Proposal

March 10, 2020

1 Proposal for Starbucks Capstone Project

1.1 Domain Background

Starbucks, an American Coffee Company, which offers discount to mobile users who purchase items through the mobile app.

As a retail company that sells products to its consumers, increasing sales would be a major concern for them, and as a businesses look forward to increasing sales and revenue every year, establishing means to help with this using technology would be at the top of their list.

Using a simplified version of the real Starbucks app based on a simulator we seek to be able to determine demographic groups of consumers and also how they best respond to offers.

1.2 Problem Statement

The goal of this project is to determine among demographic groups what **offer** they best respond to.

> An **offer** is an advertisement for a drink or an actual offer such as a Buy One Get One Free or a discount sent to users to spike interest in the product.

A potential solution would be using data gotten on demographics and transactions conducted by consumers of starbucks products and building a model to determine what offer to offer to specific consumers.

1.3 Datasets and Input

The dataset being used contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. ### 1. portfolio.json This contains offer ids and meta data about each offer (duration, type, etc.)

```
In [38]: import pandas as pd
         portfolio = pd.read_json('portfolio.json', orient='records', lines=True)
         portfolio.head(1)
```

```
Out[38]:                  channels  difficulty  duration  \
         0  [email, mobile, social]          10         7

                                      id offer_type  reward
         0  ae264e3637204a6fb9bb56bc8210ddfd       bogo      10
```

1.3.1 2. profile.json

This contains the demographic data for each customer.

```
In [39]: profile = pd.read_json('profile.json', orient='records', lines=True)
         profile.head(1)
```

```
Out[39]:    age  became_member_on gender                                id  income
         0  118           20170212   None  68be06ca386d4c31939f3a4f0e3dd783     NaN
```

1.3.2 3. transcript.json

This contains records for transactions, offers received, offers viewed, and offers completed

```
In [40]: transcript = pd.read_json('transcript.json', orient='records', lines=True)
         transcript.head(1)
```

```
Out[40]:             event                            person  time  \
         0  offer received  78afa995795e4d85b5d9ceeca43f5fef     0

                                                 value
         0  {'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}
```

```
In [41]: # Merge the transcript dataframe with the portfolio dataframe (metadata of offers)
         transcript['offer_id'] = transcript['value'].apply(lambda value: value.get('offer id'
         transcript['amount'] = transcript['value'].apply(lambda value: value.get('amount'))
         transcript.drop('value', axis=1, inplace=True)
         transcript.head(1)
```

```
Out[41]:             event                            person  time  \
         0  offer received  78afa995795e4d85b5d9ceeca43f5fef     0

                                offer_id  amount
         0  9b98b8c7a33c4b65b9aebfe6a799e6d9     NaN
```

```
In [42]: merged = transcript.merge(portfolio.rename(dict(id='offer_id'), axis=1), on='offer_id
         print(merged.shape[0], transcript.shape[0])
         merged.head(1)
```

```
306534 306534
```

```
Out[42]:             event                            person  time  \
         0  offer received  78afa995795e4d85b5d9ceeca43f5fef     0

                                offer_id  amount              channels  difficulty  \
         0  9b98b8c7a33c4b65b9aebfe6a799e6d9     NaN  [web, email, mobile]         5.0

            duration offer_type  reward
         0       7.0       bogo     5.0
```

```
In [43]: merged['offer_type'].fillna('None', inplace=True)
         merged['amount'].fillna('None', inplace=True)
         merged['offer_type'].value_counts(normalize=True)

Out[43]: None             0.562848
         bogo             0.182518
         discount         0.169599
         informational    0.085035
         Name: offer_type, dtype: float64
```

1.4 Solution Statement

A solution I propose would be to apply a Classification Machine Learning model after thorough analysis on demography data and transactions to determine if a user would respond to a particular offer or not.

With the information that there are 4 offers:

- Bogo
- Informational
- Discount
- None

This makes our model a multi-label classification model which would be useful in determine which of the offers (classes) listed above that a potential consumer would respond best to.

With the knowledge that some consumers do not purchase an item with an offer in mind, they do not need to recieve any offer, is the reason why **None** is also listed as a class.

1.5 Benchmark Model

Since this solution entails solving as a multiclass classification problem, it would be evaluated and weighted by each class.

The F-score can also used for evaluating Multiclass classification problems. In this setup, the final score is obtained by micro-averaging (biased by class frequency) or macro-averaging (taking all classes as equally important). For macro-averaging, two different formulas have been used by applicants: the F-score of (arithmetic) class-wise precision and recall means or the arithmetic mean of class-wise F-scores, where the latter exhibits more desirable properties. [1]

1.6 Evaluation Metrics

An evaluation metric that would be used on this problem would be the F1-score, specifically the Macro F1 metric [2]

1.7 Project Design

First, would be to merge all datasets together, `transcript.json` with both the `porfolio.json` and `profile.json`, as this all hold data that describe entities describe in the transcript.

Second would involve data analysis to determine what features are useful for determining what offer a consumer best responds.

N.B while it is not very important to construct a model once thorough data analysis has been done, I suggest doing this to enable prediction of what a class a consumer would be in.

Using a Classification Model that would be trained on the data we will build a multi-category classifier that would be able to identify which offers consumers would respond to best.

Since there are 4 classes we are checking for which seems to be well distributed except for the

```
None            56,2848
bogo            18.2518
discount        16.9599
informational   8.5035
```

A (Random Forest)[3] would be used in this problem as this is a very useful ensemble algorithm. This classification model would be used to predict which offer a demographic would best respond to.

1.8 References

1. Wikipedia (F1 Score)

   https://en.wikipedia.org/wiki/F1_score

2. Macro F1

   https://arxiv.org/abs/1911.03347

3. Random Forest

   https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn