Data Analytics
KKTV Term Project
[IMUseless]

資管三 陳瑾叡 B09705016
資管三 吳驊祐 B09705009

Outline

1. Overview
2. Preprocessing
3. Model
4. Results
5. Conclusion
6. Future work
7. Reference

Overview

Overview

	user_id	device_id	session_id	title_id	event_time	played_duration	action_trigger	platform	title_in_simulcast	internet_connection_type
0	0	525	2328	384	1.648950e+09	1361	1	0	0	1
1	0	525	2328	384	1.648950e+09	2	0	0	0	1
2	0	525	2400	68	1.648952e+09	2	0	0	0	1
3	0	525	2400	68	1.648952e+09	20	9	0	0	1
4	0	532	2401	68	1.648952e+09	8	10	2	1	4
...	...	...	...	...	...	...	...	...	...	...
9714098	30459	139403	4218064	113	1.663082e+09	2157	1	0	0	1
9714099	30459	139403	4220179	113	1.663156e+09	2743	1	0	0	1
9714100	30459	139403	4220179	113	1.663159e+09	2723	1	0	0	1
9714101	30459	139403	4222301	113	1.663242e+09	2699	1	0	0	1
9714102	30459	139403	4222301	10	1.663242e+09	3	0	0	1	1

9714103 rows × 10 columns

Overview

Our Goal is to find the pattern of the user using these features

- Place more effort on duration, separate them to weeks, days or pure slots
- The other feature may contain useful information, we have to try it out.
- It’s a time series problem
- The amount of the data is affordable **(9714103 rows × 10 columns)**

Overview

Before entering the complete dataset, we use lstm to evaluate the light

dataset as a baseline, so as to ensure that if the subsequent preprocessing to the “duration” feature is more effective or not

The result baseline AUC score on light-LSTM is **0.80311**

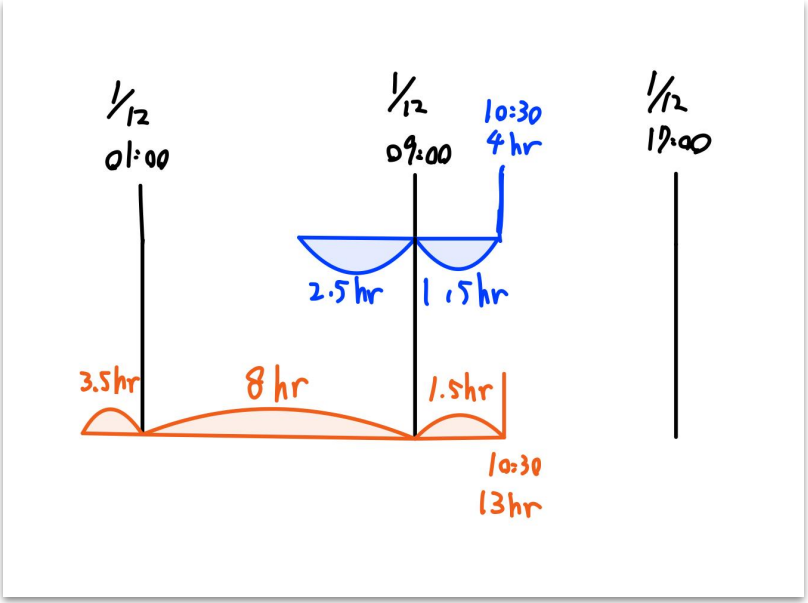
Preprocessing

Preprocess - duration

Since the pre-defined time slot has divided each day to 8, 8, 4, 4 hours

We follow the format and convert each user, everyday's duration to the corresponding slot.

If the duration is too long, we add the rest to previous slots.



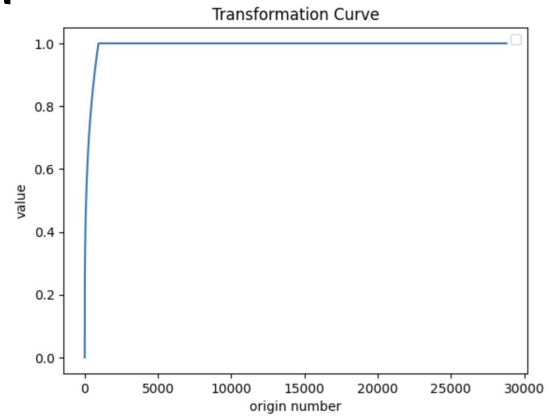
Preprocess - cutting slots

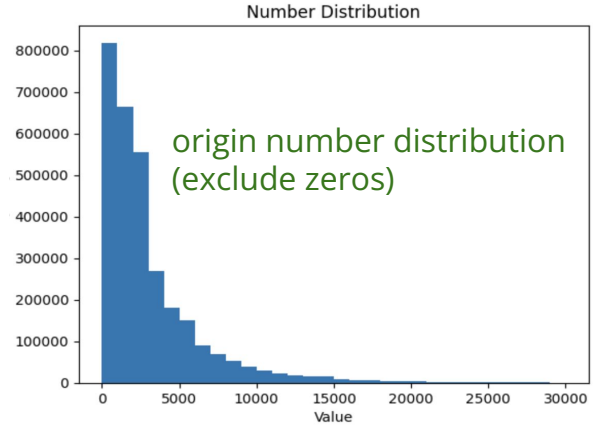
To this point, we've created a new dataset with shape like light dataset, there're 2 things left to do create a complete duration dataset.

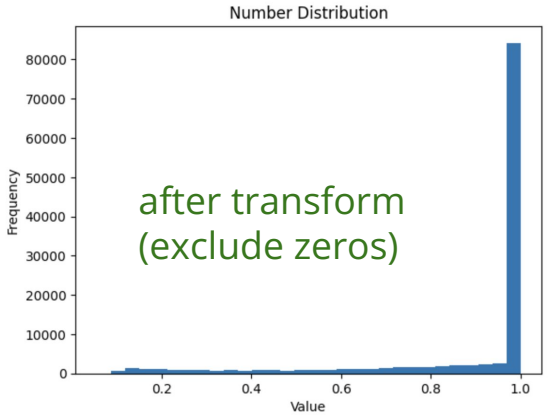
- Conduct different conversion function on seconds to create a duration dataframe with all the values between 0 and 1
- Decide the size of our input, means that how many slot is in an iteration

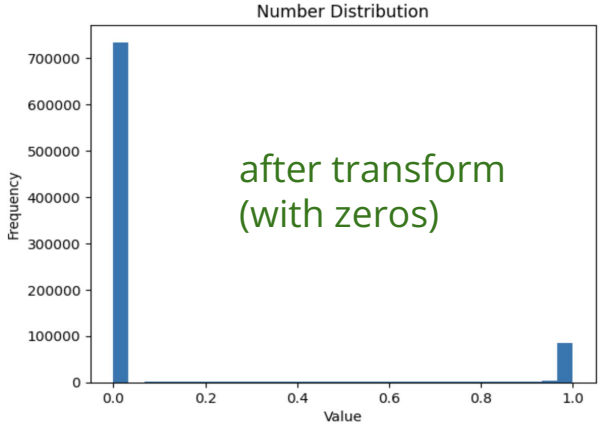
preprocess-duration

Conversion function: min(n0.3 / (28800**0.2),1)**









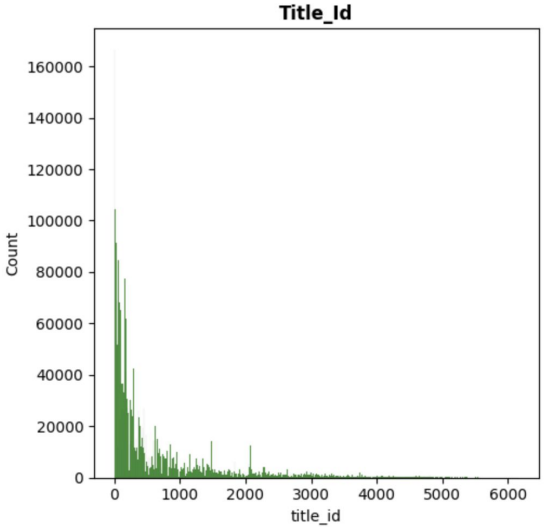
preprocess - duration (week, day, slot)

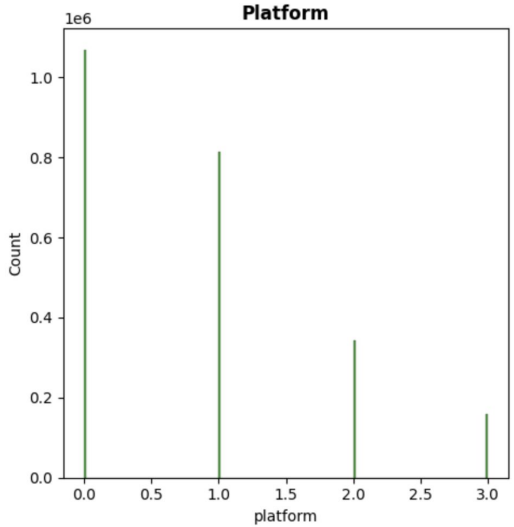
We tried different ways to divide the 1036 time slots, and feed them to LSTM

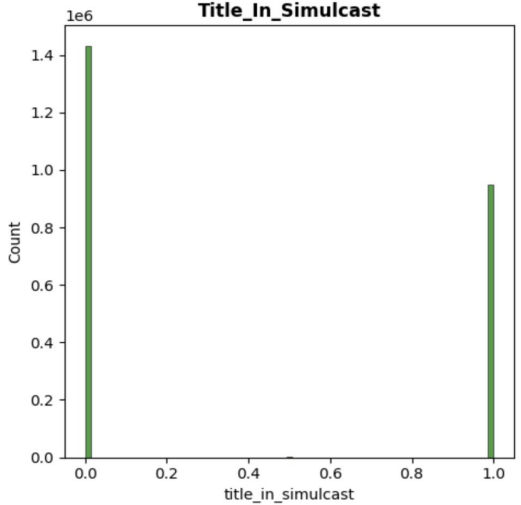
(input_size, sequence_length) = (1036, 1), (28, 37), (4, 259), (1, 1036)

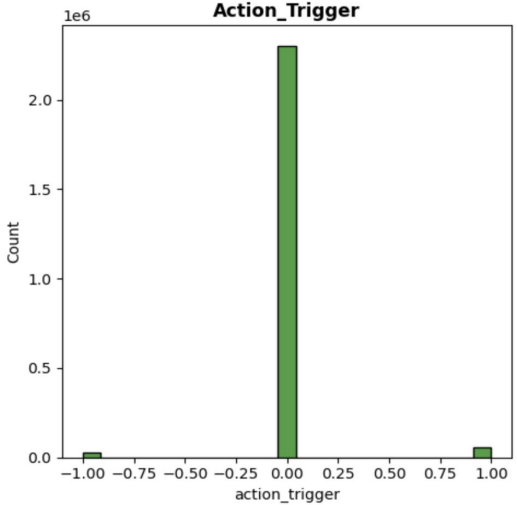
Observation : 4 slots as input behaves well (each input is a **day**'s vector)

score : **0.83356** (public)

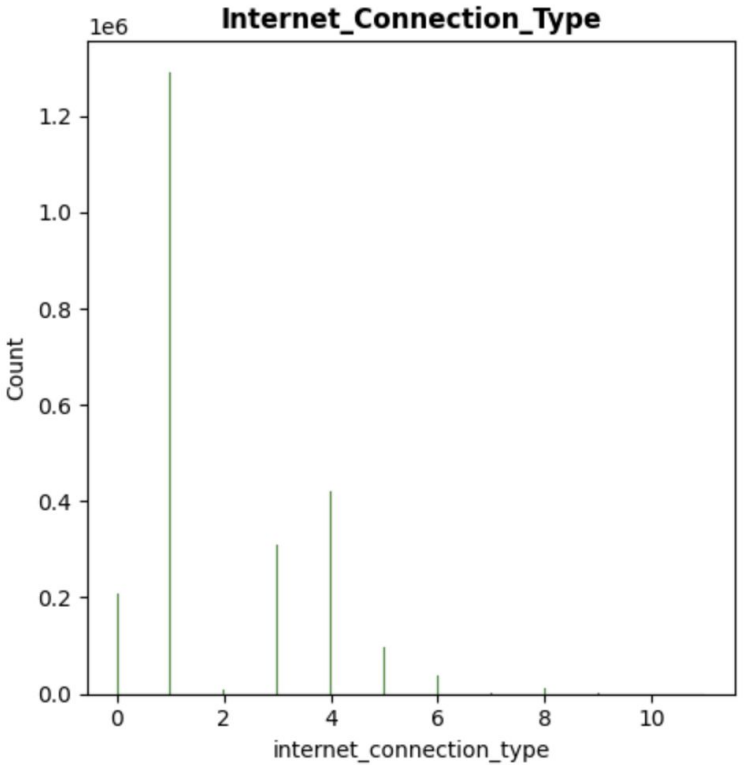








Nominal feature distribution



preprocess - other features

Rough Encoding , calculate value :

- platform : one-hot encoding (4-dim)
  - simply keep the last appearence in the day
- internet_connection :
  - (0~3)->1 *(cellular)* , (4~7)->-1 *(online,wifi)* , (8~11)->0 *(unknown)*
  - simply keep the last appearence in the day
- title_in_simulcast :
  - 2->0.5*(unknown)*, 1->1 *(yes)* , 0->0 *(no)*
  - Take mean value of the day
- action_trigger :
  - Take mean value in the day

```
action_trigger: {
  'api error': '0',
  "cast': '1',
  'error': '2',
  'interrupt': '3',
  'leave': '4',
  'limit access': '5',
  "limit single device playing': '6',
  'network error': '7',
  'next episode': '8',
  'pause': '9',
  'player error': '10',
  'previous episode': '11',
  'seek': '12',
  'video ended': '13',
}
```

preprocess - other features

Encode title in two ways :

1. Keep the most 10 popular titles (longest sum of played_duration) only.

   Rank them with number like 50, 45, ... 5. Others is encoded to 0

2. Sort the title by their popularity, and encode them in the descending order (most popular has highest number)

preprocess - result

After several combination of features and scaling ...

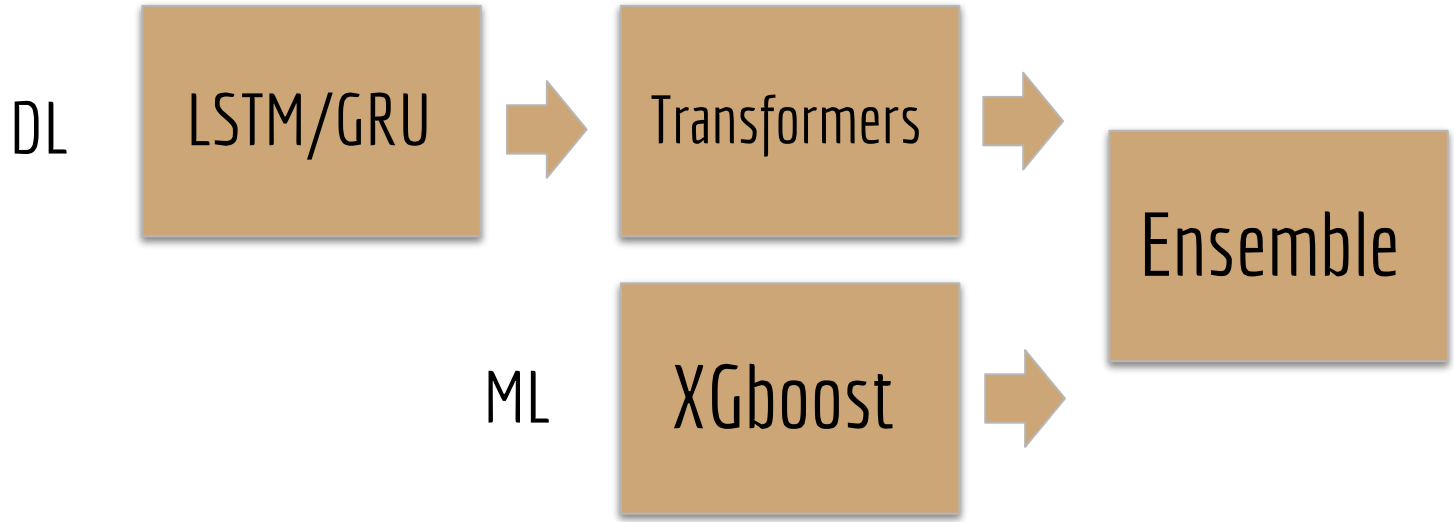
The score didn't improve, maybe it's because we didn't observe the relationship between Origin features / Transformd features with the target. Or use method like SHAP to inspect the feature importance.

However, we decided to keep only [**title_in_simulcast , platform**] to use in ensemble.

To increase the diversity and generalization of the ensemble model.

Model

Roadmap



XGBoost

- Popular method for Kaggle contest
- Gradient boost (With Random Search Params
  - (n + 1)th tree fix the problem of nth tree.
- Origin score : **0.83235** (public)
- But there's no consideration about time?
  - Add time weights to the data
- best score : **0.83592** (public)

```
arr_1 = np.repeat(1, 924)
arr_2 = np.repeat(6, 28)
arr_3 = np.repeat(10, 28)
arr_6 = np.repeat(20, 28)
arr_10 = np.repeat(36, 28)
```

focus on the last few weeks (36, 20, 10, 6, 1, 1...)

```
params = {'learning_rate': 0.008, 'n_estimators': 650, 'max_depth': 8, 'min_child_weight': 3.5,
                'seed': 64, 'subsample': 0.65, 'colsample_bytree': 0.7, 'gamma': 0.001, 'reg_alpha': 0, 'reg_lambda': 11,
                'tree_method':'gpu_hist'}
# multioutputregressor = MultiOutputRegressor(xgb.XGBRegressor(objective='binary:logistic',
#                                                              eval_metric='auc',**params)).fit(train_X, train_y)

xgboost_model = xgb.XGBRegressor(objective='binary:logistic', eval_metric='auc', **params)
multioutputregressor = MultiOutputRegressor(xgboost_model)
multioutputregressor.fit(X, y, verbose=False, feature_weights=feature_weights)
```

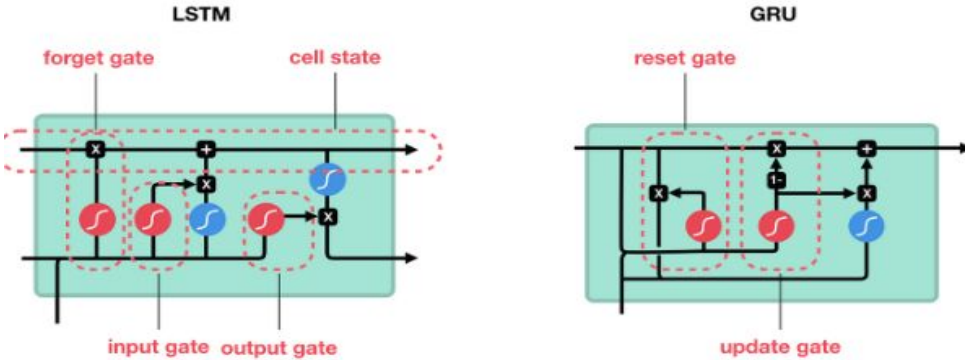
DL: Shared settings

- Loss function: BCEloss
- Optimizer: AdamW
- Dropout is used
- Activation function: Sigmoid
- Use 4 slots as a day
- Train-valid ratio: 4:1
- To better compare the results, we don’t change the above settings.

LSTM / GRU

- They are similar, but GRU is simpler
- Based on the results we got, GRU is slightly better.
- 2 layers, 256 hidden cell

score : LSTM(**0.83356**) / GRU(**0.83502**)



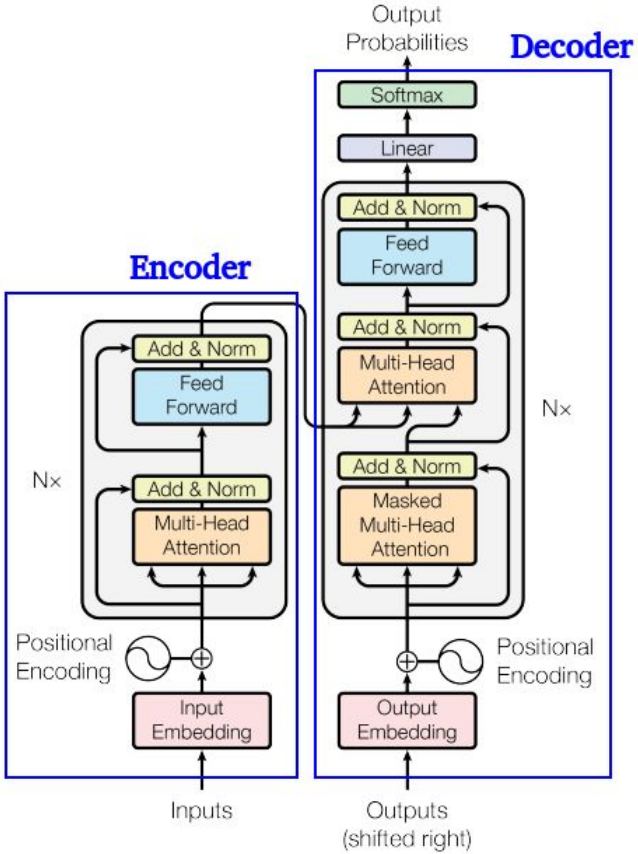
Transformers

Attention is all you need

- Not sequential, avoid recursion
  - May better perceive long term memory than RNN
- Allows parallel computation, train faster
- Positional embeddings

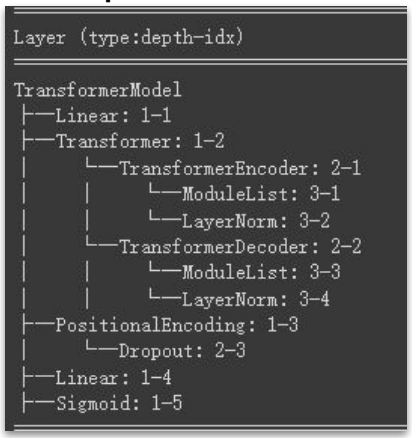
$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right)$$



Transformers

- Duration is sequential, just like sentence.
- Inputs
  - encoder : train data (259 days)
  - decoder : 259th day + test data (7 days)
- Receive the 259th day, and try to predict 260th day, and so on the 7 days are predicted.



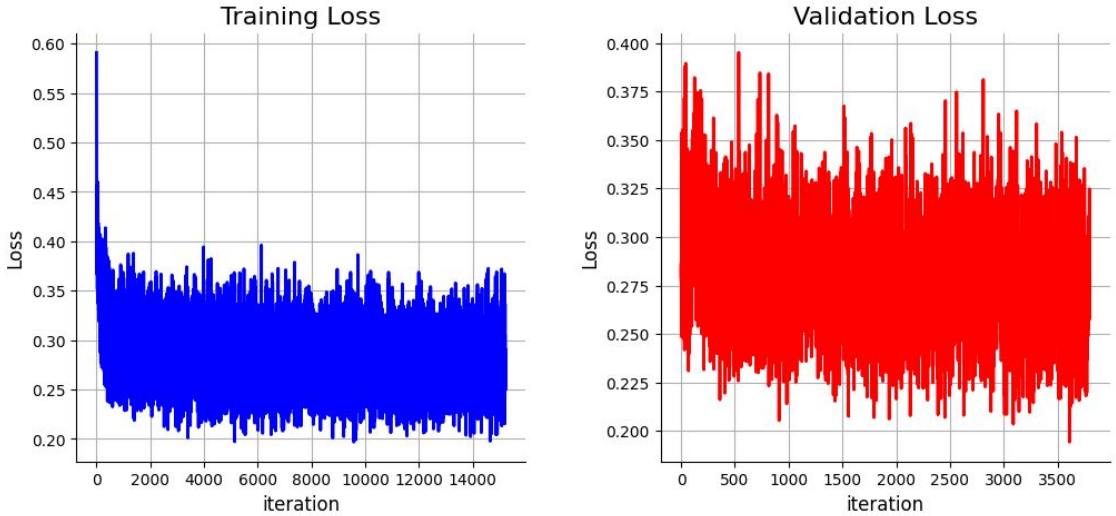
E.g. Inputs =「我愛台大」, outputs = “I love NTU”

When training, with the info from inputs, we give decoder <bos> to predict ‘I’ , and so on.

<bos> -> I
<bos> I -> love
<bos> I love -> NTU

Bottleneck

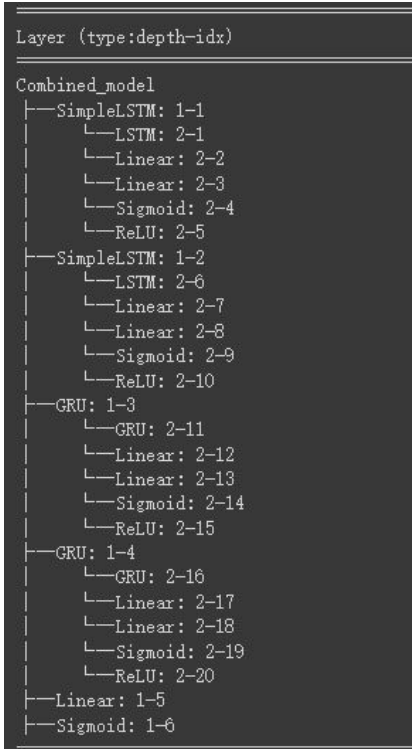
- All the models stuck at 0.83
- With same data input, no single model can outperform others.
- The model seem to be unable to learn “new” things.
- Increase learning rate : overfitting



The loss converge at around 0.3.

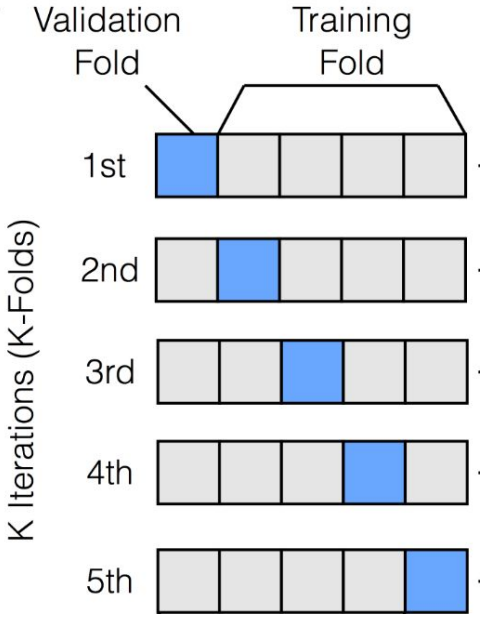
Ensemble time

- LSTM * 2 + GRU * 2 + output of Transformers & XGBoost
- Ensemble model learns the “weights” of each model’s output.
- For LSTM, GRU models :
  - Load the model weights trained before as initialization
  - Train together with the ensemble model.
- For Transformer, XGBoost :
  - Take the train data’s prediction as ensemble model’s train input
- To make LSTM and GRU more different, the input of LSTM is combination of time and nominal data.



Cross-validation

- For LSTM, GRU, and Transformer
- Apply 5-fold cross-validation
- Select two best models
  - Try to fully cover data's information



Results

Score

Model	Data	Score (highest)
LSTM	light	0.803
LSTM	nominal only	0.701
LSTM	duration only	0.82
GRU	duration only	0.831
Transformer	duration only	0.833
XGBoost	duration only	0.832
XGBoost (weights)	duration only	0.835
★ Ensemble	duration and nominal	0.838

Results

- Training with nominal data resulted in bad performance.
  - We give up pure nominal data as input
- The combination of models has best score.
  - Different models indeed contain different information
- None of them reach over 0.84
  - Maybe it’s not about models at this point.
  - The way we extract information from data may be improved.
- Change of duration formula impact the results
  - It’s the evidence of how we deal with data is crucial.

Conclusion

Findings-Data

- During Cross-validation, Fold 2 & 4 are always better.
  - It may be the most wholesome fold.
- The complete data contains several problems
  - There’s duration > 3 days
  - Some nominal data is imbalance
- Without proper preprocessing, nominal data may become noise
  - When input contains both duration & nominal, it performs worse.

Findings-Model

- Time features seem to be good, while other features aren't
  - But nominal features should be included to reach higher score.
- RNN models have similar performance, better try other models for different views.
- Since XGBoost behaves well, we should also consider the possibilities of other traditional ML methods
- The training time differs a lot.
  - LSTM > GRU > Transformer
  - Less complexity or parallel computation

Findings-Behavior

- At the beginning, we use the lstm model on the light dataset.
  - Due to our mistake, we once shuffled the test dataloader, the score is 0.7.
  - People are misplaced but still over 0.5.
  - It seems people still have similar watching habits
- Model tend to give low probability
  - Duration are unlikely to be 1
  - Imbalance 0 & 1
  - Those with more than 0.5 may have actually higher chance.
- Platform and whether or not the show is ongoing are important factors.

Future work

Future work - preprocess

Since the result of only duration is good, it’s close to the top rank teams, we didn’t put much effort on discovering the other features, and decided to focus on our model.

However, if time permitted, we might use more ways to discuss each relationship’s hidden pattern with our target, and try some heuristic way to encode them. Like a time descending way to deal with title is a direction that I always wanted to do but we didn’t have enough time. Such a disappointment !

Future work - Model

We tried different models, but we didn’t spend much time tuning hyperparameters. After listening to others’ presentations, we think that we should try out more possible combinations. Our models are not in the best tuning situation it can reach.

Besides, we only used models that’s about sequence handling, CNN was in consideration but not implemented at the end. The layers of the models should also be tested and compared.

In conclusion, we tried several combination of models and preprocessed data, and some of them are not shown in this presentation. It will be better if we test them in a systematic way.

Reference

References

Vaswani, A., et al. (2017). "Attention is all you need." Advances in neural information processing systems 30.

KKTV 1st solution

KKTV 2nd solution

XGB

XGB Feature Weight

XGB Evaluate

XGB multiple proba output

XGB multiple proba output-2

Scale Feature

References

LSTM, GRU

LSTM structure, input shape

LSTM code pytorch

LSTM code pytorch-1

LSTM code pytorch-2

LSTM code pytorch-3

LSTM code pytorch-4

BCE loss function

Why use BCE

References

cross-validation

Transformer - postional encoding

Transformer

Why transformer

How to use Transformer Networks to build a Forecasting model | by Youness Mansar | Towards Data Science

Block-Recurrent Transformer: LSTM and Transformer Combined | by Nikos Kafritsas | Towards Data Science

A detailed guide to PyTorch’s nn.Transformer() module. | by Daniel Melchor | Towards Data Science

Block-Recurrent Transformer: LSTM and Transformer Combined | by Nikos Kafritsas | Towards Data Science

RNN, LSTM or transformers in time-series? | ResearchGate

Code for this project

Note: The links don't contain all the trials and errors we have conducted. We picked some for showcase only.

LSTM:

https://colab.research.google.com/drive/1akhmtt8FPJZyPc9tDEu4Mc7O9FJgDb_e?usp=sharing

Transformer:

https://colab.research.google.com/drive/1z-5yJPCDDm0qMK__DIyPAbaDVD4wlQcG?usp=sharing

Ensemble:

https://colab.research.google.com/drive/1_mw379mr04v9k0gKEwq9e9-74mVHGKIQ?usp=sharing

Work distribution

- B09705009 吳驊祐:
  - Preprocess
  - LSTM
  - XGB
- B09705016 陳瑾叡:
  - GRU
  - Transformer
  - Ensemble
- Mutual work
  - Slides
  - Presentation

[IMUseless]

Thanks for your time!