Starbucks Capstone Challenge

❖ Problem Introduction

Starbucks Corporation is an American multinational chain of coffeehouses and roastery reserves headquartered in Seattle, Washington. It is the world's largest coffeehouse chain. As of November 2021, the company had 33,833 stores in 80 countries, 15,444 of which were located in the United States.

The given data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. Once every few days, Starbucks sends out an offer to users of the mobile app. An offer can be merely an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users might not receive any offer during certain weeks. Not all users receive the same offer, and that is the challenge to solve with this data set.

Given transactional data showing user purchases made on the app including the timestamp of purchase and the amount of money spent on a purchase. This transactional data also has a record for each offer that a user receives as well as a record for when a user actually views the offer. There are also records for when a user completes an offer.

❖ Strategy to Solve the Problem

This data set is a simplified version of the real Starbucks app because the underlying simulator only has one product whereas Starbucks actually sells dozens of products. The task is to combine transaction, demographic and offer data to determine which demographic groups respond best to which offer type.

In order to solve the problem three machine learning models: k-Nearest Neighbors (kNN), Decision Tree Classifier and Random Forest Classifier. Whichever performed the best would be taken into consideration for the feature selection.

❖ Metrics

In order to evaluate how good the model is, the F1 score will be calculated. The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. The F1 score is primarily used to compare the performance of two classifiers, for example, suppose that classifier A has a higher recall, and classifier B has higher precision. The closer the score is to 100, the better the model.

❖ EDA (Exploratory Data Analysis)

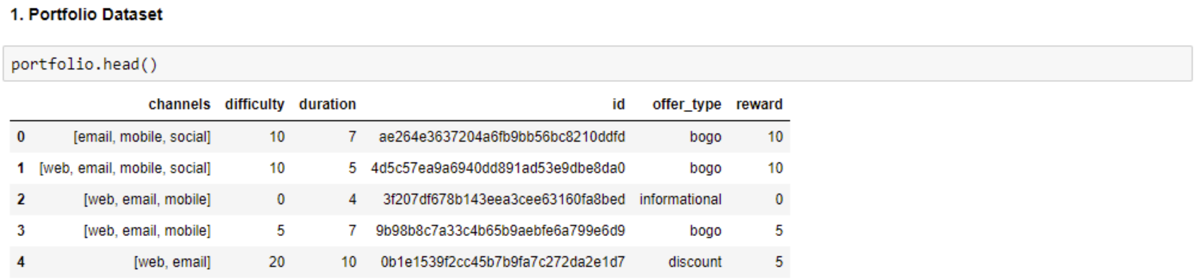
The data is contained in three files:

- portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)
- profile.json - demographic data for each customer
- transcript.json - records for transactions, offers received, offers viewed, and offers completed

Here is the schema and explanation of each variable in the files:

portfolio.json

- 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)



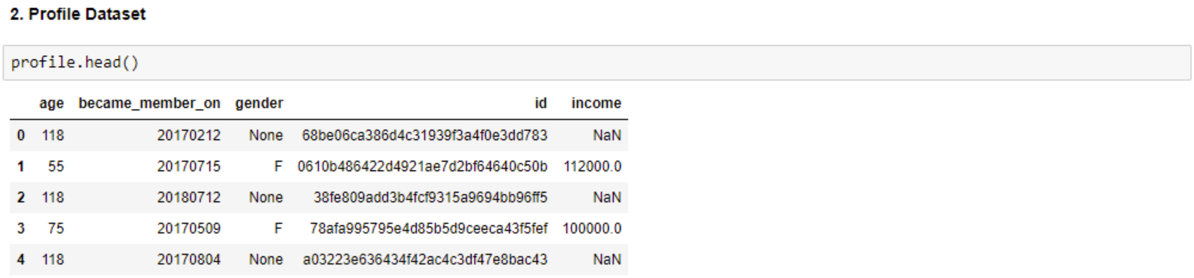
1. Portfolio Dataset

```
portfolio.head()
```

	channels	difficulty	duration	id	offer_type	reward
0	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10
1	[web, email, mobile, social]	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10
2	[web, email, mobile]	0	4	3f207df678b143eea3cee63160fa8bed	informational	0
3	[web, email, mobile]	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5
4	[web, email]	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	5

profile.json

- 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



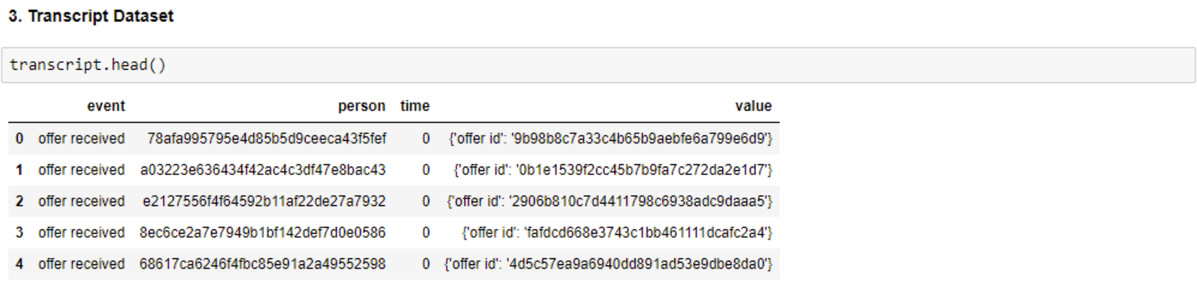
2. Profile Dataset

```
profile.head()
```

	age	became_member_on	gender	id	income
0	118	20170212	None	68be06ca386d4c31939f3a4f0e3dd783	NaN
1	55	20170715	F	0610b486422d4921ae7d2bf64640c50b	112000.0
2	118	20180712	None	38fe809add3b4fcf9315a9694bb96ff5	NaN
3	75	20170509	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0
4	118	20170804	None	a03223e636434f42ac4c3df47e8bac43	NaN

transcript.json

- 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



3. Transcript Dataset

```
transcript.head()
```

	event	person	time	value
0	offer received	78afa995795e4d85b5d9ceeca43f5fef	0	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}
1	offer received	a03223e636434f42ac4c3df47e8bac43	0	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}
2	offer received	e2127556f4f64592b11af22de27a7932	0	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}
3	offer received	8ec6ce2a7e7949b1bf142def7d0e0586	0	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}
4	offer received	68617ca6246f4fbc85e91a2a49552598	0	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}

Figure 1.

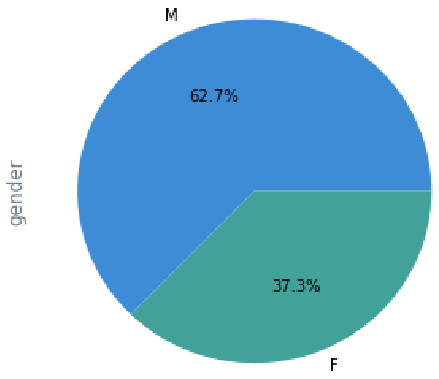


Figure 2.

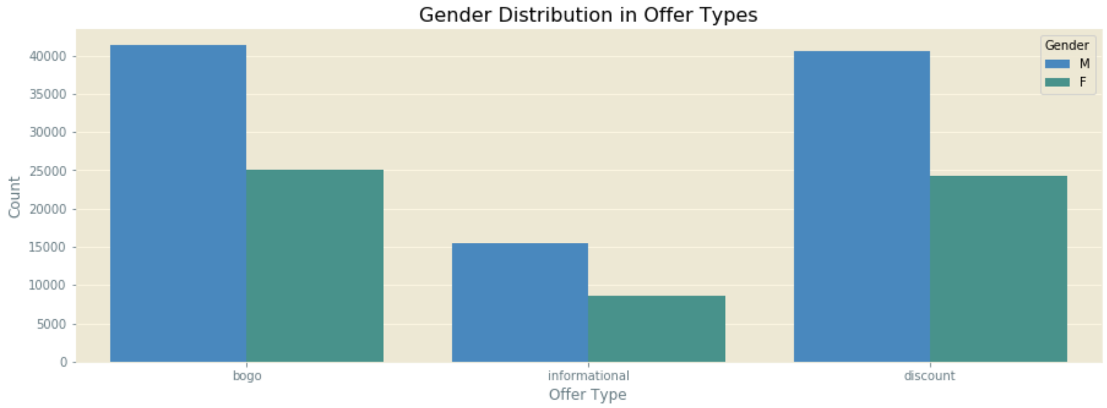


Figure 3.

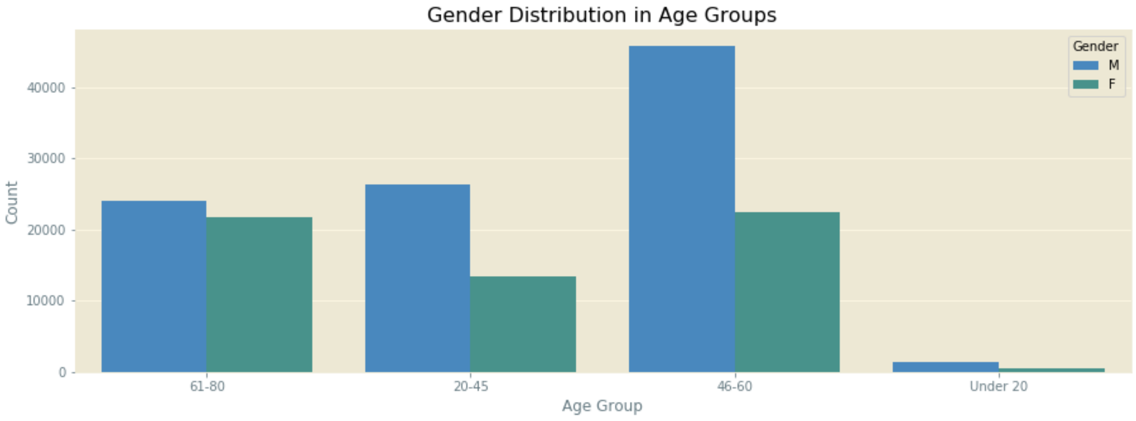
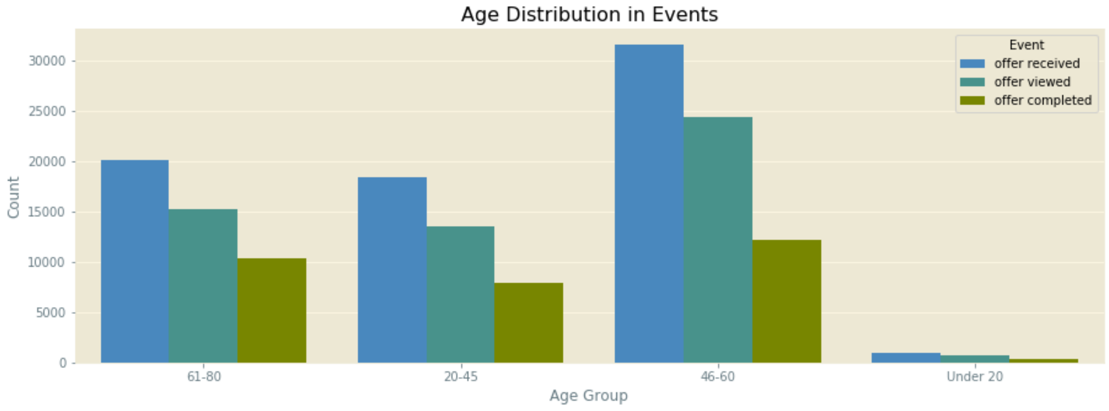


Figure 4.



- Males represent 62.7% and females 37.3% of the customers who use the Starbucks app (Figure 1)
- Both males and females in age group from 46 to 60 use the app the most (Figure 3)
- The most popular and preferred offers are discount and bogo (Figure 2)
- There is a lower number of customers who actually complete the offer as compared to the ones who just view & ignore it (Figure 4)

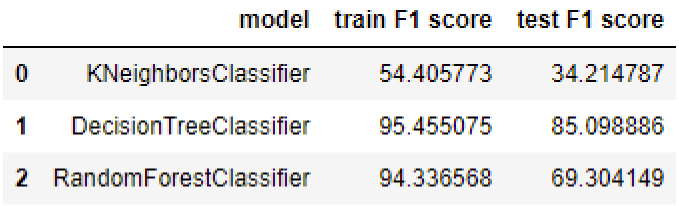
❖ Modelling

For this analysis, three classifiers are built: KNeighborsClassifier as a benchmark to evaluate the F1 score metric in comparison to other two classifiers: DecisionTreeClassifier and RandomForestClassifier.

❖ Hyperparameter Tuning

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. For this analysis, hyperparameter tuning has not been performed - future improvement to maximize model's predictive accuracy.a

❖ Results



	model	train F1 score	test F1 score
0	KNeighborsClassifier	54.405773	34.214787
1	DecisionTreeClassifier	95.455075	85.098886
2	RandomForestClassifier	94.336568	69.304149

❖ Conclusion/Reflection

As we can see from the above table both models DecisionTreeClassifier and RandomForestClassifier performed better than KneighborsClassifier. DecisionTreeClassifier got the best score: the train F1 score is 95% and the test F1 score is 85%, while RandomForestClassifier got 94% of the train F1 score and 69% of the test F1 score. The scores are pretty good to use for the classification purpose to predict whether a customer will respond to an offer.

❖ Improvement

To improve the model's performance we could run a longer experiment having more customers in our data set. We could collect demographic data, for example, location where one of the Starbucks products has been purchased, scores of drinks reviews, day and at what time the transaction has been made etc. Also, we could also perform A/B testing experiments between similar users in order to see what offers are the most effective.

❖ References

- https://learn.udacity.com/nanodegrees/nd025/parts/cd1971/lessons/81217aa8-0633-455b-89ee-0a19bc1563ca/concepts/76d802ca-2deb-4fd0-a707-8dc0e0f3260c
- https://www.educative.io/answers/what-is-the-f1-score
- https://scikit-learn.org/stable/tutorial/index.html
- https://scikit-learn.org/stable/modules/preprocessing.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- https://scikit-learn.org/stable/modules/tree.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- https://stackoverflow.com/questions/70322857/convert-list-using-zip-to-dictionary
- https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html