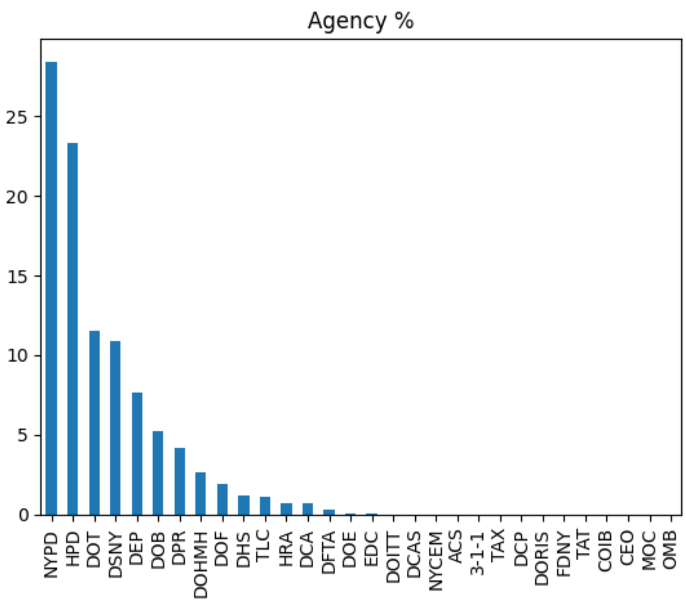
1. Introduction

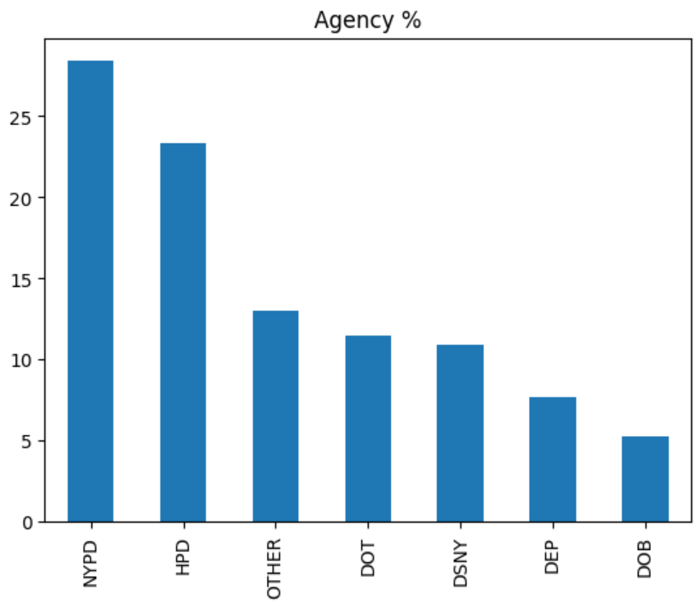
The 311 helpline, initially launched in Baltimore in 1996, diverts non-emergency concerns from 911, providing access to various municipal services. Over 70 U.S. cities offer 311 services, experiencing a surge in demand (which could be validated by the trend component in our dataset). This exercise explores the potential to forecast 311 call daily volume using (1) the information from the 311 call itself (2) combined with weather information as leading factors. Given the rich spatial and temporal information in both datasets, we experimented with various ways to extract valuable information in the feature engineering stage and implemented three model families with different levels of complexity to compare the out-of-sample forecasting power.

Section 2 describes the data processing steps and key preliminary visualizations to understand the patterns of 311 data and weather data in sequential order. In section 3, we implemented various pipelines to evaluate the feature importance when predicting contemporary 311 call volume in a hybrid manner of unsupervised and supervised learning. In section 4, we built a time series model using the information from 311 data only as a benchmark. Section 5 describes our proposed linear and nonlinear family models to predict the future 7-day 311 call volume using lagged features. Section 6 concludes and discusses potential questions from the audience.

2. Data Exploration and Processing

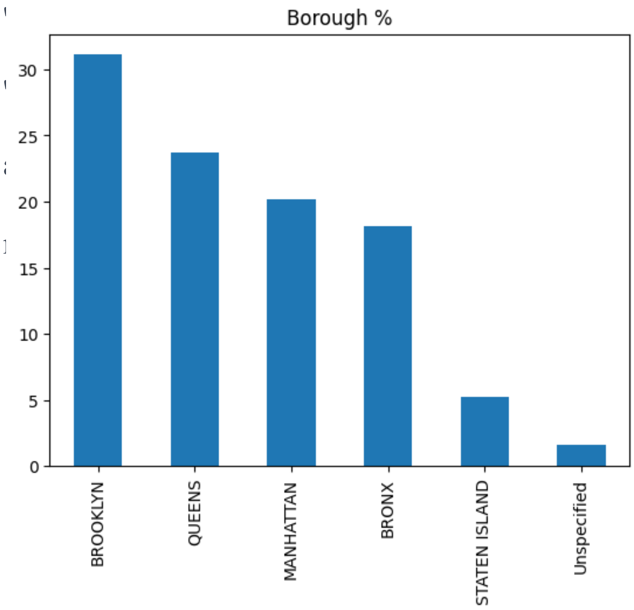
In this section, we start with several important categorical variables in 311 data to understand the underlying reason that (a) a specific complaint happens (b) in a specific region (c) during a specific season, then (d) processed by a specific agency.





Out of all agency types, the NYPD/HPD respond to the majority (approximately 25% each) in New York City.

Conversely, the bottom 24 agencies (('DPR', 'DOHMH', 'DOF', 'DHS', 'TLC', 'HRA', 'DCA', 'DFTA', 'DOE',



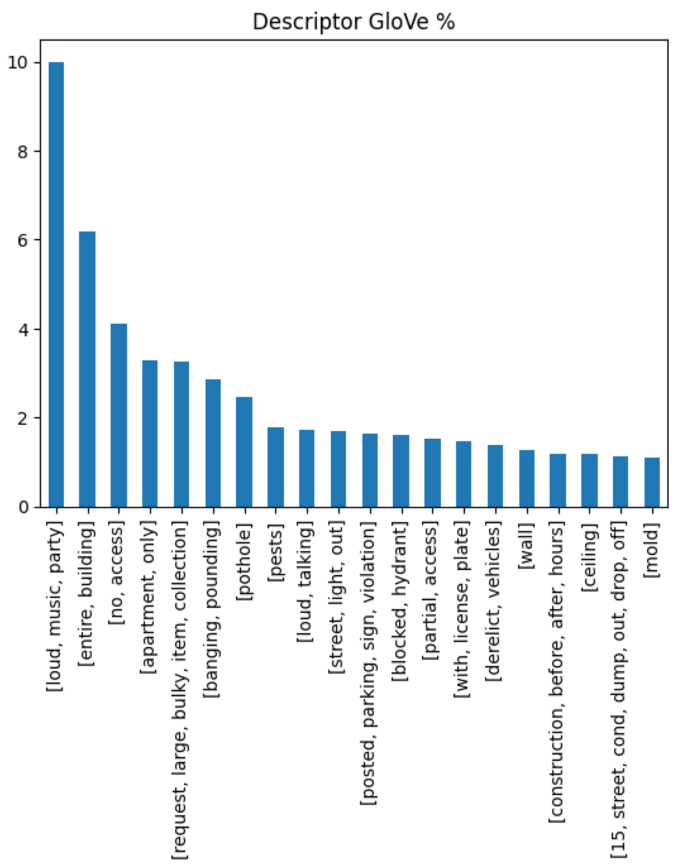
l-1', 'T

d respc

wn lool

ntire sa

Agency	Borough	Complaint Type	
NYPD	BROOKLYN	Noise - Residential	198167
	BRONX	Noise - Residential	167881
	MANHATTAN	Noise - Residential	148578
	QUEENS	Noise - Residential	132889
	MANHATTAN	Noise - Street/Sidewalk	83985
DEP	MANHATTAN	Noise	83967
NYPD	BROOKLYN	Noise - Street/Sidewalk	57972
	MANHATTAN	Noise - Commercial	55203
	BROOKLYN	Noise - Commercial	48055
DEP	BROOKLYN	Noise	46472
NYPD	BRONX	Noise - Street/Sidewalk	42021
DEP	QUEENS	Noise	31413
NYPD	MANHATTAN	Noise - Vehicle	27125
	BROOKLYN	Noise - Vehicle	23729
	QUEENS	Noise - Commercial	22702
	STATEN ISLAND	Noise - Residential	20696
	QUEENS	Noise - Street/Sidewalk	20310
		Noise - Vehicle	17768
	BRONX	Noise - Vehicle	15902
		Noise - Commercial	11980
DEP	BRONX	Noise	9338
	STATEN ISLAND	Noise	5249

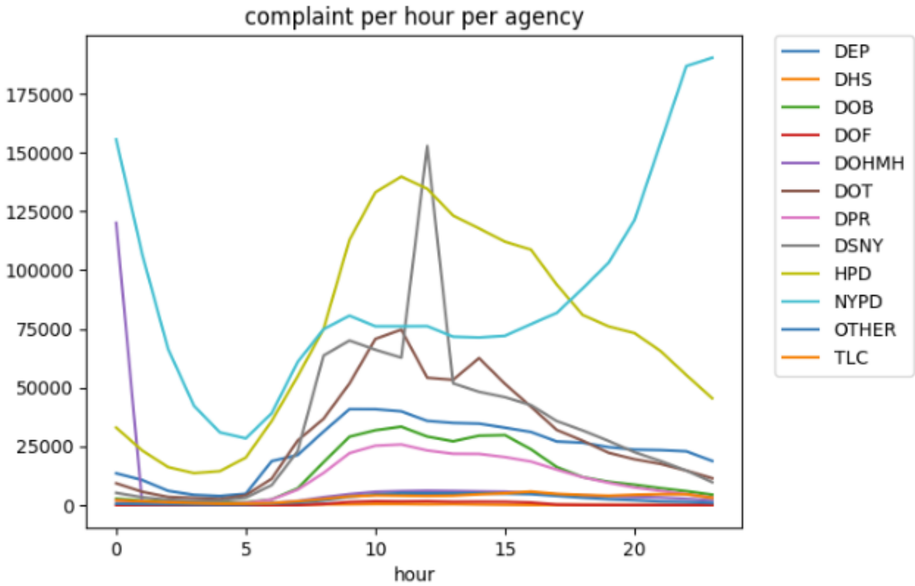


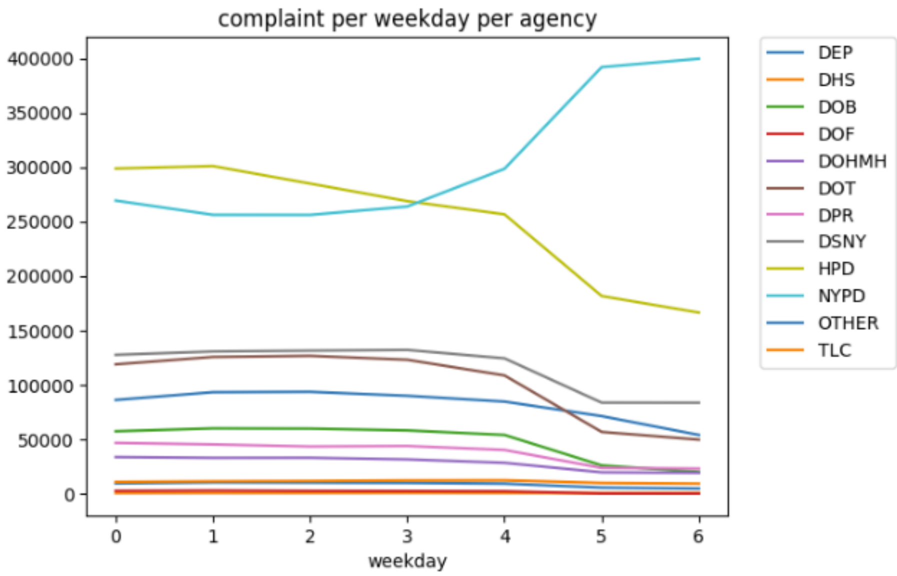
In the Borough category, we saw there are only 6 types and the proportion is well spread. We further run double-grouping to get the above table: noise is the top type of complaint coming from various channels. NYPD received the most calls across all the boroughs. In terms of location, Manhattan's top contributor is street noise while other areas' are residential. DEP generally comes with a top-level type denoted as "noise". As a further step, we take extra care on the Descriptor column, which brings 1000+ types (instead of the Resolution Description since its content is pretty standardized), and use Google GloVe to encode the text info to 200-dim word embedding. The resulting word embedding could help us understand 200+ types in the Compliant column. The most frequent list of words matched by GloVe is shown in the figure above: there are multiple themes going on in each single complaint script, such as 'noise x inside', 'noise x outside', 'vehicle x illegal plate', 'vehicle x illegal parking'. Hence, we run a hierarchical clustering algorithm using GloVe with other 311 features and report the detailed result in section 3.

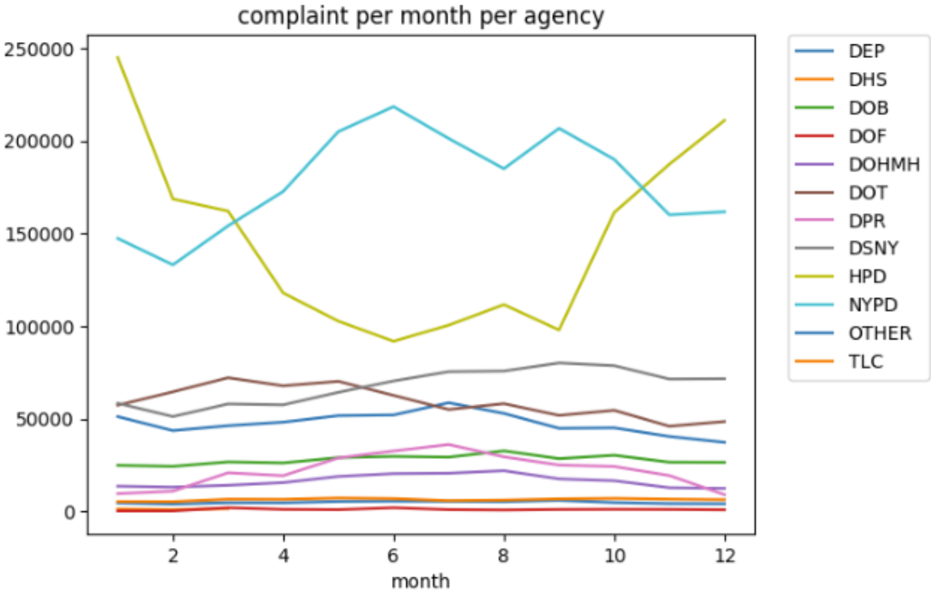
Agency	Complaint Type	
NYPD	Noise - Residential	668223
HPD	Heat/Hot Water	658390
NYPD	Illegal Parking	427948
	Blocked Driveway	391037
HPD	Unsanitary Condition	245925
NYPD	Noise - Street/Sidewalk	206837
HPD	Paint/Plaster	176766
	Plumbing	155705
NYPD	Noise - Commercial	140768
HPD	Door/Window	110528
	Water Leak	103281
NYPD	Derelict Vehicle	102206
	Noise - Vehicle	86544
HPD	Electric	81385
	General	77840

For the two top contributors in the agency category, we plot the map with the complaint type as hue. NYPD is mainly driven by noise and vehicles, while HPD is driven by water and electricity. A natural hypothesis is that for different types of complaints, their data generating process may be influenced by various seasonality and weather impacts. We further plot the time series per agency at various time frequencies.

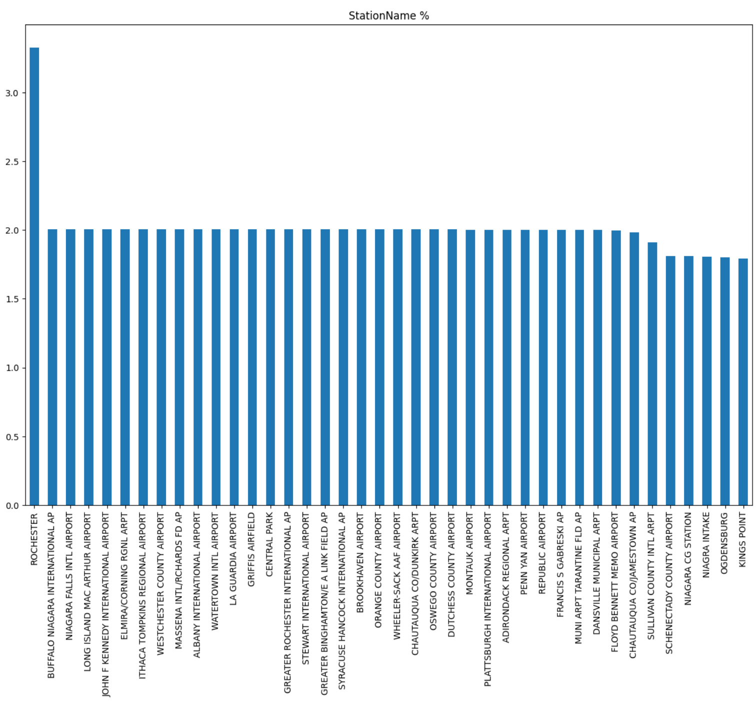
Only the NYPD receives most calls during nighttime (weekends), while others receive most during daytime (weekday). NYPD volume peaked in summer, HPD volume peaked in winter, while there is no significant annual seasonality for other agencies. This further consolidates our decision to add the agency feature in the complaint type clustering task in the next stage.

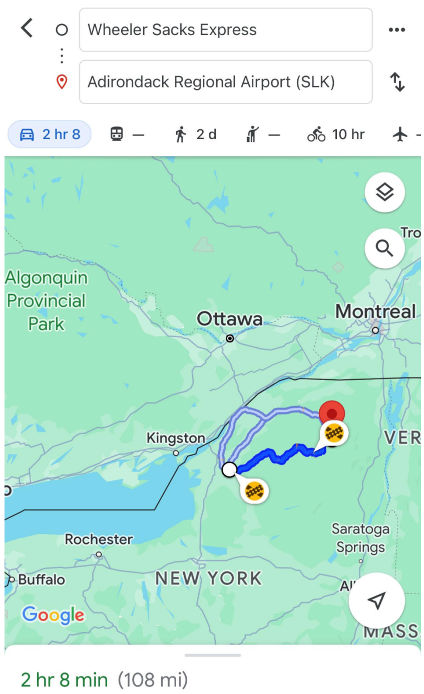

complaint per hour per agency
175000
150000
125000
100000
75000
50000
25000
0
0
5
10
15
20
hour
DEP
DHS
DOB
DOF
DOHMH
DOT
DPR
DSNY
HPD
NYPD
OTHER
TLC


complaint per weekday per agency
400000
350000
300000
250000
200000
150000
100000
50000
0
0
1
2
3
4
5
6
weekday
DEP
DHS
DOB
DOF
DOHMH
DOT
DPR
DSNY
HPD
NYPD
OTHER
TLC



After getting a grasp of the 311 call data distribution, we added the weather data to our analysis and conducted similar preliminary visualization. Given 50+ weather stations, we use the nearest neighbor method to merge the weather feature with a corresponding community board to keep the spatial weather variation as much as we can (instead of only using the 6 boroughs). Specifically, we choose the number of stations K = 5 to avoid severe missing value issues since we don't have the entire weather station coverage on each day, at the minor cost of smoothing out some spatial variation.





Similarly, we draw the time series plot for various numeric weather features to investigate seasonality issue (see the appendix). Temperature, dewpoint, and snow depth exhibit the most significant annual seasonality, while wind features come second, and precipitation remains last. For the two categorical variables (Rain and

SnowIce), we observe a high linear correlation among stations (pearson 0.86, spearman 0.92), and there is considerable overlap in the top 10 table. However, we decided to keep both since they are influenced by multiple weather factors in a super nonlinear way. For example, Wheeler is top Rain station, which also appears in the top SnowIce table due to its high latitude and proximity to Lake Ontario. On other hand, Adirondack is top SnowIce station, and it does not appear in top Rain list due to its even higher latitude and inland location.

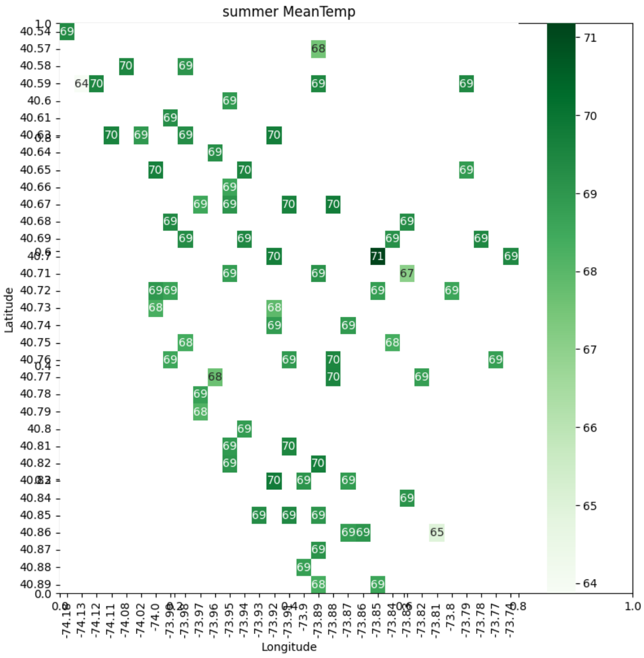
StationName	Rain
WHEELER-SACK AAF AIRPORT	3.864816839803171
SYRACUSE HANCOCK INTERNATIONAL AP	3.3385729907053032
GRIFFIS AIRFIELD	3.2463094587206123
BUFFALO NIAGARA INTERNATIONAL AP	3.171131765992345
OSWEGO COUNTY AIRPORT	3.1506287588846362
MUNI ARPT TARANTINE FLD AP	3.1267085839256423
GREATER ROCHESTER INTERNATIONAL AP	3.1027884089666484
ITHACA TOMPKINS REGIONAL AIRPORT	3.0583652268999453
NIAGARA FALLS INTL AIRPORT	3.048113723346091
CHAUTAUQUA CO/DUNKIRK ARPT	3.044696555494806

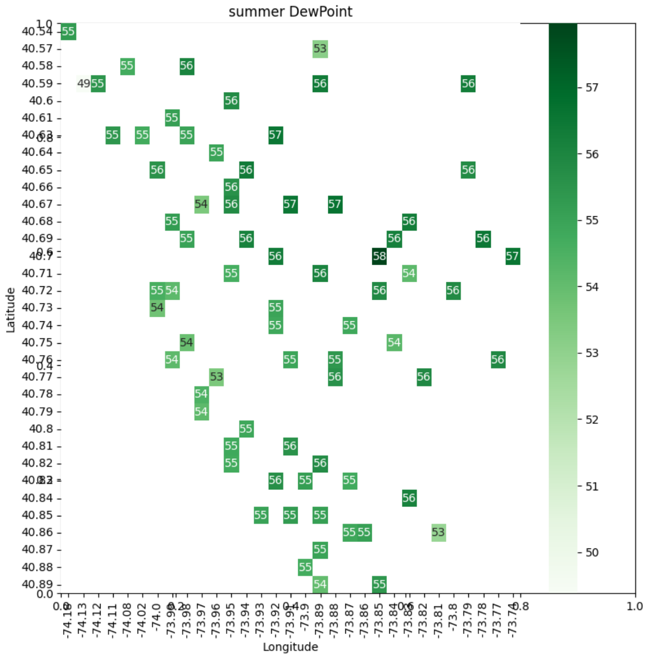
StationName	SnowIce
ADIRONDACK REGIONAL ARPT	5.3299692501774025
SYRACUSE HANCOCK INTERNATIONAL AP	4.257667744224553
GREATER ROCHESTER INTERNATIONAL AP	4.202475754947567
MUNI ARPT TARANTINE FLD AP	4.155168335567295
BUFFALO NIAGARA INTERNATIONAL AP	4.139399195773871
WHEELER-SACK AAF AIRPORT	4.139399195773871
GREATER BINGHAMTON/E A LINK FIELD AP	4.013246077426476
ITHACA TOMPKINS REGIONAL AIRPORT	3.8870929590790824
GRIFFIS AIRFIELD	3.8397855396988096
OSWEGO COUNTY AIRPORT	3.8161318300086733

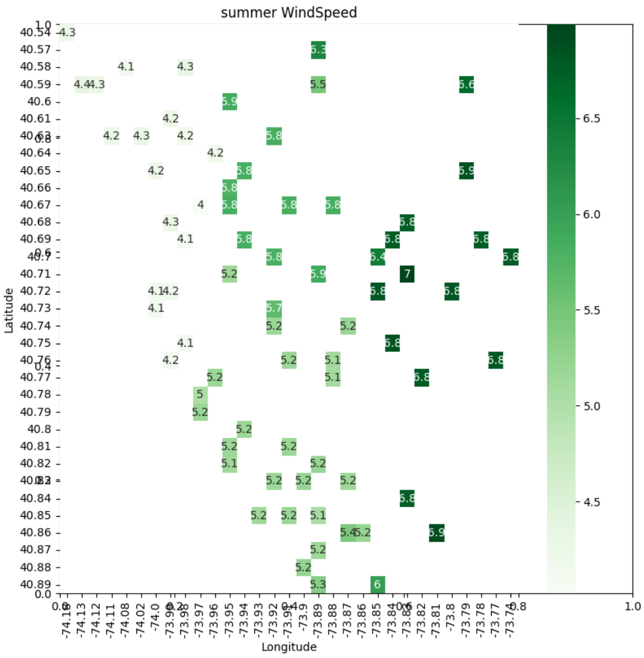
For the numeric features, we plot the map by summer and winter and make sense of the seasonality pattern.

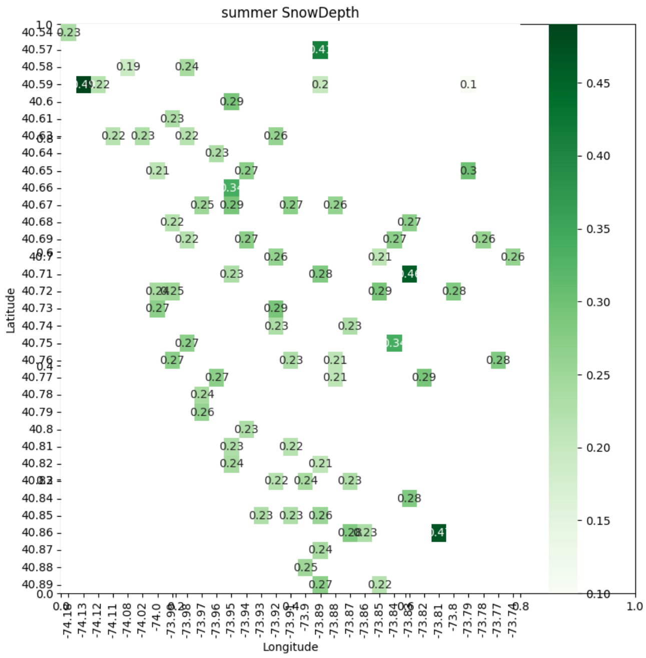
(a) The area with the highest temperature is located in Manhattan, Queens, Brooklyn, and Staten Island, while the "left inland" and "right Atlantic ocean coast" are cooler.
(b) DewPoint is consistently lower in winter and highly correlated with temperature across areas.
(c) Wind Speed and snow depth show similar geographical distribution, where the "right Atlantic ocean coast" area has the largest values given the same latitude.

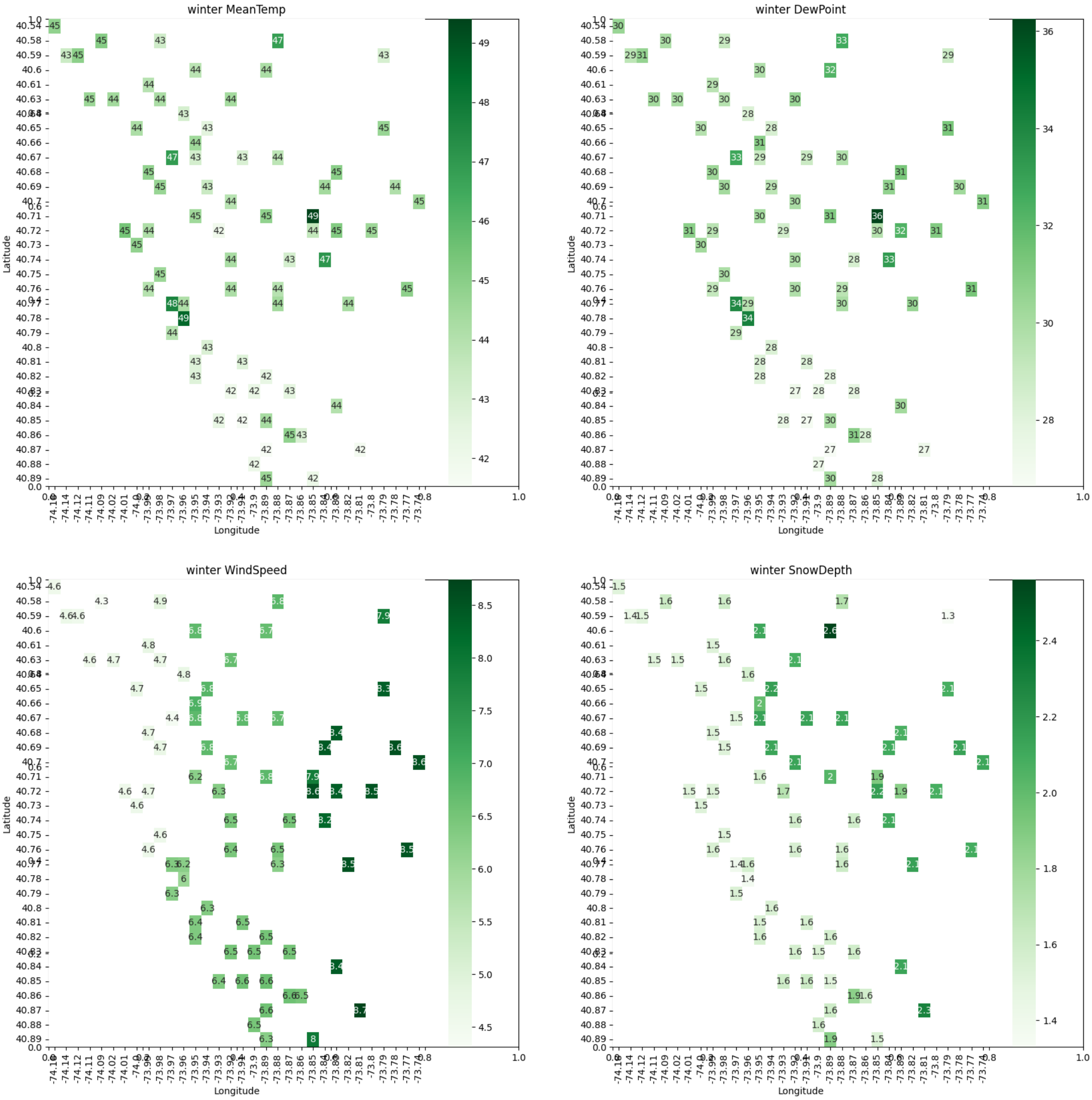
The above pattern further consolidates our (1) nearest-neighbor-based data merging to improve SNR (2) keep complaint cluster information in the modeling to leverage the spatial variation.


summer MeanTemp
Latitude
Longitude


summer DewPoint
Latitude
Longitude


summer WindSpeed
Latitude
Longitude


summer SnowDepth
Latitude
Longitude



3. Feature Evaluation and Visualization

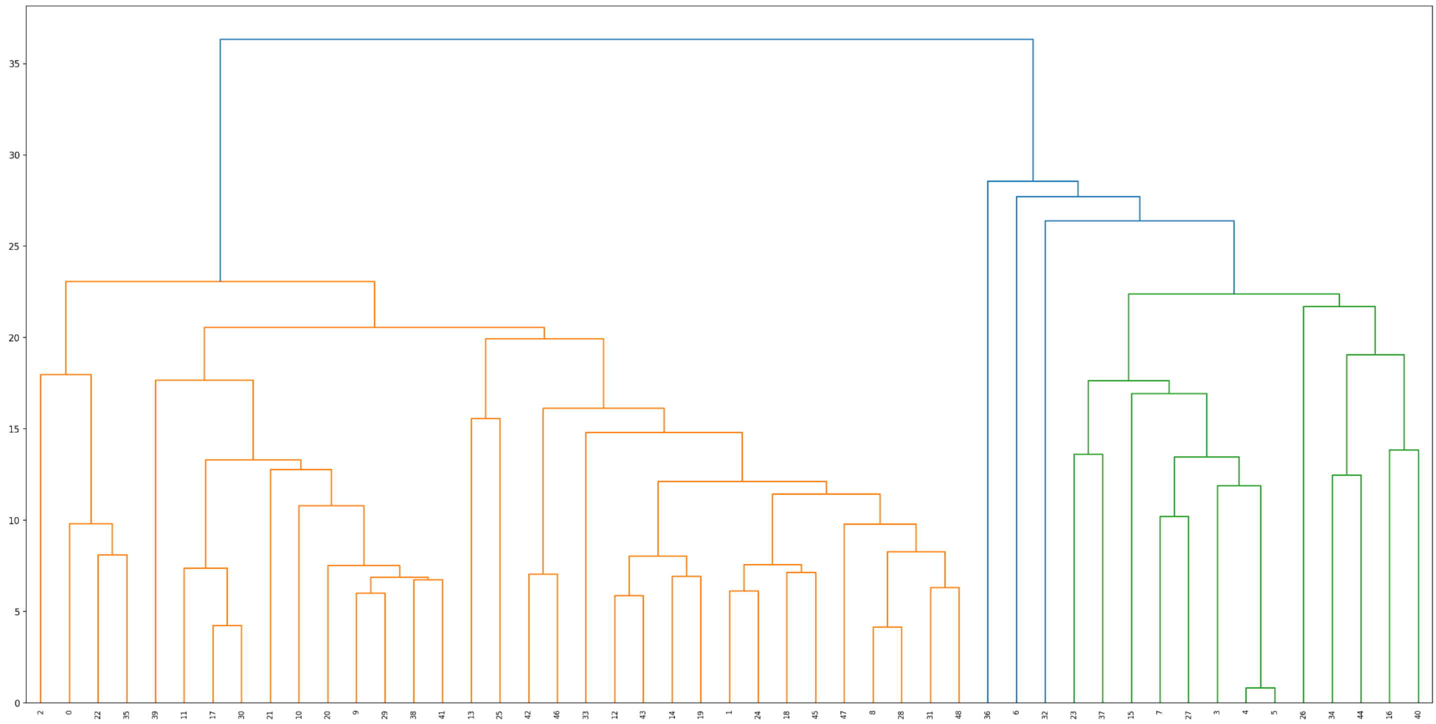
We would like to incorporate spatial weather data to predict the call volume in our proposed models. An intuitive starting point is to determine which themes in the complaint type column could be better predicted by such predictors. For example, during a hurricane, there may be more instances of trees damaged, extreme traffic, and power outages than usual. On the other hand, weather dynamics might not significantly impact issues like residential noise. Therefore, for such issues, we anticipate seeing more predictive power from time series analysis in our baseline model, which only utilizes information from the 311 data.

In this section, we first employ manifold learning and hierarchical clustering to visualize the complaint types. Then, we use random forest to assess the feature importance for predicting categorical classification as well as the target call volume. Both the unsupervised and supervised approaches can help us gauge the feature selection for our proposed models.

In the unsupervised learning part, we leverage t-SNE and PCA to visualize the GloVe embedding combined with other variables into low-dimensional space. Given the computational cost for such a large dataset, we use stratified sampling to get the right amount of samples while keeping it representative.

We do the visualization using 2 feature sets: *(a) GloVe embedding, duration, location, (b) GloVe embedding, seasonality at various frequencies, agency encoding*. See the appendix for more figures.

Given the large number of labels, manifold learning didn't perform well, we seek clustering to reduce the number of complaint types to 5 (based on 30-dim feature distance and sample size in each cluster).



Cluster 1: 24%

['Appliance', 'Air Quality', 'Indoor Air Quality', 'Safety', 'Snow', 'Door', 'Flooring/Stairs', 'Paint/Plaster', 'Illegal Parking', 'Dirty Conditions', 'Graffiti', 'Curb Condition', 'Other Enforcement', 'Sidewalk Condition', 'Street']

Cluster 2: 37%

['Electronics Waste', 'Missed Collection (All Materials)', 'Taxi Complaint', 'Vehicle', 'Rodent', 'Electric', 'Traffic', 'Electronics Waste Appointment', 'General', 'Animal', 'Maintenance Or Facility', 'Food', 'Unsanitary Condition', 'Vending', 'Consumer Complaint', 'OTHER', 'Plumb', 'Water']

Cluster 3: 7%

['Sanitation Condition', 'Building/Use', 'Request Large Bulky Item Collection']

Cluster 4: 10%

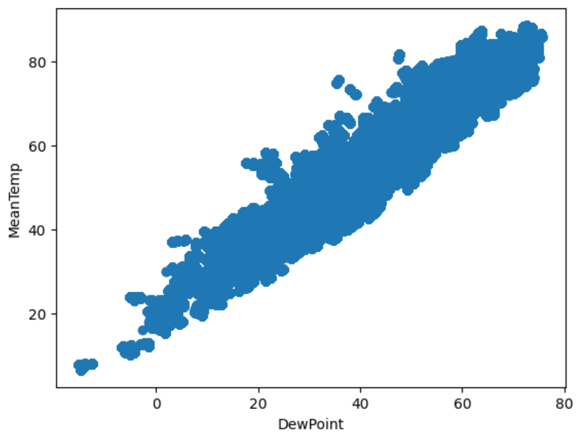
['Lead', 'Sewer', 'Elevator', 'Construction', 'Non-Emergency Police Matter', 'Blocked Driveway', 'Broken Muni Meter', 'Broken Parking Meter']

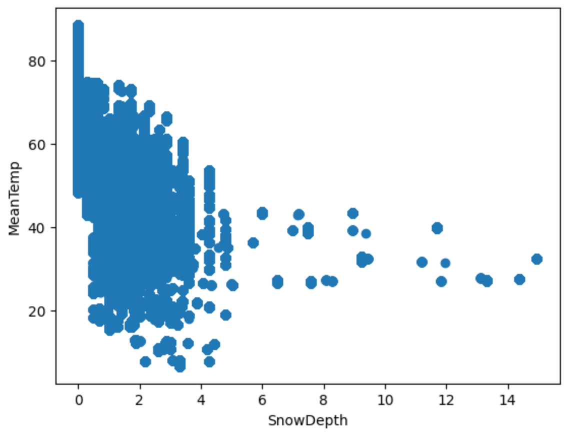
Cluster 5: 22%

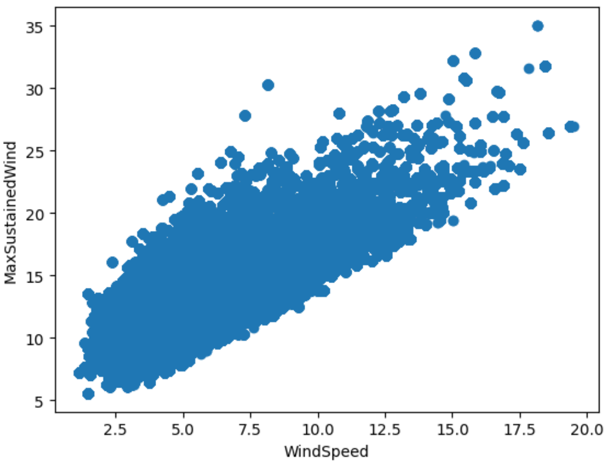
['Noise', 'Root/Sewer/Sidewalk Condition', 'Tree', 'Emergency Response Team (Ert)', 'Special Projects Inspection Team (Spit)']

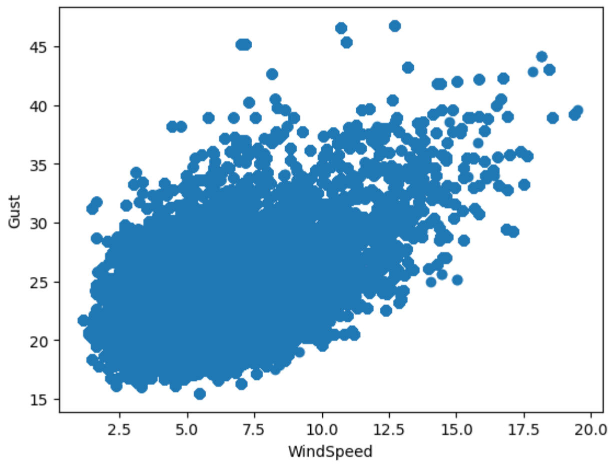
In the next supervised learning stage, we have two types of grouping in the merged dataset, each with a handful of distinct values: Borough and Complaint Cluster. The question we would like to answer is: in the span of weather features (i.e. *['IsWinter_Winter', 'IsWeekend_Weekend', 'Up', 'Down', 'DewPoint', 'Precipitation', 'Wind_1', 'Wind_2', 'Rain', 'SnowDepth', 'SnowIce']*), which grouping criteria is more of a "clean-cut" among different groups. Someone will vote for the Borough if they think the variation across geography locations dominate different levels of forecasting power on 311 call. The other side who voted for Complaint Cluster believes the text information extracted from word embedding in the previous unsupervised learning stage matters, which will help differentiate the nature of different types of events. Hence, we carry out some analysis on the grouping scheme before conducting the standard feature evaluation when the target variable is the contemporary daily call volume.

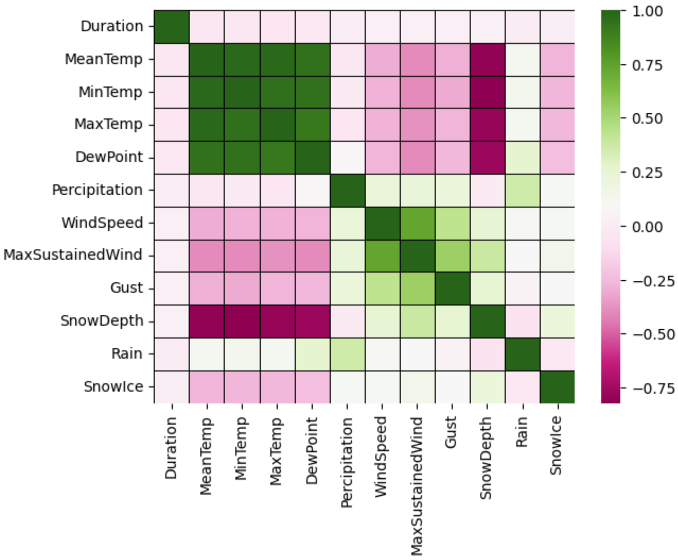
We created features 'Up'/'Down' to replace 'MaxTemp'/'MinTemp', then removed MeanTemp due to its large correlation with DewPoint and SnowDepth. Besides, we took the first 2PC of wind group to reduce collinearity.

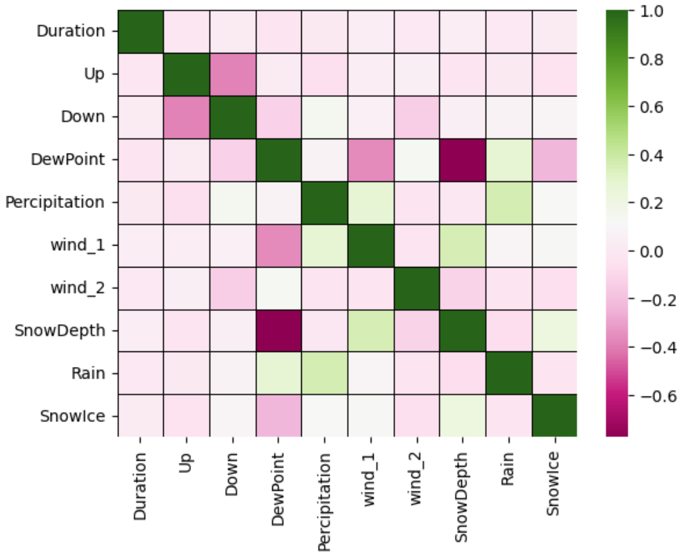


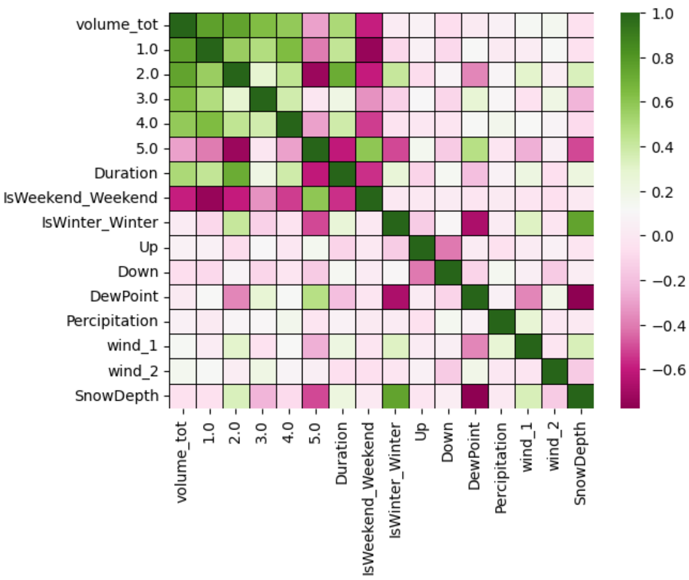


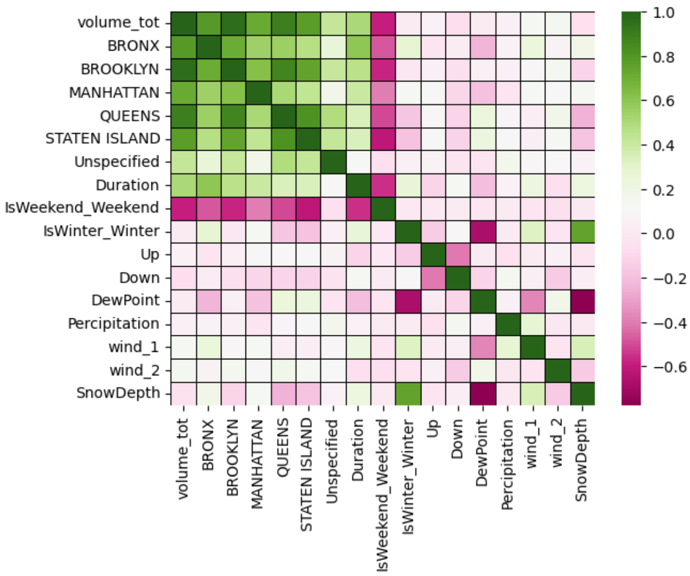

MaxSustainedWind
WindSpeed


Gust
WindSpeed


Duration
MeanTemp
MinTemp
MaxTemp
DewPoint
Percipitation
WindSpeed
MaxSustainedWind
Gust
SnowDepth
Rain
SnowIce


Duration
Up
Down
DewPoint
Percipitation
wind_1
wind_2
SnowDepth
Rain
SnowIce


volume_tot
1.0
2.0
3.0
4.0
5.0
Duration
IsWeekend_Weekend
IsWinter_Winter
Up
Down
DewPoint
Percipitation
wind_1
wind_2
SnowDepth

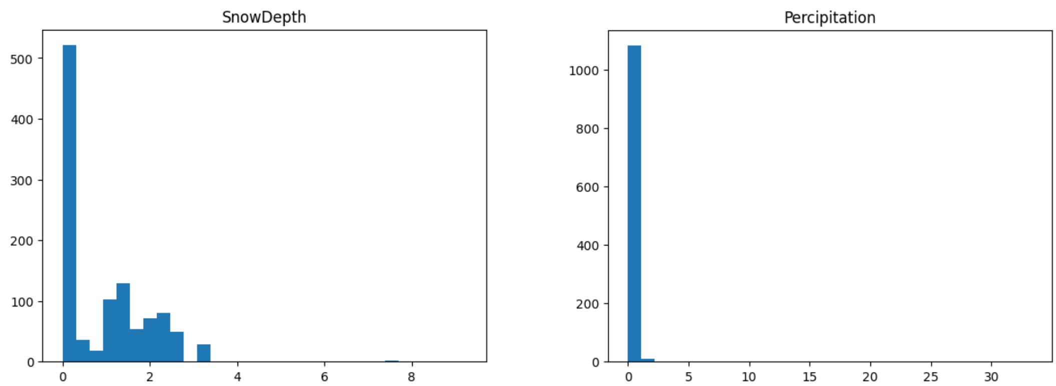

volume_tot
BRONX
BROOKLYN
MANHATTAN
QUEENS
STATEN ISLAND
Unspecified
Duration
IsWeekend_Weekend
IsWinter_Winter
Up
Down
DewPoint
Percipitation
wind_1
wind_2
SnowDepth

By fitting a random forest model to classify Borough, we achieve an out-of-sample accuracy of 0.56. In contrast, for the Complaint Cluster classification task, we obtain an out-of-sample accuracy of 0.37. This indicates weather feature can explain more variation across Boroughs. Please refer to appendix for more details.

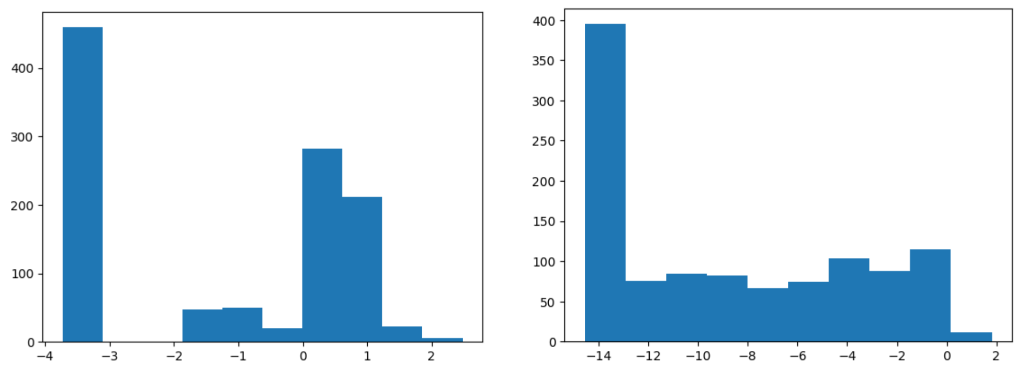
In the above correlation matrices, we saw the total volume has a positive correlation with all boroughs. Most of the borough-level volume has a positive correlation with complaint processing duration, negatively correlated with IsWeekend_Weekend and IsWinter_Winter. Only Bronx volume is positively correlated with IsWinter_Winter.

There is a more interesting pattern in the complaint cluster-level breakdown: cluster 5 volume is negatively correlated to all other clusters as well as the total volume. It demonstrates cluster 5 are those short-lived/ emergent complaints that happened over the weekend during summertime. This can also be validated by its negative correlation with SnowDepth and positive correlation with DewPoint. Cluster 2 is the opposite side: long-existing problems happened during wintertime.

Lastly, it is worth mentioning that we conducted a Box-Cox transformation on the Precipitation and SnowDepth variables due to their highly positively-skewed distribution before proceeding with standardization in the modeling stage.

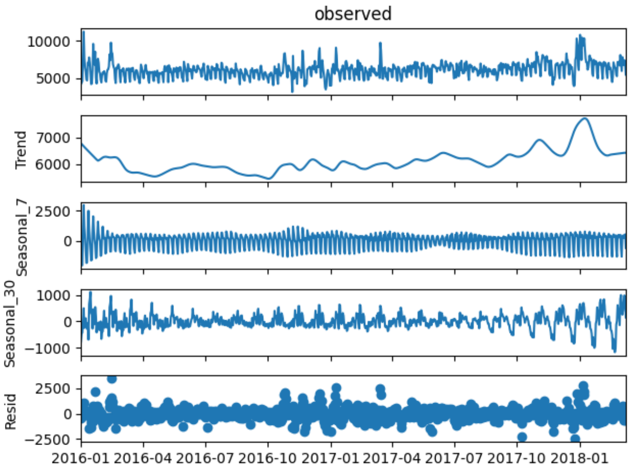


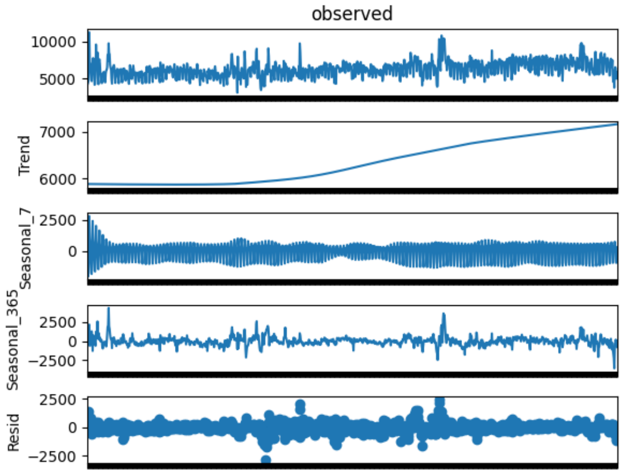
After box-cox transform:



4. Baseline Time Series Analysis (SARIMA)

In this section, we reported the forecasting result when using information solely from 311 call data using various implementation vehicles. We started with the time series decomposition plot below, to understand the long-term trend, and weekly/monthly/annual seasonalities if there are any.

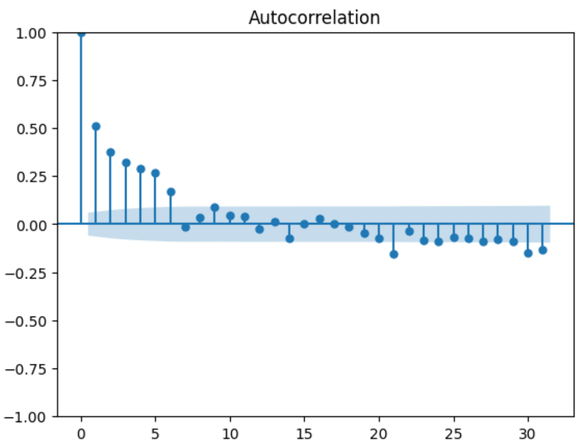


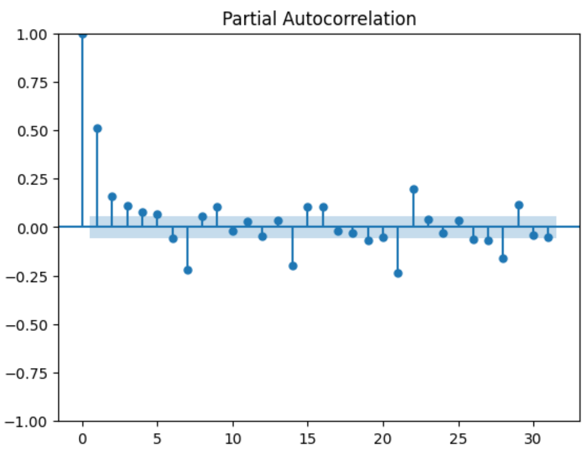


The baseline model 1 is selected by the auto_arima function: *ARIMA(1,0,0)(2,0,1)[7] intercep*t, where we conduct model fitting at each single run in a weekly rolling manner. In our baseline model 2, we fixed the above model setting for the prediction across different clusters, then manually added them up to get the total volume as the final output. In our proposed linear and nonlinear models in the next section, we also did some time series modeling and plotted the following diagnostic chart to gauge how much lagged information we may include (ADF test and ACF / PACF plot on the residual series).

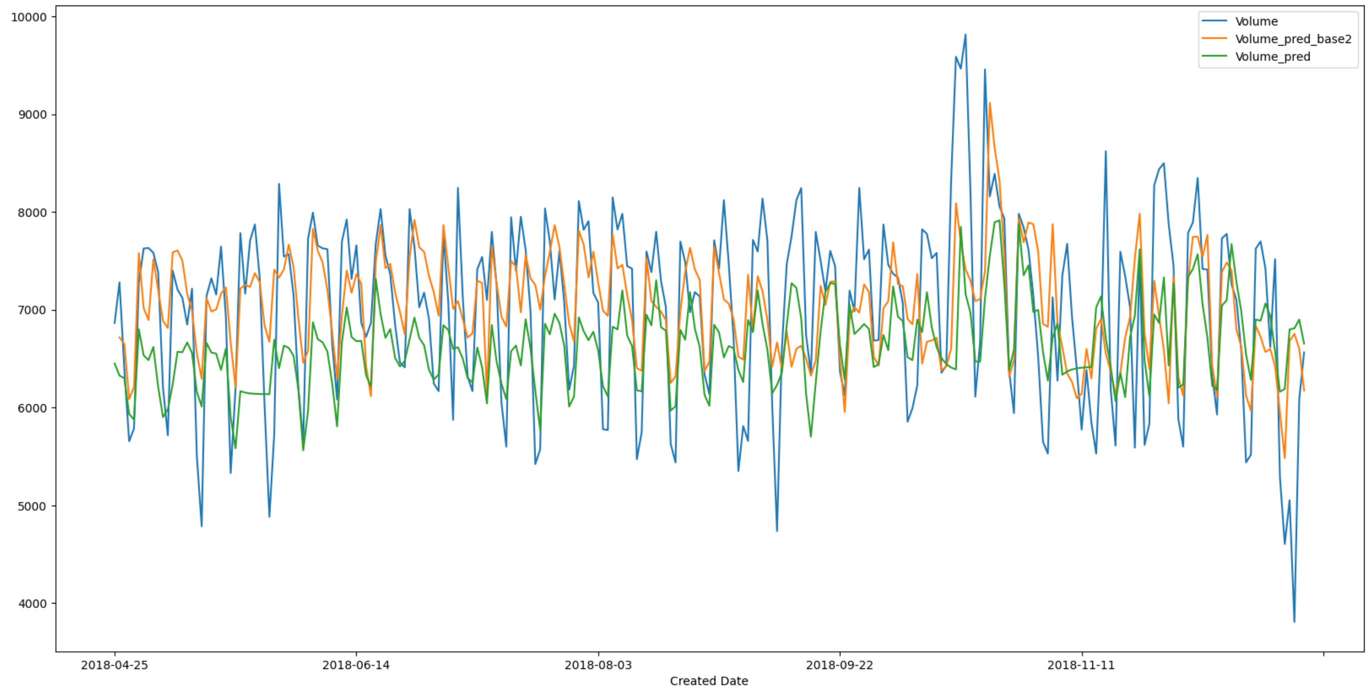
ADF Statistic: -6.70, p-value: 0.00, Critical Values: 1%: -3.43, 5%: -2.86, 10%: -2.56.

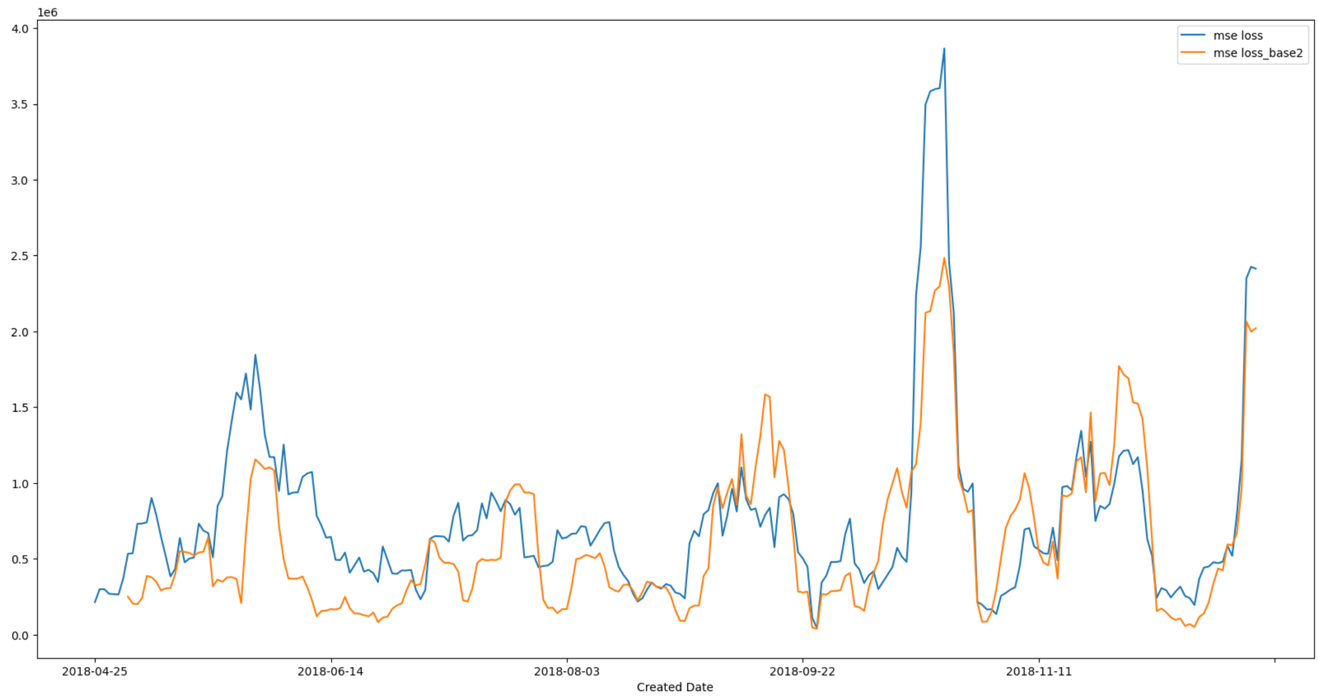
ARIMA(3,0,6) for the residual only.





We define our loss metric as the daily mean squared error in a trailing 7-day window in the test set for all the models. The results of baseline model 1 (loss = 757356) and 2 (loss = 616343) are reported as follows. Baseline model 2 can capture a more accurate magnitude of short-term shock compared to model 1. We also reported the time series prediction result for each cluster; please refer to the appendix for details.





5. Linear Predictive Models (PLS) and Nonlinear Family (LSTM)

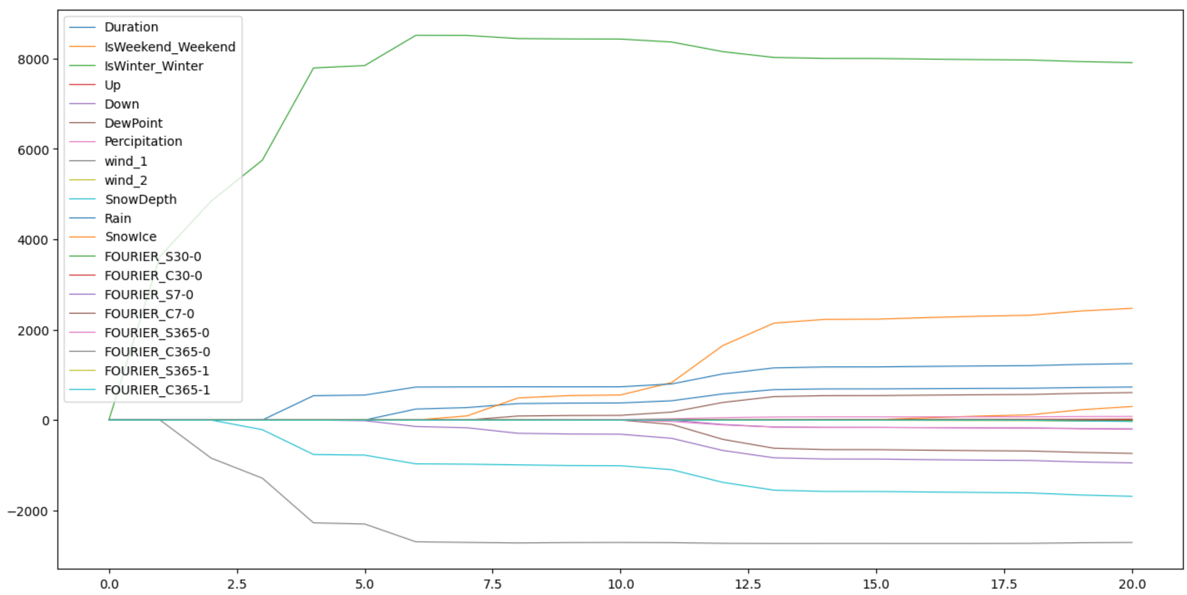
In this section, we proposed both linear and nonlinear models to predict daily 311 call volume with weather data. We formulated the problem in a multi-dimensional target prediction fashion (e.g. multiple boroughs), where the dimensionality of target and predictor are T and P, respectively. Given the data format, Partial Least Square (PLS hereafter) and Long-Short Term Memory (LSTM hereafter) are the appropriate models with a good balance between model complexity and forecasting power.

Given the result of the feature evaluation, in addition to the existing weekly/monthly seasonality components in baseline models, we add the annual seasonality from all the numeric weather features into our feature set. Before inputting our prepared dataset to either linear or nonlinear models, we conduct standardization on most of the numeric features to ensure model stability.

We first do a rough train test split, where the first 821 (621) samples are the training set and the remaining are the testing set for PLS (LSTM). In order to get some qualitative understanding on the impact of different features, we report the summary of ridge regression when the target variable is: (a) the total call volume each day (b) daily breakdown by borough (c) daily breakdown by cluster. In the later stage, we will conduct a rolling window train-test split to evaluate our model performance for PLS (d) and LSTM (e). For the results of (b) and (c), please see the appendix.

(a)

We start with a feature set: ['Duration', 'IsWeekend_Weekend', 'IsWinter_Winter', 'Up', 'Down', 'DewPoint', 'Percipitation', 'wind_1', 'wind_2', 'SnowDepth', 'Rain', 'SnowIce', 'FOURIER_S30-0', 'FOURIER_C30-0', 'FOURIER_S7-0', 'FOURIER_C7-0', 'FOURIER_S365-0', 'FOURIER_C365-0', 'FOURIER_S365-1', 'FOURIER_C365-1'], then use regularized regression to drop redundant features. By Lasso path, we drop the 3 bottom-ranked Fourier features and further drop 2 more insignificant features 'Up' and 'FOURIER_S365-1' due to its insignificance in Ridge regression.



	0
IsWinter_Winter	1
FOURIER_C365-0	2
SnowDepth	3
Rain	4
FOURIER_S7-0	5
Duration	6
IsWeekend_Weekend	7
FOURIER_C7-0	8
FOURIER_S365-0	9
Percipitation	10
DewPoint	11
Down	12
Up	13
FOURIER_S365-1	14
SnowIce	15
wind_1	16
FOURIER_S30-0	17
FOURIER_C365-1	18
wind_2	19
FOURIER_C30-0	20

```
Residuals:
       Min        1Q   Median       3Q       Max
-3395.0362 -441.3723 25.9031 453.0345 2251.6609


Coefficients:
               Estimate  Std. Error   t value   p value
_intercept   6219.279727   59.205627  105.0454  0.000000
x1            387.369513   32.499473   11.9193  0.000000
x2           -406.519163  103.574381   -3.9249  0.000094
x3            118.861129  108.587048    1.0946  0.274006
x4              6.858049   28.071883    0.2443  0.807057
x5            -88.887071   28.517972   -3.1169  0.001892
x6           -374.643973   52.872711   -7.0858  0.000000
x7             25.378694   28.224012    0.8992  0.368816
x8            119.453245   28.774068    4.1514  0.000036
x9             49.370241   26.433434    1.8677  0.062158
x10          -265.354267   62.013523   -4.2790  0.000021
x11           -94.704479   62.849904   -1.5068  0.132238
x12          -254.677390  110.893874   -2.2966  0.021894
x13          -208.877312   31.061003   -6.7247  0.000000
x14            21.455329   29.413179    0.7294  0.465937
x15           -80.383463   37.118614   -2.1656  0.030631
x16           -90.288688   69.667592   -1.2960  0.195343
x17           -12.297923   25.989125   -0.4732  0.636200
---
R-squared:  0.49791,    Adjusted R-squared:  0.48728
F-statistic: 46.84 on 17 features
```

The ridge regularization alpha is selected to be 10 from cross-validation, resulted in 15-dim feature set ['Duration', 'IsWeekend_Weekend', 'IsWinter_Winter', 'Down', 'DewPoint', 'Precipitation', 'wind_1', 'wind_2', 'SnowDepth', 'Rain', 'SnowIce', 'FOURIER_S7-0', 'FOURIER_C7-0', 'FOURIER_S365-0', 'FOURIER_C365-0']. Duration definitely significantly contributes to the surge of total volume since that means the working capacity among agencies is not enough during peak season. The upward temperature momentum 'Up' is not significant but the downside 'Down' is negatively correlated with volume, which means an extreme downward intraday movement will lead to more calls. Wind_1 and Wind_2 both positively contribute to more calls. In terms of seasonality, only weekly and annual Fourier terms are kept at the end.

(d)

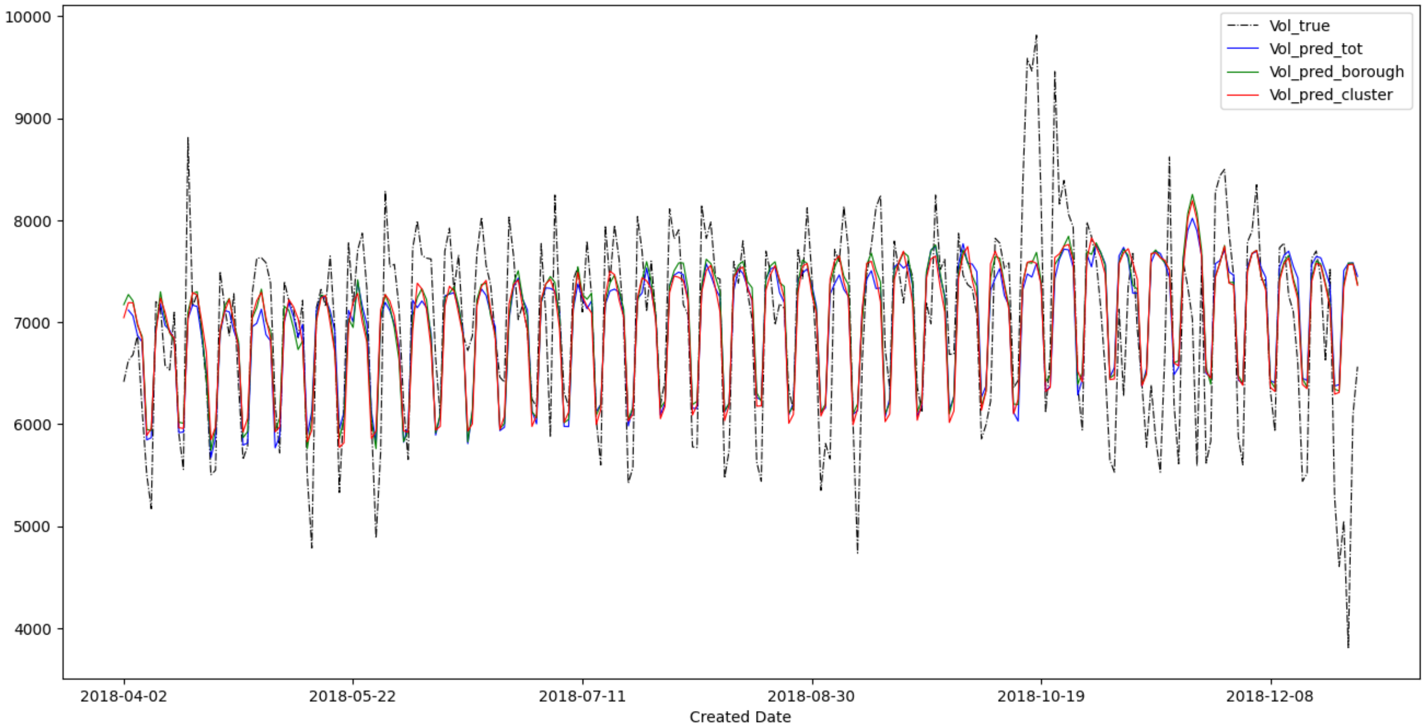
In this subsection on linear models, we initially conducted an out-of-sample test using the 15-dimensional feature, and the proposed models did not significantly outperform the baseline models. Therefore, considering our previous time series diagnosis, we decided to reintroduce the time series components of call volume itself to capture the trend at various frequencies, resulting in a slight improvement in performance.

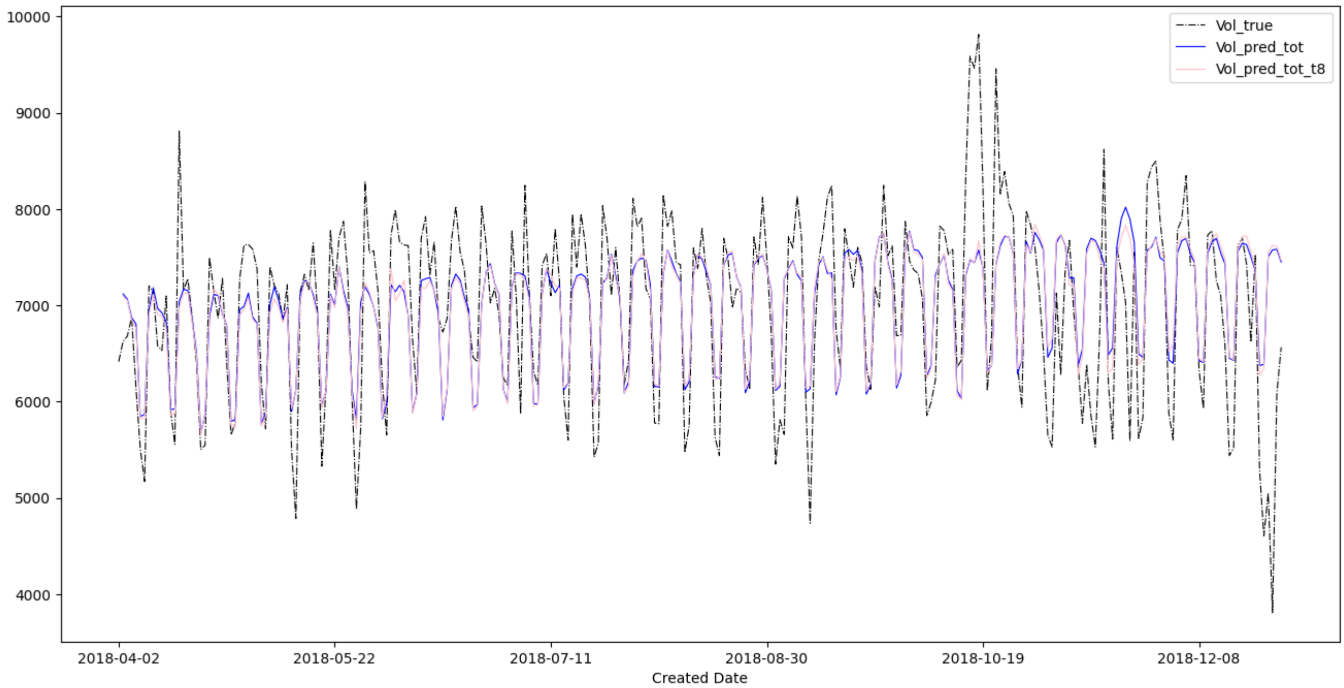
Both of our two multi-dimensional forecasting approaches outperform the single-dimensional target forecasting, with the borough breakdown slightly outperforming the cluster breakdown. This further validates the hypothesis that the data generating process of total volume is not single-dimensional, aligning with the findings from the baseline models.

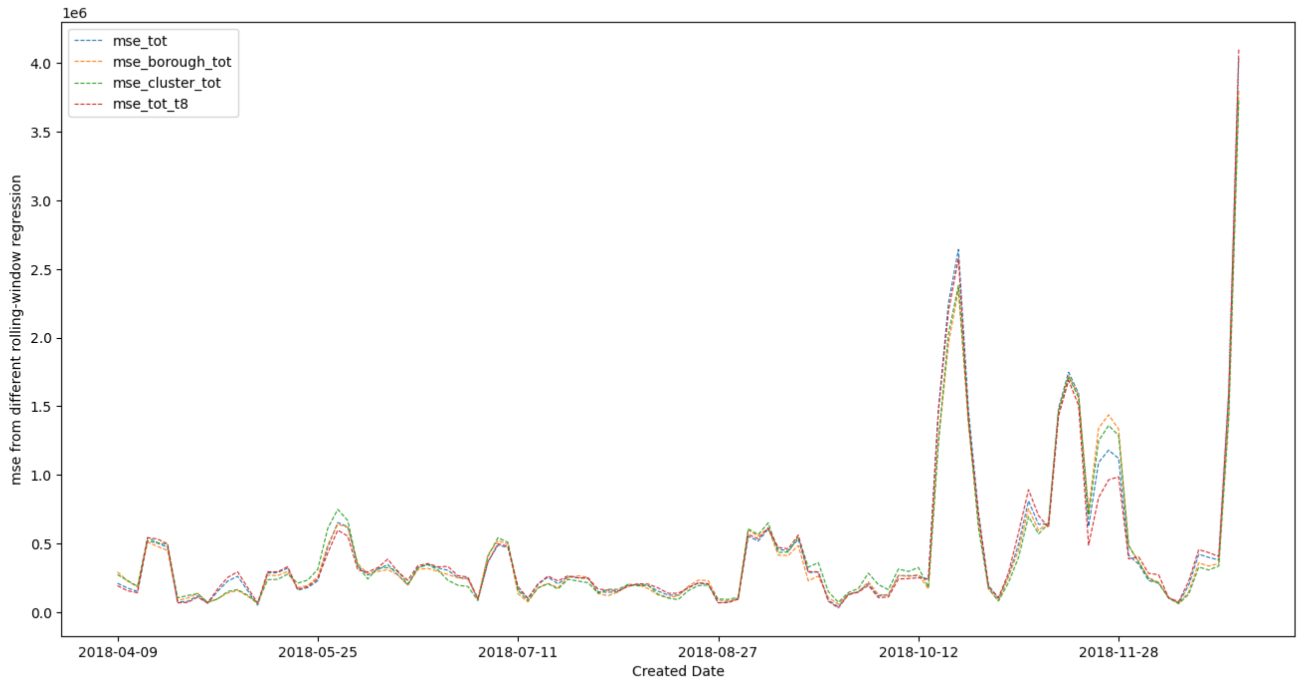
feature set: ['Duration', 'IsWeekend_Weekend', 'IsWinter_Winter', 'Down', 'DewPoint', 'Precipitation', 'wind_1', 'wind_2', 'SnowDepth', 'Rain', 'SnowIce', 'FOURIER_S7-0', 'FOURIER_C7-0', 'FOURIER_S365-0', 'FOURIER_C365-0'] + long-term trend + short-term residual

The results of linear models are reported as follows.

Forecasting method	volume total	volume total with time series component	volume total with borough breakdown	volume total with cluster breakdown
daily MSE	476658	474845	462760	466575





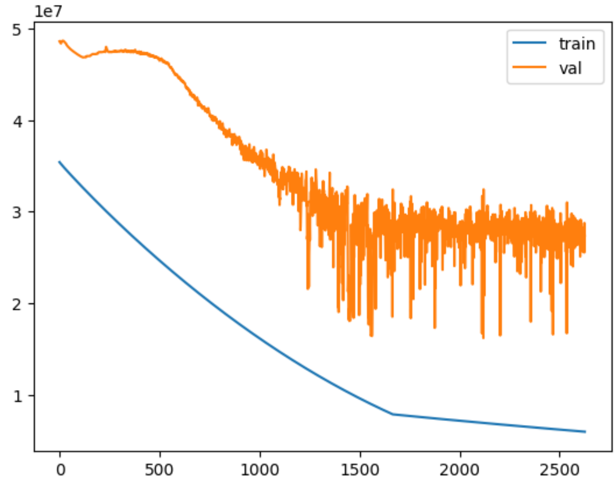


(e)

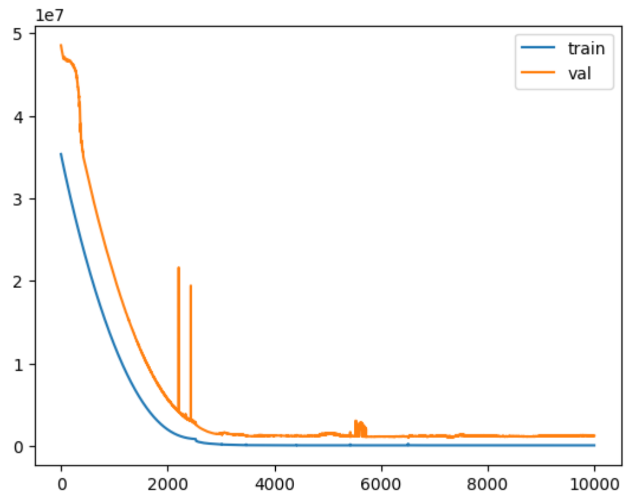
Using LSTM on V100 GPU, initially, we only include the long-term trend as an extra exogenous predictor since we assume it can pick up the short-term residual automatically. After calibrating the hyper-parameters of the network, we fixed the architecture and added the T-8/9/10 residual series into the LSTM input as an enhancement. We carried out an extensive hyper-parameters search and reported a few of them in this subsection. For all other network training details, please see the appendix. We use an early stopping heuristic for each varying parameter combination, so the number of epochs in each of the following plots varies.

Fixed parameter: *batch_size = 30, LSTM dropout = 0.5, Adam Optimizer betas=(0.9, 0.999), lookback = 360*.

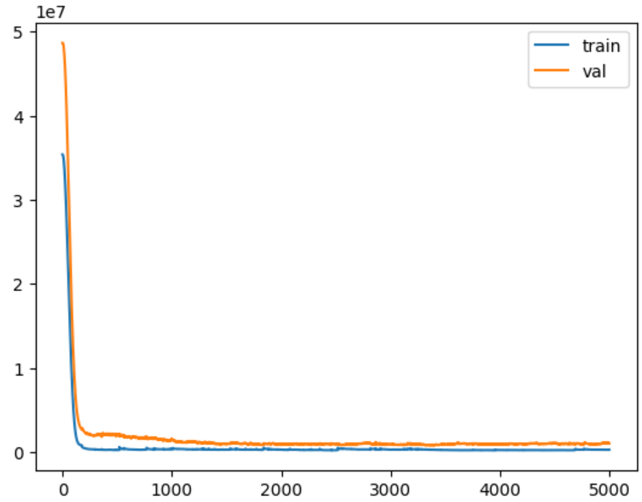
We started with *hidden_size = 50, num_layers = 2, lr = 0.005, weight_decay = 0.2*. The low learning rate and small regularization effect could not lead to a decent test set performance, so we adjusted these two parameters as well as the *hidden_size* as the next try.



When adjusting to *hidden_size = 32, num_layers = 2, lr = 0.01, weight_decay = 0.5*, the loss chart looks better, but still suggests our model is too "large". So we further tune the *hidden_size and num_layers*.



After reducing the model size, there is still a little gap between training loss and testing loss in the stable stage. We rethink the LSTM architecture to better capture the linear and nonlinear effects separately, to avoid mixing them too early in the process. After re-designing the network, we got the result as below. Specifically, we treat the short-term residual as nonlinear while the long-term trend as linear input, which is finally combined with the LSTM output using weather features to generate the volume prediction. The trained model file has been included in the submission package for reference.

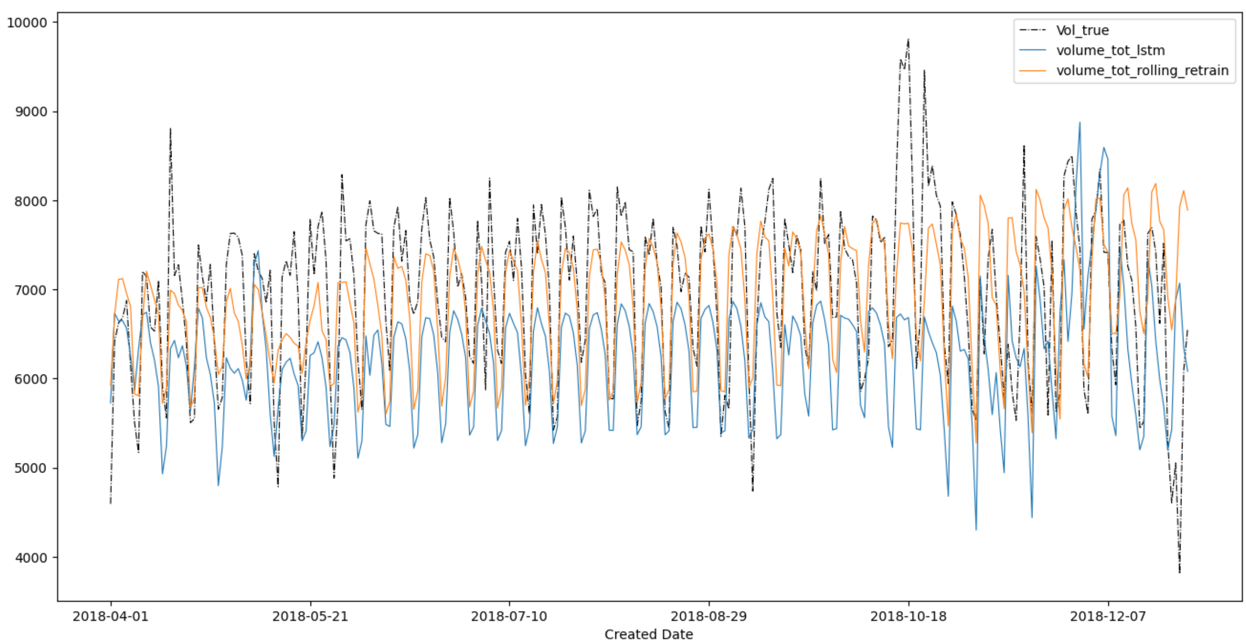


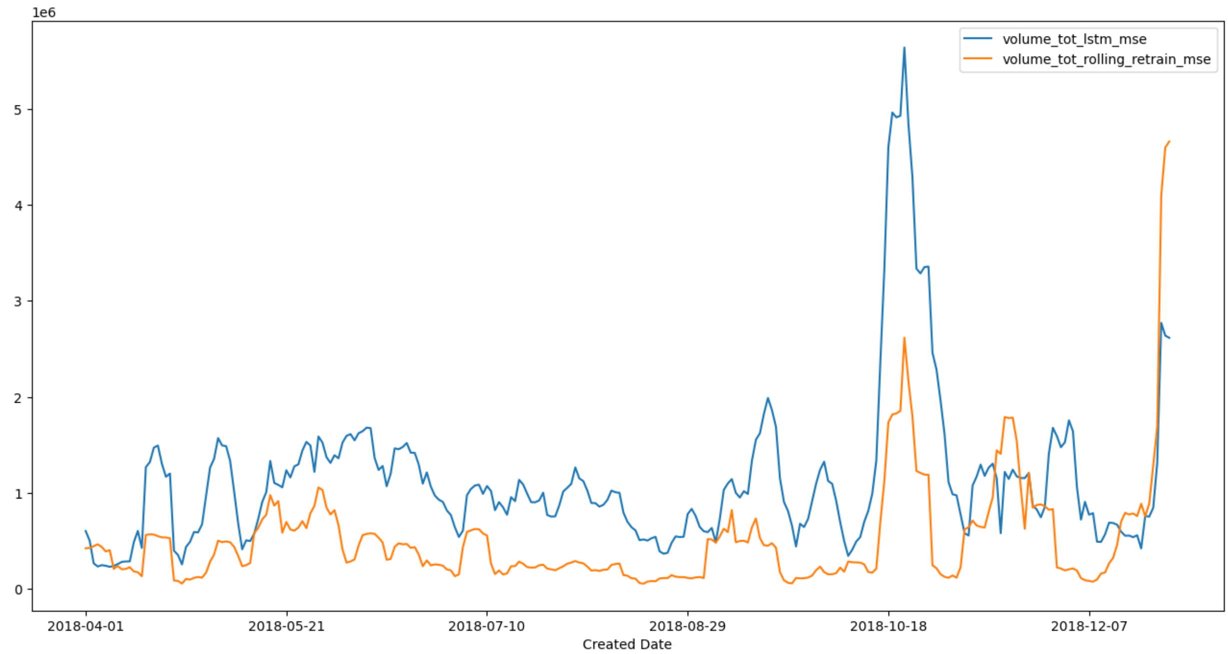

```
Epoch 3100: train MSE 317420.2812, test MSE 899817.2500
Epoch 3200: train MSE 361139.2812, test MSE 968935.1875
Epoch 3300: train MSE 320471.5000, test MSE 892837.5000
Epoch 3400: train MSE 282766.5312, test MSE 861252.5000
Epoch 3500: train MSE 256830.3281, test MSE 930555.1250
Epoch 3600: train MSE 241756.7500, test MSE 991217.6875
Epoch 3700: train MSE 234950.0156, test MSE 964106.0000
Epoch 3800: train MSE 248947.3906, test MSE 1004176.4375
Epoch 3900: train MSE 238783.4062, test MSE 902151.1250
Epoch 4000: train MSE 251044.7188, test MSE 959029.5625
Epoch 4100: train MSE 248914.5312, test MSE 1008493.7500
Epoch 4200: train MSE 246616.6562, test MSE 974870.8125
Epoch 4300: train MSE 241287.6406, test MSE 1015644.9375
Epoch 4400: train MSE 252380.9688, test MSE 1000422.8125
Epoch 4500: train MSE 245600.2969, test MSE 1066082.0000
Epoch 4600: train MSE 242570.9531, test MSE 1008281.9375
Epoch 4700: train MSE 323780.6875, test MSE 1024560.3750
Epoch 4800: train MSE 333738.1875, test MSE 1012966.1875
Epoch 4900: train MSE 300870.3750, test MSE 970894.4375
```

```python
dl_model = CSFB_Model()
dl_model.load_state_dict(torch.load(PATH))
dl_model.to(device)
```

```
CSFB_Model(
  (lstm): LSTM(15, 4, num_layers=2, batch_first=True, dropout=0.5)
  (linear_ts): Linear(in_features=3, out_features=1, bias=True)
  (linear_lstm): Linear(in_features=4, out_features=1, bias=True)
  (linear_out): Linear(in_features=3, out_features=1, bias=True)
  (bn): BatchNorm1d(1, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (drop): Dropout(p=0.1, inplace=False)
  (ts_relu): LeakyReLU(negative_slope=0.1)
)
```

The final prediction result is as follows, where the *volume_tot_lstm* is out-of-sample testing directly using our fixed trained model from the training set, the *volume_tot_rolling_retrain* is a further fine-tuned version of the trained model on a rolling weekly basis.





Model Setting	hidden_size = 4, num_layer = 2, small ts regularization, linear residual	hidden_size = 4, num_layer = 2, small ts regularization, LeakyReLu residual	hidden_size = 4, num_layer = 2, normal ts regularization, ReLu residual
volume_tot_lstm_mse	1.52e+06	9.83e+05	1.14e+06
volume_tot_rolling_retrain_mse	7.03e+05	7.34e+05	5.18e+05

The rolling fine-tuned outperforms the fixed network for all the architectures. The first column shows linear residual is not representative enough so we have a large loss from the fixed network. The second column replaces the linear residual with a LeakRelu layer, while keeping the same account of regularization. That is where we saw the smallest gap between fixed network and weekly fine-tuning. The third column adds more nonlinearity to capture more transient signals in the weekly fine-tuning version, at the cost of a moderate loss of fixed network.