**Group 3 – Final Report**

**Boston Covid-19 Prediction**

Trieu Vo

[vo.trieu@northeastern.edu](mailto:vo.trieu@northeastern.edu)

Felix Castaneda Guzman

[castaneda.f@northeastern.edu](mailto:castaneda.f@northeastern.edu)

Karishma Abhijit Bhalshankar

[bhalshankar.ka@northeastern.edu](mailto:bhalshankar.ka@northeastern.edu)

College of Professional Studies, Northeastern University

ALY 6150: Healthcare/Pharmaceutical Data and Applications

Prof. Roy Wada

**Introduction**

COVID-19 is a contagious respiratory disease that can lead to serious illness in some people. COVID-19 has easily transmitted through respiratory droplets and aerosol particles that an infected person releases when they cough, sneeze, sing, speak and breathe. As of 2019, the global and statewide impact of the COVID-19 pandemic has been severe. The risk of severe COVID-19 illness, hospitalization, and death have been significantly reduced due to widespread vaccination and immunity in the population.

In this Final Project, our goal is to predict new Covid-19 cases 15 days into the future using multiple predictors from 15 days ago. There are questions that we want to find answers to:

* How can we predict future Covid-19 cases in the next 15 days using the past information?
* What is the necessary information to predict new Covid-19 cases efficiently?

We use different Machine Learning models such as VAR, SARIMA and ARIMA to predict new cases and visualize them clearly to show the difference between our prediction and ground truth.

**Dataset**

We use the Boston Covid-19 dataset collected by the Boston Public Health Commission to build models and predict new Covid-19 cases. This dataset can be downloaded directly from the source: <https://bphc-dashboard.shinyapps.io/BPHC-dashboard/>. It contains 17,727 observations of Covid19 metrics over the past three years, from 2020-01-01 to 2022-06-14, equivalent to 856 days. It has 13 features, a combination of all Boston COVID-19 dashboard datasets published on the same website. These features tell us how fast COVID-19 is spreading and how many new cases we see each day.

There are 13 columns in this dataset and their description are as follows:

|  |  |  |
| --- | --- | --- |
|  | **Column Name** | **Description** |
|  | Window\_title | Types of measures (Tests, Test Positivity, Neighborhood, Daily Case Count,…) |
|  | Title | Titles of tables in each measure |
|  | Description1 | Description of tables (Dates, COVID-19 Cases, CLI Visits, Age, Neighborhood, Race/Ethnicity, City and Sex, COVID-19) |
|  | Category1 | Regions in Boston (Boston, Roslindale, Dorchester,...) |
|  | Description2 | Measures, Week range, Race/Ethnicity |
|  | Category2 | All-Boston Tests, Community Tests, College-Ordered Tests, Threshold |
|  | Value | Values of Windown\_title (float) |
|  | Value\_unit | Unit of values (cases, percentage, 7-day moving average, deaths,…) |
|  | Value\_1 | Values of Windown\_title (float) |
|  | Value\_1\_unit | Unit of values (cases, percentage, 7-day moving average, deaths,…) |
|  | Comparison | Lower than the rest of Boston, Higher than the rest of Boston, Similar to the rest of Boston,…) |
|  | Timestamp | Dates |
|  | Chart\_info | Introduction of the dataset (Data release version) |

**Analysis**

**Count all unique values in “Window\_title”**

**Text

Description automatically generated**

We spotted six subsets from the feature “Window title.” All of them have the same among of records and timestamps, so it would be ideal for merging purposes.

1. COVID-19 Positive Tests
2. COVID-19 Emergency Department Visits
3. Number of Confirmed Adult Hospitalizations due to COVID-19
4. Number of Adult Hospitalizations due to COVID-19
5. Occupied Adult Non-Surge ICU Beds at Boston Hospitals, All Patients
6. Percentage of Occupied Adult Non-Surge ICU Beds at Boston Hospitals, All Patients

**Create a new dataframe for “COVID-19 Tests”**

We separate the “Covid-19 Tests” data from the dataset and create a new dataframe for it. We also convert Category1 of Covid-19 Tests to datetime data type.

A picture containing text

Description automatically generated

**Create a new dataframe for “Current All-Boston COVID-19 Tests”**

We separate the “All-Boston COVID-19 Tests” data from the “Covid-19 Tests” dataframe and create a new dataframe for it. Then we check missing values, fill them with 0, sort the dataset by dates, and reset the index.

Table

Description automatically generated

**Create a new dataframe for "Current Community COVID-19 Tests"**

We separate the “Current Community COVID-19 Tests” data from the “Covid-19 Tests” dataframe and create a new dataframe for it. Then we check missing values, fill them with 0, sort the dataset by dates, and reset the index.

Table

Description automatically generated

**Create a new dataframe for "** **Current College-Ordered COVID-19 Tests"**

We separate the “Current College-Ordered COVID-19 Tests” data from the “Covid-19 Tests” dataframe and create a new dataframe for it. Then we check missing values, fill them with 0, sort the dataset by dates, and reset the index.

Table

Description automatically generated

**Merge all three dataframe into one new dataframe**

We merge all three dataframe “All Boston Tests”, “Community Tests” and “College Tests” into one new dataframe.

Graphical user interface, text, application

Description automatically generated

**Create a new dataframe for “COVID-19 Emergency Department Visits”**

We separate the “COVID-19 Emergency Department Visits” data from the dataset and create a new dataframe for it. Then we check missing values, fill them with 0, sort the dataset by dates, and reset the index.

Table

Description automatically generated

**Create a new dataframe for “Number of Adult Hospitalizations due to COVID-19”**

We separate the “Number of Adult Hospitalizations due to COVID-19” data from the dataset and create a new dataframe for it. Then we check missing values, fill them with 0, sort the dataset by dates, and reset the index.

Background pattern

Description automatically generated with medium confidence

**Create a new dataframe for “Occupied Adult Non-Surge ICU Beds at Boston Hospitals, All Patients”**

We separate the “Occupied Adult Non-Surge ICU Beds at Boston Hospitals, All Patients” data from the dataset and create a new dataframe for it. Then we check missing values, fill them with 0, sort the dataset by dates, and reset the index.

Table

Description automatically generated

**Create a new dataframe for “COVID-19-Like Illness Boston Emergency Department Visits”**

We separate the “COVID-19-Like Illness Boston Emergency Department Visits” data from the dataset and create a new dataframe for it. Then we check missing values, fill them with 0, sort the dataset by dates, and reset the index.

Table

Description automatically generated

**Combine new dataset**

We also wanted to add new features to the dataset, so we downloaded a historical from the Trans Stats (<https://www.transtats.bts.gov/Data_Elements.aspx?Data=1>). It is a historical record of passenger’s arrival at the Logan Airport from 2020 to 2022, so we could match the time series of Boston Positive Cases.

Table

Description automatically generated

**Merge all subset dataframes**

After separating different dataframes as above, we merge them all into one large dataframe which contains necessary information to predict new Covid-19 cases.

**Create new variables**

We extract the year from “Dates” column and create a new column that contains the year only (2020, 2021, 2022).

Additionally, we created a new column with the status of mask policy in Boston Based on The Commonwealth of Massachusetts Executive Office of Health and Human Services Department of Public Health. Mask became mandatory in public places from May 6, 2020, and it was lifted on Feb 28, 2022. So this column takes 3 values “No mask”, “Mask mandated” and “Optional”

Graphical user interface, application

Description automatically generated

**Visualize the target variable**

Our target variable is the number of Positive Tests in Boston from 2020 until June 2022.

Chart, histogram

Description automatically generated

**Handle categorical variables**

We convert “Mask” and “Year” to dummy variables and drop all unnecessary object variables.

Table

Description automatically generated

**Create a correlation matrix**

Chart

Description automatically generated

We can see that:

* Year 2022 has higher correlation with the target variable Positive Cases
* Number of Covid19 Tests in the community of Boston has also more correlation with the target
* In order to tune the model we can only pick community-Test feature to predict Positive Cases in Boston area.

**Create descriptive statistics tables**

We createdescriptive statistics tables for all samples and for each subset in the dataset.

**For all samples:**

**A picture containing graphical user interface

Description automatically generated**

**Group by Mask Policy:**

**A picture containing text

Description automatically generated**

**Group By Year:**

Text

Description automatically generated

**Building XGBoost model**

We use XGBoost model because this is the most popular model used in Kaggle. We use it to predict New Positive Cases In Boston randomly. We split the training set and the test set with ratio 8:2. We set up these parameters: n\_estimators = 100, learning\_rate = 0.08, gamma = 0, subsample = 0.75, colsample\_bytree = 1, max\_depth = 10.

Then we evaluate predictions using MAE, MSE, RMSE, R^2. It predicts very well on the training set (R^2 = 99%) and pretty well on the test set (R^2 = 92%).

Text

Description automatically generated Text, letter

Description automatically generated

We also visualize the Feature Importance to predict Positive Cases. Emergency Department Visits, Community Tests, Year\_2022 and No Mask are the key features that most influenced the prediction of positive tests. They fit the current situation in Boston.

Chart, bar chart

Description automatically generated

**Building VAR model to predict new Covid-19 Cases**

We visualize the main features that we use in the VAR model to predict.

Calendar

Description automatically generated

Then we build a function called “grangers\_causation\_matrix()” to Check Granger Causality of all possible combinations of the Time series. The rows are the response variable, columns are predictors. The values in the are the P-Values. P-Values lesser than the significance level (0.05), implies the Null Hypothesis that the coefficients of the corresponding past values is zero, that is, the X does not cause Y can be rejected.

Graphical user interface

Description automatically generated

Next, we build a function called “cointegration\_test()” to Perform Johanson's Cointegration Test and Report Summary.

After that, we split the dataset into a training set and a test set. The training set has 817 observations, while the test set has 15 observations, which is correspondent to the future 15 days.

We build a function called “adfuller\_test()” to Perform ADFuller to test for Stationarity of given series and print report. We see that “Positive Tests”, “International” and “College-Tests” columns are stationary. In constrast, "Emergency Department Visits", "Number of Adult Hospitalizations due to COVID-19", "DOMESTIC", "Boston-Tests" and "Community-Tests" are Non-Stationary. We need to convert them to Stationary before inputting them to the VAR model.

Next, we select the Order (P) of VAR model.

Table

Description automatically generated with medium confidence

Now we train the VAR model with selected Order (P).

Table

Description automatically generated

We check for Serial Correlation of Residuals (Errors) using Durbin Watson Statistic.

Graphical user interface, text

Description automatically generated

We get the lag order and then input data for prediction. After forecasting, we get the result but we need to convert them back to original scale so that we can compare our prediction with the ground truth. To do this, we build a function called “invert\_transformation()”. Finally, we visualize our results and ground truth of different features to see the difference.

Graphical user interface

Description automatically generated with low confidence

We can see that our model predict pretty well variables such as “Positive Tests”, “Number of Adult Hospitalizations due to COVID-19”, “Boston-Tests” and “Community-Tests”. However, it predicts badly on other variables.

We evaluate our predictions using different metrics such as MAPE (Mean absolute percentage error), ME (Mean Error), MAE (Mean Absolute Error), MPE (Mean percentage error), RMSE (Root Mean Square Error), CORR (Correlation), MINMAX. Here are the evaluation results.

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

Text, table

Description automatically generated

**Building SARIMA model to predict new Covid-19 Cases**

We split the dataset into predictor variables and a target variable. The target variable is the “Positive Tests”.

Analyzing the chart, we can observe that the time-series has seasonality pattern. January has a peak of Cases, at least for the last 2 years. There is an upward trend over the years as well.

Chart, histogram

Description automatically generated

Using the “sm.tsa.seasonal\_decompose” command from the pylab library we can decompose the time-series into three distinct components: trend, seasonality, and noise.

Timeline

Description automatically generated

We can see that the Boston Covid-19 Positive Cases Time Series is Stationary. It does not show a clear trend. Its values fluctuate around its mean value.

Now we build the SARIMA model. The model’s notation is SARIMA(p, d, q).(P,D,Q)m. These three parameters account for seasonality, trend, and noise in data.

p = Seasonality, d = trend, and q = noise.

Text

Description automatically generated

We use AIC (Akaike Information Criterion) to estimate the quality of each model, relative to each of the other models. The low AIC means the better model.

Since our time series does not show trend we will set the model order = (1,0,1). After training the model, we get the model as described below.

Graphical user interface, chart, application

Description automatically generated

With the diagnostic above we can visualize important information as the distribution and the Auto correlation function ACF (correlogram). Values upward the “0” has little correlation over the time series data. Values near to “1” demonstrates strongest correlation.

Next, we use the SARIMA model to predict new cases on the test set and compare that to ground truth. We visualize the result using matplotlib library.

Chart, histogram

Description automatically generated

Our model predicts pretty well on the training set.

Now we use it to predict new cases on the test set in 15 days into the future (from 2022/5/29 to 2022/6/13).

Chart, line chart

Description automatically generated

We can see that it predicts pretty well although it does not really match the ground truth line.

Then we use the model to predict new cases in 15 days after 2022/6/13, which we don’t have the dataset. We can see that the predicted line fluctuates and keep under 200 cases.

Chart, line chart

Description automatically generated

…

Chart, line chart

Description automatically generated

**Building the ARIMA model to predict new cases**

We also use ARIMA model to predict new cases. ARIMA model is Auto regressive model. It can be a simple, multiple or non-linear regression model. It’s similar to SARIMA model but it does not focus on Seasonality as SARIMA model.

After training the ARIMA model, we use it to predict the training set again and compare the results with the ground truth. We use matplotlib library to visualize the difference between them.

Chart

Description automatically generated

We can see that the model predicts pretty well on the training set.

Now we use it to predict new cases on the test set in 15 days into the future (from 2022/5/29 to 2022/6/13).

Chart, line chart

Description automatically generated

We can see that the model predicts very well on the test set. The predicted line and the actual line are nearly matched.

Now we use the ARIMA model to predict new cases in 15 days after 2022/6/13, which we don’t have the dataset. The result is shown as below. We can see that the predicted line is pretty similar to the predicted line of SARIMA model. They both fluctuate and keep under 200 cases.

Chart, line chart

Description automatically generated

**Conclusion**

In this final project, we use the Boston Covid-19 dataset collected by the Boston Public Health Commission and the Logan International Airport dataset collected by the Bureau of Transportation Statistics.

We go through different data science steps to preprocess the data, such as Data Collection, Data Cleaning, Data Manipulation, Data Visualization, and Data Reduction.

We build 4 models to predict the new Covid-19 cases 15 days into the future. They are VAR, SARIMA, ARIMA, and XGBoost models. ARIMA model predicts very well. The actual line resembles the prediction line. The outcomes of SARIMA and VAR models are acceptable. The XGBoost model also gives a high R^2 score on the test set (92%).

We use Feature Importance to find out how features affect the new Covid-19 cases. Emergency Department Visits, Community Tests, Year\_2022, and No Mask are the key features that most influenced the prediction of positive tests. They are consistent with the current situation in Boston.

**References**

Health, D. of P., & Executive Office of Health and Human Services. (2022, June 5). *Covid-19 response reporting*. Mass.gov. Retrieved June 5, 2022, from <https://www.mass.gov/info-details/covid-19-response-reporting>

Sharma, H. (2020, October 28). *Visualizing linear, Ridge, and lasso regression performance*. Medium. Retrieved June 5, 2022, from [https://towardsdatascience.com/visualizing-linear-ridge-and-lasso-regression-performance-6dda7affa251](https://towardsdatascience.com/visualizing-linear-ridge-and-lasso-regression-performance-6dda7affa251%20)

Selva Prabhakaran July 7, 2019 Vector Autoregression (VAR) – Comprehensive Guide with Examples in Python <https://www.machinelearningplus.com/time-series/vector-autoregression-examples-python/>

Selva Prabhakaran August 22, 2021 ARIMA Model – Complete Guide to Time Series Forecasting in Python <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>

Raphael Bubolz Larrosa Jul 15, 2019 How to forecast sales with Python using SARIMA model <https://towardsdatascience.com/how-to-forecast-sales-with-python-using-sarima-model-ba600992fa7d>