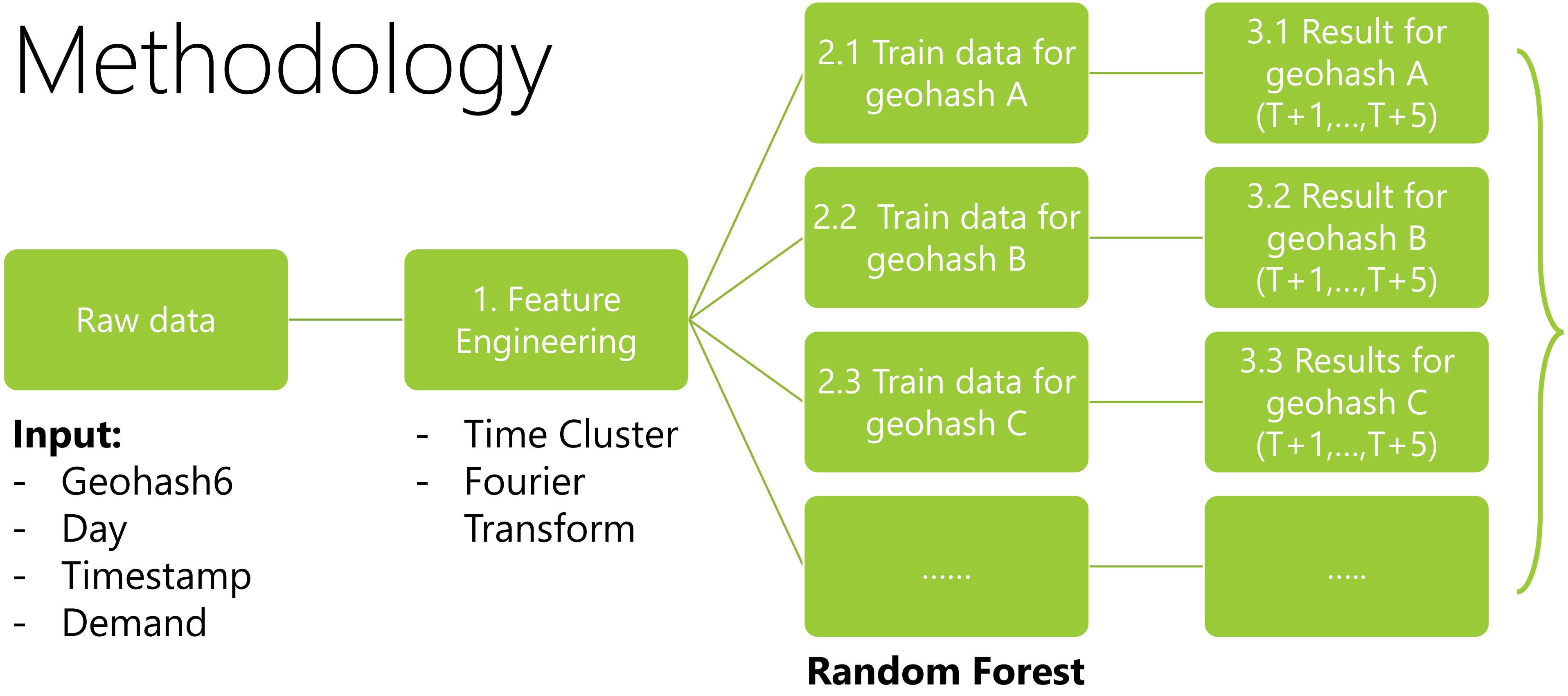
AI for S.E.A Challenge: Traffic Management

Nurvirta Monarizqa

nmonarizqa@windowslive.com | nmonarizqa.github.io

Methodology



Raw data

Input:

- Geohash6
- Day
- Timestamp
- Demand

1. Feature Engineering

- Time Cluster
- Fourier Transform

2.1 Train data for geohash A

2.2 Train data for geohash B

2.3 Train data for geohash C

……

Random Forest

3.1 Result for geohash A (T+1,…,T+5)

3.2 Result for geohash B (T+1,…,T+5)

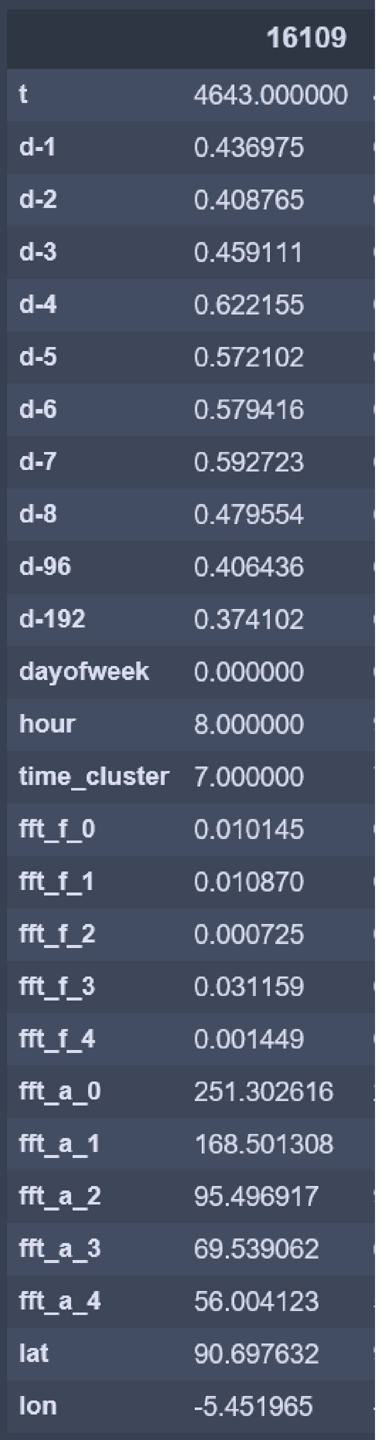
3.3 Results for geohash C (T+1,…,T+5)

…..

Result:

- Geohash6
- Day
- Timestamp
- Demand

To predict the demand, each data point was first modified into usable training data through feature engineering. For each geohash6, a random forest model was used to train the data, then applied into data for T+1,…,T+5 to predict the future demand. Therefore, there will be *n* number of different random forest used to predict the demand with *n* is the number of unique geohash6.



	16109
t	4643.000000
d-1	0.436975
d-2	0.408765
d-3	0.459111
d-4	0.622155
d-5	0.572102
d-6	0.579416
d-7	0.592723
d-8	0.479554
d-96	0.406436
d-192	0.374102
dayofweek	0.000000
hour	8.000000
time_cluster	7.000000
fft_f_0	0.010145
fft_f_1	0.010870
fft_f_2	0.000725
fft_f_3	0.031159
fft_f_4	0.001449
fft_a_0	251.302616
fft_a_1	168.501308
fft_a_2	95.496917
fft_a_3	69.539062
fft_a_4	56.004123
lat	90.697632
lon	-5.451965

1. Feature Engineering

There are 26 features used in this prediction.

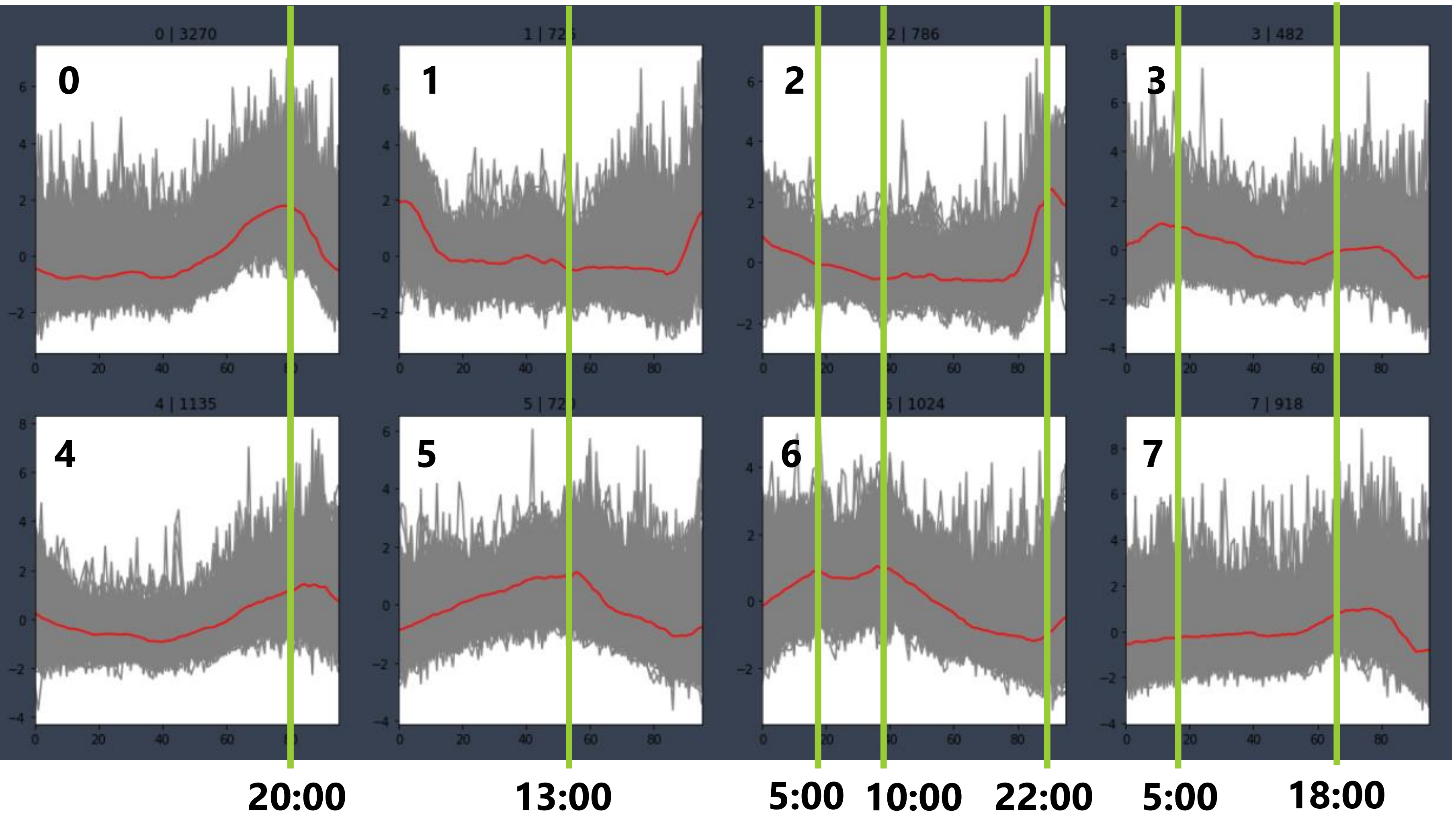
In time series prediction, seasonality matters. In the case of traffic, the most common seasonality pattern is daily, then weekly, and season (summer, fall, winter, spring). Since it is South East Asia, season was assumed to be not present. Therefore, ***dayofweek*** was created to capture weekly seasonality, ***hour*** was created to capture daily seasonality, and ***t*** was created to differentiate each period.

Fourier transform was also included to capture seasonality pattern for each location (geohash6). Time series was assumed to be the sum of many sinusoidal function, therefore, the pattern can be “captured” using fast-fourier transform. For each location, the top 5 frequencies (which are when the seasonality is most apparent) was extracted (***fft_f_0 to fft_f_4***) as well as the amplitudes (***fft_a_0 to fft_a_4***).

Previous demands were believed to be important factors. Therefore, the previous value (***d-1***), second to the last *(**d-2**)*, up to ***d-8*** were also included. To capture daily and weekly seasonality, ***d-96, d-192, d-672*** were also included.

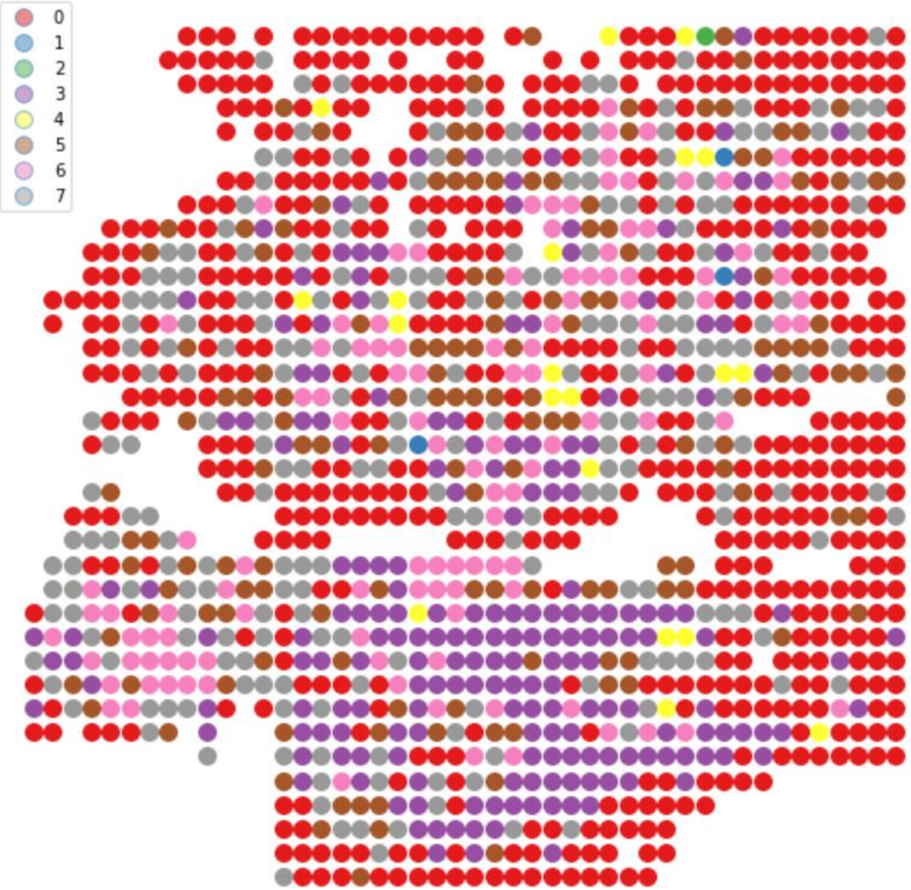
Location also matters. To capture the importance of locations, the geohash6 was geocoded into its latitude (***lat***) and longitude (***lon***). Another variable that was used is *time_cluster* which will be explained in the next page.

1. Feature Engineering: Clusters

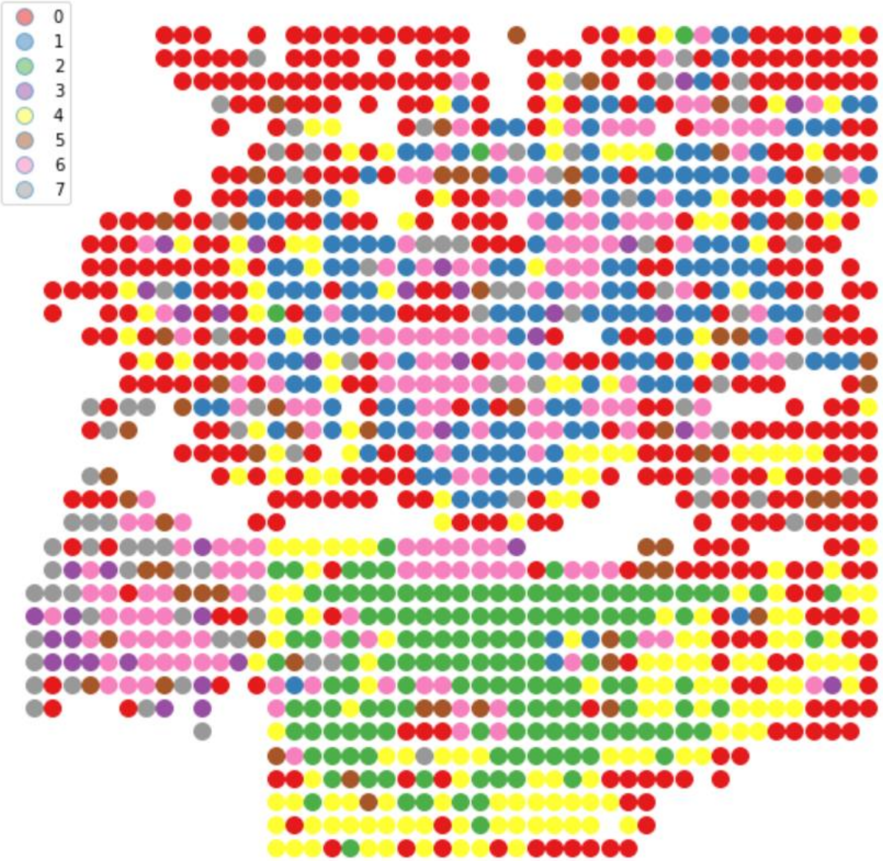


For each geohash6 and each day, the timeseries was normalized and grouped into one of these eight clusters. Since the landuse data was unknown, these daily timeseries clusters were able to capture the daily pattern and possibly identify the type of the location. Cluster 3 for example, has two peaks during morning and evening, so it is possible that cluster 3 indicates CBD (central business district) areas. The cluster then stored in to ***time_cluster*** variable

Clustering



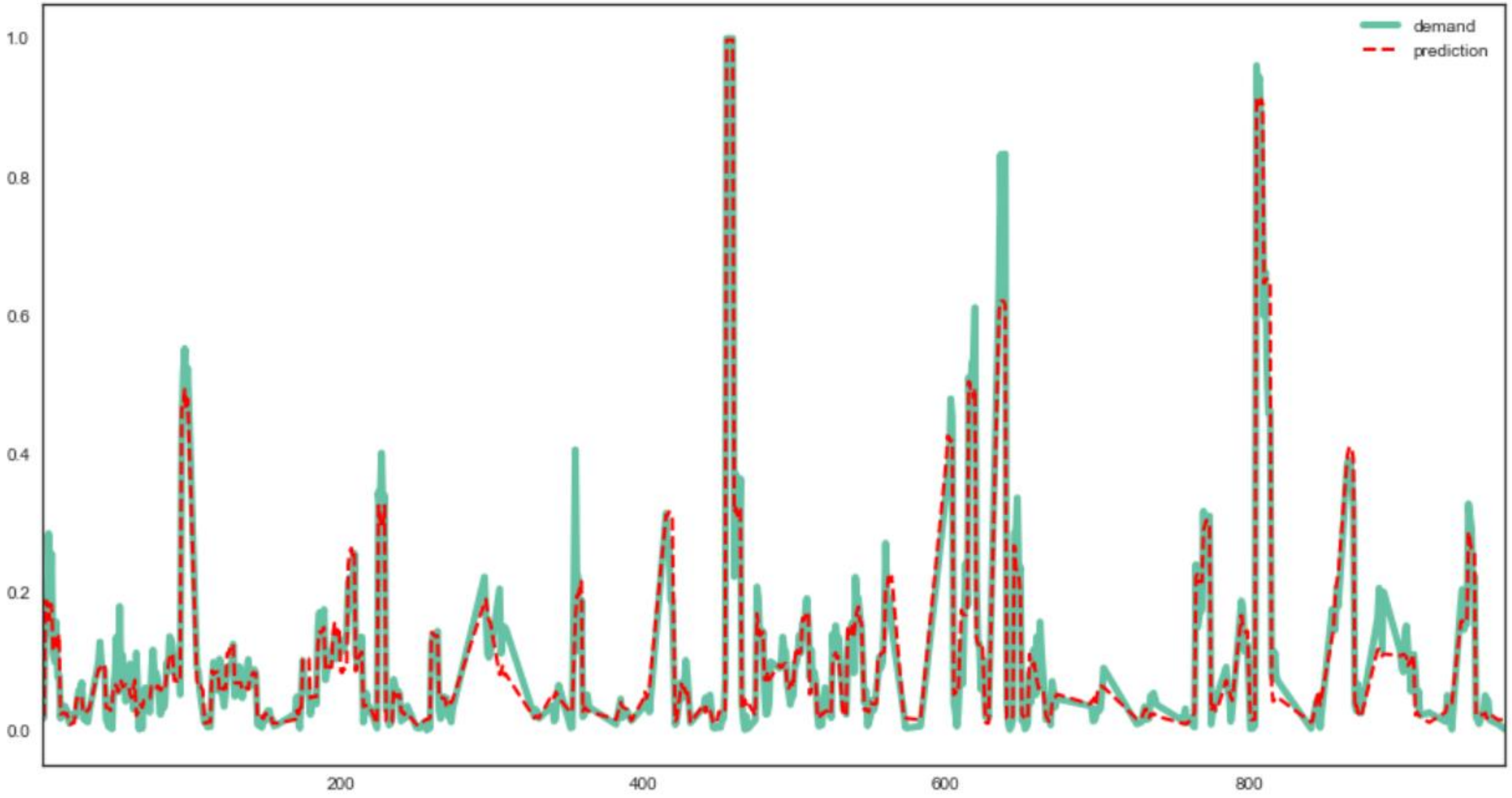
Spatial Pattern of *dayofweek==5*



Spatial Pattern of *dayofweek==0*

Here is an example of how time_cluster can help us identify the type of the location. During a specific day, cluster 3 (purple) that has peak at 5:00 and 18:00 dominating the southern areas, while on different day, those areas are dominated by cluster 2 (green) that has peak at 22:00-23:00.

2&3. Training and Predicting Data



Demand vs prediction of a set of sample data

As each location (geohash6) has specific pattern, *n* number of random forest was trained to fit *n* set of timeseries which *n* equals number of unique geohash6 in the data.

Random forest regression was used and fine-tuned so that it returns good accuracy while not overfit and maintain a reasonable computation time.

Then, for each geohash6, next five period of demand were calculated.

Resources

- Script: https://github.com/nmonarizqa/grab-tm
- Prerequisites: geohash, scikit-learn
- Python: 2.7
- Usage: `python predict_demand.py [filename_input]`
- Example: `python predict_demand.py sample.csv`