PCIC 2021: Causal Inference and Recommendation

Alexander Yetta Team

South China University of Technology

0 Team Introduction

Alexander Yetta team.

There is only one member in the team. I am a solo player.

Di He, Master of South China University of Technology, data mining enthusiast, has won the following awards:

1. **Huawei GDE2020 Global Developer Competition: Core Network KPI Anomaly Detection TOP1 (solo);**
2. **Huawei HDC2020 Network Artificial Intelligence Hackathon: Hard Disk Anomaly Detection TOP2 (solo);**
3. **Mango TV's 2nd International Audio and Video Algorithm Invitational: Video Anti-leech Prediction TOP2 (team);**
4. **Baidu AI STUDIO: Anti-fraud prediction of advertising traffic TOP2 (solo);**

and so on.

1 Task Introduction

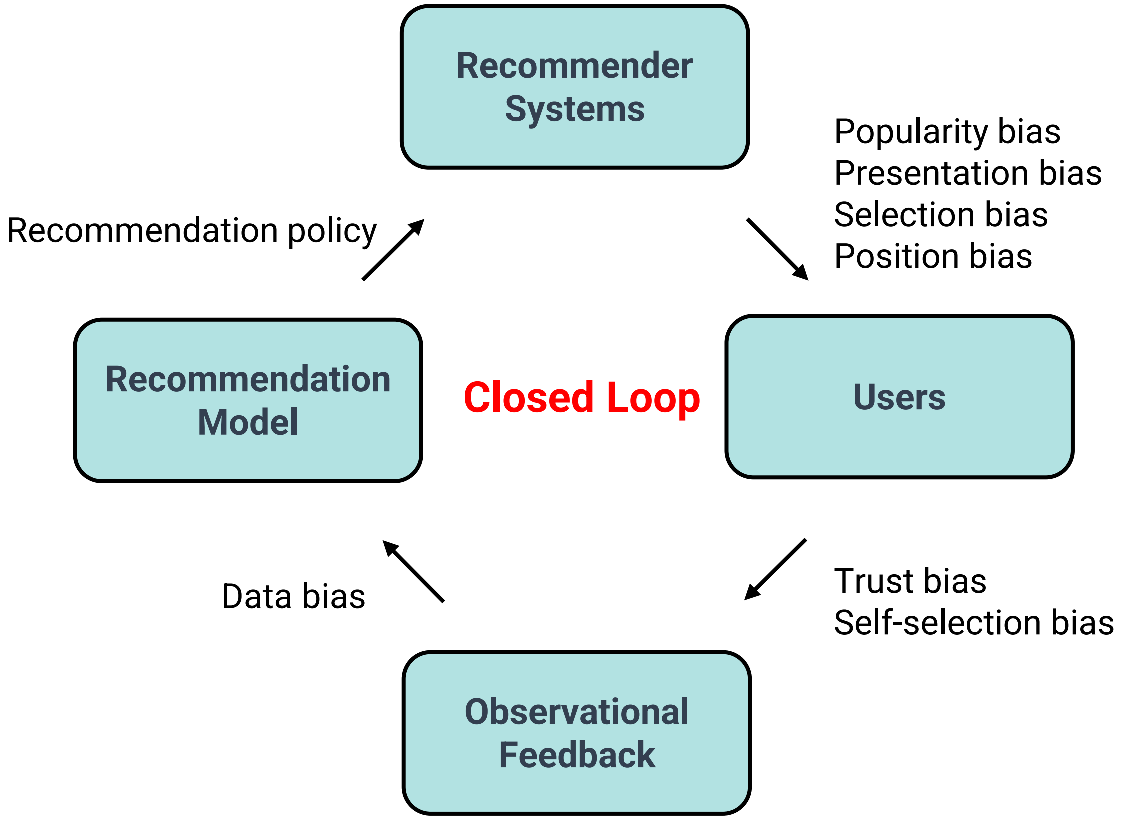
Task:
Estimate a user's preference to a particular movie tag (like or dislike), instead of predicting the rating of a particular user-movie pair.

Given:
1. Rating: ratings for some (user,movie) pairs;
2. Bigtag: observed tags that users labeled to movies;
3. Choicetag: random experiment, similar to Bigtag；
4. Movie: basic information of movies;

Evaluation Metric:
Metric is the AUC (Area Under ROC Curve) between the ground truth and the predictions of the submission. A method of calculating AUC is as follows:

$$AUC = \frac{\sum_{M*N} I\left(P_{pos}, P_{nega}\right)}{M*N}, \quad \text{where } I\left(P_{pos}, P_{nega}\right) = \begin{cases} 1, P_{pos} > P_{nega} \\ 0.5, P_{pos} = P_{nega} \\ 0, P_{pos} < P_{nega} \end{cases}$$

where M is the number of true positive samples, N is the number of true negative samples.



Recommender Systems

Popularity bias
Presentation bias
Selection bias
Position bias

Users

Trust bias
Self-selection bias

Observational Feedback

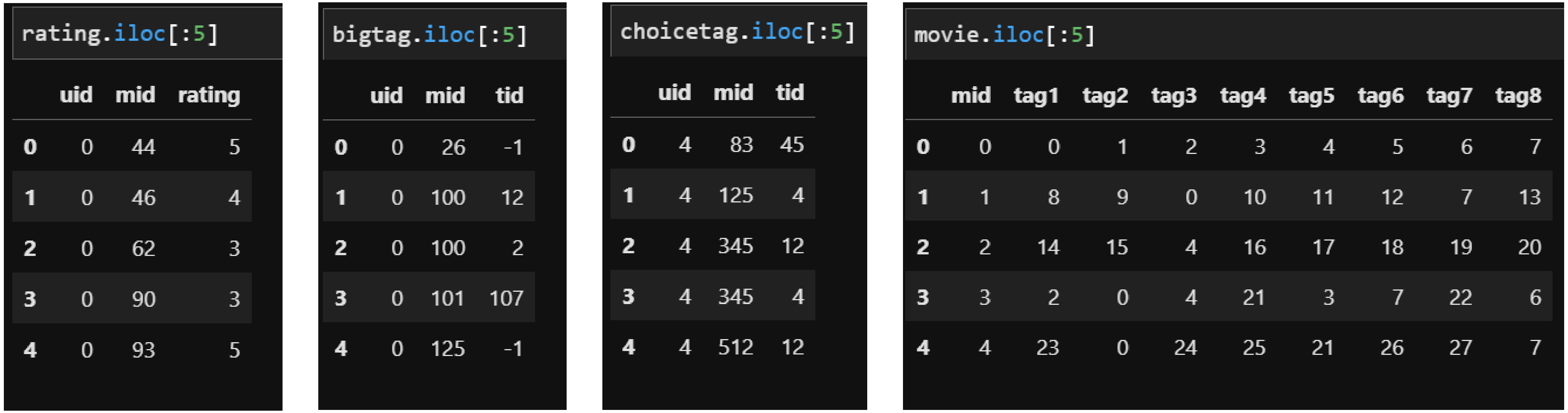
Data bias

Recommendation Model

Recommendation policy

Closed Loop

2 Problem Understanding



```
rating.iloc[:5]
```

	uid	mid	rating
0	0	44	5
1	0	46	4
2	0	62	3
3	0	90	3
4	0	93	5

```
bigtag.iloc[:5]
```

	uid	mid	tid
0	0	26	-1
1	0	100	12
2	0	100	2
3	0	101	107
4	0	125	-1

```
choicetag.iloc[:5]
```

	uid	mid	tid
0	4	83	45
1	4	125	4
2	4	345	12
3	4	345	4
4	4	512	12

```
movie.iloc[:5]
```

	mid	tag1	tag2	tag3	tag4	tag5	tag6	tag7	tag8
0	0	0	1	2	3	4	5	6	7
1	1	8	9	0	10	11	12	7	13
2	2	14	15	4	16	17	18	19	20
3	3	2	0	4	21	3	7	22	6
4	4	23	0	24	25	21	26	27	7

Fig.1 Sample Data (where uid is userid, mid is movieid, tid is tagid)

Essence of the problem:

In this task, the user's preferences for some movies are known, and some tags that users labeled to movies are known.

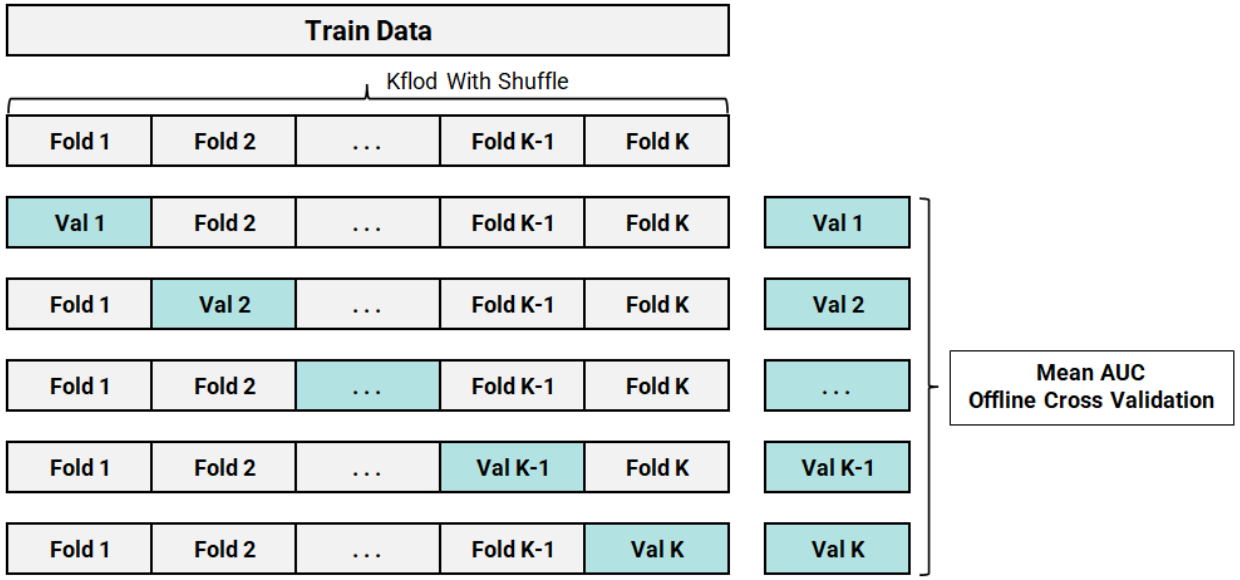
The task is to evaluate users’ preferences for unknown movie tags.

Therefore, we can deal with this problem from the following aspects:

1. Information of users;
2. Information of tags;
3. Information of user-tag pairs;

3 Solution Summary

3.1 Offline Validation



Train Data

Kflod With Shuffle

Fold 1 | Fold 2 | . . . | Fold K-1 | Fold K

Val 1 | Fold 2 | . . . | Fold K-1 | Fold K — Val 1

Fold 1 | Val 2 | . . . | Fold K-1 | Fold K — Val 2

Fold 1 | Fold 2 | . . . | Fold K-1 | Fold K — . . .

Fold 1 | Fold 2 | . . . | Val K-1 | Fold K — Val K-1

Fold 1 | Fold 2 | . . . | Fold K-1 | Val K — Val K

Mean AUC
Offline Cross Validation

Fig. 2 offline validation scheme

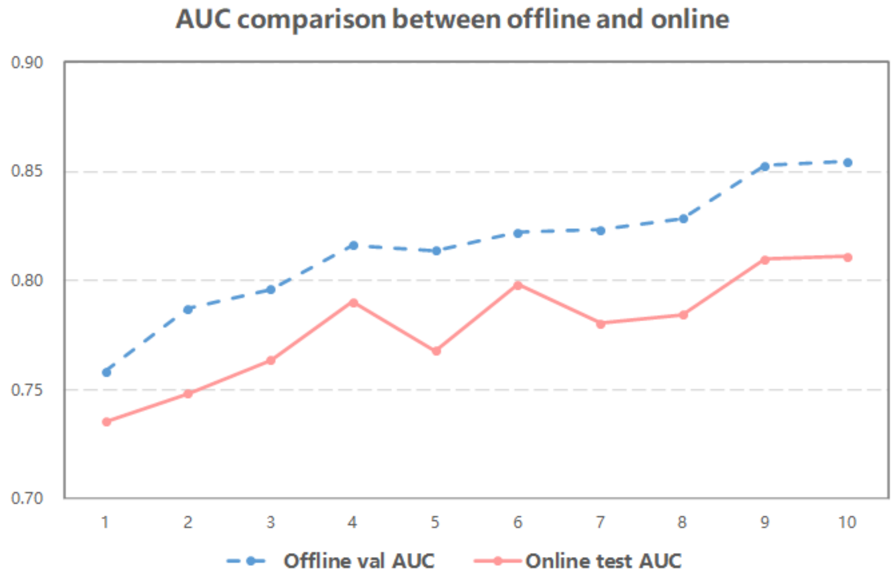


Fig. 3 AUC comparison between offline and online (Phase 1)

In order to make full use of training data and obtain more reliable offline validation, cross-validation with shuffling is adopted as the local validation scheme. The process is shown in Fig.2.

Fig.3 is the comparison of online and offline scores in phase 1. Obviously, this is a reliable offline validation solution. When the AUC of offline cross-validation increases by more than 0.005, there is good consistency between online and offline. Therefore, modeling experiments with offline AUC improvement less than 0.005 can be considered as unreliable random improvement. And almost all modeling experiments with an improvement of less than 0.005 will not be used in the final solution. This has brought a good generalization ability to my modeling scheme, and in the end my scheme is ranked 1st in both phase 1 and phase 2.

3 Solution Summary

3.2 Feature Engineering

As mentioned in the section 2: Problem Understanding. Feature engineering can be performed on the information from users, tags, and user-tag pairs.

3.2.1 Information of users

I tried to construct a lot of features from the user's perspective, such as:
the max, min, median, count, nunique, and std of each user’s rating of the movies,
the total number of tags or movies labeled by each user, etc.
However, all the features or feature sets of user information cannot bring the improvement that can pass offline verification (local cross validation score increased by more than 0.005). Perhaps due to the existence of bias, user-related features cannot improve the performance of the model, and these features may even reduce the performance of the model. Therefore, my final solution does not contain any variables constructed only by users.

3.2.2 Information of tags

I constructed tag variables such as:
the total number of movies for each tag,
the total number of movies tagged in a specific location such as tag1 or tag2,
the total number of tags for each tag by the user, etc.
The features of the tags is very useful and is the core variables of the final model.

3 Solution Summary

3.2 Feature Engineering

3.2.3 Information of user-tag pairs

The bigtag data set is a data set that directly contains user-tag pairs. As shown in Fig.4, if a user has labeled a certain tag of a movie, it means that the user is interested in this tag.

However, there are very few user-tag pairs in the bigtag data set that match the sample to be predicted. Only **4.02%** of the user-tag pairs in the training set exist in the bigtag data set, and the proportion in the test dataset of Phase 1 is only **1.98%**, the proportion in the test dataset of Phase 2 is **even 0%.**

Therefore, **how to obtain more user-tag information is of great significance.**

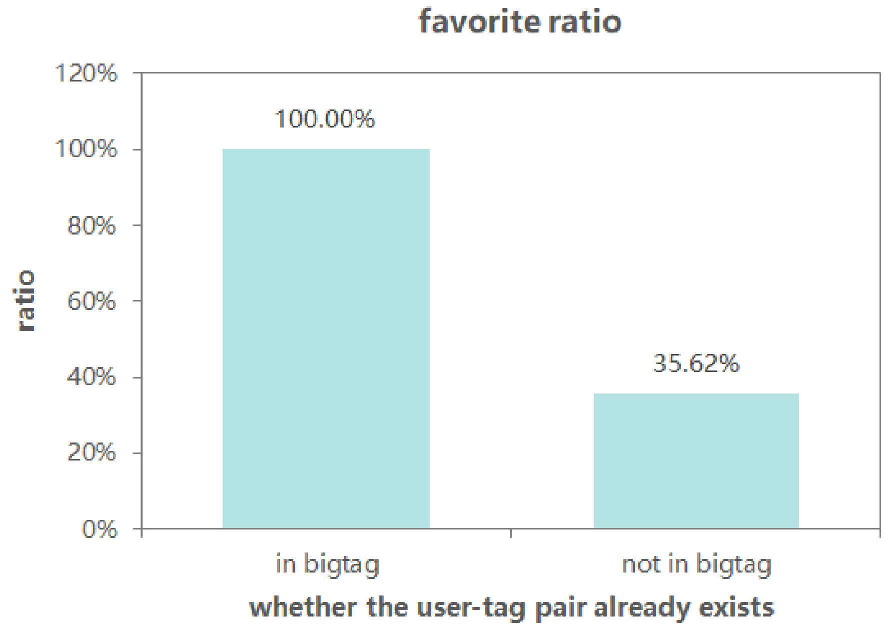


Fig. 4 the favorite ratio of existing user-tag pairs that already existed in bigtag dataset

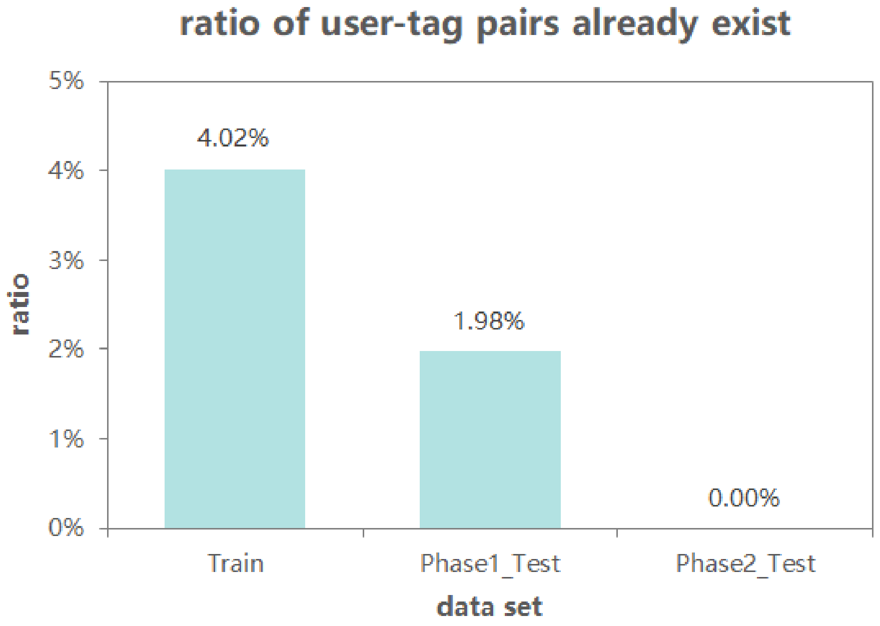


Fig.5 ratio of user-tag pairs already exist in test

3 Solution Summary

3.2 Feature Engineering

3.2.3 Information of user-tag pairs

Although the tags labeled by users are very limited. But based on the behaviors tagged by users, the affiliation relationship between movies and tags, and the ratings that users give to movies, we can construct a huge graph network correlation relationship (Fig.6). As shown in Fig.7, based on the graph network, we can construct various paths from a user to various tags. By making these paths into features, we can get a lot of user-tag pairs information. These variables are the **most critical part** of my solution.

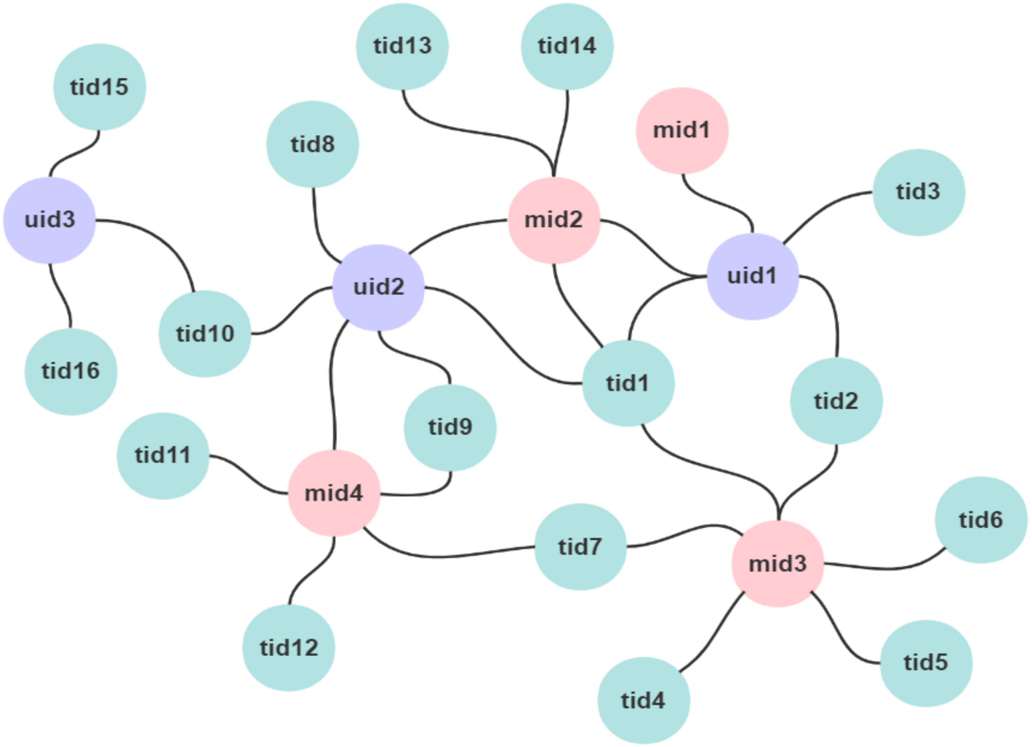


Fig.6 graph network of users, movies, and tags

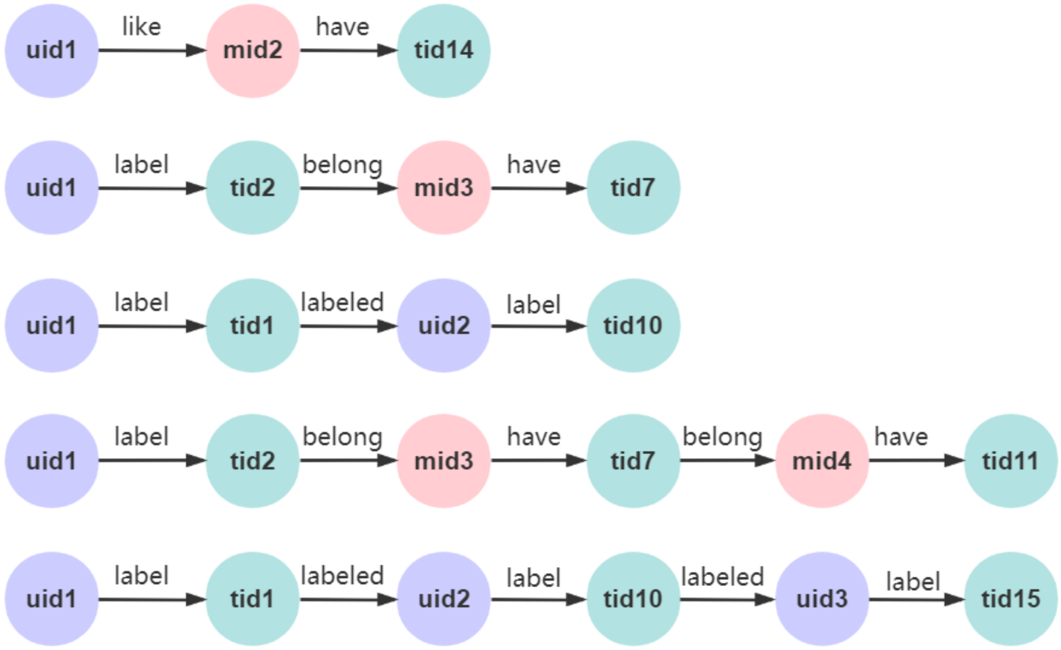

uid1 like mid2 have tid14
uid1 label tid2 belong mid3 have tid7
uid1 label tid1 labeled uid2 label tid10
uid1 label tid2 belong mid3 have tid7 belong mid4 have tid11
uid1 label tid1 labeled uid2 label tid10 labeled uid3 label tid15

Fig.7 path of user to other tags

3 Solution Summary

3.2 Feature Engineering

3.2.4 Feature selection

No additional feature selection is required

Since I have a robust local cross-validation scheme. And in the process of exploring data, I will construct a set of features based on a same idea to conduct data experiments. Since I have a lot of discoveries and ideas, I conducted a lot of data experiments in the process of exploring data. Once the idea fails local cross-validation, I will delete the idea and all features related to the idea. Therefore, when modeling, the variables that the model can use are all useful variables, and no additional feature screening is required.

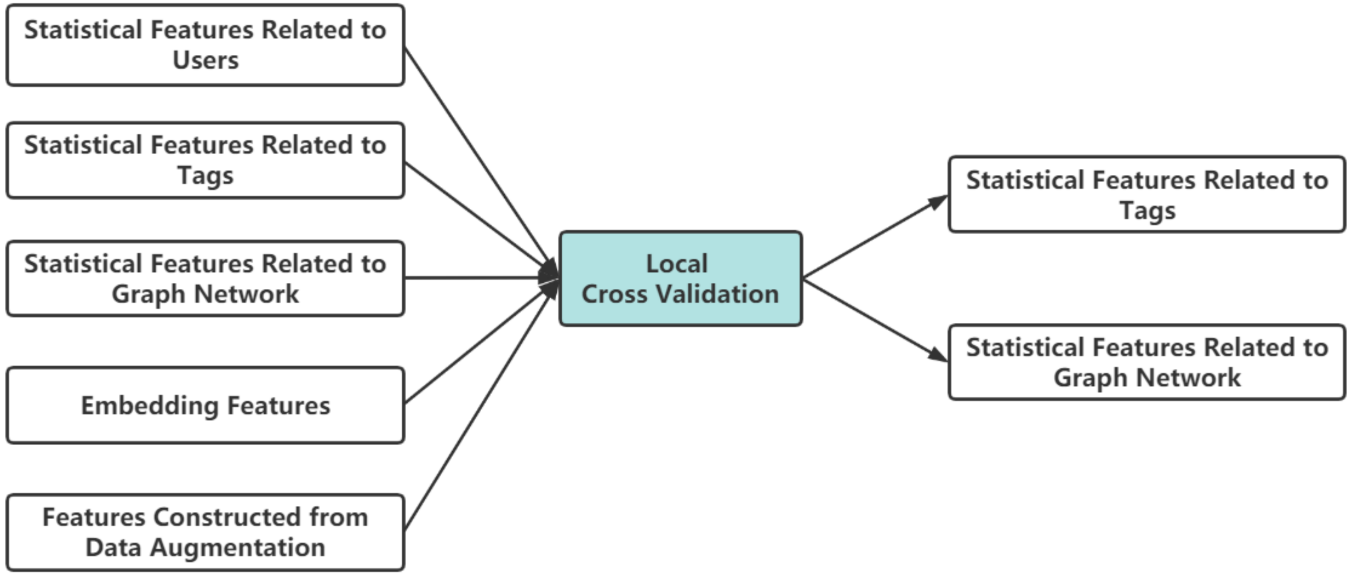

Statistical Features Related to Users
Statistical Features Related to Tags
Statistical Features Related to Graph Network
Embedding Features
Features Constructed from Data Augmentation
Local Cross Validation
Statistical Features Related to Tags
Statistical Features Related to Graph Network

Fig.8 Feature engineering scheme

3 Solution Summary

3.3 Model Selection

GBDT (Gradient Boosting Decision Tree) is a boosting ensemble learning algorithm based on decision tree. The excellent algorithm framework based on GBDT includes XGBoost[1] developed by Tianqi Chen at the University of Washington, CatBoost[2] developed by Yandex, LightGBM[3] developed by Microsoft, etc.

Due to the advantages of fast training speed, high fitting accuracy, and strong generalization ability, LightGBM has been widely used in various structured data tasks after being open sourced in 2017, and achieved excellent performance:

IJCAI 2018 Alimama International Advertising Algorithm Competition
Rank1: https://github.com/plantsgo/ijcai-2018
Rank2: https://github.com/YouChouNoBB/ijcai-18-top2-single-mole-solution
Rank3: https://github.com/luoda888/2018-IJCAI-top3

WSDM 2018 KKBox's Music Recommendation Challenge
Rank1: https://github.com/lystdo/codes-for-wsdm-cup-music-rec-1st-place-solution

KDD Cup 2020 Challenges for Modern E-Commerce Platform: Debiasing
Rank1: https://github.com/aister2020/KDDCUP_2020_Debiasing_1st_Place

PAKDD 2021 2nd Alibaba Cloud AIOps Competition
Rank1: https://github.com/ji1ai1/202101-PAKDD2021

The LightGBM algorithm framework based on GBDT is also adopted by my solution.

[1] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, 2016.
[2] Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin "CatBoost: gradient boosting with categorical features support". Workshop on ML Systems at NIPS 2017.
[3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 3149-3157.

3 Solution Summary

3.4 Model Framework

The final model framework of my solution is shown in Figure 9. The framework is very simple.

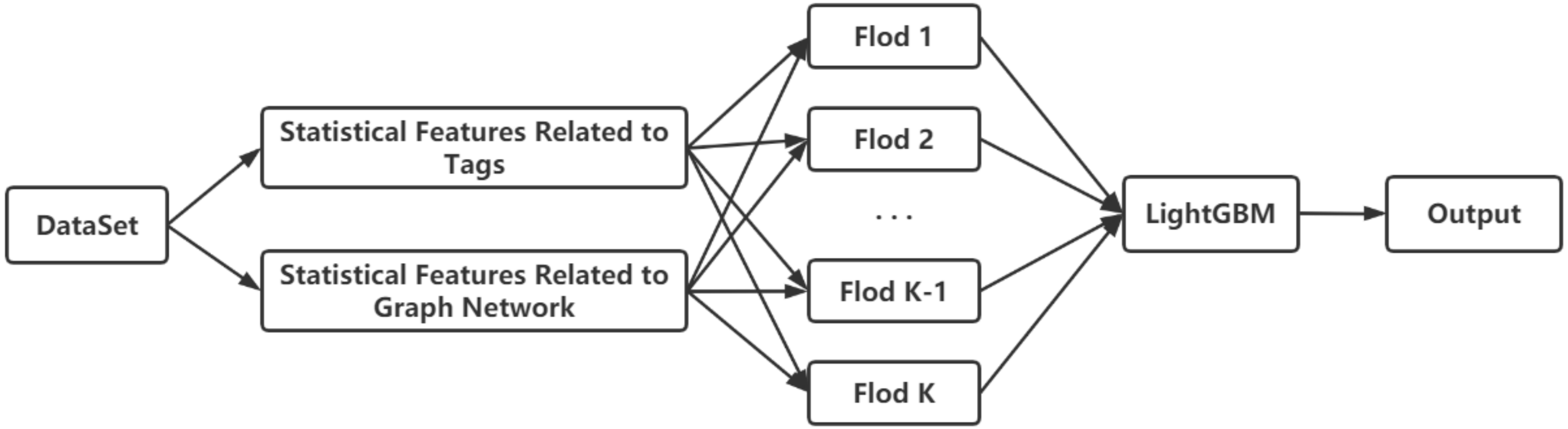

DataSet
Statistical Features Related to Tags
Statistical Features Related to Graph Network
Flod 1
Flod 2
. . .
Flod K-1
Flod K
LightGBM
Output

Fig.9 Model Framework

3 Solution Summary

3.5 Feature Importance

Fig.10 shows the importance of the Top 20 features of the model.

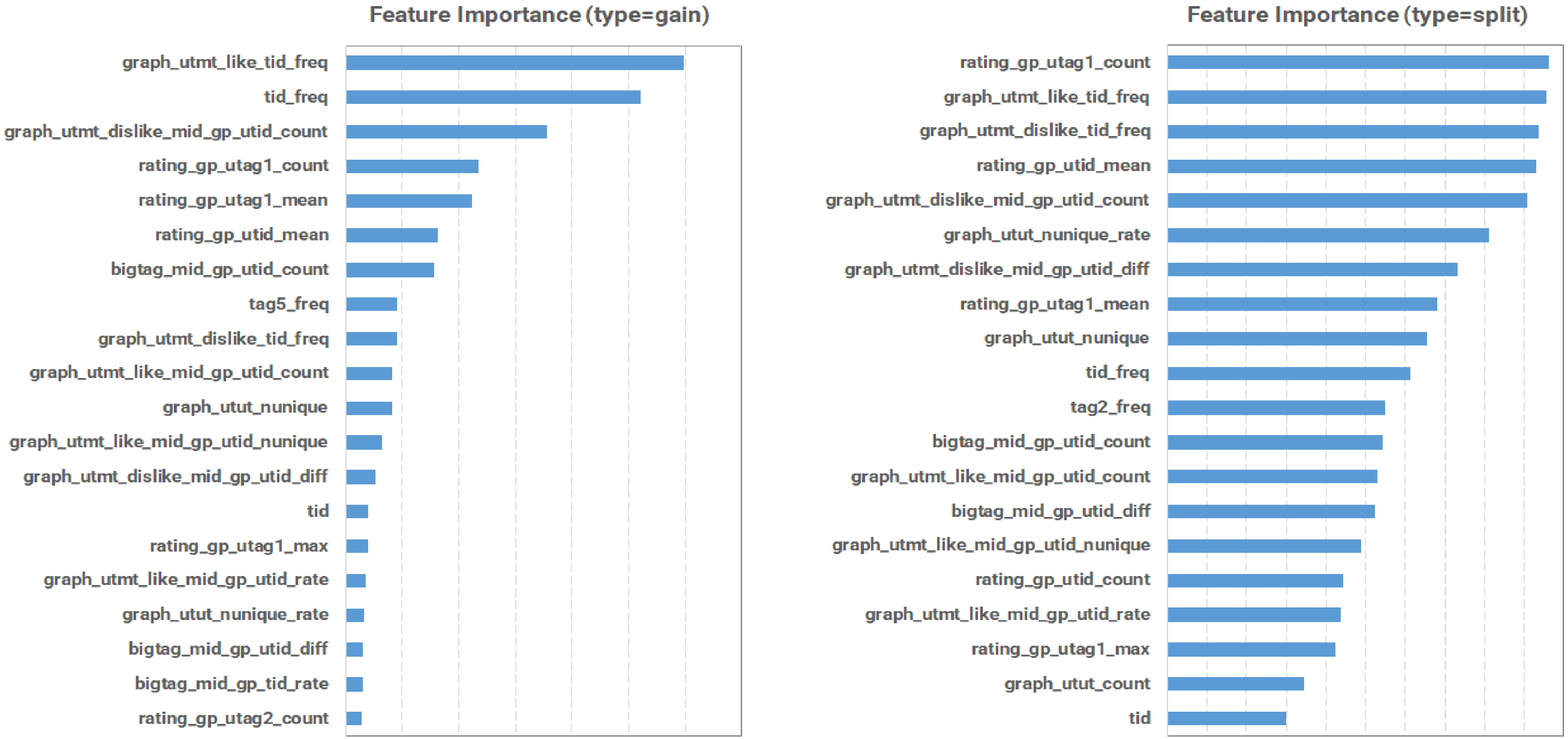
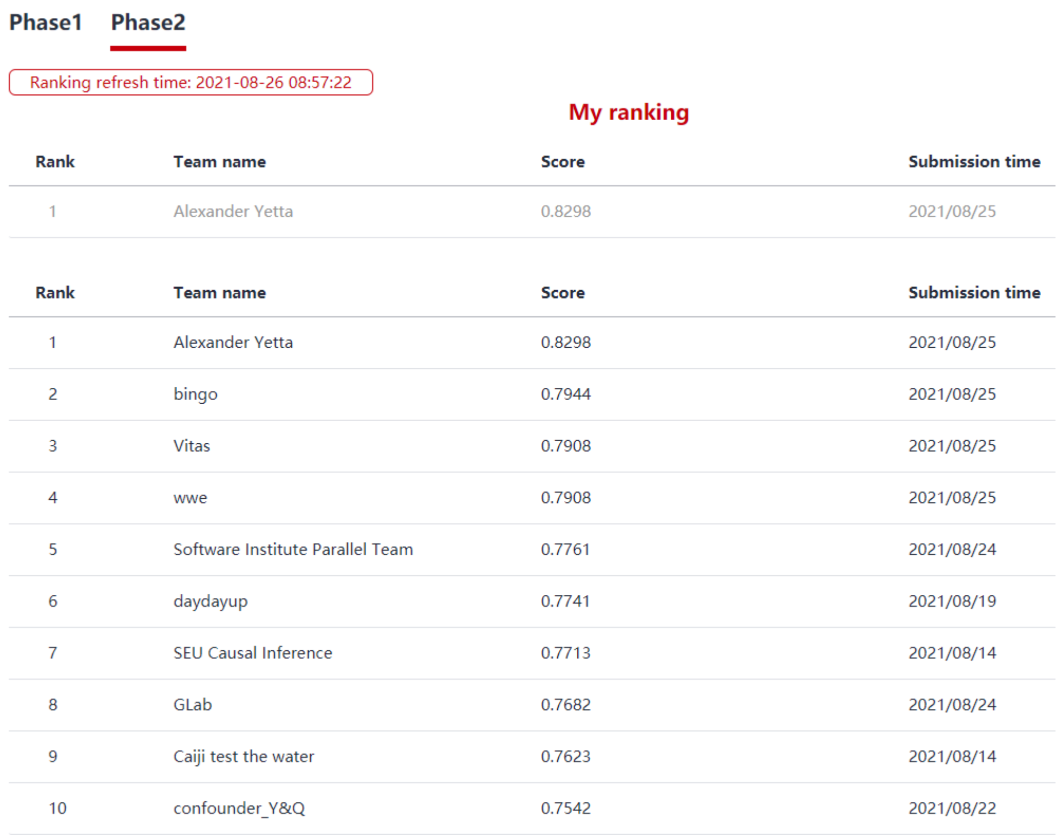


Fig.10 Feature Importance

4 Achieved Result

My solution got rank1 in both phase 1 and phase 2 of this task. And the final performance in the test set is far ahead of other teams.



Phase1 Phase2

Ranking refresh time: 2021-08-26 08:57:22

My ranking

Rank	Team name	Score	Submission time
1	Alexander Yetta	0.8298	2021/08/25

Rank	Team name	Score	Submission time
1	Alexander Yetta	0.8298	2021/08/25
2	bingo	0.7944	2021/08/25
3	Vitas	0.7908	2021/08/25
4	wwe	0.7908	2021/08/25
5	Software Institute Parallel Team	0.7761	2021/08/24
6	daydayup	0.7741	2021/08/19
7	SEU Causal Inference	0.7713	2021/08/14
8	GLab	0.7682	2021/08/24
9	Caiji test the water	0.7623	2021/08/14
10	confounder_Y&Q	0.7542	2021/08/22

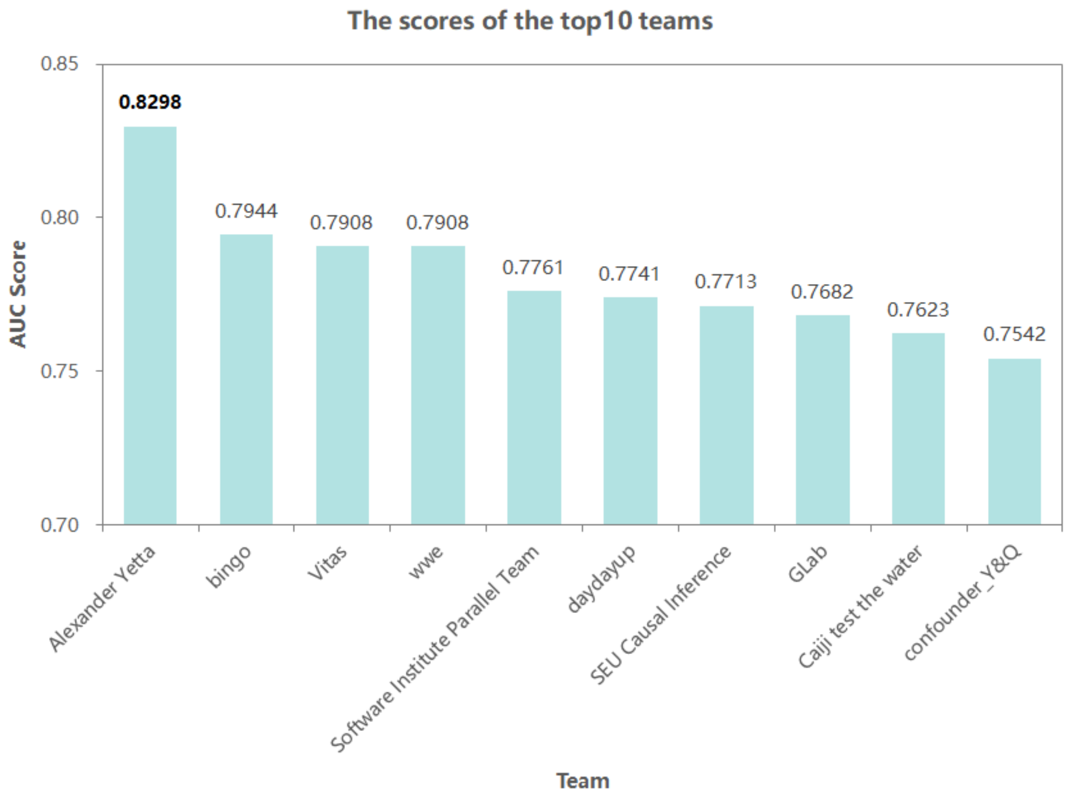


Fig.11 The final result of the task