PREDICTING YEAR-OVER-YEAR CHANGE IN FOOT TRAFFIC DUE TO COVID-19

Tamar Novetsky (trn219), Alene Rhea (akr435), and Michael Stanley (mhs592)

NYU Center for Data Science, Machine Learning, Spring 2020

1. INTRODUCTION

The COVID-19 pandemic has transformed gathering sites into transmission sites. There is a new need to predict where people are gathering, in order to enforce social distancing, supply PPE, and establish guidelines and recommendations. The goal of this project is to predict changes in foot traffic to points of interest within key centers of the US pandemic.

Specifically, we aim to predict the year-over-year change in weekly foot traffic for thousands of specific locations (e.g., individual stores, airports, etc.) based on the type of business, the timing of COVID-19 regulations in the area, the demographics and economy of the surrounding area, and the weather. We have acquired 15 months of weekly foot traffic data for millions of locations in the US from SafeGraph. We have incorporated a number of additional datasets to generate the feature set described above.

Predicting change in foot traffic is a regression problem. The primary focus is on prediction rather than interpretability, so we evaluate regression models over a range of complexity. The data has a temporal aspect, as each week is considered a separate sample, and this temporality is considered when defining new features, as well as training, validation, and test sets.

We found xgboost to be the best performing model and that it generalizes well to the test set. The xgboost model clearly outperformed the linear baseline, with 80% lower MAE on the validation set. A custom evaluation metric allows the model to be fairly robust to outliers. Feature importance analysis indicates that features from many different sources improved model performance.

2. APPROACH

The cornerstone of our approach is the integration of features from many different datasets in order to construct a holistic model of changes occurring during the COVID-19 pandemic. Our analysis is focused on five key urban centers of the US pandemic: New York City, Seattle, New Orleans, Detroit, and Atlanta. The impact of COVID-19 is likely very different between urban and rural areas. We chose to focus on urban centers so that urban-rural differences did not outweigh other factors. We chose cities in different states to emphasize the impact of government regulation on foot traffic. Each row in our dataset represents a single location-week (e.g., "Starbucks on Water St., March 8-14, 2020"). Modeling weekly traffic instead of daily traffic removes the impact of weekends, and provides a smoothing effect.

Our target variable is the year-over-year change in weekly foot traffic for the next week. No data from the next week is used in prediction. For the Starbucks on Water Street example, our target variable would be:

$$\frac{\text{(\# Visits: 3/15-3/21, 2020)} - \text{(\# Visits, 3/17-3/23, 2019)}}{\text{1+ (\# Visits: 3/17-3/23, 2019)}} \quad (1)$$

The distribution of our target variable is heavily right-skewed. The target variable is a percentage change, so has a skewed range of $[-1, +\infty]$. In order to prevent the undue influence of extreme, positive outliers, we experimented with capping the target variable at different values. (See Table 1.) Baseline tests were run at several cap levels, to track the impact of changes on different target distributions. Capping the target variable of the validation or test set is equivalent to replacing y labels in the evaluation metric formula with $\min(y, c)$, where c is the cap level. Thus, evaluation metrics cannot be compared across cap levels. We ultimately chose to cap the target variable at 5, corresponding to a 500% increase in foot traffic from 2019 to 2020.

Cap	mean	std	min	median	max
None	-0.3934	3.0143	-0.9989	-0.6932	584.0
100	-0.4053	2.0663	-0.9989	-0.6932	100.0
10	-0.4537	0.9886	-0.9989	-0.6932	10.0
5	-0.4787	0.7718	-0.9989	-0.6932	5.0
2	-0.5152	0.5667	-0.9989	-0.6932	2.0
1	-0.5419	0.4637	-0.9989	-0.6932	1.0

Table 1. Distribution of the target variable by cap on outliers.

We used a non-random, time-based split to divide our data into training, validation, and test sets. We chose this method because we were predicting behavior in the same places over time, and wanted to avoid data leakage from future weeks. The training set spans March 8 - April 4, 2020, the validation set covers April 5-11, and the test set covers April 12-18.

Our baseline model is a simple, un-regularized linear regression model using only minimal SafeGraph data. Our primary experiments were the addition of new features, from external datasets and engineered from the SafeGraph data. We also experimented with l1 and l2 regularization, Ada-Boost, random forests, gradient-boosting, xgboost, and multi-layer perceptrons. Features were standardized before modeling so that regularization worked consistently across the feature space.

We used 5-fold gridsearch on the training data to evaluate a host of hyperparameter combinations for each algorithm. For each model class, the optimal hyperparameters were re-trained on the full training set and applied to the validation set. A final model was selected based on mean absolute error on the validation set, re-trained on the full training and validation set, and evaluated on the test set. MAE, mean squared error (MSE) and R-squared (r^2) metrics were calculated for training and validation, but MAE was used for model selection to reduce the impact of outliers.

Because the data comprise a time series, we experimented with including features representing several past weeks worth of data, including previous weeks' targets. We also experimented with one-hot encoding of NAICS codes. While NAICS codes are somewhat directional (111111 is closer to 111112 than it is to 555555), they are also categorical. After evaluating models with and without one-hot encoding, we found that one-hot

encoding did not improve model performance, but vastly increased the size of the feature space, and so we removed the one-hot features.

2.1 Datasets

Below is a list of data sources, a summary of the data they provide, and a brief description of the data processing performed:

- SafeGraph: Aggregated anonymous location data from about 35 million mobile devices, detailing visits to about 4 million places of interest in the US.
- Keystone Strategy: non-pharmaceutical (government) interventions
- U.S. Department of Housing and Urban Development: USPS ZIP-FIPS crosswalk
- New York Times: Daily COVID-19 case and death counts at the county level.
- United States Census Bureau: Employment and population demographics by zip code, number of establishments by NAICS code and zip code.
- National Oceanic and Atmospheric Administration: Historical daily temperature and weather patterns by weather station.
- CivicSpace Labs: Latitude and longitude of US zip codes, used to identify nearest weather station.

The features we derived from these datasets are described in detail in Appendix A.

SafeGraph. Our target variable was engineered using SafeGraph Weekly Patterns and Historic Monthly Patterns data. In order to compare weekly data from 2020 with monthly data from 2019, we de-aggregated the "visits_by_day" vector to form one row per day and calculated the daily visitors and other desired features (e.g. number of brands in the 'same_day_related_brand frequent itemset' dictionary). Daily records were then aggregated weekly, replicating the release schedule of the Weekly Patterns data to ensure that that no data leakage would occur when lagged features were introduced. During aggregation, we took the first value of location-based features which do not change over time (e.g., zip code), summed the number of daily visits to the place of interest (POI), and took the median of all visit-based features which are constant per week in 2020 (e.g., maximum number of visits to the POI in any single hour of the week) or per month in 2019 (e.g. median distance from home travelled by visitors to POI, taken over all trips in a month). The 2019 and 2020 datasets were then joined, and year over year change in visits for each location-week was calculated: (2020 visits - 2019 visits) / (2019 visits + 1). Note that when calculating the year-over-year change, the visit numbers were all padded by 1, to prevent division by zero.

The final step in preparing the SafeGraph data was to add lagged features to improve time-series analysis. First, to ensure that we could add any arbitrary look-back window, we filtered the dataset to include only those locations which appeared in every week of the data. Then, we appended to each week the next week's "change_in_visits" feature; this would serve as our target variable. We also appended all visit-based features from the next week in 2019. Finally, we added all visit-based features from the previous week of both 2019 and 2020.

The addition of lagged features reduced the size of our dataset, as we had to exclude the first and last weeks due to missing data. Using next-week's "change_in_visits" feature was crucial for our approach; using visit-based SafeGraph features to predict the same week's change in visits would constitute flagrant data leakage. The decision to additionally use backward-looking lagged features was based on the notion that giving the model access to each previous week would be more valuable than giving it an additional week of training data from the very beginning of the pandemic. This trade-off was tested in our baseline models.

NAICS codes. The North American Industry Classification System (NAICS) is a coding system used to classify business establishments by type of economic activity. Since NAICS codes indicate different levels of detail at different numbers of digits (the first 2 digits of a NAICS code describe a sector, the first 3 digits describe a subsector, etc.), we produced 2, 3, 4, and 5-digit slices of the 6-digit "naics_code" feature and used each slice as a new feature. Other NAICS code-specific features were added as well (e.g., business count by NAICS code for each zip code).

Government Interventions. We created features for each week and ZIP code indicating what percentage of the week each intervention was in effect locally. The government interventions we used as features were social distancing mandates, shelter-in-place orders, closing of public venues, school closures, and non-essential services closures. These interventions were selected due to availability of effective start date information for each of the counties in our dataset.

Weather. Weather data is collected and maintained for individual weather stations. In order to generate historical weather features, we manually determined the closest weather station to the geographical center of each zip code. Many weather stations do not store daily weather records, so we filled gaps with the next nearest weather station where necessary.

US Census Bureau. The Census Bureau reports the number of establishments in each NAICS code at the zip code level, but the data is incomplete. Where the NAICS code breakdown was not available, we filled missing fields by multiplying the overall number of establishments in the zip code by the average NAICS code breakdown in the surrounding county. The Census Bureau also reports number of employed people per zip code, but the figure is a range (e.g., 500-999). We used the midpoint of each range as the estimated number of employed people per zip code.

Data Merging. Governmental interventions, COVID-19 case and death counts, and population demographics were all initially tabulated by county, not zip code, so we mapped from county to zip code using the County-zip Crosswalk database provided by the U.S. Department of Housing and Urban Development. Once we had everything tabulated by zip, we joined these features with the SafeGraph data using the combination of zip code and week as the join key for a left join.

In total, we generated 63 features for 222,263 location-week samples.

3. EXPERIMENTS

3.1 Baseline Experiments: Feature Selection and Outlier Capping

The most basic model we produced was an out-of-the-box, unregularized linear regression using only the previous target variable of each week (i.e., 'change_in_visits') as its singular feature. We gradually added features to that simple model: first

the SafeGraph features without any lagged variables, then the lagged SafeGraph variables, and finally all of the external data. For each of these datasets, we tested the training and validation set performance with target outliers uncapped, and capped at 1, 2, 5, 10, and 100 (See Tables 4 and 5 in Appendix B.) Findings are summarized in Table 2.

	Best Val R2	Best Val MSE	Best Val MAE
No cap	SG with lag	Full dataset	Previous target
Cap=100	SG with lag	Full dataset	SG with lag
Cap=10	SG with lag	SG with lag	SG with lag
Cap=5	SG with lag	SG with lag	SG with lag
Cap=2	SG with lag	SG with lag	SG with lag
Cap=1	SG with lag	Full dataset	SG with lag

Table 2. Best model per cap level (SG=SafeGraph)

It's clear that the baseline model struggles to model outliers. With no cap on outliers, MSE explodes, as expected. Less expected is that with no cap, each additional set of variables makes the validation MAE worse. The parity we see between the previous target and lag features makes sense, because the previous target is in fact a lag feature. It seems that these features, most of which are inherently similar to the actual target variable, are the only ones which are able to provide the specific variance necessary to model outliers. At all cap levels besides None, we find that the addition of lag variables improves validation performance by all metrics. This finding is significant, because adding lag variables also corresponds to reducing the size of the training set by 20%. As discussed above in the "Approach" section, it is not appropriate to compare evaluation metrics from different outlier caps, as the caps effectively change the definitions of the metrics. To choose a cap level, we can look for a natural "elbow" in the data; we find one at a cap of 5 (Figure 1).

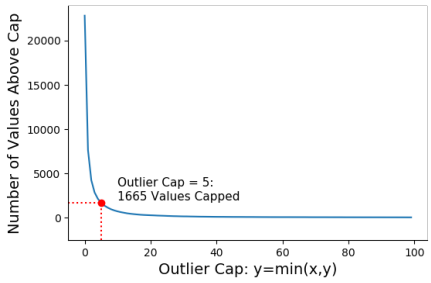

Figure 1. Outlier count for different caps.

At cap=5, MAE and MSE both improve on the training set with each additional dataset, but they both peak on the validation set with lagged SafeGraph features, getting worse when external features are added. This indicates that the regressor is overfitting on the external features.

Moving forward into our experiments with more sophisticated models, we will use cap=5 and the full dataset, including external features. We suspect that the observed overfit on the external features is due to approximation error. We hypothesize that non-linear models will be better able to capture these complex relationships, and will indeed benefit from the addition of external data.

Because the training, validation, and test sets were split temporally and not randomly, the variance of the target variable differs substantially across them, as shown in Table 3. Models are only ever compared over the same dataset, so this should not impact model selection, but it is relevant when discussing overfitting and generalization to the validation and test sets. In short, the variance of the validation set is lower than training or test, which makes models appear to generalize to the validation set better and to the test set worse than they actually do. The R-squared metric somewhat accounts for this impact.

Dataset	Variance
Training (March 8-April 4)	0.6360
Validation (April 5-11)	0.4683
Test (April 12-18)	0.5488

Table 3. Target variable variance for each dataset.

3.2 Results of Hyperparameter Experiments

	MAE	MSE	R^2
Lasso	0.4861	0.3778	0.2357
Ridge	0.4836	0.3764	0.2396
XGBoostRegressor	0.0469	0.1333	0.9262
Random Forest	**0.0165**	**0.0668**	**0.9740**
Gradient Boosting	0.0868	0.1636	0.8636
Adaboost	0.2261	0.2981	0.6445
MLP	0.1480	0.1915	0.7672

Table 4. Training Metrics for each model with optimal hyperparameters. Bolding indicates best model of each metric.

Tables 4 and 5 report the evaluation metrics on the training and validation sets, respectively, for each model using the optimal hyperparameters determined by gridsearch. We see that non-linear models significantly outperform Lasso and Ridge regression, and that tree-based ensemble models perform best on the validation set. Comparing training and validation error, we see that Random Forest and MLP overfit the training set the most, as the error of these models increases most during validation. Interestingly, a few models have lower MAE and MSE on validation than on training, likely due to the lower overall variance in the validation set (0.47 for validation vs. 0.64 for training).

	MAE	MSE	R^2
Lasso	0.3920	0.3766	0.1628
Ridge	0.3912	0.3755	0.1645
XGBoostRegressor	**0.0745**	**0.1326**	**0.8409**
Random Forest	0.0779	0.1338	0.8337
Gradient Boosting	0.0818	0.1627	0.8254
Adaboost	0.1663	0.2512	0.6449
MLP	0.1739	0.2446	0.6286

Table 5. Validation Metrics for each model with optimal hyperparameters. Bolding indicates best model of each metric.

Notably, the xgboost model has 80% lower MAE than the baseline on the validation set (0.0745 compared to 0.3742).

xgboost had the lowest validation MAE and was chosen as our final model. Random Forest and Gradient Boosting had only marginally worse results on the validation set. Table 6 reports the xgboost evaluation metrics on the test set, after re-training on the full training and validation sets. The results are encouraging: test MAE for xgboost is 36% higher than validation MAE, but this is partially due to variance in the test set being 17% higher than in the validation set. This is apparent in the 3% decrease in R-squared value, which accounts for total variance. The test error indicates that the model generalizes quite well to the test data and continues to dramatically outperform the benchmarks.

	MAE	MSE	R^2
Train	0.0469	0.1333	0.9262
Validation	0.0745	0.1326	0.8409
Test	0.1571	0.1020	0.8142

Table 6. Evaluation metrics for xgboost.

3.3 Error Analysis

As discussed above, when we re-trained xgboost on the full dataset (training and validation), we obtained very good results that were similar to, although slightly worse than, the errors on the training and validation sets. The results are tabulated in Table 6.

We also examined the residuals on the test data as a function of target variable, shown in Figure 2. The residuals are smallest on average close to the average target variable, -0.5, and get larger in spread and maximum value at higher values of the target, corresponding to outliers in the data. This is encouraging - our model focuses on the range [-1, 1], and is not skewed too heavily by outliers. The MAE of test data with target variable less than 1 is 0.1256, 20% lower than the overall MAE of 0.1571.

We also find that the more extreme errors in the test set differ by target variable. Instances with target variable closer to 0, if the prediction is far from the target, are more likely to have been overestimated, and therefore the residual is strongly negative. Conversely, outlier data points with high target variable values are more likely to be underestimated by our model, corresponding to strongly positive residuals.

Figure 2. Residuals on test set as a function of target variable.

4. DISCUSSION

4.1 Evaluation of Findings

The results above show that the xgboost model trained on data from a wide range of sources clearly outperforms the linear baseline and has notable predictive power for foot traffic. Other ensemble models performed nearly as well, which indicates that the approach taken here is repeatable.

4.2 Feature Importance

Like most tree-based models, xgboost provides an indication of feature importance, defined as the average reduction in standard deviation for splits using that feature. Not surprisingly, the most important feature is the previous week's target variable, "change_in_visits." If visits are down 10% this week, it is reasonable to expect visits next week to be down a similar amount. The next feature, 'visits_2019_nextweek' actually contributes directly to the target variable. Historic visit counts account for several other top features, which indicates that businesses at different nominal traffic levels may be impacted differently. Other top features are government regulations ("school_closure_pct" and "social_distancing_pct"), surrounding economics ("est" and "emp"), and visitor demographics ("num_visitor_home_cbgs", "num_visitor_country_of_origin_lastweek"). The diversity of top features reinforces the value of aggregating a wide variety of datasets.

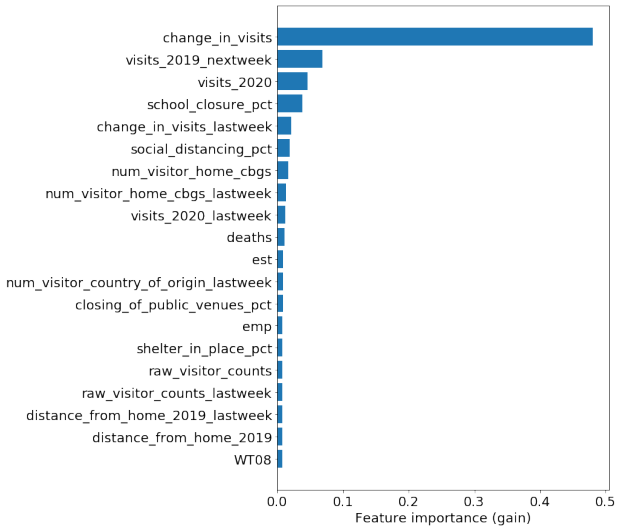

Figure 3. Top 20 features by XGBRegressor importance.

4.3 Future Work

The feature importance of our final xgboost model demonstrates the value of bringing together a diverse set of features when predicting the impact of COVID-19. Our strategy can be expanded to incorporate relevant data from other domains, such as geotagged Twitter sentiment analysis, election history and polling data, and additional healthcare systems data. SafeGraph's own upcoming data release will provide a rich source of new features, as well, including social distancing metrics and the square footage and census block group of each location. This data release will include Weekly Patterns going back to January 2019, which would allow us to expand our training set back through January and February.

Additional training data will make larger lag windows viable. Experiments could be run to find the optimal look-back window (i.e., the number of past weeks of 2020 data and past/future weeks of 2019 data to append to every row). The inherent trade-off between lag window and dataset size could be analyzed with careful, lag-aware bootstrapping analysis. Lagged features could also be added for the external datasets. Our multi-layer perceptron did not perform as well as our tree-based ensemble methods, but that might well change with further hyperparameter tuning. Different activation functions could be tested, along with a broad range of network parameter combinations. We could also try to implement a non-strongly connected neural net. A literature review would be helpful to identify potential parameter configurations, as neural nets are notoriously difficult to tune from scratch.

These proposed approaches could be iteratively explored using each new Weekly Patterns data release as a test set. Because SafeGraph's data is aggregated and released weekly, there is no need to adapt our method to online learning. Rather, the data release schedule dovetails with our methodology to form a convenient experimental protocol.

SOFTWARE

This project was conducted in Python and utilized the following modules: numpy, pandas, matplotlib, scikit-learn, xgboost. Source code is available at github.com/akrhea/covid-foot-traffic.

REFERENCES

1. Keystone Strategy. "Coronavirus City And County Non-pharmaceutical Intervention Rollout Date Dataset." Last Revised: April 22, 2020. Available at URL: https://github.com/Keystone-Strategy/covid19-intervention-data

2. New York Times, 2020, "Coronavirus (Covid-19) Data in the United States". Available at URL: https://github.com/nytimes/covid-19-data

3. United States Census Bureau. "County Business Patterns: 2017: Complete ZIP Code Industry Detail File." Last Revised: December 12, 2019. Available at URL: https://www.census.gov/data/datasets/2017/econ/cbp/2017-cbp.html

4. United States Census Bureau. "County Business Patterns: 2017: Complete ZIP Code Totals File." Last Revised: December 12, 2019. Available at URL: https://www.census.gov/data/datasets/2017/econ/cbp/2017-cbp.html

5. United States Census Bureau. "County Population Totals: 2010-2019: All Data 2019." Available at URL: https://www.census.gov/data/tables/time-series/demo/popest/2010s-counties-total.html

6. United States Department of Housing and Urban Development. "HUD USPS ZIP Code Crosswalk Files." Available at URL: https://www.huduser.gov/portal/datasets/usps_crosswalk.html

7. National Oceanic and Atmospheric Administration. "Climate Data Online: Daily Summaries." 2020. Available at URL: https://www.ncdc.noaa.gov/cdo-web/

8. CivicSpace Labs, "US Zip Code Latitude and Longitude". Last updated February 9, 2018. Available at URL: https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/table/

APPENDIX A: FEATURE DESCRIPTIONS

week: Week number of the year (Sunday as the first day of the week), as int. (All days in a new year preceding the first Sunday are considered to be in week 0.) Ranges from 11-16. For merging only; dropped before modeling.

postal_code: 5-digit zip-code. Int. For merging only; dropped before modeling.

naics_2: 2-digit NAICS code. String. '0' indicates missing; naics_2_0 dropped after one-hot encoding. Engineered from SafeGraph data.

naics_3: 3-digit NAICS code. String. '0' indicates missing; naics_3_0 dropped after one-hot encoding. Engineered from SafeGraph data.

naics_4: 4-digit NAICS code. String. '0' indicates missing; naics_4_0 dropped after one-hot encoding. Engineered from SafeGraph data.

naics_5: 5-digit NAICS code. String. '0' indicates missing; naics_5_0 dropped after one-hot encoding. Engineered from SafeGraph data.

naics_code: Full 6-digit NAICS code. String. '0' indicates missing; naics_code_0 dropped after one-hot encoding. From SafeGraph data.

naics_2_num_biz: Number of established businesses matching 2-digit NAICS code in zip code, 2017. From United States Census Bureau data.

naics_3_num_biz: Number of established businesses matching 3-digit NAICS code in zip code, 2017. From United States Census Bureau data.

naics_4_num_biz: Number of established businesses matching 4-digit NAICS code in zip code, 2017. From United States Census Bureau data.

naics_5_num_biz: Number of established businesses match 5-digit NAICS code in zip code, 2017. From United States Census Bureau data.

naics_6_num_biz: Number of established businesses matching full 6-digit NAICS code in zip code, 2017. From United States Census Bureau data.

visits_2019: Total number of visits to the POI over this week in 2019. Engineered from SafeGraph data.

visits_2019_lastweek: Total number of visits to the POI over last week in 2019. Engineered from SafeGraph data.

visits_2019_nextweek: Total number of visits to the POI over next week in 2019. Engineered from SafeGraph data.

visits_2020: Total number of visits to the POI over this week in 2020. Engineered from SafeGraph data.

visits_2020_lastweek: Total number of visits to the POI over last week in 2020. Engineered from SafeGraph data.

change_in_visits: (visits_2020 - visits_2019) / (visits_2019+1). Engineered from SafeGraph data.

change_in_visits_lastweek: (visits_2020_lastweek - visits_2019_lastweek) / (visits_2019_lastweek+1). Engineered from SafeGraph data.

target: The dependent variable. Equivalent to change_in_visits_nextweek: (visits_2020_nextweek - visits_2019_nextweek) / (visits_2019_nextweek+1). Engineered from SafeGraph data.

distance_from_home_2019: Median distance from home travelled by visitors to POI (of visitors whose home Safegraph has identified) in meters. Original median taken over the month; median taken again over this week in 2019. Engineered from SafeGraph data.

distance_from_home_2019_lastweek: Median distance from home travelled by visitors to POI (of visitors whose home Safegraph has identified) in meters. Original median taken over the month; median taken again over last week in 2019. Engineered from SafeGraph data.

distance_from_home_2019_nextweek: Median distance from home travelled by visitors to POI (of visitors whose home Safegraph has identified) in meters. Original median taken over the month; median taken again over next week in 2019. Engineered from SafeGraph data.

distance_from_home_2019_missing: Binary flag indicating distance_from_home_2019 was missing and has been replaced by training median (weeks 11-15). Engineered from SafeGraph data.

distance_from_home_2019_missing_lastweek: Binary flag indicating distance_from_home_2019_lastweek was missing and has been replaced by training median (weeks 11-15). Engineered from SafeGraph data.

distance_from_home_2019_missing_nextweek: Binary flag indicating distance_from_home_2019_nextweek was missing and has been replaced by training median (weeks 11-15). Engineered from SafeGraph data.

max_hourly_visits: The maximum number of visits to the POI in a single hour of this week in 2020. Engineered from SafeGraph data.

max_hourly_visits_lastweek: The maximum number of visits to the POI in a single hour of last week in 2020. Engineered from SafeGraph data.

median_dwell_2019: Median minimum dwell time in minutes. Original median taken over the month; median taken again over this week in 2019. Engineered from SafeGraph data.

median_dwell_2019_lastweek: Median minimum dwell time in minutes. Original median taken over the month; median taken again over last week in 2019. Engineered from SafeGraph data.

median_dwell_2019_nextweek: Median minimum dwell time in minutes. Original median taken over the month; median taken again over next week in 2019. Engineered from SafeGraph data.

median_dwell_2020: Median minimum dwell time in minutes, over this week in 2020.

median_dwell_2020_lastweek: Median minimum dwell time in minutes, over last week in 2020. Engineered from SafeGraph data.

num_related_same_day_brand_2019: Number of other brands that the visitors to this POI visited on the same day as the visit to this POI where customer overlap differs by at least 5% from the SafeGraph national average, median for this week in 2019. Engineered from SafeGraph data.

num_related_same_day_brand_2019_lastweek: Number of other brands that the visitors to this POI visited on the same day as the visit to this POI where customer overlap differs by at least 5% from the SafeGraph national average, median for last week in 2019. Engineered from SafeGraph data.

num_related_same_day_brand_2019_nextweek: Number of other brands that the visitors to this POI visited on the same day as the visit to this POI where customer overlap differs by at least 5% from the SafeGraph national average, median for next week in 2019. Engineered from SafeGraph data.

num_related_same_day_brand_2020: Number of other brands that the visitors to this POI visited on the same day as the visit to this POI where customer overlap differs by at least 5% from the SafeGraph national average, median for this week in 2020. Engineered from SafeGraph data.

num_related_same_day_brand_2020_lastweek: Number of other brands that the visitors to this POI visited on the same day as the visit to this POI where customer overlap differs by at least 5% from the SafeGraph national average, median for last week in 2020. Engineered from SafeGraph data.

num_visitor_country_of_origin: Number of home countries with 5 or more visitors to the POI this week in 2020. Engineered from SafeGraph data.

num_visitor_country_of_origin_lastweek: Number of home countries with 5 or more visitors to the POI last week in 2020. Engineered from SafeGraph data.

num_visitor_home_cbgs: Number of home census block groups with 5 or more visitors to the POI this week in 2020. Engineered from SafeGraph data.

num_visitor_home_cbgs_lastweek: Number of home census block groups with 5 or more visitors to the POI last week in 2020. Engineered from SafeGraph data.

raw_visitor_counts: Number of unique visitors from SafeGraph's panel to this POI during this week in 2020. Engineered from SafeGraph data.

raw_visitor_counts_lastweek: Number of unique visitors from SafeGraph's panel to this POI during last week in 2020. Engineered from SafeGraph data.

TMIN: Min temperature (F) of the week. Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

TMAX: Max temperature (F) of the week. Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

PRCP: Inches of precipitation for the week. Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

SNOW: Inches of snowfall for the week. Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

WT01: Number of days with fog, ice fog, or freezing fog (may include heavy fog). Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

WT02: Number of days with heavy fog or heaving freezing fog (not always distinguished from fog). Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

WT03: Number of days with thunder. Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

WT04: Number of days with ice pellets, sleet, snow pellets, or small hail. Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

WT06: Number of days with glaze or rime. Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

WT08: Number of days with smoke or haze. Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

WT11: Number of days with high or damaging winds. Engineered from National Oceanic and Atmospheric Administration and CivicSpace Labs data.

cases: Number of COVID cases in the FIPS county, running total up to that week. From New York Times data.

deaths: Number of COVID deaths in the FIPS county, running total up to that week. From New York Times data.

POPESTIMATE2019: Estimated population in FIPS county in 2019. From United States Census Bureau data.

Pop_pct_chg_2019: Estimated year-over-year population change for FIPS county, 2018 to 2019. From United States Census Bureau data.

emp: Number of employed persons in zip code, 2017. From United States Census Bureau data.

est: Number of established businesses in zip code, 2017. From United States Census Bureau data.

closing_of_public_venues_pct: Portion of week in which closure of public venues was in effect for this zip code. Closure of public venues is defined as a government order closing gathering venues for in-person service, such as restaurants, bars, and theaters. Engineered from Keystone Strategy data.

non-essential_services_closure_pct: Portion of week in which non-essential services closure was in effect for this zip code. Non-essential services closure is defined as a government order closing non-essential services and shops. Engineered from Keystone Strategy data.

school_closure_pct: Portion of week in which closure of schools and universities was in effect for this zip code. Engineered from Keystone Strategy data.

shelter_in_place_pct: Portion of week in which a shelter in place order was in effect for this zip code. Shelter in place is defined as an order indicating that people should shelter in their homes except for essential reasons. Engineered from Keystone Strategy data.

social_distancing_pct: Portion of week in which social distancing was in effect for this zip code. Social distancing is defined as a mandate of at least 6 feet between people. Engineered from Keystone Strategy data.

APPENDIX B: OUTLIER CAP COMPARISONS

Outlier Cap	Dataset	Train R2	Train MSE	Train MAE
No cap	Previous target only	0.640898	7.724115	0.450076
	SG without lag	0.644967	7.636591	0.455493
	SG with lag	0.661174	3.880314	0.366743
	Full dataset	0.661591	3.255859	0.351224
Cap=1	Previous target only	0.039996	0.257263	0.383273
	SG without lag	0.220832	0.208802	0.339230
	SG with lag	0.260287	0.171187	0.293761
	Full dataset	0.243215	0.170585	0.291978
Cap=2	Previous target only	0.056657	0.394342	0.430185
	SG without lag	0.222160	0.325158	0.387359
	SG with lag	0.260144	0.262191	0.331597
	Full dataset	0.236838	0.259539	0.328155
Cap=5	Previous target only	0.093560	0.758363	0.494951
	SG without lag	0.230357	0.643913	0.458689
	SG with lag	0.271977	0.496196	0.383934
	Full dataset	0.241795	0.482256	0.376391
Cap=10	Previous target only	0.136065	1.255914	0.535752
	SG without lag	0.246081	1.095983	0.508804
	SG with lag	0.296034	0.807206	0.416582
	Full dataset	0.264640	0.771622	0.405525
Cap=100	Previous target only	0.395070	4.694559	0.539570
	SG without lag	0.421073	4.492761	0.542507
	SG with lag	0.524254	2.613492	0.404590
	Full dataset	0.502205	2.239160	0.392759

Figure 4. Training set metrics for different outlier caps and datasets.

Outlier Cap	Dataset	Val R2	Val MSE	Val MAE
No cap	Previous target only	0.604408	2.886780	0.315256
	SG without lag	0.606194	2.873745	0.337933
	SG with lag	0.614091	2.816122	0.347874
	Full dataset	0.584153	2.526766	0.373648
Cap=1	Previous target only	-0.127136	0.204343	0.374237
	SG without lag	0.120659	0.159419	0.304102
	SG with lag	0.163677	0.151620	0.275704
	Full dataset	0.139318	0.151381	0.281535
Cap=2	Previous target only	-0.073640	0.288253	0.415347
	SG without lag	0.132617	0.232876	0.340742
	SG with lag	0.185846	0.218585	0.307205
	Full dataset	0.146106	0.219831	0.317297
Cap=5	Previous target only	0.014488	0.492993	0.469943
	SG without lag	0.158051	0.421177	0.393280
	SG with lag	0.227065	0.386653	0.350471
	Full dataset	0.166884	0.390127	0.374229
Cap=10	Previous target only	0.088449	0.741425	0.500879
	SG without lag	0.188259	0.660243	0.427800
	SG with lag	0.268528	0.594955	0.379105
	Full dataset	0.195151	0.600374	0.422827
Cap=100	Previous target only	0.399833	2.489540	0.468988
	SG without lag	0.404260	2.471175	0.428775
	SG with lag	0.504778	2.054218	0.384693
	Full dataset	0.426866	1.972305	0.412221

Figure 5. Validation set metrics for different outlier caps and datasets.