User Analysis: CTR Prediction on Features & Behaviors

Yingfan Duan, Hazel Gu, Haotian Wu,
Han Yu, Dawei Zhao

Agenda

01
Business Problem
Value & Impact

02
Data
EDA & Preprocessing
Feature Engineering

03
Model Mining
LightGBM
Random Forest

04
Model Findings & Tests
Model Evaluation
Feature Importance

05
Discussion
Key Takeaways

06
Future Works
Potential Improvements

01
Business Problem

Business Problem

Which ~~guy~~ **ad** would you choose?

Advertiser

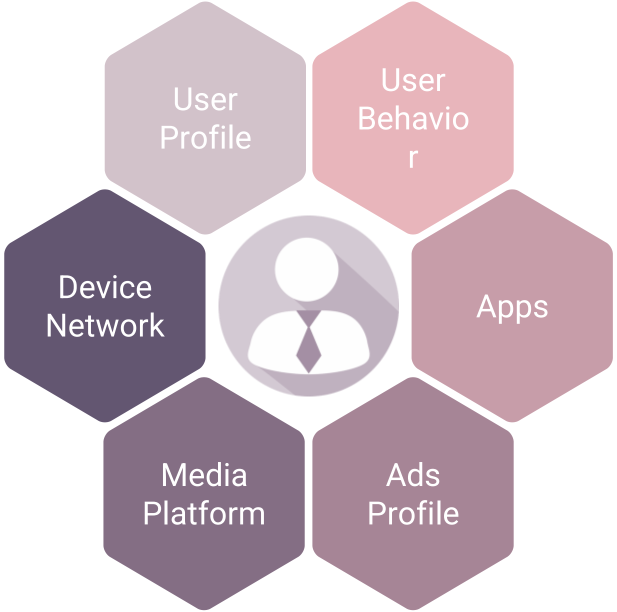
Attempt to figure out the recipe for high CTR observations for **Precision Marketing**

Project Objective

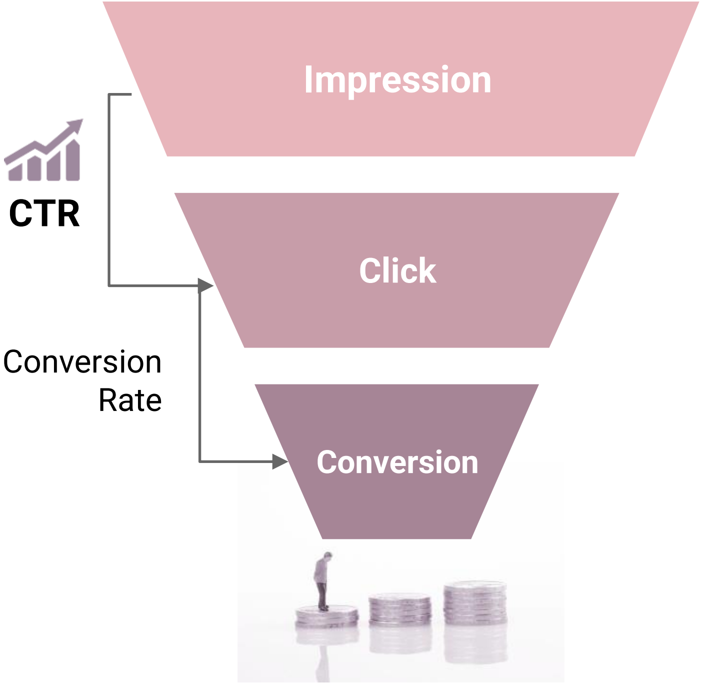
Predict 7th day's Click-Through Rate based on prior consecutive 6 days

Business Value

Precision Marketing



Monetization



02

Data & EDA

Dataset Introduction

Size & Shape	
Size	456 MB
Rows & Columns	#3M #36
Target Variable	‘Label’ (1 = clicked)

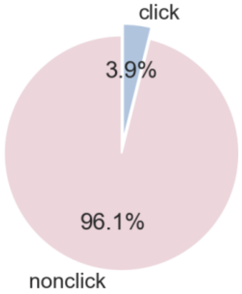
Features Groups	
User	uid age gender net_type …
Ads	adv_id slot_id Inter_type_cd …
Apps	spread_app_id app_first_class his_app_size …

Stats	
Dtype	All numerical Int64 / float64
Missing	Represented by -1
Unique	uid: #1.05M adv_id: #5796

EDA

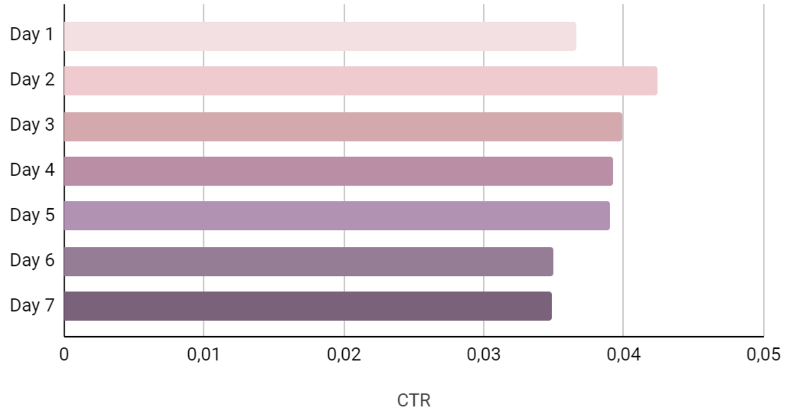
01 Pie Chart

Very Imbalanced target variable



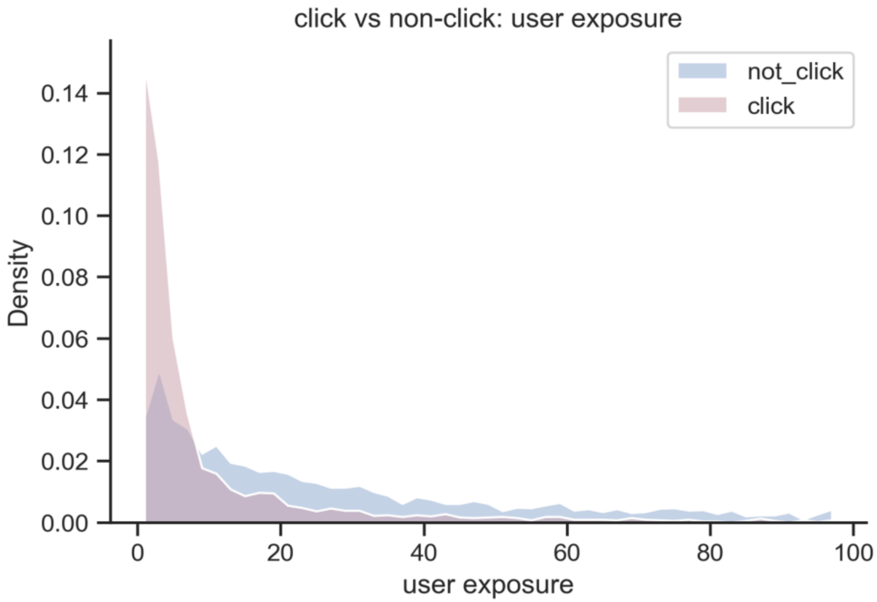
02 Bar chart

CTR of Day 6 & 7 are lower,
Day 2 has the highest CTR



03 Histplot

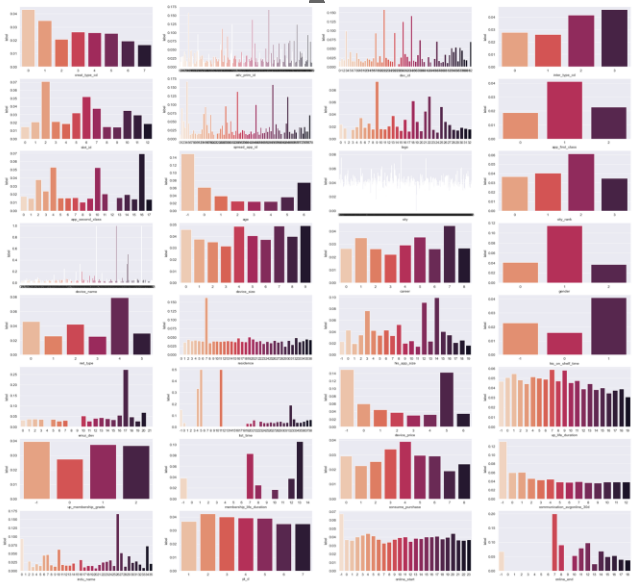
User with low impression are much more likely to click

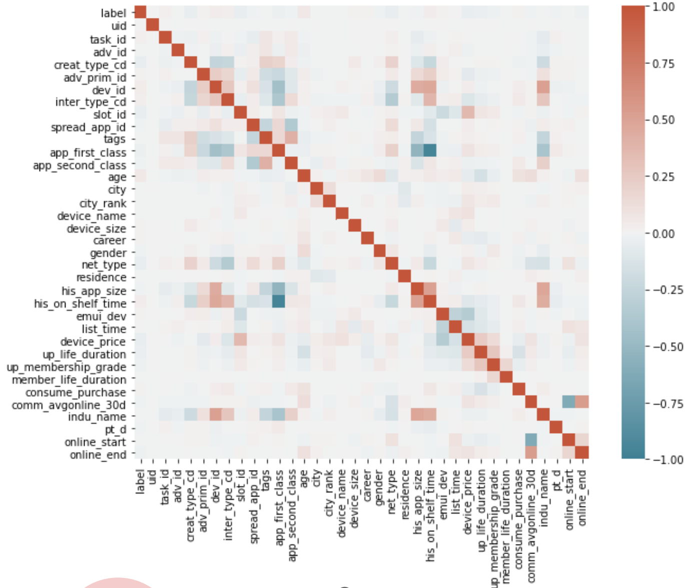


EDA

04

CTR Distribution





05

Correlation Heatmap

Feature Engineering Overview

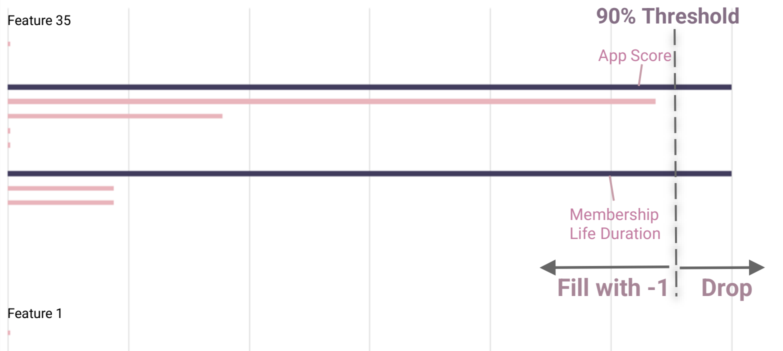
35 Features

Data Preprocessing		Feature Engineering	
01	Missing Value/ Basic Feature Extraction	01	User Exposure Feature
02	Categorical Data Encoder	02	Cross Effect Feature
03	Ordinal Data Encoder	03	CTR Feature
04	Memory Reduction	04	Embedding Feature

Data Preprocessing

01 Missing Value/ Basic Feature Extraction

Missing Value



Basic Feature Extraction

Online Time: Text	4^5^6^7^8^9^10^11^12^ 13^14^15^16^17^18^19^20
Start Time	4
End Time	20

02 Nominal Data Encoder

City

Gender

Advertisement ID

Device ID

Label Encoder except -1

Data Preprocessing

03 Ordinal Data Encoder

- Not actually continuous

Mobile Device Launch Time

Device Price/Size

Put into Buckets

Equally Spaced

Frequency Considered

04 Memory Reduction

Data Structure	
Int8	[-128 , 127]
Int16	[-32768 ,32767]
Int32	[-2146483648,...]

Find appropriate data type for each feature → 80% Memory Reduced

Feature Engineering : Exposure and Interaction

Exposure: Count	Interaction: Crossing Count
❑ Compute the count for each feature value per day (both train and validation set: Day 1 - Day 7) ❑ Example: User ‘A’ appeared 3 times on Day 1 ❑ Apply and create the Count Variables to **some features** (User side/Advertisement side/Media(app) side)	❑ Compute the count for each feature pair per day (both train and validation set: Day 1 - Day 7) ❑ Example: User ‘A’ + Advertisement ‘Apple’ appeared 5 times on Day 1 ❑ Apply and create the Crossing features to **some pair generated by user profile and advertisement characteristics**

Feature Engineering : CTR - Related

CTR	Previous Day CTR
❑ Using the 'LABEL' column (mean) ❑ The CTR for train (day 1 - day 6) is computed using its own day's label mean ❑ The CTR for validation (day 7) is evaluated using the overall label mean of the rest days ❑ Apply and create the CTR features to **every features** in the data set	❑ Using the 'LABEL' column (mean) ❑ Calculate the CTR based on the previous day's label mean ❑ Set day 6 as the previous day of both day 1 (train) and day 7 (validation) ❑ Apply and create the PREVDAY_CTR features to **every features** in the data set

Feature Engineering : Embedding

Word2Vec

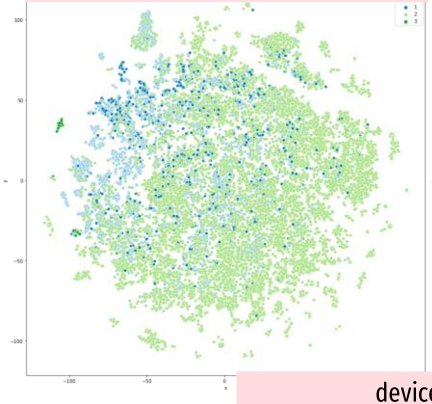
- ❑ Convert data into numerical matrix
- ❑ Use SKIP - GRAM for Word2Vec
- ❑ Set embedding size = 8
- ❑ Primarily apply to User & Ads related features cross-relationship with others

```
Example:
User id & Adv id       Adv id & User age
User id & Adv tags     Adv id & Adv App id
Adv id & Residence     Adv id & City rank
```

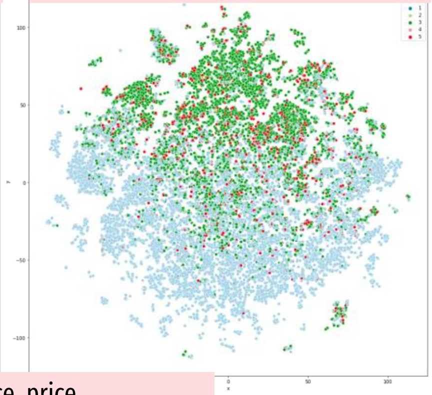
Embedding: T-SNE Visualization

With obvious patterns

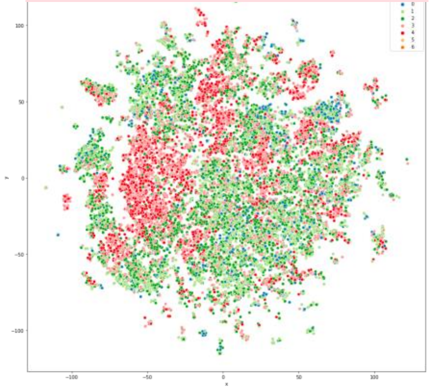
inter_type



net_type

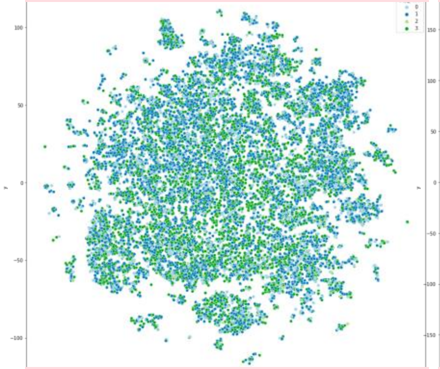


device_price

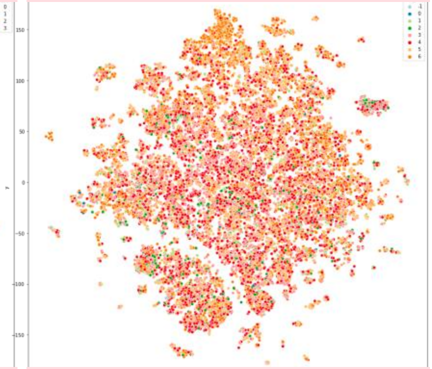


Without patterns

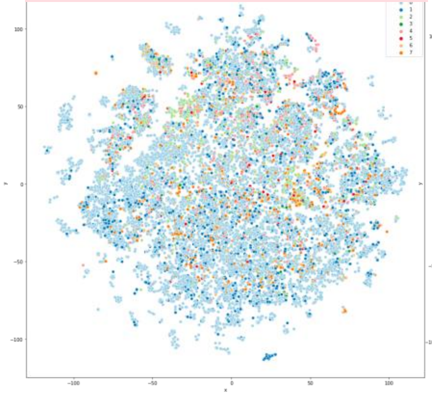
city_rank



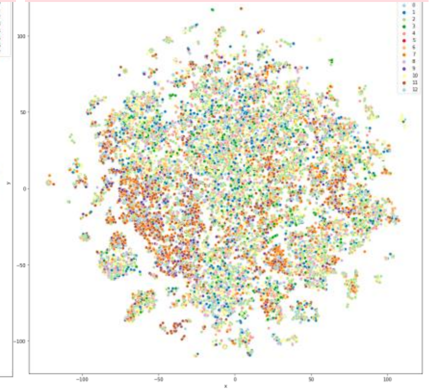
age



creat_type



slot_id



03
Model Mining

LightGBM Introduction

LightGBM (Light Gradient Boosting)

- ❑ Developed by Microsoft in 2016
- ❑ Distributed Gradient Boosting
- ❑ Decision Tree Algorithm
- ❑ Used for classification, ranking, etc.
- ❑ Improve performance and scalability
- ❑ Gradient-Based One-Side Sampling (GOSS)
- ❑ Exclusive Feature Bunding (EFB)
- ❑ **ALWAYS used for CTR prediction problem & high-dimensional data**

Histogram based algorithm

buckets continuous features into discrete bins

EFB

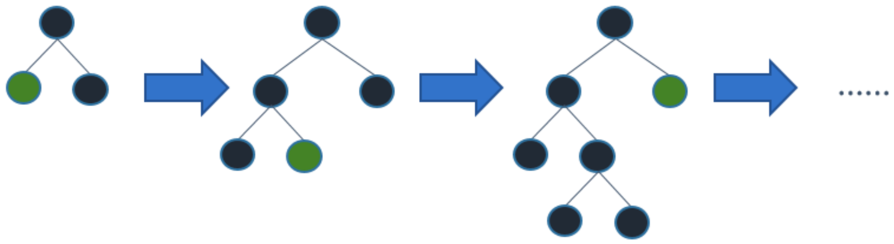
Dimension reduction by bundling features together

GOSS

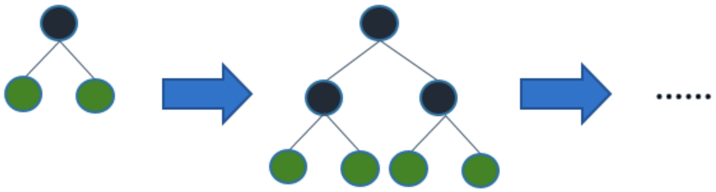
Retains large gradients while random sampling small gradients

LightGBM & Random Forest

LightGBM		Random Forest
Boosting Sample according to error rate	01	**Bagging** Sample drawn with replacement
Leaf - Wise Avoid overfitting with smaller computation cost	02	**Level - Wise** Good for engineering optimization



Leaf-wise tree growth



Level-wise tree growth

04
Model Findings

Model Framework

01 Train Valid split
Use day 1-6 for training , use day 7 for validation

02 Undersample
Sample negative training samples by 50%

Final model

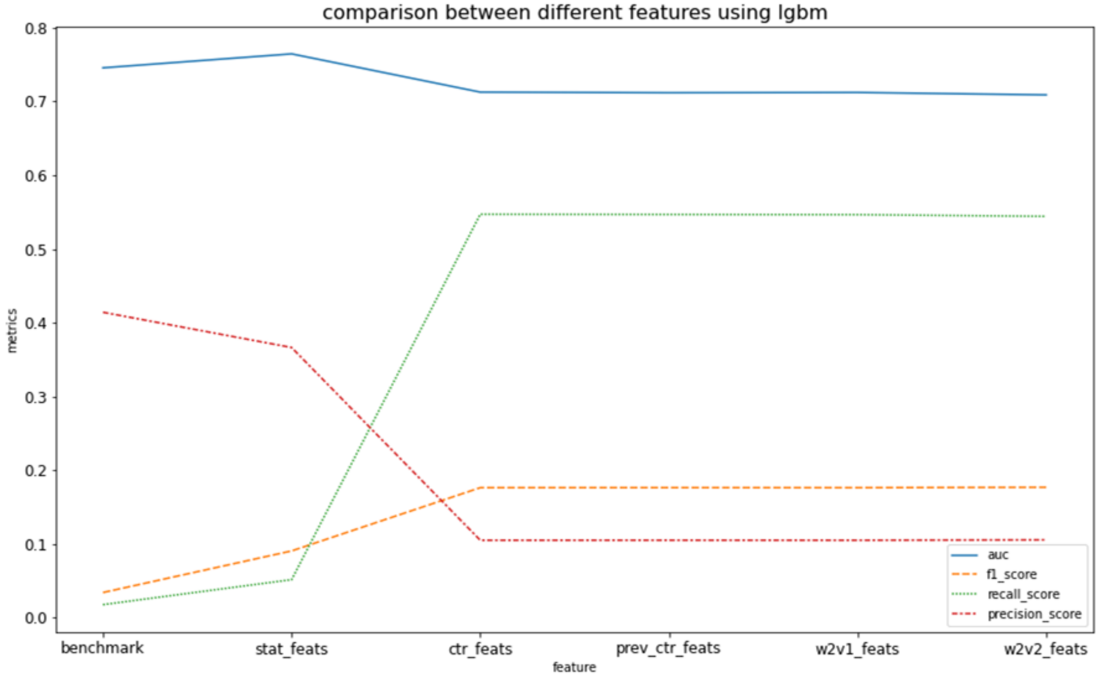
03 Feature selection
Use LightGBM to compare different features

04 Model selection
Use AUC to compare lgbm and random forest

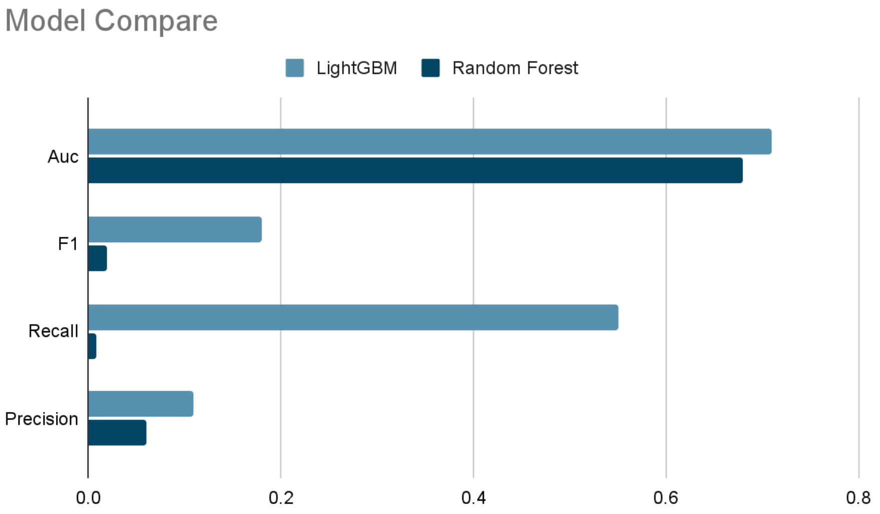
Feature Performance Comparison

LightGBM	
Stat features	AUC + 0.02
CTR features	Recall + 0.5
W2V features	AUC increases

We selected original features, stat features, all ctr features and one set of embedding features for final model.



Model Selection



Choose LightGBM as our final model

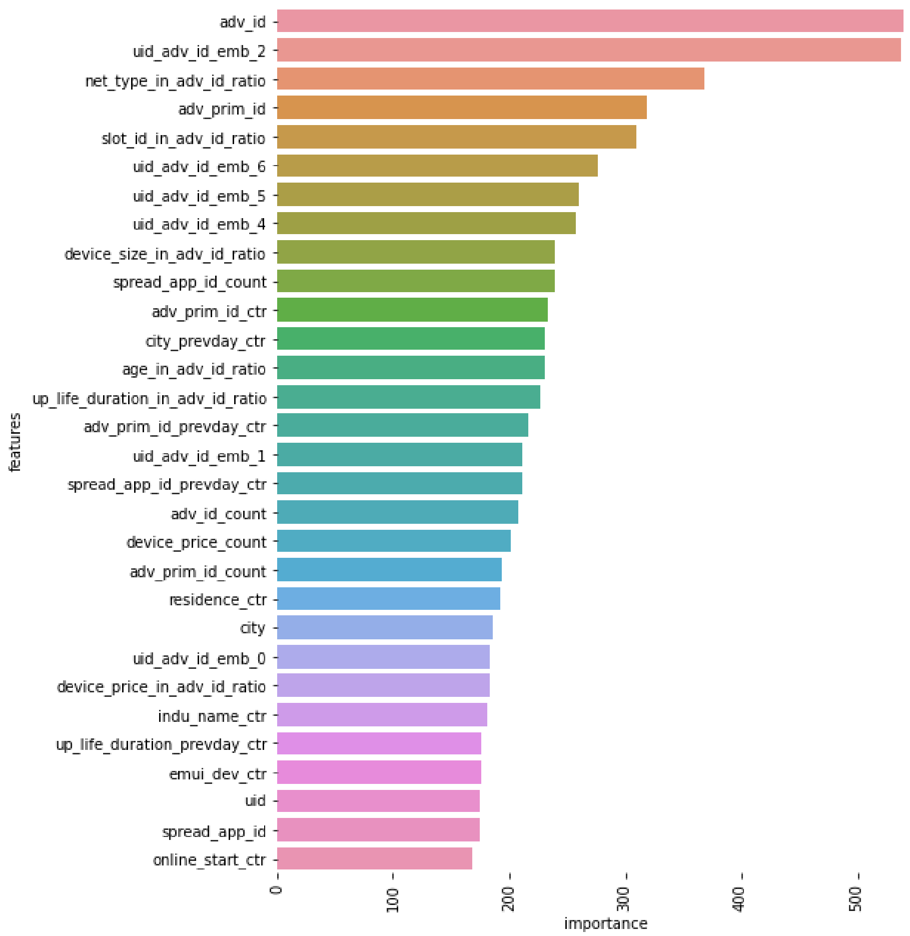
Parameter Tuning

Strategy: Bayesian Optimization

boosting_type	goss	is_unbalanced	True
lambda_l1	1.79	learning_rate	0.05
max_depth	11	lambda_l2	4.75
num_leaves	179	bagging_fraction	1.0
colsample_bytree	0.5	min_child_sample	15

Best Model

AUC	0.79	Recall	0.66
F1	0.18	Precision	0.11



Detailed Feature Importance

01
8 customer related features

Focus on personalized ads, especially users' age, career, residence, etc;

02
Embedding features

Use ad embedding to represent users are effective features for prediction

03
Slot_id feature

Ad position impacts CTR greatly

04
4 device related features

Device type, price, size and net type impact CTR

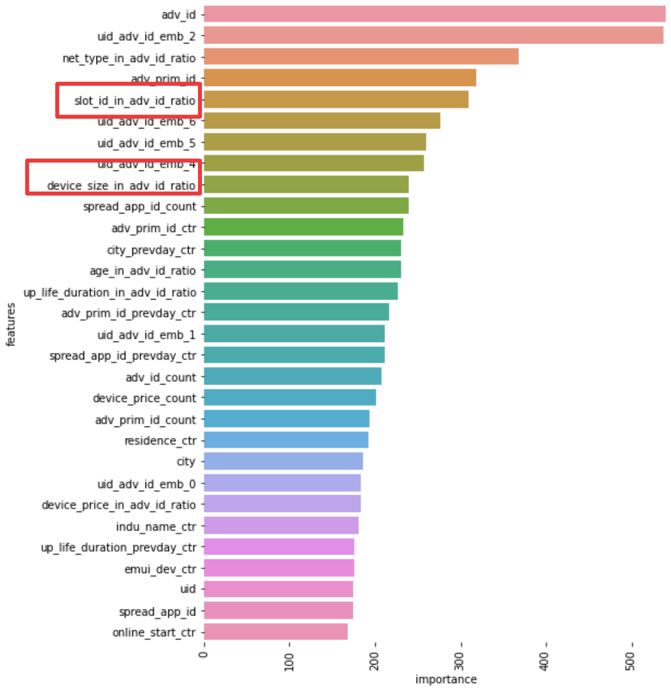
05
Conclusion
Future work

Findings & Recommendations

- **slot_id related feature**
  - Deeper analysis on ads position on Apps
- **device-related features**
  - Have customized ads for different types of mobile devices

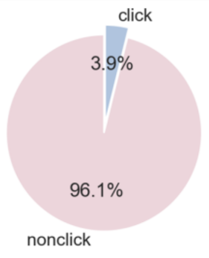
Challenges

- **Enormous data size** - needs high computational power
- **Masked data** - can only conclude on feature importance, but unable to generate literal recommendations
- **Imbalance issue** - 96.1% vs. 3.9%; did perform SMOTE and undersampled the data, but was still unbalanced



Size	456 MB

Dtype	All numerical Int64 / float64



Future Works

Project Improvements

- More delicate parameter tuning
- Embedding is an important technique to predict CTR, but did not lift our model's performance up although deemed as important. We could try more combinations of embedded features and see how they can positively affect our model performance
- Could add weight to days when predicting day 7

Future extensions

- Geospatial location analysis - city and province names in original dataset are masked. Had we have unmasked geographical information, we could potentially analyze CTR trend within different regions
- Time of day analysis - we only know which day each record was on within a 7-day period. Had we have the specific time of day information about when the ads were pushed to users, we could do analysis on CTR patterns throughout different periods of a day

Thank You For Listening!