starbucks XGB

February 21, 2020

Table of Contents

0.1 Project Overview

This project revolves around the Starbucks dataset provided through Udacity. I chose this problem because I have some experience in predicting customer outcomes from self-reported data as it relates to product use and engagement. I was also interested in the extent that demographics will actually have in offer completion. Does income or age matter at all in terms of who completes offers or who simply walks in a buys something.

From the Udacity project outline:

Your task is to combine transaction, demographic and offer data to determine which demographic groups respond best to which offer type.

- As such this project will try to predict each of the four event types: transact, complete offer, view offer, and receive offer. With a model that can predict transaction, completion, view, and receive then we can see what types of features have predictive power for a given outcome.
- It machine learning terms, this is a multilabel classification task. Tree boosted algorithms such as XGB perform very well specially given the size of the dataset and relatively large number of features.

0.1.1 Data and Input

The datasets are provided by Udacity and Starbucks. They include: - profile.csv - portfolio.csv - transcript.csv

These will provide the basis for the multilabel classifier. No further inputs will be needed.

0.1.2 Problem Statement

I will build a tree boosted algorithm to predict who will complete and offer, who will transact, and who will view/recieve but not complete.

- These are major events that can be tracked directly with the data. If a customer chooses to not opt into an offer and completes is anyway, that is only a win for business as they have spent more money and recieved $0 in reward. It also tells us something about their purchase habits. If they only transact, then again they have spent money with the need for reward. Lastly, if they complete an offer after viewing, then they are responding well to advertising and are likely spending more, or at least the same, amount of money to receive the incentive.
- Additionally, by knowing who is likely to complete, transact, view, or recieve only then we can analyze the business practices around duration, marketing, and difficulty to determine changes in engagment for those who do not complete or transact.

Once a model is build and evaluated, we can view the feature importance to understand what features are important to a given outcome.

- Feature importance will be a great window into know which of the many data points collected hold the most predictive power for the each class label.

0.1.3 Expected Result

I would expect this model to be able to predict fairly well the transaction and completion classes as they are very distinict and likely correalate to very different habits of a buyer. Offer view and offer receive are highly similar in root behavior. Your phone pings you with a push notification, you look at the offer, you get distracted and forget completely. In that fraction of a second you have received and viewed the offer and gone about your day. This categoricall different from opting into an offer, using the app to purchase towards that offer and then (possibly) cashing in on the reward.

- Expectations are low for predicting views/receives in terms of accuracy, precision, and recall
- Expectations are high for predicting completion and transaction in terms of accuracy, precision, and recall

0.2 Metrics

The primary metrics that will evaluate this model are accuracy, precision, and recall.

- Classification of views/receives of offer will likely have some form the precision/recell trade-off. Either the model will have high precision- it's percent correct, and low recall- it's ability to find cases, or the reverse.
- Classification of transaction and completion will likely be very good. The class labels are fairly well balanced so we would expect precision, accuracy, and recall to be well balance if the model can predict classes well.

0.3 Evaluation

XGB multilabel classifier preformance cited below. As expected, transactions and completions are well predicted on the hold-out set. For offer views and received, the model cannot predict these classes very well. Further work could be completed to determine a better solution for disecting these groups.

```
           precision    recall  f1-score   support

       0       0.88      0.17      0.28      7546
       1       0.47      0.97      0.63      5689
       2       1.00      1.00      1.00      3416
       3       1.00      1.00      1.00     14003

accuracy                           0.79     30654
```

0.4 Data Preparation

After combining all the dataset for profile, transcript, and portfolio we have a dataframe with that is ready for analysis. Below is an excerpt from that dataframe. I suppressed the customer and offer ids for sake of space saving, but they are contained with the dataframe.

```
   age  became_member_on gender   income           event  time  reward_amount  \
0  118          20170212   None      NaN  offer received   168            NaN
1  118          20170212   None      NaN    offer viewed   216            NaN
2   68          20180426      M  70000.0  offer received     0            NaN
3   68          20180426      M  70000.0    offer viewed    18            NaN
4  118          20170925   None      NaN  offer received   408            NaN

   transaction_amount              channels  difficulty  duration offer_type  \
0                 NaN  [web, email, mobile]        10.0       7.0   discount
1                 NaN  [web, email, mobile]        10.0       7.0   discount
2                 NaN  [web, email, mobile]        10.0       7.0   discount
3                 NaN  [web, email, mobile]        10.0       7.0   discount
4                 NaN  [web, email, mobile]        10.0       7.0   discount

   reward
0     2.0
```

```
1     2.0
2     2.0
3     2.0
4     2.0
```

0.5 Categorical encoding scheme

0.5.1 Channels column

The channels column wil be converted to length. All channels have `[web, email]` which will be two. `[web, email, mobile]` is three, and `[web, email, mobile, social]` is now 4.

0.5.2 Offer type column

The offer types are also numeric now `bogo` = 1, `discount` = 2, and `informational` = 3

0.5.3 Gender column

`Male` = 1, `Female` = 3, `Other`=3, and None = 0 Other and none may be in some way linked via identity, but I'll keep them separate in the event that it is significant.

0.5.4 Target: Event encoding

Offer recieved and offer viewed, while important for demographics, is probably not all that important in so much as they both may not convert to a sale or offer completion. As such, I'll group them together and try to predict completion, transaction, or none.

TARGET Encoding: `offer received = 0`, `offer viewed = 1`, `offer completed = 2`, `transaction` = 3

0.5.5 Offer id and customer id

These unique identitfiers will be replaced by integer hash ids so they can be tracked.

0.6 Imputation

Income value will be filled with the median value only about 11% of the income data is missing, so this is both safe and unlikely to bias the model in a meaningful way. To be on the safe side, I'll fill the values inside the cross validation folds so as to spread out the filled values relative to the CV fold.

0.6.1 Filling NaN values

```
age                        0
became_member_on           0
gender                     0
id_customer                0
income                 33772
event                      0
time                       0
reward_amount         272955
transaction_amount    167581
offer_id                   0
channels              172532
difficulty            172532
duration              172532
offer_type                 0
reward                172532
dtype: int64
```

Channels, difficulty, duration, and reward all have the same number of missing. Reward amount can be zeroed for Null values, as those were likely 0 reward given. I assume that Null values for transaction amounts were either zero dollars or some other interaction that was outside the ability to track. Thus this remaining values will be filled with zeros.

```
age                       0
became_member_on          0
gender                    0
id_customer               0
income                33772
event                     0
time                      0
reward_amount             0
transaction_amount        0
offer_id                  0
channels                  0
difficulty                0
duration                  0
offer_type                0
reward                    0
dtype: int64
```

Our final, completed dataframe is shown below for the first five observations. Again, customer and offer id are suppressed for readablity.

```
   age  became_member_on  gender   income  event  time  reward_amount  \
0  118          20170212       0      NaN      0   168            0.0
1  118          20170212       0      NaN      1   216            0.0
2   68          20180426       1  70000.0      0     0            0.0
3   68          20180426       1  70000.0      1    18            0.0
4  118          20170925       0      NaN      0   408            0.0

   transaction_amount  channels  difficulty  duration  offer_type  reward
0                 0.0       3.0        10.0       7.0           2     2.0
1                 0.0       3.0        10.0       7.0           2     2.0
2                 0.0       3.0        10.0       7.0           2     2.0
3                 0.0       3.0        10.0       7.0           2     2.0
4                 0.0       3.0        10.0       7.0           2     2.0
```

0.7 Multilabel Classification

The goal for this project is to predict who will transact, complete offers, receive offer, and view offers. This is a multilabel classification supervised learning task with four classes stated above.

0.7.1 Benchmark Model - Random Forest Classifier

The performance metrics for the benchmark Random Forest Classifier are detailed below

```
{'model_id': 0, 'model': RandomForestClassifier(bootstrap=True, class_weight=None, criterion='
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False), 'f1_score': {'mean': 0.5848, 'sd': 0.0},
```

The benchmark model is pretty poor in terms of precision, recall, and f1 score. The accuracy is okay at 85%.

We will revisit this benchmark model two more times. Once after training the XGB model on the training data and again when evaluating the test data.

0.7.2 Random Search XGB model

A random search was performed for hyperparameters of the XGB model and returned the top result. They are as follows

```
Out[44]: {'model_id': 1,
          'model': XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                         colsample_bytree=0.6799999999999999, gamma=4.6,
                         learning_rate=0.017857142857142856, max_delta_step=0, max_depth=9,
                         min_child_weight=7, missing=None, n_estimators=379, n_jobs=1,
                         nthread=None, objective='multi:softprob', random_state=9450,
                         reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                         silent=True, subsample=0.89),
          'f1_score': {'mean': 0.8697, 'sd': 0.0},
          'recall': {'mean': 0.7777, 'sd': 0.0056},
          'precision': {'mean': 1.0, 'sd': 0.0},
          'specificity': {'mean': 1.0, 'sd': 0.0},
          'balanced_accuracy': {'mean': 0.8888, 'sd': 0.0028},
          'accuracy': {'mean': 0.9581, 'sd': 0.0011}}
```

We see that this is a significant increase in performance over the benchmark metrics.

0.7.3 Model Evaluation - Test Set

Model performance metrics for the test set are describe below. The first output is the confusion matrix for eacf class and the second output is the classification report displaying metrics for recall, precision, accuracy, and f1 score

```
[[ 1274  6272     0     0]
 [  176  5513     0     0]
 [    0     0  3416     0]
 [    0     0     0 14003]]

              precision    recall  f1-score   support

           0       0.88      0.17      0.28      7546
           1       0.47      0.97      0.63      5689
           2       1.00      1.00      1.00      3416
           3       1.00      1.00      1.00     14003

    accuracy                           0.79     30654
   macro avg       0.84      0.78      0.73     30654
weighted avg       0.87      0.79      0.76     30654


accuracy score: 78.97%
```

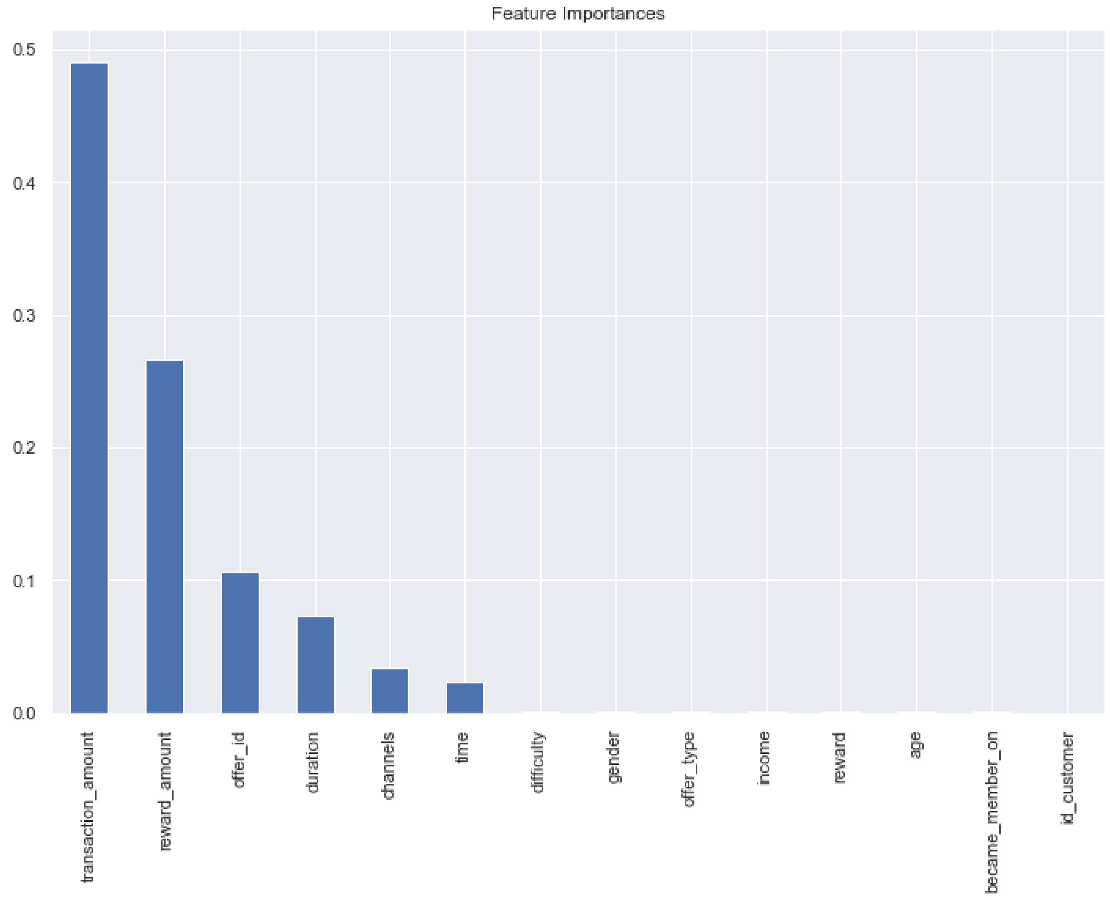
Events are numeric as such:
`offer received` = 0, `offer viewed` = 1, `offer completed` = 2, `transaction` = 3

It is expected that the model will lose some preformance under new observations. This is especially the case when the income values have not been imputed into the test set nor has the test set been scaled in the same way that training set has. In a production environment we would want to continue that imputation in a per batch manner to maintain the same conservative approach for filling missing income values.

This is a great result for predicting something like a transaction without offer and completing offers. If the goal is to find who will buy without offer and who will buy with an offer, then this is pretty good.

0.7.4 Feature Importance to Class Label

The plot below shows the relative importance that each feature in the dataset holds to overall prediction of class label. We see that business level features are the strongest predictors for each class not demographics.



The top features that seem to dictate the likely hood a customer will transact, complete offer, view, or receive. One of the strongest predictors is reward amount, duration, the specific offer, and transaction amount. Secondary predictors apprear to be offer type, marketing saturation, and difficulty. The second and third models evaluated tell a similar story. Since offer id carries a heavy predictive power, future work could be to determine the significance of each offer for completion. Reward amount was a top predictor as well, so that is also an avenue for further work. Which amount if optimal for completion.

Interestingly enough; age, gender, income, and offer type carry no (or little) predictive power. There is some literature support this idea re: Martens and Provost (2011); at a certain point sociodemogrpahics add no additional predictive power when compared to transaction activity.

1 Conclusion

The top features that seem to dictate the likely hood a customer will transact, complete offer, view, or receive. One of the strongest predictors is reward amount, duration, the specific offer, and transaction amount. Secondary predictors apprear to be offer type, marketing saturation, and difficulty. The second and third models evaluated tell a similar story. Since offer id carries a heavy predictive power, future work could be to determine the significance of each offer for completion. Reward amount was a top predictor as well, so that is also an avenue for further work. Which amount if optimal for completion.

Interestingly enough; age, gender, income, and offer type carry no (or little) predictive power. There is some literature support this idea re: Martens and Provost (2011); at a certain point sociodemogrpahics add no additional predictive power when compared to transaction activity.

1.1 Citation

- Martens, D., & Provost, F. (2011). Pseudo-social network targeting from consumer transaction data. Workinng paper CeDER-11-05, New York University - Stern School of Business.