Definition

Project Overview

Offer optimization is an interesting marketing problem for any consumer product based company and can have a direct impact on the revenue and the cost incurred by the company. Sending less offers would remove competitive edge and could result in less activity from customers whereas sending too many offers can be overkill and might even have a negative effect on the customer. It is important to provide the right offer to each user at an individual level to maximize profits.

Historically marketing has been done at a universal level across all customers without any personalization. However, this is really inefficient and these days personalized marketing campaigns are becoming more prevalent. Personalized marketing is done by understanding the different groups of customers through data and providing them with the right offer at the right time. One of the ways in which this problem is solved is through customer segmentation. For example, segmenting customers based on demographics and providing a different offer to each of these segments could help drive revenues for a company. Sophisticated segmentation could further be done on the order patterns of the customers using unsupervised techniques such as kmeans to make the offer strategy more effective. However, even segmentation is limited in its strength since the customers within a segment also could be different and might require different offers. To address this issue a more supervised machine learning approach is used where using historical data whether a given offer would be effective on a customer is predicted. This is the approach I would be using as well.

Through this project I hope to gain insights on the patterns of consumers in a consumer product company. I believe as I go through solving this problem, I would understand the salient features of the data and the assumptions that I need to keep in my mind. I would also understand the techniques that work on this kind of data which would give me the confidence to solve future problems in the same domain and help me become a holistic data scientist. I will also be joining as a data scientist in the marketing tech team of a food delivery company early next year, hence I feel this project would be a great learning opportunity for me.

Problem Statement

The aim of this project is to predict which users would respond to a particular offer. Hence, for a user-offer combination I would try to predict a 1 if that offer is going to be effective on the user or 0 if the offer is not effective. The definition of effective here means that the user has to both view and complete the offer before it expires.

Evaluation Metrics
Since this is a classification problem the metric that can be used to evaluate the model can be accuracy. Also, since the cost of a false negative (Not giving an offer given that the customer would have responded positively) high here a combination of recall and accuracy also can be considered as a metric. Furthermore, the auc score of the classification model is also considered.

Analysis

Data Exploration

The dataset used contains information on 17000 users and their activities with the Starbucks app over a period of 30 days. The activities include transactions made by the user, offers received by the user, offers viewed by the user, and offers completed by the user. An offers dataset provides details on the different offers that have been sent to the users regarding their durability and difficulty. A users dataset provides information about their demographics. The same offer could be offered to a user multiple times during this period. In these cases, I consider only the last offer that was sent to the user in the case when this user-offer combination hasn't been effective even once. For cases where a user-offer combination is effective I consider the first offer that was sent to the user. An effective offer is an offer that has been viewed and completed by the user.

During my exploration of the dataset I also found that around 4000 user offer combinations haven't still been completed. Hence, I have removed these combinations from my model as we do not know if in the future these offers would be effective or not.

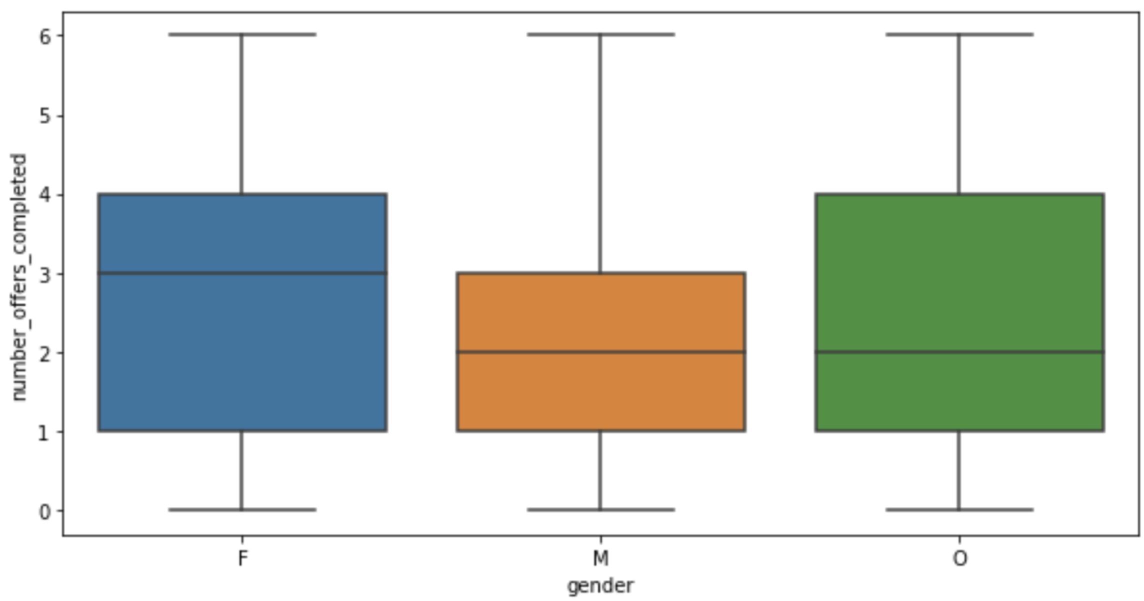
The demographics dataset has around 2000 rows in which the gender, income and the age of the user is unknown. I have imputed the missing gender with 'O'. I have further imputed the age and income of these users with their respective median values of the 'O' gender.

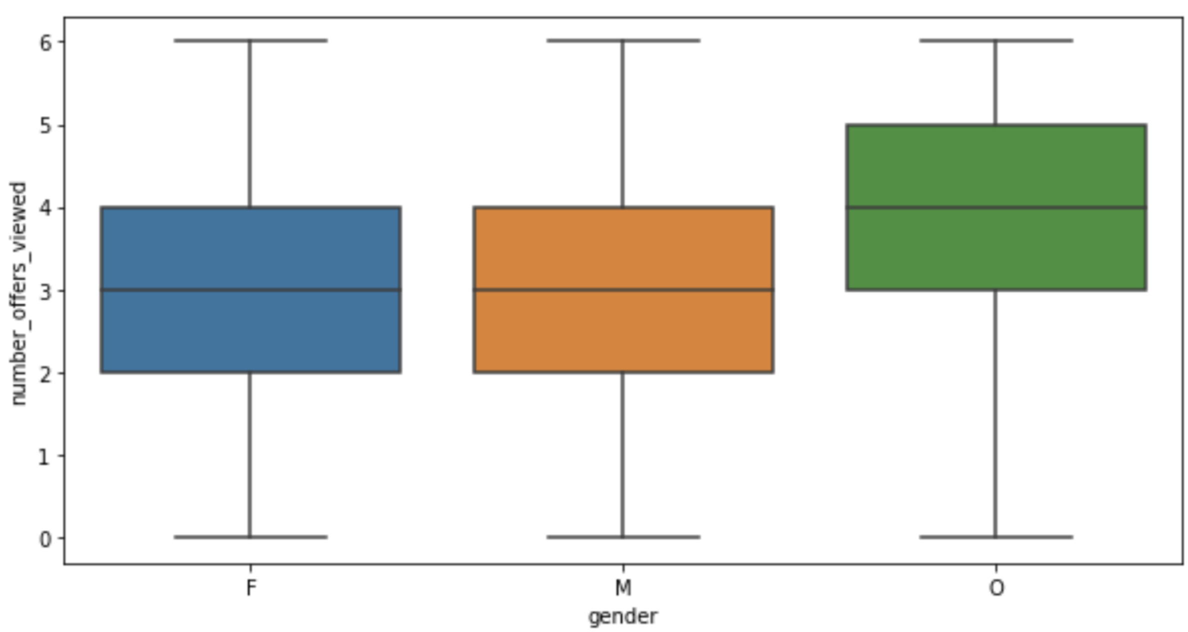
The following are the general statistics I have about the data

1. Number of females - 6129, Number of males - 8484 and their respective average incomes are 71,300$ and 61,200$. Their respective average age is 58 and 52.
2. There are a total of 10 different offers of which 4 are bogo type, 4 are discount type, and 2 are informational.
3. Average duration of bogo offers - 6 days, Average duration of discount offers - 8.5 days, Average reward for bogo offers - 7.5$, Average reward for discount offers - 3$
4. In the 30 days of the given data a person has received 4.4 offers on an average, viewed 3.3 offers on an average, and completed 2 offers on an average
5. The average amount spent per transaction is 13$

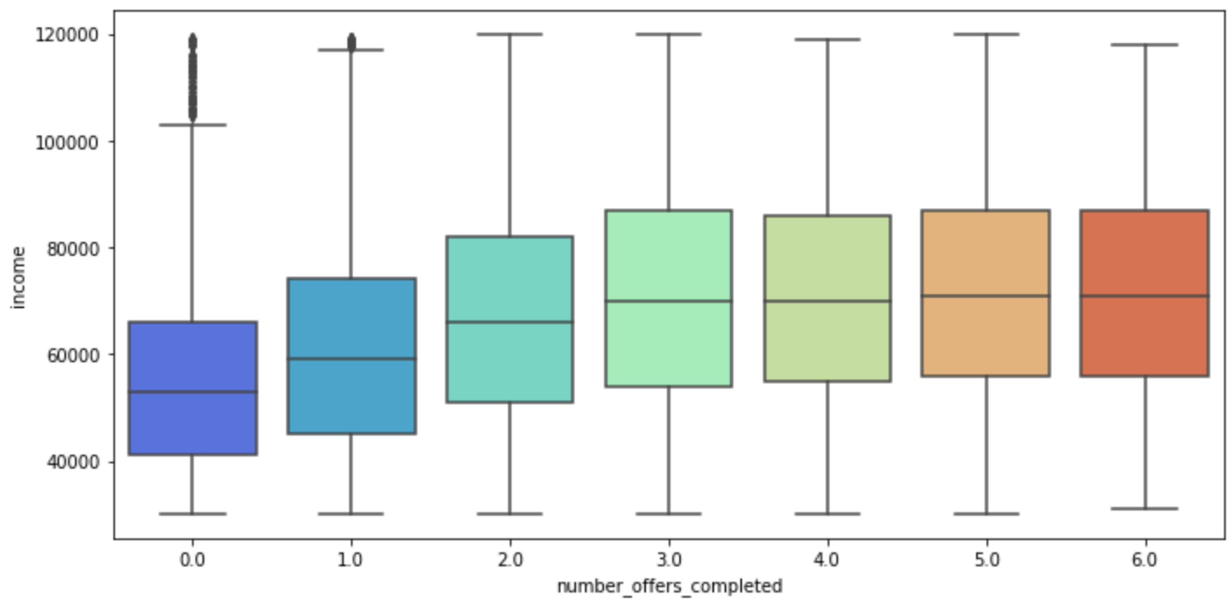
Exploratory Visualization

I explored the offer effectiveness with respect to different features of the input data such as the gender, income, offer type etc. to gain familiarity with the data which I hoped will further inform my modelling.

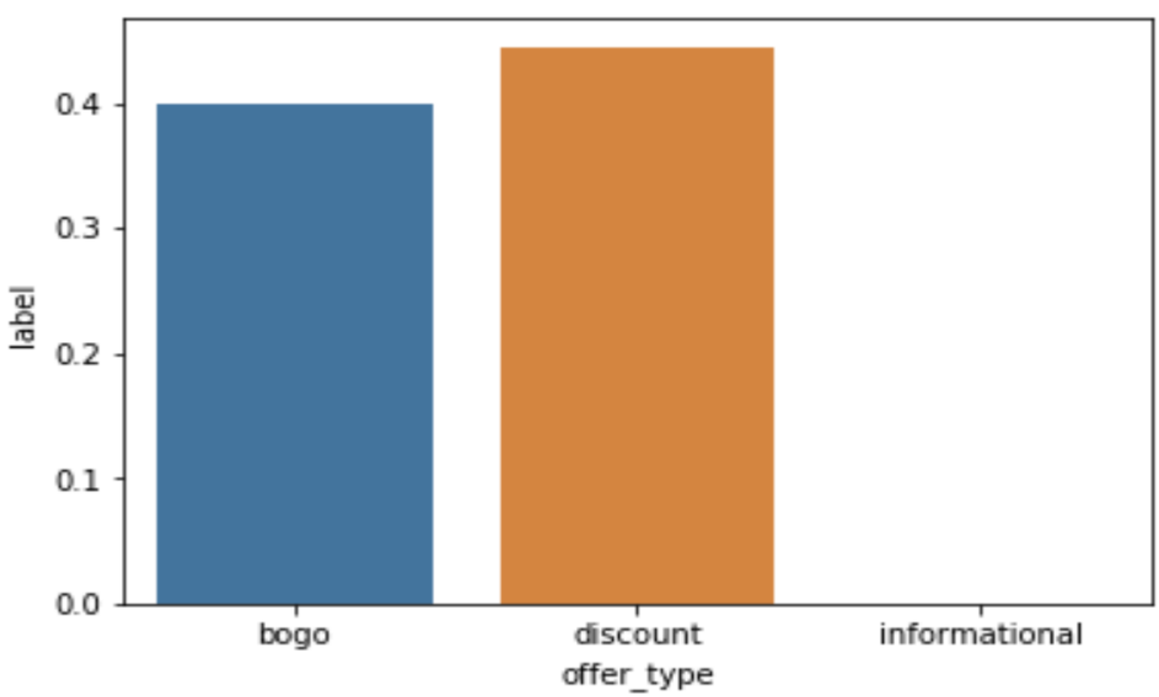




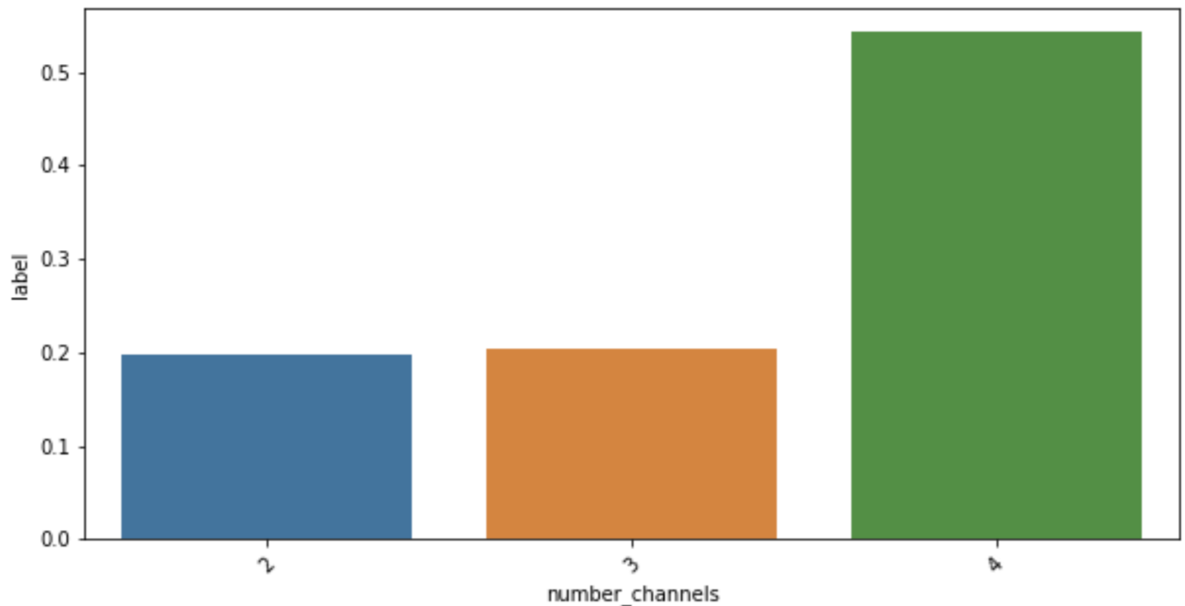
From the above two graphs we can conclude that females tend to complete a higher number of offers even though the number of offers viewed is similar to males. Hence, the offer effectiveness on females must be higher.



People who have completed a higher number of offers tend to have a higher income. This could mean that there is some scope in the offer design to make it more accesible to the lower income people.



The offer type also has an impact on the effectiveness of the offer. We can see that the discount type tends to be more effective than the bogo type.



The number of marketing channels as well as an impact on the effectiveness of the offer as seen on the above graph. Most of the marketing channels for offers include email, web and mobile. However, when social is included as well the effectiveness increases.

More such explorations are present in the exploration notebooks present on the github link of this project.

Algorithms and Techniques

The problem can be solved using a classification approach. I have used tree based algorithms like Random Forest and XGBoost to develop the classification model. I preferred tree based algorithms due to the varying scales of the input features. We have features like income and age which are on different scales. So, in order to use these features without any preprocessing I used tree based algorithms. Classification techniques such as neural networks and knn can be used as well after converting all the features to the same scale.

Benchmark

This is an unbalanced classification problem as there are more instances of an offer not being effective. I have randomly split the data into train and test; I used the latter dataset to evaluate my model and decide the best one. The benchmark model is a naive model where we just predict the majority class for all the data points in the test dataset. This results in an accuracy of 63%. My aim would be to improve upon this naive model with more advanced models and appropriate feature engineering.

Methodology

Data Preprocessing

The first step in the preparation of the required data was to create the training data of effective and non effective user offer combinations. From the transcript data all the offers that were viewed by the user and completed before the expiry were considered effective. There were around 4205 user offer combinations that haven't expired yet and not yet completed. Hence these were removed from the modelling. Also, there were cases where a user would get the same offer multiple times; Only the recent offer is taken into consideration for such cases.
The final dataset had 59083 entries and these were further divided into train and test.

The following were the preprocessing done on the Profile data

1. Converted the values of age 118 to null
2. Imputed the Null value in gender with 'O'
3. Imputed the null values of age and income with the respective median value of the O group
4. Created a membership_days column form the became_member_on column since a date column cannot be directly used in any machine learning algorithm
5. One hot encoded the gender column

The following were the preprocessing done on the portfolio data

1. One hot encoded the channels and offer type column
2. Created a number of channels feature
3. Created a feature called difficulty ratio as well which the difficulty divided by the duration

The notebooks **create_dataset** and **create_model_data** creates the final modelling data and runs the model as well.

Implementation

Once the preprocessing was done I created features related to the historical order patterns of the user hoping that these signals would further determine if an offer would be effective or not on a particular user. These were the features engineered

1. Number of orders
2. Total amount spent on the orders
3. Number of offers received, viewed, and completed
4. Number of orders during an offer period
5. Amount spent during an offer period

Then, the datasets were consolidated by joining on the appropriate keys to have the final training data. It consisted of a total of 23 features after the engineering.
After this, I trained a random forest model with the y variable as the effective or not effective label and X variables as the features created above. I implemented a grid search hyper

parameter tuning as well to improve the results. The model with hyperparameter tuning is taken from the **ml_utils.py** file where the helper functions have been created.
The challenging part about the implementation process was in the data preparation step. Since the transcript data was just a sequence of user events over time, manipulating it to the required format required a lot of thought. For example, the feature engineering of the historical order patterns had to be calculated only till the time the offer was received by the user. The reason for this is because ideally we would have to send the offer to a user who has a higher probability of viewing and completing it. The information available to us will only be till the time we plan to send the offer. Hence the features were calculated only till the time received for all entries in the dataset. I have created a function called agg_fun which does precisely this in the **create_model notebook.**

Refinement

The first model that I used for solving the problem was a xgboost as tree based algorithms require minimum preprocessing and this provided an accuracy of 68%. This accuracy is 5% more than the benchmark established by the naive method.
The steps taken to improve this to the final model accuracy of 76% are the following

1. Since the xgboost algorithm is sensitive to hyperparameters I wanted to check how a random forest model would perform. The default parameters resulted in an improvement to 72%
2. Additional features like difficulty ratio and reward ratio also helped in improving the accuracy to 75%.
3. Hyperparameter tuning was used to improve the accuracy further to 76%

Results

Model Evaluation and Validation

The classification models were evaluated on the common classification metrics such as accuracy, precision and recall on the test dataset. Further the auc score of the model was also calculated.
The model’s accuracy on the training dataset and testing dataset are both 76%. This implies that there is no overfitting and there is scope for improvement by using a more complex model.

The following is the classification report of the model

```
              precision    recall  f1-score   support

           0       0.91      0.70      0.79     12445
           1       0.62      0.87      0.73      7053

    accuracy                           0.76     19498
   macro avg       0.76      0.79      0.76     19498
weighted avg       0.80      0.76      0.77     19498
```

From the classification report we can see that the recall for the positive class is 87%. Hence we are able to capture most of the instances where the offer on a user is effective. However the model's precision is low. This is by design as from a business point of view I believe the error for a false positive(predicting that offer is effective when it is not) is lower than that of a false negative (predicting that offer is not effective when in fact it is).

The following are the final parameters of the Random Forest model used

```
RandomForestClassifier(class_weight={0: 1, 1: 2}, max_depth=8, max_features=0.5,
                       min_samples_leaf=10, n_estimators=400, random_state=42)
```

Given how similar the training and testing scores are, and also the working of the random forest algorithm I am confident that it will be robust to the changes in the input data and continue to provide reliable predictions.

Justification

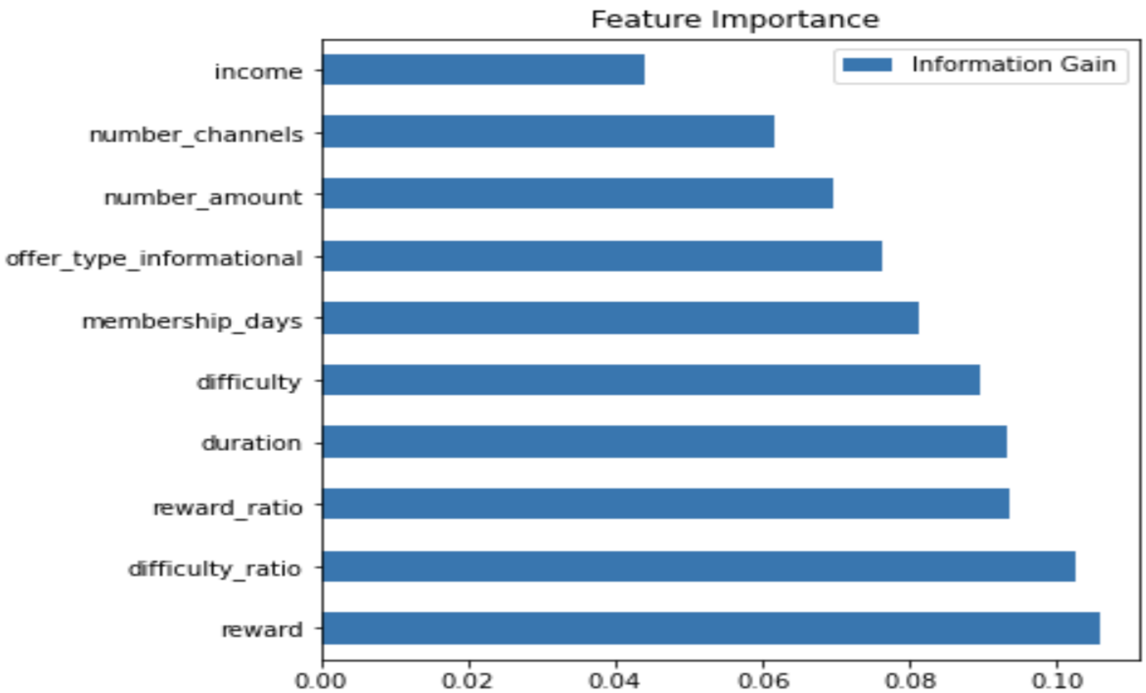
The machine learning solution proposed above gives an accuracy of 76% which is 13% percentage points more than the naive prediction. Since the model is providing similar results on unseen data as well we know that this solution is robust and will provide similar results for other unseen data. The naive model just predicts the majority class (0) for all the entries in the test data, whereas the machine learning approach learns from the historical data to predict whether an offer would be effective on a user in a more intelligent way.

Although the machine learning model solution is an improvement over the naive solution, it is not perfect as its accuracy is not 100%. We can try to further improve this model and I will propose a few ways in the improvement section.

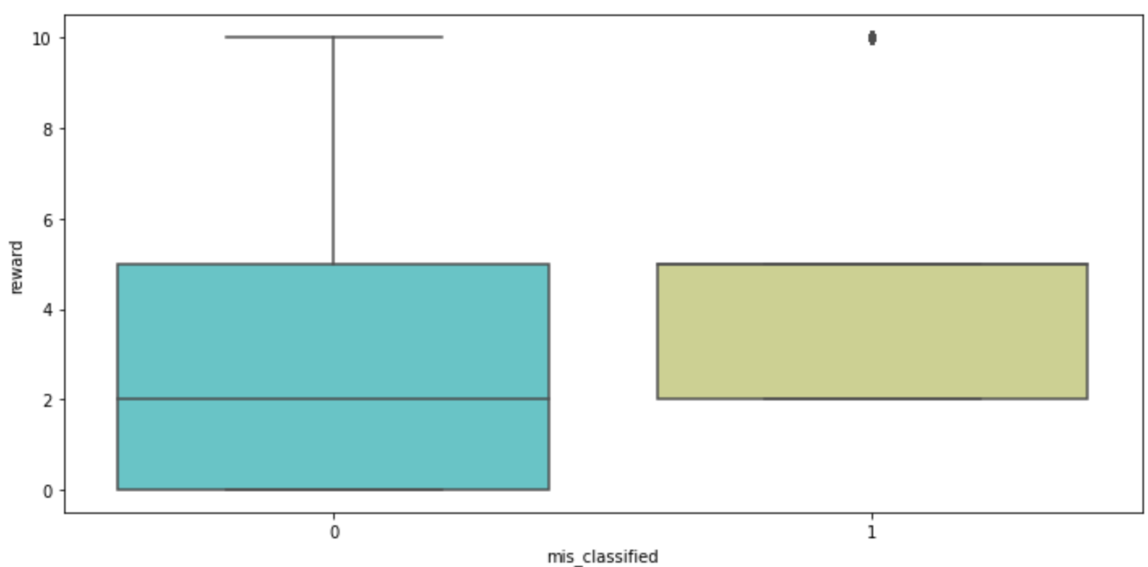
Conclusion

Free form Visualization

To conclude, the final model learns from the patterns of the data and does a better job in predicting whether an offer would be effective or not on a user. The model also provides the feature importance metrics which can help us understand which features had a higher impact on the model's prediction.



From the above graph we can see that the offer features play a very important role in predicting whether an offer is effective or not and the user related features are of lesser importance. This could also result in explaining the why the precision of our predictions are low. Digging deeper into the false positives predicted by the model we see the following



The false positive predictions have a higher reward and the model is mainly driven by that to make the positive prediction as reward is the most important feature shown in the feature importance. I believe stronger user related signals can help alleviate this issue with the model.

Reflection

The following steps were followed to solve the offer optimization problem in Starbucks

1. I understood the business problem thoroughly and the characteristics of the data that was given to solve the problem.
2. I performed exploration on the data to get comfortable with the data and understand its salient features.
3. After this, the training dataset was created with the following columns - person, offer_id,time and label. Label is the flag which tells us whether the offer on a user was effective or not. An offer is effective when the person viewed and completed the offer before its expiry.
4. Once this was done, I preprocessed and feature engineered the variables that I would use for my modelling from all the datasets involved. I then consolidated this data to get the final modelling data.
5. The modelling data was divided into train and test. A random forest model was used to predict whether an offer is effective or not.
6. The model was then evaluated on various metrics like accuracy, precision, recall, and auc score.

The interesting and simultaneously the most interesting aspects of the project was the preprocessing and feature engineering step. It required me to understand the input datasets in depth to transform the data into a form that can be consumed by a machine learning model. Once the logic was decided, coding it on python was also an interesting task and I believe I have learnt a lot from this.

I believe that the process followed in my solution is generic enough to be used in any offer optimization problem given the input datasets are in a similar form.

Improvement

I propose the following ways to improve the above provided solution -

1. More data - The feature importance shows that offer related features play a more important role than the user related features like the historical order patterns. I believe this is because the order patterns are all calculated on the promotion period of the 30 days given in the dataset. I believe more historical data about the order patterns of the user will play a significant role in determining whether an offer would be effective on a particular user. More demographic data like the location and the marital status of the individual can be useful signals.
2. Segmentation modelling - It could be the case that users in our dataset are significantly different from each other in terms of their ordering patterns and it might be useful to put them into separate groups. Once this clustering activity is done we can then run a model for each group. This helps in reducing the noise created by one group on another and can improve the results.
3. More advanced models - Models like neural networks or ensembling two different models could also be used to improve the results.

References

1. https://towardsdatascience.com/starbucks-offer-optimisation-cdf9bcedd48a
2. https://medium.datadriveninvestor.com/offer-optimization-using-machine-learning-46a1f5d1b59b