Final Project Report:

Forecasting COVID-19 Pandemic in America

CS 4774-002

Fall 2020

Allen Lang (al8he)

I would like to thank Professor Qi and teaching assistants Jack, Zhe, and Arshdeep for their guidance and assistance for this project and for this semester.

Motivation

As winter weather sets in, holiday season begins across the World. This is a time of increased travel, family gatherings, and celebrations relative to the rest of the year. However, the COVID-19 pandemic has already dramatically altered the lives of everyone around the world, and it will no doubt change how people celebrate this upcoming holiday season. The purpose of this project is to analyze how major holidays and events have already been impacted by the virus and formulate forecast predictions of COVID-19 cases during this time period.

Background

Of the public celebrated holidays in America, the two most popular are occurring before the end of this year: Thanksgiving and Christmas. Although it was originally a Christian holiday, people celebrate it for non-religious purposes as well: approximately 9 out of 10 Americans celebrate the event in 2017[1]. For Thanksgiving, 85% of Americans celebrate it[2]. It is a time of personal and economic importance, as people spend on average $998 on gifts to give friends and family at gatherings[3]. Although both can bring joy to people, outside circumstances can dampen the holiday mood. During the Great Recession, the last period of global turmoil, a survey showed that around 75% of Americans felt that economic troubles made them celebrate Christmas less happily[4]. The COVID-19 has and will be another source of stress.

Due to its highly contagious nature and long incubation period, attempts to contain the spread of the virus have been difficult. In the United States, restrictions to foreigners and foreign travel began in February and March of 2020. Domestically, states began applying lockdowns (or stay-at-home orders) in March, closing non-essential institutions like gyms, restaurants, and movie theaters. Many schools and businesses instituted remote work programs. For the rest of this current year, the contentious topic over the pandemic has been whether or not to lift these precautionary restrictions. States began the re-opening process in June in preparation for summer, but a recent rise in cases domestically and in Europe has brought back discussions about another lockdown or travel restrictions.

Travel typically rises during this holiday, as people go home to visit family or go on individual trips. Based on a 2001 survey, 90% of all travelers used personal vehicles and 5 to 6% used air travel[5]. Air travel can be monitored through the Transportation Security Administration

(TSA) checkpoint counts, while the Department of Transportation (DOT) keeps track of travel data in its Bureau of Transportation Statistics[6,7]. Governments will keep track of these numbers when deciding on COVID-19 responses during the next couple of months. The Center for Disease Control (CDC) has already warned that the virus spread faster indoors, which is a place many people will be spending during the holidays and winter[8]. No one knows yet how much the seasonal change will exacerbate the pandemic, and this has led to uncertainty for everyone: from governments debating re-introduction of unpopular restrictions to people debating proceeding with annual holiday traditions.

Time-series data is a field that is high in demand but lacks a large pool of experts comfortable with generating statistical models. Forecasting is the act of generating predictions with a time scale and with these time-series data. Common examples are weather forecasts and trends in energy consumption that inherently have a periodic seasonality relative to time[9]. Contemporary models like ARIMA and exponential smoothing can be difficult to properly tune, and some require the data to be stationary (an average of the data that remains relatively constant). These two issues were the main reason why Facebook Core Data Science Team created the Prophet library for Python. It simplifies the construction and tuning process while ensuring that its models can handle idiosyncratic data[10].

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{1}$$

From documentation, the Prophet model is broken up into four different components that are shown in Equation 1: trend $g(t)$, seasonality $s(t)$, holidays $h(t)$, and white noise ϵ_t. Trend represents the non-periodic changes, and its growth is either linear or logistic. This is accomplished with changepoints, which break up the time model into a piecewise function. Increasing changepoints will increase the number of subfunctions and increase the ability to fit a model, but may lead to overfitting from overcomplexity. Seasonality represents periodic changes and is modeled with a Fourier Series. Prophet allows for seasonality of varying periods; some examples are daily, weekly, and annually. Holidays represent irregular events that could provide shocks to the time series model. These do not necessarily have to be officially recognized holidays (although Prophet has built-in country holidays), and the user may add special events. The last term is an error term that accounts for anything that does not belong to the other three and is assumed to be normally distributed[10].

Related Work

The COVID-19 pandemic has generated an influx of data sources and efforts to build interpretations of these values. Every state now has its own COVID-19 tracker that monitors case numbers, deaths, hospitalizations and other numbers by geography and socioeconomic status. News sites like the New York Times dedicate special sections specifically for COVID-19, offering case count trackers, news updates, and progress on vaccine development. The website COVID Exit Strategy takes in quantitative data and provides qualitative evaluations of the pandemic in each state[11].

Forecasting COVID-19 is the process of using past and current trends to predict future case numbers. The CDC has partnered data analytics teams from universities and companies to produce models that attempt to forecast the spread of COVID. As of now, 47 teams provide models for deaths forecasting, 32 provide for case forecasting, and 12 for hospitalization forecasting. All forecasting models make predictions for up to 4 weeks into the future. A common model used in each has been the SEIR, a compartmental method used to simplify infectious disease modeling. This particular model is helpful because of COVID’s long incubation period, as it classifies people into susceptible, exposed, infectious, and recovered[12].

Claim/Target Task

The claim for this project is that, during holiday times, COVID-19 spreads faster than normal. The task will be to observe the pandemic in the United States during holidays such as Memorial Day and Fourth of July, and to derive any possible quantitative results from the analysis. Potential analysis may also look at large gatherings, such as the Minnesota George Floyd protests.

Intuitive Figures

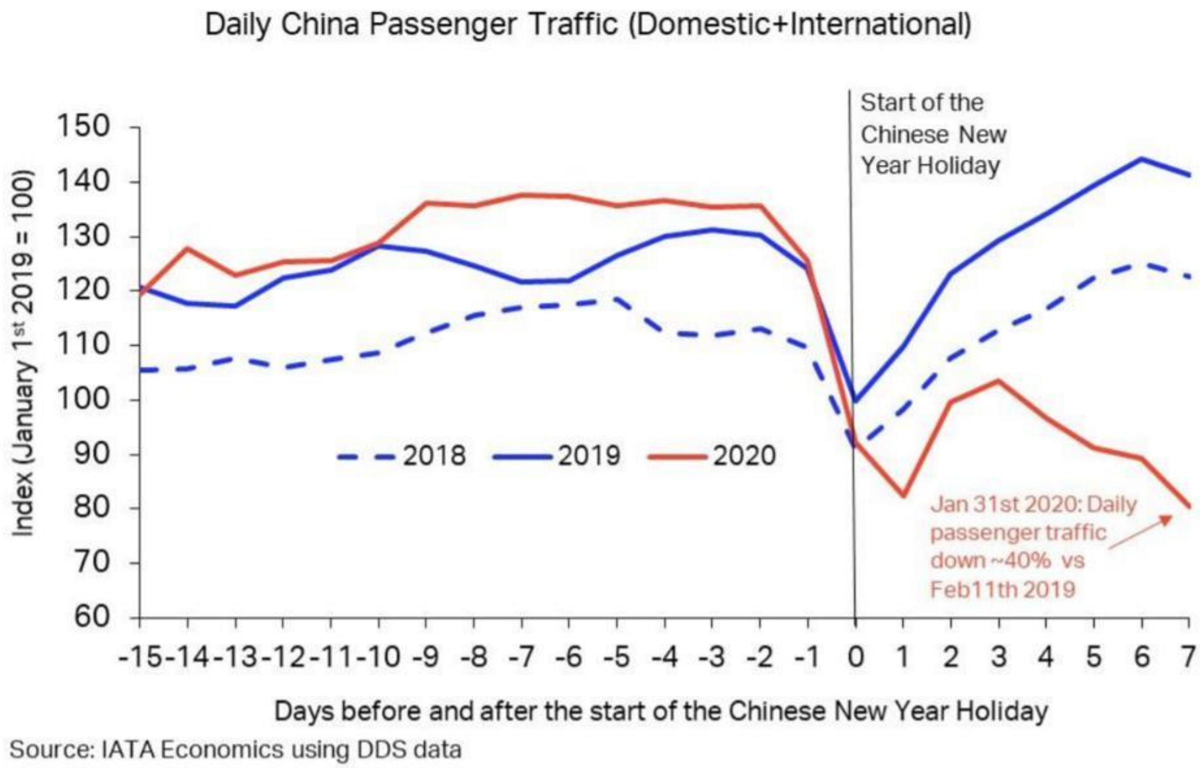

Figure 1. Normal passenger traffic during Chinese New Year Holiday compared to 2020, shortly before beginning of lockdown[13].

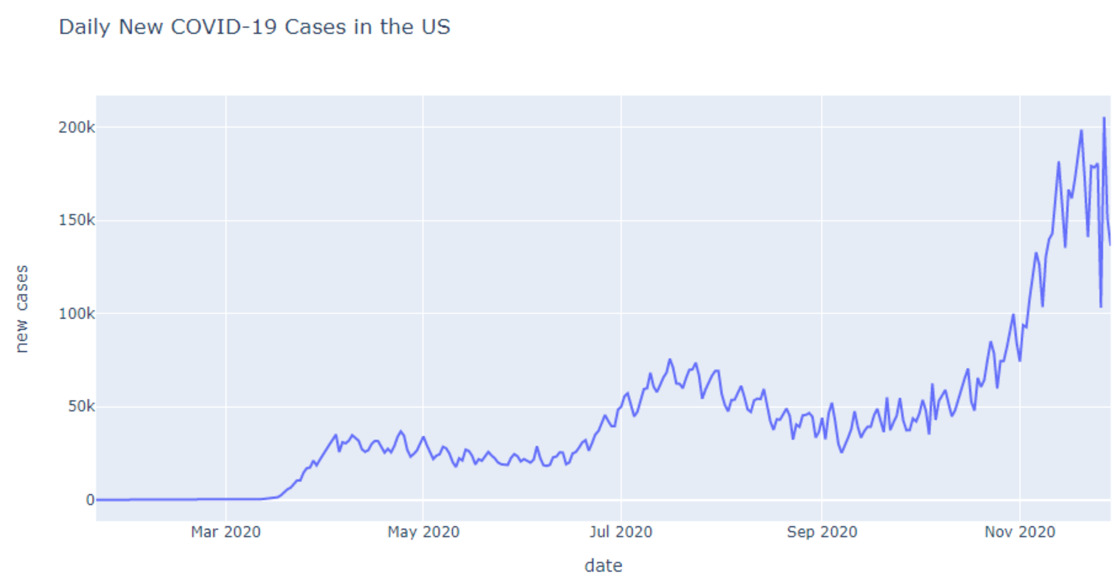

Figure 2. Number of new cases in the United States up to end of November. As the weather becomes colder, cases have begun rising across the country[14]. How should governments respond to this and the upcoming holiday season?

Proposed solution

The current plan of action is to use Pandas module and analyze current COVID-19 data sets for the United States as a whole. There exist many datasets of this kind, so the data specifically from New York Times will be. Google Mobility Data will be used to estimate travel during pre-pandemic and post-pandemic periods, and it features usage rates of grocery stores, hospitals, and recreational areas[15]. This will also be used to quantify movement of Americans during holiday seasons. A list of holidays celebrated in America will also be used to finetune the model[16]. Finally, Oxford's indices on government responses to the pandemic will be used as an additional parameter[17]. Prophet will be used to forecast the predictions for future holidays[18]. Prophet provides data visualizations, but Plotly Express will also be used for its plots[18,19].

Contributions

1. Build/tune model that accounts for holidays and special when forecasting COVID-19 cases.
2. Better understanding of various factors like mobility, holidays, government response can contribute to the new COVID-19 cases

The hope is that such a model can contribute to the community by better showing effects of holidays and special events on COVID-19 spread. Additionally, it would also like to clarify the effect that other time-series data like mobility trends can have on time-series forecasting.

Implementation

The following list is an overview of the code implementation.

1. Data Cleaning and Formatting
2. Baseline model
3. Hyperparameter Tuning
    a. Part 1: Without holidays
    b. Part 2: With national holidays
    c. Part 3: With national holidays and special events
    d. Part 4: With national holidays and special events (manual tuning)
    e. Part 5: With regressors
4. Prediction

Data Cleaning and Baseline Model

The first step is to import the data sets (which will be discussed below) and format DataFrames. Some datasets may have missing values and must be handled properly. Formatting

DataFrames is important because Prophet only takes in a certain type of DataFrame into its model. The time column must name "ds" and be a pandas datetime object; the output column must be named "y". The second step is to split the data into train and test sets, and train a Prophet model with the default parameters on the training data. The root mean squared error (RMSE) is found for the training and testing set from the subsequent forecasted model. This is a common metric used in time-series analysis[22]. This will be used for comparison in step 4.

Hyperparameter tuning

The third step is hyperparameter tuning, and involves five parts. Each will create another train test split for validation purposes using RMSE of testing set as a metric. Part 1 tunes the model without holidays, and focuses on parameters that influence trend and seasonality. Changepoints are areas in the graph that vary in trend, and its scale can be modified such that the model is more or less flexible at each changepoint. The same can be done for cyclical seasonality. Since Fourier series represent periodic seasonality, the Fourier series' order can be tuned as well. The best model is taken as the one with the lowest test RMSE, and its train and test root mean squared error (RMSE) are found.

Table I

Hyperparameter Grid for Part 1 of Tuning

Hyperparameter	**Description**	**Range of Values**
Seasonality	type of seasonality	['additive','multiplicative']
Seasonality scale	flexibility of seasonality	[0.1,1,10]
Changepoint scale	flexibility of trend between changepoints	[0.1, 0.2, 0.3, 0.5]
Fourier order	order of the Fourier series used for seasonality	[6, 8, 10, 12]

Part 2, 3, 4 looks at holidays. All three use the best model from part 1. Part 2 only takes national holidays celebrated in the United States from a Github repository. Part 3 adds in special events such as Election Day, Black Lives Matter protests, and periods in which the quarantine was lessened. The last event was determined by using Oxford's stringency data and determining days in which the stringency index dropped. Both holiday lists are shown in the Appendix. Part 4 is an extension to part 3, and attempts to manually tune the hyperparameters for part 3's set of holidays. All three sections will have their best RMSE taken based on the testing set. Of the three, the best one is will have its hyperparameters used for part 5.

There are three holiday hyperparameters. Like seasonality and changepoint, holidays also have a scale parameter that quantifies flexibility and significance of a holiday towards model forecasting. Lower and upper window adjust how far the holiday effect will be accounted for in the model by measure of days. For example, a lower window of -2 would mean that the effect of the holiday extends two days back. An upper window of 2 would mean the effect extends two days forward.

Table II

Hyperparameter Grid for Part 2, 3 of Tuning

Hyperparameter	Description	Range of Values
Lower window	How far holiday effect goes backwards	[-1, -2, -3, -4]
Upper window	How far holiday effect goes forwards	[1, 2, 3, 4]
Holiday Priority scale	Flexibility of holiday event	[0.05, 1, 5, 10]

Part 5 looks at regressors, specifically other time-series data that can be made to help forecast COVID-19 cases. This will include six metrics from Google's mobility report and an additional metric, the stringency index mentioned above. Regressors are treated similar to seasonality, and the priority scale and the mode (additive/multiplicative) can be changed. This step of tuning tried each regressor one at a time. Afterwards, the best of each of the seven were manually tuned to find the best combination from that set. The best model was found using RMSE of testing set.

Table III

Hyperparameter Grid for Part 2, 3 of Tuning

Hyperparameter	Description	Range of Values
Type	Type of regressor	[-1, -2, -3, -4]
Upper window	Data used	['stringency', 'retail', 'residential', 'work', 'park', 'grocery', 'transit']
Regressor Priority scale	Flexibility of regressor	[1, 5, 10]

The fourth and final step is to look at the performance of the model on the original train and test split using the best hyperparameters from each step. The test RMSE will then be compared with that of the baseline model.

While some notebooks have used Prophet for COVID-19, the novelty of this method is that it incorporates many factors using the regressor feature. It also attempts to crudely fine tune holiday parameters using human intuition. For example, holidays on Fridays should have upper windows extending to the weekends. It also attempts to observe the effect of social movements such as Black Lives Matter and how it has impacted the COVID-19 pandemic. Lastly, it incorporates Google mobility data as a way of capturing the movement of people during the pandemic.

Technical challenges to the implementation involved getting used to working with time-series analysis because there was no prior experience with it. Another issue was getting used to the different Python libraries used, specifically Plotly and Prophet. A notable challenge for Prophet was finding a suitable method for validation. Prophet offers a cross validation method for time-series data, but the process itself was not understood well enough to justify its use. As such, the implementation had to be varied so that a simple train-test split was used instead.

Data Summary

Four data sets were used for the project. All data sets used were real world data sets or attempts to quantify real world phenomenon such as government response. Simulated data was not chosen as the purpose of study was to test the ability to predict and forecast into the future using given data. Simulated data would have complicated the study by adding assumptions, which runs counter to the contributions of this study in being easy to understand.

The first was New York Times' COVID-19 case and death count for the United States. There existed COVID-19 cases by state and by county, but the main dataset used was the case count that aggregates all of the states' cases into one single number for a given day. The dataset also includes a survey of mask usage per county, but that was not used either as the survey only lasted for the first week of July[14]. The second dataset is from a Github repository that keeps track of a list of holidays that each country in the world celebrates[16]. The data was filtered to only give a list of US holidays.

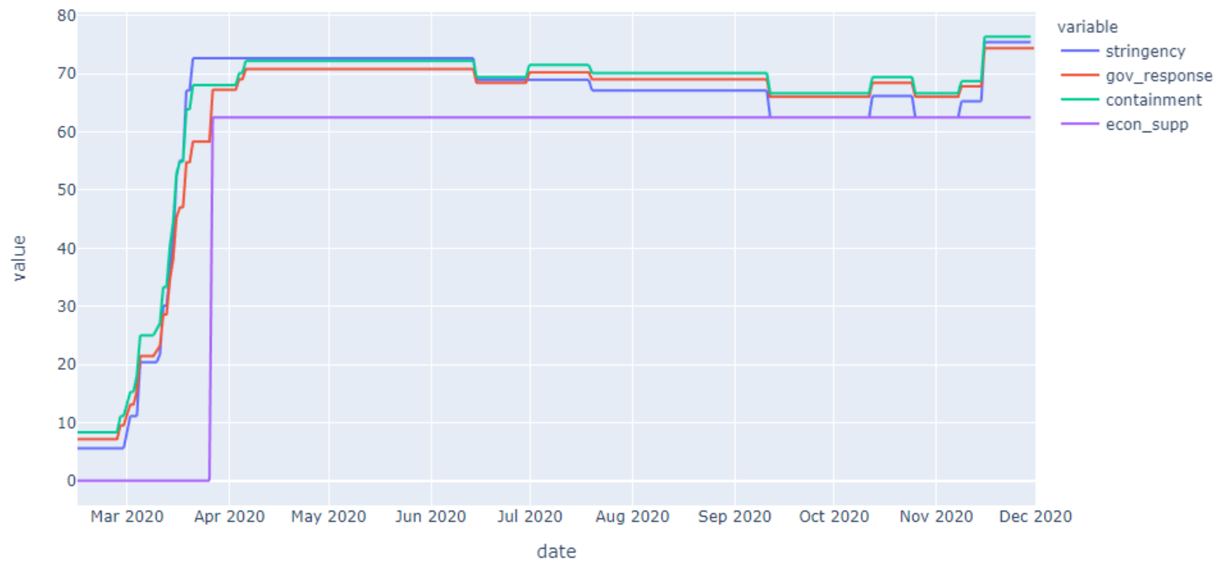

Figure 3. Four main metrics from Oxford's Government Response Tracker. Since all four seem to be closely correlated, only the stringency variable will be used since it is similar to the other three[17].

The third dataset is from University of Oxford's Blavatnik School of Government. A group at the school has developed the Oxford COVID-19 Government Response Tracker. It specifically tracks a government's overall response, stringency of lockdowns, economic support (stimulus relief packages), and containment (ability to test and contact trace) metrics[17].

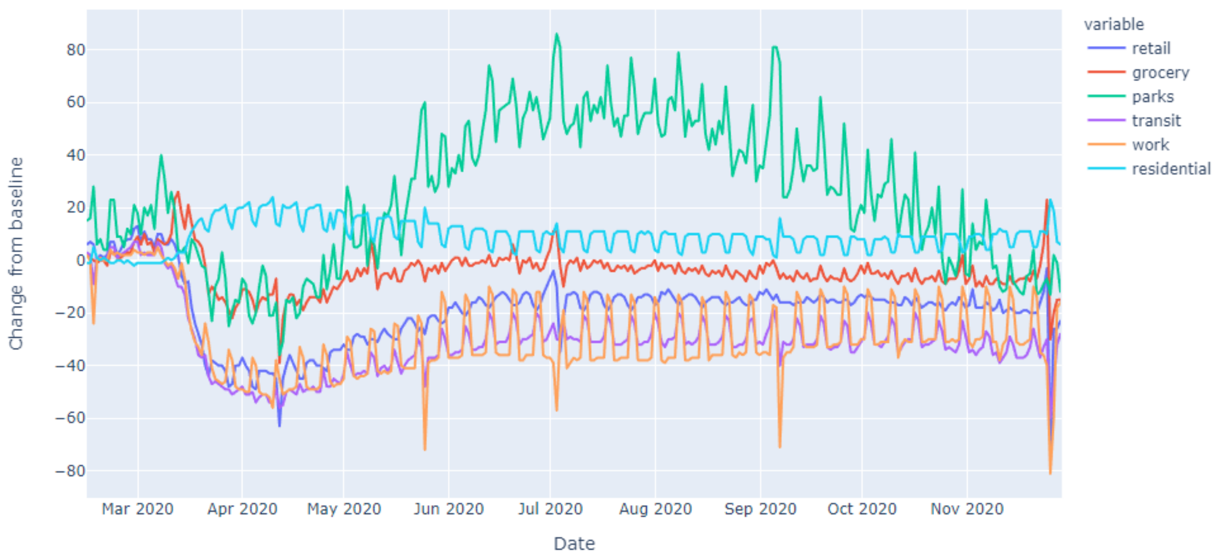

Figure 4. Six mobility variables in the United States from Google's COVID-19 Mobility Report. Dates range from February 15th to November 29th.[15]

The fourth data is from Google and its Mobility Report. As a response to COVID-19 pandemic, Google has released mobility reports of countries across the world in various metrics: grocery, transit, retail, parks, work, and residential. From January to mid-February, Google measured a baseline. Subsequent dates following that baseline period would feature a percentage change from that baseline. Figure 4 shows the mobility trends of the six variables in the United States from mid-February to end of November[15].

Experimental Results

A baseline Prophet model was produced with the default constructor of a Prophet object. It has a linear growth trend and automatic changepoints with a scale of 0.05. It disables yearly and daily seasonality while its weekly seasonality is additive and has a scale of 10. There are no holidays in the model. The model does not appear to fit very well to the new COVID-19 cases. The train and test RMSE will be shown near the end of this section.

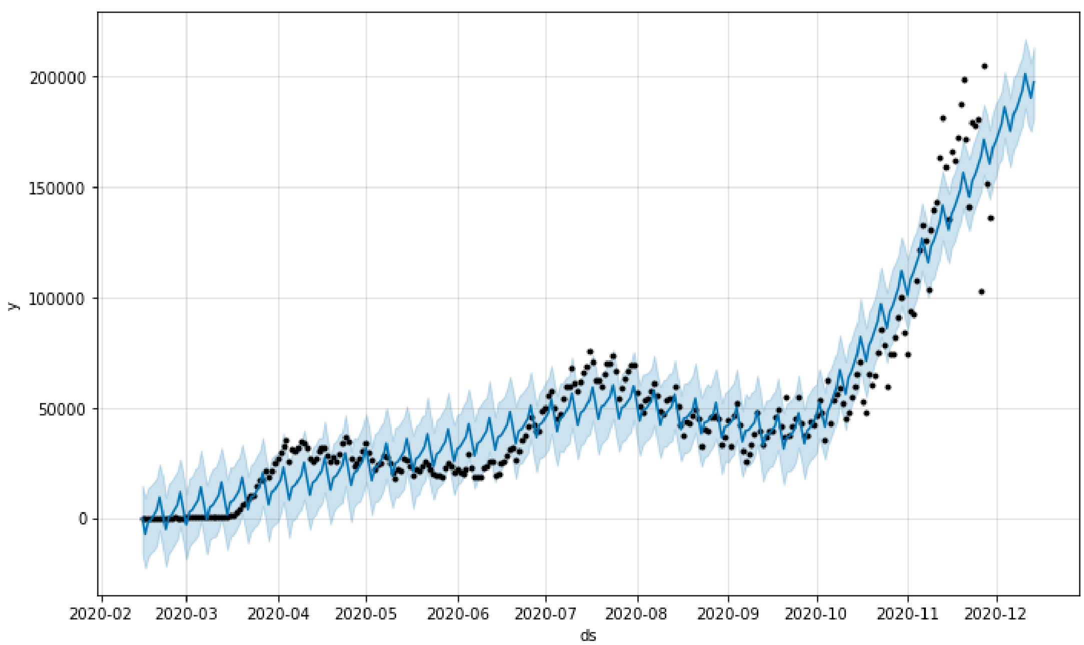

Figure 5. Prediction forecast using the baseline model. RMSE results showed in Table V.

The first part of hyperparameter tuning focuses only on seasonality and trends without any holidays. The hyperparameter combination is shown as a grid in Table I. The trend of the hyperparameter showed that higher Fourier order, changepoint scale, and seasonality scale led to smaller error. Multiplicative seasonality performed better than additive seasonality. Comparing the test RMSE, the best model was chosen as the configuration with smallest RMSE. The best model had multiplicative seasonality, a seasonality scale of 10, a changepoint scale of 0.5, and a Fourier order of 12.

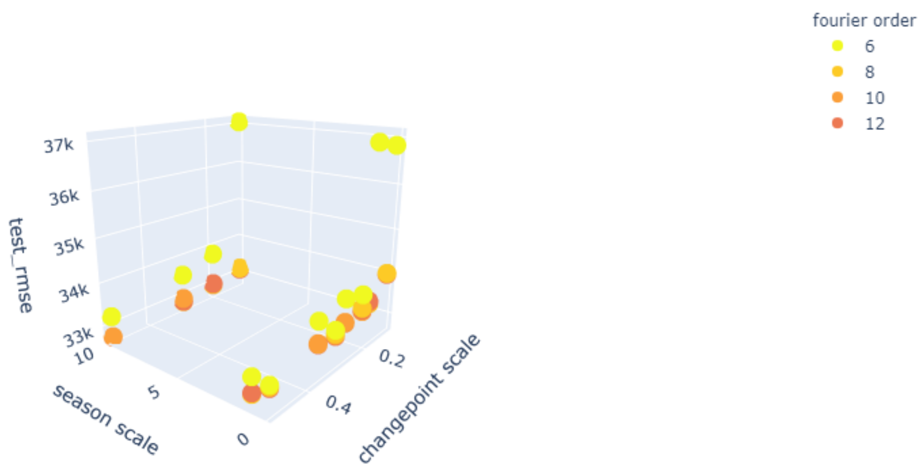

Figure 6. Hyperparameter tuning visualization for part 1 for the multiplicative mode, which performed better than additive seasonality. Performance showed sharp increase as changepoint scale decreased.

The second part of hyperparameter tuning uses the best model from part 1, and begins to tune the holidays. The hyperparameter combination is shown in Table II. There was a definitive relation in which larger lower window (more negative) and smaller upper window led to an improvement in model performance. When the model was not performing well (large upper window and small lower window), smaller holiday scales were better. The opposite is the case for the complement, in which larger holiday scales are better performing. Comparing the test RMSE, the best parameter set was a lower window of -4, upper window of 1, and a holiday scale of 5.

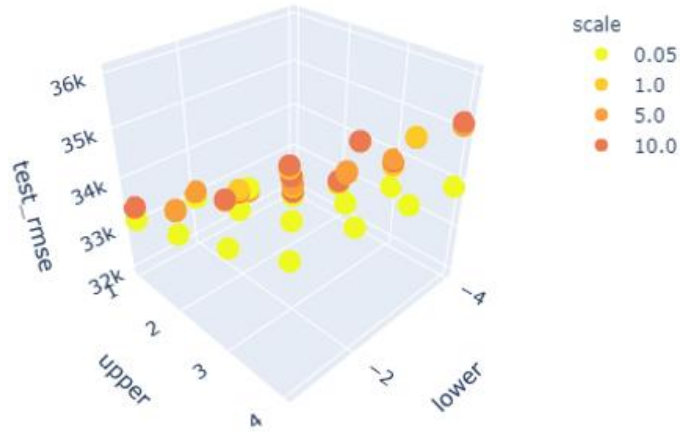

Figure 7. Hyperparameter tuning visualization for part 2 while using the best parameters from part 1 for the other parameters. RMSE was highest when upper was 4 and lower was -1 and lowest when upper was -1 and lower was 4.

The third part of hyperparameter tuning focused on the same set of parameters, but with an expanded pool of holidays that featured mass gatherings and protests. The set is also represented by Table II. Tuning results showed a similar trend as that of part 2, in which larger lower windows and smaller upper windows performed the best. The best model had a lower window of -4, upper window of 1, and a holiday scale of 10. The fourth part of hyperparameter tuning was a manual attempt to tune the expanded holiday list used in part 3. The parameters of each holiday were set individually based on historical and cultural significance.

The final part of hyperparameter tuning used regressors from other time-series data. The hyperparameter grid is in Table III. All of the Google mobility metrics performed better on multiplicative while the stringency index performed better on additive. There seemed to be no pattern regarding the regressor's priority scale. The regressors that performed the best were residential and retail. The best type/scale pair for each regressor were taken, and a manual tuning of the seven regressors found that residential, retail, and work combined made the best model. Unlike parts 1 and 2 with Figures 6 and 7, a visualization is not shown for this part's results. Instead, a data table can be found in the Appendix.

Table IV

Results of Hyperparameter tuning

Tuning Stage	Train RMSE	Test RMSE
Part 1: Without holidays	3171.63	32726.62
Part 2: With holidays	2886.28	32057.55
Part 3: With holidays/events	2786.01	32712.98
Part 4: With holidays/events manual	3024.23	32700.53
Part 5: With regressors	2850.94	30731.03

Table IV shows the results of the hyperparameter tuning. For parts 2, 3, and 4, the best parameter set was with the one without any special events or any manual tuning: part 2. The final model is shown below:

Seasonality and Trends:
- Seasonality: multiplicative
- Seasonality scale: 10
- Changepoint scale: 0.5
- Fourier order: 12

Holidays (no additional events or individual manual tuning):
- Lower window: -4
- Upper window: 1
- Holiday Prior Scale: 5

Regressors:
- Work: multiplicative, scale = 1
- Retail: multiplicative, scale = 1
- Residential: multiplicative, scale = 5

The final model was run on the original train test split that the baseline model was run on, and a prediction forecast was generated. Table V lists the RMSE results and Figure 8 shows the prediction forecast. While the forecast does a good job at fitting to the training data compared to the baseline, it still does a poor job at fitting to the testing data for the November period. It appears to match the trend well (lows to lows and highs to highs), but it seems to fail in matching the magnitude of the trend (off by 30,000 to 40,000 cases). Some health experts attributed to rapid rise of COVID-19 in November to relaxation of restrictions and increased activity in public spaces[20]. The study attempted to account for the loosening of government restrictions, but it appears to not be enough. The Google Mobility Reports also did not seem to be able to account for this increase in activity.

Table V

Comparison between baseline and tuned model

Model	Train RMSE	Test RMSE
Baseline	11186.34	73059.23
Fine-tuned	4137.23	56454.53

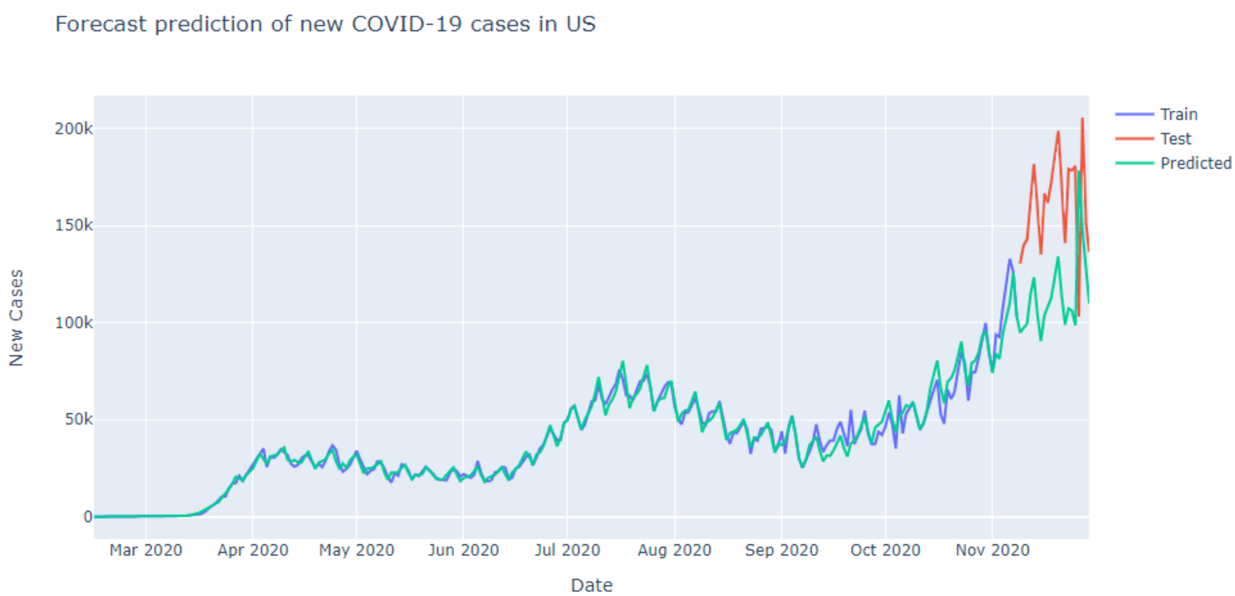

Figure 8. Prediction forecast on train and test data using the fine-tuned model. It has trouble fitting the magnitude of COVID-19 cases, but matches the trends.

Experimental Analysis

Based off of the test RMSE, the part 1 tuning stage performed worse than all of the three stages that introduced holidays. This is indicative of the initial claim that national holidays do have an impact on the rate of COVID-19 transmission in a country. However, the inclusion of certain special events manually did not improve the model in comparison to just the list of national holidays. This suggests that special events may not have the all-encompassing seasonality effect that normal national holidays have. Black Lives Matter and Election Day may be large events with gatherings, but may be regionally concentrated (ex. Black Lives Matter protests being concentrated in urban areas). Another observation is that there is a clear distinction that larger lower window and smaller upper windows improve models with holidays (for both part 2 and part 3). This could mean that more people are testing before a holiday so that they can feel safe when they visit family or attend gatherings. The small upper window suggests that few people are testing after they attended a gathering as they feel assured with testing beforehand. Another observation is that the model improved more with the addition of regressors from part 1 than with just the inclusion of holidays. This suggests that time-series forecasting may benefit from other time-series data as input.

An ablation study was performed to deconstruct the model to the summation shown in Equation 1 in the Background section. There exist three terms: trend, seasonality, and holidays. During the hyperparameter tuning, trend was controlled by the changepoint scale, seasonality was controlled by weekly seasonality, and holidays were controlled by input of national holidays and special events. Regressors influenced both trend and seasonality as the regressor itself was a time-series. Certain components of the best model from experimental results section will be kept constant while removing the others from the model. Table VI shows the results of this process.

Table VI

Comparison between baseline and tuned model

Component	Train RMSE	Test RMSE
Only changepoints	4791.24	35134.07
Only seasonality	12746.42	28125.25
Only holidays	12461.87	26278.22
Only regressors	12226.58	29195.24

The Prophet documentation mentions the changepoint scale as the most important parameter in a Prophet model. It cannot be turned off, so instead it was set to a very small number (0.001). This means the model will produce very little changepoints so the overall curve is very inflexible. Keeping only changepoints fits the training data the best, but is actually worse on the testing data. However, one shortcoming of this project was that the train test split was very low because the total amount of data points was few (less than 300). The other three fitting the test data well may be attributed to small sample size. The test data (month of November) is also far easier to fit as COVID-19 cases at the time primarily went up. The train data is much harder to fit as COVID-19 cases rose and declined from this period. Increasing changepoint scale would greatly improve the model because it will allow for greater flexibility.

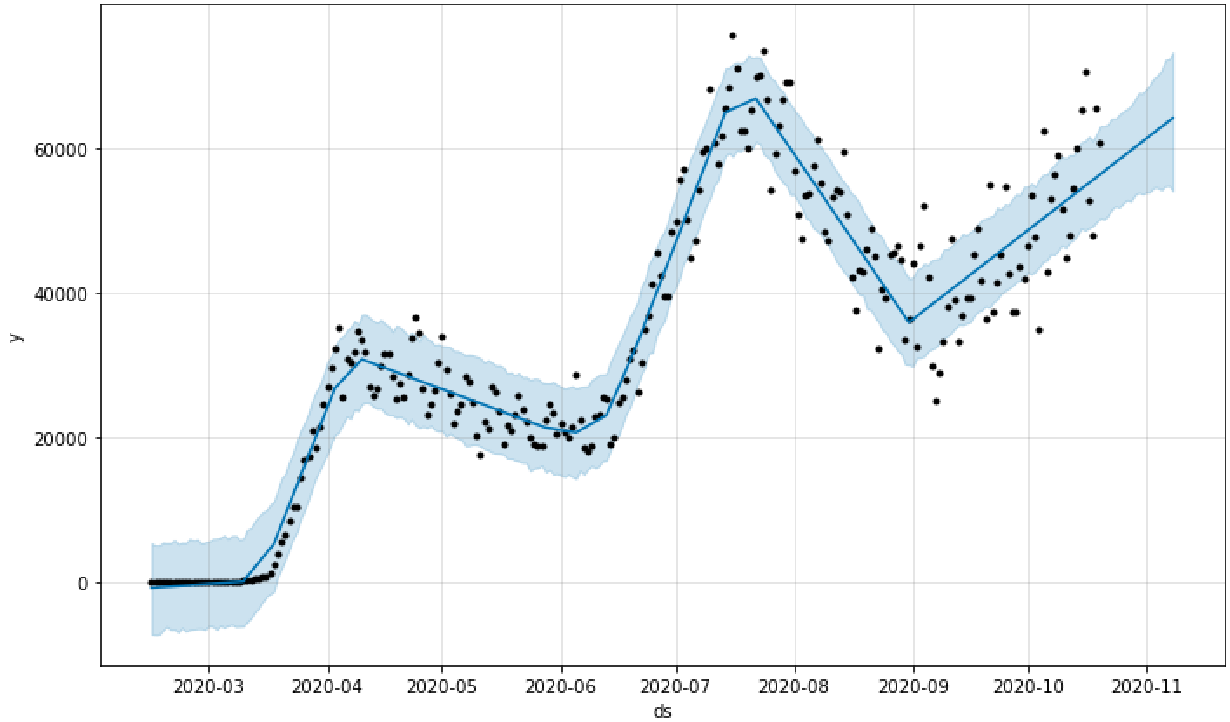

Figure 9. Performance of Prophet model when only the changepoints are left at its optimal value and the rest of the components are minimized or removed. Changepoints appear to be an important foundation to getting a good fit.

Figure 9 captures these conclusions when only the optimal changepoint is kept, as it generally matches well with the different trends from March to November. This was because the changepoint scale provided was the right amount to not underfit the training data. In contrast, Figure 10 shows the forecast when only regressors are kept. Without the ability for a model to add many changepoints, the Prophet model has trouble capturing the trends of a dataset. From this study, it has been seen that the trend and changepoint seems to have the biggest influence on fitting good models by producing an ok fit. The other components exist to fine tune the model.

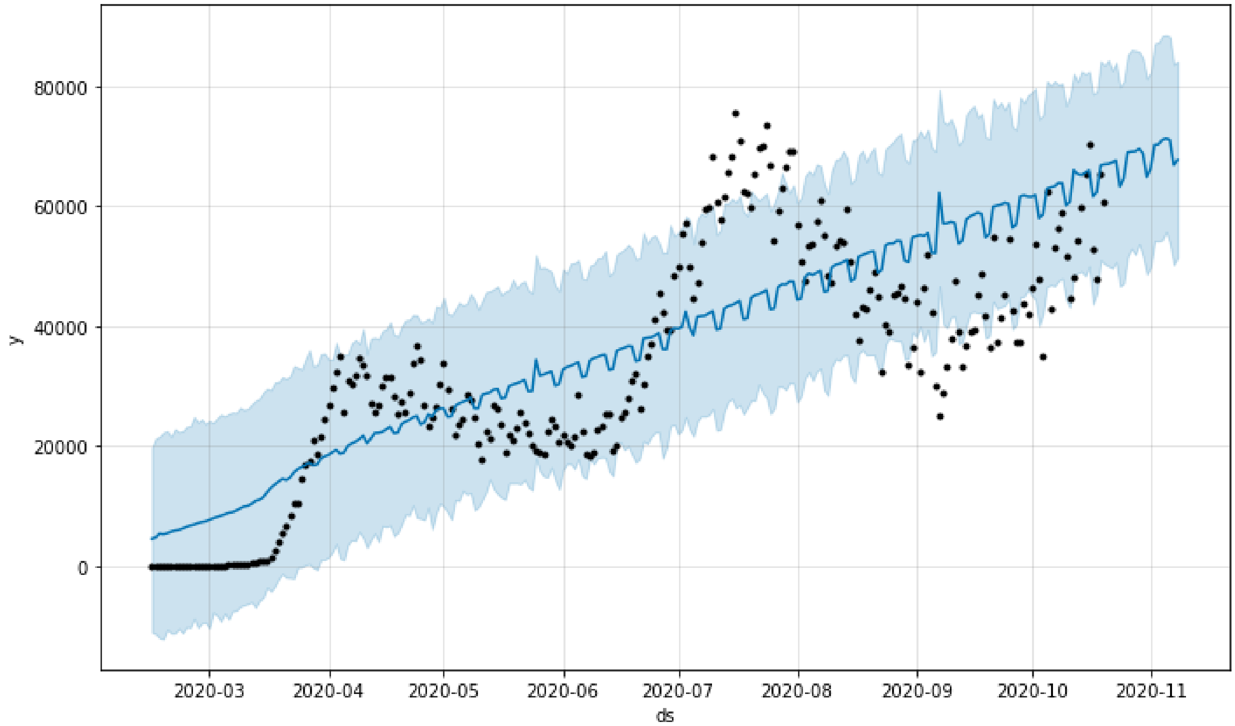

Figure 10. Performance of Prophet model when only the regressors are left at its optimal value and rest of the components are either minimized or removed. Without changepoints, the model cannot appropriately fit to the training data.

Conclusion and Future Work

The goal of the project was to determine the effect of various parameters like pedestrian mobility and national holidays on the spread of COVID-19 in the United States. Facebook's Prophet module was used to develop time-series model that could forecast daily new cases of COVID-19. A model with default parameters was used as a baseline and compared to models that varied in parameters regarding trend, seasonality, and holiday. It was found that models with the holidays performed better than those without, but only when the given holiday list has been carefully selected. Regressors from other time-series data was found to be helpful in improving model performance. From an ablation study, it was determined that the changepoint scale relating to trend was a very important feature to ensuring the Prophet model could fit general time-series data shapes. However, a fine-tuned Prophet model requires optimal changepoints, seasonality, holidays, and regressors to produce a good forecast on both training and testing data.

Since the fine-tuned model captured the trend of the testing data (November) but not its magnitude, extensions could look into more inputs. Since time-series models performed well, more input of time-series data capturing the lessening of restriction could be fed into regressors. Figure 4 showed that the parks metric seemed to be more related with good weather, so temperature could be included into the model. While special events like the BLM protests did not improve the model, future work should narrow it down to individual states. Many COVID-19 regulation decisions are left to state governments, and Oxford's stringency index may not capture those nuances. This could perhaps help the model better fit the magnitude of the COVID-19 cases during November as shown in Figure 8.

Other future work within the context of Prophet module could focus on improvements in project implementation. Mentioned in the implementation section was Prophet's cross validation method. Cross validation is more useful than a one-time split for small data sets like the daily COVID-19 cases because it uses all of the training data. Other metrics besides RMSE should be explored, as metrics like percent error are more intuitive stand-alone quantifications of accuracy. However, metrics like mean absolute percentage error (MAPE) can be skewed when there exist values close to or equal to zero. Extensions can look into the inclusion of modified metrics like weighted absolute percentage error (WAPE) or symmetric MAPE (sMAPE)[22].

A recent paper performed an empirical study between machine learning models and classical models on forecasting, and found that the machine learning models performed worse[21]. Future projects should benchmark and compare the Prophet model to classical statistical models like ARIMA and machine learning models like neural networks.

References

1. https://www.pewresearch.org/fact-tank/2017/12/18/5-facts-about-christmas-in-america/
2. https://web.archive.org/web/20161113140542/http://www.newspolls.org/articles/19586
3. https://nrf.com/insights/holiday-and-seasonal-trends/winter-holidays
4. https://news.gallup.com/poll/113566/us-christmas-not-just-christians.aspx
5. https://www.bts.dot.gov/sites/bts.dot.gov/files/legacy/publications/america_on_the_go/us_holiday_travel/pdf/entire.pdf
6. https://www.tsa.gov/coronavirus/passenger-throughput
7. https://www.bts.gov/covid-19
8. https://www.nytimes.com/2020/10/29/opinion/coronavirus-holiday-travel-thanksgiving.html
9. https://towardsdatascience.com/energy-consumption-time-series-forecasting-with-python-and-lstm-deep-learning-model-7952e2f9a796
10. https://peerj.com/preprints/3190/
11. https://www.covidexitstrategy.org/
12. https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/forecasting.html
13. https://www.forbes.com/sites/grantmartin/2020/02/08/industry-data-shows-steep-drop-in-chinese-air-travel-demand-after-coronavirus-outbreak/?sh=5fbe32313710
14. https://www.kaggle.com/paultimothymooney/nytimes-covid19-data
15. https://www.google.com/covid19/mobility/
16. https://github.com/dr-prodigy/python-holidays
17. https://www.bsg.ox.ac.uk/research/research-projects/coronavirus-government-response-tracker
18. https://facebook.github.io/prophet/docs/quick_start.html
19. https://plotly.com/python/plotly-express/
20. https://www.forbes.com/sites/brucelee/2020/11/23/why-are-covid-19-coronavirus-cases-increasing-here-are-7-reasons/?sh=72ecbcdb4aa8
21. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0194889
22. https://otexts.com/fpp2/accuracy.html

Appendix

	ds	holiday
0	2020-02-17	Washington's Birthday
1	2020-05-25	Memorial Day
2	2020-07-04	Independence Day
3	2020-09-07	Labor Day
4	2020-10-12	Columbus Day
5	2020-11-11	Veterans Day
6	2020-11-26	Thanksgiving
7	2020-12-25	Christmas Day

Figure 11. Holiday list for part 2 hyperparameter tuning, only has national holidays.

	ds	holiday
0	2020-02-17	Washington's Birthday
1	2020-05-25	Memorial Day
2	2020-06-01	BLM Protests
3	2020-06-15	Lessening of Quarantine Measures
4	2020-07-04	Independence Day
5	2020-07-20	Lessening of Quarantine Measures
6	2020-09-07	Labor Day
7	2020-09-12	Lessening of Quarantine Measures
8	2020-10-12	Columbus Day
9	2020-10-26	Lessening of Quarantine Measures
10	2020-10-31	Halloween
11	2020-11-03	Election Day
12	2020-11-11	Veterans Day
13	2020-11-26	Thanksgiving
14	2020-12-25	Christmas Day

Figure 12. Holiday list with special events for part 3 and 4 tuning.

Table VII

Hyperparameter tuning for regressors

mod	reg	reg scale	train_rmse	test_rmse
additive	stringency	1	2878.536555	32142.79238
additive	stringency	5	2871.548877	32088.63397
additive	stringency	10	2868.716731	32060.65448
additive	retail	1	2879.361154	32025.26637
additive	retail	5	2875.756973	32008.09869
additive	retail	10	2877.308986	31930.24329
additive	grocery	1	2891.233534	32029.61362
additive	grocery	5	2894.917712	32053.39764
additive	grocery	10	2890.511685	32038.46493
additive	parks	1	2890.843264	32091.96136
additive	parks	5	2894.532806	32115.00014
additive	parks	10	2892.011341	32089.08964
additive	transit	1	2886.856663	32215.01271
additive	transit	5	2889.742732	32207.64628
additive	transit	10	2888.622361	32208.82395
additive	work	1	2882.384825	32173.80759
additive	work	5	2880.919443	32194.85944
additive	work	10	2882.155998	32182.62581
additive	residential	1	2884.071491	32149.96845
additive	residential	5	2889.775688	32227.70254
additive	residential	10	2889.616871	32185.23435
multiplicative	stringency	1	2738.391381	33193.4313
multiplicative	stringency	5	2734.128444	33329.87357
multiplicative	stringency	10	2733.469053	33380.92882
multiplicative	retail	1	2864.827428	31687.39924
multiplicative	retail	5	2869.382488	31765.29539
multiplicative	retail	10	2864.394654	31704.17238
multiplicative	grocery	1	2881.53267	31865.71195
multiplicative	grocery	5	2889.183841	31892.22957
multiplicative	grocery	10	2890.267332	31880.50468
multiplicative	parks	1	2895.670182	31871.69023
multiplicative	parks	5	2892.074894	31896.86679
multiplicative	parks	10	2892.387417	31830.31385
multiplicative	transit	1	2891.542925	31814.811
multiplicative	transit	5	2896.568171	31907.00801
multiplicative	transit	10	2887.990438	31796.09231
multiplicative	work	1	2896.565343	32053.70813

multiplicative	work	5	2893.786063	32107.72256
multiplicative	work	10	2895.735517	32080.20506
multiplicative	residential	1	2883.03317	31292.07376
multiplicative	residential	5	2881.727647	31185.8623
multiplicative	residential	10	2886.066135	31335.23204