Ads Click-Through Rate (CTR)

Team **ML11**
Team Members: Nakshatram Shreyas (4th year)
Yeeshukant Singh (2nd year)

Brief Introduction

What is CTR and its Significance?

CTR stands for Click Through Rate and is the number of clicks that your ad receives divided by the number of times your ad is shown: **clicks÷impressions** = CTR.

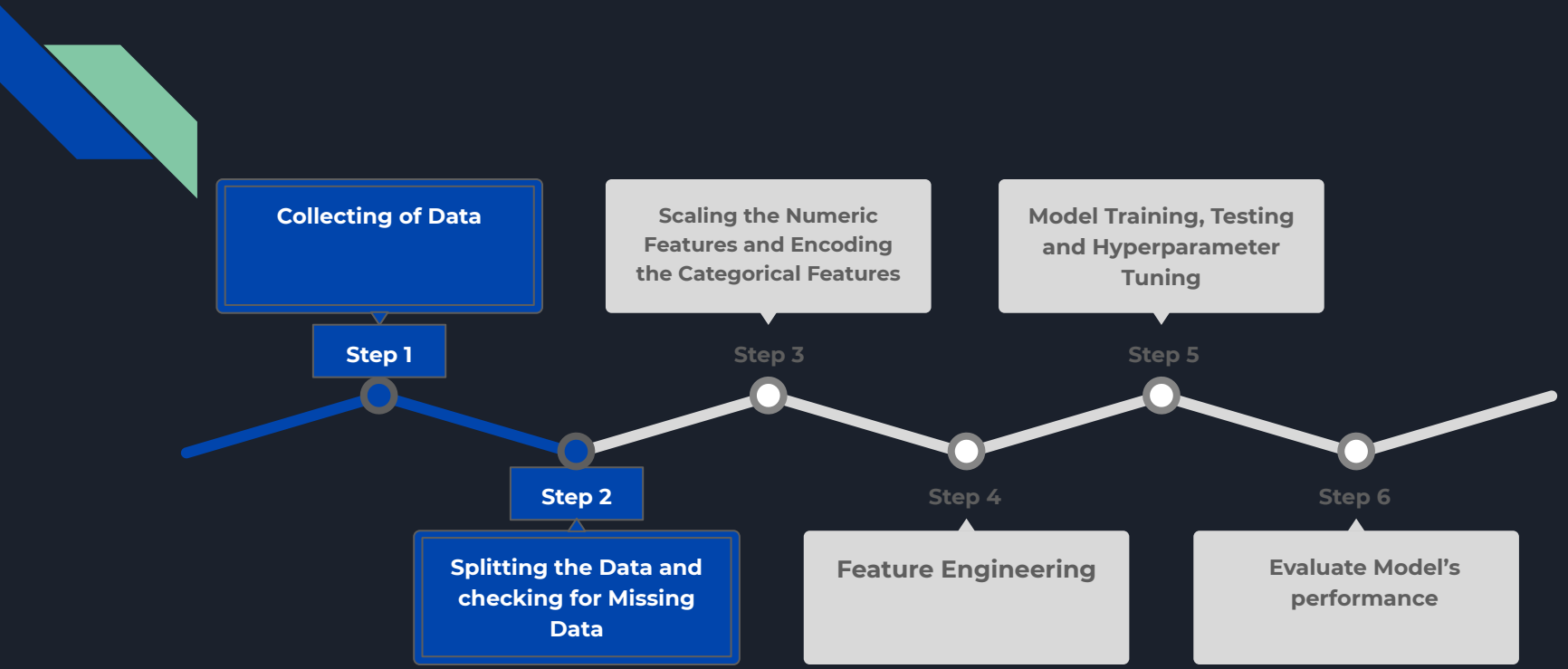
For example, if you had 5 clicks and 100 impressions, then your CTR would be 5%.

In online advertising, click-through rate (CTR) is a very important metric for evaluating keywords and ads performance. As a result, click prediction systems are essential and widely used for sponsored search and real-time bidding.

Problem Statement

Given a user and the page he/she is visiting, find the probability that he/she will click on a given ad. The goal of this analysis is to benchmark the most accurate ML algorithms for CTR estimation. The data set consists of 10 days of click-through data, ordered chronologically. Non-clicks and clicks are subsampled according to different strategies.

Approach


Collecting of Data
Step 1
Step 2
Splitting the Data and checking for Missing Data
Step 3
Scaling the Numeric Features and Encoding the Categorical Features
Step 4
Feature Engineering
Step 5
Model Training, Testing and Hyperparameter Tuning
Step 6
Evaluate Model's performance

Data Collection

The data available for the task was gigantic with around 40 million entries which can not be read at once by normal systems. Therefore we selected 3 million random entries from starting till the end.

Selecting random entries throughout the data ensured that no information is missed out of data and there is no bias which would have otherwise resulted in poor performance of our model.

No Null/NaN values in data.

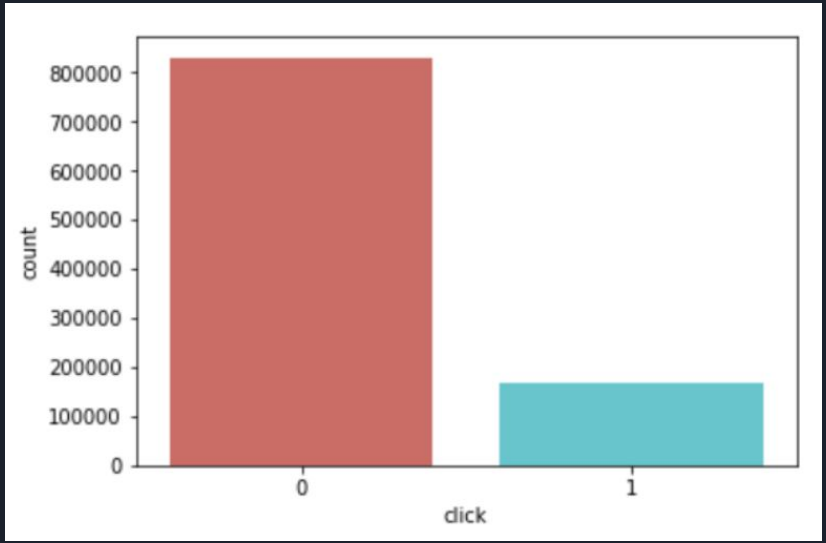
Independent Variables:

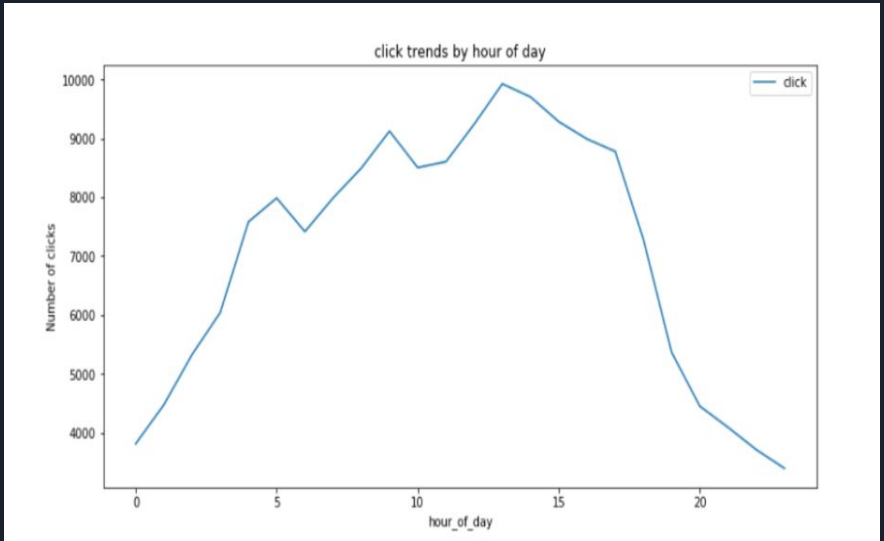
- **id: ad identifier** — **type: int**
- **hour:** — **type: datetime**
- **C1 -- anonymized categorical variable** — **type: int**
- **banner_pos** — **type: int**
- **site_id** — **type: object**
- **site_domain** — **type: object**
- **site_category** — **type: object**
- **app_id** — **type: object**
- **app_domain** — **type: object**
- **app_category** — **type: object**
- **device_id** — **type: object**
- **device_ip** — **type: object**
- **device_model** — **type: object**
- **device_type** — **type: int**
- **device_conn_type** — **type: int**
- **C14-C21 - anonymized categorical variables** — **type: int**

Dependent Variable:

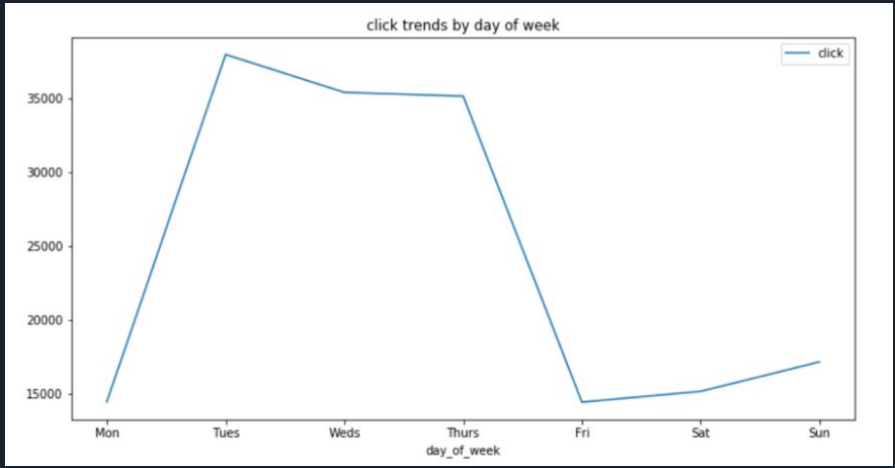
- **click: 0/1 for non-click/click** — **type: int**

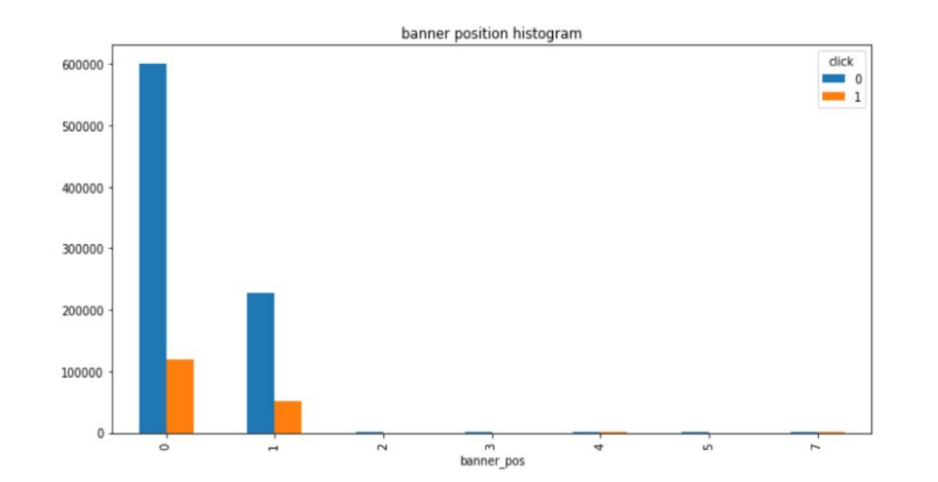
Visualization

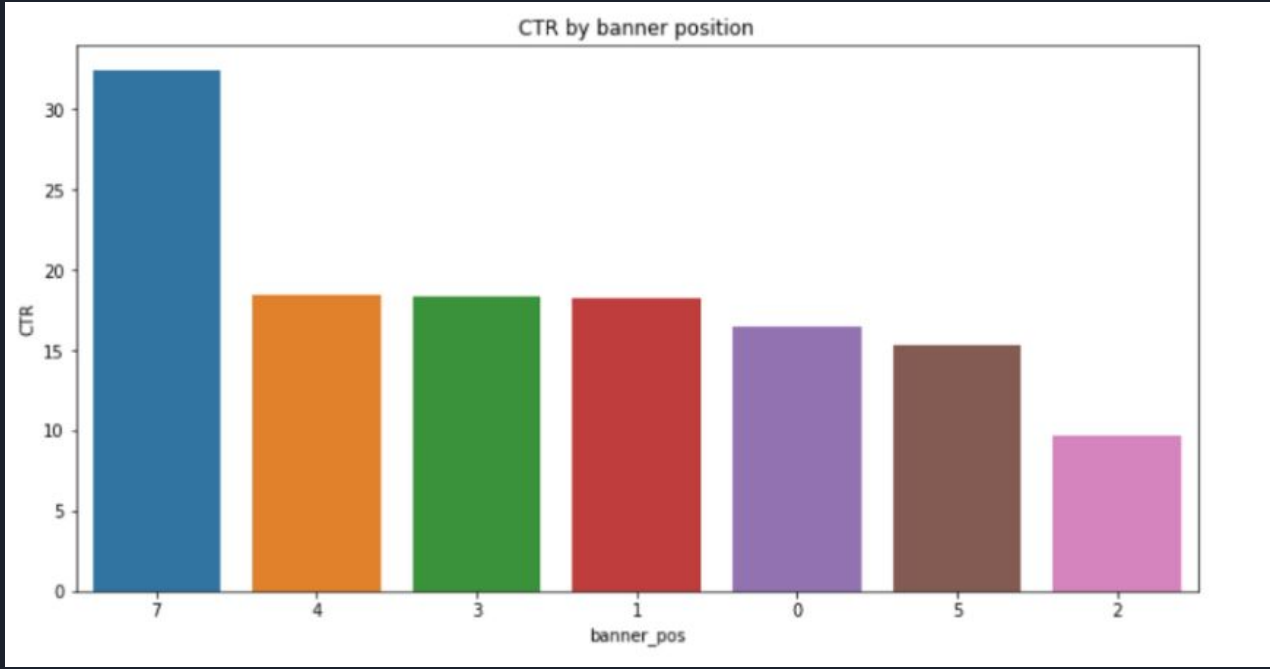


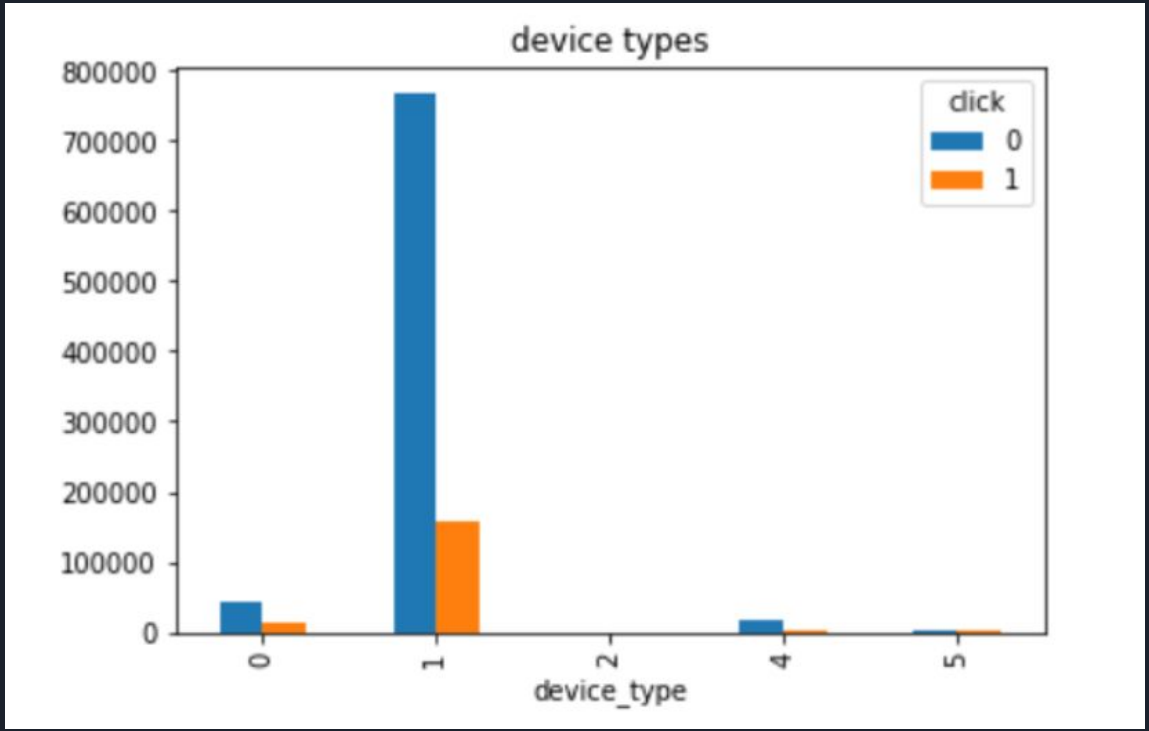


Overall CTR for the data was around 17%


click trends by day of week
click
35000
30000
25000
20000
15000
Mon
Tues
Weds
Thurs
Fri
Sat
Sun
day_of_week


banner position histogram
click
0
1
600000
500000
400000
300000
200000
100000
0
0
1
2
3
4
5
7
banner_pos


CTR by banner position
CTR
30
25
20
15
10
5
0
7
4
3
1
0
5
2
banner_pos


device types
800000
700000
600000
500000
400000
300000
200000
100000
0
click
0
1
0
1
2
4
5
device_type

Data Processing and Feature Engineering

The data had 23 feature columns out of which 9 where categorical features.

These Categorical features could not be used as it is in the model. To overcome this we used a hash function to hash these features along with the ***Sklearn Standard Scaler.***

Next task was to reduce the number of features in the data to reduce the curse of dimensionality.

For this we obtained Mutual Information scores and tried different combinations of features which would increase their MI score.

Continued...

Two feature combinations with improved results were identified.

- device_info = banner_pos+ device_model+ device_conn_type
- user_info = device_ip + device_model + device_id.

Consequently only the combinations were retained and individual features were dropped.

This modification improved classification and, the overall precision of model.

To further reduce the number of features we dropped the features like ‘id’ and ‘hour’ which provided no help in classification.

Model Training, Testing and Evaluation

Three different models were used with different configurations for training and testing, which are:

- Decision Tree Classifier
- Random forest Classifier
- XGBoost Classifier.

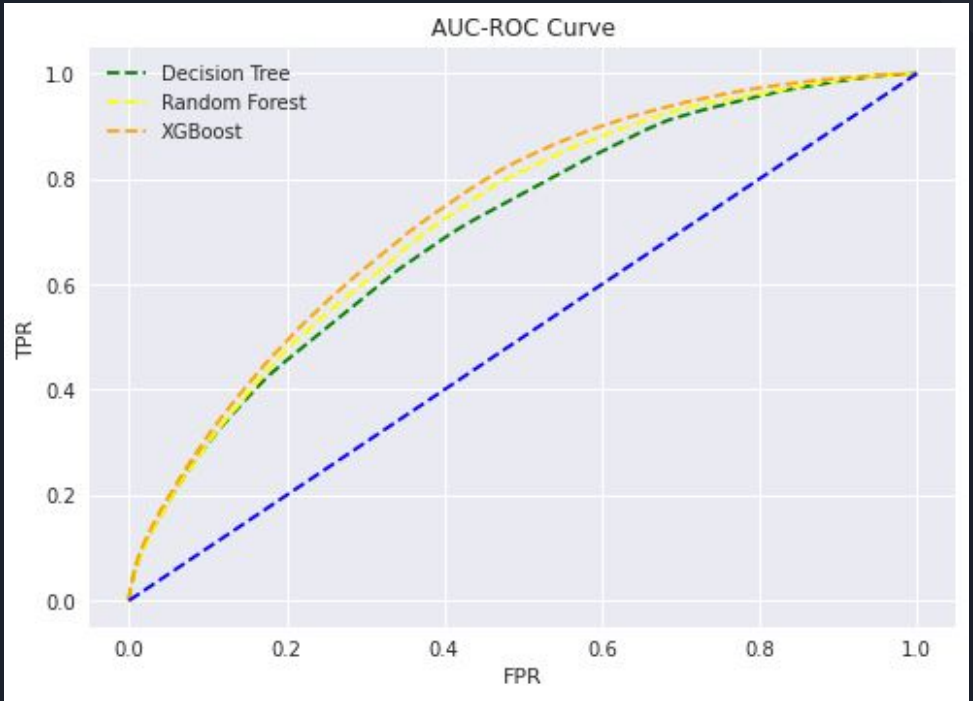
Of all these XGBoost performed better in classifying with better ROC AUC score and cross validation scores.

For cross validation, Kfold cross validation technique was used.

By this, the precision score achieved was about **80.02%** (with cross validation)

Continued...

	Precision(with Cross validation)	ROC AUC score
Decision Tree	0.7947	0.7052
Random Forest	0.7950	0.7162
XGBoost	0.8002	0.7380


AUC-ROC Curve
Decision Tree
Random Forest
XGBoost
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0

Application

Since CTR helps in knowing the performance of Ads, using this model with some changes and integration with any Web application advertisers can target the end users.

Both users and advertisers will be benefited from this. Users will receive more related ads and advertisers will be able to improve their ads performance.

THANK YOU