Machine Learning in Intelligent Transportation

Team: TNT_000

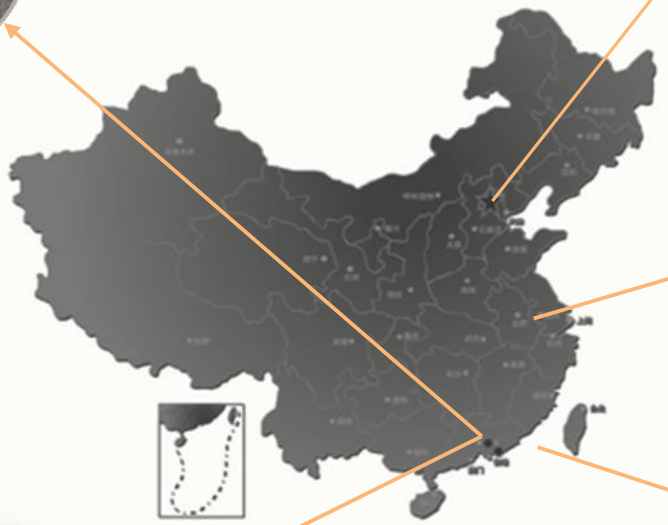
Contents

1. Team Members
2. Scenario & Solution
3. Feature Engineering
4. Model in Deep learning & XGBoost
5. Improved XGB with Boosting
6. Achievement Roadmap

Team Members

罗志鹏-北京大学
Microsoft

杨亚涛-中山大学
InplusLab



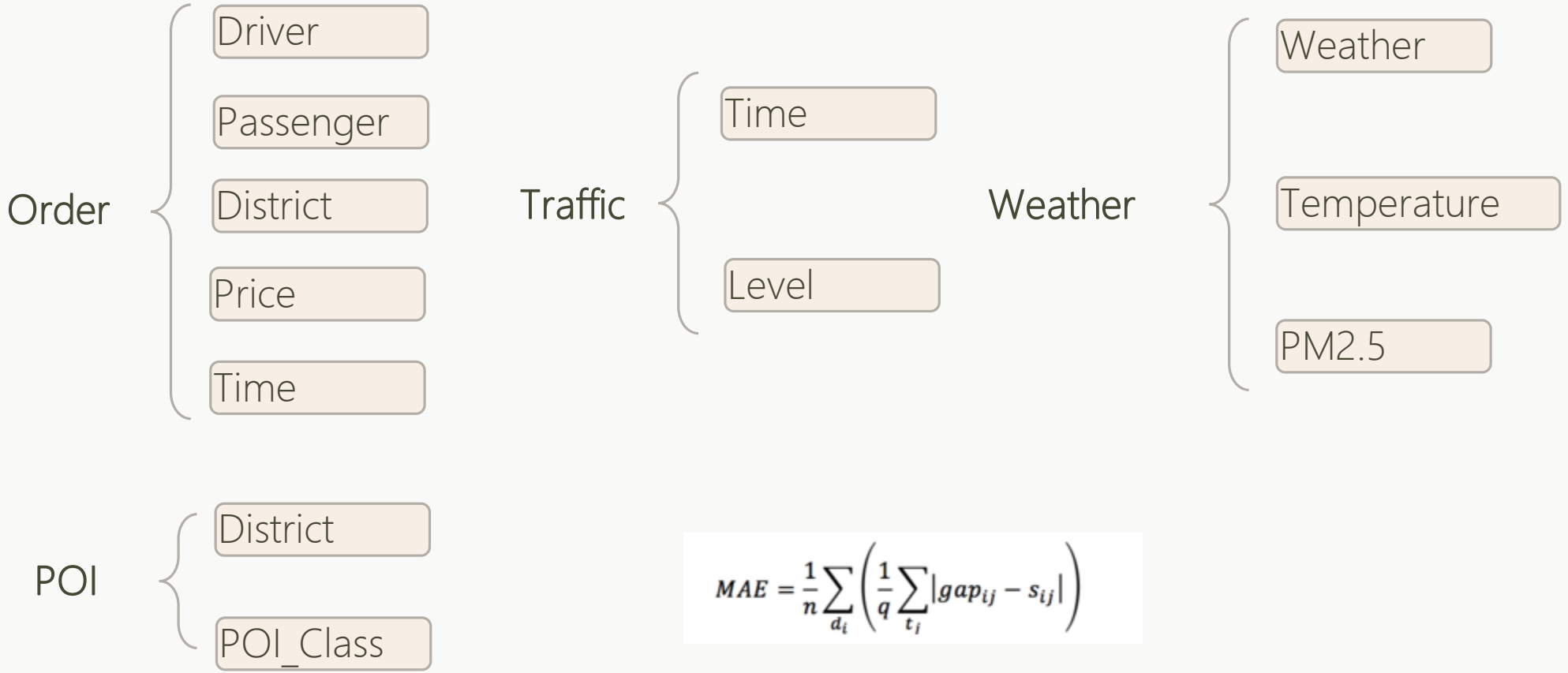
李超-西北工业大学
海康威视

黄春振-中山大学
InplusLab

赵伟-中国科学院
Tencent

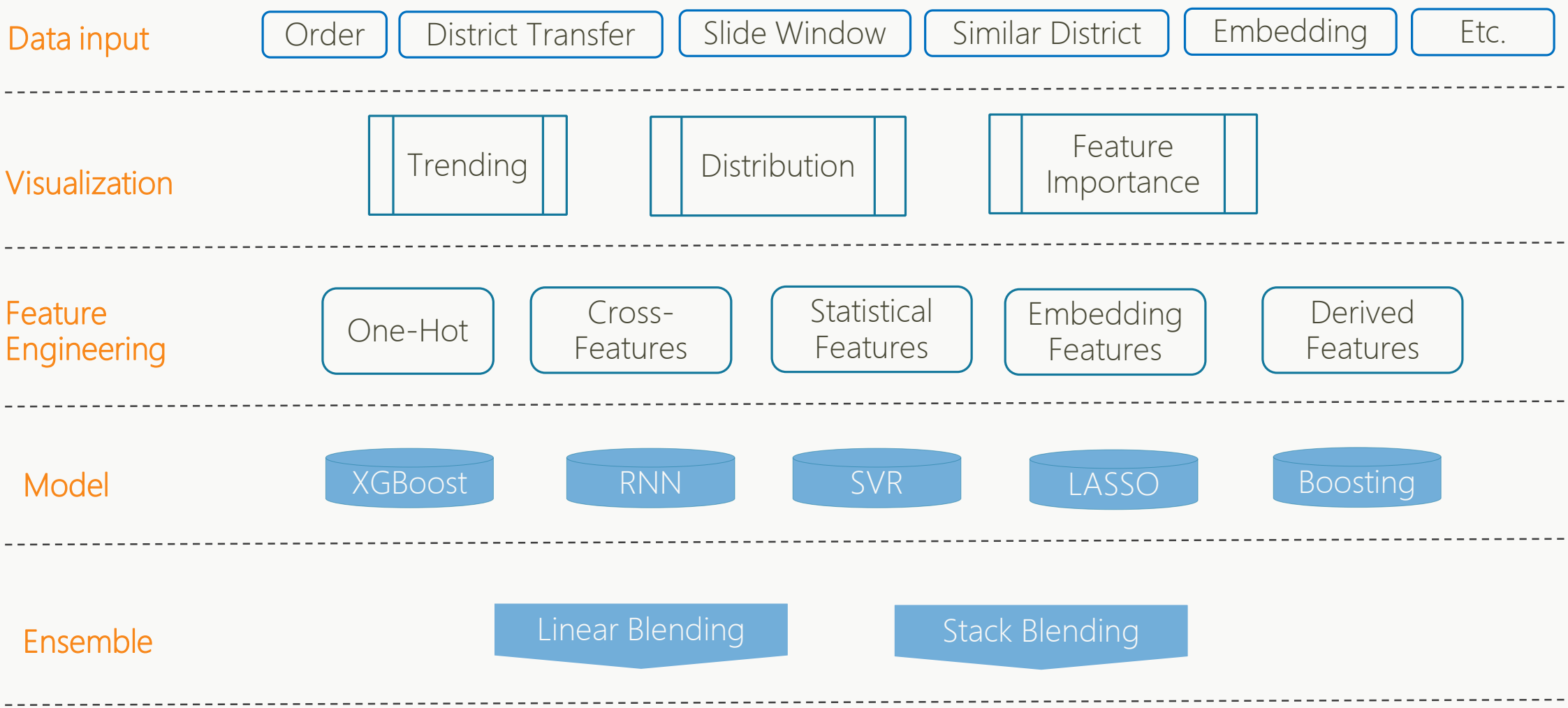
Scenario

Datasets: The training and testing data of two seasons.

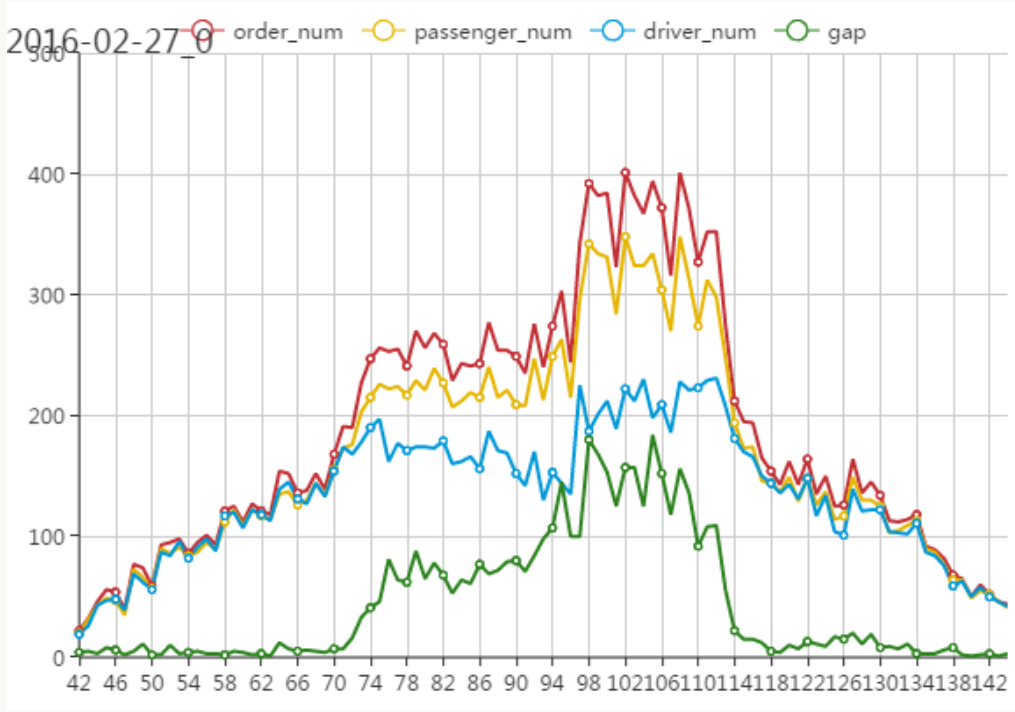


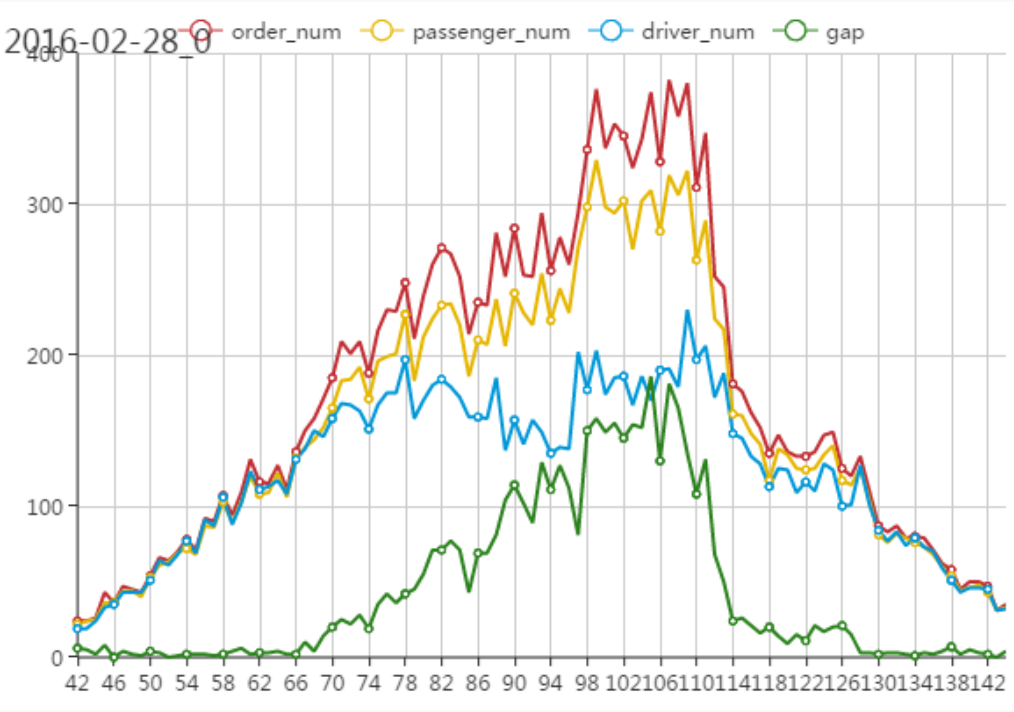
$$MAE = \frac{1}{n}\sum_{d_i}\left(\frac{1}{q}\sum_{t_i}|gap_{ij} - s_{ij}|\right)$$

Solution for Intelligent Transportation

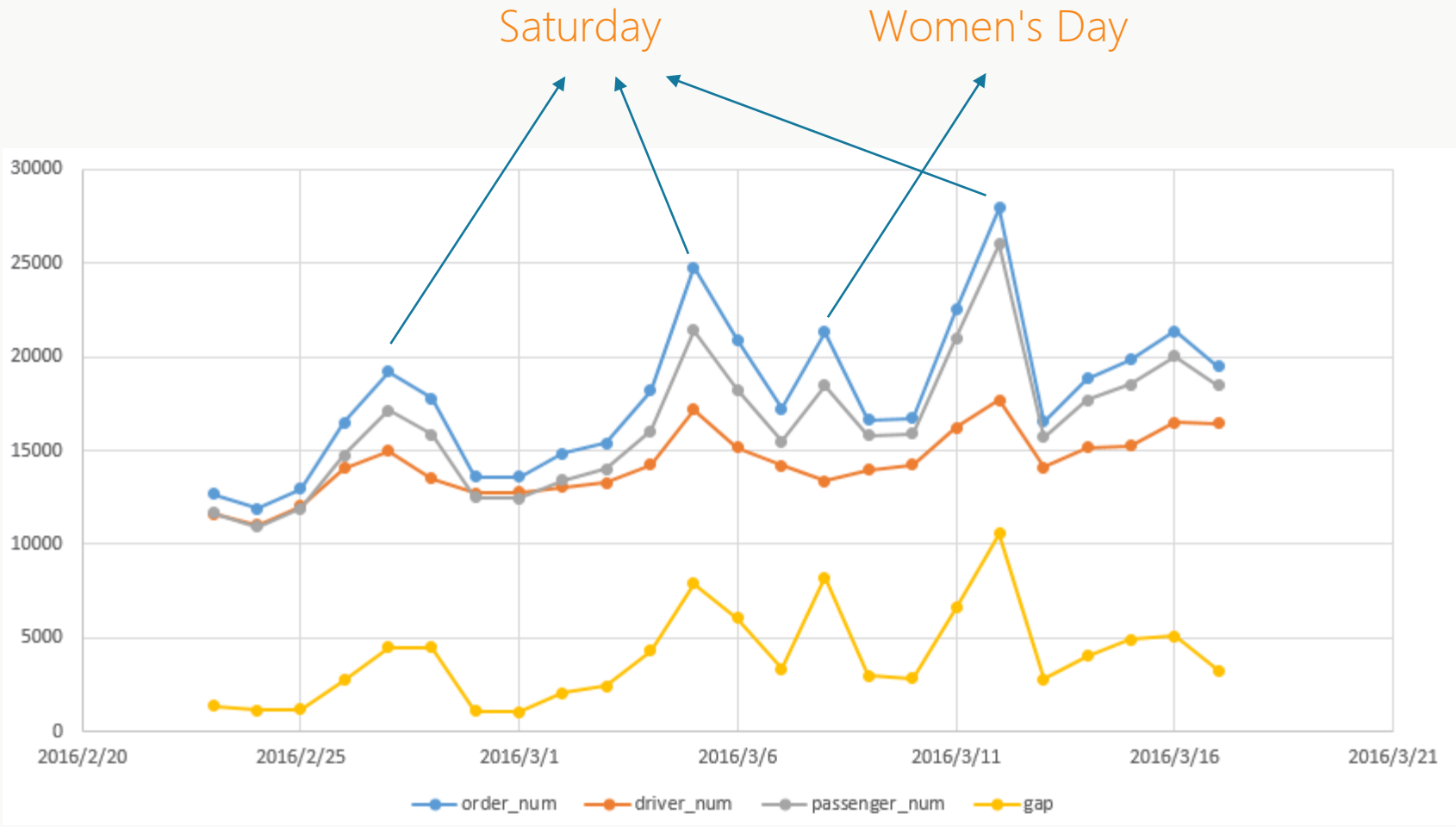


Data Visualization about Trending





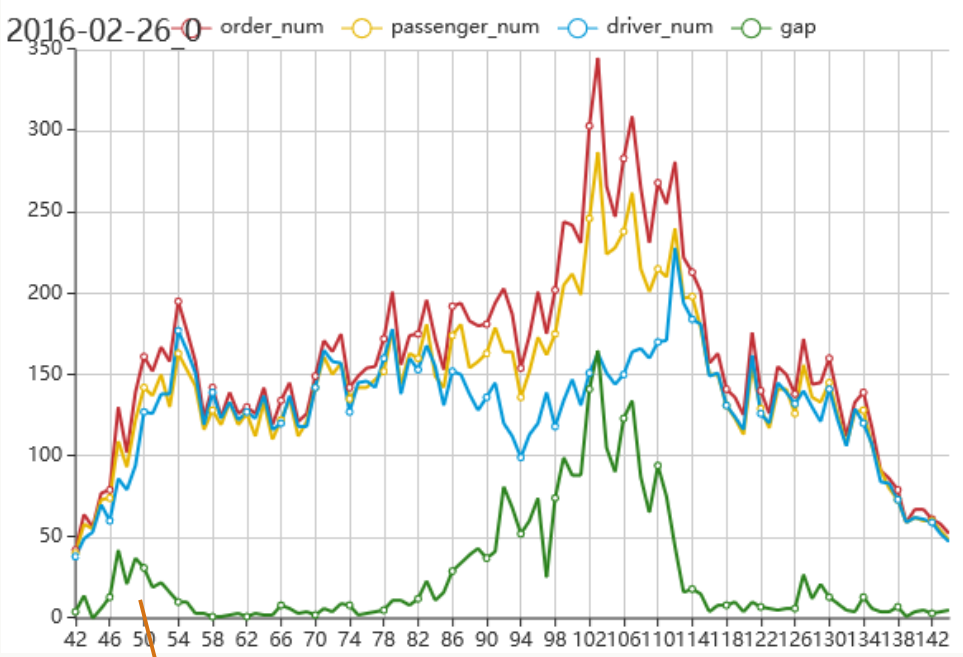
Trending during Season 2



Trending during 2.21-3.17 at district 8

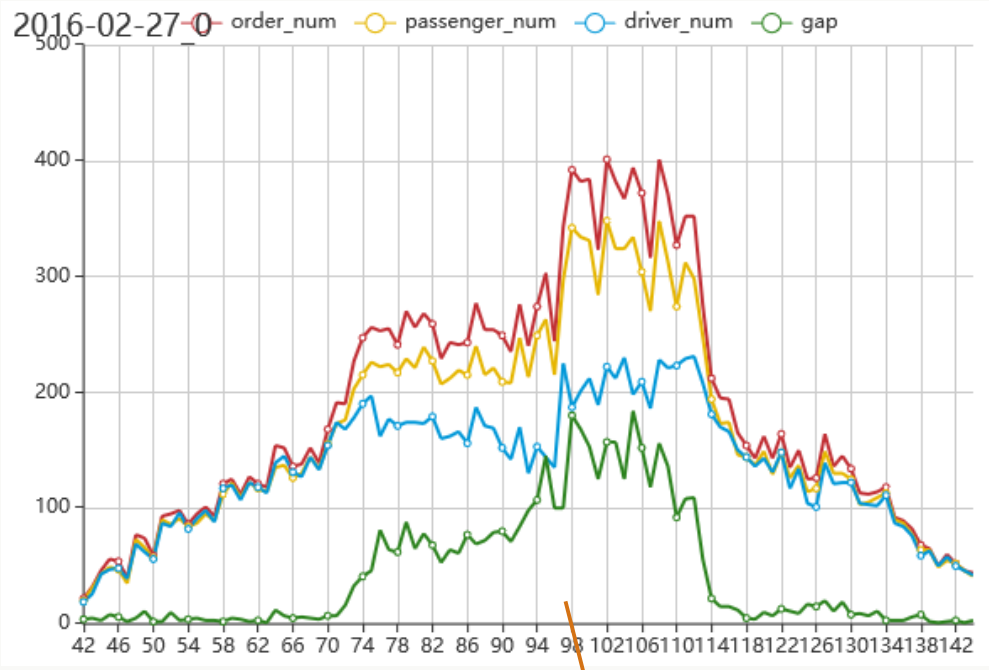
Workday vs Weekend

Friday



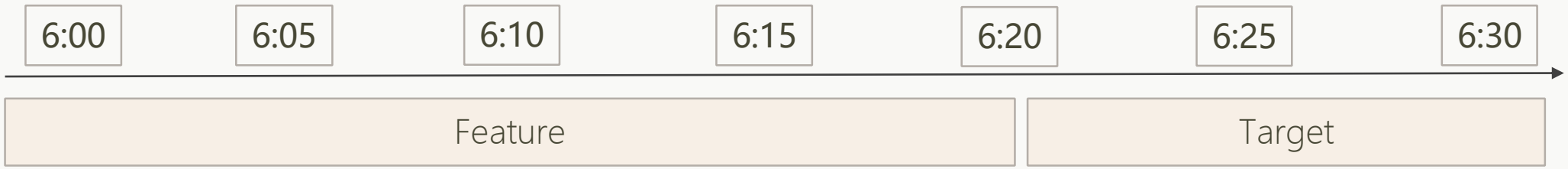
Higher gap from 7am-9am at workday

Saturday



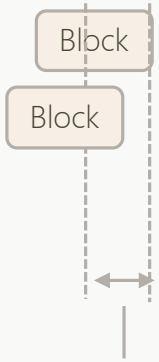
Higher gap from 12am-7pm on Saturday

Feature Engineering



Sliding time window to enlarge dimensions.

20 minutes for features -> 10 minutes for target



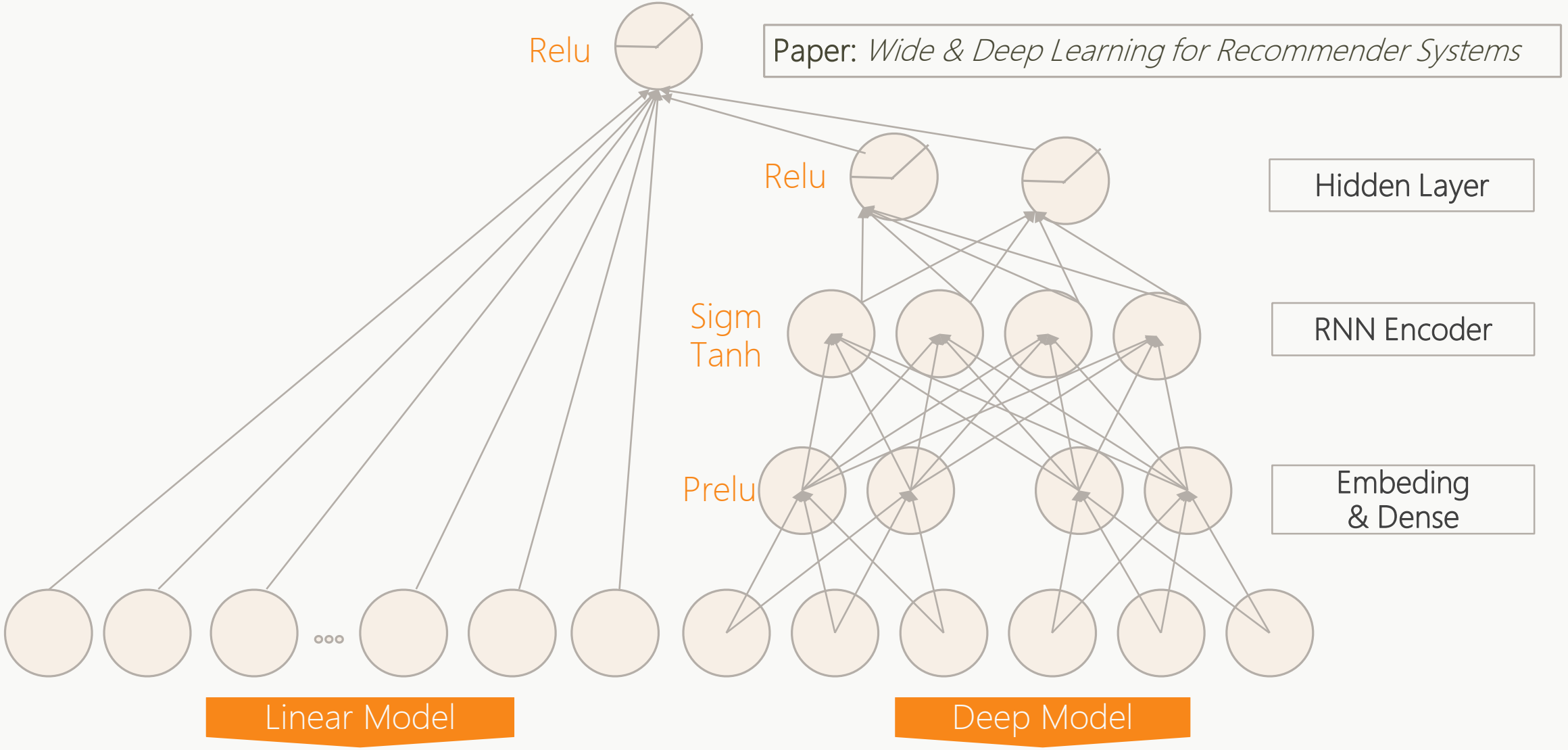
Generate features by blocks of time series.

By sliding 1 minute, all the blocks can be splitted.

Feature Engineering

- Order
  - Gap-Order
  - Gap-Passenger
  - Driver
  - Gap rate
- Embeding
  - District-Dense
  - Time Slice
  - Week Dense
- Slide window
  - Gap-Order
  - Gap-Passenger
  - Driver
  - Gap rate
- Others
  - Weather
  - Traffic Jam
- District Transfer
  - Order-Cnt
  - Average Price
- Similar District
  - Gap-Order
  - Gap-Passenger

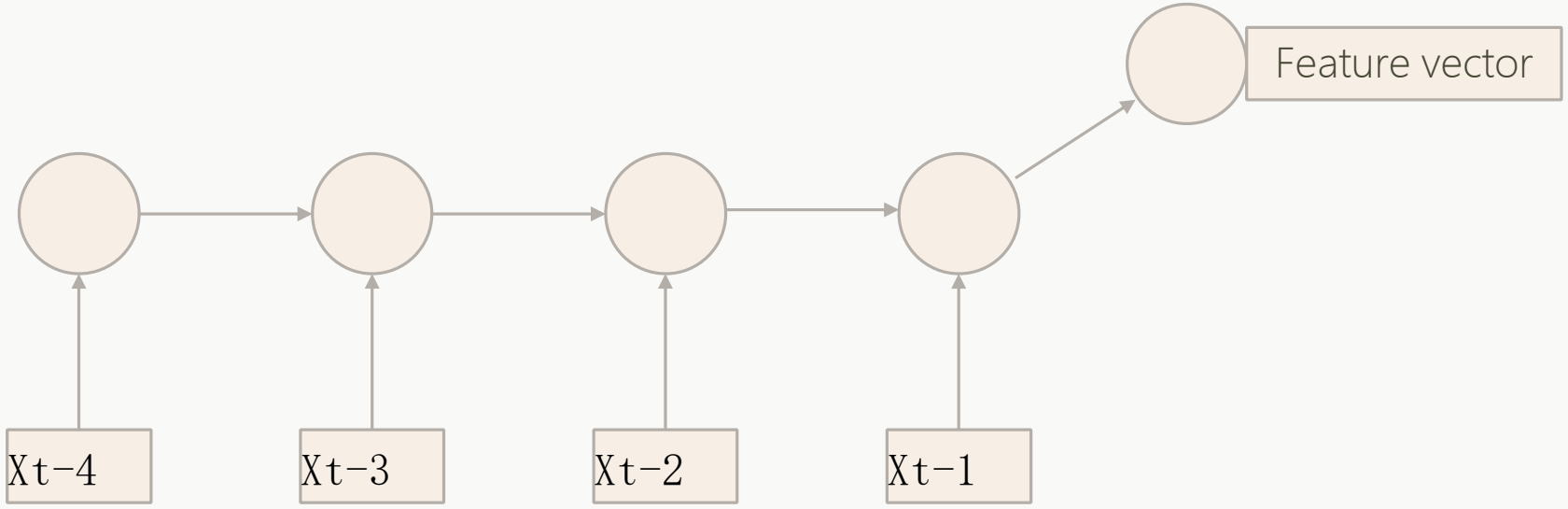
Deep Learning in Intelligent Transportation



RNN Encoder

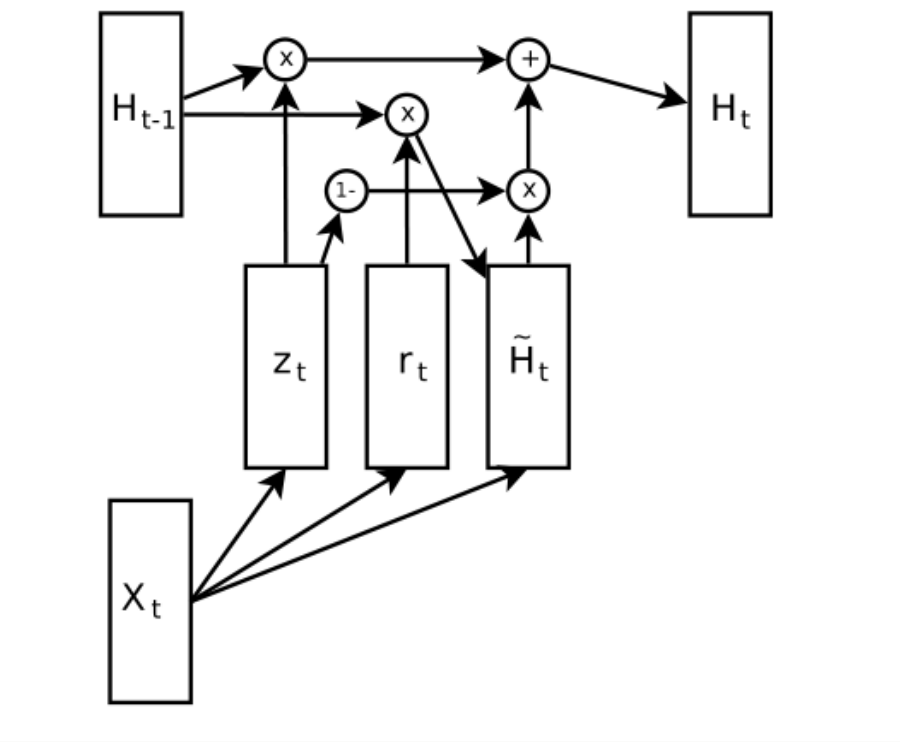
1. Abstract features of 20 minutes splitted by 5 minutes of four.

2. By using GRU model, we encoded time series feature as fixed-length feature vector.



GRU Model

Performance of GRU is same as LSTM, but computing speed of GRU is more faster.



$$
\begin{aligned}
r_t &= \text{sigm}\left(W_{\text{xr}}x_t + W_{\text{hr}}h_{t-1} + b_{\text{r}}\right) \\
z_t &= \text{sigm}\left(W_{\text{xz}}x_t + W_{\text{hz}}h_{t-1} + b_{\text{z}}\right) \\
\tilde{h}_t &= \tanh\left(W_{\text{xh}}x_t + W_{\text{hh}}(r_t \odot h_{t-1}) + b_{\text{h}}\right) \\
h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t
\end{aligned}
$$

An Empirical Exploration of Recurrent Network Architectures

XGBoost & LR with Boosting way

XGB learn the trainning data by default weights equally, but not consider some instances that are hard to learn. So we adjust the weights of instances by the boosting way.

Boosting

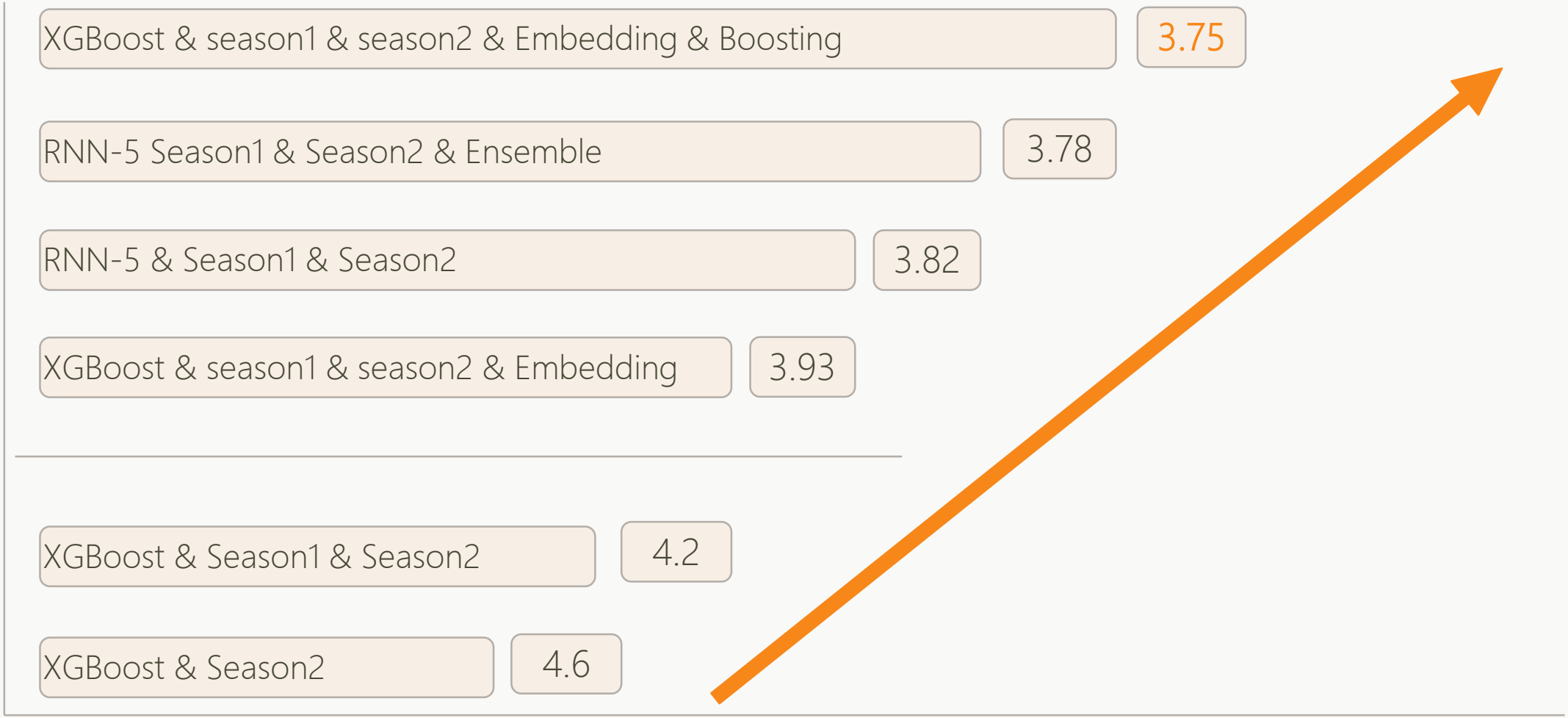
- Reweight instances
- Sampling by new weight
- Optimization coefficents

```
Start boosting training ...
Iter =  1 error =  0.4777267 beta = 0.2282228 w =  1.477433 gt_cnt =  74812 le_cnt =  81788
Iter =  2 error =  0.7693082 beta = 0.5918352 w =  0.5245271 gt_cnt =  72496 le_cnt =  84104
Iter =  3 error =  0.8408543 beta = 0.7070359 w =  0.3466738 gt_cnt =  72639 le_cnt =  83961
Iter =  4 error =  0.8821781 beta = 0.7782382 w =  0.2507226 gt_cnt =  73201 le_cnt =  83399
Iter =  5 error =  0.8985979 beta = 0.8074782 w =  0.2138392 gt_cnt =  71792 le_cnt =  84808
Iter =  6 error =  0.9024258 beta = 0.8143723 w =  0.2053377 gt_cnt =  69970 le_cnt =  86630
Iter =  7 error =  0.9098775 beta = 0.827877 w =  0.1888907 gt_cnt =  69056 le_cnt =  87544
Iter =  8 error =  0.9210467 beta = 0.848327 w =  0.1644891 gt_cnt =  68797 le_cnt =  87803
Iter =  9 error =  0.9294001 beta = 0.8637845 w =  0.1464319 gt cnt =  68573 le cnt =  88027
```

Paper: *Big Error Margin Boosting Algorithm Feely, 2000*

Paper: *AdaBoost+: An Ensemble Learning Approach for Estimating*

Achievement Roadmap

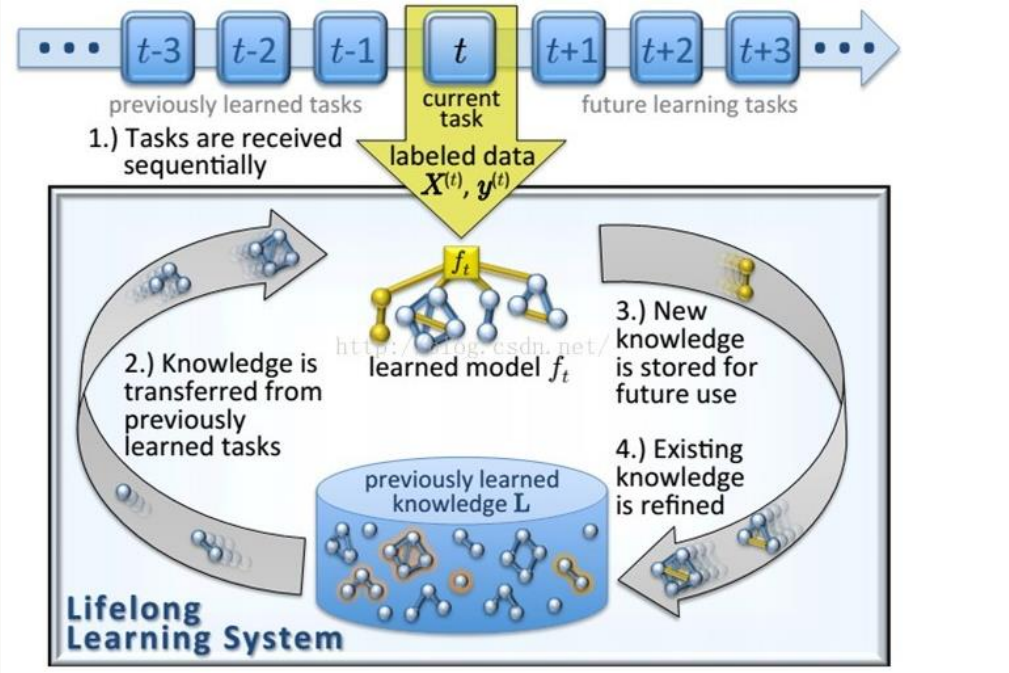


Not yet…

Transfer Learning

Multi-Task Learning

Lifelong Machine Learning



Thank you for your listening.

滴滴
滴滴一下 美好出行

滴滴Di-Tech算法大赛