Udacity Machine Learning Nanodegree
Capstone Project Proposal

Mohamed Hesham Ibrahim Abdalla

November 3, 2020

1 Project Overview

This is my submission of the capstone project of Udacity Machine Learning Nanodegree. In this project, a simulated dataset of Starbucks reward app is used to visually analyze and predict the type of customers that are willing to complete the advertised offers in the Starbucks app. This analysis is an important step to avoid showing offers to the wrong customer groups who will not complete the offer or even get annoyed from the app completely. Also, saving resources and time by matching offers with the customers who are more willing to complete them.

2 Problem Statement

This project tackles the issue of identifying the customer type that could be interested in a certain offer. This is done by answering one important question: if a customer has certain properties, would he complete a certain offer?

I will use data visualization techniques to analyze the data and hence know what group of customers could pay attention or interest to a certain offer. In addition, I will develop a classifier to detect whether or not a certain customer would be interested in a certain offer.

3 Metrics

In this project, F1-score is used to quantify how well a model is. F1-score gives equal weights to both the positive and negative class (this customer will complete the offer (1), this customer will not complete the offer (0)) which is needed since the dataset has almost balanced class ratio.

4 Datasets and Inputs

This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. The data will be used in the context of Udacity Machine Learning Nanodegree capstone project.

datsets to be used in this project:
portfolio.json: containing offer ids and meta data about each offer (10 rows x 6 columns)

- 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.json: demographic data for each customer (17000 rows x 5 columns)

- 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.json: records for transactions, offers received, offers viewed, and offers completed (306534 rows $\times$ 4 columns)

- 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

The mentioned datsets could be merged and combined to statically analyze the demographic groups that could be interested in a certain offer.

5 Data Exploration

This section statistically analyze the datsets and its features.

portfolio.json:

- This datset does not have any null values nor duplicate values.
- 'offer_type' feature has three offer types BOGO, discount and informational.
- 'difficulty', 'reward' and 'duration' features are more of a category feature rather than a continuous value.
- 'channels' feature is a list of a combination of the strings ["email", "social", "mobile", "web"]. This feature must be transformed to one-hot encoded features to be able to use it for classification.
- After merging the transcript dataset with the portfolio dataset and grouping the data by offer id, it was found that most viewed offer is the offer with the id customers receive the offer, view it and complete it. On the other hand, 6.37% of the customers receive the offer but do not view it and still complete it. Furthermore, 27.25% of the customers receive the offer, view it but do not complete it.

Fig. 1: A sample from the dataset 'portfolio.json'.

```
[ ] portfolio.sample(5)
```

	reward	channels	difficulty	duration	offer_type	id
3	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9
1	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0
5	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2
0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd
6	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4

transcript.json:

- This datset does not have any null values nor duplicate values.
- 'value' feature includes a dict object which is in the form {'amount':int}, {'offer_id':'id'} or {'offer id':'id'}. 'offer id' or 'offer_id' only show up if the event feature of an instance has the word 'offer'. Otherwise, 'amount' key is shown in the value dict object. This feature must be transformed to include the amount and offer id values as separate columns.
- 'event' feature is a categorical feature with values 'transaction ', 'offer completed', 'offer viewed' and 'offer received'. This feature must be one-hot encoded for classification.
- After merging the transcript dataset with the profile dataset and grouping the data by offer id and person id, it was found that around 33.71% of the customers receive the offer, view it and complete it. On the other hand, 6.37% of the customers receive the offer but do not view it and still complete it. Furthermore, 27.25% of the customers receive the offer, view it but do not complete it.
- 'time' feature is an integer value of 336.3 mean and 200.3 standard deviation. This feature will be removed when the data is merged since it is problematic to include it when all the events are listed in one instance.

Fig. 2: A sample from the dataset 'transcript.json'.

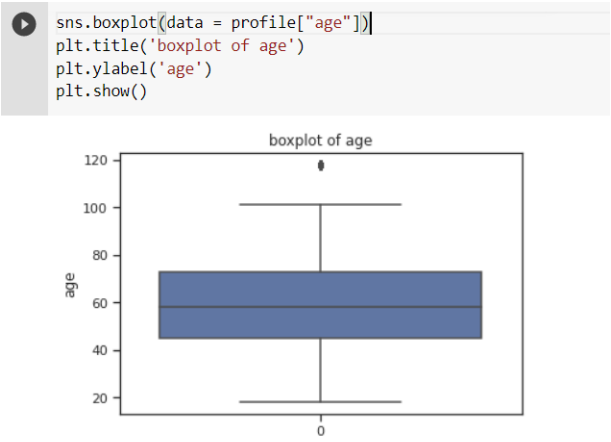
```
[ ] transcript.sample(5)
```

	person	event	value	time
51968	3124144d8d414799aa9a384fb62be3f3	transaction	{'amount': 8.27}	156
292612	7d4b80a7ba3541f49fbccbdade389014	transaction	{'amount': 12.25}	654
290985	7b58f6fe0a654970a4299a84fc36d8be	transaction	{'amount': 1.67}	648
102519	7322a6c01c874585b342ee85d245cdfe	transaction	{'amount': 15.48}	282
41582	6dba14f698ae4030ab7354cd5cfe7119	offer completed	{'offer_id': 'fafdcd668e3743c1bb461111dcafc2a4...	96

profile.json:

- This datset has any null values in both the gender and income features.
- 'became_member_on' is a date object in an integer form. This feature must be transformed to a number of days since registration of the customer in order to make the classification task easier.
- 'age' feature has ages between 18 and 118 with a mean of 62.5 and standard deviation of 26.7. After investigation, it was found that the value 118 is an outlier which is found in 12.79% of the dataset. Also, the instances that have age value of 118 have a null or NaN value in the 'gender' and 'income' features. The instances of with age 118 will be removed from the dataset completely since they also show misleading values in both 'gender' and 'income' features.

Fig. 3: Boxplot of the 'age' feature.



- 'gender' feature is categorical feature with the values 'O', 'F' and 'M'. This feature must be transformed to one-hot encoded features for the classification task.
- 'income' feature is continuous value with a range between 30k and 120k with a tick of 1k and a mean of 65k and standard deviation of 21k. This feature must be scaled for the classification task.
- as stated before, both 'income' and 'gender' features have null or NaN values when the 'age' feature of the instance is 118. The instances with the missing values will be removed.
- When merging 'transcript.json' with 'profile.json' datasets, we found that there are around 33,772 more instances in 'transcript.json' compared to 'profile.json'. Therefore, These instances were removed since they don't map to any values in 'profile.json' dataset.

Fig. 4: A sample from the dataset 'profile.json'.

```
[298] profile.sample(5)
```

	gender	age	id	became_member_on	income
14685	None	118	e8d790013bbe488ab860acd314b44f7f	20160322	NaN
4040	None	118	9bbca859a7ff43db981b10737391e550	20171104	NaN
2625	M	62	151b12c2cfa5443f9c92a53ba9ada28b	20170329	33000.0
4133	None	118	b22d1083630c43288cf7d032eaeda6e0	20150803	NaN
14850	M	66	b5a19a184eb04c08a7c92804a88a4cf6	20160426	71000.0

Fig. 5: Statistics of 'profile.json' dataset before reduction and removal of nulls.

```
[396] profile_reduced.describe()
```

	age	became_member_on	income
count	14825.000000	1.482500e+04	14825.000000
mean	54.393524	2.016689e+07	65404.991568
std	17.383705	1.188565e+04	21598.299410
min	18.000000	2.013073e+07	30000.000000
25%	42.000000	2.016052e+07	49000.000000
50%	55.000000	2.017080e+07	64000.000000
75%	66.000000	2.017123e+07	80000.000000
max	101.000000	2.018073e+07	120000.000000

Fig. 6: Statistics of 'profile.json' dataset after reduction and removal of nulls.

```
[300] profile.describe()
```

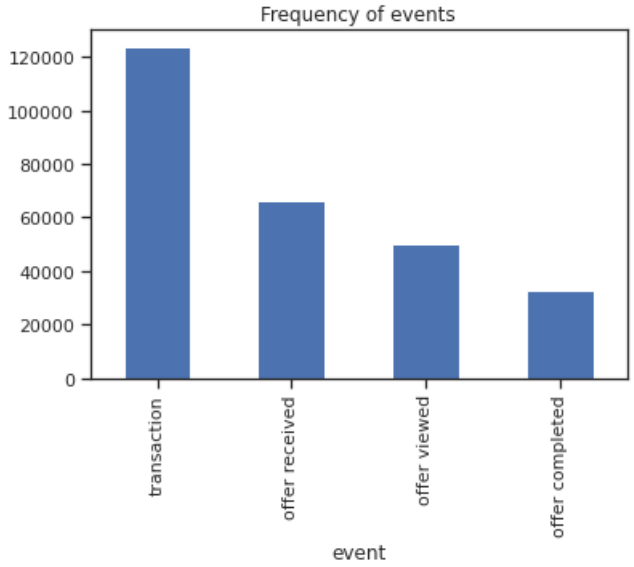
	age	became_member_on	income
count	17000.000000	1.700000e+04	14825.000000
mean	62.531412	2.016703e+07	65404.991568
std	26.738580	1.167750e+04	21598.299410
min	18.000000	2.013073e+07	30000.000000
25%	45.000000	2.016053e+07	49000.000000
50%	58.000000	2.017080e+07	64000.000000
75%	73.000000	2.017123e+07	80000.000000
max	118.000000	2.018073e+07	120000.000000

6 Exploratory Visualization

While the used notebook has many visualization, this report includes only the most important and relevant ones.

6.1 General Transcript Dataset Plots

As mentioned before, the 'event' feature of the transcript dataset has the values 'transaction', 'offer received', 'offer completed' and 'offer viewed'. The following plot compares between the number of events. The plot clearly shows that most of the events done are transaction events and the least done ones are the offer completion events. Also, this plot shows that more than half of the offers that was viewed get completed.



The following table shows the most and the least frequently seen, completed and received offers. The most completed and viewed offer is a discount offer that has a difficult of 10 and a reward of 2. On the other hand, the most received offer is the least viewed offer with a difficulty of 20 and a reward of 5.

max and min row with the column: offer completed

	offer completed	offer received	offer viewed	reward	channels	difficulty	duration	offer_type	id
max	5003	6652	6407	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4
min	0	6657	3487	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed

max and min row with the column: offer received

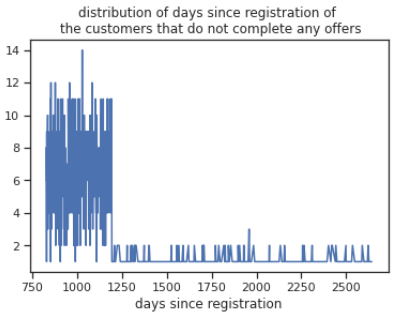
	offer completed	offer received	offer viewed	reward	channels	difficulty	duration	offer_type	id
max	3386	6726	2215	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7
min	4103	6576	6310	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d

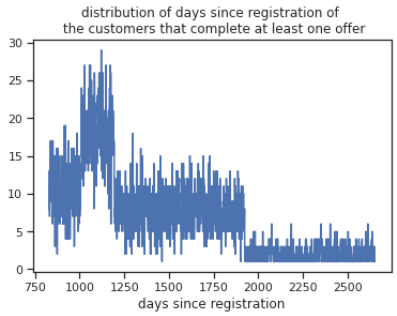
max and min row with the column: offer viewed

	offer completed	offer received	offer viewed	reward	channels	difficulty	duration	offer_type	id
max	5003	6652	6407	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4
min	3386	6726	2215	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7

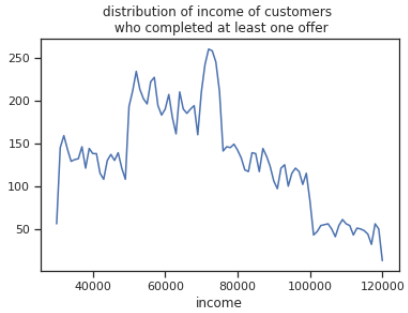
6.2 General Profile Dataset Plots

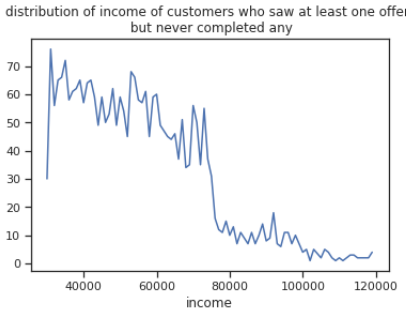
The following plots show a line plot of the distribution of days since registration of the customers. The left plot shows that customers who do not complete any offers are usually members since 1000 days. On the other hand, the right plot shows that customers who complete at least one offer have been members since 1000 to 1250 days. Also, the plot shows that in general customers who complete any offer could be members since 750 days or even more. That is why the left plot is not really useful to get a good insight.





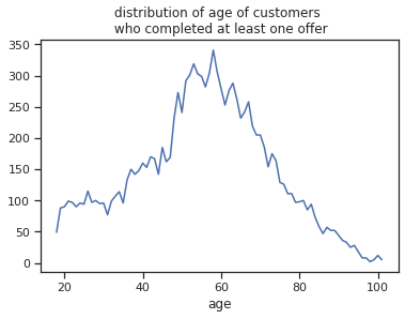
The following plots show a line plot of the distribution of income of the customers. The left plot shows that customers who complete at least one offer earn around 70k with an average income of 67k. Also, the plot shows that the group of customers that earn 55k are more likely to complete at least one offer. On the other hand, the right plot shows that customers who view at least one offer but never complete any are more likely to be earning less than 75k with an average income of 55k. In addition, customers with the highest income are most likely going to complete the offers.

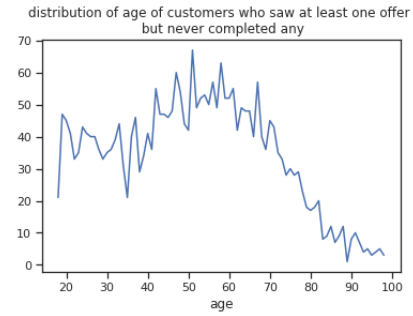




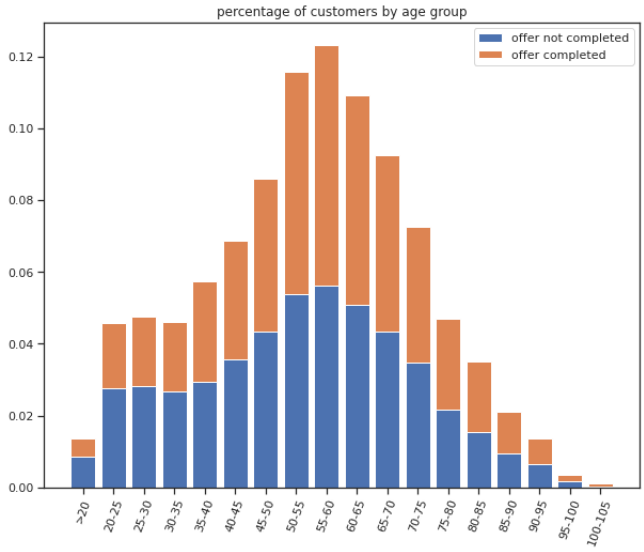
The following two plots show a line plot of the distribution of age of the customers. The left plot shows that customers who complete at least one offer earn around are around the age of 60 with an average age of 55. On the other hand, the right plot shows that customers who view at least one offer but never complete any are more likely to be around the age of 50 with an average age of 51.

When comparing the two plots, we can see that customers around age 90 are significantly more active compared to customers on age 80 and 100.

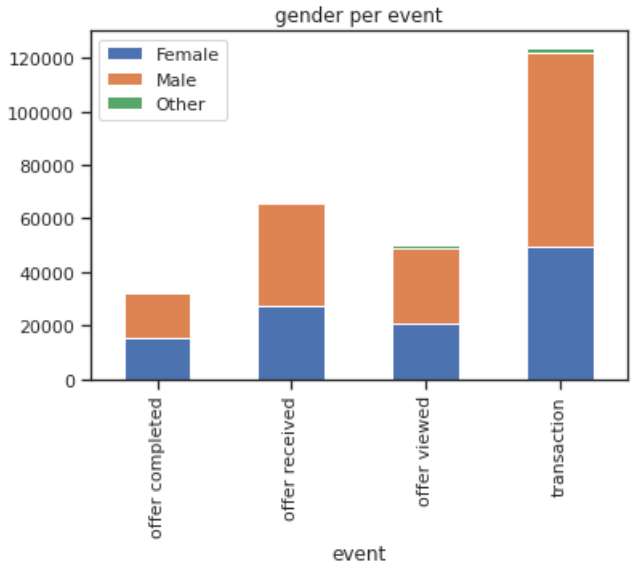




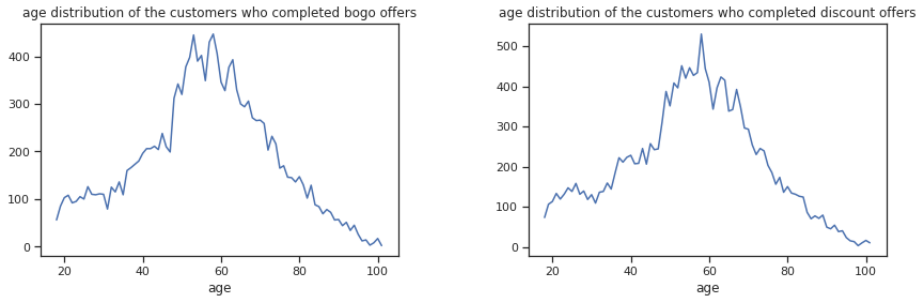
The following plot shows the percentage of customers who complete offers vs the ones who do not per age group. As visualized before, customer around 60 are more likely to complete the offers and they present the highest number of age group. Also, younger customers (less than 35) are more likely to not complete the offers.



The following bar plot shows the the gender groups per event type. In general the male gender is the most common. But when it comes to completing an offer there is almost a balance between the two genders. The other gender is low in general but they are more likely to view an offer rather than completing it.

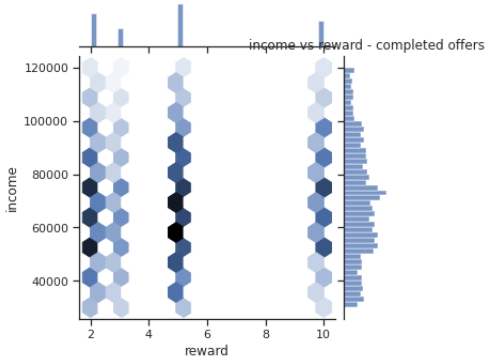


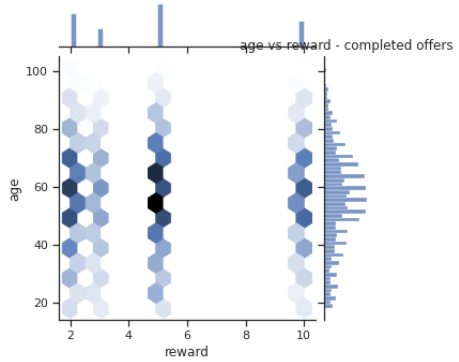
The following two plots show a line plot of the distribution of age of the customers who completes BOGO and discount offers. The left plot shows that customers of age 60 and 50 are more likely to complete BOGO offers. On the other hand, the right plot shows that customers of age 60 only are the most likely customers to complete the discount offers. However, customers of age 50 are still probable to complete the discount offers but with a probability less than BOGO offers. In general very old (<90) and young (<35) customers are more likely to not complete any offers however, younger customers are still more likely to complete the offers when compared to older customers (<60). The dataset follows a normal distribution which means that when the customer age go away from the center (age 60), this customer is more likely to not complete the offer.



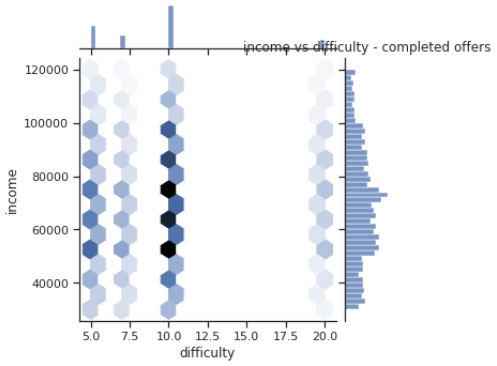
6.3 Reward and Difficulty Features Plots

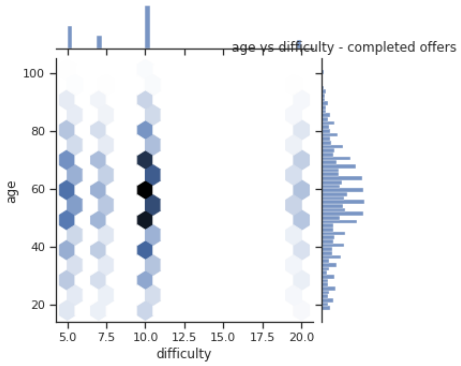
The following two plots prove that the most active groups are the ones with age 60 and income between 55k and 75k. This group is more likely to complete offers with reward level of 5. On the other hand, customers with age 40 are more likely to complete offers of reward level of 2 but customers of age 20 are more likely to complete offers of reward level of 5. In general customers are willing to complete offers with reward level of 5 with a few exception like customers of age 40.





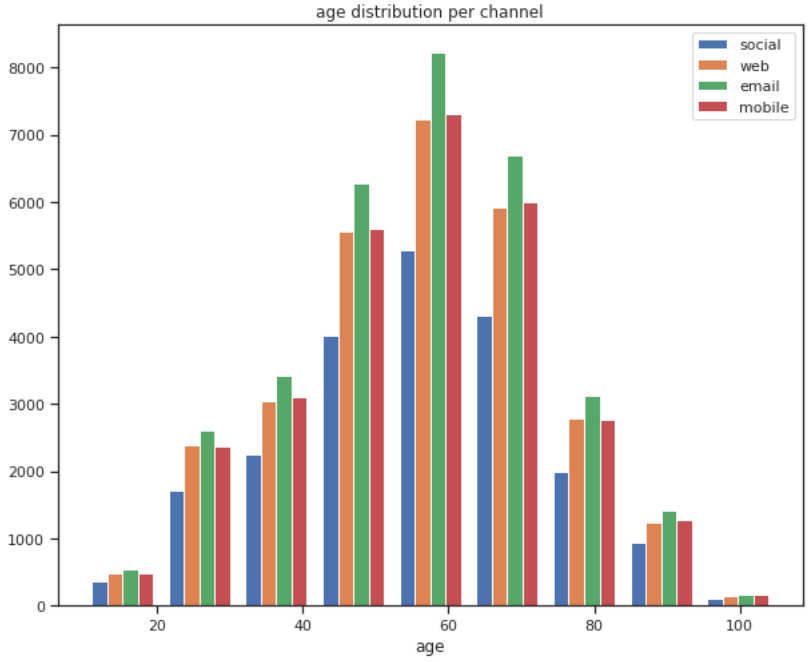
As mentioned before, the most active groups are the ones with age 60 and income between 55k and 75k. This group is more likely to complete offers with difficulty level of 10. In general customers are willing to complete offers with difficulty level of 10.



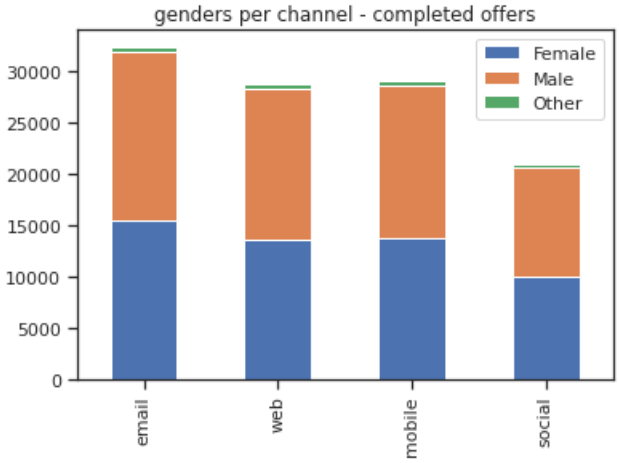


6.4 Channels Feature Plots

The following plot shows the age distribution per channel for the customers who complete offers. Email channel is by far the most used one and then mobile. most of the younger customers (< 25) complete offers that have been presented on a web channel. On the other hand older customers (> 80) complete offers that were presented on a mobile channel.

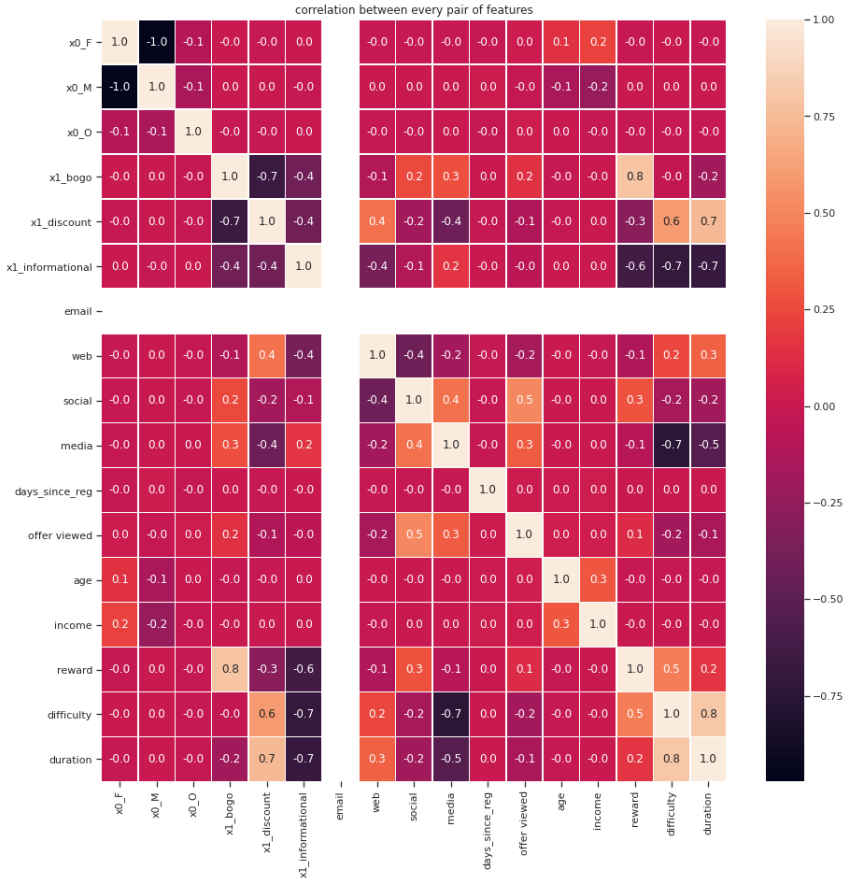


Gender differences per channel for the customers who complete offers can be seen from the following bar plot. The male gender is generally more dominant in every channel. The difference between male and female ratio in every channel is very small which means that men and women are almost alike when it comes to completing offers.

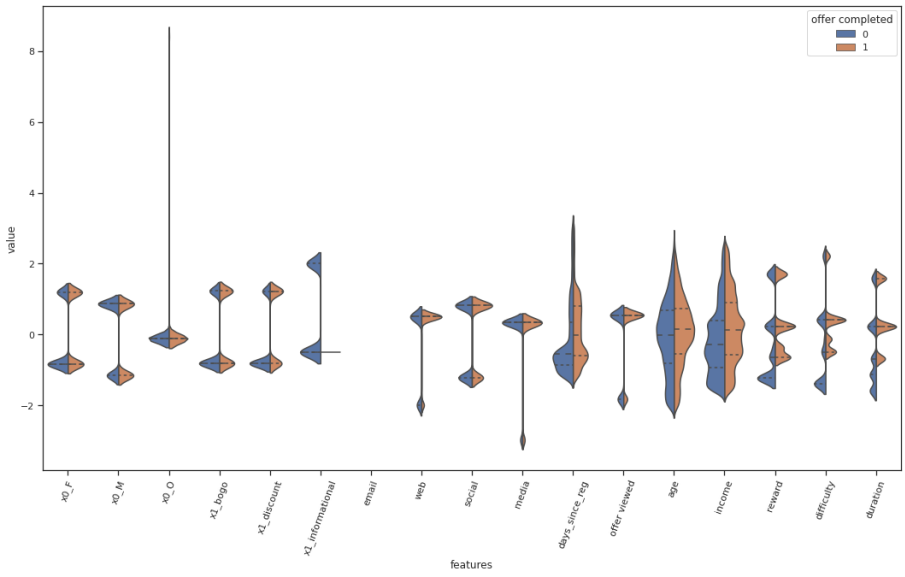


6.5 Features Correlation

To get better insights between the features, a correlation matrix heatmap is used to show correlation between every pair of features. The heatmap shows that BOGO offers are positively correlated with the reward feature. Also, discount offers are positively correlated with difficulty and duration features. Since the correlation between these features is not strong enough, no features is eliminated.



Lastly, to choose a model, a violin plot was used to see how good a feature can separate between the positive and the negative classes. Unfortunately, none of the features are able to completely separate the classes. Therefore, the dataset must be tested and evaluated with different models to get the best model for the final classification task.



7 Algorithms and Techniques

Since it was hard to find a model for the dataset just from looking at the visualization, I chose a few models to evaluate against each other and find a good final model for the classification task. The algorithms used for the classification and model selection task are: GradientBoosterClassifier, LogisticRegression, SGDClassifier, GaussianNB, RandomForestClassifier, DecisionTreeClassifier, KNeighborsClassifier. For the validation step, all previously-mentioned algorithms had the default parameters indicated by sklearn's library. In addition, any 'random-state' variable was set to 42. After validating the models, only the best ones were selected. The best models were: GradientBoostingClassifier, LogisticRegression, SGDClassifier, KNeighborsClassifier and RandomForestClassifier. The following list shows the parameters used for each model from the best models in the tuning step.

- GradientBoosterClassifier
  - learning_rate: 3/10, 4/10, 5/10 and 6/10.
  - n_estimators: 128, 129, 130, 131, 132.
  - min_samples_split: 2, 3.
  - min_samples_leaf: 6, 7, 8.
  - max_depth: 1, 2, 3.
  - max_features: 1, 2, 3, 4, 5, 6, 7, 8.
- SGDClassifier
  - loss: 'hinge', 'log', 'modified_huber', 'perceptron', 'squared_hinge'.
- LogisticRegression
  - C: 0.001,0.01,0.1,1,10,100.

- RandomForestClassifier
  - n_estimators: range from 50 to 1000 with a step of 100.
  - max_depth: 3, 4, 5, 6, 7, 8.
- GaussianNB
  - var_smoothing: 0.00000001, 0.000000001, 0.00000001.
  - priors: None.

After the model tuning step, the best parameters were:

- GradientBoosterClassifier
  - learning_rate: 0.5.
  - n_estimators: 130.
  - min_samples_split: 2.
  - min_samples_leaf: 7.
  - max_depth: 2.
  - max_features: 3.
- SGDClassifier
  - loss: 'log'.
- LogisticRegression
  - C: 1.
- RandomForestClassifier
  - n_estimators: 850.
  - max_depth: 7.
- GaussianNB
  - var_smoothing: 1e-08.
  - priors: None.

For training, the 'fit' function for the estimators were used with y value as the 'offer complete' feature and X as the other features.

8 Benchmark

For bench-marking all the previously-mentioned algorithms were used with its default values. The following list shows the F1-score of the algorithms.

- GradientBoosterClassifier: 0.805
- LogisticRegression: 0.78
- SGDClassifier: 0.78
- GaussianNB: 0.77
- KNeighborsClassifier: 0.77
- DecisionTreeClassifier: 0.74
- RandomForestClassifier: 0.78

These values were used to choose models after the model tuning step. In other words, the F1-score of the models produced after the model tuning steps were scored with the F1-score of the default-valued models to see if the model prediction was improved or not.

9 Data Preprocessing

For Data Preprocessing, five main steps were done:

- removing outliers and null values.
- one-hot encode the categorical data.
- transform date data.
- remove problematic features.
- applying standard scaler on the features for better classification.

In the first step, all the instances with age of 118 in the 'profile.json' dataset were removed as the these instances also had null values in 'income' and 'age' features. Also, all the instances that are 'transcrip.json' with a person id that does not exist in the 'profile.json' dataset were reomved.

The second step included encoding the features 'event', 'gender', 'channels' to one-hot encoded features. The following table shows the new features.

old feature	one-hot encoded features
'event'	'offer received', 'offer completed', 'offer viewed', 'transaction'
'gender'	'O', 'M', 'F'
'channels'	'email', 'mobile', 'web', 'social'

In the third step the column 'became_member_on' was transformed to the feature 'days_since_reg' which calculates the days in the form: today - 'became_member_on'.

In the fourth step, the feature 'offer received' was removed since all the offers were received. Also, the 'time' feature was removed as it was not easy to indicate the time after merging the 'transcript.json' and 'profile.json' datasets and grouping them by 'offer_id' and person id. In addition to the previously-mentioned features, all the features that indicate an id ('offer_id', 'person', 'id') were removed since it does not help in prediction. Another feature that got removed is 'transaction' since it is not directly related to the offer completion events but it could be used as an improvement in future.

Finally, the last step just applied the StandardScaler estimtor from the sklearn library to standardize the data.

10 Implementation

The implementation of the project could be summarized in 4 steps: data exploration & cleaning, data visualization, feature engineering and model selection.

In the first step, statistical analysis was used to get information about the features and the data, such as maximum and minimum values in each feature, standard deviation, mean and count. Using these data, I was able to identify outliers and null values which were removed by the end of this step.

The second step included using violin, bar, scatter, line, hex joint, heatmap plots to get more insights about the data. Some of the features had to be transformed before visualizing them. For this reason, some of the transformers were

already implemented in this step to be used later in the data engineering step. Also, the data used in the features violin plot had to be standardized to give a clear comparison between the features.

In the data engineering step, all the custom transformers (ChannelsTransformer, DateTransformer) and StandardScaler were used to construct a pipeline that can transform the merged data ('profile.json', 'transcript.json', 'portfolio.json). The channels and date transformers were not created in the beginning (at the beginning of the visualization step), but rather used as a function. On later stages, a sklearn Transformer was used to easily transform the data without repeating any code.

In the final step, the models stated in the algorithms step were trained to create a benchmark and evaluate the models against each other. Also, the data was separated into 3 sets: training, validation and test. The training set was used to fit the models. The validation set was used to tune the models and compare the models between each others. Finally, the test set was used to give a final evaluation of the model.

11 Refinement

As mentioned in the Benchmark and Algorithms and Techniques sections, a basic model with default parameters were used. The models used produced F1-scores of maximum 0.805 (GradientBoostingClassifier) but using GridSearchCV and RandomizedSearchCV of the sklearn library, the models were improved with a maximum F1-score of 0.808 (RandomForestClassifier).

12 Model Evaluation and Validation

As stated before a validation set was used to validate the model. In addition, a test set (20% of the data) was used to get the evaluation of the data. F1-score was sure for evaluation and validation since both the positive and the negative classed are almost balanced and both the negative and positive class are equally important.

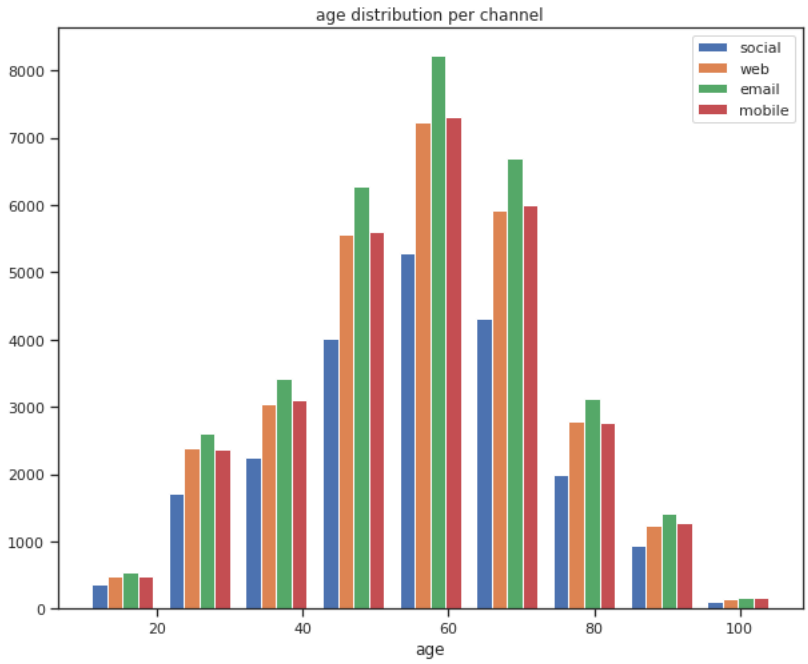
The model was expected to perform this performance, since most of the features are non-linear and do not create a good separation between the classes.

13 Justification

As stated in the previous section, the model was expected to form that way since it was hard to create a good separation between the classes. However, the model was able outperform the benchmark models.

14 Free-Form Visualization

One of the most interesting insights this project shows is that most of the active customers are around the age of 60. Also, young customers get attracted to more to offers that were offered on the web. On the other hand, older customers prefer offers that was offered on the web. However, the email channel is the most common one among all the channels of every age group.



15 Reflection

As stated before, the project can be summarized in 4 main steps:

- data exploration & cleaning.
- data visualization.
- feature engineering.
- model selection.

I found the model selection step quite difficult since the data was hard to fit and to suggest a model to. On the other hand, I found the data visualization step very interesting and rewarding.

16 Improvement

There is a very obvious room for improvement which is including the deleted features ('transaction', 'time'). Those features, I believe, could improve the prediction quite a bit. Also, trying more hyper-parameters combinations and neural networks could be a very good choice.