WSM FINAL PROJECT

OTTO – MULTI-OBJECTIVE RECOMMENDER SYSTEM

Group name: WSM_Panda

109703027 資科三 范佳琦
109703037 資科三 田展禾
109703060 資科三 盧奕潔
109208064 資科三 俞　懿
110971005 資專二 許博堯

Contents

1

TF-IDF

一種常用於資訊檢索與文字探勘的統計方法，
用來 評估「詞」對於「文件」的重要程度

WHAT IS PARQUET

CSV

- 以資料列為導向的儲存方式
- 透過**index**取得相關資料
- 適合 **Web-based system, APP**

Id	Name	Role
1	Anakin	Darth Vader
2	R2D2	Robot
3	Yoda	Jedi Knight

Parquet

- 適合資料分析
- 資料量大時，用**parquet**會更有效率

1	2	3
Anakin	R2D2	Yoda
Darth Vader	Robot	Jedi Knight

GROUPBY

Session aid

```
0           [1517085, 1563459, 1309446, 16246, 1781822, 11...
1           [424964, 1492293, 1492293, 910862, 910862, 149...
2           [763743, 137492, 504789, 137492, 795863, 37834...
3           [1425967, 1425967, 1343406, 1343406, 1343406, ...
4           [613619, 298827, 298827, 383828, 255379, 18381...
                                   ...
12899774                                     [33035, 1399483]
12899775                                   [1743151, 1760714]
12899776                                    [548599, 1737908]
12899777                                     [384045, 384045]
12899778                                      [561560, 32070]
Name: aid, Length: 12899779, dtype: object
```

GENSIM TF-IDF MODEL

1 ▪

```
dct = Dictionary(df_sess_split)  # fit dictionary

corpus = [dct.doc2bow(line) for line in df_sess_split] # convert corpus to BoW format

tfidf_model = TfidfModel(corpus)  # fit model
```

2 ▪ Transform the test set with the dictionary fitted by training df.

3 ▪ Sorted aids by tfidf scores.

Submission

- Find the top 20 popular aids.
- Recommend the aids which has been clicked, carted, or ordered by every session. If less than 20, then fill up with popular aids.
- The click, cart, and order of each session are the same.

SCORE: 0.483

2

XGBRanker

Motivation

1. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance
2. Dominates many Kaggle competitions
3. Scikit-Learn API → easy to use
    - XGBRegressor
    - XGBClassifier
    - XGBRanker
4. Support GPU

STEPS

Feature engineering → Train test split → Model training → Rank the items

Features

	session	aid	ts	type
0	0	1517085	1659304800	0
1	0	1563459	1659304904	0
2	0	1309446	1659367439	0
3	0	16246	1659367719	0
4	0	1781822	1659367871	0
...	...	...	...	...
271	0	843110	1661684298	0
272	0	938007	1661684355	0
273	0	1228848	1661684528	0
274	0	1740927	1661684942	0
275	0	161938	1661684983	0

- Features group by session
  - viewed_aid
  - click_cnt
  - cart_cnt
  - order_cnt
  - monday_action_cnt
  - tuesday_action_cnt
  - ...
  - sunday_action_cnt
  - evening_action_cnt
- Features group by aid
  - viewed_session
  - clicked_cnt
  - carted_cnt
  - ordered_cnt
  - monday_action_cnt
  - tuesday_action_cnt
  - ...
  - sunday_action_cnt
  - evening_action_cnt

Model Training

```
from xgboost import XGBRanker

model = XGBRanker(objective='rank:ndcg', n_estimators=100, random_state=0,learning_rate=0.1)
model.fit(
    X_train,
    y_train,
    group=query_list_train,
    eval_metric='ndcg',
    eval_set=[(X_test, y_test)],
    eval_group=[list(query_list_test)],
    verbose =10
)
```

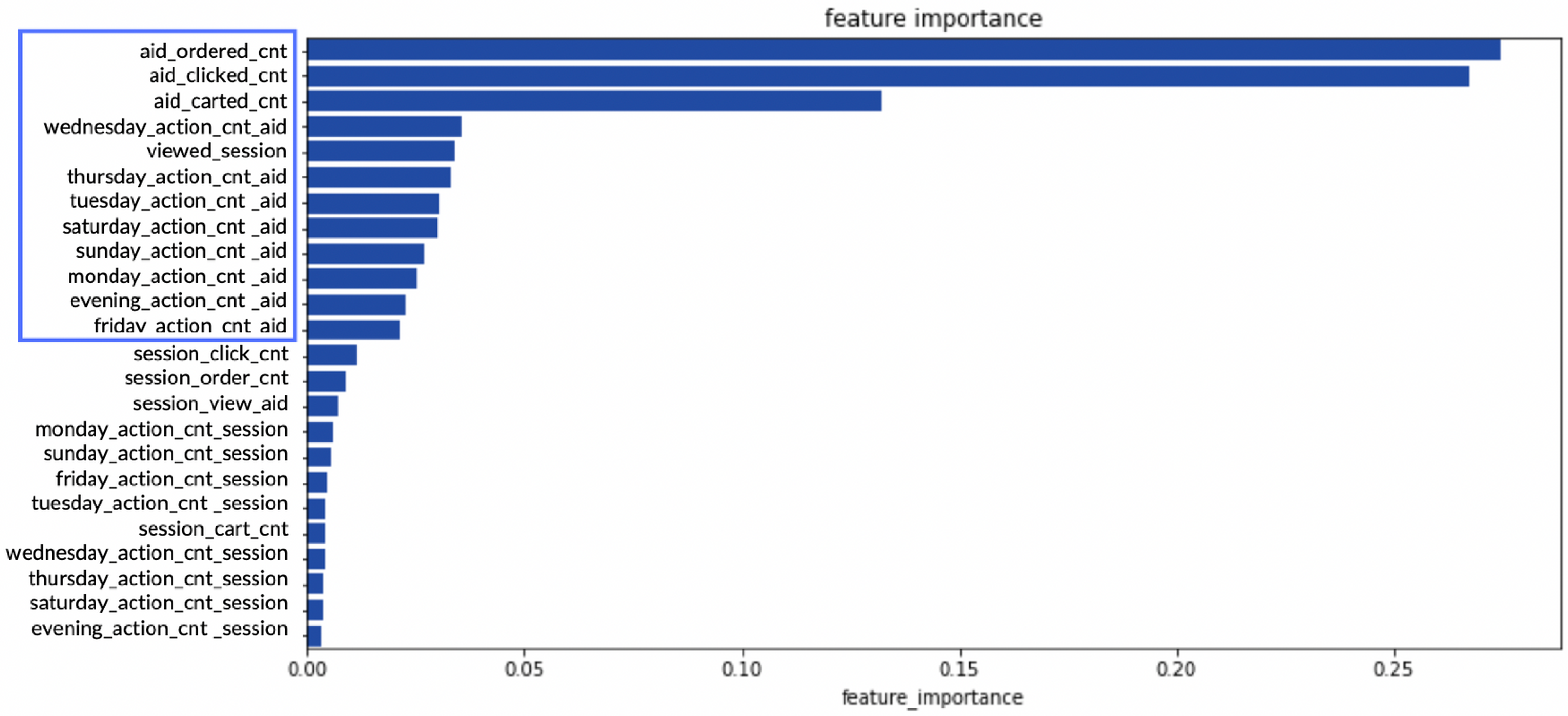
Query group information is required for ranking tasks

For example, if your original data look like:

qid	label	features
1	0	x_1
1	1	x_2
1	0	x_3
2	0	x_4
2	1	x_5
2	1	x_6
2	1	x_7

then *fit* method can be called with either *group* array as [3, 4]

Feature Importance



Rank the Aid in Test Set

Original order:
itemA, itemB, itemC, itemD, itemE, itemF

↓

XGBRanker

↓

Ranked order:
itemB, itemC, itemD, itemA, itemE, itemF

Submit click, cart, order with the same ranked items

Score: 0.462

Conclusion

- Item related features are more important than user related features (in my case).
- The features are not informative enough to capture the patterns of the users.
- Did not generate candidate items for sessions whose number of item <20, only rank the existed items.

3

Word2Vec

STEPS

1. Grab the aids of train and test in session units, and organize them into a two-dimensional list as training data
2. Use the training data as parameters to train gensim.Word2Vec
3. Use annoy to look for nearest neighbors in the embedding space
4. Grab the most recent aid of each session and look their top 20 neighbors

DETAILS

Word2Vec

Training algorithm：CBOW

ItemA, ItemB, ______ , ItemD, ItemE

- 通過前後**aid**來預測當前值
- 訓練速度較**Skip-gram**快

最終每個**aid**皆有一個陣列，可用來表示和其他**aid**之間的關係

Annoy

AnnoyIndex

- **nearest neigbor search**
- **Euclidean distance**
- 改善**gensim**內建的**.most_similar()**太慢的問題

Conclusion

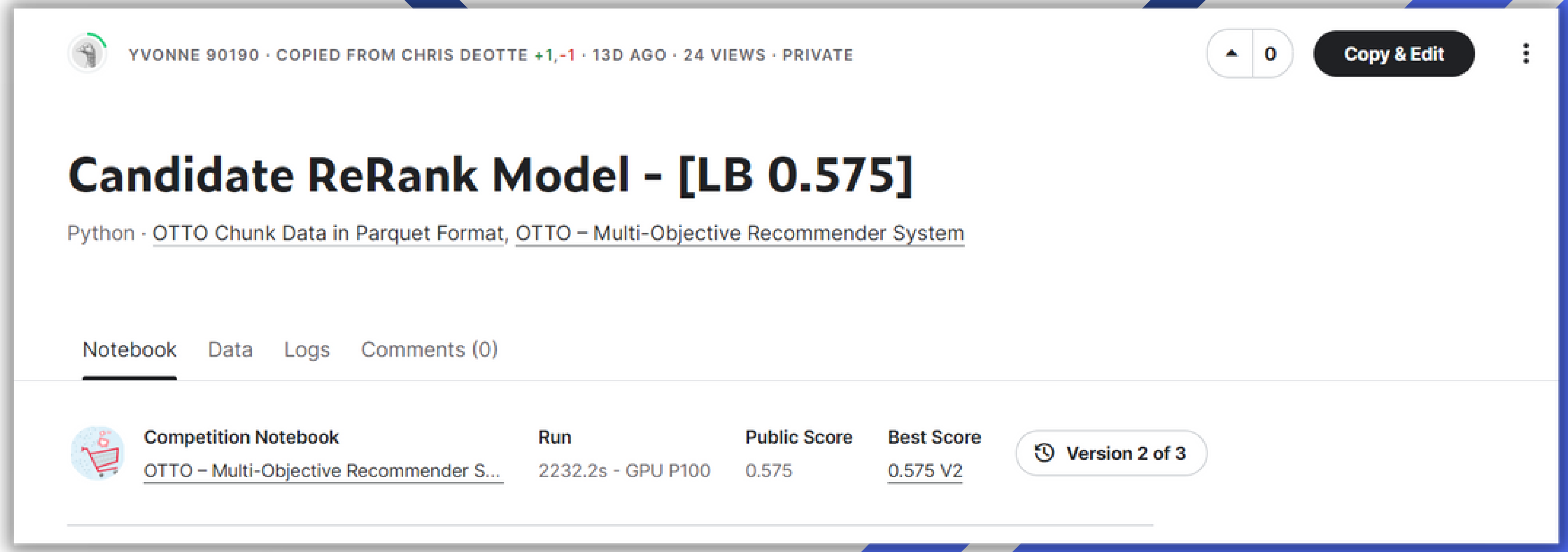
- The scores of click, cart, order are the same because session type is not considered
- Because the context is considered, the score is relatively improved
- The final score is 0.521

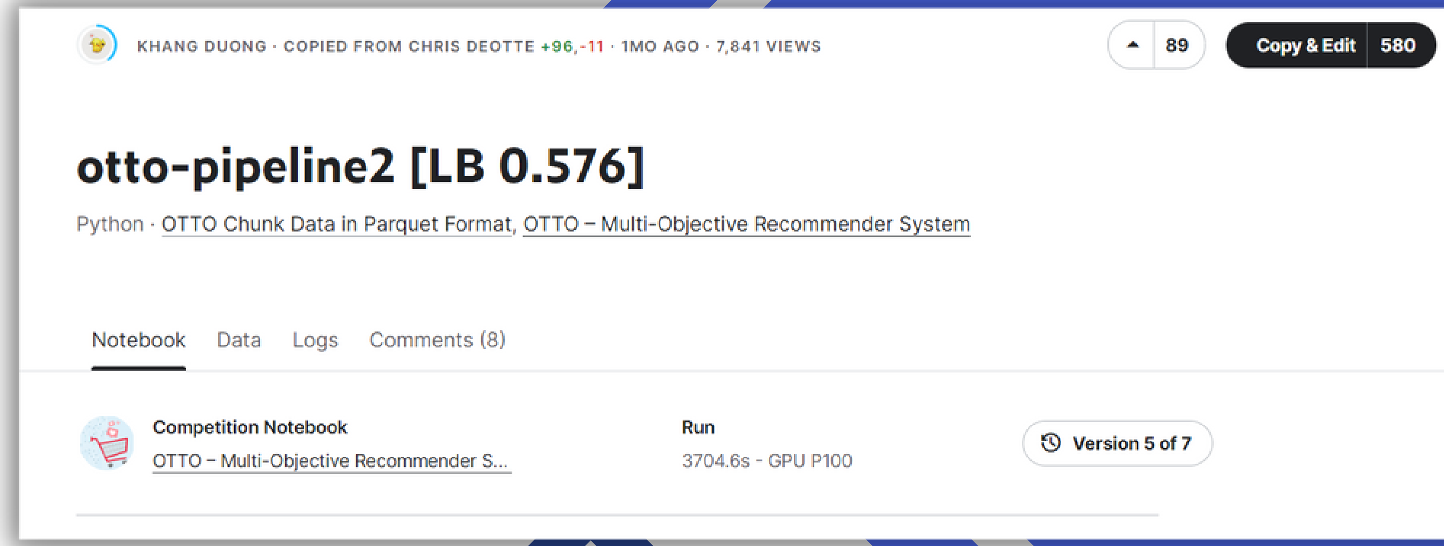
4

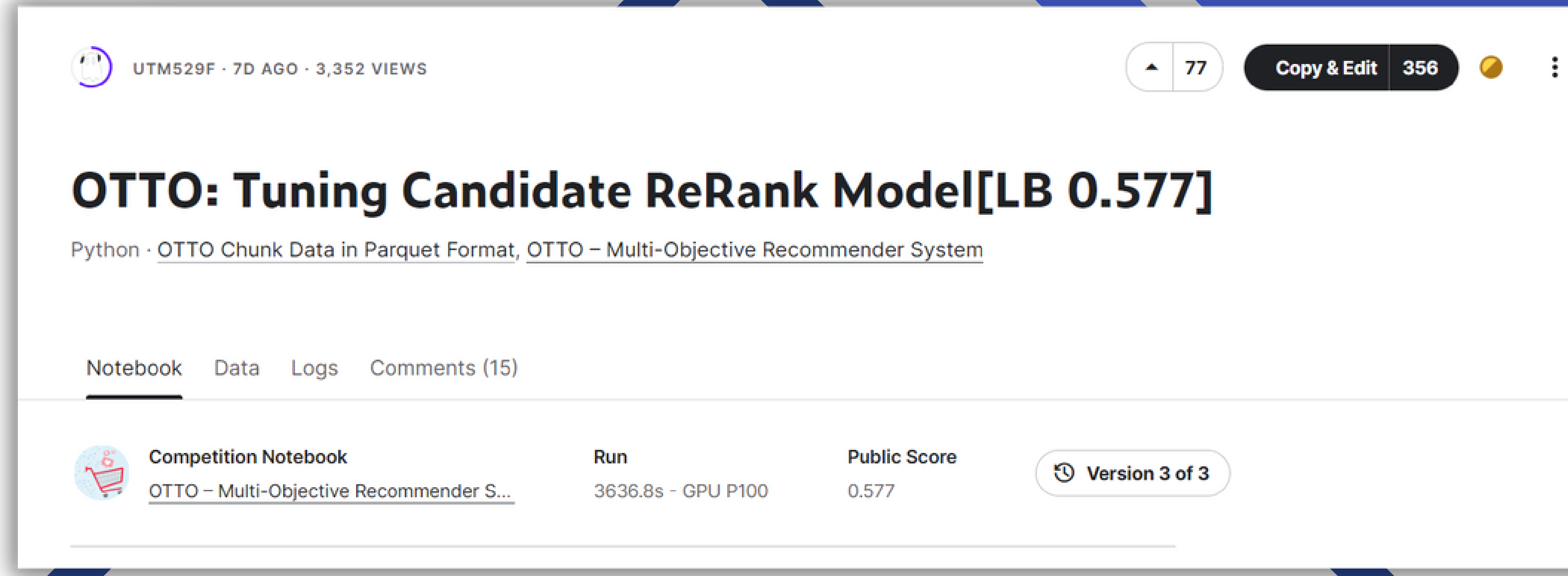
Conclusion

ENSEMBLE

- Combine the results of 3 public notebooks
- Take session_type as a unit, and vote for each aid
- The weight of votes is 0.6, 0.8, 1 according to the score of the notebooks
- According to the sum of votes, re-assign the session_type with a new order of aids







THANK YOU FOR LISTENING!