Advertisement Click Prediction

Paneri Patel
Data Science Institute
https://github.com/Paneripatel/DATA1030P

Introduction

Today for companies to sell their products and maximize their revenue they need to dive deep into user engagement for products based on different demographics like what age of people are currently looking to buy what type of products, their location plays an important role, what are they really looking for based on their history, and different genders have different kind of shopping preferences, so to better market products to the right type of audience and get more users to click the advertisements and potentially buy the product, I have developed a model which will predict if a user will click on ads popped on their device based on various factors taken into consideration, such as what is the users age, what time of the day they are looking at their device and if an advertisement pops will they click, what type of device they are looking the ad on, gender, browsing history, and where the advertising might be placed on screen for better marketing. These insights can help us understand user engagement and accurate predictions can get more revenue to the company with better sales. My approach not only focuses on evaluating machine learning models but also provides interpretability, allowing advertisers to identify key factors influencing user behavior.

The dataset is collected from the Kaggle website. My dataset has about 10,000 entries with 9 columns, User's ID, Full Name, Gender, Age, Advertisement position, device type, time of the day(more like morning, afternoon, evening and not time series), browsing history, and click (1-yes, 0-No) helping us understand if the user clicked the advertisement or not. It is a classification problem. My dataset has roughly about 66% of the fraction of features with missing values.

There have been several approaches made to make a good predictive model for advertisement click prediction, several can be found on Kaggle. Those show an accuracy of 60%-80% with different models trained, some with imputing missing values while some with dropping missing values, also some with using models that can handle missing values directly like the XGBoost model.

EDA

A quick analysis of the overall engagement of users based on clicks, around 6500 users have clicked the advertisement and around 3500 have not.

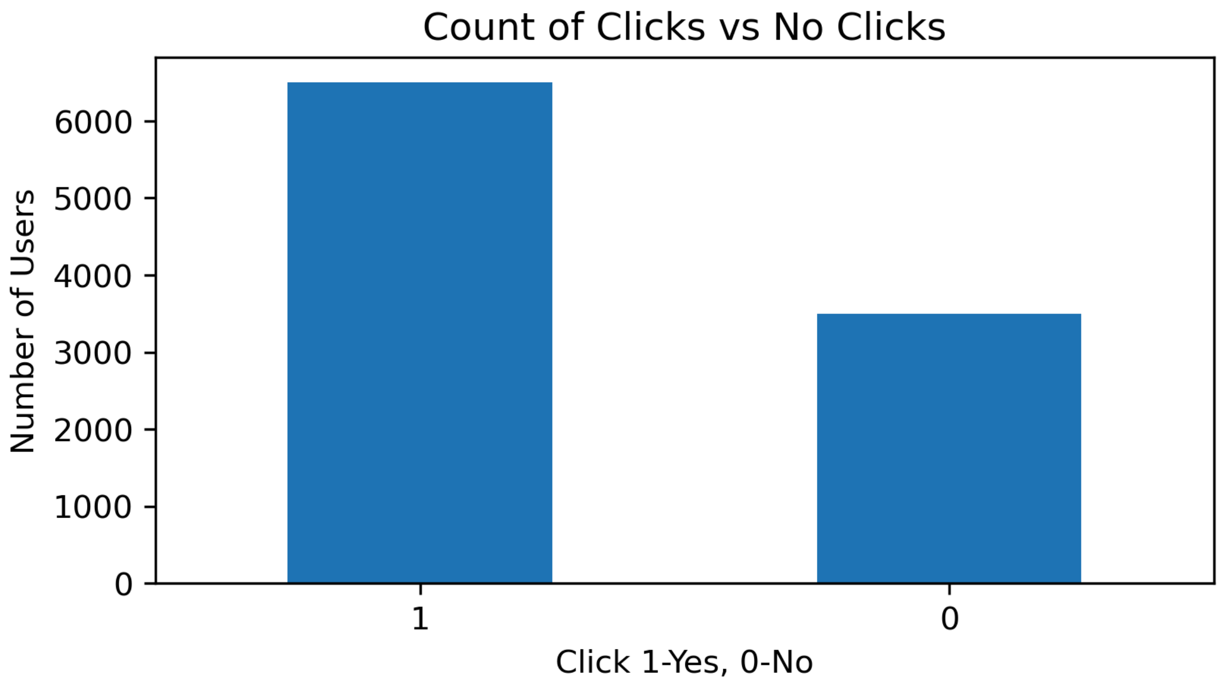

Figure 1. The above plot shows how many users clicked the advertisement. (1-Yes and 0-No)

The proportion of advertisement clicks based on browsing history helps identify which types of browsing behavior are more likely to lead us to advertisement clicks which is critical for targeting advertisements.

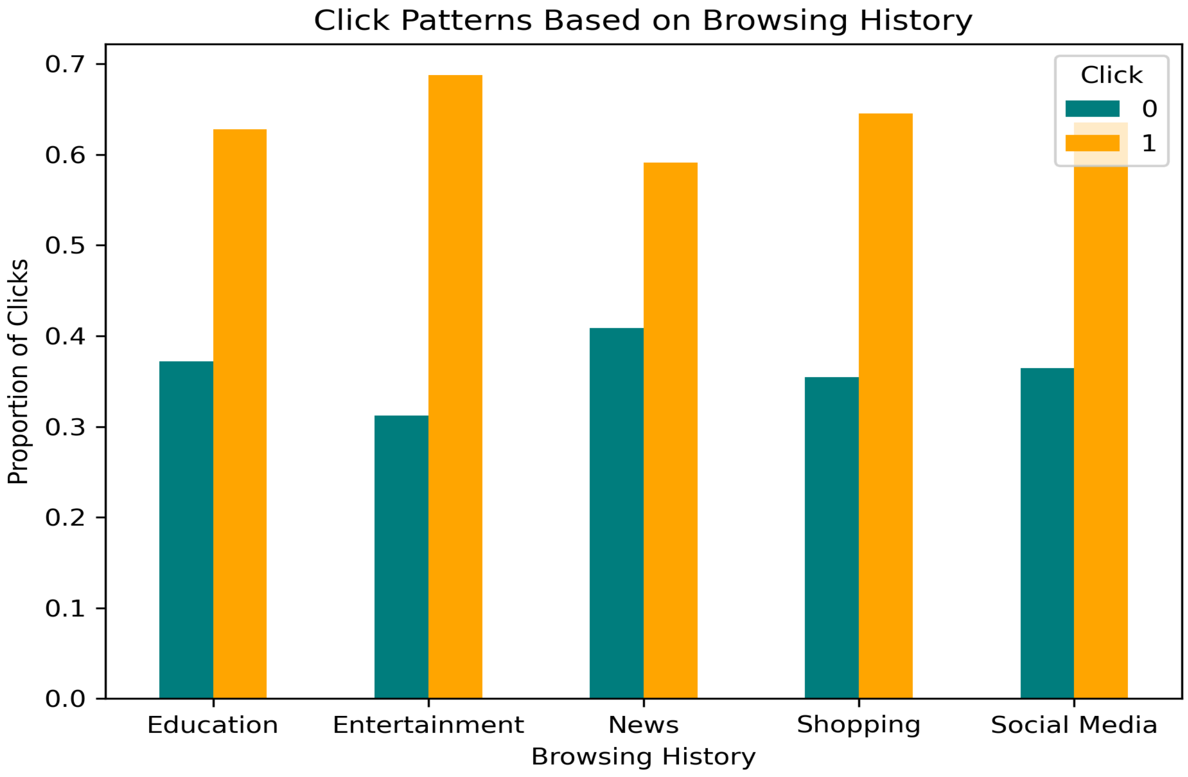

Figure 2. The above plot shows the proportion of advertisement clicks based on users browsing history. (1-Yes and 0-No)

Age plays an important role when it comes to buying products through advertisements, as people of different ages are more focused on different types of their needs. Older people like to buy more medical products that can comfort them, teens and adults would be more into shopping, middle-aged people would be more into buying things for toddlers, etc. The below figure helps us understand how age influences advertisement click behavior.

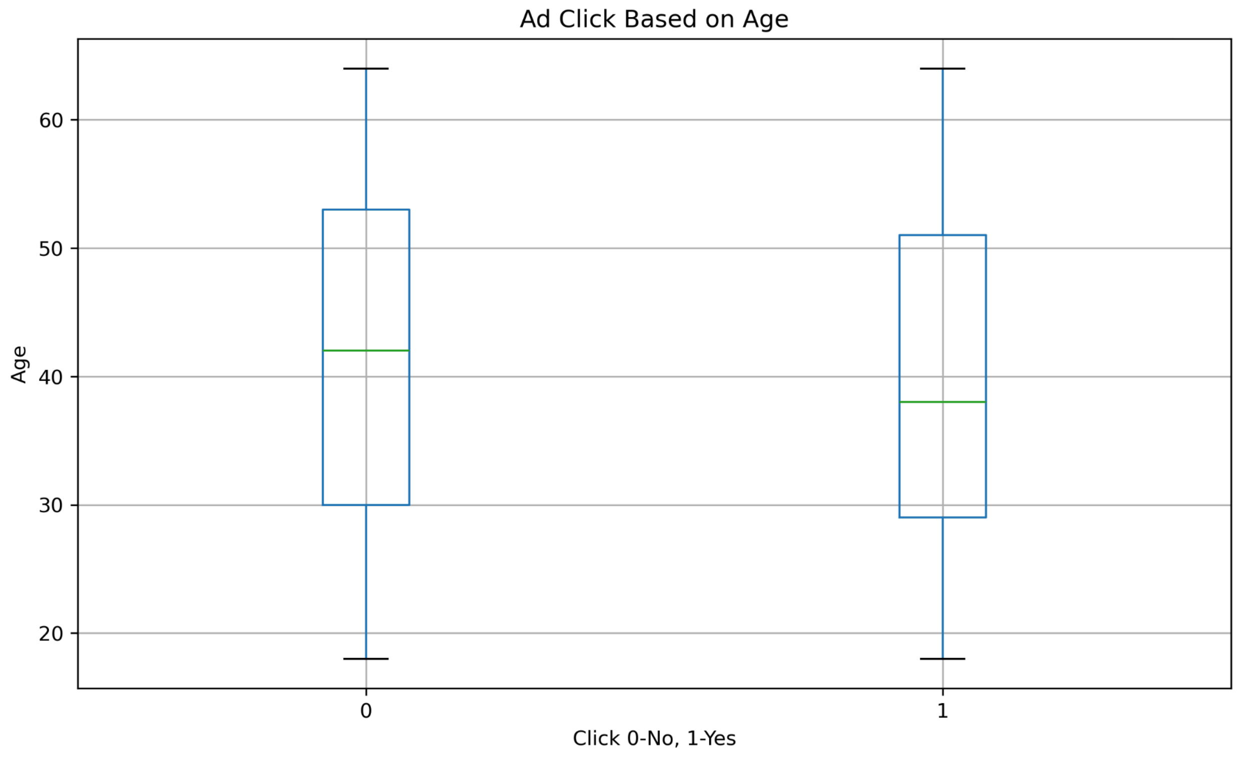

Figure 3. Advertisement clicks based on the user's age.

Below plot, browsing history vs age group helps us understand how different age groups engage with different types of content which helps us understand what types of advertisements for that particular age group should be placed on what type of browsing behaviors.

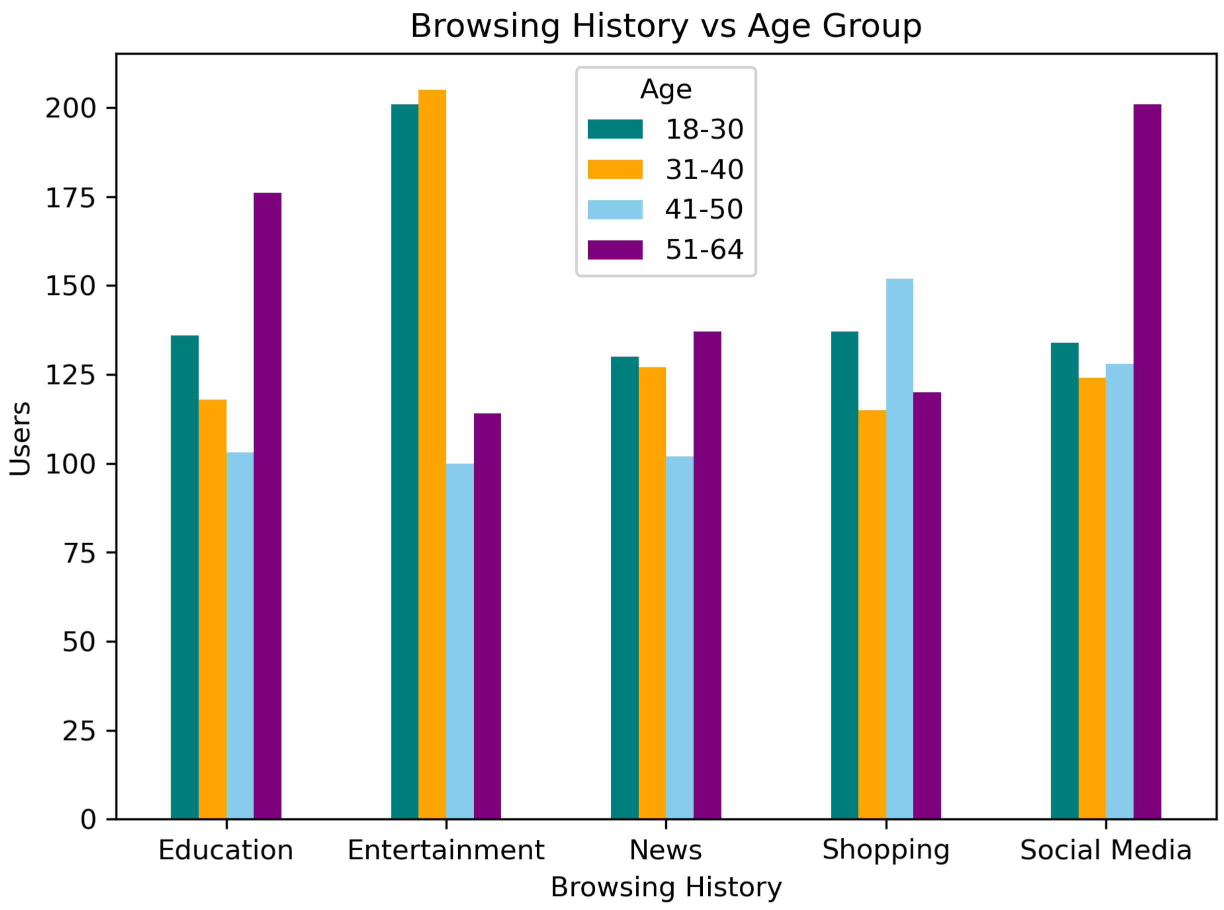

Figure 4. Browsing history of users based on age group.

Some of the EDA insights that I found with various other plots are that younger users in their mid-20s-30s are more likely to click on advertisements compared to older users, younger users are more focused on education and entertainment, making it the top place to add relevant advertisements there. Afternoon and evening are the peak times for advertisement engagement with users being less likely to engage in the morning. Unexpectedly users engaging while watching news and educational content show the highest likelihood of clicking advertisements, lesser during entertainment and shopping content.

Methods & Results

Data splitting - Initially I used GroupShuffleSplit to split the train-test. 80% training + validation, 20% test. Then I used StratifiedGroupKFold for the train and validation split. 5 folds cross-validation on training + validation set. Preventing data leakage, providing robust validation, and accounting for user-level dependencies.

Preprocessing - Dropped column id and full_name as it adds no value. For categorical data(gender, device_type, ad_position, browsing_history, time_of_day) I imputed missing values with a constant value “missing” and encoded using OneHotEncoder, to ignore the missing values as XGBoost directly handles missing values and for other ML algorithms I used reduced feature options on my dataset through which I was able to get 63 patterns to train my ML algorithms on. And for continuous data(age) I used StandardScaler to normalize distributions.

ML Pipeline - I have used ML algorithms- XGBoost, Lasso, Ridge, ElasticNet, and KNN on my dataset over 5 different random states to evaluate which model can have the best accuracy in terms of prediction if the user clicked the advertisement based on demographics. To optimize model performance hyperparameter tuning was done by exploring parameter combinations and GridSearchCV was used to extract the best parameters. The below table shows the parameters I tuned for each model with the bold ones giving the best result.

Models	**Parameters**
Lasso	alpha: 0.001, **0.01**, 0.1, 1, 10
Ridge	alpha: 0.001, 0.01, 0.1, 1, **10**
ElasticNet	alpha: 0.001, **0.01**, 0.1, 1, 10 l1_ratio: 0.1, **0.5**, 0.9
XGBoost	learning_rate: 0.01, **0.03**, 0.05 colsample_bytree: **0.9**, 1.0 subsample: **0.6**, 0.7, 0.8
KNN	n_neighbors: 3, 5, **7** weights: uniform, **distance**

Table 1. The table gives us insights into hyperparameters used to tune various ML Algorithms.

To evaluate my model's performance I chose the f1-score as my metric as I have classification problems and my dataset is imbalanced and through the f1 score I can get the balance of both precision and recall, ensuring both false positives and false negatives are accounted for. Below I have a plot that shows all model's performance, f1 score, and standard deviation with the baseline model.

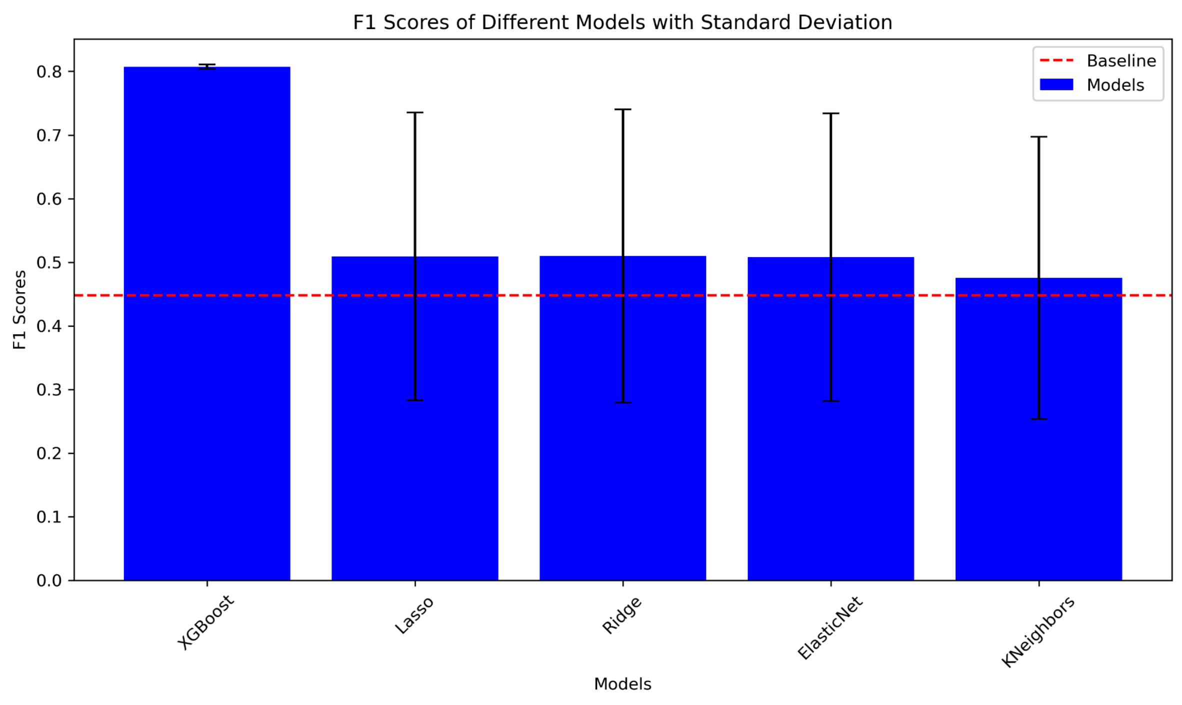

Figure 5. Bar graph showing test scores of different algorithms with F1 and Standard Deviation.

Models	**F-1 Score**	**Standard Deviation**
XGBoost	0.8107	0.0033
Lasso	0.5089	0.2262
Ridge	0.5096	0.2306
ElasticNet	0.5077	0.2262
KNeighbors	0.4752	0.2219
Baseline	0.4478	0.0071

Table 2. F1 Scores with Standard Deviation

As we can see my best model performance is by XGBoost, which has an f1 score of 0.8107 and a standard deviation of 0.0033 which evaluates to XGBoost relatively performing well over random states and different parameters and handling a lot of missing values. While my other models have performed nearly the same as the baseline model and have f1 scores in the same range. The accuracy of my XGBoost model is 0.7234 while that of the baseline model is 0.4525, which shows XGBoost has relatively better accuracy than the baseline model. Below is the confusion matrix which helps us understand how XGBoost has correctly predicted clicks vs falsely predicted.

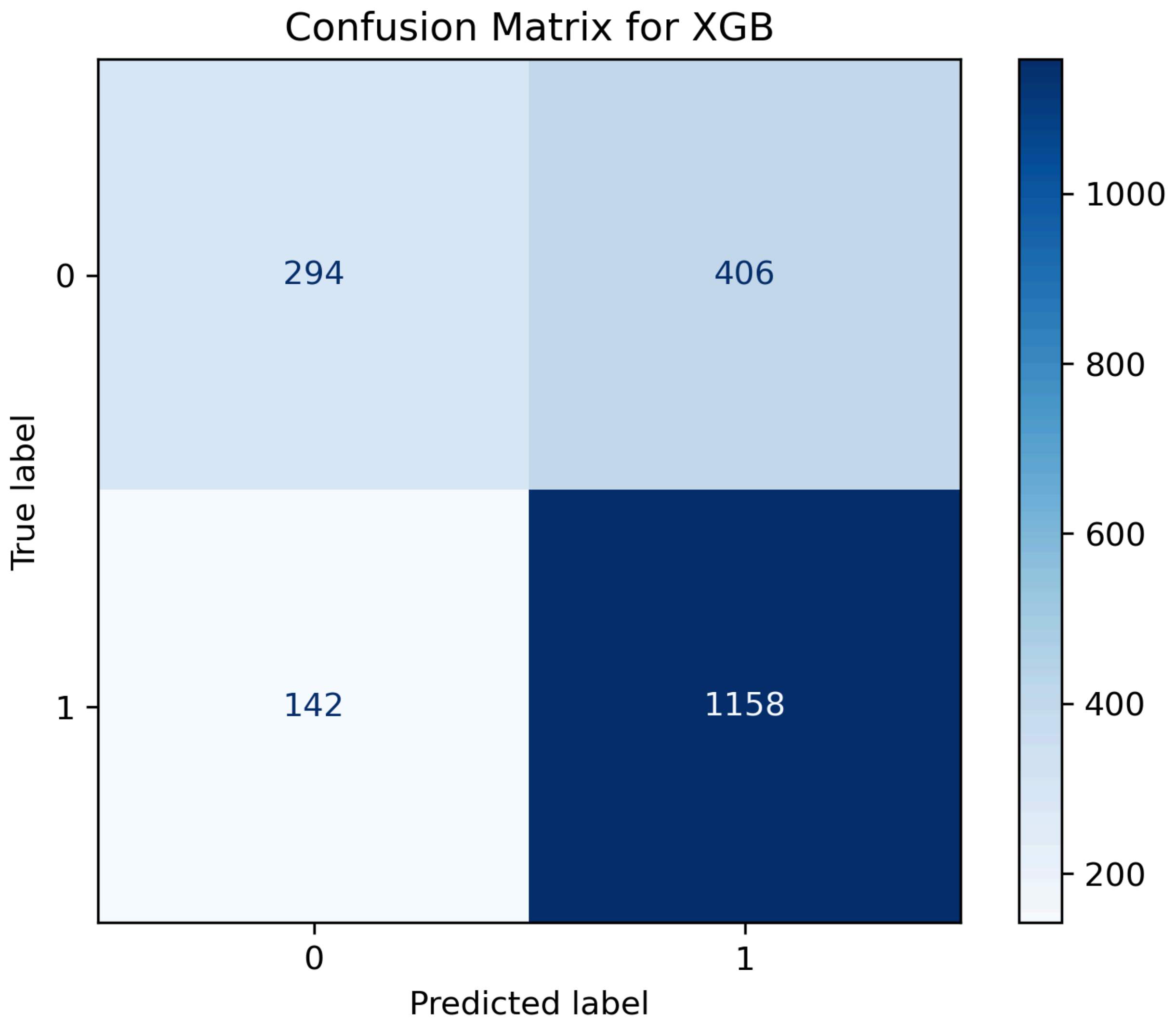

Figure 6. Confusion matrix for XGB.

XGBoost model was able to correctly predict that 1158 users have clicked the advertisement and 294 users haven't clicked the advertisement based on its training on the train set. We can clearly see how my model is a little biased in predicting the majority class 1-yes for clicks, as it predicts 406 users who have clicked but they really did not click the advertisement.

Calculating global feature importance helped me get more insights on which features contributed most to the prediction. I calculated global feature importance with SHAP, Permutation, and XGBoost's 5 different metrics - gain, weight, cover, total_gain, and total_cover. After conducting feature analysis over various metrics I was able to see that the user's age, browsing history, and device type significantly played an important role in prediction. Where in different browsing history type and device type have been varied but top common is browsing history-news. The most common least important feature I found is advertisement position as it didn't even show up in any importance metric and gender being the lowest on graphs. Below is the graph that shows the feature importance by SHAP.

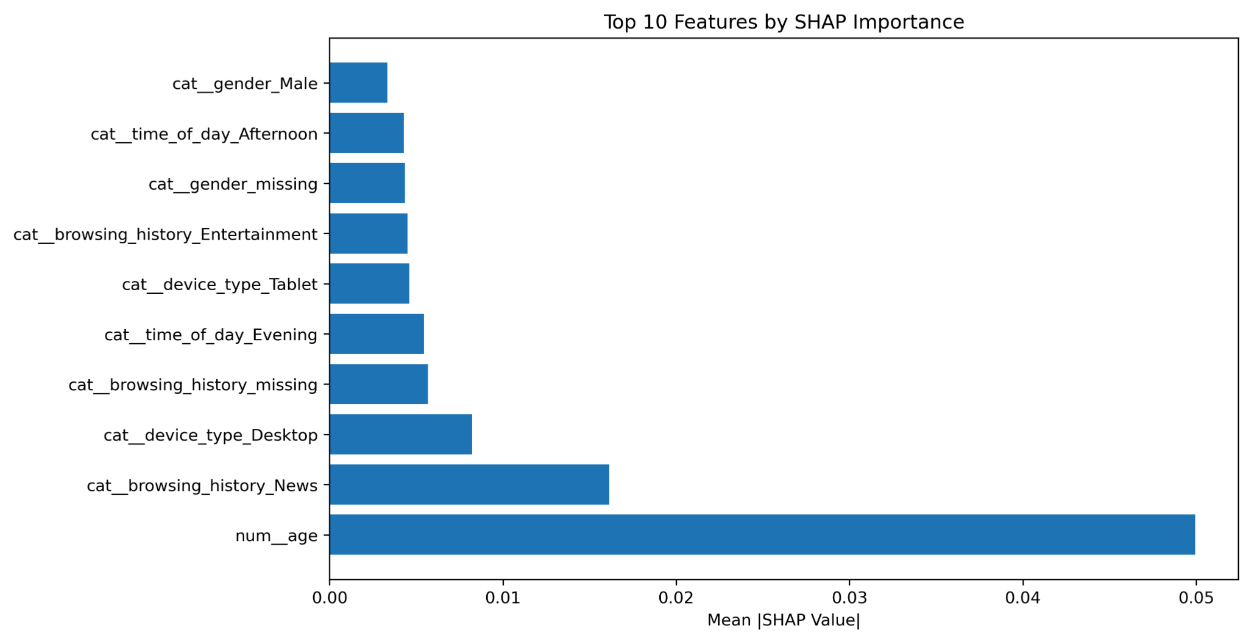

Figure 7. Top 10 Features by SHAP Importance.

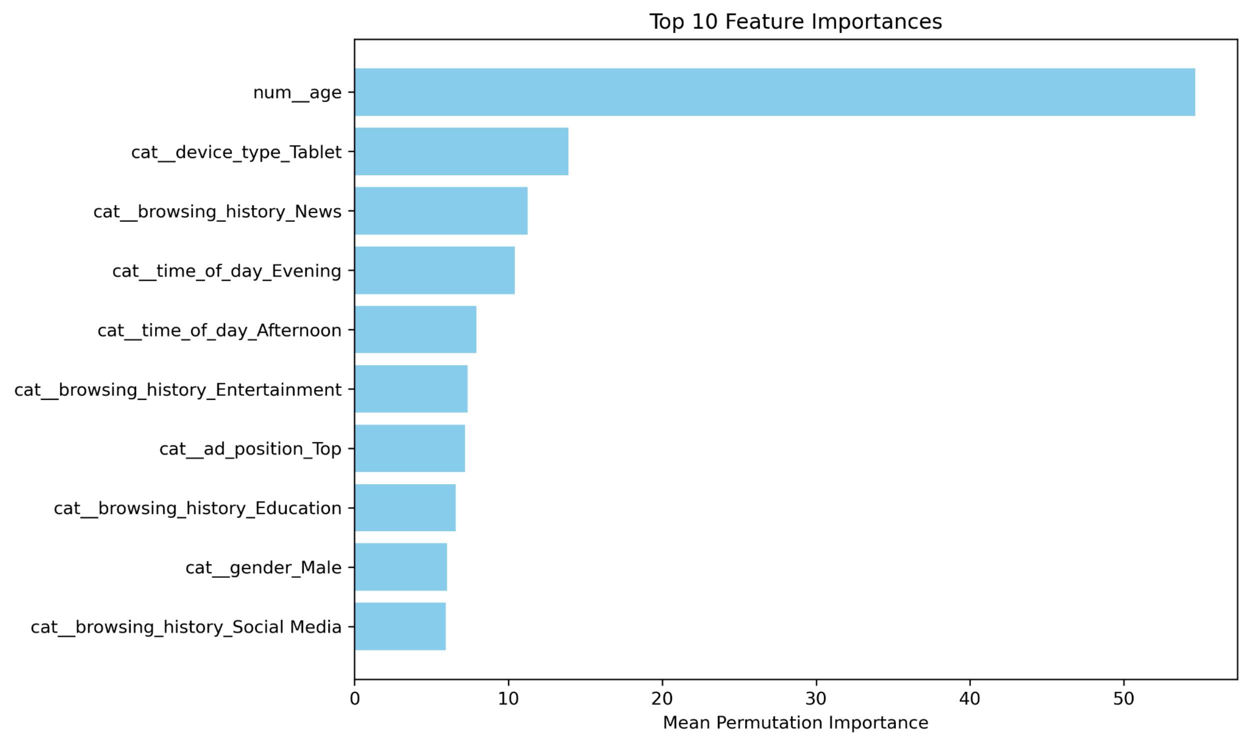

Figure 8. Top 10 Features by Permutation Importance.

To know the local feature importance and get insights into individual predictions, I calculated using SHAP values for local features for different indexes, here I would like to show my findings for the 200th index, we can see the model's average prediction across the entire dataset is 0.65 and on this specific index the features are pushing the prediction to lower end, features like device_type_Desktop is positively contributing to the prediction but age is lowering the prediction.

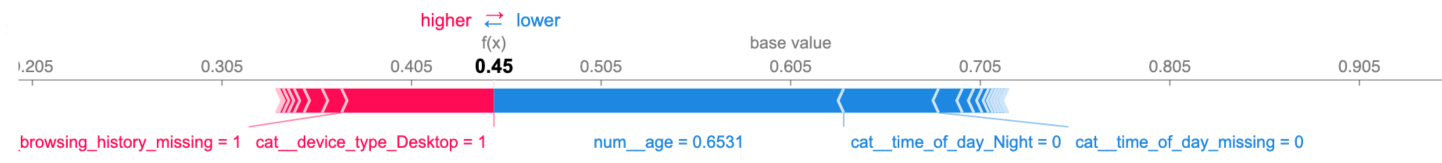

Figure 9. Local Feature Importance.

Outlook

To improve my model's interpretability I would like to first tune more hyperparameters in all the models I have tried, as due to my personal device limitation I wasn't able to explore more in-depth detail on various parameters and some other models like Random Forest and SVM. To improve my best-performing model's prediction, I want to tune more hyperparameters with refinement and specifically go in depth where my model started performing better like for example, if my model has the best parameters at 0.6 for subsample, I want to dig in deep and see the range on how it performs from 0.61…- 0.69. In future, I will also try to learn and use other machine-learning techniques and models like LightGradientBoostingMachine which can handle missing values as well. Additionally regularly keep updating data as it gets updated, assuming we can better train our model on more data and accurate data. Also, some additional data such as location, or previous history of clicks based on their searches/detailed browsing history over social media can help in making a better predictive model.

References

[1] https://www.kaggle.com/code/pavankumar4757/ad-click-prediction
[2] https://www.kaggle.com/code/ahmedashraf299/ad-click-dataset-using-randomforest-acc-72
[3] https://www.kaggle.com/code/codecavalier/ad-clicks-predictions-eda-mastery-analytics
[4]https://datascience.stackexchange.com/questions/67167/which-models-can-handle-null-values
[5]https://datascience.stackexchange.com/questions/67167/which-models-can-handle-null-values