CS 760: Machine Learning Fall 2020

Predicting COVID Occurrence Using Surrounding Localities Information

AUTHORS: Matthew Viens, Ranganath Selagamsetty

GO GREEN. AVOID PRINTING, OR PRINT 2-SIDED OR MULTIPAGE.

As part of the pervasiveness of COVID-19 during 2020, there have been many approaches to predict the spread of the virus on scales ranging from worldwide to the local level. As the data on virus occurrence moves from the global aggregate to the local scale, more locations that have no data reported on either virus occurrence or virus absence appear. The analysis herein proposes several predictors that can leverage ZIP Code level data about the surrounding localities to estimate virus occurrence statistics using Nearest Neighbors Regression and Nearest Neighbors augmented Linear Regression. As the results will show on data from the State of Ohio, locality specific virus occurrence statistics are not wholly replaceable by surrounding locality information, though in some situations near parity can be achieved with some locality specific information.

1 Introduction

Since the start of the COVID-19 Pandemic, there have been many different attempts at predicting almost anything to do with COVID including Occurrence Rates, Risk Factors, Death Rates, and Vaccine Efficacy. The facet of COVID prediction that is studied herein attempts to break down macro-scale global pandemic predictions into locality based predictions from the surrounding localities of the COVID Occurrence, hereafter referred to as the Absolute or Rate values depending on context. There are many different ways of looking at the connections between localities at varying levels of granularity and complexity to predict COVID occurrence, one of the most important choices being what level of locality is chosen to study. In the United States, there are various ways of grouping localities, including neighborhoods, townships, ZIP codes, counties, and states in a rough ordering of increasing scale. The choice of which level of locality to study was driven by both the availability of data the scale of localization. From a balancing of these two factors, ZIP Codes were chosen as the level of locality based information to study and hereafter ZIP Codes and localities will be used interchangeably.

After the choice of ZIP code based data, the next natural question was to determine what tools to use to build predictors. Three different predictors were developed for this: a reference Linear Predictor model (Ref-Linear), a Linear Predictor model including Distance information (Dist-Linear), and Nearest Neighbors Regression model using Distance information (Dist-NNR). Ref-Linear serves as a baseline predictor of performance for estimating COVID occurrence in a ZIP Code when there is historical information about that specific ZIP Code. Dist-Linear is the first of the two predictors that includes information about the neighboring ZIP Codes and that information is applied in a Linear Regression process. Dist-NNR is the second of the two predictors that includes information about the neighboring ZIP Codes and it only uses the distance to the nearest K neighbors and the COVID occurrence statistic of interest to build the predictor.

The three predictors were chosen to provide the capability to analyze several distinctions. The Ref-Linear model and its corresponding error results provide insight into how robustly historical information about

COVID occurrence in a locality can be used to predict occurrence over other time periods. The Dist-Linear model and its error results provide insight into how surrounding locality COVID occurrence information can be used in place of historical context to predict occurrence over matching time intervals when in a linear structure along with information about localities that is not COVID specific. The Dist-NNR model and its corresponding error results provide insight into how much locality COVID occurrence information can be replaced by looking only at the distance to the nearest K localities and their respective COVID occurrence statistics to predict over matching time intervals in a structure as simple and elegant as Nearest Neighbors.

The remainder of the paper will have the following structure. First, there will be a discussion of Related Work both on COVID specific and locality based predictors. Second, an explanation of the details of the Dataset used for analysis including sourcing, pre-processing, and synthesis of core files. Third, an exposition of what approach was used to build the predictors including algorithms, tools used, and result structure. Fourth, core results from the predictors will be shown with a full accounting of all results being found in the Data Appendix. Fifth, conclusions based off those results and directions for future work. Data and Code Appendices are also included at the end of the paper.

2 Related Work

The related work on this topic can be broken down into three major categories. First, work that is COVID specific involving the prediction of COVID occurrence statistics. Second, work that is related to using locality predictors of various results based off characteristics of surrounding localities. Third, work that is a fusion of the first and second.

In an example of the first, Yadav et. al. [1] developed a Support Vector Regression (SVR) predictor to predict the spread of the novel COVID-19 virus. The main contributions of this work were to analyze the transmission rate of the virus and correlating the coronavirus and weather conditions. Using the Pearson's Correlation, the authors reported a correlation between temperature and humidity with an increase in the likelihood of contracting the virus.

In an example of the second, Goin et. al. [2] studied the community characteristics associated with firearm violence. Using random forest machine learning algorithms, the authors linked communities with higher rates of firearm violence with over 300 community characteristics, including: climate, demographics, education attainment, marital status, etc. The study was limited to an examination of the communities the highly populated cities of California. The results showed the importance of including community based features when predicting on a community based label.

In an example of the third, Khmaissia et. al. [3] showed the importance of making use of data within a locality to predict features about that locality. In this work, the authors present an unsupervised machine learning framework to detect factors that are highly correlated to the rate of new COVID-19 cases in New York City (NYC). The authors built their dataset from the 236 socioeconomic features collected from the 177 registered ZIP codes in NYC. The results show successful prediction of COVID-19 daily increase rate based on ZIP with similar socioeconomic factors.

3 Dataset

There are three major sections to the Datasets used for the analysis herein, the COVID-19 specific features, the ZIP code information, and the additional features added for linear consideration. Each will be addressed below.

3.1 COVID-19 Data

The choice was made to limit the examination of COVID-19 data to a single US State for several reasons. First, State Departments of Health are the main collectors of data above a locality level before it becomes national data. Second, State level data provides enough data as to make the analysis compelling but is small enough for the processing to be managed and augmented with additional features of interest. Third, State level data is available and in formats that are easily parsable. After the decision to pick a single US State, it became a question of what State to select. The three that were in consideration were Wisconsin, Colorado, and Ohio. Because of the ease of accessing the Ohio data by ZIP Code directly from the Department of Health, Ohio was selected to be studied.

The Ohio COVID-19 Dataset [4] had 7 different features for each ZIP Code, which were ZIP Code Population, Case Count Cumulative, Case Count Last 30 Days, Case Count Last 14 Days, Case Count Cumulative Per 100 Thousand, Case Count Last 30 Days Per 100 Thousand, Case Count Last 14 Days Per 100 Thousand. Some lines of the Dataset only include ZIP Code and Population, and do not specify if that means no cases in that ZIP Code or that the data is for other reasons absent.

3.2 ZIP Code Data

The ZIP Code Data for distance information was provided by finding a dataset of all the ZIP Codes in the US [5] along with their associated Latitude and Longitude coordinates (Lat/Long). The Lat/Long coordinates come from the geographic centroid of the region the ZIP code covers. The Lat/Long coordinates are what is used to generate distance information from one locality to the other. The Lat/Long coordinates are the easiest way to estimate the distance between ZIP Codes since sequential ZIP Codes are not guaranteed to be adjacent or even proximate to each other. An example of this would be the ZIP Code 43210 which is in Columbus, OH. The two numerically adjacent ZIP Codes 43209 and 43211 are not geographically contiguous with each other even if they are all located in Columbus, OH. The specific method of computing distance will be addressed in 4.2 Distance Analysis.

3.3 Additional Features

Several additional features where found to add to Linear Analysis in versions of the Ref-Linear and Dist-Linear models. These features include Population Density of a ZIP Code in People per Square Mile [6], and how many universities are in a given ZIP Code along with the total number of university students [7]. The three features were chosen to see how strongly indicative they could be when combined with other distance and COVID-19 occurrence information.

3.4 Combined Dataset

The COVID-19 Data, ZIP Code Data, and Additional Features were all synthesized together into one file called, simply, Dataset from hereon. There are 1446 Zip Codes to perform analysis on in the Dataset. While there are occasional locations in the dataset where the individual features are absent, as a whole the dataset is well filled in and sufficient to proceed on to the approach to model design.

4 Approach

4.1 Linear Model Approach (Ref-Linear)

For the first implementation of a COVID-19 data predictor, a simple Linear Regression model was used to predict a current COVID-19 statistic based off of historical COVID-19 data within a ZIP code. Linear Regression was chosen as the first predictor due to its ubiquitous usage in research for similar problems. The method is incredibly flexible and robust, able to incorporate continuous valued variables to assess their interactivity. Additionally, an open-source implementation, scikit-learn, was readily available, and commonly used in the field.

Scikit-learn [8], or sklearn, implements the Linear Regression Model using Ordinary Least Squares Regression. The coefficient vector $\mathbf{W}$ is calculated by computing the following equation to minimize prediction error:

$$\mathbf{W} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

$\mathbf{W}$ has dimensionality equal to the *(number of features + 1)*. The original feature matrix $\mathbf{X}$ is augmented with an additional column whose value is one for each data sample. This allows the bias/intercept term to be calculated in the above matrix operations.

As seen from Table A.1, the first predictor was used to predict the values of the various COVID-19 features using historical COVID-19 data within a particular ZIP code. The first row shows the error rate of predicting Cumulative Absolute Occurrence by looking the absolute occurrence in the last 14 & 30 days, the current occurrence rate, and the occurrence rates in the last 14 & 30 days. The following three rows show the error rate of making this prediction using only some of the aforementioned features. The last three rows show the error rate of predicting the Cumulative Per 100K Occurrence Rate using some or all of the historical COVID-19 rate data for a given ZIP code.

The next predictor used the the non-COVID-19 features that the dataset was augmented with, namely: population, population density (per sq. mi.), number of universities, and total number of university students. Table A.1 shows the error rate of predicting the various COVID-19 features using all available data (COVID-19 and non-COVID-19) collected from a particular ZIP code.

4.2 Distance Analysis

There are two major features that should be addressed before moving on to the Distance Based Predictors. The first is the data being used for this Distance Analysis, the ZIP Codes themselves. The second is the method of computing distance between the ZIP Codes, Geodesic Distance.

4.2.1 ZIP Codes

ZIP Codes are only one way of interacting with locality data and were invented by the US Postal Service to help coordinate mail shipments. As a consequence of this, ZIP Codes are not designed for properties like comparable population or population density. The US Census Bureau uses a related concept called a ZIP Code Tabulation Area (ZCTA) [9] to address some of these issues but those values do not map directly or easily to ZIP Codes. The available data for COVID-19 occurrences in Ohio broken down by localities was done in ZIP Codes. This means that even if there are equivalents that would have better properties when used as a predictor, it would take much more raw data access than is presently available to use ZCTAs instead of ZIP Codes. The variability in population directly impacts the capabilities of ZIP Code based metrics,

a detail that will be discussed in more detail in the Results section, but can be seen in simple statistical metrics. The Ohio ZIP Codes from the Ohio Covid-19 Dataset [4] have a mean population of 9,788, a median population of 3,938, and a standard deviation in population of 12,685. The variability will cause a marked difference between metrics which are in terms of absolute population or occurrence and those which are in terms of relative occurrence rates.

4.2.2 Geodesic Distance

A simplistic model of the Earth is a perfect sphere, and distances are computed using the Great Circle Distance [10]. This formulation calculates the arc distance between two points across a sphere. This model is particularly attractive to larger datasets due to its ease of calculation.

However, the distance formulation used here makes use of Geodesic Distances, which allows the distance between any two points on any Euclidean surface to calculated. Given the ellipsoidal shape of the Earth, Geodesic distances have been proven to yield more accurate distances. The localized nature of the dataset suggests that it may make sense to use great circle distances in computing the distances between ZIP codes. However, to make the model both accurate and extensible, Geodesic distance formulation was used. Using the Geodesic distance allows data from other states to be incorporated seamlessly.

To implement this distance calculation, the open-source GeoPy [11] project was used. GeoPy implements the Geodesic distance calculation by using the WGS-84 ellipsoid model, the most globally accurate [12] model, of the Earth which takes and Lat/Long Coordinates as arguments. Distance between ZIP codes is calculated by the Geodesic distance from the center of one ZIP code to the center of the other ZIP code using the Lat/Long Coordinates from the US Zip Code Dataset [5].

4.3 Distance Based Predictors

4.3.1 Linear Model including Distance (Dist-Linear)

The final linear predictor used the information from a particular feature of neighboring ZIP codes. To implement this, the distances between the centers of various ZIP codes were computed, and a feature vector was built by polling the neighboring ZIP codes for a specific data feature. Consider the ZIP code 45883, and see the vector below showing the closest neighboring ZIP codes 45828, 45348, and 45860:

$$Distance\ from\ 45883\ in\ miles = [4.63_{45828}, 5.67_{45348}, 6.14_{45860}]$$

The following vector would then be built for the feature matrix $\mathbf{X}$ to predicting Cumulative Absolute Occurrence, for a ZIP code using three neighboring ZIP codes:

$$\mathbf{X}_{45883} = [Cumul.Abs._{45828}, Cumul.Abs._{45348}, Cumul.Abs._{45860}]$$

It's important to note that that the order of the elements in the data sample maintains the distance information of the ZIP codes. By preprocessing the distance information, the model remains linear in terms of coefficients. The Linear Regression predictor fits the data to create a coefficient matrix where each coefficient represents the relative importance of the kth neighbor's COVID-19 data.

As with the the predictor described in the 4.1, the Linear Regression model from Scikit [8] was used to implement this predictor. Table A.2 shows the error rate when predicting a COVID-19 feature incorporating

information from a specific number of neighbors. The various rows indicate differing number of neighbors to consider when evaluating the prediction.

4.3.2 Nearest Neighbors Regression (Dist-NNR)

Nearest Neighbors Regression makes use of a notion of distance between a set of reference points that have reference values and a target point for which the target value is unknown or an estimate is wanted. The mathematical description of this is shown below:

$$\hat{y}_j = \frac{\sum_{i=1}^{K} w_{n(i,j),j} y_{n(i,j)}}{\sum_{i=1}^{K} w_{n(i,j),j}}$$

Where the estimate for $\hat{y}_j$ becomes an interpolation based on the K nearest neighbors of ZIP Code j for the COVID-19 statistic of interest coming from the known y values. Where $w_{a,b}$ is the weight function between ZIP Code a and ZIP Code b. Where $n(i,j)$ is the index of the $i-th$ nearest neighbor of ZIP Code j based on the geodesic distance. The definition for this is adapted from a reference on Nearest Neighbor techniques [13].

There are several different ways of computing $w_{a,b}$ based on the distance between the two points, $d_{a,b}$. The most natural is Inverse:

$$w_{a,b,inv} = \frac{1}{d_{a,b}}$$

Another option is to use a Negative Exponential:

$$w_{a,b,exp} = e^{-d_{a,b}}$$

Since distance can be such a large value relative to these terms, some normalization notions were tried. Since distance needs to remain on $[0, \infty)$, a standard score assumption, which would assume normal distribution, does not apply. So mean and standard deviation divided versions of the previous metrics were tried.
Where $\mu = \frac{1}{N}\sum_{p,q} d_{p,q}$ and $\sigma = \sqrt{\frac{1}{N}\sum_{p,q}(d_{p,q}-\mu)^2}$ over all relevant pairs p, q where N is the number of those relevant pairs.
The Mean Divided Inverse becomes:

$$w_{a,b,mean\_inv} = \frac{\mu}{d_{a,b}}$$

The Mean Divided Negative Exponential becomes:

$$w_{a,b,mean\_exp} = e^{-\frac{d_{a,b}}{\mu}}$$

The Std Divided Inverse becomes:

$$w_{a,b,std\_inv} = \frac{\sigma}{d_{a,b}}$$

The Std Divided Negative Exponential becomes:

$$w_{a,b,std\_exp} = e^{-\frac{d_{a,b}}{\sigma}}$$

All six of these metrics were used in Dist-NNR to make different predictors based of the w values and were manually computed in a Python script written for the analysis, more details in Appendix B.

5 Results

5.1 Error Definition

The method of error rate calculation, and therefore accuracy comparisons, was standardized across predictors to use the same training and testing split by using SciKit's train_test_split() method with the same random seed of 7 and the same test_size of .25. The error definition is as follows:

$$Error\ Rate = \frac{1}{|Test|} \sum_{i \in Test} \frac{|\hat{y}_i - y_i|}{y_i}$$

5.2 Ref-Linear

The Ref-Linear results were quite accurate, with Cumulative Rate predictors based on 14 Day Rate, 30 Day Rate, and the combination of the two achieving error rates of .27, .23, and .22 respectively. This would indicate that Cumulative Rate is best predicted if using the combination of the 14 & 30 day rates. When the target statistic is Cumulative Absolute, the error rate uniformly increases. The data is shown in more depth in A.1 as Table 2.

When additional locality specific details of a given zip code is provided the Error Rate for Cumulative Rate, 14 Day Rate, and 30 Day Rate all shrink to less then .20 as shown in A.1 as Table 3.

5.3 Dist-Linear

The Dist-Linear results were markedly split between the Absolute results and the Rate results. The Absolute Occurrence Error Results were all greater than 1 for all tested K Neighbor Values 1 and 100. The Rate Occurrence Results were much better with Error rates between .22 and .36 depending on the number of neighbors used and the statistic in question. Cumulative Rate achieved best results with 5 neighbors, error of .22, and worst results with 1 neighbor, error of .29. 30 Day Rate achieved best results with 5,10, & 25 neighbors tying , error of .25, and worst results with 1 or 100 neighbors, error of .29. The 14 Day Rate achieved best results with 10 & 25 Neighbors, error of .32, and worst results with 100 neighbors. The data is shown in more depth in A.2 as Table 4.

Overall, the trend is for the Absolute Occurrence statistics to be uniformly poor predictors and the Rate Occurrence statistics to be better using 2 - 10 neighbors, with extremely small or large numbers of neighbors increasing error.

5.4 Dist-NNR

The Dist-NNR results were also split between Absolute results and Rate results. The Absolute Occurrence results all had error values above 1. The Rate Occurrence results are all less than .45. the K=1 neighbor results did not vary at all based on distance metric because of the cancellation occurring in the scaling terms across all the metrics, shown in more detail in Table 6 A.3. Hence K=1 results in the predictor copying the occurrence rates of the nearest ZIP code. The K = 2 through K = 100 results all match in terms of the accuracy of the predictors. This is likely because of the distance values between ZIP Codes increased markedly as the values beyond the first few neighbors are examined.

The K=1 results are uniformly better than any of the K $\geq$ 2 test on Absolute Occurrence with the exception of the Exponential results which are comparable to the K = 1 results or slightly better. On Rate Occurrence,

all $K \geq 2$ tested performs better than $K = 1$ with the exception of Mean Divided Negative Exponential being slightly worse. The best results for Cumulative, 14 & 30 Day Rates occur with $K \geq 2$ with all of the Inverse based methods and the error values are .29, .29, & .35 respectively. The Inverse based methods matching exactly makes sense, as the division through by a constant cancels when the sum of all weights term is including in the Nearest Neighbors Regression Definition. The data is shown in more depth in A.3 Tables 6-11.

5.5 Predictor Comparison

The comparison between the three different types of predictor are straight forward. The Ref-Linear model uniformly performed better than the Dist-Linear or Dist-NNR models when using all data from the ZIP Code by at least .05. The Dist-Linear results in the best case were able to match the Ref-Linear model when Ref-Linear was used to predict Cumulative Rate based off 14 Day Rate. So there are cases in which the Linear model using surrounding ZIP Code based information matches the Linear model that can use target ZIP Code specific information but that occurs only when the amount of target ZIP Code specific information is restricted. The Dist-NNR results tend to act as a performance lower bound on the Dist-Linear results.

6 Conclusions

6.1 Future Work

There are several directions to take additional work on the techniques described herein to pursue better performance. An examination that split Urban and Rural areas into separate datasets entirely could allow for separate Linear coefficients improving Dist-Linear results or improving Dist-NNR results by removing potential perturbations at the boundary between Urban and Rural areas. An analysis of different US States with different topographical characteristics could find better applicability in a more agrarian and sparse state like Kansas or a denser and industrialized state like Massachusetts. The Dist-NNR approach could specifically be tried with other normalization methods like min-max normalization to increase the impact of proximate but not adjacent neighbors. Hence there remain many potential future avenues for research.

6.2 Concluding Remarks

Overall, the locality specific information about COVID-19 Occurrence Statistics cannot be entirely replaced with information from the surrounding area with the techniques explored herein, there are some cases where Dist-Linear is able to match predicting COVID-19 Rate occurrence statistics if one of the other Cumulative, 14 Day, or 30 Day occurrence statistics were available. So in cases where entire rows of ZIP Code COVID-19 Occurrence information are absent, which did occur in some cases in the Ohio Coronavirus Dataset of [4], a Linear Predictor that includes distance information could be a useful estimate. Additionally, with the further parsing of the data, as considered above in Future Work, there are directions to take this analysis to potentially produce higher accuracy predictors.

References

[1] *Analysis on novel coronavirus (COVID-19) using machine learning methods*, Milind Yadav, Murukessan Perumal, Dr. M Srinivas. 2020. Chaos, Solitons, & Fractals

[2] *Predictors of firearm violence in urban communities: A machine-learning approach*, Dana E. Goin, Kara E. Rudolph, and Jennifer Ahern. 2018. ScienceDirect

[3] *An Unsupervised Machine Learning Approach to Assess the ZIP Code Level Impact of COVID-19 in NYC*, Fadoua Khmaissia, Pegah Sagheb Haghighi, Aarthe Jayaprakash, Zhenwei Wu, Sokratis Papadopoulos, Yuan Lai, and Freddy T. Nguyen. 2020. Cornell University

[4] Ohio Department of Health.
https://coronavirus.ohio.gov/wps/portal/gov/covid-19/dashboards/key-metrics/cases-by-zipcode
Retrieved 11/29/20

[5] US Zip Code Dataset
https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/table
Retrieved 11/29/20

[6] Ohio Population Density ZIP Code Rank
http://www.usa.com/rank/ohio-state–population-density–zip-code-rank.htm

[7] Ohio Universities Dataset
https://en.wikipedia.org/wiki/List_of_colleges_and_universities_in_Ohio
Retrieved 11/29/20
Manually scrapped from page
Augmented with latitude/longitude & zip code data from Google Answers on google searching university name and "lat long"
County information from augmented Wikipedia

[8] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
https://scikit-learn.org/stable/

[9] Zip Code Tabulation Areas
https://www.census.gov/programs-surveys/geography/guidance/geo-areas/zctas.html

[10] Great Circle Distance
https://en.wikipedia.org/wiki/Great-circle_distance

[11] GeoPy
https://geopy.readthedocs.io/en/stable/

[12] *Algorithm for geodesics*, Charles F. F. Karney. 2013. Springerlink.com

[13] Nearest Neighbors. Daniel L. Pimentel-Alarcon.
https://danielpimentel.github.io/teaching/CS760/lectures/CS760_9KNN.pdf

Appendices

A Detailed Results

Table 1. Feature Abbreviation Key	
Abbreviation	Expansion
Metric	Used Distance Metric
Pop.	Population
Cumul. Abs.	Cumulative Absolute Occurrence
14 Day Abs.	14 Day Absolute Occurrence
30 Day Abs.	30 Day Absolute Occurrence
Cumul. Rate	Cumulative Per 100K Occurrence Rate
14 Day Rate	14 Day Per 100K Occurrence Rate
30 Day Rate	30 Day Per 100K Occurrence Rate

A.1 Ref-Linear Results

Table 2. Ref-Linear Results Using Partial COVID Data from Zipcode		
Label	Features	Error Rate
Cumul. Abs.	14 Day Abs., 30 Day Abs., Cumul. Rate, 30 Day Rate, 14 Day Rate	0.81
Cumul. Abs.	14 Day Abs., 30 Day Abs.	0.24
Cumul. Abs.	14 Day Abs.	0.36
Cumul. Abs.	30 Day Abs.	0.27
Cumul. Rate	14 Day Rate, 30 Day Rate	0.22
Cumul. Rate	14 Day Rate	0.27
Cumul. Rate	30 Day Rate	0.23

Table 3. Ref-Linear Results Using All Data from Zipcode					
Cumul. Abs.	14 Day Abs.	30 Day Abs.	Cumul. Rate	14 Day Rate	30 Day Rate
0.75	0.31	0.29	0.16	0.17	0.11

A.2 Dist-Linear Results

K-Neighbors Error Result for Each Feature

Table 4. Dist-Linear Results						
K	Cumul. Abs.	30 Day Abs.	14 Day Abs.	Cumul. Rate	30 Day Rate	14 Day Rate
1	3.47	2.50	2.58	0.29	0.29	0.34
2	2.92	2.17	2.25	0.25	0.27	0.33
5	2.63	2.03	2.06	0.22	0.25	0.33
10	2.74	2.09	2.12	0.25	0.25	0.32
25	2.61	2.05	2.12	0.24	0.25	0.32
50	2.70	2.10	2.15	0.27	0.27	0.33
100	3.09	2.37	2.46	0.27	0.29	0.36

A.3 Dist-NNR

Results Key

Table 5. Distance Metric Abbreviation Key	
Abbreviation	Expansion
Inv	Inverse
Exp	Negative Exponential
Mean Inv	Mean Divided Inverse
Mean Exp	Mean Divided Negative Exponential
Std Inv	Std Divided Inverse
Std Exp	Std Divided Negative Exponential

K-Neighbors Error Result for Each Feature

Table 6. K = 1 Error Rates							
Metric	Pop.	Cumul. Abs.	14 Day Abs.	30 Day Abs.	Cumul. Rate	30 Day Rate	14 Day Rate
Inv	2.07	3.22	2.13	1.98	0.32	0.32	.44
Exp	2.07	3.22	2.13	1.98	0.32	0.32	.44
Mean Inv	2.07	3.22	2.13	1.98	0.32	0.32	.44
Mean Exp	2.07	3.22	2.13	1.98	0.32	0.32	.44
Std Inv	2.07	3.22	2.13	1.98	0.32	0.32	.44
Std Exp	2.07	3.22	2.13	1.98	0.32	0.32	.44

Table 7. K = 2 Error Rates							
Metric	Pop.	Cumul. Abs.	14 Day Abs.	30 Day Abs.	Cumul. Rate	30 Day Rate	14 Day Rate
Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Exp	2.05	3.20	2.11	1.96	0.31	0.30	0.41
Mean Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Mean Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36
Std Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Std Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36

Table 8. K = 5 Error Rates							
Metric	Pop.	Cumul. Abs.	14 Day Abs.	30 Day Abs.	Cumul. Rate	30 Day Rate	14 Day Rate
Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Exp	2.05	3.20	2.11	1.96	0.31	0.30	0.41
Mean Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Mean Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36
Std Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Std Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36

Table 9. K = 10 Error Rates

Metric	Pop.	Cumul. Abs.	14 Day Abs.	30 Day Abs.	Cumul. Rate	30 Day Rate	14 Day Rate
Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Exp	2.05	3.20	2.11	1.96	0.31	0.30	0.41
Mean Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Mean Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36
Std Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Std Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36

Table 10. K = 25 Error Rates

Metric	Pop.	Cumul. Abs.	14 Day Abs.	30 Day Abs.	Cumul. Rate	30 Day Rate	14 Day Rate
Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Exp	2.05	3.20	2.11	1.96	0.31	0.30	0.41
Mean Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Mean Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36
Std Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Std Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36

Table 11. K = 100 Error Rates

Metric	Pop.	Cumul. Abs.	14 Day Abs.	30 Day Abs.	Cumul. Rate	30 Day Rate	14 Day Rate
Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Exp	2.05	3.20	2.11	1.96	0.31	0.30	0.41
Mean Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Mean Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36
Std Inv	3.88	4.38	3.48	3.38	0.29	0.29	0.35
Std Exp	3.97	4.39	3.53	3.44	0.34	0.33	0.36

B Code Files

All of the code for the project is availible at `https://github.com/BujSet/CS_760_Ohio_COVID_Project`
Where BujSet is Ranganath Selagamsetty, viens-code is Matthew Viens, and solitonreachgit is also Matthew Viens (there was an error in a Git push that used an old author tag).
The ReadMe file goes over the details of running each of the predictors.
All data files used in the creation of the predictors is included in the repository.
All results files are folders for results according to what predictor they were made from.
All predictors used the same random seed (7) for train_test_split with test being 25% of the data.