Logo

Description automatically generated

Week 2 Assignment – MASSACHUSETTS COVID-19 EXPLORATORY DATA ANALYSIS  
  
Group 1

Craig Perkins – [perkins.cr@northeastern.edu](mailto:perkins.cr@northeastern.edu)

Sneha Sharma Satheesha - [satheesha.s@northeastern.edu](mailto:satheesha.s@northeastern.edu)

Pranay Bindela – [bindela.p@northeastern.edu](mailto:bindela.p@northeastern.edu)

Northeastern University

November 2022 – ALY 6150

Week 2 Assignment – MASSACHUSETTS COVID-19 EXPLORATORY DATA ANALYSIS  
  
Group 1

For our group assignment we are using statewide data from the covid tracking project: <https://covidtracking.com/data/state/massachusetts>.The dataset has daily statewide metrics across a wide variety of areas.

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| **date** | Date of the measurement |
| **state** | **ma** for every row in this dataset (column to be removed) |
| **death** | Cumulative count of death confirmed + death probably |
| **deathConfirmed** | Cumulative count of death confirmed |
| **deathIncrease** | Daily increment of death count |
| **deathProbable** | Cumulative count of probably death attributable to COVID-19 |
| **hospitalized** | Cumulative count of hospitalized (weekly) |
| **hospitalizedCumulative** | Cumulative count of hospitalized (weekly) |
| **hospitalizedCurrently** | Count currently hospitalized |
| **hospitalizedIncrease** | Daily increment of hospitalized |
| **inIcuCumulative** | Cumulative count of ICU visits attributable to COVID-19 (Blank – column to be removed) |
| **inIcuCurrently** | Current count of ICU visits attributable to COVID-19 |
| **negative** | Cumulative count of negative tests |
| **negativeIncrease** | Daily increment of negative tests |
| **negativeTestsAntibody** | (Blank – column to be removed) |
| **negativeTestsPeopleAntibody** | (Blank – column to be removed) |
| **negativeTestViral** | (Blank – column to be removed) |
| **onVentilatorCumulative** | (Blank – column to be removed) |
| **onVentilatorCurrently** | Count on ventilator currently |
| **positive** | Cumulative count of positive tests (Antibody + Antigen + Viral) |
| **positiveCasesViral** | Cumulative count of positive viral tests |
| **positiveIncrease** | Daily increment of positive increase |
| **positiveScore** | (Blank – column to be removed) |
| **positiveTestsAntibody** | (Blank – column to be removed) |
| **positiveTestsAntigen** | (Blank – column to be removed) |
| **positiveTestsPeopleAntibody** | Cumulative count of tests positive for antibodies |
| **positiveTestsPeopleAntigen** | (Blank – column to be removed) |
| **positiveTestsViral** | Cumulative count of positive tests |
| **recovered** | Cumulative count of recovered (weekly) |
| **totalTestEncountersViral** | (Blank – column to be removed) |
| **totalTestEncountersViralIncrease** | (Blank – column to be removed) |
| **totalTestResults** | Cumulative count of tests |
| **totalTestResultsIncrease** | Daily increment of tests |
| **totalTestsAntibody** | (Blank – column to be removed) |
| **totalTestsAntigen** | (Blank – column to be removed) |
| **totalTestsPeopleAntibody** | Cumulative count of antibody tests |
| **totalTestsPeopleAntigen** | Cumulative count of antigen tests |
| **totalTestsPeopleViral** | Cumulative count of viral tests |
| **totalTestsPeopleViralIncrease** | Daily increment of viral tests |
| **totalTestsViral** | How is this different to totalTestResults? |
| **totalTestsViralIncrease** | How is this different to totalTestResultsIncrease? |

Samples of rows within the dataset after removing duplicate columns, blank columns and columns that do not provide useful information:

Table

Description automatically generated

Descriptive Statistics of the dataset:

Table

Description automatically generated

The first analysis I performed was to visualize the number of positive tests by day using a line chart in pandas. Dates with N/A for the number of positive tests are treated as 0. Below is a chart of number of positive tests by day:

Chart, histogram

Description automatically generated

See the supplied jupyter notebook for more analysis, I only included the first chart of the analysis for this write-up.

**Data Cleaning**

The correlation matrix below shows below shows that the three variables with over 30% of the data missing have a strong correlation with other variables.

So, considering an imputation algorithm to impute the data is the best option here. We have used KNN imputation algorithm to replace the missing values

A picture containing chart

Description automatically generated

**KNN Imputation**

A scikit-learn class called KNNImputer is used to predict or complete the missing values in a dataset. In comparison to the simplistic strategy of filling all the data with the mean or median, the method that uses the fundamental KNN algorithm is more useful.

Chart, histogram

Description automatically generatedWe now look at other columns with missing data less than 30%. We then plot the distribution graphs to understand the distribution of variables which can help us determine the methods we can use to fill in the missing values.

For the remaining columns of the dataset, we replaced missing values with the previous values. As the professor mentioned in the class, we assume what happened today happens tomorrow.

**Descriptive Statistics after Data Cleaning**

**Table

Description automatically generated with medium confidence**

**Descriptive Statistics**

1. **Positive cases grouped by month, year and every day of the week**

Chart, bar chart

Description automatically generated

1. **Deaths grouped by month, year and every day of the week**

Chart, bar chart, PowerPoint

Description automatically generated

In the above graphs, we are comparing the deaths and positive cases seen over every month and day of the week. These graphs help us analyze the peaks and help us in forecasting when we might see a surge in COVID cases again.

**Feature Engineering**

Date time has been converted to Day, Month, and Year. We used the date and augmented data pertaining to the day. The type of day is important in a time-series problem, especially in problems like covid cases. For example, people may be more likely to go out on Saturdays and Sundays, which raises the likelihood of their becoming infected with Covid and, as a result, the number of positive cases that may emerge in the following days. As a solution, our team has opted to supplement data with details that can be generated based on date.

The new columns are 'year,"month,' 'day,' 'Year,' 'Month,' 'Week,' 'Day,' 'Dayofweek,' 'Dayofyear,' 'Is month end,' 'Is month start,' 'Is quarter end,' 'Is quarter start,' 'Is year end,' 'Is year\_end', 'Is\_year\_start'. These columns provide additional information about the type of activity that may have occurred to the model and facilitate the process of detecting patterns in the data for the model.

The other two columns added are: 'deathDecreased' and 'hospitalizedDecreased'. Some records in the dataset had negative values, suggesting a decline in that particular count from the prior days. Negative values are undesirable because they increase the possibility of diminishing gradients on the model; hence, we separated the negative values and added two extra columns to capture the notion of magnitude reduction.

**Initial Prediction and Forecast**

We divide the data into train and test data. We then apply a forecasting method to forecast the number of positive cases for the next 60 days using

ARMA

**Chart, line chart

Description automatically generated**In ARMA, the term "autoregressive" refers to the model's utilization of previous values to forecast future ones. Predicted values are specifically a weighted linear mixture of historical values. The main distinction between this sort of regression approach and linear regression is that in this case, the feature inputs are historical values.

A moving average is a weighted, linear combination of white noise terms that represents the forecasts, where white noise is a random signal. Here, it is proposed that ARMA forecasts future values by combining white noise and historical data. The behavior of market participants, such as the buying and selling of BTC, is modeled through autoregression. Wars, recessions, and political crises are all models for shock events in white noise.

Using SARIMAX, we can define an ARMA model.[2]

The below graph describes our forecast graph. Red line indicates the prediction and blue indicates the actual values

Chart, line chart

Description automatically generated

**Conclusion**

From the above graphs, we can conclude the following:

1. The peak covid positive months were January and February 2021
2. The deaths have also occurred in higher numbers during these months.
3. We forecasting model can predict with Root mean square error of 205 at this point in our project.

**References**

1. The COVID Tracking Project – Massachusetts <https://covidtracking.com/data/state/massachusetts>
2. [Sadrach Pierre](https://builtin.com/authors/sadrach-pierre) “A Guide to Time Series Forecasting in Python” retrieved from https://builtin.com/