the country.

Ethical dilemmas:

Private data: The original dataset in the github has a collection of private data the documents individuals social distancing and mobility habits. This data was collected by monitoring personal and work devices using their GPS position. Although this type of data could quantify adherence to social distancing guidelines, we worried that it would be too invasive into individual privacy.

Dem-to-rep ratio: One of the features in the dataset calculated the ratio of democrats to republicans. We were hesitant to draw correlations between political affiliation and the number of cases or deaths in a particular county because these findings would be the result of correlation not causation. However, because political affiliation is a particularly divisive issue recently, such findings could be taken out of context and used to fuel division. Ultimately, the feature proved to have little significance in our models predictions.

Additional data:

Additional data on the number of deaths over time would allow us to make more accurate conclusions. In the current dataset, we only have the number of deaths over a 10-day period in the beginning of April. Furthermore, many of the counties have 0 recorded deaths over this time period because they have yet to be affected by COVID-19. Acquiring updated counts of deaths and cases through present day would strengthen our model's predictive power.

Ethical concerns:

Because the dataset includes values for social distancing measures like implementation of stay-at-home measures and closing of restaurants, false negatives present an ethical dilemma. If our model suggests that these measures have little effect on the spread of COVID-19 when in reality they have slowed the spread of the disease, the model might support reopening the economy, endangering the lives of workers and citizens. To address this ethical concern, we can increase the threshold for declaring social distancing measures ineffective, reducing false negative findings.

Evaluation and Limitations

We changed our modeling objective from a regression problem to a classification problem. Although this allowed us to improve the accuracy of our model, it only allows us to approximate the total number of cases or deaths within one of five ranges. Additionally, our model would be unable to extrapolate predictions outside of these ranges.

Our model may be relying too heavily on population parameters. We tried to account for the effect of population by normalizing some labels and features with respect to population. However, because the spread and death toll of pandemics in the absence of social distancing measures are exponential, the model might still be overly affected by the population metrics of each county. To account for this limitation, we could re-run our analysis and training using the log of certain population features.

Surprising Discoveries

We were surprised that the date at which social distancing measures were put in place had little effect on our models' predicted number of cases/population and deaths/cases. We expected to see counties with early adoption of social distancing measures to have significantly lower numbers of cases and deaths given the exponential nature of pandemics.

Future Work

Further study of the CDC's SVI percentile score could yield interesting results. Since the SVI models a county's ability to respond to pandemics, we would like to learn which features the CDC uses in making its assessment and see if these features could be incorporated into our model. Since our models essentially try to assess which factors make a community vulnerable to disease, studying how an established government organization scored the same query could reveal holes in our analysis.