Regression, Clustering, Classification of COVID-19 in Maryland

Arun Kulkarni*, Joshua Shevitz†
* (Towson University Undergrad): dept. of Computer Science, Towson, USA, akulka1@students.towson.edu
† (Towson University Undergrad): dept. of Computer Science, Towson, USA, jshevi2@students.towson.edu

***Abstract*—This paper details an analysis of COVID-19 in Maryland. The datasets used come from the Maryland State Department of Health, as well as the 2010 Maryland Census. The first dataset includes the total number of reported, positive COVID-19 cases in every zip code in Maryland - these are provided by day, starting from April 11th 2020, and, in the version of the dataset used for this experiment, ending on April 25th, 2021. The second dataset contains Zip Code Tabulation Area (ZCTA) data from the 2010 Maryland census, and is composed of population and demographic data. The two datasets were joined on their zip code properties, yielding 449 usable entries. After some exploratory data analysis (EDA) and preprocessing, the following analyses were performed: linear regression with zip code populations, *k*-means binning of COVID-19 case data, and finally 4 different classification methods using ZCTA data as features and COVID-19 clusters as classes.**

***Index Terms*—Exploratory Data Analysis (EDA), k-nearest neighbors (KNN), decision trees, random forests, naive Bayes, holdout, partition, classification, Pandas, Python, scikit-learn**

I. Introduction

On January 9Th, 2020 WHO announced the outbreak of a pneumonia-like virus in Wuhan, China. Not long after this announcement on January 21st the first case of the Coronavirus appeared in the United states only a day after beginning to screen the airports for the virus. On this same day, a Chinese scientist, Zhong Nanshan, MD, confirmed the virus can be transmitted from person to person. [1] By the time of this discovery, it was reported that over 4 people were killed due to the virus and over 200 people were infected. At this point, the WHO still was unsure of the need to declare a pandemic-level health emergency; however it took only two days for the people of Wuhan to be quarantined after 13 more people were reported to have died from the virus, and an additional 300 became sick. On January 31, the WHO issued a global health emergency. It was at this point that the reported death toll had risen to over 200 with more than 9,000 reported infected or sick with the virus.

The finding and spread of COVID-19 has caused many problems for the people of every race, sex, ethnicity, nationality and more. With the spread of the disease finally being fought with the invention of the COVID-19 vaccines, the world has had a chance to reflect on the past year and look at how this pandemic has effectively shaken the worlds population. This research paper takes a look at the cases of COVID-19 in the state of Maryland's different zip code regions, and attempts to find correlation between different characteristics and attributes.

II. Methodology

All experiments described in this paper were performed using Python, and various data mining, machine learning, and data visualization libraries including pandas, scikit-learn, scipy, and matplotlib.

A. Dataset Descriptions

The datasets used for this experiment come from [4] and [7]. The former, from [4], is a CSV file of Maryland ZCTA demographic data from the 2010 census. This data includes factors such as total population, total number and percentage of residents by ethnicity, and total percentage of residents under 18 or over 65 for nearly all zip code regions in the state of Maryland. To reduce bias in the classification models, the percentage-based ethnicity demographics were chosen for experimentation, not the count-based demographics. This also factors in varying overall populations, and the fact that count-based demographics are derived from total population.

The second dataset, from [7], is a CSV file of the total number of reported positive COVID-19 cases in each zip code region in the state of Maryland by date. This dataset is updated multiple times a week, and begins with April 11^{11}, 2020. The version of the dataset used in this experiment ends on April 25th, 2021.

It should be noted that, henceforth in this paper when discussing "total" COVID-19 cases, the implication is that this is the total, per zip code region, as of the aforementioned date of April 25th, 2021.

B. Data Cleaning

1) COVID-19 Dataset: The COVID-19 case dataset contained poorly-labeled columns, with strange formatting of the dates. Pandas was used to convert the date labels to uniformly formatted date-time objects. Additionally, the dataset was inverted using the pandas "melt" function, with the output as one date per row and the different zip codes along the columns.

2) ZCTA Dataset: The ZCTA dataset contained several columns unnecessary to the experiments, so these columns were dropped. The columns that remained were strictly referring to demographic and population data. These numerical attributes for each zip code region, selected for this experimentation, are listed here:

1) 2010 Population
2) Percent Over 65 years of age
3) Median Age

4) Percent Under 18 years of age
5) Percent Non-Hispanic White
6) Percent Non-Hispanic Black
7) Percent Non-Hispanic American Indian
8) Percent Non-Hispanic Asian
9) Percent Non-Hispanic Native Hawaiian
10) Percent Non-Hispanic Other
11) Percent Non-Hispanic Total
12) Percent Hispanic

3) Joining Both Datasets: One obstacle faced in this experiment was the varying size of the two datasets. The dataset from [7] contained entries for a total of 516 zip codes; however the dataset from [4] contained data from 468 zip codes. The two datasets were joined using a full inner join on the zip code column, and entries with null values were dropped. This decision was made, in place of imputation, for the sake of cleanliness and focus on experimentation. Further work could be done in the area of data imputation with this dataset and the analysis of its impact.

Upon completion of the join, the dataset contained a total of 449 entries (representing 449 zip codes) with ZCTA demographics and total COVID-19 cases for each row entry.

C. EDA and Preprocessing

First, EDA was performed on the manipulated COVID-19 dataset. Figure 1 shows an empirical cumulative distribution function (ECDF) of total reported COVID-19 cases, along with the median and mean daily increases by zip code in Maryland. As one can see from the visualization, the distribution of total cases by zip code skews heavily to the right; the majority of zip codes have fewer than 2000 cases, with a near-negligible daily increase.

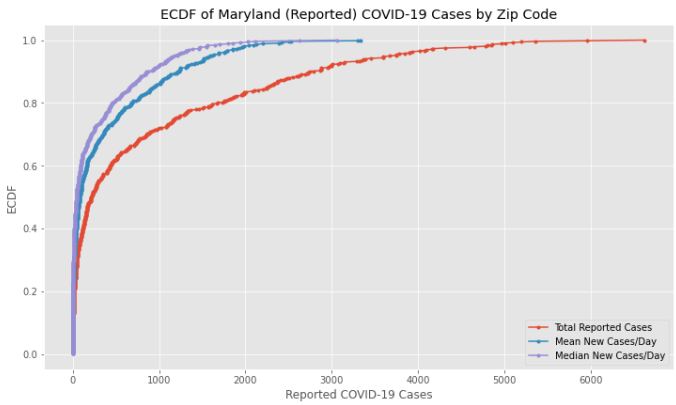

Fig. 1. ECDF of Maryland reported COVID-19 cases

Figures 2 and 3 show a box plot and bee swarm plot of COVID-19 cases in Maryland, respectively. These figures more clearly demonstrate the strong right-skewness, with the majority of zip codes having fewer than 1500 reported cases.

To test for normality, a graphical approach was used, utilizing a cumulative distribution function (CDF) of Maryland COVID-19 cases by zip code. The shape of the data is far from normal, as seen in Figure 4. Figure 4 was created using

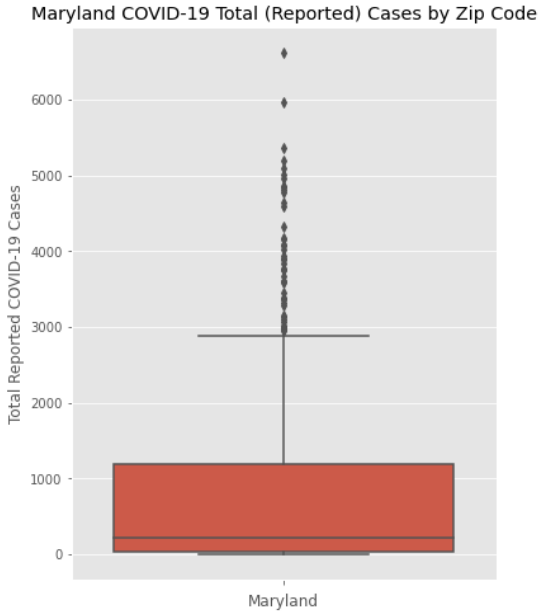

Fig. 2. Box plot of Maryland reported COVID-19 cases by zip code

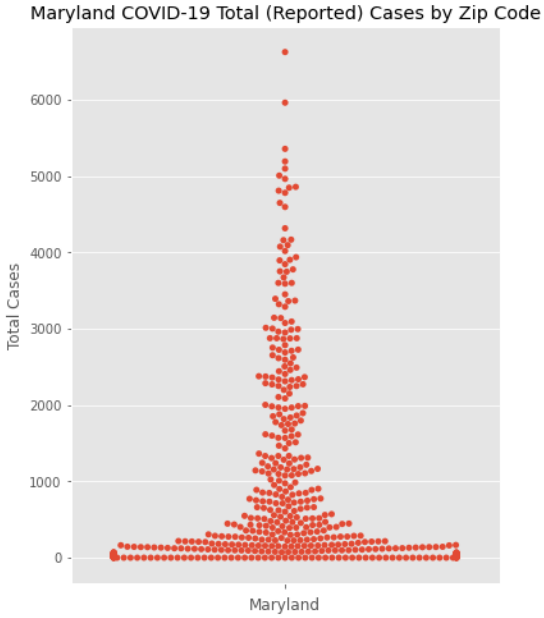

Fig. 3. Bee swarm plot of Maryland reported COVID-19 cases by zip code

a randomized normal distribution with the dataset's sample mean and standard deviation.

Next, some of the variables from the ZCTA dataset were explored. The two chosen for EDA were median age and population, as these two factors seemed to intuitively have significant impacts on the spread of any disease, including COVID-19.

Figure 5 shows a scatter plot of median age vs. total reported COVID-19 cases per zip code in Maryland; one can see that zip codes with a median age between approximately 35 and

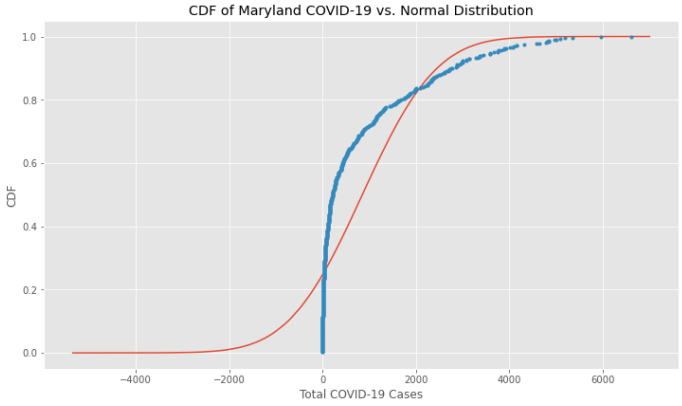

Fig. 4. CDF of Maryland reported COVID-19 cases by zip code

45 have the highest number of reported cases. The calculated Pearson correlation coefficient for this visualized relationship was approximately -.36.

Figure 6 shows another scatter plot, this one of population vs. total reported COVID-19 cases per zip code in Maryland. This plot demonstrates a strong linear relationship between population and COVID-19; the Pearson correlation coefficient was approximately .96.

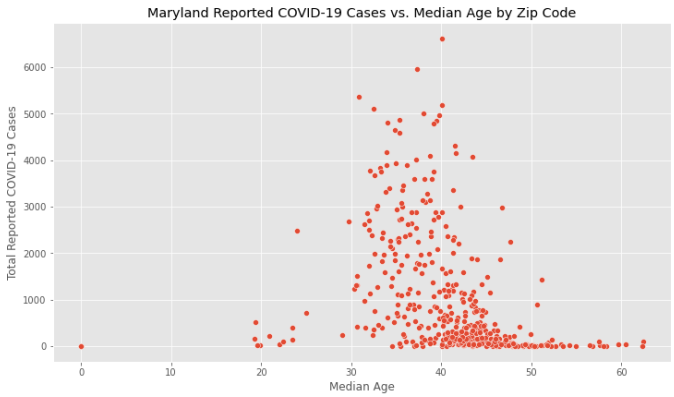

Fig. 5. Median age vs. total COVID-19 cases

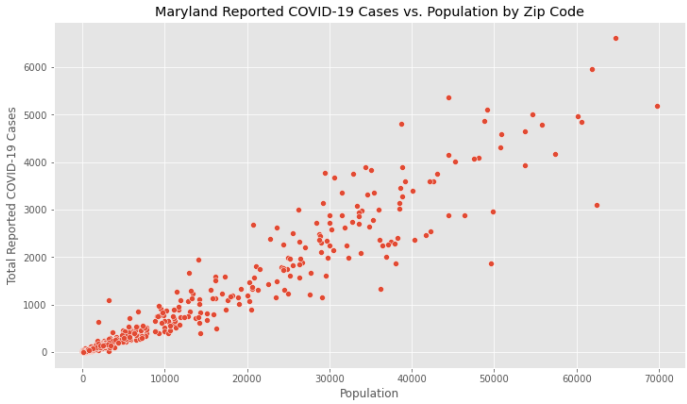

Fig. 6. Population vs. total COVID-19 cases

III. Experimentation and Results

A. Linear Regression

Due to the strong linear relationship between population and total COVID-19 cases, a single (2-D) linear regression was performed using these two variables; population was used to predict the COVID-19 case count in a zip code.

First, a linear regression line was computed and drawn, along with 200 bootstrap lines. The bootstrap lines were calculated from a randomly generated set of sample data, using the observed mean and standard deviations. Figure 7 shows the scatter plot from Figure 6 with the regression and bootstrap lines drawn; the tight variance among the bootstrap lines is an indicator of a strong linear relationship, likely leading to a high value of R^2.

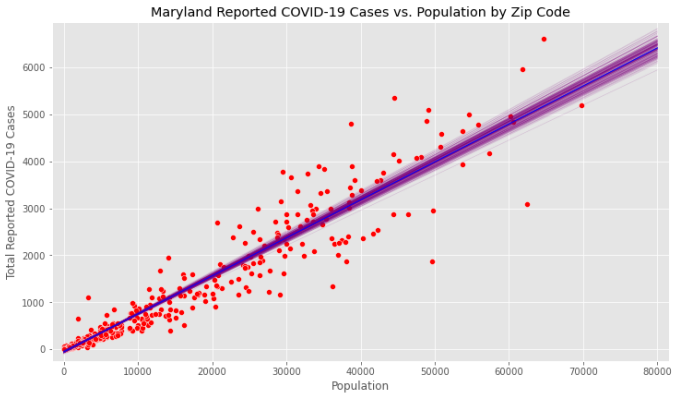

Fig. 7. Population vs. total COVID-19 cases: regression with 200 bootstrap lines

The metrics, including the value for R^2, were calculated as well, using the Python library statsmodels. Here, in Figure 8, the R^2 value can be seen as .913 with an adjusted value of .913. This indicates a strong model, with the model accounting for approximately 91.3% of variance among the samples. It should be noted as well that the metrics figure shows 447 observations due to the fact that outliers were dropped from the data prior to analysis.

OLS Regression Results

Dep. Variable:	TotalCases	R-squared:	0.913
Model:	OLS	Adj. R-squared:	0.913
Method:	Least Squares	F-statistic:	4659.
Date:	Mon, 03 May 2021	Prob (F-statistic):	7.10e-238
Time:	21:46:40	Log-Likelihood:	-3291.3
No. Observations:	447	AIC:	6587.
Df Residuals:	445	BIC:	6595.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-58.4500	23.693	-2.467	0.014	-105.015	-11.885
x1	0.0810	0.001	68.255	0.000	0.079	0.083

Omnibus:	73.779	Durbin-Watson:	1.383
Prob(Omnibus):	0.000	Jarque-Bera (JB):	954.001
Skew:	0.054	Prob(JB):	6.94e-208
Kurtosis:	10.156	Cond. No.	2.62e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.62e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Fig. 8. Statsmodels regression metrics

Next, a prediction question was posed using the linear regression model: in a hypothetical zip code region with a population of 15,000, what would be the total number of cases? Using the model, the prediction yielded a value of approximately 1,156 - a likely accurate prediction, given the high value of R^2.

B. k-Means Binning

Since the main focus of this experiment was classification, categorical variables representing COVID-19 case data were required. To solve this problem, *k*-means binning was performed on the set of total COVID-19 cases by zip code.

First, the elbow method was used to determine the optimal value of *k*. This was performed using KMeans from scikit-learn, and yielded the figure shown in Figure 9.

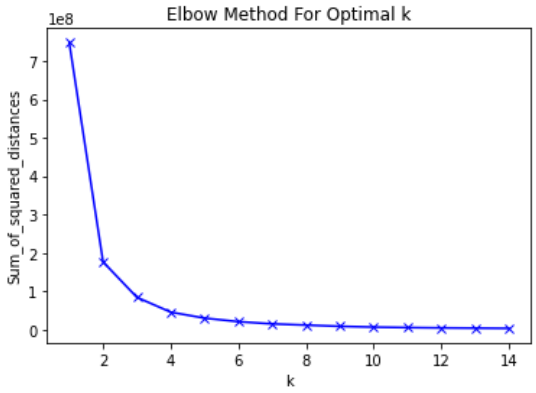

Fig. 9. Elbow method to find optimal *k* for clustering

It can be seen visually that the "elbow" of the graph occurs at *k*=3, so 3 was chosen for the number of clusters. With the three clusters now determined, the cluster centers were set at approximately 289.6, 2158.3, and 4250.8; this can also be interpreted as "low" (cluster 0), "medium" (cluster 1), and "high" (cluster 2) numbers of COVID-19 cases. The numeric labels (0, 1, 2) were converted to string literals to avoid bias in the classifiers, and joined to the dataset. To get a better idea of the shape of the clusters, Figure 10 shows a histogram of the counts of the different clusters. Cluster 0 has the highest count, followed by cluster 1 and then cluster 2. This makes sense and aligns with the previous EDA, as the COVID-19 case totals skew right.

C. Classification Algorithms

This section outlines the preprocessing involved prior to using the algorithms, the classification algorithms used, and their results. The accuracy metrics used to gauge the performance of the different classifiers include measures of precision, recall, F_1 score, and overall accuracy. The weighted averages are discussed, as they take into consideration the counts of the different class labels in the datasets. These metrics are all calculated using *metrics* from the Python library scikit-learn.

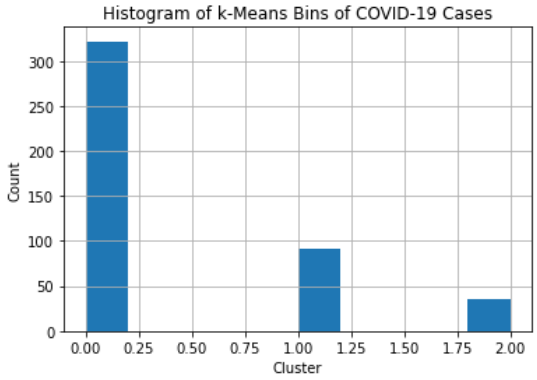

Fig. 10. Histogram of *k*-means cluster counts

1) Normalization and Partitioning: Prior to the training and testing of classifiers, the data was normalized (using min-max normalization) and partitioned using the holdout method along a randomized 70-30 split. In other words, 70% of the data was randomly chosen for training, and the remaining 30% was chosen for testing. The features selected for testing include population, median age, percent of the population over 65 years of age, percent of the population under 18 years of age, and 8 different metrics regarding the percentage of different ethnic groups in each zip code region.

Min-max normalization was used prior to partitioning on the selected feature set. This decision was made due to the wide variety of data ranges and different units used; for example, the range of the population feature was significantly greater than the range of any percentage measurement. The normalization process was implemented to eliminate bias among the classifiers, so that all features would be given equal weights.

After partitioning, a statistical *t*-test was used to validate the means of the subsets created by the partition. Using a *p*-value of .05, we could not reject the null hypothesis that the means of the training and testing subsets were identical for all features except for one: the percentage of the population under 18 years of age. It was speculated that this was due to the random state used or a high variance for that particular feature. Also, it should be noted that the *t*-tests were performed without the assumption of equal variances.

2) K-Nearest Neighbors: This algorithm is an example of instance learning, as training set records are first stored. When a new instance is introduced, the distance between this new instance and a predetermined *k* number of neighbors is calculated. The neighbors are sorted by distance, and the nearest neighbors are determined based on the *k*-th minimum distance. A simple majority of the class of the *k*-nearest neighbors is then used as the prediction value of the new instance.

The number *k* in this experiment was determined by finding the local minima of the mean error as *k* increased. A value of *k* that is too small can lead to underfitting and a value of *k* that is too large can lead to overfitting, so choosing optimal values for *k* is important. As shown by the plot of the mean

error rate vs. k in Figure 11, the value k=7 was found to be the minimum and was therefore selected for experimentation.

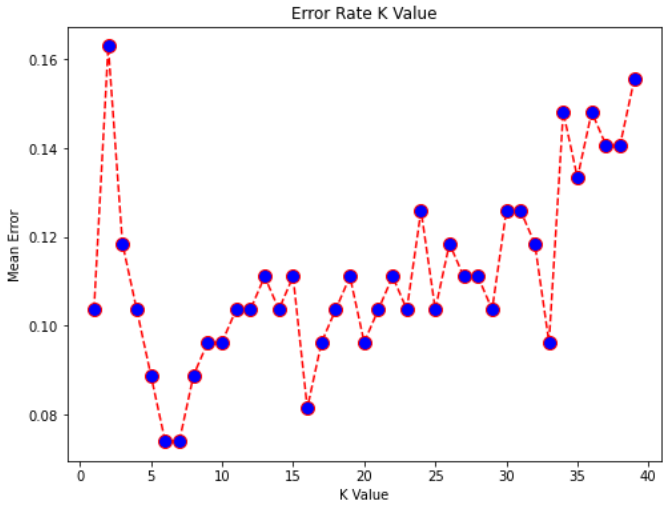

Fig. 11. Mean-error curve as k increases

After training and testing using the preprocessed data, the KNN classifier yielded values of .93, .93, .93, and .93 for the weighted average precision, recall, F_1 score, and accuracy, respectively. The breakdown of the KNN results by cluster can be seen in Figure 12.

	precision	recall	f1-score	support
Cluster0	0.71	0.83	0.77	12
Cluster1	0.98	0.97	0.97	87
Cluster2	0.89	0.86	0.87	36
accuracy			0.93	135
macro avg	0.86	0.89	0.87	135
weighted avg	0.93	0.93	0.93	135

Fig. 12. KNN result metrics using k=7

3) Decision Tree: Decision trees use a flowchart-like structure. They are a collection of decision nodes connected by branches that extend downward from a root node to a set of terminating leaf nodes. Each node represents a different test on an attribute, each branch represents an outcome of a test, and finally each leaf node holds a class label.

Each tree is constructed using a top-down, recursive divide-and-conquer algorithm. All training examples start at the root, and then an algorithm is used to partition the numerical attributes (features). The algorithm used in this case is CART, using a Gini index. The CART algorithm for feature selection is the default algorithm used by scikit-learn, and provides strong binary splits. Entropy can be used as the split criteria as well, and this was done in addition to using CART.

Using CART for feature selection and splitting, the decision tree classifier yielded values of .95, .95, .95, and .95 for the weighted average precision, recall, F_1 score, and accuracy, respectively. The breakdown of the CART decision tree results by cluster can be seen in Figure 13.

	precision	recall	f1-score	support
Cluster0	0.80	1.00	0.89	12
Cluster1	0.96	1.00	0.98	87
Cluster2	1.00	0.81	0.89	36
accuracy			0.95	135
macro avg	0.92	0.94	0.92	135
weighted avg	0.95	0.95	0.95	135

Fig. 13. Decision tree result metrics using CART

Using entropy (the ID3 algorithm), the decision tree yielded values of .97, .96, .96, and .96 for the weighted average precision, recall, F_1 score, and accuracy, respectively. The breakdown of the ID3 decision tree results by cluster can be seen in Figure 14.

	precision	recall	f1-score	support
Cluster0	0.80	1.00	0.89	12
Cluster1	0.98	1.00	0.99	87
Cluster2	1.00	0.86	0.93	36
accuracy			0.96	135
macro avg	0.93	0.95	0.93	135
weighted avg	0.97	0.96	0.96	135

Fig. 14. Decision tree result metrics using ID3

4) Naive Bayes: A naive Bayesian classifier is a simple probabilistic classifier based on Bayes' theorem. It assumes naive independence between features, and calculates conditional probabilities based on every feature for each new sample. It uses P(H), the prior probability, along with P(X), the observed probability of a sample, to calculate P(X given H), which is the the likelihood of a sample belonging to a class. This is calculated for each new sample for each class, and the highest P(X given H) wins.

The naive Bayesian classifier used in this experiment yielded values of .91, .90, .90, and .90 for the weighted average precision, recall, F_1 score, and accuracy, respectively. The breakdown of the naive Bayesian classifier results by cluster can be seen in Figure 15.

	precision	recall	f1-score	support
Cluster0	0.65	0.92	0.76	12
Cluster1	0.98	0.94	0.96	87
Cluster2	0.82	0.78	0.80	36
accuracy			0.90	135
macro avg	0.82	0.88	0.84	135
weighted avg	0.91	0.90	0.90	135

Fig. 15. Naive Bayesian classifier result metrics

5) Random Forest: A random forest classifier is an ensemble method that uses a collection of decision trees. The random forest is generated using random selections of attributes at each node to determine splits. During the classification, each randomly generated tree votes and the most popular class is assigned to the test instance. For this experimentation, random forests of 10, 100, and 1000 trees were used.

All three had comparable performance metrics; however, the random forest with 100 estimators had the best performance with values of .96, .96, .96, and .96 for the weighted averages of the precision, recall, F_1 score, and accuracy, respectively. Figure 16 shows the result metrics table for the random forest with 100 estimators; the other two are omitted from this paper for brevity, but are included in the Jupyter notebooks in [8].

	precision	recall	f1-score	support
Cluster0	0.92	1.00	0.96	12
Cluster1	0.96	1.00	0.98	87
Cluster2	1.00	0.86	0.93	36
accuracy			0.96	135
macro avg	0.96	0.95	0.95	135
weighted avg	0.96	0.96	0.96	135

Fig. 16. Random forest with 100 estimators result metrics

IV. Conclusion

COVID-19 has affected the world in significant ways for over a year - many factors lead to the different models of COVID-19 contagion, and these factors can be difficult to determine in conventional ways.

One goal of this experimentation was to try to determine if simple demographic data could be used to predict the rate of COVID-19 infection in a given area, and the results are promising. A simple linear regression with population yielded an R^2 of over .9, implying that population is a somewhat decent predictor of COVID-19. With the prediction question posed, it was also found that in a hypothetical zip code chosen at random, where the population was around 15,000 the number of cases was found using the regression model. The total yielded amount for this was approximately 1,156, which as was stated earlier in the paper, was a likely prediction given the value of R^2 was 91.3%.

The classifiers all had reasonably-high weighted average result metrics; however, the breakdowns of the metrics by cluster told a different story. Of all classifiers used in this experiment, the random forest with 100 estimators had the best macro performance, with precision, recall, and F_1 score values all above .85 (most above .90) for each cluster individually. It can be surmised that, of the clusters, accurate labeling of Clusters 0 and 2 would be the most important in an application setting, since Cluster 0 represents areas of *low* COVID-19 and Cluster 2 represents areas of *high* COVID-19. Therefore, the decision tree(s) and random forests could prove to be the most useful, since they most accurately labeled zip code regions in Cluster 2, or those with *high* numbers of COVID-19.

When looking at the median age per zip code, the model shows that the highest cases per zip code happened between the ages of 35 and 45, with a scattering of cases happening to the populations aged between 45 and 55. **Note: while the point of this research was to find correlation and predictive models of COVID-19 cases per zip code, it can be speculated and further researched whether the reasons for this are poor pre-cautionary measures between these age groups or lowered immune responses due to age or possible health correlations within those age groups.*

References

[1] A. J. M. C. Staff, "A Timeline of COVID-19 Developments in 2020," AJMC, 01-Jan-2021. [Online]. Available: https://www.ajmc.com/view/a-timeline-of-covid19-developments-in-2020.

[2] Allen, J., Robles, F., Aufrichtig, A., and Almukhtar, S., *Maryland Coronavirus Map and Case Count.* The New York Times, 01-Apr-2020. [Online]. Available: https://www.nytimes.com/interactive/2021/us/maryland-covid-cases.html. [Accessed: 04-Apr-2021].

[3] "The Data Maryland ," The COVID Tracking Project. [Online]. Available: https://covidtracking.com/data/state/maryland. [Accessed: 05-Apr-2021].

[4] "Maryland Census Data - ZIP Code Tabulation Areas (ZCTAs)," *Maryland's GIS Data Catalog.* [Online]. Available: https://bit.ly/3nO17Dt. [Accessed: 03-May-2021].

[5] McGuire, M. and Liao, W. COSC467 Python Labs [Source code].

[6] MD COVID-19 Data Dashboard. [Online]. Available: https://coronavirus.maryland.gov/datasets/md-covid-19-data-dashboard. [Accessed: 05-Apr-2021].'

[7] "MDCOVID19 MASTER ZIP CODE CASES," *Coronavirus.* [Online]. Available: https://coronavirus.maryland.gov/datasets/mdcovid19-master-zip-code-cases. [Accessed: 03-May-2021].

[8] Kulkarni, A. and Shevitz, J. *MD_COVID_EDA_Part1, MD_COVID_EDA_Part2, ZCTA_Data_Cleaner, COSC567_FinalProject_DataCleaning, and COSC_467_FinalProject_Clustering_Classification.* [Source code].