2018 JD Competition Population dynamics census and prediction

Team Name: Cheese&Chips

Division: Israel

Teammates: Haibin Yu (PhD student, haibin@u.nus.edu)
Zhongxiang Dai (PhD student, daiz9109@gmail.com)
Yizhou Chen (PhD student, leon798775190@gmail.com)

University: National University of Singapore, School of Computing,
13 Computing Drive, Singapore, 117417

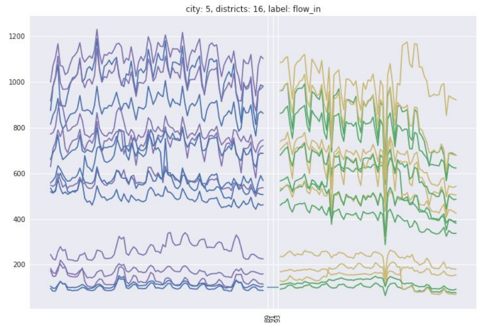
Data Visualization and Exploration

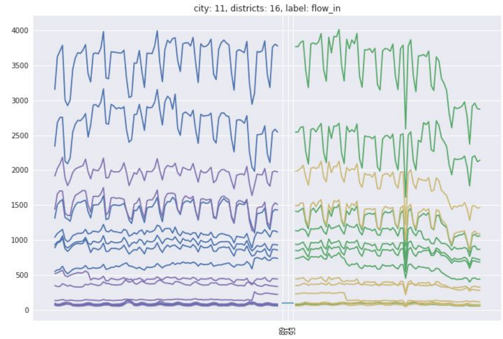
Before delving into developing time series prediction models, it is necessary for us to check how the data look like. Therefore, we read the file “flow_train.csv” to check the various statistics,

	dwell	flow_in	flow_out
count	32844	32844	32844
mean	4.0911	199.6082	199.6082
std	8.3844	393.3869	393.3573
min	0.1080	5.5960	5.5320
50%	1.5880	74.5240	74.5480
max	95.2640	4010.2840	4015.9280

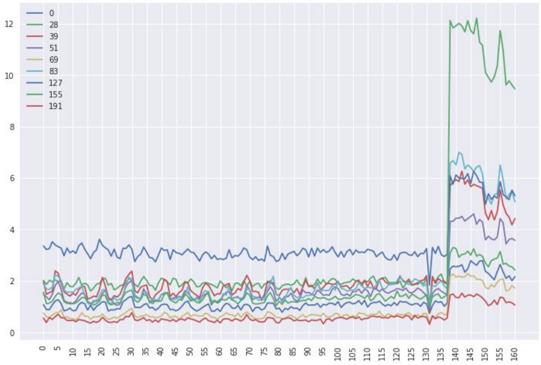
As we can discover from the table above, the statistics of flow in and flow out are almost the same. These two labels are nearly identical. Therefore, the prediction for flow in and flow out should also be the same. Such macro fact is the result of the property of transitions where the flows between two districts (A to B or B to A) are always similar.

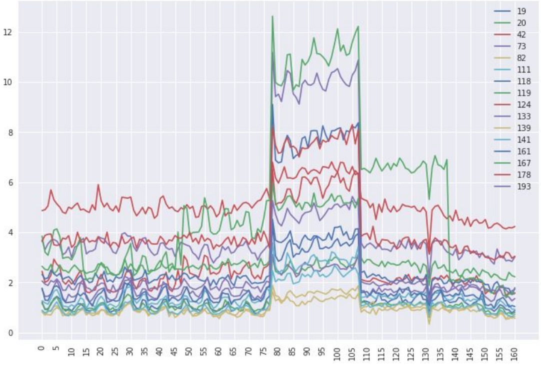
The dataset contains 13 cities and 204 districts. We would like to discover the correlation between each city or each district. Here is an example of label flow in for different districts for different cities:





The plot gives us a hint that there are a lot of similarities among different districts since they look very similar to others (seasonality and trend). Then we decide to check the pearson correlation between districts. The pearson correlation coefficient value is set to be 0.85 (with high linear correlation). By doing so, we find very interesting patterns among the dataset as shown in the following figures:





We find out that 60+ districts have this clear pattern we refer as “Squared Wave”. The data values located in the “Squared Wave” are very different from the values outside. The prediction model we developed targets to extract this pattern and makes our predictive results more accurate and robust.

Another very interesting property is that we find out the starting(ending) timestamp of the “Squared Wave” are always [14, 48, 78, 108, 138] days after the first day in record (without counting missing dates). The intervals allow us to predict the rough location of the next timestamp.

Pre-processing

Before applying the prediction models *for the last 10 days*, we applied a pre-processing step to the time series by removing the square waves discovered during data exploration (as displayed in the figures in the previous section).

Specifically, we **(1)** identified those time series with the appearance of square waves, **(2)** marked the beginning and ending of these square waves, **(3)** removed those marked sections, and **(4)** concatenated the remaining sections as the new pre-processed time series. We completely discarded the sections corresponding to the square waves because we observed that, for each time series, the patterns of the square waves differ largely from those of the other sections. Note that for those square waves that started before the prediction period yet did not end before that, we still completely removed the square wave since we observed that doing so led to significant performance improvement, which might be caused by immediate drop of these time series in the prediction period (explained later). Also note that we did not use this pre-processing technique when predicting the missing 5 days in the middle, since some of the square waves span across these 5 days.

Prediction Models

Linear Regression Autoregressive Model

Our final results make use of linear regression model. The models treat dwell, flow-in and flow-out separately.

In the first stage, we predict the missing 5 days by a linear regression model LR1. LR1 takes input from the starting H days before the missing 5 days and F days after the missing 5 days and output the prediction of the missing 5 days. H and F are hyperparameter obtained by grid search on the local validation set and the value is H=7 and F=8 . We also grid search for the time for the start point S and end point E of the training and the value is S=0 and E=14. After obtaining those hyperparameters, we retrain the model and make prediction of the missing 5 days. For LR1, we did not do pre-processing of the data.

We predict the raw value of the last 10 days using another linear regression model LR2. For LR2, pre-processing is applied as described above. Additionally, the time series take in the prediction of LR1 as the ground truth for the missing 5 days. We again do a grid search on H and S mentioned earlier and in this model H=3 and S=12.

We notice that square waves might started before the prediction period of the last 10 days yet did not end before that. We suspect based on the periodic nature of the square wave that a sudden drop might occur within the prediction period. This gives rise to another hyper parameter to tune, which is the date that the abruption happens. After we validate our results in the leaderboard we found that the drop is most likely to occur right at the beginning of the last 10 days. Thus these square waves does not need further special treatment.

In this stage, we found that if we did not do the pre-processing, the leaderboard error is around 0.2320. With pre-processing, the error can be reduced to around 0.1947. Another linear regression model LR3 is used for the refinement of prediction, which is introduced in the post-processing section.

ARIMA

We also attempted the Autoregressive Integrated Moving Average (ARIMA) model, which is a commonly used model in statistics for time series prediction. In particular, we used the *auto_arima* function from the pmdarima Python package, which, in addition to running the standard ARIMA model, also performs a search over the hyper-parameters of the ARIMA model such as the number of autoregressive steps, the number of moving average steps, etc. In particular, for each time series, we ran two types of ARIMA models for predicting the middle 5 days and the last 10 days respectively.

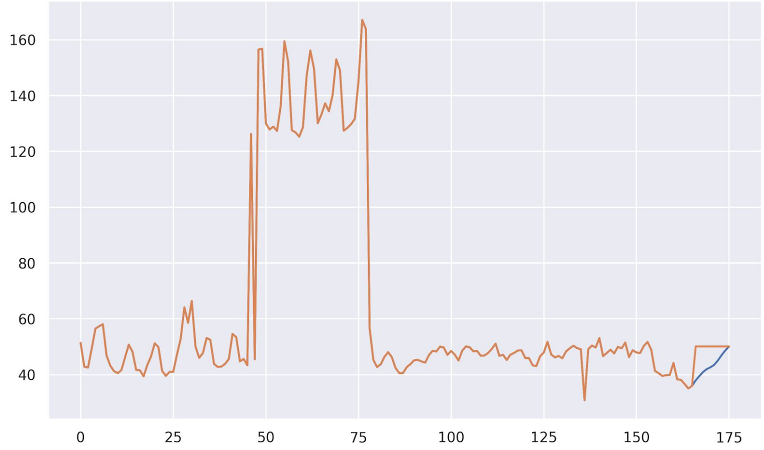
Specifically, since the missing 5 days are in the middle of the time series, we only used the data from the first 88 days (right before the missing data) to fit the ARIMA, and predicted the values of the next 5 days. On the other hand, regarding the last 10 days, we simply ignored these 5 missing days in the middle, and used the entire time series to fit the ARIMA model and predicted the values of the last 10 days. We also tried inserting the predicted values of the 5 missing days (from the previously described arima model) in the middle and re-fitting the ARIMA model, however, the validation performance dropped dramatically, so we abandoned this approach.

Unfortunately, early into the second phase of the competition, we observed that the ARIMA model did not perform competitively with the linear regression model described earlier, and, to our surprise, ensembling (averaging) with the results from linear regression deteriorated the performance, so we did not use ARIMA models in the following predictions.

Post-processing

We also applied some post-processing steps to improve our predictions of the last 10 days. As a result of our observations of **(1)** the presence of square waves in many districts and **(2)** the fact that removing the square waves that are close to the prediction period significantly improved the performance **(3)** the timestamp near the prediction period is likely to occur right at the beginning of the last 10 days, we suspected that for some of the time series, there might exist the beginning of square waves (abrupt rise) during the prediction period. Therefore, we tried to identify the time series in which this might happen, and applied post-processing to test the validity of this conjecture.

In particular, we ran the linear regression model LR3 (the blue curve in the figure below) to predict the last 10 days as described earlier *without removing the square waves*; then, we identified those districts whose predictions exhibited a clear (monotonic) increasing trend, and treated those time series as candidates in which abrupt rise might happen. Subsequently, for each time series, we recorded the maximum predicted value among the 10 predictions, and fixed every prediction of the last 10 days to this value, thus creating an artificial abrupt increase in the prediction period. As a result, LR3 served as the predictor for the sudden rise. The post-processing procedure can be illustrated by the figure below, in which the blue curve on the right end represents the original prediction using LR3, and the flat prediction above the blue curve is the predicted values submitted after post-processing.



We found that doing this further improved the prediction performance, which we believe corroborates our conjecture that the abrupt rise happens in at least some of the regions we identified. Post-processing brought down the leaderboard error to around 0.1887.

Conclusion

Throughout this competition, we learned the importance of data visualization/exploration and drawing insights from the preliminary data analysis, because some inherent patterns of the data can turn out to be critical for reliable prediction performance.