Data Mining Project:

Short Term Road Traffic State Prediction with GPS Data

UV F3B213A

Jiao GUO
Xuanqi HUANG
Xiaorui HUO

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Content

1. Introduction

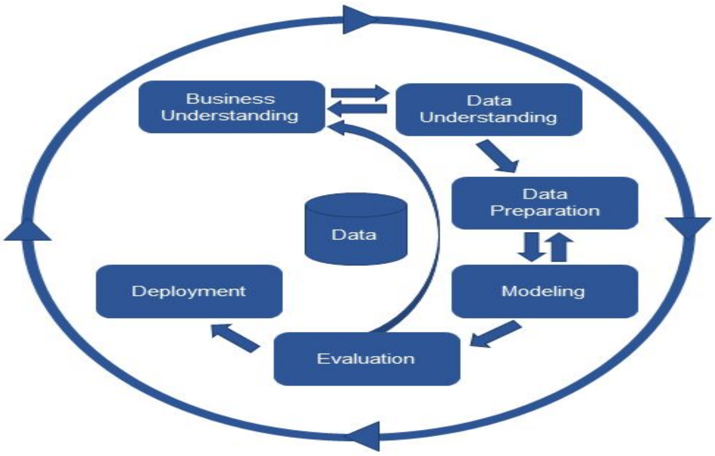
Didi company is a chinese Uber-like company, each vehicle receives lots of orders each day from clients who request the rides from departure to destination. With the data collecting system, DiDi company possesses a database of client orders, in which the details of each order are recorded. Including the vehicle information, GPS location records every few seconds, as well as time and location details about departure and arrival.

In Chengdu, a city in China, like the other medium-size cities, the traffic can be considerably bad, particularly at citie's centre in the daily rush hours. The traffic congestion usually influence significantly trip durations. The long trip duration, on one hand, potentially has a negative impact on the client satisfaction, on another hand, it makes a direct effect on company's revenue. Hence, the prediction of congestion in the near upcoming time is quite important. With the congestion information, Didi company can inform the car driver of the congestion condition in the near future, according to which, the driver can adjust its driving route and potentially reduce the trip time.

To measure and modelize the congestion, the mean spatial speed (abv: MSSpeed) is been proposed by our tutor, which estimates the congestion condition at a specific zone in a specific time slot. Due to the difficulty of calculation, the goal in this period is to construct models and predict the mean spatial speed.

2. Methodology

Cross-industry standard process for data mining, known as CRISP-DM, is an open standard process model that describes common approaches used by data mining experts.[1] Therefore, in order to carry out our project better, our project is based on methodology CRISP-DM.



- Business understanding: The purpose of our project is to estimate the congestion condition at a specific zone in a specific time slot by predicting the mean spatial speed, which will help improve the traffic congestion in the city.
- Data understanding: We visualized the data, checked for missing values and outliers and explored the data.
- Data preparation: We first calculated the mean spatial speed, and then selected the features used for modeling.
- Modeling: For modeling, we chose Xgboost and LSTM model.
- Evaluation: The way of evaluation is mean square error.
- Deployment: We can add more feature or implement new models. Besides, we can use Big Data methods (Spark / Hadoop) for parallelization and speed of execution.

3. Data Understanding

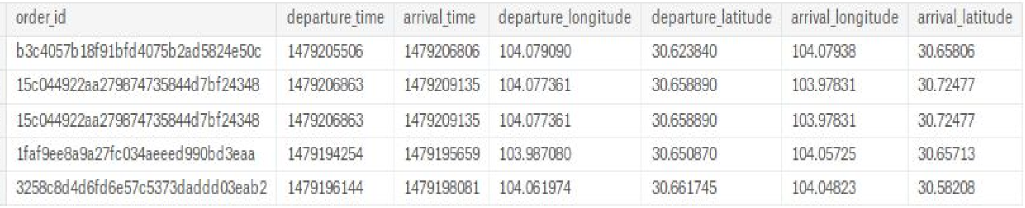
This section gives a global view of data, including the descriptive presentation which leads to a better comprehension for analysis of errors, anomalies, observations and statistical analysis.

There are two types of daily data, with a time range from 2016/11/05 to 2016/11/30.

- order_yyyymmdd :
  - Format: 'com.databricks.spark.csv'
  - Size: about 20MB
  - Columns: order_id, depature_time, arrival_time, departure_longitude, departure_latitude, arrival_longitude, arrival_latitude
  - Explication: Details about each order.
- gps_yyyymmdd :
  - Format : 'com.databricks.spark.csv'
  - Size : about 3.3GB
  - Columns: vehicle_id, order_id, universal_time, longitude, latitude
  - Explication: The GPS track for each order every few seconds.

3.1 Table Order

This table is mainly to record the information of orders, including the location and time of the start and end of the order , and order's id. An example of data as shown in table 3.1.



order_id	departure_time	arrival_time	departure_longitude	departure_latitude	arrival_longitude	arrival_latitude
b3c4057b18f91bfd4075b2ad5824e50c	1479205506	1479206806	104.079090	30.623840	104.07938	30.65806
15c044922aa279874735844d7bf24348	1479206863	1479209135	104.077361	30.658890	103.97831	30.72477
15c044922aa279874735844d7bf24348	1479206863	1479209135	104.077361	30.658890	103.97831	30.72477
1faf9ee8a9a27fc034aeeed990bd3eaa	1479194254	1479195659	103.987080	30.650870	104.05725	30.65713
3258c8d4d6fd6e57c5373daddd03eab2	1479196144	1479198081	104.061974	30.661745	104.04823	30.58208

Table 3.1 : Order data example

3.1.1 Data exploration

Data exploration in this sector is based on file order_20161115. In order to better understand this order’s table, we will explore the number of orders per hour, the time interval of each order, and the location range of orders.

3.1.1.1 Order visualization

As shown in the figure 3.2, from 7h to 19h, there are a lot of orders which implies that there are a lot of people using DIDI services. However, from 0h to 6h, the number of orders is relatively small.

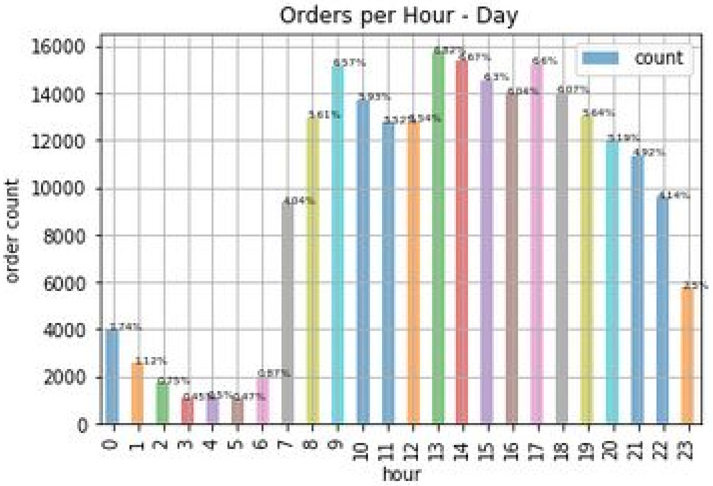
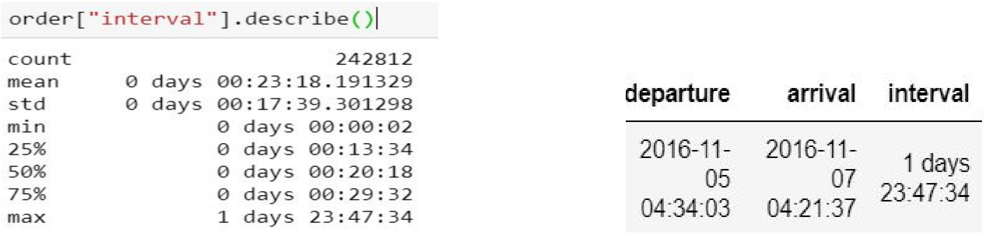


Figure 3.2 : Order counts per hour

3.1.1.2 Time interval of order

We calculated the interval between departure time and arrival time for each order and found an outlier with its time interval more than one day. This table is shown below :



```
order["interval"].describe()

count                      242812
mean       0 days 00:23:18.191329
std        0 days 00:17:39.301298
min               0 days 00:00:02
25%               0 days 00:13:34
50%               0 days 00:20:18
75%               0 days 00:29:32
max               1 days 23:47:34
```

departure	arrival	interval
2016-11-05 04:34:03	2016-11-07 04:21:37	1 days 23:47:34

Figure 3.3 : Time interval of orders

3.1.1.3 Order location range

We can observe from the following figures that the arrival position of the order has some outliers, for example, there are some points whose longitude and latitude are 0.

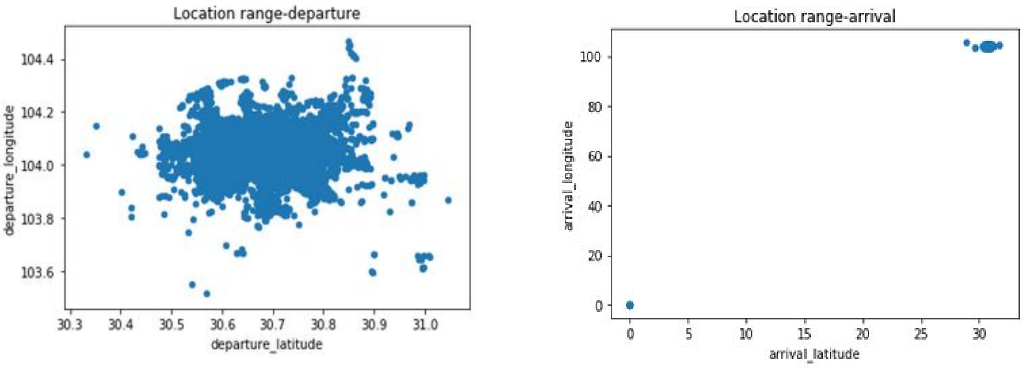


Figure 3.4 : Scatter plot of latitude and longitude--Order

After removing the outliers, we plot the location range of the order on the map of Chengdu. The red contour represents the location range of the order table, and the blue contour represents the location range of the gps table. We can find that the location range of the gps table is only a small part of the location range of the order table. So for the next steps, we only consider using the gps table.

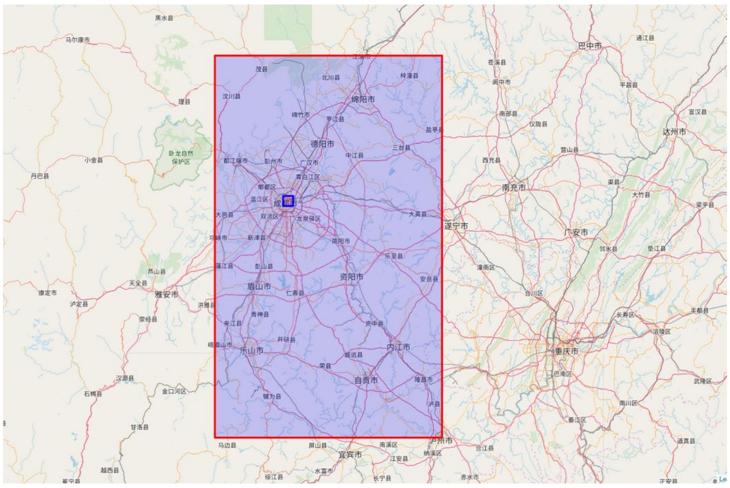
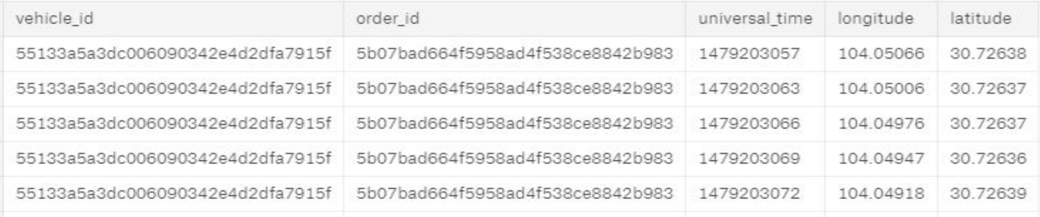


Figure 3.5 : location range displayed on map(order table and gps table)

3.2 Table GPS

The GPS table contains GPS location track information, which means the vehicle geographic locations of each trip, represented by latitude and longitude. They are recorded every few seconds. In order to have a better understanding of gps tabel, we firstly choose and analyse one daily file, and then combine daily files for global analysis.



vehicle_id	order_id	universal_time	longitude	latitude
55133a5a3dc006090342e4d2dfa7915f	5b07bad664f5958ad4f538ce8842b983	1479203057	104.05066	30.72638
55133a5a3dc006090342e4d2dfa7915f	5b07bad664f5958ad4f538ce8842b983	1479203063	104.05006	30.72637
55133a5a3dc006090342e4d2dfa7915f	5b07bad664f5958ad4f538ce8842b983	1479203066	104.04976	30.72637
55133a5a3dc006090342e4d2dfa7915f	5b07bad664f5958ad4f538ce8842b983	1479203069	104.04947	30.72636
55133a5a3dc006090342e4d2dfa7915f	5b07bad664f5958ad4f538ce8842b983	1479203072	104.04918	30.72639

Table 3.6 : GPS data example

3.2.1 Data exploration

Data exploration in this section is based on file gps_20161115. Generally speaking, data in GPS table has high quality, quite coherent, without missing value.

3.2.1.1 GPS location range

Following Figure 3.7 shows that the GPS data distributes approximately in a rectangular. Geographic locations in GPS table are only part of trips data, taking out from some complete trip records.

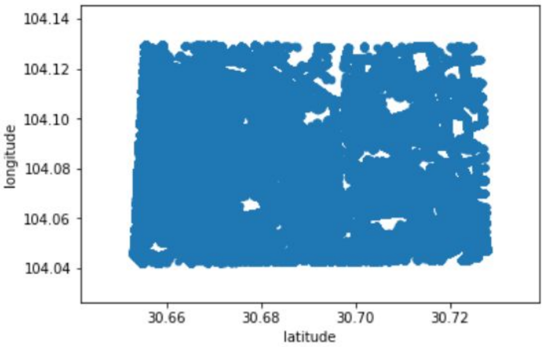


Figure 3.7 : Scatter plot of latitude and longitude-GPS

Taking out the four border points, which are maximum and minimum of longitude and latitude, Figure 3.8 shows the visualization of this GPS zone in city map. GPS range contains downtown area, suburb area, and the area between downtown and suburbs. GPS data distributes at east-north corner of Chengdu city.

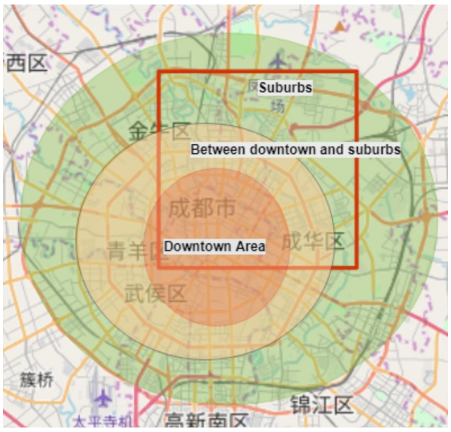


Figure 3.8 : GPS signal range displayed on Chengdu map

3.2.1.2 GPS signal visualization

The figure shows GPS signals number for each hour of 2016-11-15. From the plot, it can be seen that from hours 1 to 7, there are a few GPS signals which implies the traffic is not busy.

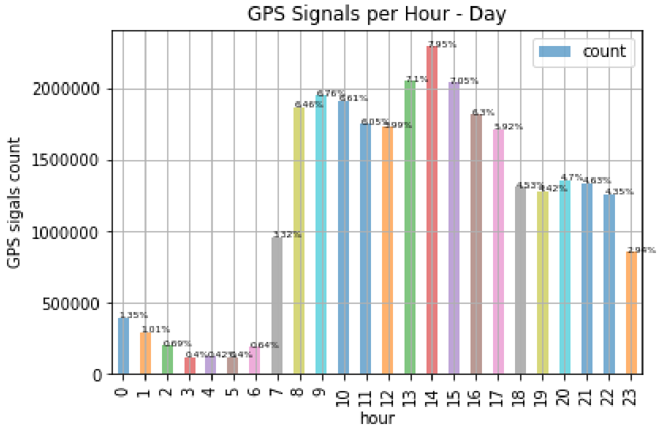
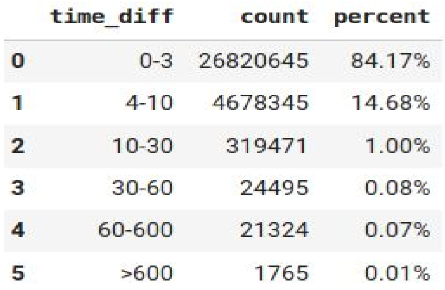


Figure 3.9 : GPS signal number per hour

3.2.1.3 GPS signal time interval

GPS signal of an order is tracked every few seconds, following table shows the basic distribution of tracking time interval, which is the time interval between two neighbouring GPS signals. More than 98% of tracking time interval are within 10 seconds, which means GPS data is quite time intensive.



	time_diff	count	percent
0	0-3	26820645	84.17%
1	4-10	4678345	14.68%
2	10-30	319471	1.00%
3	30-60	24495	0.08%
4	60-600	21324	0.07%
5	>600	1765	0.01%

Table 3.10 : GPS signal tracking time interval

3.2.1.4 GPS signal counts per order

There are 208,406 orders in daily GPS file of 20161115, the mean GPS count per order is 158. As show in following boxplot, the distribution of GPS count per order is skewed and there are outliers.

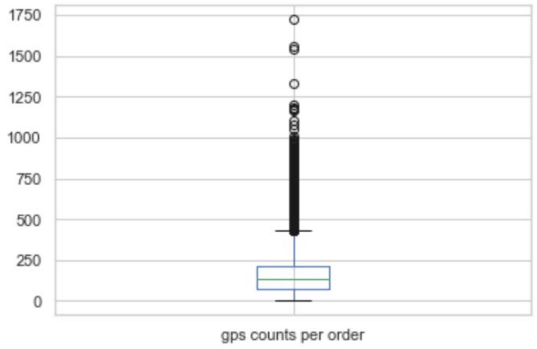
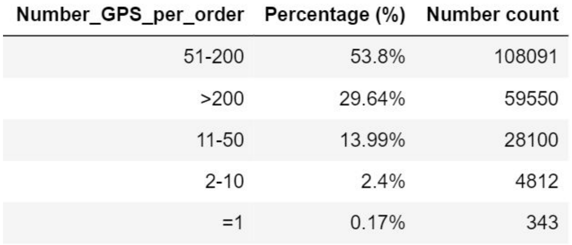


Table 3.11: Boxplot of GPS number per order with mean = 158

The table 3.12 showing that there are about 0.17% of orders contain only one GPS point.



Number_GPS_per_order	Percentage (%)	Number count
51-200	53.8%	108091
>200	29.64%	59550
11-50	13.99%	28100
2-10	2.4%	4812
=1	0.17%	343

Table 3.12 : GPS number per order

3.2.1.5 Time duration of orders in GPS zone

The following figure show the distribution of time duration of orders within GPS range. Mean value of travel time is 9.7 min.

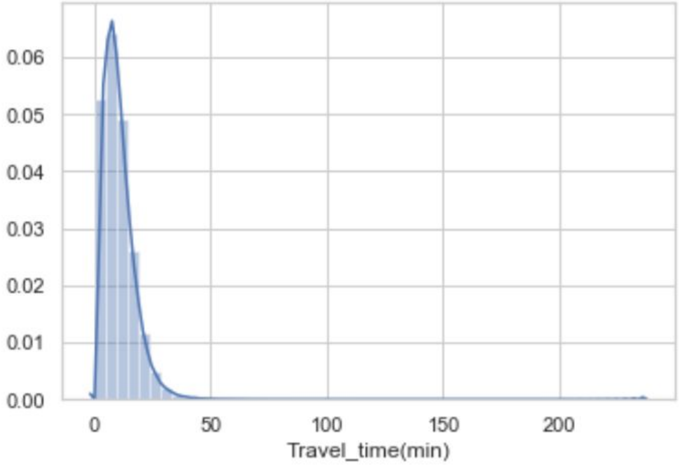
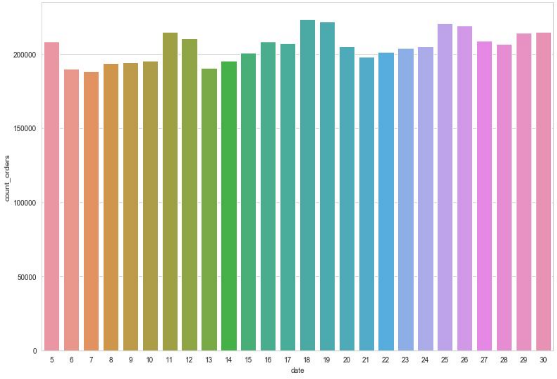


Figure 3.13 : Distribution of time duration

3.2.2 Cross file analysis

This section aims to comparer information from different day in a week.

The following figures show the orders numbers and vehicle numbers in each GPS daily file from Saturday 2016-11-05 to Wednesday 2016-11-30. The distribution in two figures are approximately the same, as well as the weekly form occurs respectively. Friday and Saturday is the most busy days in a week.



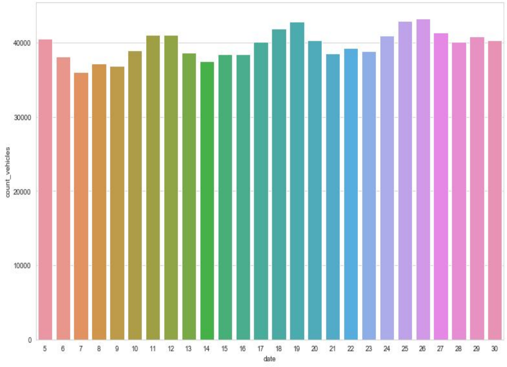


Figure 3.14 Number orders per day

Figure 3.15 Number vehicles per day

3.2.3 GPS location visualization problem and solution

Chinese government requires all local maps services to use an obfuscated, deviation-originated coordinate system[1] for data that we have, where the latitude and longitude respect the GCJ-02 code. That’s the reason why the graph has a big deviation when we use folium map to visualize geolocation trip GPS, shown in the following figure at left. In order to correctly visualize data in Map API, we propose two solutions:

Solution1: Folium is a simple used map tool, displaying geography data in a clair and interactive way. In order to correctly display GPS signals, firstly we are supposed to convert latitudes and longitudes of coordinates from system GCJ-02 to WGS-84 which is an no offset coordinate system. The following figure shows that converted data display correctly with Folium.

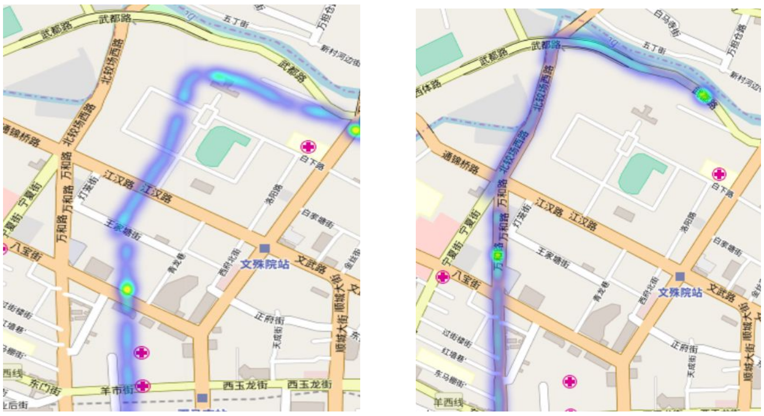


Figure 3.16 Comparison of two coordinate system

Solution 2: GaoDe Map API can be used in order to correctly display the GPS point, whose coordinates system is GCJ-02. The graph shows the visualization of a part of trip in GaoDe map API, which is precisely displayed.

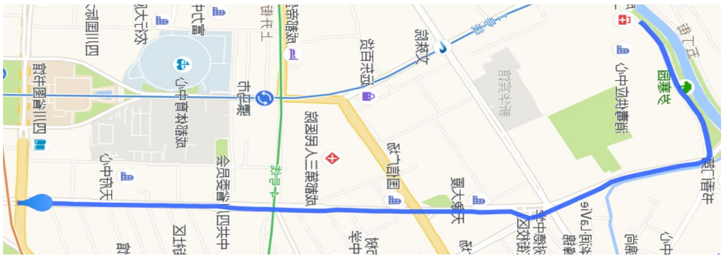


Figure 3.17 Visualization in Gaode Map

4. Data preparation

This section contains feature creations. Before modeling, the first part is to calculate mean spatial speed as the label for our machine learning model. The second part is about feature engineering.

4.1 Mean spatial speed calculation

4.1.1 Definition of mean spatial speed

Mean spatial speed is defined by the following formula:

$$GPS\,Speed = \frac{\sum Total\,Distance\,Travelled}{\sum Total\,Time\,Spent}$$

Temporary mean speed is calculated using an average of instantaneous speed ,which is not a global way to characterize the mean speed in a specific zone. Mean spatial speed takes all the distance travelled and time spent information of all vehicles in a specific geographic zone and in a specific time slot, which can be more accurate and representative than the classic temporal mean space.

4.1.2 Zone division

Inspired by the ancient developper, the approach we implemented is to divide the area into **N * N** small squares firstly (due to the limit of computation power, we fixed N = 8), which has enough GPS signals to represent the whole traffic condition in this zone. We notice that GPS signals are centralized in each main avenue, for example, first and second ring road, city surrounded highway and city centre.

The figure below illustrates the heatmap of 300K gps signals sampled randomly from the file gps_20161115. The level of color red represents the quantity of gps signals number. (Dynamic heatmap figure can be found in Annexe)

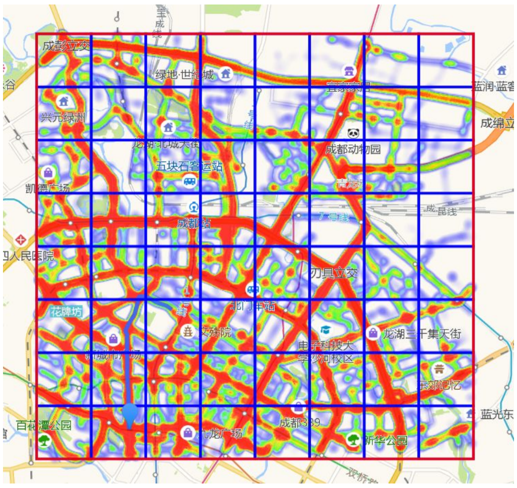


Figure 4.1 : gps signals heatmap with 300K samples in a day

In the figure 4.1, the surrounding red square shows the range of gps signals in GPS table and the blue lines show the line of division for 8*8 squares.

Combing the dynamic heatmap, we have a global understanding of traffic situation in Chengdu:

1. The traffic condition are getting better as moving away from the city centre.
2. The majority of vehicles are diving in the main road.
3. Some places like traffic centres might always be in a bad condition even if they are far away from city. (Eg: Train station, Airport, Long distance bus Station, commercial centre, industrial centre)
4. Many overpasses are already built in each Ring Road. Obviously, There will be more vehicles driving on overpass than below. Besides, vehicles are driven quicker on overpass

In order to better represent the traffic condition with Mean Spatial Speed, we are supposed to insure high homogeneity in the same zone(Eg: driving direction, Speed limit, etc).

Generally, we divide Chengdu by its 3 Ring Road [8] shown by the Figure below in left. Then we separate more precisely each zone with their location.

- Zone 1, 2 and 3 are inside the First Ring Road;
- Zone 4, 5 represente the First Ring Road;
- Zone 6, 7 and 8 represente the Second Ring Road;
- Zone 9 to 13 are outside the Second Ring Road.

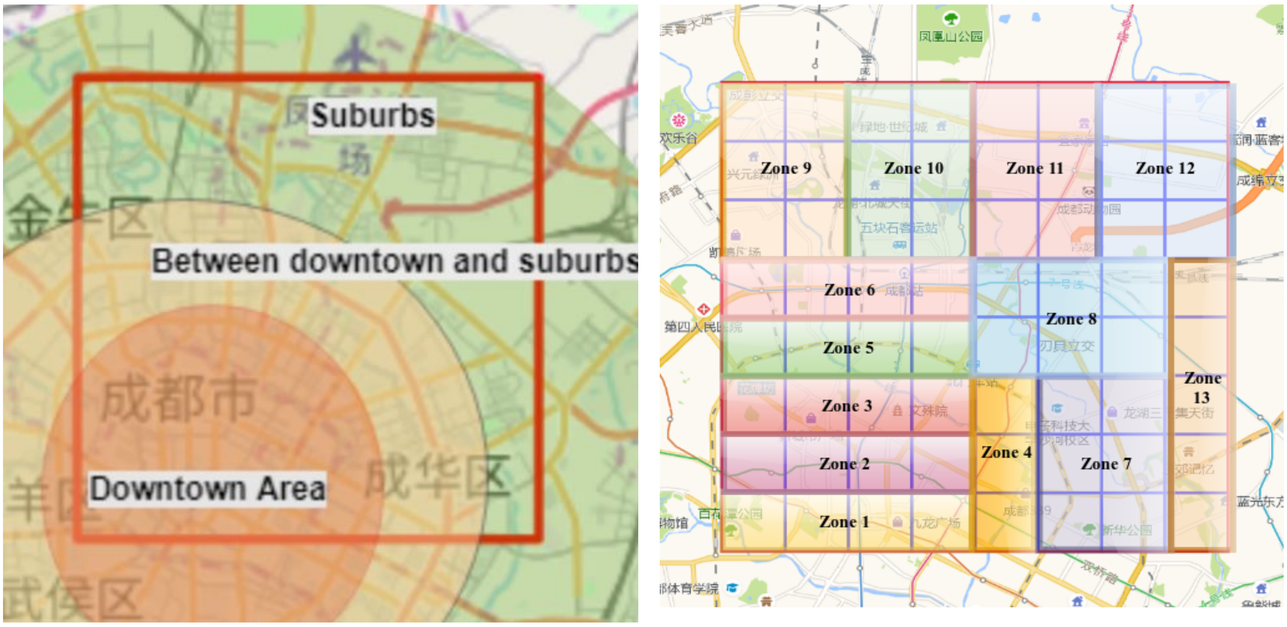


Figure 4.2 : zone division

With this zone division,

- We successfully put the most busy downtown area into zone 1 and 2
- Zones representing the Ring Road have the same direction with Roads in order to minimize the influence below the overpass.
- We avoid putting the special place in the boundary of the zone, so only one zone will be influenced by the special place (like Airports, commercial centre).

4.1.3 Time division

Length of time slot is fixed to 5min for the sake of simplicity, which means 00:00 - 00:05 is the first time slot in a day, and 23:55 -00:00 the last timeslot.

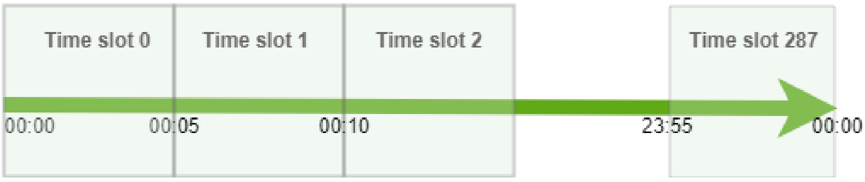


Figure 4.3 : Time slot presentation

4.1.4 Distance between two neighbouring GPS positions

There can be an error in calculation of distance between two GPS position,if cars turn around in a corner. For example, if two GPS position are not at the same road as shown in the following figure, it will cause obviously an error unignorable by using haversine distance which return the shortest distance between two geographical coordinates.

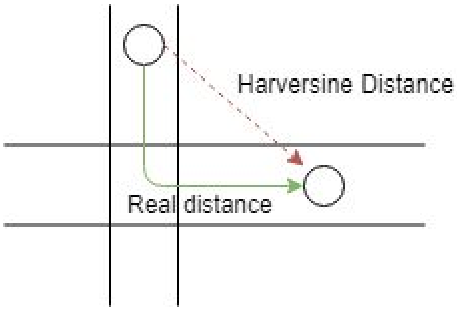


Figure 4.4 : Error with haversine distance

We propose to use GaoDe API[3] request, with the same coordinate system as in GPS table, as mentioned before. Gaode Map is a chinese widely-used map tools, used as default in DiDi App. It's more accurate than other APIs. With using GaoDe API, we can easily have the shortest distance between two coordinates (GPS signals) without considering actual traffic condition.

Traffic limit of this API is 30,000 request /day /account. Our team applied 3 accounts for this project which means we can consult 90,000 request /day to GaoDe API. We hence decided to request distance for neighbourhood GPS signal pair those whose time interval more than 60s, in order to limit API request per file. In a nutshell, the distance between GPS signals is the result of API request or the distance Haversine is used for calculations, depending on the time interval between them.

4.1.5 GPS signal across geographical zone

For the across zone GPS signals, as shown in the following graph, the distance and time interval between C and D will be dropped. Data loss rate is about 1.16% in GPS file.

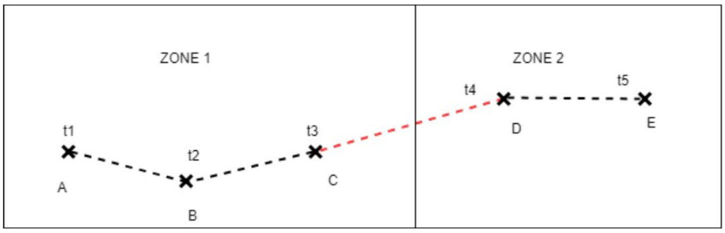


Figure 4.4 : GPS signal cross geographic zone

4.1.6 GPS signal across time slots

As shown in the graph, from A to B, the GPS signal across three time slots, the distance should be distributed according to time proportion. About 1.17% of data have time slot crossing problem. And about 80% pairs of GPS signal in the same order have a time interval <= 15s. Considering efficiency and accuracy, the across time slot problem will be dealt in the condition that the time interval larger than 15 seconds.

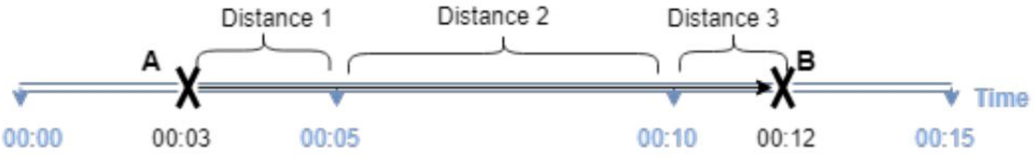


Figure 4.5 : GPS signal cross time slot

4.1.7 Deal with GPS signal cross timeslot and cross zone

We calculated the average of all GPS speed within the zone in a five-minute. For clarity, the following table shows an example of problematique cross zone issues.

ID	Current Timestamp	Last Timestamp	Current time slot	Last time slot	zone	Last zone	Time interval (current - last)	Distance (current - last)
1	00:02 (departure of an order)	NAN	0	NAN	11	NAN	NAN	NAN
2	00:12	00:02	2	0	12	11	10min	100m
3	00:14	00:12	2	2	12	12	2min	20m
4	00:21	00:14	4	2	12	12	7min	70m

Table 4.6 : Cross zone GPS signals example

Algorithms for resolving the across zone and across time slot problem represents in the following:

```
Move down the last timestamp, zone, time slot to current timestamp
If zone != last zone:    [Cross zone problem #ID = 1]
        Delete row
If zone == last zone and timeslot != last timeslot: [Cross timeslot problem #ID = 4]
        Add new timeslot rows
        Distribute distance to new row
```

The following is the result of our algorithm :

ID	Current Timestamp	Last Timestamp	Current time slot	Last time slot	zone	Last zone	Time interval	Distance (current - last)
1	00:02 (departure of an order)	NAN	0	NAN	11	NAN	NAN	NAN
2 #Deleted	00:12	00:02	2	0	12	11	10min	100m
3	00:14	00:12	2	2	12	12	2min	20m
5	00:14:59	00:14	2	2	12	12	1min	70* (1/7) = 10m
6	00:19:59	00:15	3	2	12	12	5min	70* (5/7) = 50m
4	00:21	00:20	4	3	12	12	1min	70* (1/7) = 10m

Table 4.7: Result after zone and time slot algorithm

4.1.8 Dropped data resume

In order to keep the mean spatial speed more accurate, the following data considered as error signals will be dropped out before model construction.

Dropout condition	When	Loss rate (ex: gps_20161114)
Orders whose gps signal count < 4	Before calculating spatial speed	0.015%
GPS tracking data interval > 20 min	Before calculating spatial speed	0.0003%
Cross zone GPS signals	Before calculating spatial speed	1.1124%

Table 4.8 : Dropped data resume

4.1.9 Visualization of mean spatial speed

The figure below shows the mean spatial speed in every five minute for each zone. We can observe that the speed changed in each zone are very similar. From 0h to 6h, the speed is faster. But in those periods from 7h to 9h, which corresponds to Morning-Rush, the speed begins to decline. The same condition happens from 17h to 19h , which corresponds to Evening-Rush. Then the speed starts to rise from 20h to 23h.
Although the speed changes in each zone are similar, the speed in different zones varies. For Zones 1 to 5, which belong to 1st circle of Chengdu , the speed is relatively slow. For Zones 6 to 8 which are in the 2nd circle of Chengdu, the speed is moderate. For Zone 9 to 13 ,which are outside the 2nd circle of Chengdu, the speed is relatively high.

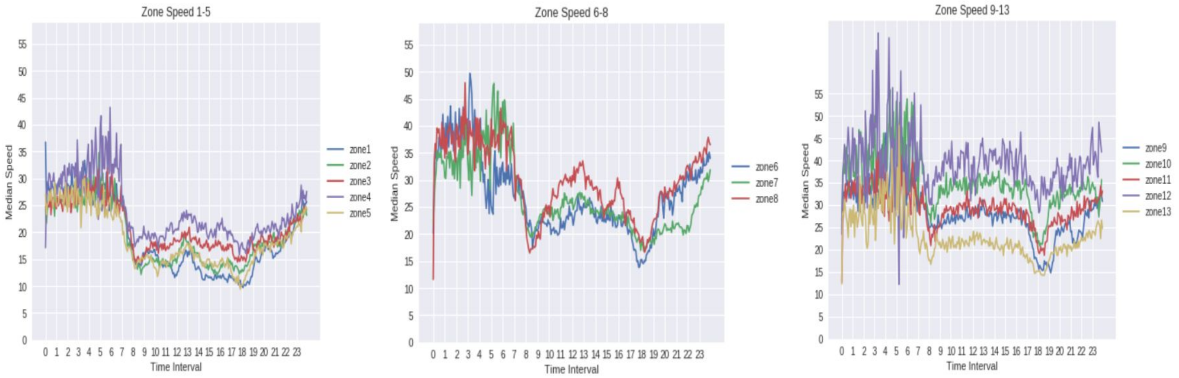


Figure 4.9 : Zone speed visualization

The distribution of speed is shown in the following figure, it slightly skewed to left. To be strict, speed distribution can not be considered as a normal, according to the Q-Q plots in the figure 4.10, moreover, it doesn't pass the normality tests.

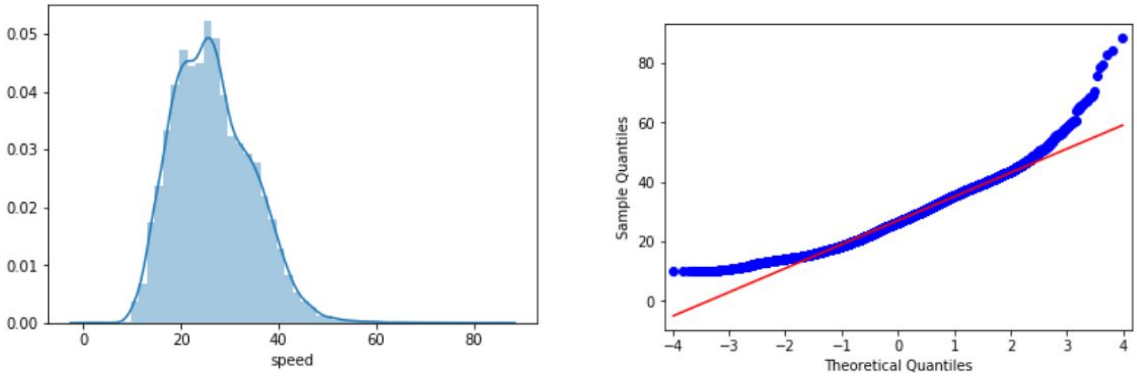


Figure 4.10 : Speed distribution and Q-Q plot

4.2 Feature engineering

The figure below shows features and the label we selected, as well as its relationship, which will be used in our model. Features include zone, weekday, number of orders, number of vehicles, time slot and group of traffic hours. Our label is mean spatial speed in every five minutes.

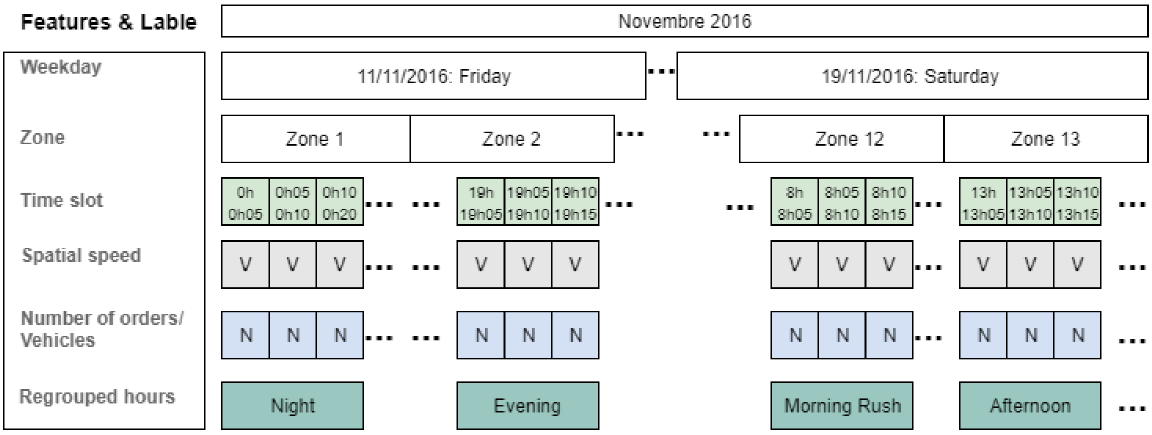
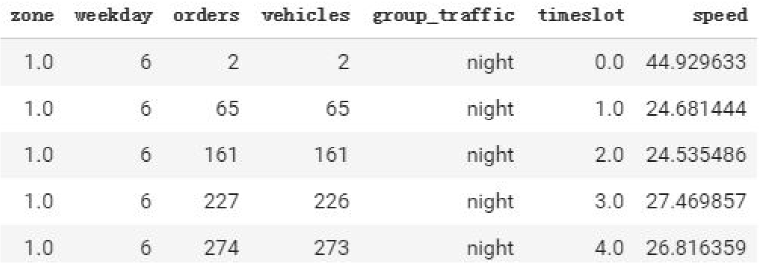


Figure 4.11 : Speed data Structure



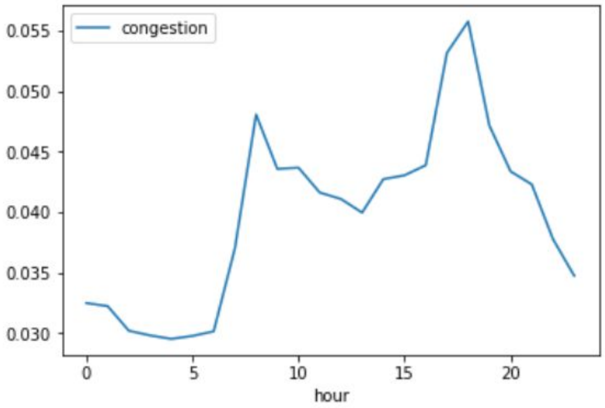
zone	weekday	orders	vehicles	group_traffic	timeslot	speed
1.0	6	2	2	night	0.0	44.929633
1.0	6	65	65	night	1.0	24.681444
1.0	6	161	161	night	2.0	24.535486
1.0	6	227	226	night	3.0	27.469857
1.0	6	274	273	night	4.0	26.816359

Figure 4.12 : Speed data example

4.2.1 Feature adding - Grouped traffic hours

The following graph shows the average congestion index of each hour, grouped traffic hour is created according to its distribution. All time slots in a day are regrouped to form 6 subgroups, as show in the table 4.14.

$$Congestion\ index = \frac{1}{Spatial\ Speed}$$



Time period	Grouped hours
23:00- 6:59	Night
7:00- 8:59	Morning rush
9:00-12:59	Morning work
13:00-15:59	Afternoon
16:00 - 18:59	Evening rush
19:00-22:59	Evening

Figure 4.13 : congestion per hour

Table 4.14: Rules for grouping traffic hours

5. Model construction and evaluation

5.1 Method classic

5.1.1 XGBoost : Model construction

In order to build an efficient, accurate and fast-running model, we choose XGBoost. This algorithms is built for the tradeoff between accuracy and training time. Moreover, it's feasible to run a XGBoost in personal laptop with multi-core.

- Package: XGBoost, sklearn, pandas, seaborn
- Model: XGReggressor
- Data Shape: (6 features + 1 target) * 26180 rows
    - Feature Selection:
        - Quantitative Features: vehicles, orders
        - Categorical Features: timeslot, zone, weekday, group traffic(identify the period of a day), using one-hot coding in model.
    - Target: Mean Spatial Speed (Continue Value)
- Parameters optimized:
    - n_estimator = 50
    - learning_rate = 0.03
    - max_depth = 7
    - min_child_weight = 8

In order to control Overfitting problem in such a classic Machine Learning model, the cross validation set will be a quite useful methods. So we splitted our dataset into 3 parts:

- 60% training set,
- 20% cross validation set,
- 20% test set.

However, we still limited the max depth of each tree, and tuning regularization factor. But the most efficient way is to expand our dataset.
Learning Curve is shown in the following figure.

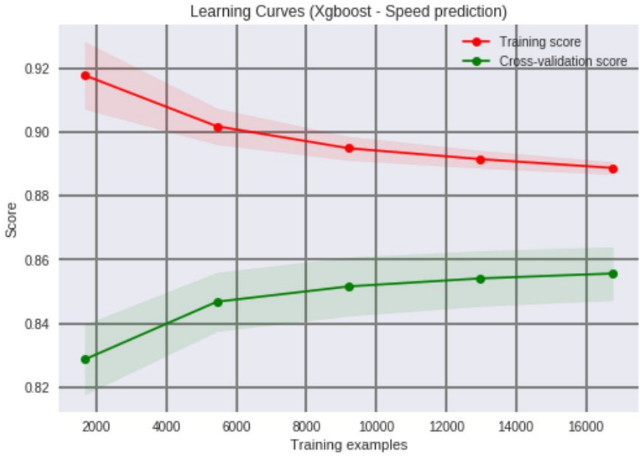


Figure 5.1: Learning Curves (XGBoost regression)

5.1.2 Prediction distribution

As shown below, this figure presents that the prediction are quite similar with the ground truth.

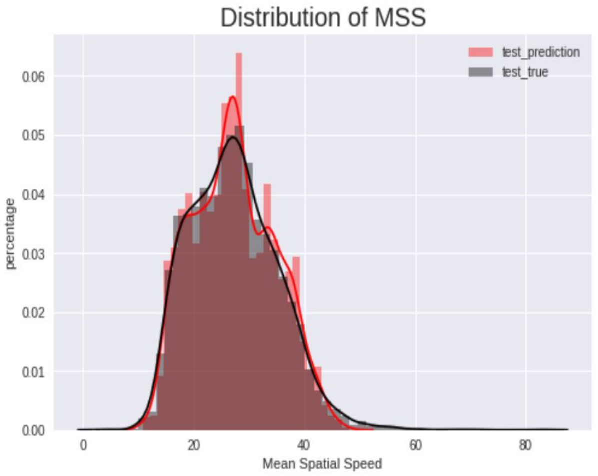


Figure 5.2 MSSpeed Prediction and True Value Distribution

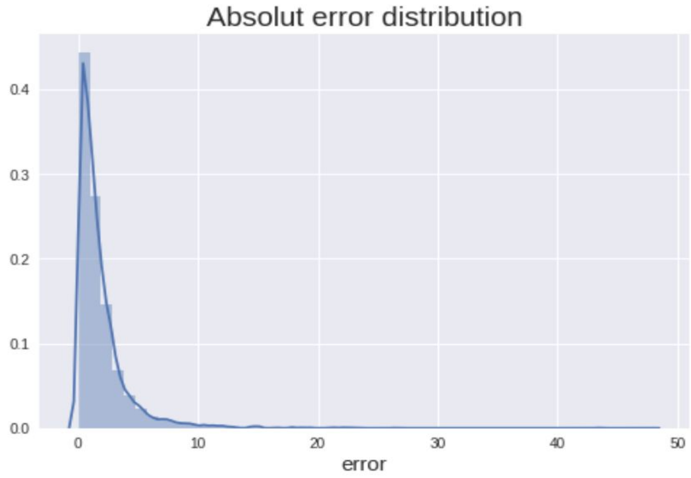


Figure 5.3 Spatial Speed Distribution

5.1.3 Evaluation Metrics

The definition of evaluation metric is defined in the following.

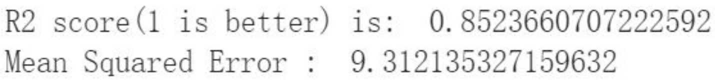
- **Mean Squared Error (MSE)**

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

- **Coefficience of Determination (R2 score)**

$$SS_t = \sum_{i=1}^{m} (y_i - \bar{y})^2 \qquad SS_r = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \qquad R^2 \equiv 1 - \frac{SS_r}{SS_t}$$

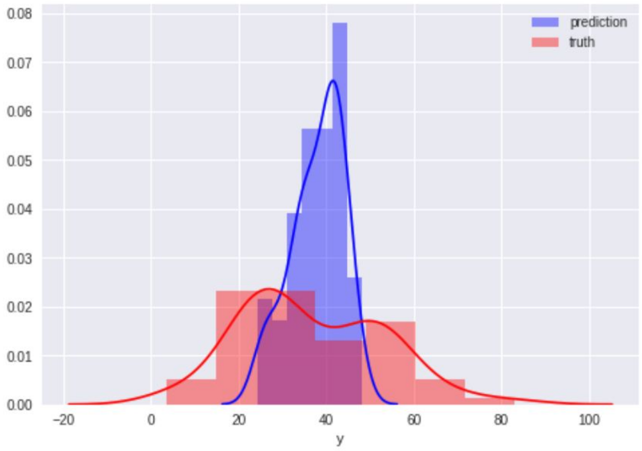
Test Dataset splited from Dataset in 20%, as for these two evaluation metrics, we have :

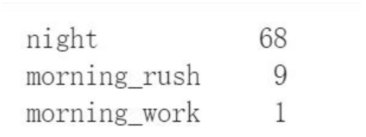


```
R2 score(1 is better) is:  0.8523660707222592
Mean Squared Error :  9.312135327159632
```

To dig more, we analyzed predictions with error more than 10, because 10 km/h would be a big change for city traffic condition.

We found that our predictions centralized in 40km/h but true values' distribution has bigger variance than these big error predictions. In order to solve the problem, we need more features to describe "in night" or to verify if the calculation of Spatial Speed made mistakes.





```
night            68
morning_rush      9
morning_work      1
```

Figure 5.4 Histogram--Error > 10

Finally, some problems need to be mentioned like categorical features in one-hot coding. Those splitted features won't be recognized as one feature by XGBRegressor model. Sometimes, the algorithms inside will subsample a part of them. As for 'zone', that will decrease differences between zones.

5.2 Multivariate time series

5.2.1 LSTM : Model construction

In each geographic zone, the congestion situation represented by mean spatial speed is a time-dependent variable, and it depends on as well as other variables like numbers of orders and vehicles numbers. Thus, predicting the mean spatial speed can also be treated as a multivariate time series problem. A temporal representation for each variable in zone 1 is shown in following figure.

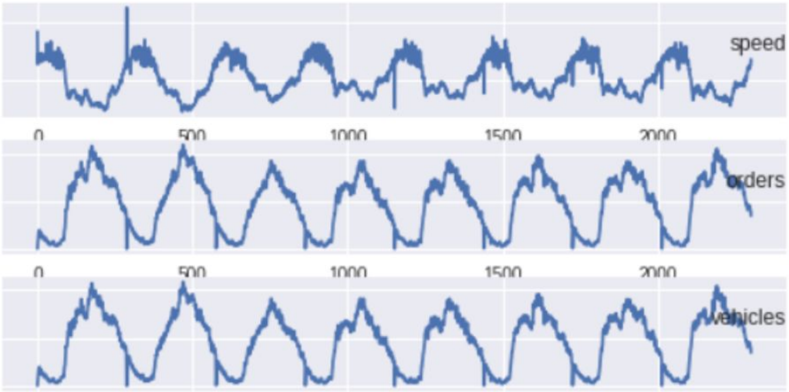
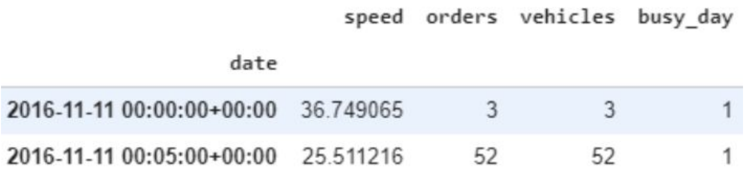


Figure 5.5: Zone 1 - Time-dependent variables

Neural networks like Long Short-Term Memory (LSTM) are able to modelize multivariate time series problem, but classical time series methods like ARIMA can be difficult to adapt to multiple input time series problems[7]. At same time, comparing to traditional RNN (recurrent neural networks), LSTM has a larger memory which is more suitable in our project. That is the reason why we have chosen the model LSTM.

The dataset is taken from the GPS data from 2011-11-11 to 2016-11-18 of zone 1, as shown in the following table. Column “date” represents the start time for the time slot where the gps signal should be. Column “busy_day” define whether current day is Friday and Saturday or not. As explained in the precedent section, these two days have more orders and vehicles than other days, we assume that it is an important feature.



date	speed	orders	vehicles	busy_day
2016-11-11 00:00:00+00:00	36.749065	3	3	1
2016-11-11 00:05:00+00:00	25.511216	52	52	1

Figure 5.6: S   Sample of dataset

In order to use LSTM, first step is to frame time sequence as a supervised learning problem. We created inputs using last 12 time slots (t-12 to t-1), which represent 1 hour, and the speed at the current time slot **t** as the output, the structure is shown in Figure 5.6;

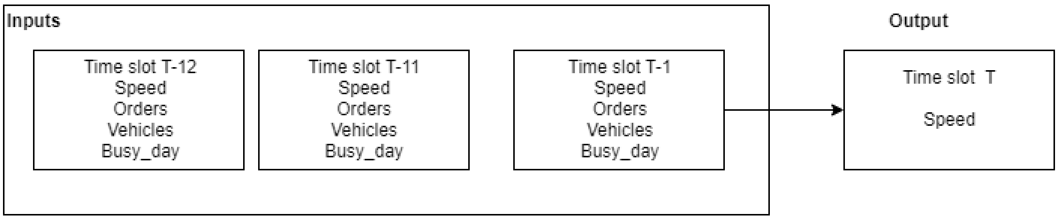
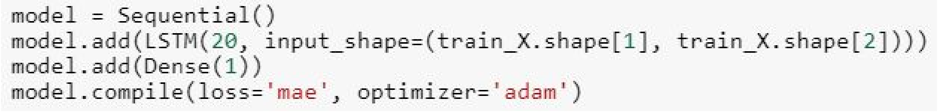


Figure 5.7: Input and output for LSTM

Network structure is represented in the following. First layer is LSTM with 12 units, following by a densely connected Neural Network layer, ending with the output.



```
model = Sequential()
model.add(LSTM(20, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')
```

Figure 5.8: LSTM network structure

5.2.2 Model evaluation and prediction

With 50 training iterations, as show in the figure 5.3, training error keeps reducing. Cross validation error also keeps decreasing, but with a little fluctuations. It shows that model fits well dataset without severe overfitting.

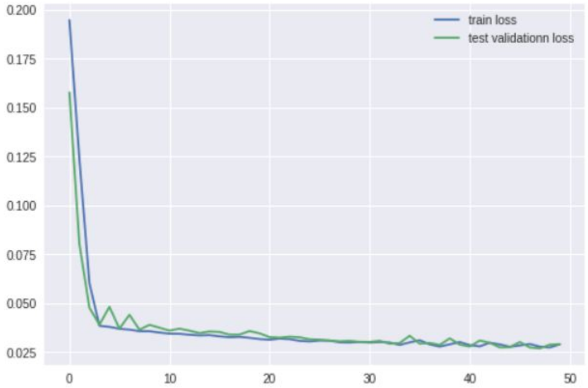
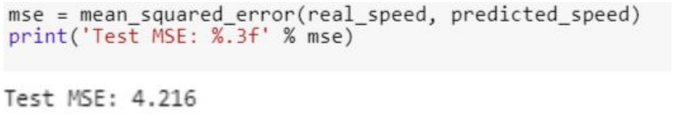


Figure 5.9: Loss on the Train and Test Dataset

Mean square error within test set is shown in the following, which is 4.21.



```
mse = mean_squared_error(real_speed, predicted_speed)
print('Test MSE: %.3f' % mse)

Test MSE: 4.216
```

Figure 5.10: Model evaluation MSE

Model is constructed using dataset from 2016-11-11 to 2016-11-18, the predictions are for day 2016-11-19, the prediction results of LSTM network is shown in the following, in green. The model is able to make consistent predictions.

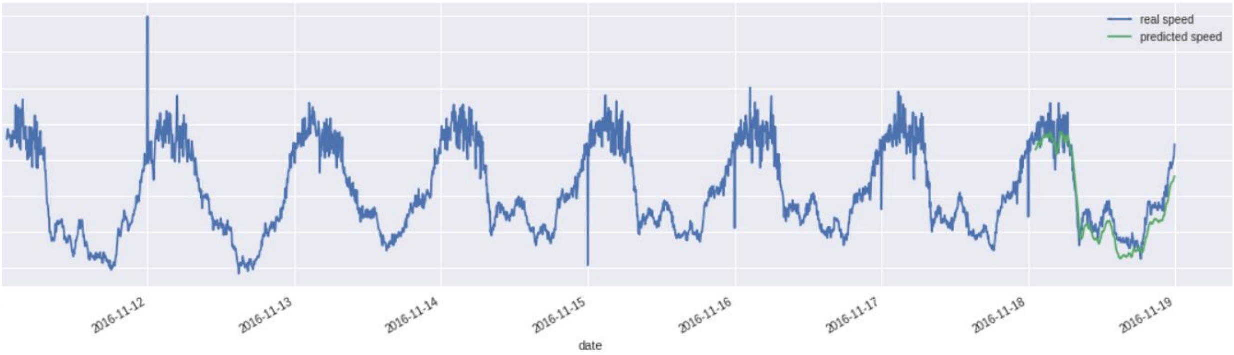


Figure 5.11: Prediciton results

6. Perspective

This section introduces some possible ways for further improvement.

- **Append new features**

There are new features which can be added into dataset, for example, different geographic zone can be labeled with commercial, school, business zone. Furthermore, there could be useful information deriving from open data resources, like weather condition, national holiday ect. More features may lead to a better result. Finally, clustering methods can be used for geographic zone creation and grouped daily hours creation.

- **Implement with better model**

The biggest problem we met in the modelisation part is that the time-consummation is a bit high during tuning XGBoost model. In the future deployment, it can be well considered to use Catboost, LGBM to accelerate the model construction, or using deep learning model to have a better performance. A lot of optimization work should be done to LSTM as well.

- **Using Big data architecture to reduce execution time**

In order to gain a better performance, we avoid using function like **apply** and **map** in Pandas, which eat a lot of RAM and CPU. With the disposal of a well performance server, the time consumption per GPS file is approximately 2h per file, which should be take into account for the future improvement. Big data architecture can be used in future work.

Core number	10
RAM	32G
Disk	300G

Table 6.1 : Server configuration

7. Conclusion

In this data mining project, following every step proposed by the methodology CRISP-DM, from Data understanding to Modelization and Prediction, we have succeeded to predict the mean spatial speed from DiDi company's order database.

In order to make sure the accuracy of Mean Spatial Speed, the calculation occupies about 90% of whole project time, which is the vital step in our project. However, we are not able to find a authoritative reference but analyze with our own background experience. Next step, we may consult more data platforms like Ali Cloud OpenData, Baidu Traffic Condition, etc, to better represente Mean Spatial Speed.

In the modelization part, classic machine learning method with XGBoost and multivariate time series with LSTM give both a good performance with low '***RMSE*** '. These two methods are not ready to be used in real cases. They highly rely on the accuracy of our calculation.

Mean spatial speed is a good manner to represent the traffic flow and congestion, it surely can be used in traffic congestion prediction.

Further improvements will be explained in our next report.

Bibliography

[1] Cross-industry standard process for data mining
https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining
[2] ChengDu
https://zh.wikipedia.org/wiki/%E6%88%90%E9%83%BD%E5%B8%82
[3] Coodinate system transfer
https://github.com/caijun/geoChina/blob/5c6284b/R/cst.R#L101-L107
[4] Gaode API
https://lbs.amap.com/api/webservice/guide/api/direction
[5] Data visualization problem
https://tools.wmflabs.org/geohack/geohack.php?language=zh&pagename=%E6%88%90%E9%83%BD%E5%B8%82&params=30_39_35_N_104_03_48_E_region:CN-51_type:city&title=%E6%88%90%E9%83%BD%E5%B8%82
[6] Baidu Smart Transportation
https://jiaotong.baidu.com/
[7] Mulitiviate time series with lstm
https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/
[8] Ring roads
https://contemporary_chinese_culture.academic.ru/659/ring_roads