Advertisement CTR Prediction

Project Proposal

DS5220 / Fall 2023 Semester

Team Members: Liyang Song, Qian Yin

Sep 23, 2023

Our Team Work

- We have established a **GitHub Repository** for team collaboration: https://github.com/LiyangSong/Advertisement-CTR-Prediction

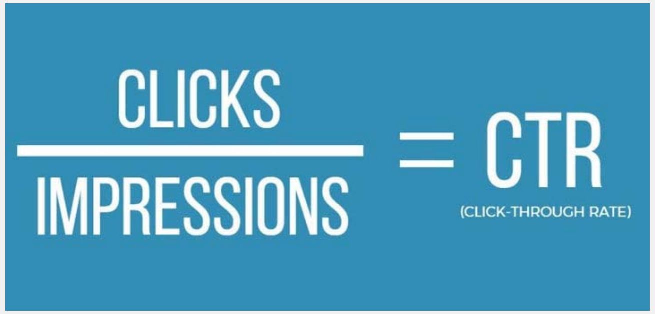
- We have established an **AWS S3 bucket** with public access, which can be used to store all large data files.

- We have scheduled team meetings using **Microsoft Teams.**

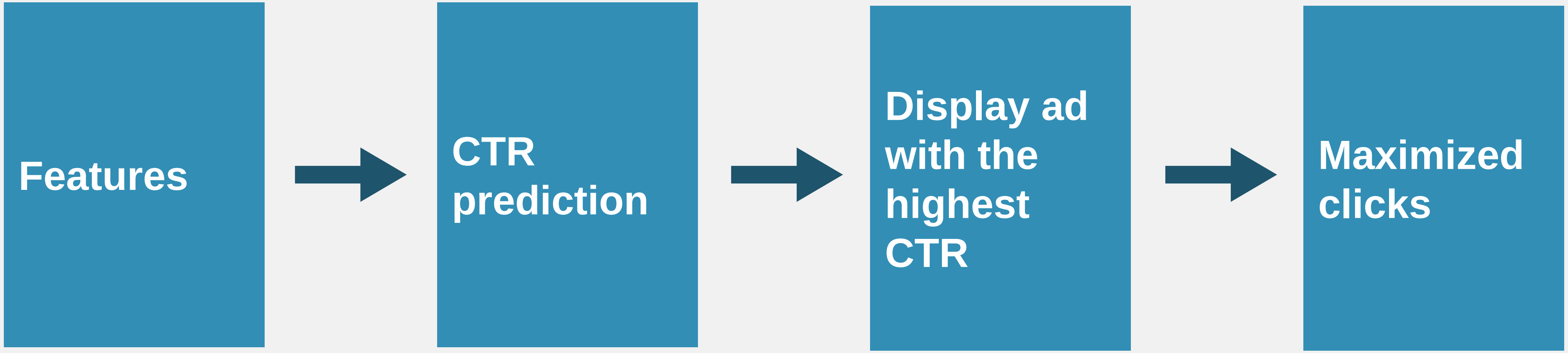
Frame the problem: Define business objective

- The goal is to **increase the efficiency of targeted advertising by accurately predicting the likelihood of a user clicking on an advertisement**, enabling improved return on investment (ROI) and lower cost on advertising. In order to reach this goal, we will try to find the best model for the **CTR (Click-Through Rate)** prediction.

- **CTR is a ratio showing how often people who see ads or free product listing end up clicking it**. CTR is the number of clicks that an ad receives divided by the number of times the ad is shown: *clicks ÷ impressions = CTR*. For example, if one ad had 5 clicks and 100 impressions, then the CTR would be 5%.



Frame the problem: How will the solution be used?



Features → CTR prediction → Display ad with the highest CTR → Maximized clicks

- In this project, we want to use **2020 DIGIX Advertisement CTR Prediction** dataset to build a model that improves advertising CTR prediction. The model can be integrated into the advertising serving system and will be used to **predict the likelihood of a click for each available ad**. And the system can then choose to **display ad with the highest predicted CTR** to the user, thereby potentially **maximizing the actual clicks and revenue**.

Frame the problem: How should performance be measured?

- Because our model predicts whether a customer will click for an advertisement or product or not, this is a **classification model** (target value - 0 or 1). For classification problems, we would consider using ones from the major performance metrics:

	Metric Definition	Align with Object
Accuracy	(Number of correct predictions)/(Total number of predictions)	Accuracy indicates how well the model predicts all the labels correctly. A high accuracy rate could be a sign for good models when the dataset is balanced.
Confusion Matrix	A table with two dimensions (Actual and Predicted) and four terms (True Positives, True Negatives, False Positives, and False Negatives)	Confusion matrix identifies classes being predicted correctly/incorrectly and types of errors being made.
Precision	(True Positives)/(True Positives + False Positives)	Precision is important for checking the correctness of the model. A precision score closer to 1 indicates that the model produces less false positive errors.
Recall	(True Positives)/(True Positives + False Negatives)	Recall is important for reducing the number of false negatives. A recall score closer to 1 indicates that the model is minimizing the false negative errors.
F1-Score	The harmonic mean of both Precision and Recall	F1-Score optimizes precision and recall. When the value of F1 is close to 1, the model is performing well in terms of both precision and recall.
AUC-ROC	The two-dimensional area under the entire ROC(curve plotted between True Positive Rate and False Positive Rate)	The AUC-ROC curve is used to visualize the performance of classification models.The higher the AUC, the better the performance of the model.

Frame the problem: Minimum performance needed

- At the current step, the minimum performance needed to reach the business objective is unclear. It depends on what the company is seeking for and what specific goal the company sets as a priority. In general, **a Click-Through Rate higher than the current one** is the minimum performance requirement.

Frame the problem: Assumptions made so far

1

- **Data is representative:** The provided dataset is representative of the overall user base and ad base, and the sampling method of ads is random and no sampling bias. This will make sure the trained model can generalize well to unseen real-world data.
- **Verify:** Can be verified by comparing dataset with broader business data. However, it is out of our reach in this project.

2

- **Data is relevant:** Features in the dataset have predictive power for CTR, and not too many irrelevant ones are included. Irrelevant features can introduce noise to the model and impact the prediction power.
- **Verify:** Can be verified by correlation measurement or feature selection methods, which will be done in EDA phase.

Frame the problem: Assumptions made so far

3

- **Data is good-quality**: Not too many outliers or missing values. Poor quality data can make system difficult to detect underlying patterns, and make trained model perform unpredictably in production.
- **Verify:** Box plots can be used to identify outliers; missing values can be calculated as percentages for each feature, which will be done in EDA phase.

4

- **External factors are constant**: External factors that may generate impacts on CTR remain constant over time. The model's performance may degrade over time if external factors are considered constant during training, but actually change over time.
- **Verify:** As the dataset contain only 7 consecutive days, it is hard to verify external factors' change without additional data. It can also be validated through monitoring the model's performance after deploying in production.

Get the data: Data source and format

- Data Set:
https://www.kaggle.com/datasets/louischen7/2020-digix-advertisement-ctr-prediction

  - The dataset contains advertising behavior data collected from seven consecutive days. Detailed data field descriptions: *data_fields.json*
  - The raw data file is stored in CSV format and has been compressed and uploaded to out team's **AWS S3 bucket** with public access.
  - Use *download_data.ipynb* to download and unzip the raw data file.
  - The raw data is stored in CSV format with columns spit by “|”. It can be easily read by *pandas.read_csv(file_path, sep="|")* and transferred to a pandas dataframe.

Get the data: Train and test set split

- The train and test set split has completed using *train_test_split.ipynb* with a test size fraction of 0.2.
- The dataset is **heavily imbalanced** with over 95% zero values in **'label'** column, this will introduce a risk that the minority class might not be adequately represented in either the training set or the test set (or both). This can lead to models that are poorly generalized or validated. To counteract this, **stratified sampling** is applied.
- Users can reproduce the split process by rerun the *train_test_split.ipynb*, but we will recommend downloading the split train and test data sets using *download_data.ipynb* for better performance.