CRITEO CLICK LOG DATASET ANALYSIS

A Project Report

Presented to

Dr. Nidhi Rastogi

Rochester Institute of Technology

By

Somonnoy Banerjee sb7238@g.rit.edu

Niranjan Ambekar na2883@g.rit.edu

Rakshitha Gade Rozario rg7037@g.rit.edu

Rahul Kengeri rk1087@g.rit.edu

TABLE OF CONTENTS

Development of question/hypothesis

Internet advertisement has become one of the most prominent and effective ways to advertise products, services, or even events. Therefore it is essential for advertisers to effectively place their ads on a website in order to maximize the number of users that will view or click it.

CTR or click-through-rate estimates the probability of a user clicking a potential advertisement. CTR prediction can be used to place ads on a website more efficiently and also can be used to display more relevant ads to users. Generally higher the CTR the more effective the ad placement is. Click-through-rate can be calculated as the number of times a click is made on the ad, divided by the number of times the ad is shown expressed as a percentage.

CTR = Number of clicks-throughs / Number of impressions * 100 (%).

Using data from the Criteo-AI labs which contains feedback for millions of display ads we can build a predictive model that can determine whether an ad will be clicked or not. This kind of model can be used for personalized advertising and recommender systems.

In this project we are using several machine learning models like Logistic Regression, SVM Classifier, XGBoost, Random Forest, KNN, Naive Bayes, Cat Boost, Ada Boost and doing a comparative analysis on their performance on the Criteo-AI lab dataset in order to find the best ML model that works with the given data.

Data research

Went through various sources to look for a dataset of size at least 500 GB. Tried to look into different problem statements to identify places where automation was required using data in order to provide value to the end user of our Data Science project with valuable insights in order to validate and make informed decisions accordingly. Scraped data related to whiskey from the https://www.thewhiskyexchange.com/ and created a dataset accordingly. Created a dataset by scraping data from different websites. Came to know about data scraping and the laws governing scraping. Started looking for questions that concern organizations. This is the point when we came across the question stated above. After getting to know about the requirement of the project, which was to find a dataset large enough and work on that, we came across the website of Criteo AI Labs which works on AI projects. The website of Criteo AI Labs contained a dataset related to Click Logs. It was a 1 TB dataset. Organizations trying to sell their products focus a lot on Click-Through rates. This enables them to increase their revenue. The dataset of Criteo Labs is extremely relevant in this regard. The dataset contains 39 features, 13 of which have Integers values while 26 of which have categorical values. One feature contained 1 or 0 indicating whether an advertisement had been clicked or not respectively.

Literature review

S No.	Paper Title	Author(s)	Methodology	Year of Publication
1.	Click-Through Rate Prediction Using Feature Engineered Boosting Algorithms	Mohamadreza Bakhtyari, Saye Mirzaei	XDBoost algorithm for prediction inference with limited raw data and time.	2021
2.	A Novel CTR Prediction Model Based On DeepFM For Taobao Data	LinShu Li; Jianbo Hong; Sitao Min; Yunfan Xue	DeepFM model which is a combination of FM Component and Deep Component is proposed, which is an end to end model and does not need manual feature engineering.	2021
3.	CTR Prediction Models Considering the Dynamics of User Interest	Hailong Zhang, Jinyao Yan, Yuan Zhang	Deep-based dynamic interest perception network(DIPN) is compared with state-of-the-art models.	2020
4.	Convolutional Neural Networks based ClickThrough Rate Prediction with Multiple Feature Sequences	P. Chan, X. Hu, L. Zhao, D. Yeung, D. Liu and L. Xiao	A Deep Interest Network (DIN) is used by designing a local activation unit to adaptively learn the representation of user interests from historical behaviors with respect to a certain ad.	2018
5.	ETCF: An Ensemble Model for CTR Prediction	Xiaokang Qiu, Yuan Zuo, Guannan Liu	An ensemble model named ETCF is proposed which cascades GBDT with gcForest to	2018

			tackle the practical problems of CTR prediction and do not need much hyper-parameter tuning work to realize its best performance	
6.	ETCF: An Ensemble Model for CTR Prediction	Xiaokang Qiu, Yuan Zuo, Guannan Liu	Ensemble Trees and Cascading Forests cascades gradient boosting decision trees with gcForest to tackle the problems of CTR prediction and do not need much hyper-parameter tuning work to realize its best performance.	2018
7.	Predicting clicks: CTR estimation of advertisements using Logistic Regression classifier	Rohit Kumar, Sneha Manjunath Naik, Vani D Naik, Smita Shiralli, Sunil V.G, Moula Husain	Logistic Regression was implemented on a one week advertisement data of size around 25 GB by considering position and impression as predictor variables.	2015

Analysis strategy

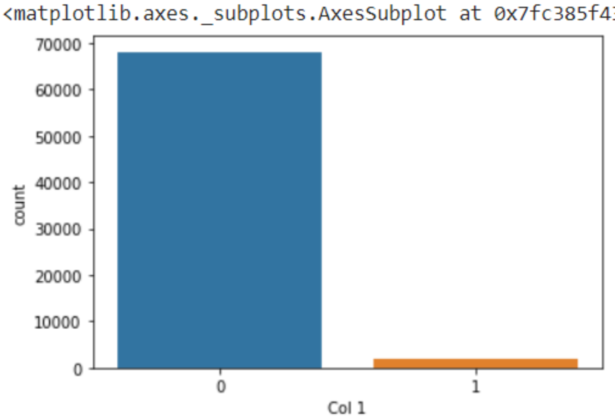
Our main target is to predict whether a user will click on an ad or not. The Criteo-lab dataset contained data over a period of 24 days, and there are 13 columns containing numerical data and 26 columns containing categorical features. The semantics of the columns have been retracted in order to anonymize the data. Even the categorical data has been hashed onto 32 bits, hence the data does not give any information. Our dataset had a lot of null values. We also realized that users rarely click on ads, so our dataset was severely imbalanced. We dropped those columns which had too many null values. For the remaining null values, for numerical columns, if the number of unique values was very large, we imputed with the median and if it was less(Compared to a given threshold), we imputed with the mode. For the categorical features, we imputed the null values with the mode. Next, we applied label encoding for the categorical features. Next, we standardized the numerical columns using Standard Scaler. Next, we tried evaluating different classification models learned in class such as Logistic Regression, KNN, Boosting Algorithms, Ensemble methods etc. Since our dataset was so imbalanced, we were getting exceptional accuracy (95%+) but poor precision and recall. We decided to include those metrics for building our models as well. We tried a bunch of different techniques to deal with the imbalance problem. Specifically, we implemented oversampling, undersampling, SMOTE technique etc. These techniques helped improve the performance of our models, resulting in better precision, recall, and a slightly poor accuracy. Performance on test data continued improving with the adoption of these techniques. Finally, we did a comparative analysis of all the models and found out that Ensemble Models, Boosting Algorithms performed the best on our dataset.

Analysis code

Our code comprises 4 sections namely feature encoding, feature extraction, model building and prediction. In feature encoding, we have converted the 26 categorical columns into numerical values in order to prepare the data for the machine learning models. We also handled missing values by either dropping the columns or imputing the values based on the number of missing values and the number of unique values in that column. We also perform standardization to eliminate extraneous data points. Our data was also incredibly imbalanced with most of the data points leaning toward one class. In order to handle this we have implemented oversampling techniques which duplicate the data points of the minority class. Next, we split the data into training and testing data in order to train the models and evaluate their performance. Each model is compared based on their accuracy, precision and F1 score in order to compare them.

```
[ ]  sns.countplot(x = 'Col 1', data=y_train)

<matplotlib.axes._subplots.AxesSubplot at 0x7fc385f43950>
```



Work planning and organization of each team member

Somonnoy did the job of retrieving the data. Then we did EDA together. Rahul did the data visualization. Rakshitha surveyed various papers related to our topic. We divided different ML models amongst ourselves, as mentioned in the presentation. Niranjan dealt with data imbalance.

Individual Contribution

We began our project by brainstorming ideas and also looking for potentially large datasets. While searching for large datasets I specifically went through a lot of research papers and found that the datasets used were not suitable. Finally after agreeing on the Criteo Click Log Dataset, I tried to understand and work on the first 1000 rows from the day_0.gz file. Since the dataset consisted of 26 categorical columns which were encoded, I found it challenging to interpret it as not much information could be gained. After performing feature engineering, I tried to implement Naive Bayes classifier and CatBoost classifier algorithms. The main reason behind choosing Naive Bayes although it's considered to be a very simple algorithm is because it is not sensitive to irrelevant features. Since the dataset had encoded categorical features and no knowledge of the categorical features could be obtained, I wanted to see how the model would work when each feature is given its own importance. Since the working of the naive bayes classifier algorithm is simple, it gave the lowest accuracy among all the other models. Moving onto the CatBoost classifier algorithm, it is gradient boosted decision trees. As the name suggests Categorical Boosting Algorithm, it is apt to use for categorical variables. CatBoost algorithm requires minimal to no feature engineering implemented before training the model. It is incredibly easy to use and efficient. Also considering the initial requirement of a large dataset, I would say that CatBoost would be among the algorithms that works best.

Conclusion

The structure of the project with intricate details was discussed. Specific benchmarks were placed to evaluate the effectiveness of each step in our project ranging from data extraction to model building. Comparative analysis of a series of algorithms was undertaken and results stored accordingly. The preliminary study was challenging due to the enormous data at disposal and the data also had a lot of missing values. However, feature engineering techniques and sampling strategies were implemented in order to deal with the missing values and improve the performance of the models.