Predicting COVID-19 Cases Leveraging Sentiment Analysis From Twitter Data

University of Toronto
MIE 1516: Structural Learning and Inference
Armando Ordorica

April 13, 2020

Abstract

The goal of this project is to analyze the evolution of Covid-19 cases and determine whether adding sentiment analysis data from Twitter improves the prediction of daily cases rather than using an autoregressive model on the signal itself. For consistency, our analysis will focus on New York City area only in an attempt to isolate other external variables driven by difference in geo-location and habits pertaining to Twitter usage. The end objective is to establish a framework that could help health officials and policy makers in predicting the upcoming number of cases in each region.

1 Time Series Forecasting with ARIMA

ARIMA, short for 'AutoRegressive Integrated Moving Average', is a forecasting algorithm based on the idea that the information in the past values of the time series can alone be used to predict the future values [1]. Any 'non-seasonal' time series that exhibits patterns and is not a random white noise signal can be modeled with ARIMA. We will use this model to predict the number of daily cases of Covid-19 data in New York City area using only information found in the signal.

The data was obtained from this repository, which feeds the 2019 Novel Coronavirus Visual Dashboard operated by the Johns Hopkins University

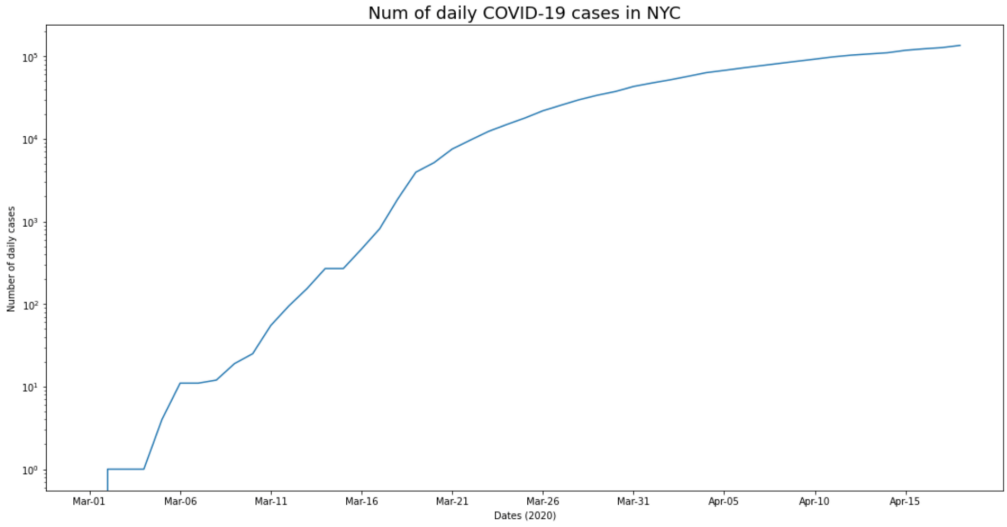

Figure 1: Chart showing number of daily COVID 19 cases in NYC.[2]

Center for Systems Science and Engineering (JHU CSSE). The date range is March 1st, 2020 to April 18th, 2020 [2].

As we can see in the chart below, the number of daily cases does not seem too noisy, so we expect the signal to be relatively straightforward to predict.

1.1 Hyperparameter Tuning for ARIMA

There are three main parameters to tune: p, which is the order of the AR term, q the order of the MA term and finally d, which is the number of differencing required to make the time series stationary. As we can see on the autocorrelation plots shown in Figure 2, the autocorrelation function approaches the shape of the delta function at an order of differencing $d = 2$.

To achieve stationarity in the time series, I have set $d = 2$. I set $p = 1$, which refers to the number of lags of Y to be used as predictor. As shown on Figure 3, there were very few points outside the significance area in first order differencing, therefore $p = 1$ seemed adequate.

Finally, I used $q = 0$ which represents the number of lagged forecast errors that should go into the ARIMA Model.The autocorrelation function informs

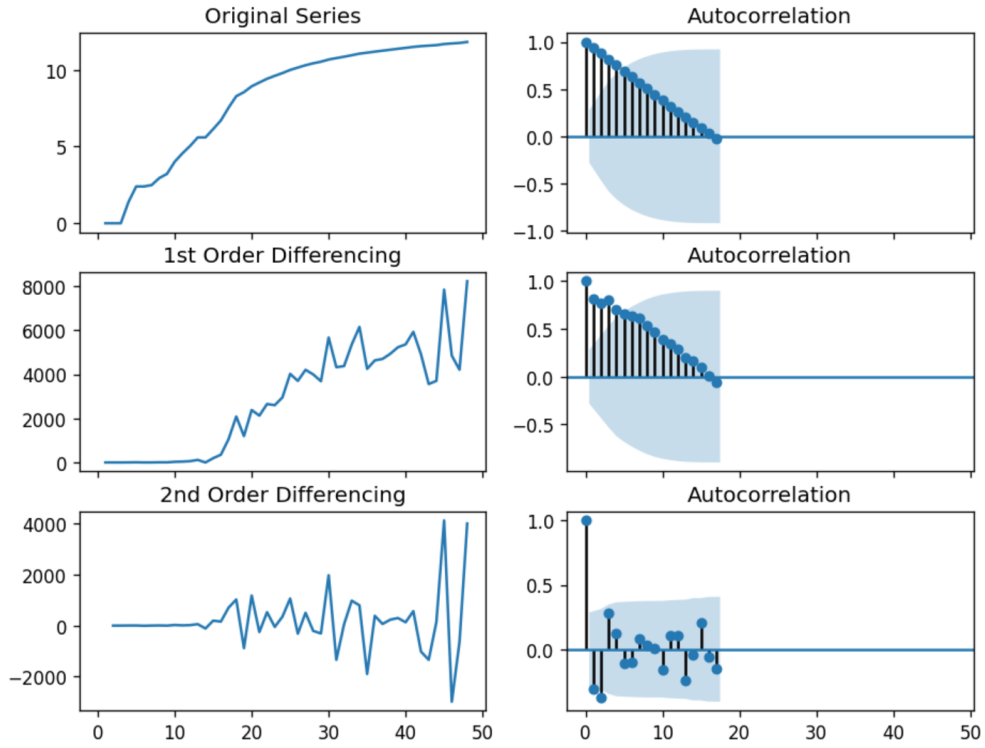

Figure 2: Autocorrelation functions used to determine the order of differencing d.

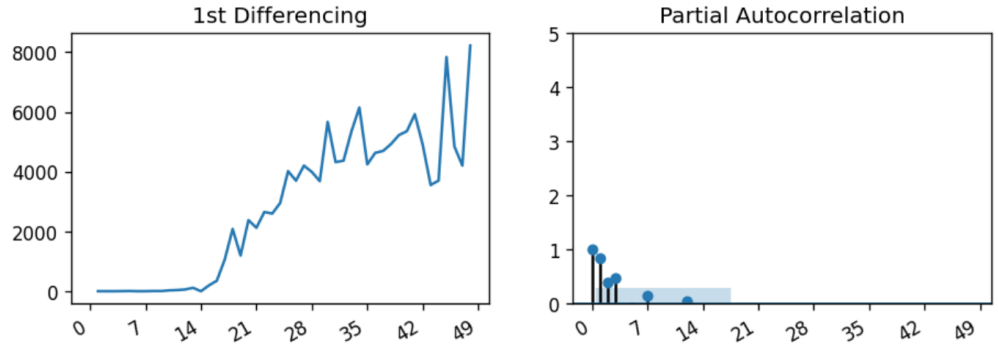

Figure 3: Partial autocorrelation function used to determine the order of the autoregressive component p.

how many moving average terms are required to remove any autocorrelation in the stationarized series as shown on figure 4.

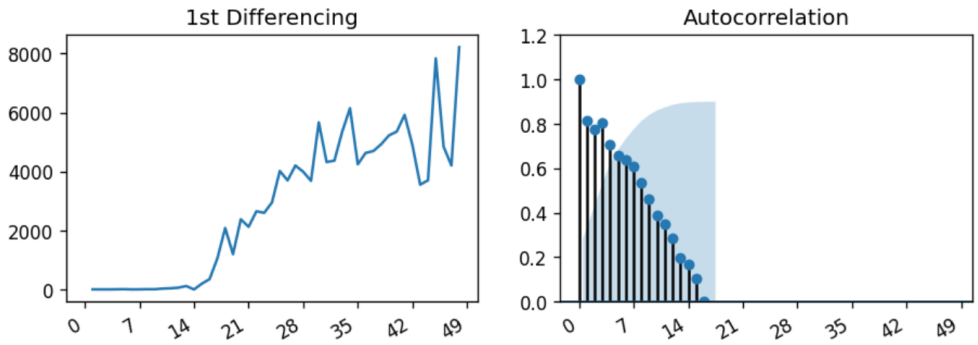

Figure 4: Partial autocorrelation function used to determine the order of the moving average component q.

For more details on the choice of p, q, and d, please consult the comments and the autocorrelation plots available in the Jupyter notebook attached.

Figure 5 shows the summary of the model with $order = (1, 2, 0) = (p, d, q)$.

As shown on figure 6, the mean seems constant around 0 and the variance seems uniform, which means that the parameters were set correctly to adjust for stationarity in this process.

1.2 Results and Validation

The table shown on figure 7 compares the log of predicted vs observed values. In addition, the cumulative mean square loss is plotted on figure 8.

We can see that even for far ahead values (10 out of sample values), the error increases, yet remains low. It is expected that error increases with time since it is a function of the variance, which also increases with time. This is because the further away the values are, the less auto-correlated they will be to the function under study.

ARIMA Model Results

Dep. Variable:	D2.y	No. Observations:	36
Model:	ARIMA(1, 2, 0)	Log Likelihood	-295.879
Method:	css-mle	S.D. of innovations	892.002
Date:	Mon, 20 Apr 2020	AIC	597.757
Time:	16:34:43	BIC	602.508
Sample:	2	HQIC	599.416

	coef	std err	z	P>|z|	[0.025	0.975]
const	4.8247	93.363	0.052	0.959	-178.163	187.812
ar.L1.D2.y	-0.6094	0.126	-4.818	0.000	-0.857	-0.361

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-1.6410	+0.0000j	1.6410	0.5000

Figure 5: Summary of ARIMA model results

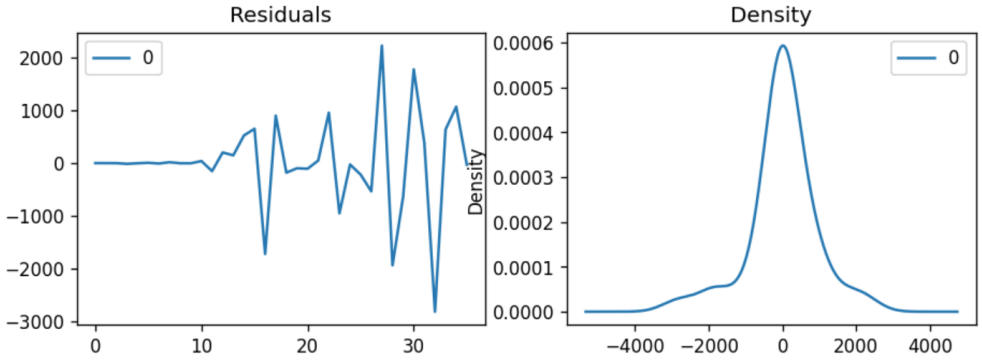

Figure 6: Mean and variance on residual plots.

	true_values	predicted_values	mse_loss	cum_mse_loss
Apr-09	4.939659	4.938860	6.387048e-07	6.387048e-07
Apr-10	4.965597	4.964430	1.362380e-06	2.001085e-06
Apr-11	4.992589	4.989340	1.055700e-05	1.255808e-05
Apr-12	5.013713	5.013734	4.136637e-10	1.255850e-05
Apr-13	5.028421	5.037586	8.399611e-05	9.655460e-05
Apr-14	5.043225	5.060961	3.145871e-04	4.111417e-04
Apr-15	5.072992	5.083868	1.182888e-04	5.294306e-04
Apr-16	5.090420	5.106343	2.535266e-04	7.829571e-04
Apr-17	5.105006	5.128401	5.473485e-04	1.330306e-03
Apr-18	5.132170	5.150068	3.203324e-04	1.650638e-03

Figure 7: Chart comparing log values of predicted vs observed values on out of sample prediction

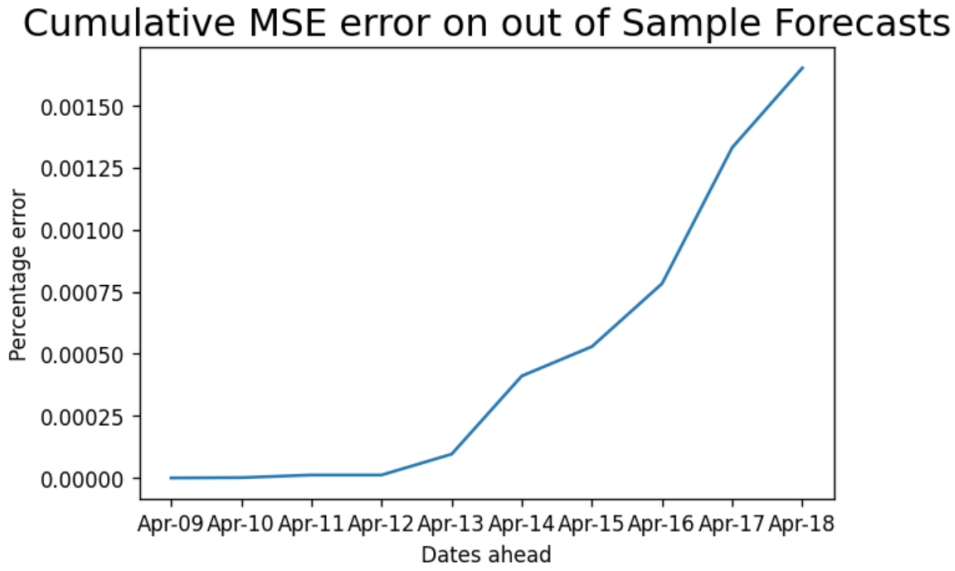

Figure 8: Cumulative MSE error ARIMA

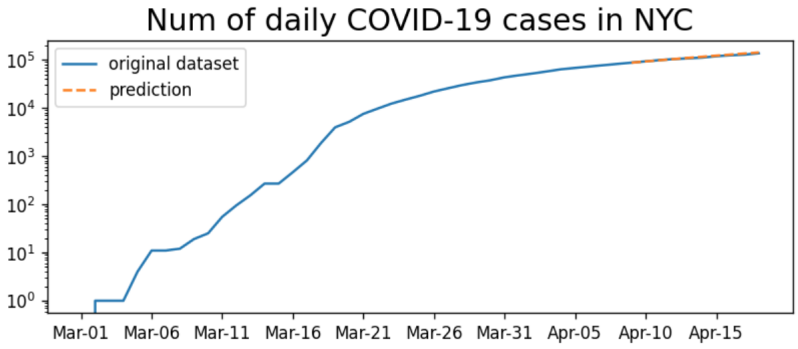

Figure 9: Out of sample prediction of 10 days

Finally, overlapping the out of sample prediction with the original dataset, we can see that the prediction matches closely the observed values as shown on 9.

2 Twitter Sentiment Analysis

The question that this section aims to answer is whether more cases leads to more Tweets or vice-versa and whether Twitter data can be used to approximate the number of daily cases of new COVID-19 infections.

In this section, we classify Tweets in three main categories: positive, neutral, and negative sentiments. These categories were chosen to simplify classification and achieve better model stability given that the free data available was limited (only a few thousand tweets per day in NYC area). After classifying Tweets in one of those three categories of sentiments, we generate time series to plot the count of those sentiments over time.

2.1 Definition of Base Population

The dataset was obtained from Kaggle [4] and contains the Tweets of users who have applied the following hashtags: `#coronavirus`, `#covid19`,`#covid_19`, `#coronavirusoutbreak`, `#coronavirusPandemic`, `#epitwitter`, and `#ihavecorona`. Note that the date range for this dataset is more limited and only ranges from 2020-03-04 to 2020-03-28 as opposed to the WHO data

which ranges from 2020-03-01 to 2020-04-18 used for the section above for the daily COVID-19 cases in NYC.

To filter for New York City area, I only kept values where the `place_full_name` column contained the values `['New York', 'NY', 'ny', 'NYC', 'nyc']`. Following this approach, boroughs of NYC such as `Queens, NY` and `Brooklyn, NY` where included.

It is worth noting that about 90% of the Twitter users don't share their location with Twitter, which drastically reduced the data availability to only a few thousands of Tweets per day.

2.2 Results

The word cloud on Figure 10. As expected, the most common words are highly relevant to the COVID-19 pandemic and describe the emotional aura of the situation.

To obtain the polarity of each Tweet, I used the Python library `TextBlob`, which is a library for processing textual data [3]. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

`TextBlob` classifies each string of text (Tweets) in positive, negative, and neutral based on their polarity and subjectivity. If the polarity is > 0, it is considered positive, < 0 is considered negative, and $== 0$ is considered neutral.

The sample of Tweets shown on Figure 11 were classified as `Positive`. We can see that words such as `awesome, best, friend, perfectly, proud, happy` are some of the main nouns that led the model to classify them as such. Likewise, we can see a sample of the negatively classified tweets on Figure 12. We can see that words such as `disgusting, insane, hand_covering_face_emoji, panic, horrific` and `sick` were key contributors for the corresponding Tweets to be classified as negative.

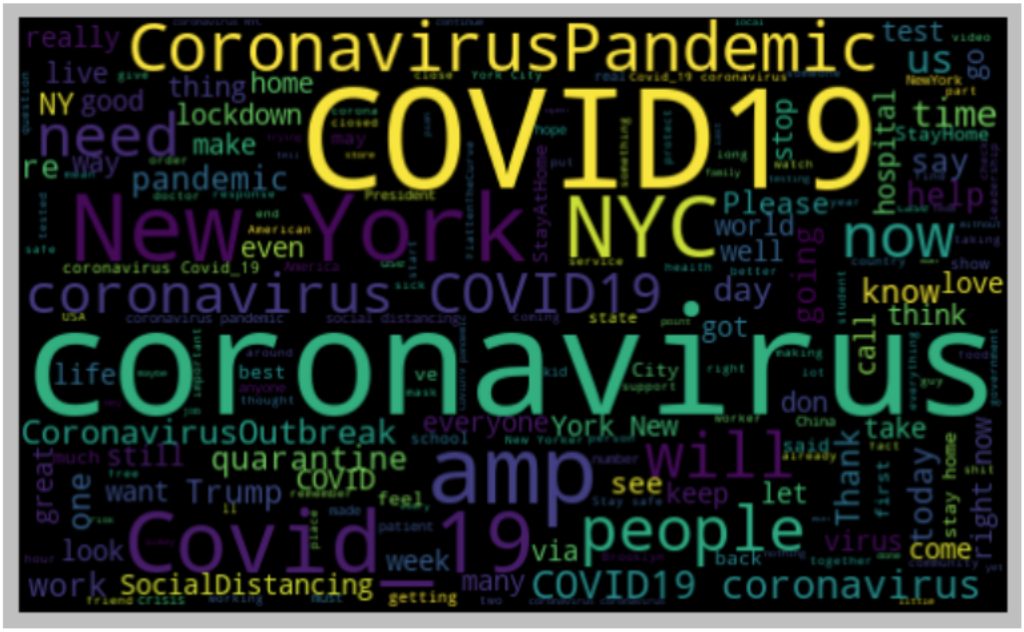

Figure 10: Word cloud for Tweets on New York City area related to the COVID-19 pandemic.

1) are awesome! COVID19

2) Is this the best decision to move ahead with meeting? Cc COVID19 coronavirus

3) A Doctor friend passed this along and suggested I share.

It illustrates perfectly how the virus infects you and answers a lot of questions you may have.

CoronavirusPandemic

4) Mother knows best! coronavirus TrumpRecession CoronavirusOutbreak

5) Proud to organize with my students! sickout closenyschools CloseTheSchools coronapocolypse coronavirus COVID—19 COVID19 NYCDOE

6) One of the best Memes of the WuhanVirus CoronaVirus ChineseFlu 👇👇👇👇👇

7) We should require everyone to read/watch this before their phones will work again. That would be the best national emergency alert system eve

8) Happy to announce that 46 will be joining us for pancchat tonight! Tune in at 8pm EST to tweet all things COVID19 CoronavirusOutbreak covidnc

9) This seems like a perfect time to separate and split NYS into the proposed 3 districts! COVID19 4NY

Figure 11: Sample of Tweets with sentiment classified as "Positive".

On aggregate, 47% of Tweets were positive, 20% were negative, and the remaining 33% was classified as neutral. The histogram on Figure 13 shows the aggregate distribution over sentiments for the Tweets considered within the specified time range. Furthermore, Figure 14 shows the number of Tweets by sentiment over time. We can see that after an inflection point

1) SUNY CUNY you guys literally DONT CARE about the safety of your students. This is absolutely DISGUSTING.

Why does it need to affect a STUDENT for you guys to CloseTheSchools CoronavirusOutbreak coronaVirus Coronavirusny

2) Geraldo Rivera Clashes With Dan Bongino Over Trump Travel Ban...These ppl are insane! coronavirus

3) Yyyooooooo y'all motherfuckers are RUTHLESS 😭😭😭!!!!!!

🤦🏾‍♀️🤦🏾‍♀️🤦🏾‍♀️ coronavirus

4) This is just disgusting. COVID19 theranos SoftBank patenttroll

5) This is outrageous & U wonder why there is panic over COVID19... StayHome StopTheSpread FlattenTheCurve WashYourHands DontPanic YouGott

6) In Syria's prisons, 10000s languish in horrific humanitarian conditions. Even one case of COVID19 there would be catastrophic. That's why h

7) what's your plan for when employees get sick!!! No plan for paid time off, no safety plans in place. Employees with coughs are told they s

8) 🗣 shout out to all you nasty ass people that don't wash your hands
CoronavirusPandemic

9) Make it so I don't have to go to work, but then close all the bars...
You're a cruel plague, Covid. You're a cruel plague.

coronavirus COVID19 panic relax work drink breathe plague

Figure 12: Sample of Tweets with sentiment classified as "Negative".

around 2020-03-09, the positive and neutral sentiments started to dominate the Tweets with negative polarity.

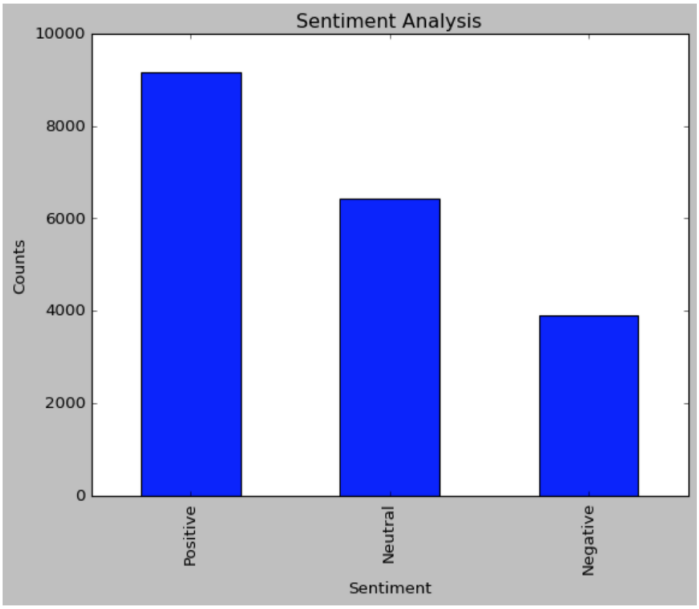

Figure 13: Distribution of sentiment classification of Tweets related to the COVID-19 pandemic in the NYC area between 2020-03-04 and 2020-03-28.

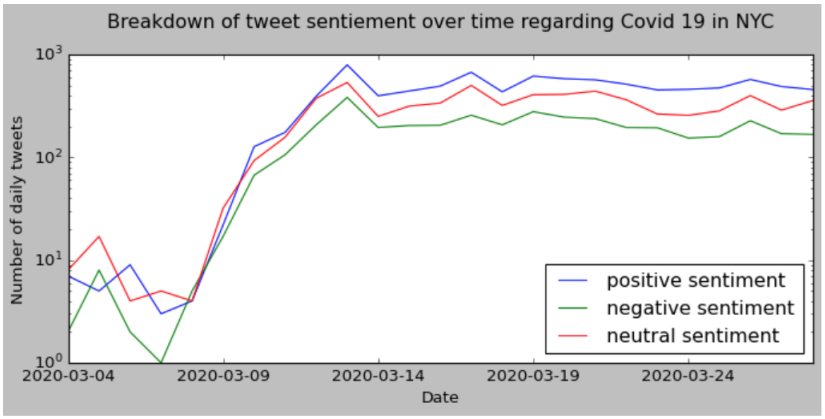

Figure 14: Time series breakdown of sentiment evolution over time related to the COVID-19 pandemic in the NYC area between 2020-03-04 and 2020-03-28.

3 Regressing time series of daily cases with Twitter data

The goal of this section is to determine whether feeding the time series from the Twitter sentiment analysis can improve the prediction on daily COVID-19 cases for New York City area.

3.1 Data Cleaning

As we can see on Figure 15, the number of Tweets increased exponentially at a similar rate together with the number of cases in New York City area.

A list of random numbers was created to randomly split train and test data. I decided to use this approach, also known as *bagging*, to make this model more generalizable and avoid overfitting given that there were only 25 data points available (2020-03-04 to 2020-03-25).

I normalized the data by subtracting the mean and dividing by the standard deviation of each feature. Normalized data leads to more stable gradient flow due to less volatility in the elements of the weight vector.

date	positive tweets	negative tweets	neutral tweets	Num of Cases in NY
2020-03-04	7	2	8	1
2020-03-05	5	8	17	4
2020-03-06	9	2	4	11
2020-03-07	3	1	5	11
2020-03-08	4	5	4	12
2020-03-09	22	17	32	19
2020-03-10	127	67	93	25
2020-03-11	175	106	157	55
2020-03-12	393	207	375	95
2020-03-13	794	383	535	154

Figure 15: This figure shows a snapshot of the schema of the table containing merged data from Twitter sentiment analysis and number of daily COVID-19 cases in NYC area.

As shown on Figure 16 the input variables were `['positive tweets', 'negative tweets', 'neutral tweets']`. Further, the target variable we were trying to infer from the input variables was `['Num of Cases in NY']`.

3.2 Results

Fitting a linear regression to these predictions did not yield great results as can be shown on Figure 17. At least not good enough to beat the ARIMA model.

Even though the number of COVID-19 related Tweets seemed to be increasing exponentially on par with the number of COVID-19 cases in early March, there was a point where the number of COVID-19 cases grew at a much higher acceleration rate than the number of Tweets. This is clearly seen on the MSE loss plot on Figure , where the loss exponentially increased past mid-March. Even though the number of Tweets grew, the number of users prone to tweeting about COVID-19 saturated much quicker than the

```
X_train = train[['positive tweets', 'negative tweets',
       'neutral tweets']]
X_train =  X_train.sort_index()
X_test = test[['positive tweets', 'negative tweets',
       'neutral tweets']]
X_test =  X_test.sort_index()
```

```
Y_train = train[['Num of Cases in NY']]
Y_train =  Y_train.sort_index()
Y_test =  test[['Num of Cases in NY']]
Y_test =  Y_test.sort_index()
```

Figure 16: Snapshot of the Python code showing the choice of input and output variables.

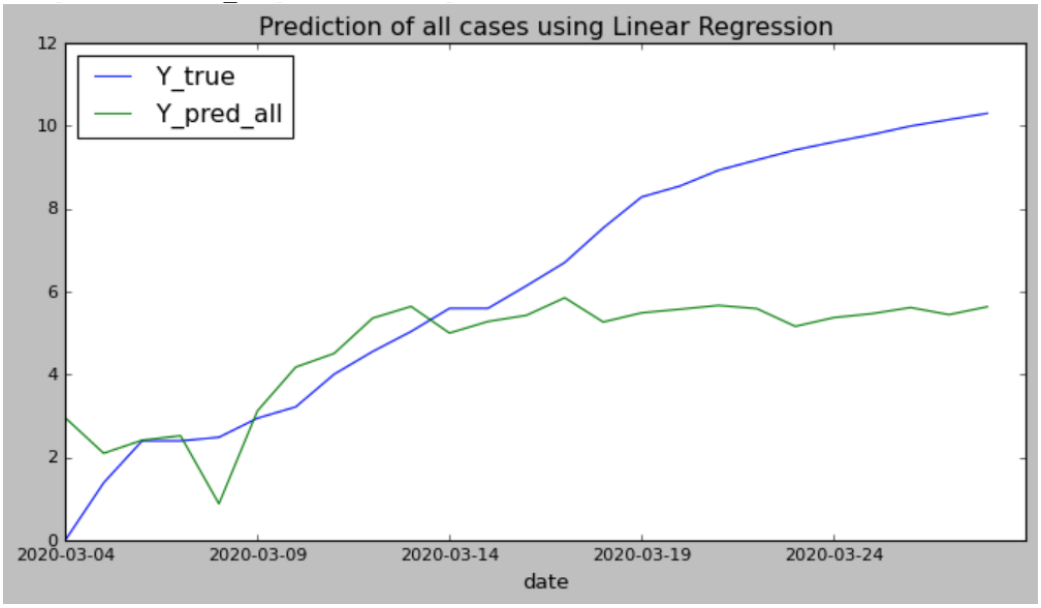

Figure 17: Log Scale plot comparing the predicted values with observed values as outputted by the linear regression model.

number of cases.

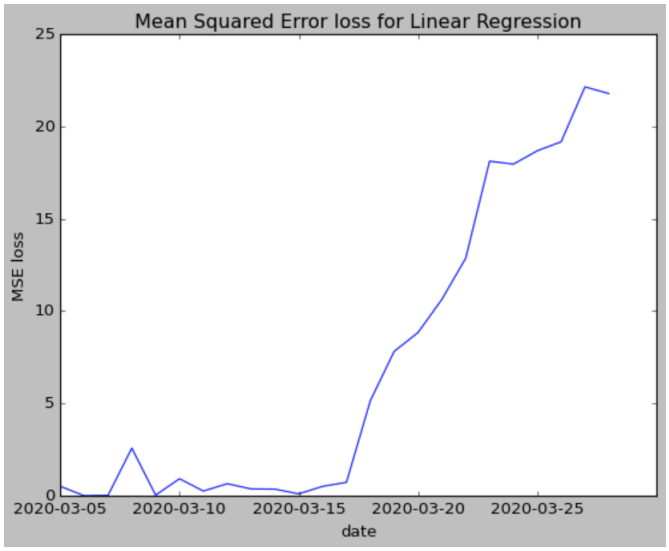

Figure 18: Mean Squared Error Loss function for Linear Regression

4 Conclusion

An autoregressive model such as ARIMA is adequate to model epidemiological data. Autoregressive models are recommended when the process is not cyclic and wide-sense-stationary with purely Gaussian noise. COVID-19's epidemiological data seemed to adhere to these conditions and the ARIMA model yielded favourable results with an elegant balance between simplicity and accuracy.

The regression on the time series from the sentiment analysis performed on Twitter data did not add anything valuable. Even though tweets and number of cases grew at a similar rate for early march, the number of cases picked up acceleration and escaped to divergence around the 14th of March, which was hard for a linear model to capture.

Further recommendations include using a more complex model such as a multilayer perceptron and include nonlinearities to model the relationship between number of tweets and number of cases. A nonlinearity would be required to detect the switchpoint that occurred near 2020-03-15 and adapt the weights accordingly. However, it was not deemed necessary to improve

this analysis given that the results from the ARIMA model were already very accurate.

References

[1] Machine Learning Plus *ARIMA Model – Complete Guide to Time Series Forecasting in Python* 2018

[2] John Hopkins Repo WHO data *John Hopkins University Github Repository for COVID-19 data from World Health Organization* 2020

[3] TextBlob Python Package *TextBlob: Simplified Text Processing — TextBlob 0.15.2 documentation* 2020

[4] Kaggle Covid-19 Twitter dataset *Coronavirus (covid19) Tweets* 2020

[5] Github link to source code *Armando Ordorica* 2020