Capstone Project Proposal

Domain Background:

Startbucks has 1 month data from offers they have made to their clients. These offers are made on an individual level, and each individual may have gotten more than one offer. From all of these offers, the starbucks datasets allow us to see which offers had the best overall reach, which offers converted to sales, and which offers had negative effect. The objective of this project is to analyze this dataset to better understand the offers, and to be able to model and predict which offers will have better results based on a given demographics. This will allow startbucks to be able to send promotions with better intelligence. Also, this will allow starbucks to not lose money, if they send a promotion, and the promotion has no impact if the person buys something or not.

Problem Statement:

From this project, there are a couple of questions which we would like to answer.

- What is the best offer, not just population as a whole, but also individually too.
- Which offers influences a person to spend more money
- Which offers have negative impact.
- Do different offers affect different demographics differently
- Finally, the biggest question we would like to answer, is “Give a certain demographics (Age, Income, Gender, And time since membership) can we predict if an offer will lead to increased sales?”
    - Along similar lines, we can also have a model that given (Age, Income, Gender, And time since membership, offer), we can predict if the offer will have a postive or no effect.

Datasets and Inputs:

The input dataset consists of simulated data that mimics customers behavior on the starbucks mobile app. Each customer receives different offers. This is all kept track inside the dataset. Offers have a validity period, and expire in differnt times. The data also consists of transactional data, showing user purchaseds made on the app.
Obtained form the descrition, here is what the initial dataset consists of
*
*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)

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

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

From this dataset, cleaning will be done. With cleaning, the first thing that will be done is to make sense of the data, and obtain some metrics for the past.

The next part, will be to create a new dataset. This new dataset will be separated by the inputs (age, gender, income) and how much each person spent given an offer. From this it may be simplified to binary, if an offer was affective or not.

Along these lines, the dataset will be separated só that given a certain demographics, the offer with the best results will be counted. This can help with predictions as to which offer to give a given person. This part is a little tricker, and more data analyes will need to be done to be able to come p with how to separate this data.

After cleaning some dataset, a csv file was made. The follow shows a pd.head() of a random sample. We can see from the data if an offer led to a transaction or not. This will give us an idea of how much was spent with a given offer. From here we can clean this more to obtain the persons information and train the model to see if the offer will cause a transaction.

	Unnamed: 0	event	person	time	offer id	reward
118878	210711	offer received	78e971bddb2842208b226249593b424f	504	5a8bc65990b245e5a138643cd4eb9837	NaN
11860	11860	offer received	258df44b49644e28873287c18a2d3474	0	2906b810c7d4411798c6938adc9daaa5	NaN
130332	227530	offer completed	7af54ea6250c4aa0941733ec8a18b821	522	2906b810c7d4411798c6938adc9daaa5	2.0
37580	63503	offer received	159ba13189d54daeaa55f4f8399d9779	168	fafdcd668e3743c1bb461111dcafc2a4	NaN
55457	110032	offer viewed	95e86569c15d455fa02041dd12f3d308	330	9b98b8c7a33c4b65b9aebfe6a799e6d9	NaN

	event	person	time	value
100	offer received	65f06566a2a14f64b003964f211635e1	0	{'offer id': '3f207df678b143eea3cee63160fa8bed'}
29569	offer viewed	65f06566a2a14f64b003964f211635e1	48	{'offer id': '3f207df678b143eea3cee63160fa8bed'}
29570	transaction	65f06566a2a14f64b003964f211635e1	48	{'amount': 2.98}
42072	transaction	65f06566a2a14f64b003964f211635e1	102	{'amount': 7.5600000000000005}
47626	transaction	65f06566a2a14f64b003964f211635e1	132	{'amount': 1.6400000000000001}
52302	transaction	65f06566a2a14f64b003964f211635e1	162	{'amount': 1.6600000000000001}
53271	offer received	65f06566a2a14f64b003964f211635e1	168	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}
69459	offer viewed	65f06566a2a14f64b003964f211635e1	174	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}

Another interesting thing, is that from the cleaned data, we know that
35% of the clients viewed the offer
45% received but did not view
20% had viewed and had a transaction.

This already gives us an idea and a baseline. If we can improve this 20%, we can say that the model is successfull or not. Also, from these data, we can acutally trrain the model to predict these different classes. Or we may need to experiment, maybe the clients that did not view the offer, we shouldn't focus on them. Therefore, we could train the dataset with the clients that viewed the offer and did not spend, and those that viewed and spent money.

Solution statement:
The solution for the problem is two fold:
Undertand and obtain metrics that show how successful past offers were and
A model that predicts given age, gender, income, offer, if the client will be positively affected by the offer, or if the offer will have no impact if the client buys or doesn't buy along with a model that decides on a offer to give, given the same inputs as mentioned above.

The model that we will be focusing on will be an Xgboost model. Xgboost models have shown to work well with labelled datasets. After clearning the dataset, we will be able to seperate the dataset from features to labels. The label being if the the given demographics if the offer was successfull or not.

Benchmark model
For the model well be training, we will be using an xgboost model to make the predictions. For the bench mark model, the model can train with a simple logistic regression to predict if the offer will. Even simpler, random selecter to pick a random offer to give a person. From these benchmark, the accuracy of positive results vs negative results can be measured, along with the total amount spent. These two metrics can be good enough.

Evaluation Metrics:
For the model part of this project, the main metrics we will be measuring will be confusion matrix with precisin, recall, and final accuracy. With these metrics, we can conclude well enough how many false positives / false negatives we have, and overall how our model is working.

Another metric that can be used is the total amount spent. To measure this, it will be tricker as our data is only past, there is not way to no future and do A/B testing.

Project Design:

Part1:
For the first part of the project, the design is going to be simple. Analyze the data, and look for interesting trends. For this part, we hope to gain insights into what worked and what didn't. And get a general distribution for the dataset.

Part2:
For the second part, the part to train a model to predict if an offer will have positive results, and to predict which offer is best given a demographics, there are a couple steps that must be taken.

1. **Create The dataset:**
    1. by analysing the data we have, we will need to create a dataset that will allow us to see if given offers were affective or not. For this we will create two datasets. One which will be for prediction. This will have the inputs, and the offer given, and we will predict if the offer was successfull or not.
    2. The next dataset will be the inputs, and based on a demographics, which offer had the best impact. To create this dataset, we will needed to change the input data to classes. And from here, the output will be based on a certain amount of classes, which offer was the best.
    3. Next the dataset will need to be separated for Training, validation, and test. Probably will use 70% for training, 15% for validation, and 15% for test. This is subject to change after an analysis is done.

4. Preprocessing for data may not be needed as much. We will test of encoding the inputs will make a difference. The only preprocessing step needed may be to normalize the input values. All these values can be normalized from 0-1. However, for xgboost this is not always needed, só we will test these different cases.

2. Get Baseline and Train Model

1. The next step once there is data, will be to run the baseline models to get simple results. These results will give us an idea from where to move and the decisions to make
2. Once the baseline is obtained, the model will be trained with xgboost, hyperparameters will be tested.
3. Once we are happy with the validation results, the model will run on the test set to see how well the model did.
4. The model we will be using with be imported as follow

```
1. import xgboost as xgb
   xgb_model = xgb.XGBClassifier(objective="multi:softprob",
   random_state=42)
2. xgb_model.fit(X, y) #Assume X and y as feature and label
3. y_pred = xgb_model.predict(X,y)
```

5. If Needed, we can use kfolds for training, however, I hypothesize that there is enough data that this will not be needed. However, if deemed needed this will be used.

3. Obtain Metrics

1. Metrics will be obtained on the validation dataset.
2. As mentioned above, we will use accuracy, precision, and recall.
3. We will also use correlation matrix
    1. this can be obtained with either pandas or sklearn.