Starbucks Offer Analysis Capstone
by Quintin Sheridan

Project Motivation

In this project, I analyze a set of Starbucks promotional offers and how different users respond to them. The goal of the project is to try and find trends in how users respond to different offers and the communication channels used to notify users of the offers. When a company sends promotional offers to users, it is important for them to know that the offers they are sending are having a positive effect on customer purchases and company profits. Marketing campaigns are quite costly, and therefore If a company is going to send an offer, they will want to know it will likely increase the users tendency to make future purchases with the company.

In the project, there are three datasets:

portfolio

- **id (string) - offer id**
- **offer_type (string) - type of offer ie BOGO, discount, informational**
- **difficulty (int) - minimum required spend to complete an offer**
- **reward (int) - reward given for completing an offer**
- **duration (int) - time for offer to be open, in days**
- **channels (list of strings)**

profile

- **age (int) - age of the customer**
- **became_member_on (int) - date when customer created an app account**
- **gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)**
- **id (str) - customer id**
- **income (float) - customer's income**

transcript

- **event (str) - record description (ie transaction, offer received, offer viewed, etc.)**
- **person (str) - customer id**
- **time (int) - time in hours since start of test. The data begins at time t=0**
- **value - (dict of strings) - either an offer id or transaction amount depending on the record**

Portfolio contains all the information about the different offers being sent. In this example project, there are three different types of promotional offers: BOGO, discount, and informational offers. Informational offers do not provide the user with discount opportunities, but they are intended to stimulate user interest and purchasing. Discount and BOGO offers have a duration period during which the users can successfully complete the offer and receive a reward, either a free item or a discount for spending in excess of a required amount within the offer duration.

Profile contains all of the information that users self disclose about themselves. The marketing campaign will want to customize which offers are sent to which users to increase their spending habits and thus the company profit. When a user makes an account, they are not required to disclose any information and a significant portion of the user base does not provide any of their information.

Transcript contains a record of all user events. Users can receive, view, and complete offers in addition to making purchases. Each time an event occurs, the time of the event is recorded. It is possible that the same user can receive the same offer more than once. It is also possible for a user to complete an offer more than once. Therefore, a large difficulty in analyzing the dataset is to figure out which ‘offer viewed’ and ‘offer completed’ events correspond to which ‘offer received’ events. More on this will be discussed later in the analysis portions of this project.

Project Plan

I have chosen to evaluate the dataset by performing two different sets of analysis. In the first portion of analysis, I wanted to evaluate the aggregate behavior of users with regards to the different offers. I wanted to see which offers were viewed more than average, which offers were completed more than average, which offers were both viewed and completed more than average, which offers were completed without being viewed more than average, and which offers were ignored more than average.

If a marketing team is trying to increase brand awareness, they will want to send users the offers that they are most likely to see. Different offers are sent by different channels: social, mobile, web, and email. Each offer is sent out via multiple communication channels. Not all users receive the same offers. It is therefore challenging to determine which offer types and which channels are most likely to end up being viewed by a user.

A business succeeds when it is able to use its marketing strategies to increase customer purchasing. However, offers can be completed without ever being viewed by a user, and therefore we can not directly determine how effective different offers are by simply examining which users receive which offers and how they make purchases in turn. A successful marketing campaign will aim to maximize offer view rates and the completion of viewed offers while minimizing offer completions where the offers are never viewed by the user.

In the second portion of the project, I create a machine learning model to predict if a certain user receives a completable offer, how will they respond to that offer. There are four different outcomes that can occur when an a user receives and offer:

1. The user can ignore the offer
2. The user can view the offer, but fail to make purchases to complete the offer
3. The user can view an offer and then make purchases to complete the offer
4. The user can make purchases and complete an offer without ever viewing the offer

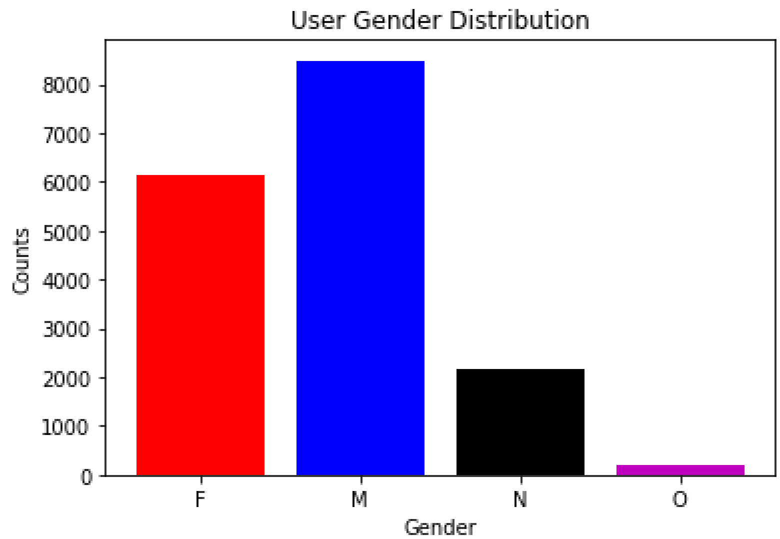
This is a classic example of a multiclass classification problem. By having a model to predict user behavior when receiving an offer, we can assess each offer for each user and determine which users should be sent which offers or which users should not receive an offer due to the fact that they may complete it without ever viewing it.

Exploratory Data Analysis

profile

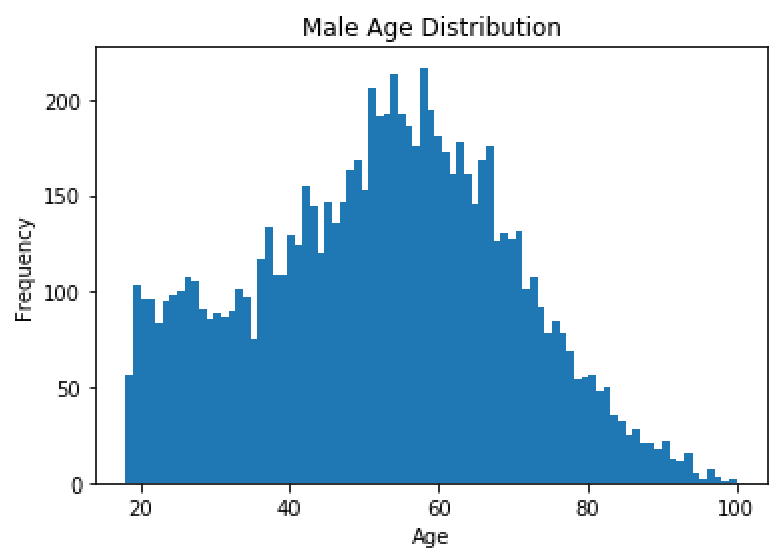
The first thing that I wanted to examine was the characteristics of the user base. When I examined user ages, I found that a large portion of the users had ages of 118, which does not seem physically possible. Upon examining the dataset, I found that all users of age 118 had not entered any information other than their ages. Therefore, I concluded that a user must enter an age over age 18 to be eligible for membership, but the users were not required to enter any other information. 2175 of the 17000 total users had not entered any of their profile information. This is a significant portion of the user base, so I did not want to simply ignore these users when analyzing the transcript data.

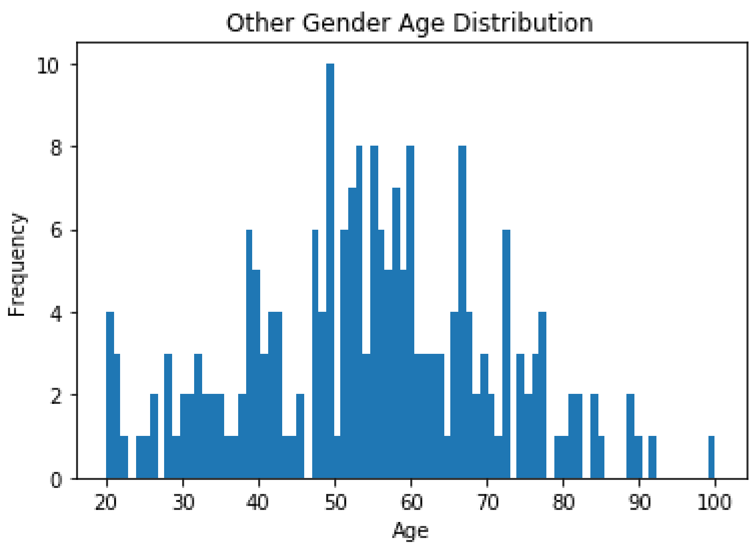
The following plot shows a distribution of users by gender.



We can see that the majority of the users are male, and that the portion of other gender users is small compared to the other categories.

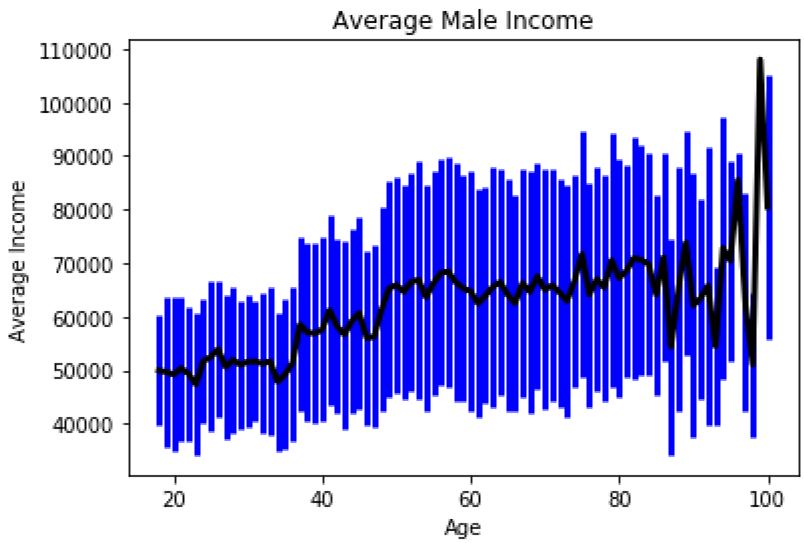
The next thing I wanted to examine was how the age and income distributions varied for the different genders in order to decide on the best approach for feature engineering for the multiclass classifier. The following plots show the age distributions of the users by gender:

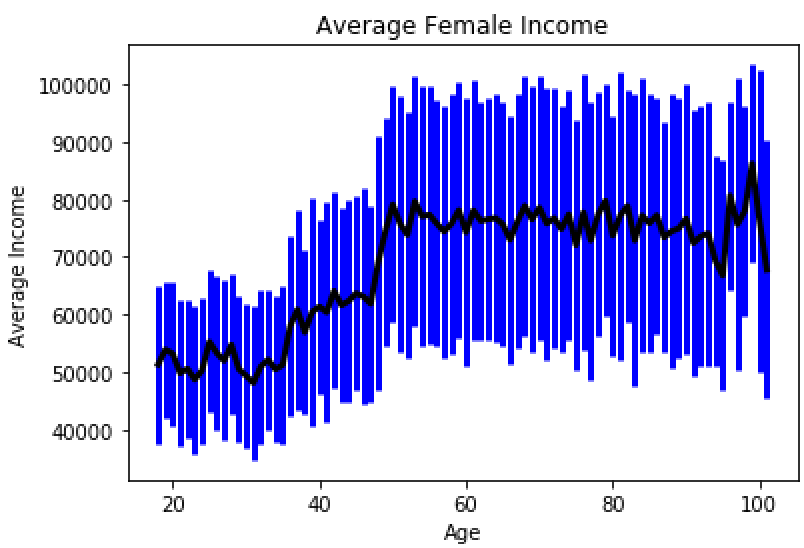


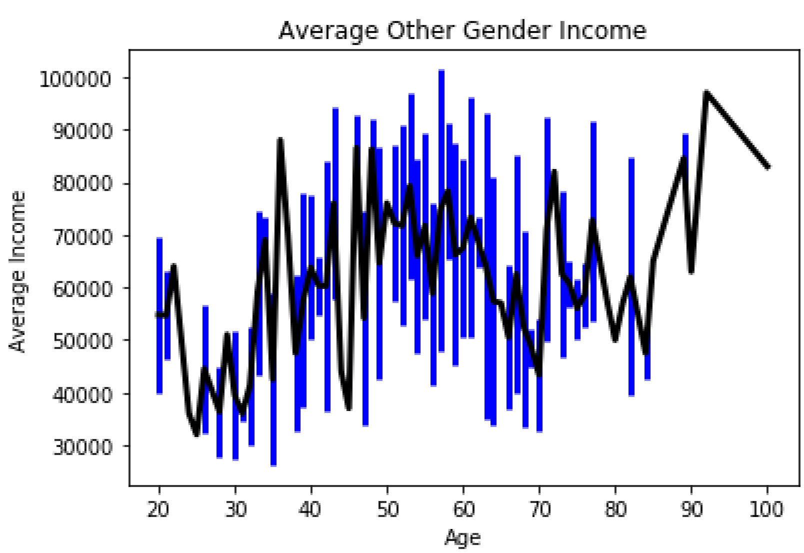


We can see that for all genders, the user age is approximately normally distributed around an average age of 60.

The following plots show income Vs. age for the different users with vertical lines that represent the standard deviation of income at a given age.

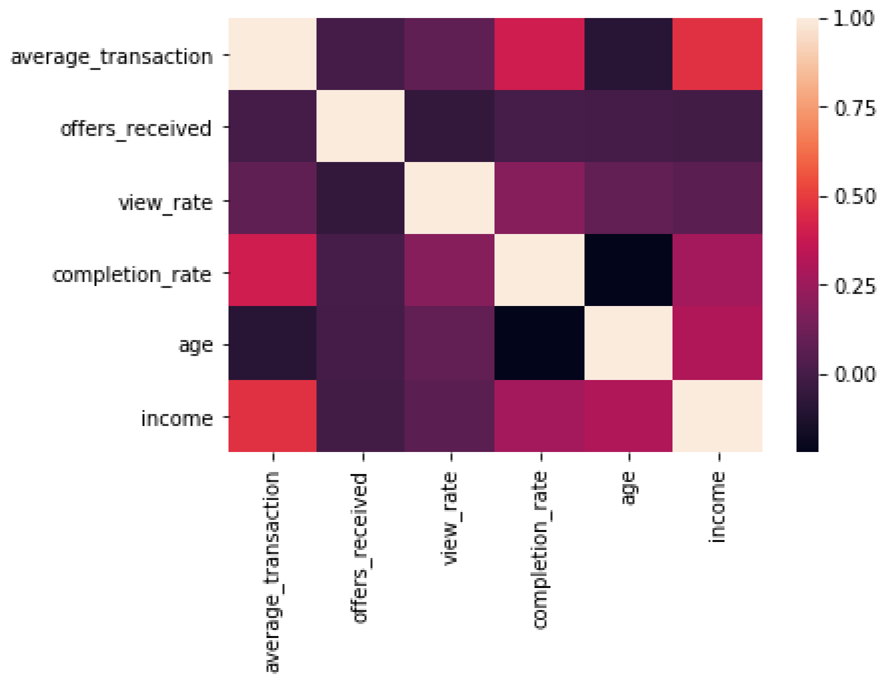

Average Male Income
Average Income
110000
100000
90000
80000
70000
60000
50000
40000
20
40
60
80
100
Age


Average Female Income
Average Income
100000
90000
80000
70000
60000
50000
40000
20
40
60
80
100
Age


Average Other Gender Income
Average Income
100000
90000
80000
70000
60000
50000
40000
30000
20
30
40
50
60
70
80
90
100
Age

We can see that for all users, income increases from about 18-50 years old where it plateaus. We can also see that there is a significant variance of income within any particular demographic. From this exploration, I decided that it would be best to represent both income and age as categorical variables. This allows for the creation of categories for users that did not input any user information so that their transcript events can be included in the machine learning model. If a large portion of the user base does not input any information, it would be good to still be able to serve customized offers for these users.

The next thing that I wanted to examine was the user spending habits. Below is a heat map showing the correlation between user attributes, their spending habbits, and how they view and complete offers.



We can see that users who make more money are likely to spend more money. We can also see that offers are more likely to be completed if they are viewed. Unsurprisingly, users that spend more in a single transaction are more likely to complete an offer.

Project Difficulties and Strategies

The most difficult part of this project was to determine which events, ‘offer received’, ‘offer viewed’, and ‘offer received corresponded to each other. The difficulty is that we do not a have a primary key on which to join ie. we can not simply perform a join of datasets as **offers_received_df.merge(offers_completed, how=’left’, on=’key’)**. When I first tried to join on [user_id, offer_id] I found that it did not work because some users will receive the same offer more than once. For example, if a user received the same offer twice and completed it twice, a merge on **[‘user_id’, ‘offer_id’]** would result in four rows for this combination of **[‘user_id’, ‘offer_id’]** . Therefore, I thought I could just merge and remove offers through selective filtering by making sure that the offer was not completed before it was received and that it was completed within the offer duration. However, this strategy still resulted in data mismatches with repeats of either offer completions or offers received.

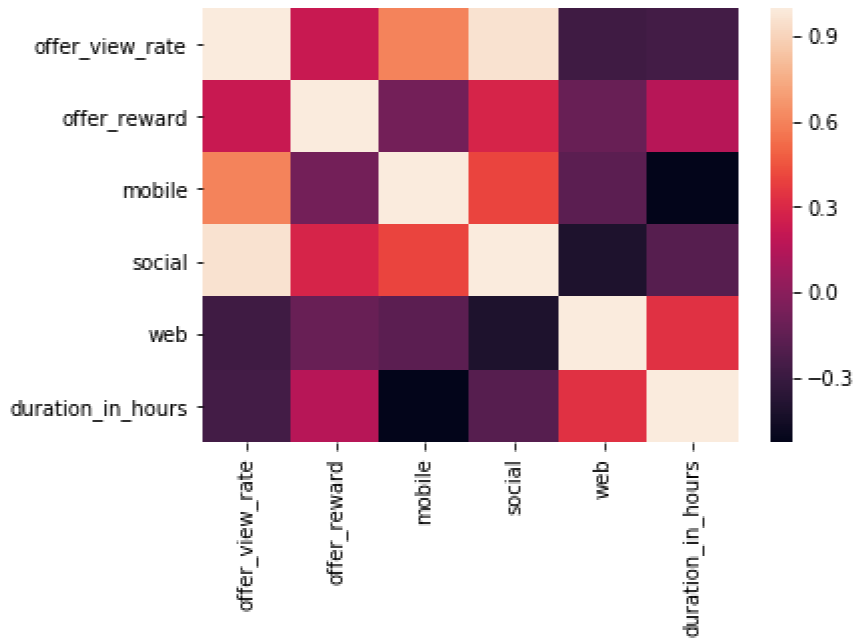
Another reason for the data mismatch is that there are duplicate rows in offers completed, where an event will have the same values for **[‘user_id, ‘offer_id’, ‘completed_time’]**. I was not sure if this is an error in the data or if it meant someone was able to complete a BOGO or discount offer more than once in the allotted offer duration. In analyzing completion rates and preparing data for the multiclass classifier, I decided to remove offer completion duplicates, which amounted to 397 of 33579 completions being reviewed. This is about 1% of the data so it should not have profound effects on data analysis and the machine learning model.

After doing some research, I thought I could use **pd.merge_asof()** which aims to perform a 1-1 left join where an exact match is performed on keys along with a closest match on another column. This could work depending o. the data. I was in fact able to use **pd.merge_asof()** to merge offers viewed with offers received with no issues. However, it is possible that the user will complete the same offer twice within the offer duration of a single offer. Both of these completions would then try to match to a single received offer causing a data mismatch and duplicated data.

Finally I came up with a strategy to resolve this issue. I assigned an event id, **received_id, viewed_id, or comleted_id** to each none-transactional event in the transcript. This way, I could check to make sure when merging data frames that I had no repeated transaction. I came up with a slower data merging operation to merge offers completed with offers received which required me to iterate through each completed offer, find its potential match with the largest gap in **‘completed_time’** and **‘received_time’** and to assign them to each other and remove that offer received event from future matches. This method prevents duplicate assignment of offers received to offers completed and vice versa, but it is not a vectorized implementation and would therefore suffer from scalability issues if deployed to a real world machine learning application. This project would be more suitable for real world application if I was able to make a 1-1 matching algorithm that solved time constraints, but I am not sure if this is even possible.

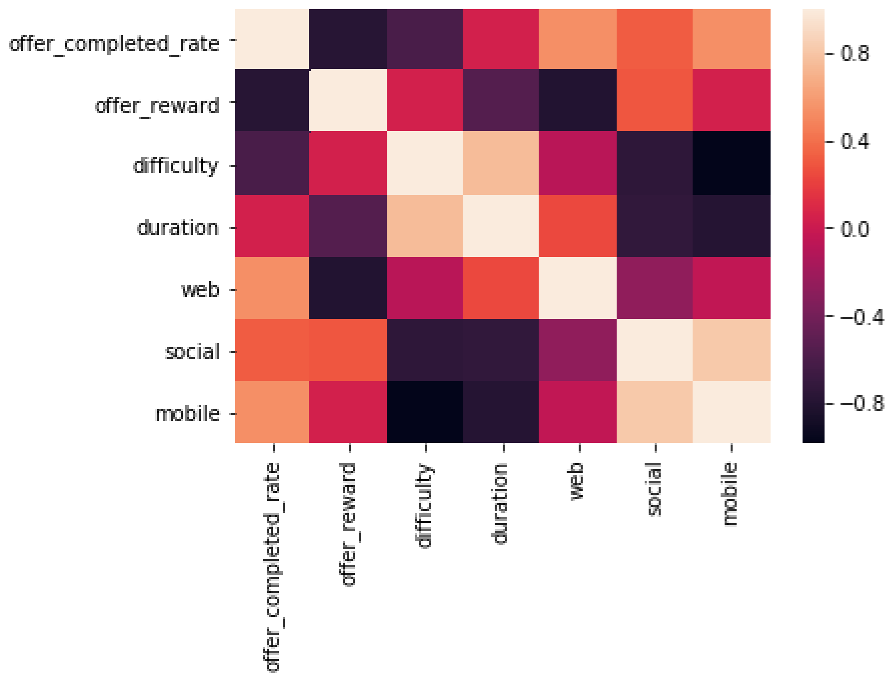
Offer Rates

After successfully merging data sets, I was able to examine user-offer interaction rates to see which offers were under/over performing the average offer performance. First, let's examine offer view rates.

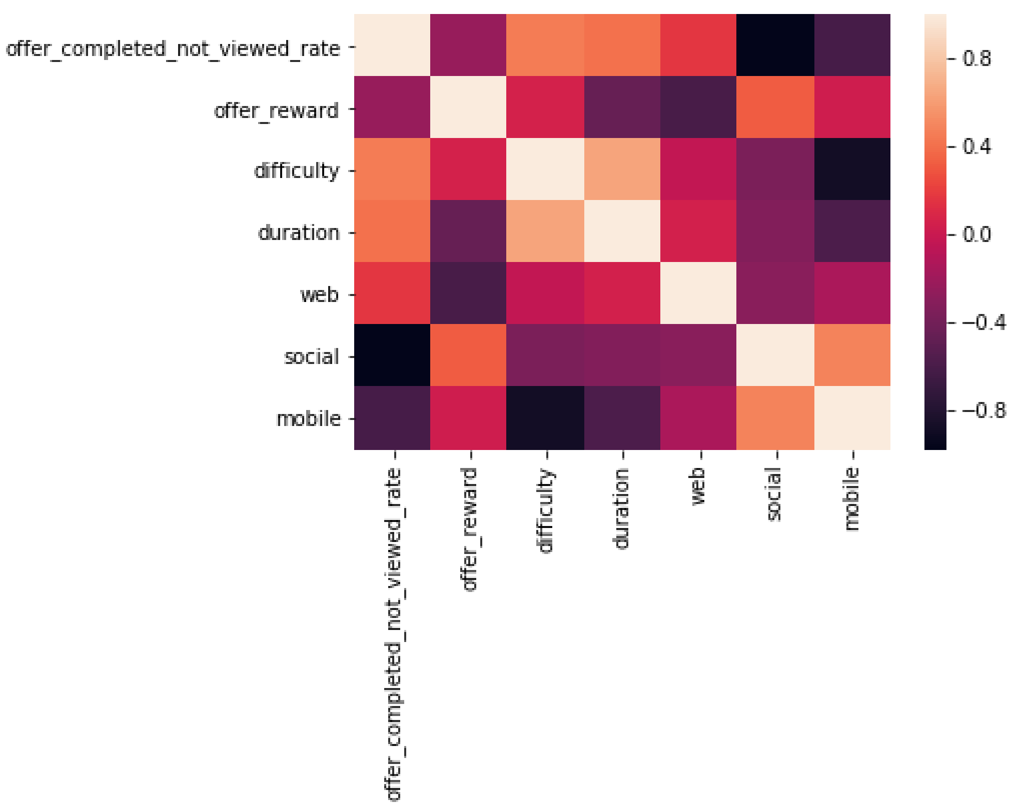


Offers are more likely to be viewed if they use the ‘social’ and ‘mobile’ channel. Interestingly, offers are not more likely to be viewed if they have a longer duration.

Next we can look at the offer completion rates.



We can see that offers are more likely to be completed if they have a longer duration. Similar to offer viewing rates, offer completion rates are higher for social and mobile offers. The harder and offer is and the higher its reward, the less likely a user is to complete it.



The longer an offer lasts, the more likely it is that the offer will be completed without ever being viewed. We can see that web offers are more likely to be completed without being viewed. Therefore, this is not an effective marketing channel.

Offer Response Prediction

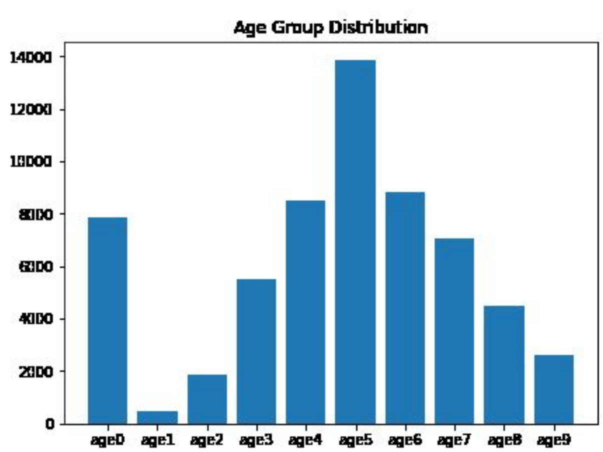
After assigning each received offer to an end state, viewed but not completed, viewed and completed, completed but not viewed, or ignored I was able to set up a multiclass classifier. I chose to assess how the decision tree would perform in the evaluation. The output labels for the classification were viewed but not completed, viewed and completed, completed but not viewed, and ignored. Only one label was assigned to each row in the dataset. The features for the dataset were a mix of transcript details as well as features for the users. The transcript feature came from offer ids. The offer feature selected were:

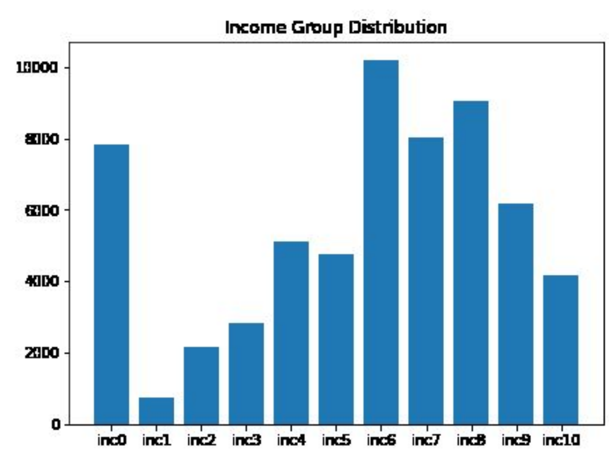
[**'difficulty', 'duration', 'bogo', 'discount', 'reward', 'web', 'mobile', 'social', 'duration_in_hours'**]

The email channel was removed as a feature because all of the offers were sent via the email channel so it does not help to discern the different offers. The time that an offer was received was not included because I wanted to find a model that would easily generalize to a new user.

As mentioned earlier, user age, gender, and salary were transformed to categorical variables in order to include users that had not entered any profile information. The date that users became a member was removed as a feature because all of the users joined on the same date so it does not help to discern the different event outcomes.

Let’s examine the distribution of received offers amongst the user age and income groups





We can see that not all of the users receive the same amount of offers. We can see that offer receival rates are not the same for all users. This is likely due to marketing strategies within the company. Users that would typically be working class (ages 25-35, incomes $40-50K) receive significantly less offers than other user groups. This is probably due to the fact that these are regular users and therefore, one would not expect them to visit stores more frequently and make more purchases if given offers.

I have chosen to use three different classifiers that work well with categorical features: Random Forest Classifier, Logistic Regression, and Categorical Naive Bayes. First, I wanted to test and see how these three models performed using the default hyperparameter arguments. I used the f1 score, the confusion matrix, and the overall model accuracy in order to evaluate model performance. The results are summarized below:

RandomForestClassifier:

{0: 'viewed and completed', 1: 'completed but not viewed', 2: 'viewed but not completed', 3: 'ignored'}

	precision	**recall**	**f1-score**	**support**
0	0.53	0.64	0.58	4486
1	0.31	0.20	0.24	1233
2	0.60	0.51	0.55	4554
3	0.48	0.52	0.50	1936
accuracy			0.53	12209
macro avg	0.48	0.47	0.47	12209
weighted avg	0.52	0.53	0.52	12209

Confusion Matrix:

	0	1	2	3
0	[2878	221	1041	346]
1	[466	242	144	381]
2	[1736	135	2328	355]
3	[372	176	382	1006]

RandomForestClassifier Accuracy: 0.529

LogisticRegression:

{0: 'viewed and completed', 1: 'completed but not viewed', 2: 'viewed but not completed', 3: 'ignored'}

	precision	**recall**	**f1-score**	**support**
0	0.53	0.63	0.58	4486
1	0.34	0.15	0.21	1233

	precision	recall	f1-score	support
2	0.60	0.53	0.56	4554
3	0.47	0.54	0.50	1936
accuracy			0.53	12209
macro avg	0.48	0.47	0.46	12209
weighted avg	0.53	0.53	0.52	12209

Confusion Matrix:

	0	1	2	3
0	[2822	143	1124	397]
1	[473	190	146	424]
2	[1642	94	2428	390]
3	[383	128	372	1053]

LogisticRegression Accuracy: 0.532

Evaluating default performance of **CategoricalNB**:
{0: 'viewed and completed', 1: 'completed but not viewed', 2: 'viewed but not completed', 3: 'ignored'}

	precision	recall	f1-score	support
0	0.56	0.48	0.52	4486
1	0.25	0.33	0.29	1233
2	0.57	0.55	0.56	4554
3	0.45	0.56	0.50	1936
accuracy			0.50	12209
macro avg	0.46	0.48	0.47	12209
weighted avg	0.52	0.50	0.51	12209

Confusion Matrix:

	0	1	2	3
0	[2162	548	1366	410]
1	[252	409	164	408]
2	[1257	325	2486	486]
3	[196	335	324	1081]

CategoricalNB Accuracy: 0.503

We can see that all of the models are performing at about a 50% accuracy for the classification task. While this is not great, it is better than the random case of ~25%. In choosing the best model for this task, we need to consider the different classes. It is not imperative that we accurately classify ignored offers as sending an offer that ends up being ignored will not result in lost revenue for the company. We can see that Categorical Naive Bayes (CNB) does the best job of classifying offers that are completed but not viewed. It is able to successfully classify the most of these and when it errors, it tends to classify the offer as an ignored offer. This is a good result because in either case, you probably would not end up sending the offer to the customer. However, CNB also does the worst job of correctly classifying offers that are viewed and then completed, which could end up resulting in lost revenues if used to recommend offers. Logistic regression and RandomForest have nearly identical performance with RandomForest performing slightly better at classifying offers that are completed without ever being viewed.

I decided to try and improve the RandomForest model by using RandomizedGridSearchCV. However, an extensive search of different hyperparameters was only able to improve the model performance by a negligible amount, 52.9% accuracy to 53.3% accuracy. This suggests limited correlation between the model features and labels. However, it would likely be difficult for a marketing expert to perform better at assessing how a user will respond to an offer.

Conclusions

In this project, I examined how users respond to different Starbucks offers. I analyzed which offers were under/over the average in terms of the resulting user interaction. I found that social and mobile offers are the most likely to be viewed, and also to be viewed and then completed. I found that web offers tend to be completed without ever being viewed, so the web is not an effective marketing channel. Offers that have a high reward and a long duration are not more likely to be completed after being viewed, but they are more likely to be completed without being viewed.

I found that there is not a lot of correlation between user profile information, offer attributes, and the final outcome of a user receiving an offer. My classification model was only able to achieve about 50% accuracy. While this is significantly better than random, it is not highly effective and there are many improvements that could be made to the model. It is beneficial, however, to be able to predict how a user will respond to a given offer before you send the user an offer and the classification model created here could be used in part of a recommender system.

Future Improvements

This project only touched on a handful of potential ways to analyze the data. The project could be improved in several ways. One way to improve the project, would be to examine which users have similar habits ie. to perform a clustering analysis to see how many different customer segments exist. This would aid in any targeted marketing campaign because you could evaluate new offers experimentally by sending different offers to subgroups of each market segment and then evaluating how the segments respond to the offers.

The ultimate goal of a business is to maximize profits. In this project, I simply examined aggregate behavior of users to the different offers and then predicted how a given user will likely respond to a given offer. However, to maximize company profits it is necessary to evaluate user spending habits when no offers are sent to them. If sending users offers does not in fact increase their spending then it decreases profit margins for the company. By chaining multiple machine learning algorithms together, it would be possible to create an offer recommendation engine that aims to maximize profits. When we are analyzing proffits, we also need to consider how much money the company gains/looses for different offers. For example, if a user is completing BOGO offers, how likely would they have been to make the same purchase of two or more items if they never received a BOGO offer. If the user likely would have made multiple purchases regardless of the offer, sending them a BOGO offer would reduce company profit margins. However, if they would not likely make multiple purchases, the effects of BOGO offers on profit margins would be negligible due to the low cost of making a beverage. When the company issues discount offers, they need to be careful that it is actually changing customer spending habits. If a customer would likely spend the same amount with or without a discount offer, then it does not make sense to send them an offer. An important part of analyzing who should receive which offer would be to try and identify which users are regulars (people who go to starbucks on a daily basis) and which users visit stores every now and then. For regular users, you would want to give them free items with purchase rather than discounts upon meeting a threshold, because you will not likely increase the rate at which they make purchases. On the other hand, you might like to give infrequent customers discounts for spending a certain amount to try and make them visit stores more frequently and become regular users.

There are several ways in which sales data such as this can be analyzed and a complete analysis was beyond the scope of this project. However, several of the data preparation and analysis techniques used here are easily extensible to other business and different marketing campaigns. It is important when analyzing data such as this to first explore the data and then to assess which valuable questions about the data you think you can answer and then to focus on obtaining those results. The further you analyze the data, the more ideas you will have for future analysis and model improvements.