



# Combination of Diverse Ranking Models for Personalized Expedia Hotel Searches

binghsu & MLRush & BrickMover Team

## **TEAM MEMBERS:**

**XUDONG LIU, BING XU, YUYU ZHANG, QIANG YAN,  
LIANG PANG, QIANG LI, HANXIAO SUN, BIN WANG**

# BACKGROUND

- Learning to rank hotels to maximize purchases

在北京 (及周邊) 找到749間飯店 網上預訂或致電 3077 4857

排列方式: 旅客評級 酒店名稱 酒店星級 **最熱門**

酒店 (平均價格)	3 星級 (平均價格)	4 星級 (平均價格)	5 星級 (平均價格)
HK\$852	HK\$383	HK\$701	HK\$1,332



**北京麗晶酒店** ★★★★★  
Regent Beijing  
4.6 /5 (190 則評價)  
Beijing (東城 - 王府井) - 地圖  
3077 4857

HK\$1,845  
**HK\$1,093**  
平均每晚  
Expedia 特別價格  
12/20 至 12/23  
✓ 免費取消

選擇日期



**王府半島酒店** ★★★★★  
The Peninsula Beijing  
4.4 /5 (188 則評價)  
Beijing (東城 - 王府井) - 地圖  
3077 4857

HK\$1,527  
**HK\$1,130**  
平均每晚  
Expedia 特別價格  
12/03 至 12/06

選擇日期

預訂您的旅程

☒ 酒店 ☐ 機票 + 酒店 ☐ 航班 ☐ 租車

預訂旅遊套票  
可慳高達 30%!

酒店  
搜尋以下城市、地標、機場附近的酒店:

北京



入住:  退房:   
yyyy/mm/dd yyyy/mm/dd

客房數目:  成人 (18+)  小童 (0-17)  
1 客房 1 2 0


顯示其他選項

**HK\$ 至抵價保證!** 搜尋酒店

北京麗晶酒店 選擇日期  
Regent Beijing ★★★★★ ✓ 至抵價保證  
99 Jinbao St Dongcheng Dist. Beijing, 100005 中國 3077 4857



**無得彈!**  
96% 所有書寫  
4.6 /5 Expedia 旅客評級  
根據 190 篇真實評論而定

 TripAdvisor 旅客評級  
根據 880 篇評論而定

查看所有照片

# BASIC ANALYSIS

- What we have?
  - Hotel features
  - Query features
  - User features
  - Compare features

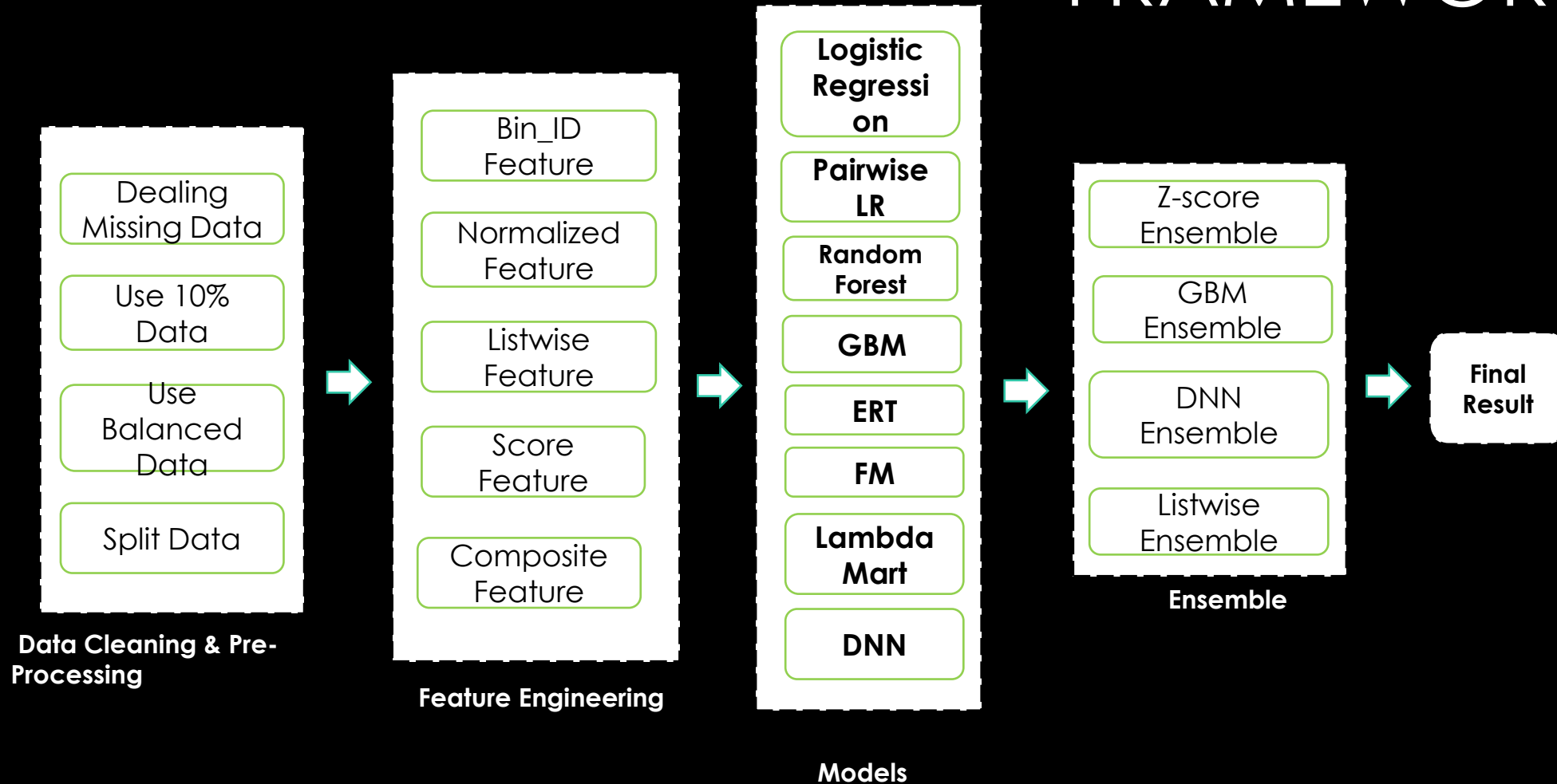
Cat	Num
Train	9917530
Test	6622629
Query in Train	336334
Query in Test	266230
Hotel Count in Train	136886
Hotel Count in Test	132888
...	...



# TO SOLVE

- How to deal with the complex data.
- How to generate good features from these data.
- How to make these features work in a model.
- How to make models train fast in such a big data.
- How to ensemble models.

# FRAMEWORKS



- 
- Preprocess

# PREPROCESS





# PREPROCESS(CONT.)

- Dealing with Missing Data

*Use the first quartile to represent the missing data*

- Use 10% Data

*Randomly sample 10% by srch\_id*

- Use Balanced Data

*Choose balanced positive and negative data*

- Split Data

*Split data by prop\_country\_id*



- 
- Features

# FEATURES

## Keypoints

Features are built in different methods

Some special features including listwise features and score features

Different features fit different models

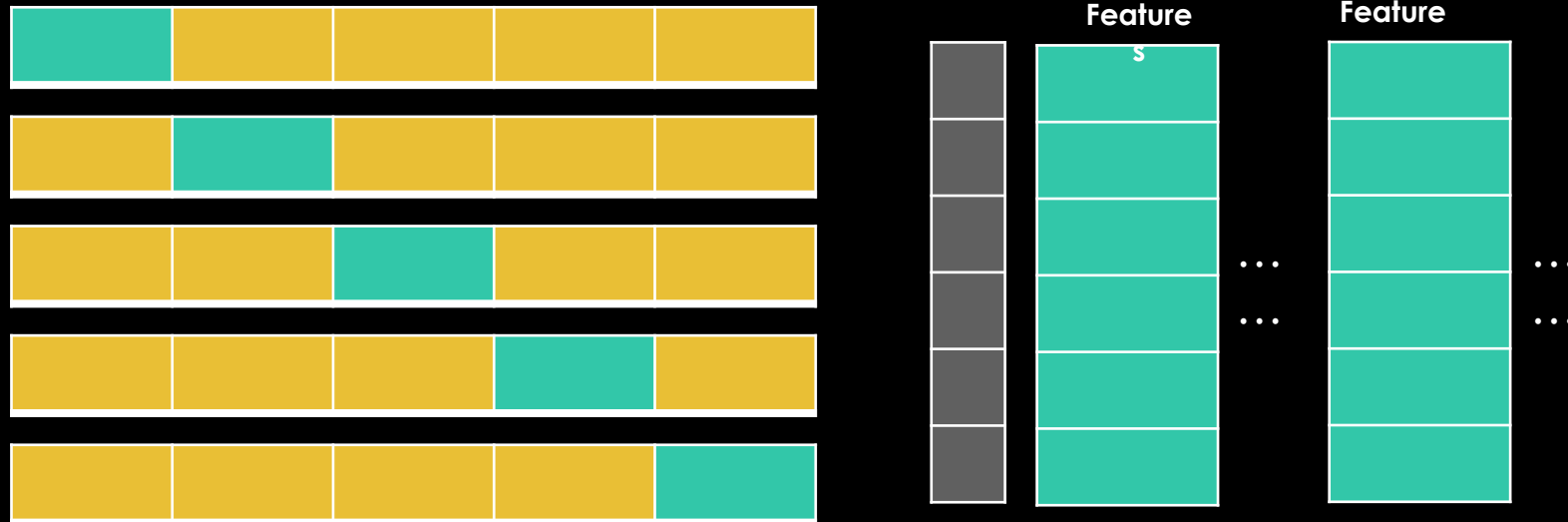
Cat.	Key Features
Bin_ID Feature	Prop_id Search_destination_id Srch_room_count Srch_booking_window
Normalized Feature	Price_usd Prop_location_score1 Prop_location_score2
Listwise Feature	Price_rank Price_diff_rank Star_rank
Score Feature	Fm_score Lr_score
Composite Feature	Roomcount_Bookwindow Adultcount_Childrencount

# FEATURES (CONT.)

- Listwise feature calculate the rank value in a query.
- Composite Feature

$$F1\_F2 = F1 * \max(\text{Max}(F2)) + F2$$

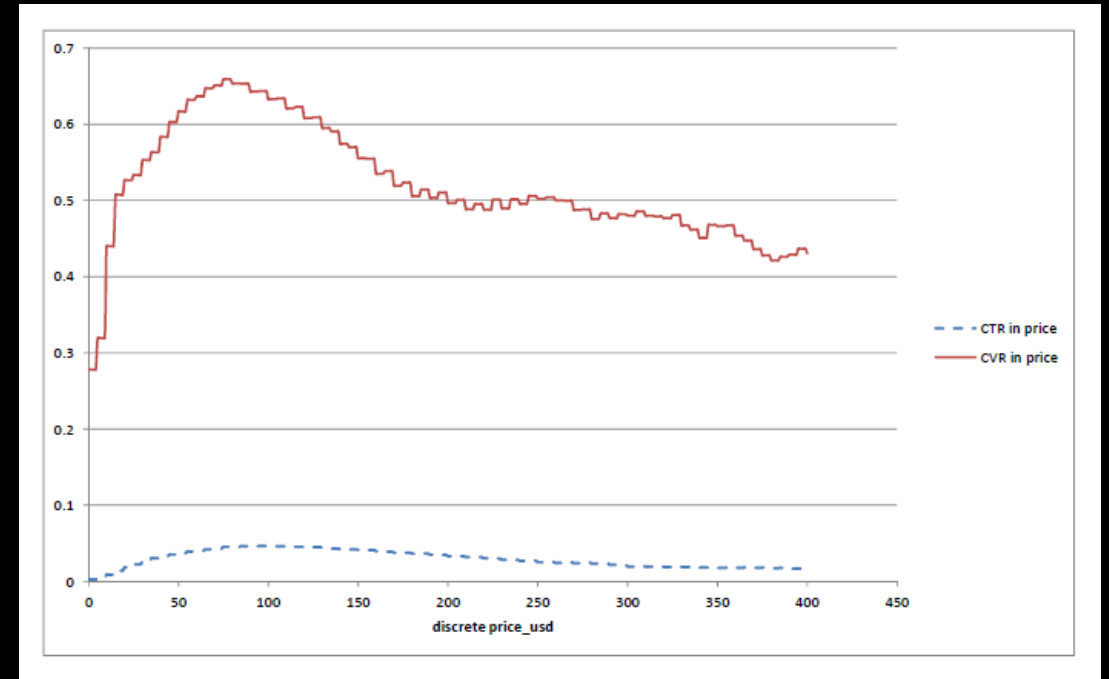
- Score Feature



# FEATURES(CONT.)

- How to define good features?
- How to find good features?
- Good features in our models.

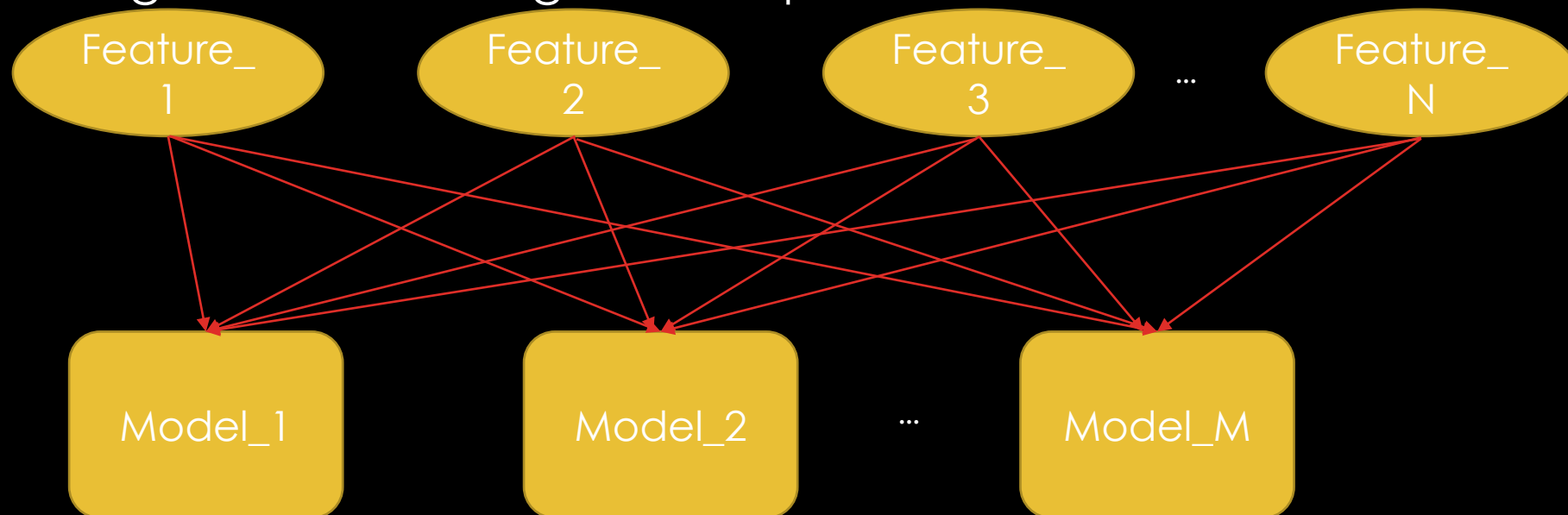
Feature_Name
Price_usd
Prop_location_score1
Prop_location_score2



- 
- Models

# MODELS

- Models should be used in the right way.
- Different features fit different models.
- Background knowledge and experiments are needed.



# MODELS(CONT.)

Pairwise & Listwise Models	NDCG@38 on Validation Set
Pairwise Logistic Regression	0.51
Lambda Mart	0.5243

Pointwise Models	NDCG@38 on Validation Set
Logistic Regression	0.52
Random Forest	0.52
Gradient Boosting Machine	0.52477
Extremely Randomized Trees	0.51699
Factorization Machine	0.5171



# MODELS (CONT.)

Pointwise Models	Features Fitted
Logistic Regression	Bin_ID Feature, Normalized Feature, Listwise Feature
Random Forest and other tree based	All Feature with Listwise Feature and Composite Feature
Factorization Machine	Bin_ID Feature, Location Feature, Listwise Feature
Pairwise Logistic Regression	Bin_ID Feature, Listwise Feature, Normalized Location Feature and Composite Feature
Lambda Mart	Score Feature, Normalized Feature, Listwise Feature

# MODELS (CONT.)

- Factorization Machine
  - A long time to train.
  - A lot of feature engineering work.
  - Listwise features bring good result.
  - Bin or Normalized.

Bin_ID_Feature	Normalized_Feature	Rank_Feature
Prob_id	Price_usd	Price_rank
Srch_destination_id	Prop_location_score1	Price_diff_rank
Srch_room_count	Prop_location_score2	Star_rank

# MODELS (CONT.)

- LambdaMART

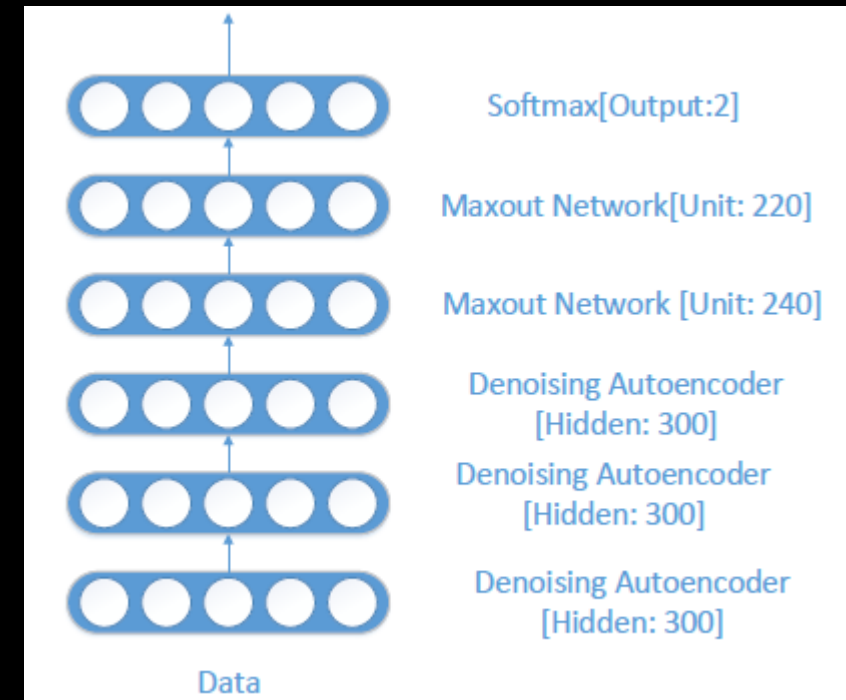
- Listwise model.
- Some features(e.g. time and locations) won't work.
- Score features work well.

Score_Feature	Normalized_Feature	Rank_Feature
Fm_score	Price_usd	Price_rank
LR_score	Prop_location_score1	Price_diff_rank
	Prop_location_score2	Star_rank

# MODELS (CONT.)

- Deep Learning Approach (Failure Case)

1. Softmax fails in the unbalanced case
2. Highly composite features contribute little (similar results found in combination random forest)



- 
- Ensemble models

# ENSEMBLE MODELS

- Why Ensemble?

Single model can't get the best result.

- Key points of Ensemble?

Diverse

Model Diverse

Feature Diverse

- How to Ensemble?

# ENSEMBLE MODELS(CONT.)

- Z-score: simplest but work best

$$Z(x) = \frac{x - \bar{x}}{\sigma(x)} = \frac{x - \bar{x}}{\sqrt{(x - \bar{x})^2/n}}$$

- GBM Ensemble





# ENSEMBLE MODELS (CONT.)

- Deep Learning Ensemble
  - Dropout Logistic Regression (0.527)
- Listwise Ensemble
  - LambdaMart (Final Model, 0.53102)

# TRAINING TIME IN SUMMARY

Single Model	Who	Training Time
Linear Model	Qiang Yan	Minutes to an hour
Random Forest	Bing Xu	4 hours
Factorization Machine	Xudong Liu & Liang Pang	10 hours
GBM/LambdaMART	Yuyu Zhang, Qiang Yan, etc	10 hours

Ensemble Model	Who	Training Time
Dropout Logistic Regression	Bing Xu	minutes
zscore	Yuyu Zhang	minutes
GBM	Liang Pang	An hour
LambdaMART	Qiang Yang	An hour

- 
- Conclusion

# CONCLUSION

- Tree based ranker (Random Forest/GBM/LambdaMart is robust in ranking hotel
- Linear Model is efficient but need more time to do feature engineering
- Deep Learning in pointwise may not be efficient
- Representation Learning is hard in this task by using trivial neural network

# FURTHER INVESTIGATE

- 1. Using Bayesian Database to deal with the missing data
- 2. Deep Neural Network in pairwise and listwise
- 3. Representation learning to solve cold start problem



# Thanks

binghsu & MLRush & BrickMover Team