2021

Machine Learning Capstone Project Final Report

Optimizing the Starbucks App's Promotion Offering Strategy

Elaine Liu

I. Project Overview

One of the challenges businesses face nowadays is the maintenance and refinement their ability to retain and convert customers with evolving preferences and tastes, particularly in industries where new products are constantly introduced to the market.

This project aims to explore and implement a machine learning approach to help the Starbucks App craft personalized promotion offerings, as well as drive the highest return for each offering.

This project utilizes a hybrid model combining decision trees with logistic regression (LR), a framework proposed by Facebook in the paper 'Practical Lessons from Predicting Clicks on Ads at Facebook'.[1] The decision tree model in this project uses the Light Gradient Boosting Machine (LightGBM) framework. The general idea for this hybrid model is that logistic regression alone is an efficient and easy-to-implement classification algorithm, but it only performs well on linearly-separable data and is not suitable for non-linear data. Decision tree models, on the other hand, are able to 'bin' each feature to a leaf node in a tree, thus transforming a continuous feature into a discrete feature. These tree features can then be effectively learned by logistic regression. Therefore, by combining decision trees with logistic regression, the hybrid model is able to achieve both training efficiency and modeling performance.

In addition, this project employs extensive feature engineering that captures customers' behavioral information by combining various offer and customer attribute datasets and leveraging historical transaction data.

By comparing against two benchmark models (one with a logistic regression model only, without tree features, and another with a boosted decision tree model only, as a classification model), this project shows that the hybrid model significantly outperforms both benchmark models in terms of Normalized Cross Entropy, the evaluation metric adopted in this project.

By referring to the feature importance generated by the boosted decision tree model, this project also shows that the most valuable features are those capturing historical transaction information about the customers or the product offerings.

II. Problem Statement

Starbucks sends out promotion offers to users through the Starbucks App once every few days. A promotion can be an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users may not receive any offers during certain weeks. Not all users receive the same offer. Every offer has a validity period before it expires.

---

[1] He, Xinran, Stuart Bowers, Joaquin Quiñonero Candela, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, et al. "Practical Lessons from Predicting Clicks on Ads at Facebook." Proceedings of 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining - ADKDD'14, 2014. https://doi.org/10.1145/2648584.2648589.

There are varying customer profiles and behaviors. Some customers might only be receptive to certain types of promotions but not others. Some customers might make a purchase regardless of whether they received an offer.

Because there are costs associated with each promotion, the goal of this project is to devise an optimized strategy using machine learning techniques that drives the highest return on each offer sent by identifying the offers with which customers are most likely to engage.

III. Solution Statement

Methodology

Target Value Definition

Since this project does not provide a target, to define a target variable, I first need to define what is considered a 'positive' action. In our context, a positive action is defined as when a customer views an offer after receiving it; redemption of the offer is not required.

The reason for defining viewing an offer as the positive action rather than redemption is that the viewing rate represents customers' engagement level. Even though the action of viewing an offer does not directly translate to dollar amounts for the business, the viewing rate is of the most important metrics for the business to track.

Hybrid Model Architecture

The hybrid model proposed by the Facebook paper and implemented in this project is a concatenation of boosted decision trees and of a probabilistic sparse linear classifier.

The boosted trees are used to transform features from continuous non-linear inputs to discrete categorical values – 'treating each individual tree as a categorical feature that takes as value the index of the leaf an instance ends up falling in.' [2] The transformed tree features are then treated as categorical inputs to the logistic regression model. The hybrid model architecture can be explained by the **Figure 1**[3] below:

[2] Ibid., 3
[3] Ibid., 2

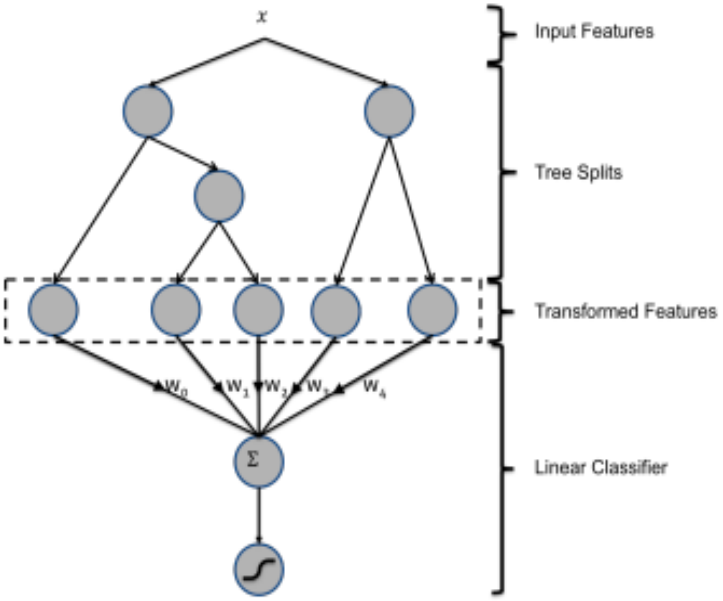

Figure 1: Hybrid model structure

Evaluation Metrics

This project uses Normalized Cross Entropy (NE) as the evaluation metric, same as in the aforementioned Facebook paper. NE is equivalent to the average log loss per impression divided by what the average log loss per impression would be if a model predicted the background click through rate (CTR) for every impression[4]. The formula is as shown below (note that the formula is slightly different from the one in the Facebook paper because the target label in this project is ∈ {0, 1}, whereas the target label in the Facebook is ∈ {-1, 1}):

$$NE = \frac{-\frac{1}{N}\sum_{i=1}^{n}(y_i \log(p_i) + (1-y_i)\log(1-p_i))}{-(p * \log(p) + (1-p) * \log(1-p))}$$

N: size of the dataset
p_i: estimated probability of viewing an offer p_i where i $= 1, 2, \ldots$ N
p: the average empirical probability of viewing an offer

Note that although there is a cost associated with every offer sent, the business would not want to miss any customers who are likely to act on the offer after receiving it, even if it means there would be a higher chance of sending offers to customers who would not be influenced by the offer. In other words, we would need to prioritize minimizing false negatives. Thus, metrics invariant to classification-threshold like AUC would not be optimal for our use case.[5]

[4] Ibid., 2

[5] Classification: ROC Curve and AUC; Machine Learning Crash Course." Google. Google. Accessed March 2, 2021. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

IV. Data Overview

Portfolio

Portfolio data contains offer IDs and metadata about each of the 10 offers. The details of the portfolio data are shown in **IX. Appendix – Portfolio**.

Profile

Profile data contains demographic data for each of the 17,000 customers. The details of the profile data are shown in **IX. Appendix – Profile**.

Transcript

Transcript data contains records for transactions, offers received, offers viewed, and offers completed. The details of the transcript data are shown in **IX. Appendix – Transcript**.

V. Implementation

Create Target Label

With a fair amount of data wrangling, each transaction was assigned a label of 1 or 0, representing whether a customer viewed an offer before transacting or not, after receiving the offer.

Among ~76,000 transactions where the customer received an offer, ~68,000 (90%) transactions show that the customer viewed the offer (label = 1) and ~17,000 (10%) transactions show that the customers did not view the offer (label = 0). Therefore, the offer view rate is about 90%.

Feature Engineering

Aside from the offer and customer features provided in the Portfolio and Profile datasets, I created additional features derived from the Transcript data that serve to uncover underlying characteristics of offers and customers that are not represented in the Portfolio and Profile datasets.

Prevent Data Leakage

One important callout is that all these features were calculated based on training (historical) data only to avoid data leakage - the model would only use past information to make predictions on future data. I divided the Transcript into training and testing - for each customer, based on the time she/he received an offer, if the time was smaller than the fourth quartile of the time period, the transactions would be included in training; otherwise, testing. These features, calculated only using training data, later are included in the testing data, so the training and testing datasets have the same feature space.

The fourth quartile is chosen to make sure that the training dataset has roughly triple the size of the testing dataset.

The following offer features were created based on training the Transcript data (details are shown in **IX. Appendix – Portfolio features based on training Transcript**):

- Average probability of each offer to be viewed (i.e. 'view_rate_portfolio')
- Average/minimum/maximum time it takes for an offer to be viewed (i.e. 'avg_duration_view_portfolio', 'min_duration_view_portfolio', 'max_duration_view_portfolio')

The following customer features were created based on training the Transcript data (details are shown in **IX. Appendix – Profile features based on training Transcript**):

- Average probability of each customer to view an offer (i.e. 'view_rate_profile')
- Average/minimum/maximum time it takes for a customer to view an offer after receiving it (i.e. 'avg_duration_view', 'min_duration_view', 'max_duration_view')
- Average/minimum/maximum amount each customer spends on a transaction (i.e. 'avg_amount', 'min_amount', 'max_amount')
- Average number of times each customer transacts (i.e. 'avg_trx_cnt')

Data Preprocessing

Log-transformation skewed features

I used log-transformation to transform the skewed features to approximately conform to normality. Although feature normalization is not likely to have an impact on the performance of the decision tree model, the benchmark model - logistic regression - is among the gradient descent based algorithms and would require the data to be scaled so the gradient descent can move smoothly toward the local minimum.

Apply One-Hot-Encoding

One-hot-encoding is applied on 'offer type' and 'gender' features. Although LightGBM offers good accuracy with integer-encoded categorical features and performs better than one-hot encoding, I chose one-hot-encoding to satisfy the requirement of the logistic regression model.

Remove highly correlated features

The issue with multi-collinearity among features does not affect the performance of LightGBM because the algorithm is based on Boosted Trees. Additionally, the regularization implemented by default in sklearn logistic regression also addresses the issue of multi-collinearity. Still, highly-correlated features were removed to reduce the feature space.

Figure 2 below shows the features left for modeling and their pairwise correlation with other features.

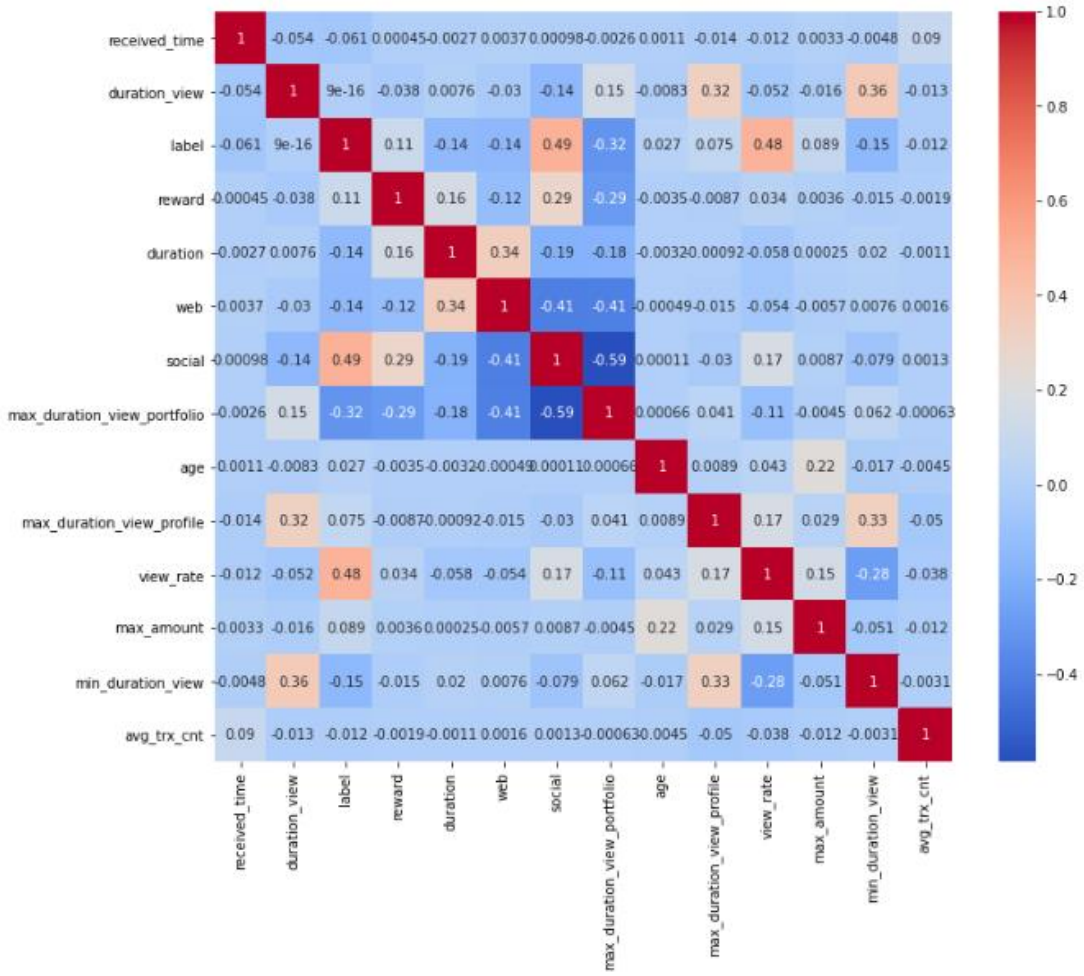

Figure 2: Correlation heat map for features left for modeling

VI. Modeling

Data

I combined the Transcript, Portfolio, and Profile data, along with additional features created, as the final dataset for the model to train and predict.

Hybrid Decision Trees & Logistic Regression Model Implementation

The decision trees model (GBDT) in this project uses the LightGBM framework and applies Bayesian Optimization, optimized on the mean of the binary log loss, to find the best hyperparameters.

The trained GBDT transforms the original features to categorical features for both training and testing data - each individual tree is a categorical feature that takes as value the index of the leaf

an instance ends up falling in[6] (i.e. tree features). Next, one-hot encoding is implemented to transform original tree features into a binary vector with 0 and 1 to indicate whether an instance ends up in a leaf. The final features to be included as input to the logistic regression is a combination of the one-hot encoded features based on tree features and the original features.

The goal of the GBDT model is not to predict the target value, but to transform features to later be included as inputs to the logistic regression.

Figure 3 below is the feature importance for the feature inputs to the decision trees model. Notably, the top 5 features that weighed more tended to be features generated from the historical Transcript data (i.e. 'max_duration_view_profile', 'max_duration_view_portfolio', 'view_rate (profile)', 'max_amount'). This project arrived at the same conclusion as in the Facebook paper - 'the most important thing is to have the right features: those capturing historical information about the user or ad dominate other types of features.'[7]

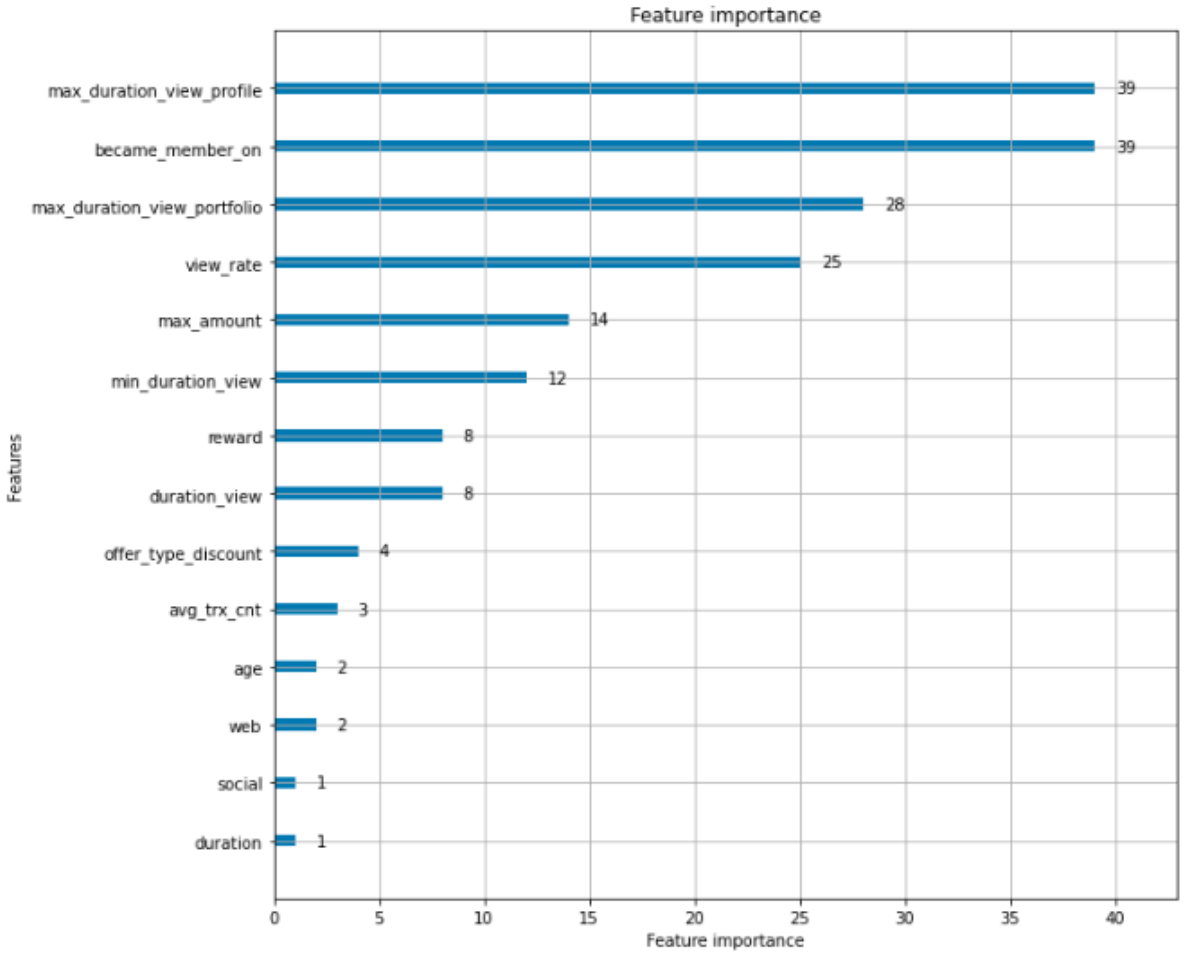

Figure 3: Feature importance generated by GBDT

[6] Ibid., 2

[7] He, Xinran, Stuart Bowers, Joaquin Quiñonero Candela, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, et al. "Practical Lessons from Predicting Clicks on Ads at Facebook." Proceedings of 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining - ADKDD'14, 2014. https://doi.org/10.1145/2648584.2648589.

Benchmark Model Implementation

The benchmark models are GBDT and LR models trained on original features that make predictions on testing data in isolation.

VII. Model Performance Result

Model Structure	Normalized Entropy (Training)	Normalized Entropy (Validation)	Normalized Entropy (Testing)
GBDT + LR	0.11%	0.11%	**0.31%**
GBDT only	0.35%	0.36%	**1.89%**
LR only	45.74%	45.64%	**83.00%**

VIII. Conclusion

Based on the model performance result, comparing the two benchmark models, the GBDT-only model achieves a significantly lower Normalized Entropy than the LR-only model. As such, the GBDT-only model is remarkably more performant than LR in this specific task. However, even though the GBDT-only model already achieves small Normalized Entropy, the hybrid (GBDT + LR) model is able to further bring down the Normalized Entropy from 1.89% to 0.31%, an impressive relative improvement to the GBDT-only model.

This project arrives at the same conclusion as the Facebook paper that combining the decision tree based model with probabilistic linear classifiers significantly improves the prediction accuracy of a model using just the probabilistic linear classifiers alone.

IX. Appendix

Portfolio

	Count	Explanation
Offer Type		
BOGO	4	
Discount	4	
Informational	2	
Difficulty		Minimum required spend to complete an offer
0	2	
5	2	
7	1	
10	4	
20	1	
Reward		Reward given for completing an offer
0	2	
2	2	
3	1	
5	3	
10	2	
Duration		Time for offer to be open, in days
3	1	
4	1	
5	2	
7	4	
10	2	
Channels		Whether the offer is sent through web email, mobile, or social channels. Offers can be sent through multiple channels
Email	10	
Mobile	9	
Web	8	
Social	6	

Profile

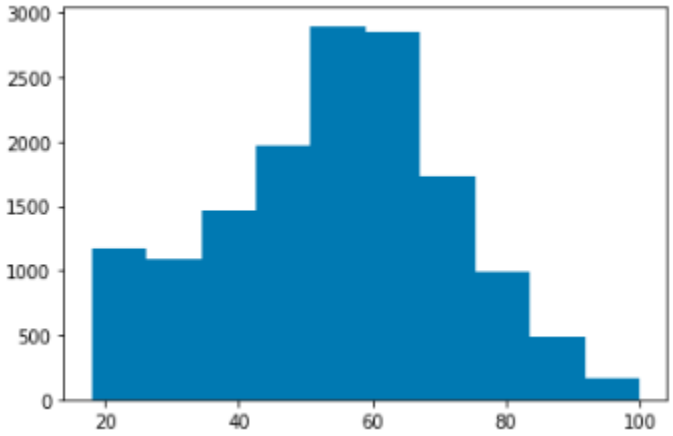

Figure 4: Distribution of customer age (extreme values/outliers removed)

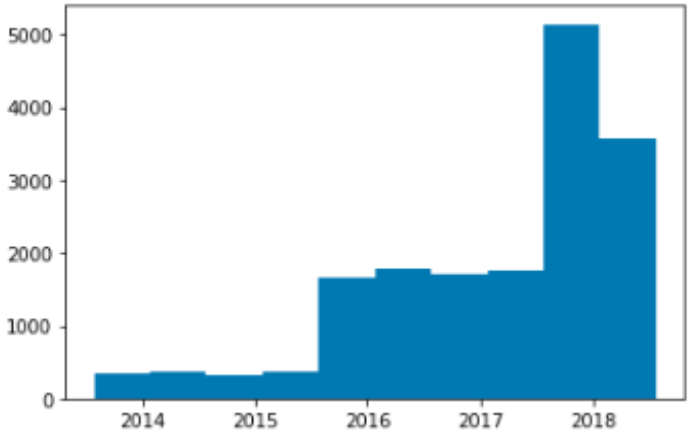

Figure 5: Distribution of when customers became members

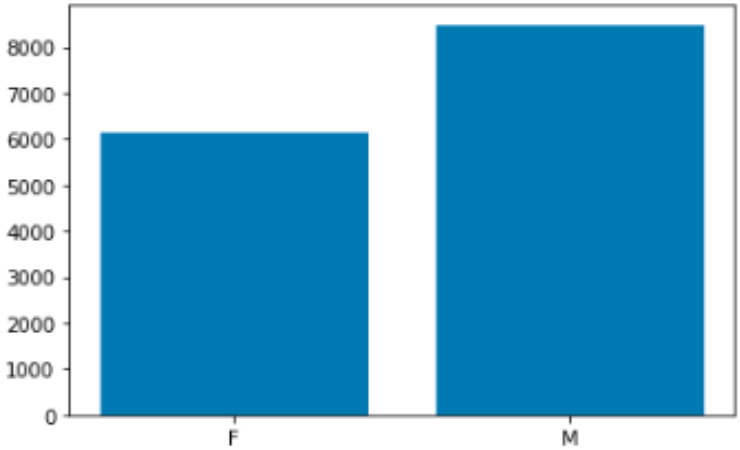

Figure 6: Distribution of customer gender

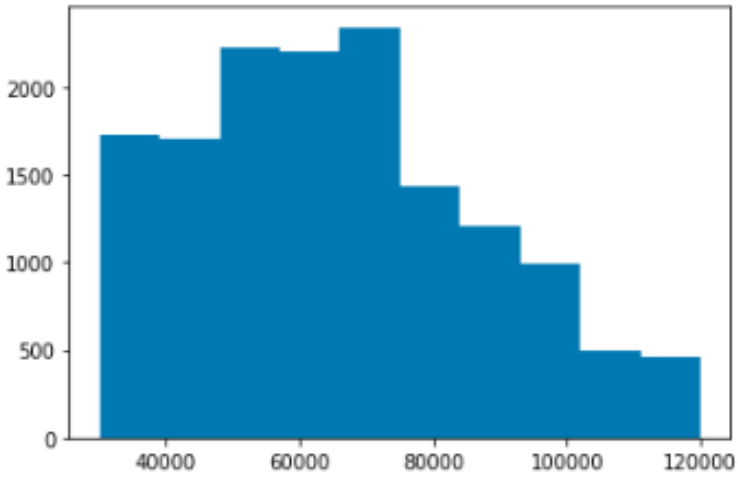

Figure 7: Distribution of customer income

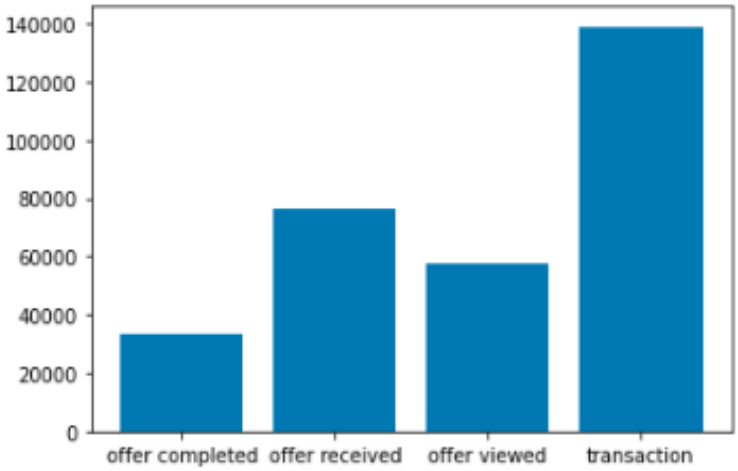

Figure 8: Distribution of events

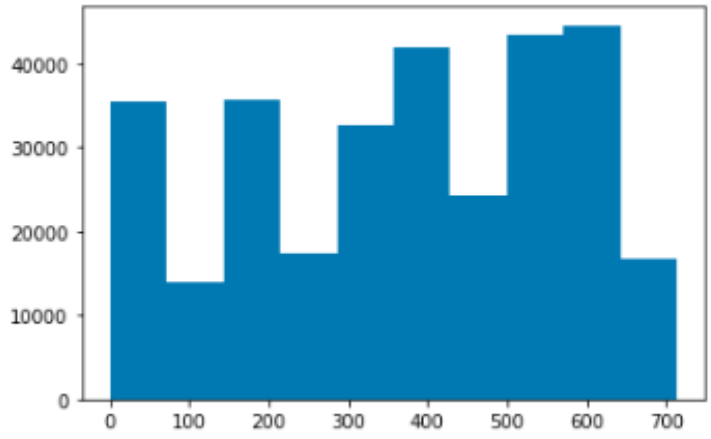

Figure 9: Distribution of time in hours since the start of the test

Transcript

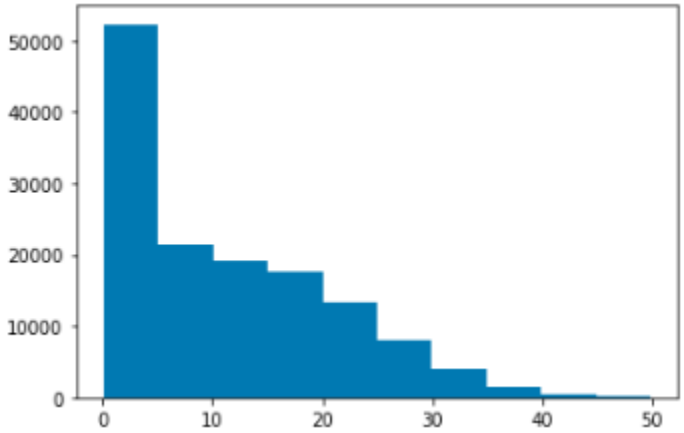

Figure 10: Distribution of transaction amount (extreme values/outliers removed)

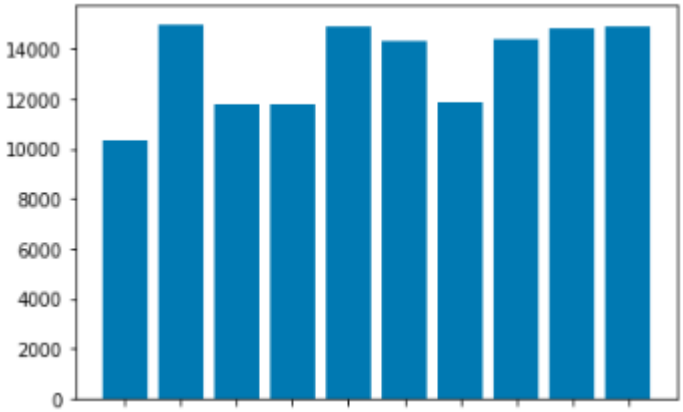

Figure 11: Distribution of offers used in transactions

Portfolio features based on training Transcript

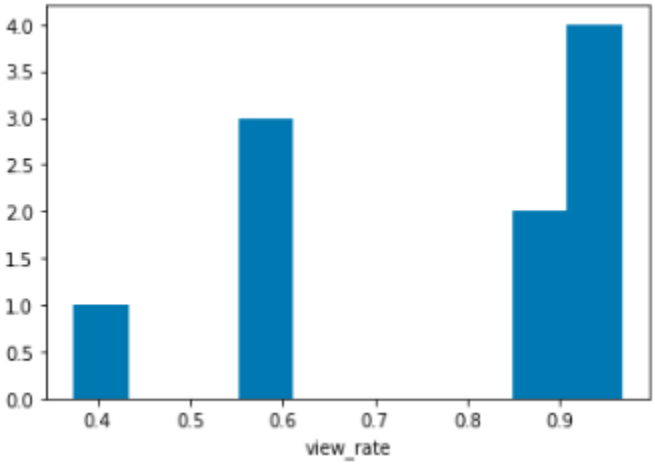

Figure 12: Average view rate of each offer

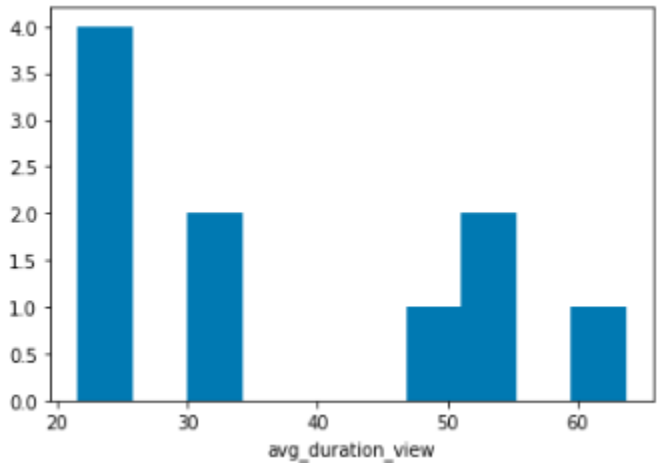

Figure 13: Average duration each offer was viewed

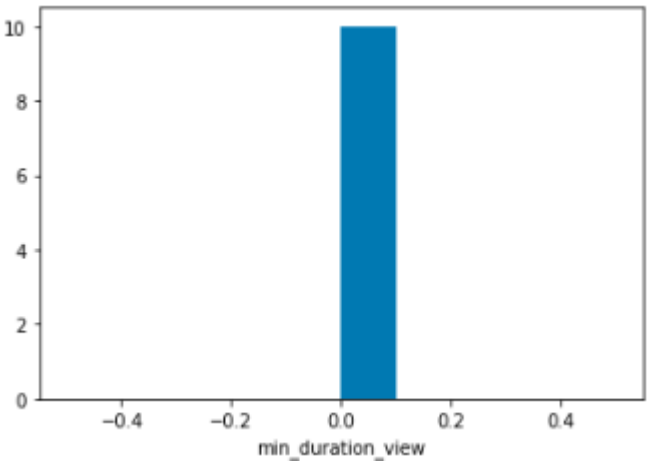

Figure 14: Minimum duration each offer was viewed

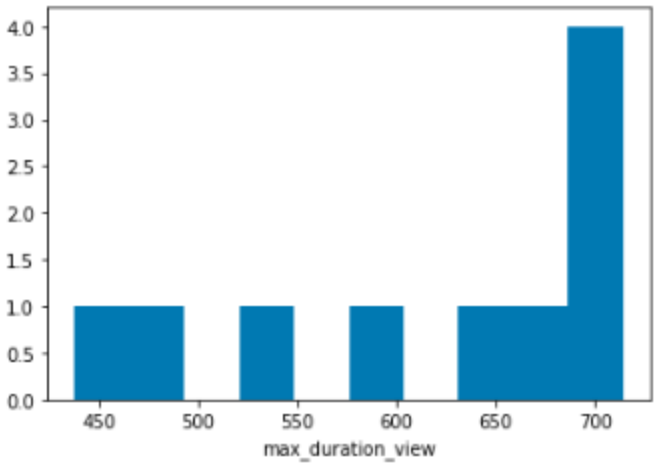

Figure 15: Maximum duration each offer was viewed

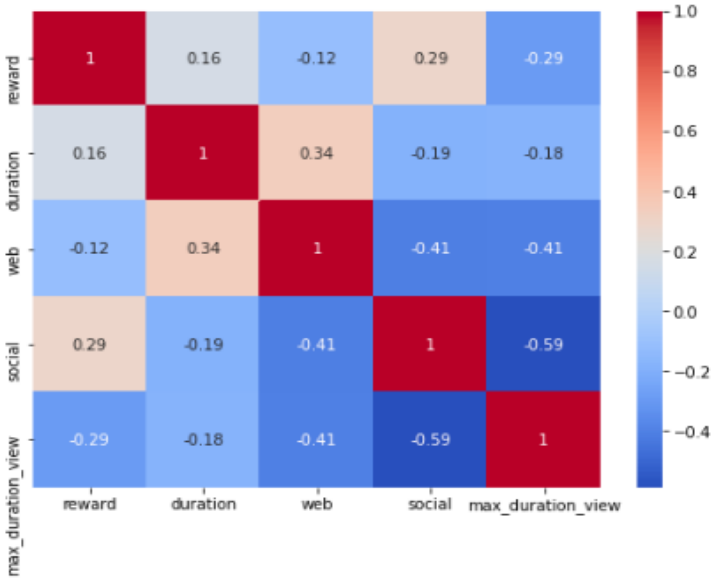

Figure 16: Correlation among all offer features

Profile features based on training Transcript

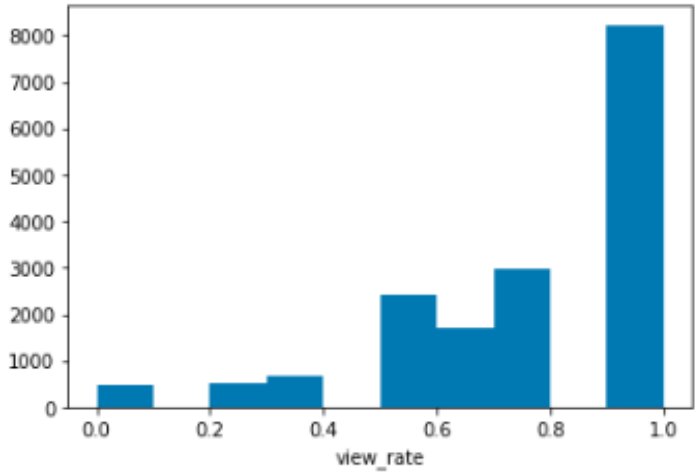

Figure 17: Average view rate of customers

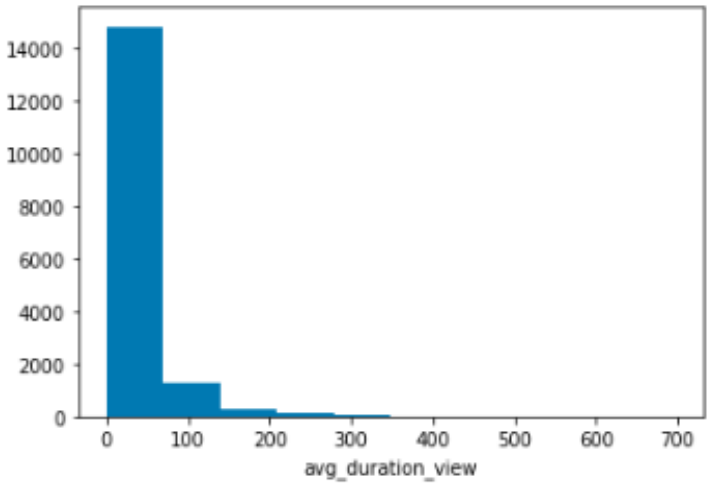

Figure 18: Average duration for which each customer viewed an offer

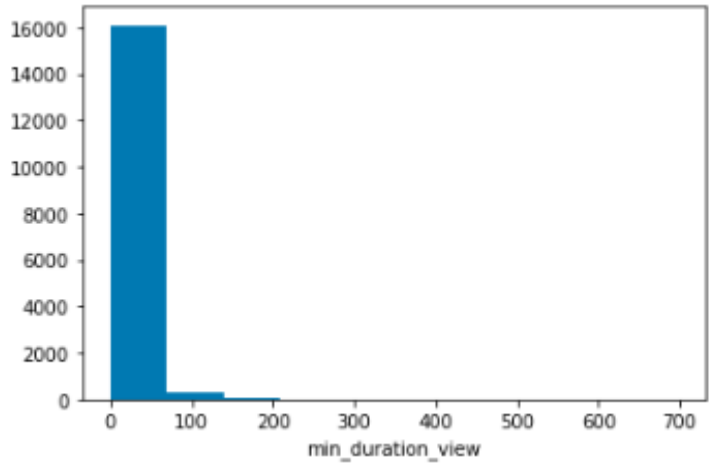

Figure 19: Minimum duration for which each customer viewed an offer

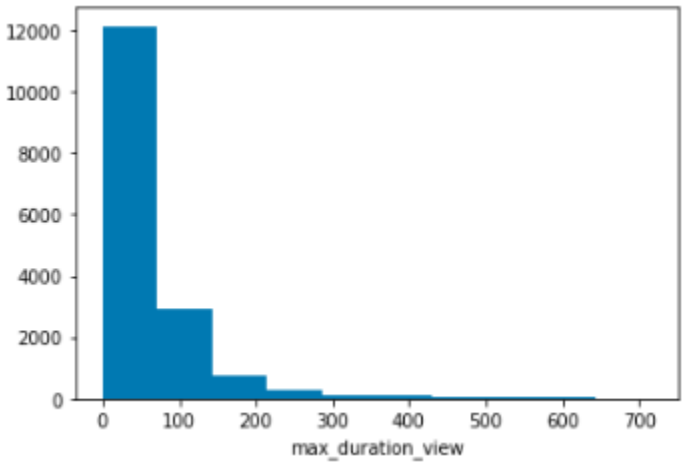

Figure 20: Maximum duration for which each customer viewed an offer

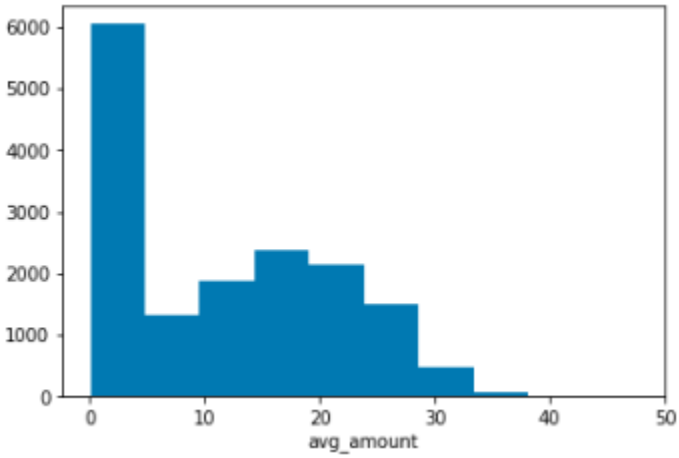

Figure 21: Average amount each customer spent on a transaction

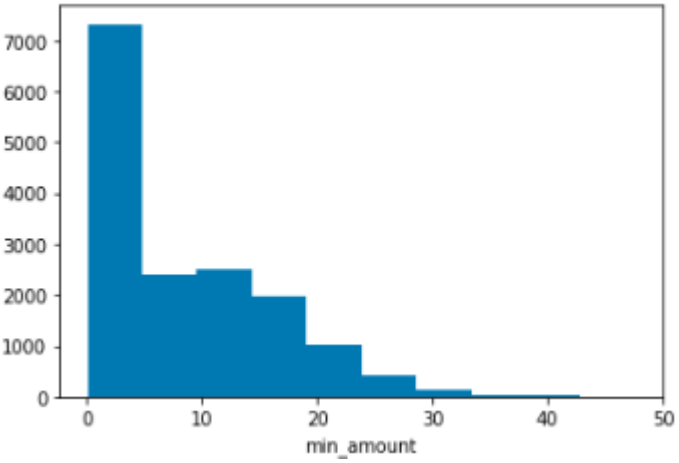

Figure 22: Minimum amount each customer spent on a transaction

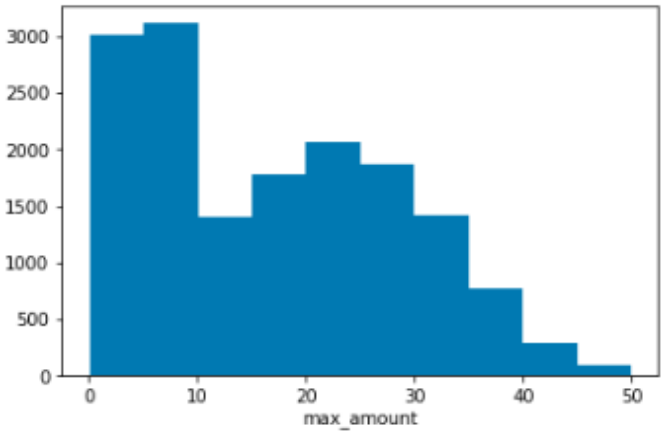

Figure 23: Maximum amount each customer spent on a transaction

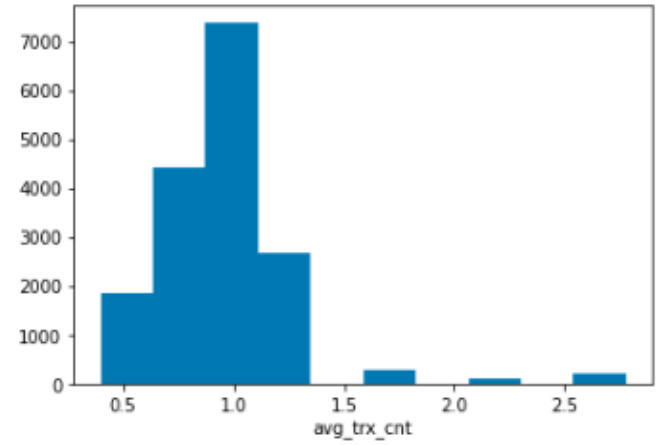

Figure 24: Average number of times a customer made a transaction

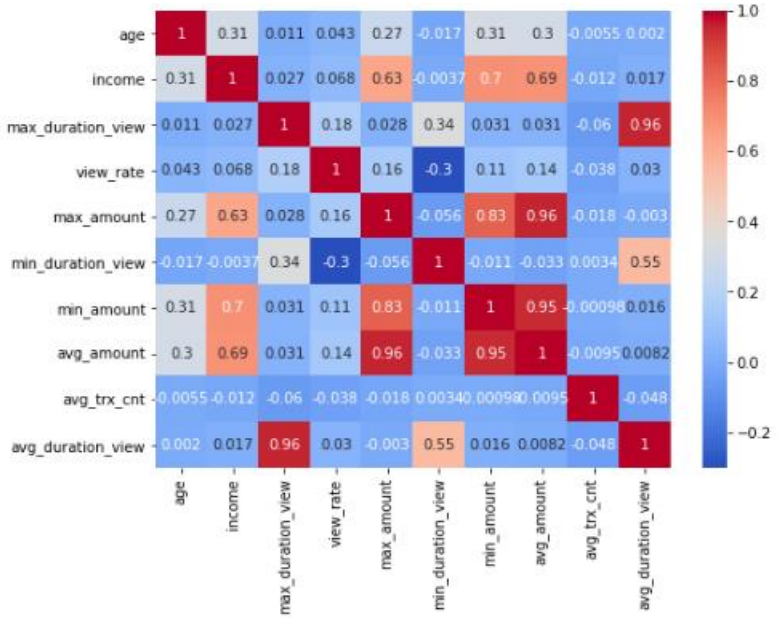

Figure 25: Correlation among all customer features