

Machine Learning Engineer Nanodegree Capstone Proposal

Chinmay Gopal August 31th , 2020

Proposal

The objective of this project is to increase Starbucks's revenue by using user data as well as spending habits to create a model which targets infrequent customers at the Company. The model aims to provide offers in an economical way to customers to make them a regular at Starbucks.

Domain Background

The 80-20 rule is prominently visible throughout many different businesses. The 80-20 rule in business economics refers to the phenomena that roughly 80% of revenue comes from about 20% of your customers. More specifically this refers to regular Starbucks customers that facilitate about 80% of the company's total revenue. Since frequency customers create so much revenue understanding these customers and their behaviour is very important for the future.

Consequently, it should come to no surprise that transforming infrequent customers into regulars is paramount to the company's future. Using data from infrequent customers such as, which offers compelled them to buy as well as which offers kept them engaged the longest, we can drastically improve the company's future revenue.

Problem Statement

The main purpose of this project is to transform infrequent customers into regular customers by using the most economical offers possible. This dynamic problem can be addressed in two major steps

- A) Using data to figure out which customers are irregulars
- B) Using prior offer and customer data to determine which offers at what frequency is needed to make that customer become a "regular".

Problem A- Though an initial EDA analysis we can create a threshold for customers who purchase Starbucks frequently. We can then get the remaining customers by eliminating people who do not match the threshold.

Problem B - The remaining data can then be run through a simple neural net or a Binary classifier with transactions, offers received, offers viewed, and offers completed as inputs. The classifier or Neural net would then output whether or not a customer would respond to the offer. Based on this output we would select what offers to send in the future.

19

Datasets and Inputs

In this project I would only need:

Offer details - offer type, difficulty, reward, duration, event, value

Customer details - Age, Date of initiation and income

Offer details are needed to differentiate the types of offers available at Starbucks. For Example, most customers might like but 1 gets 1 free offer etc. Using this information is key in determining which offers to give to our customers.

On the other hand, Customer details are key in determining what kind of a person you are. For example, a customer who is female may like certain offers more than that of a male or a richer customer might like expensive items and require fewer offers.

The data set will first be put into a large table and columns of metadata which is not required would be eliminated. These data points would then be used as features in my neural net or classifier to determine whether or not to send them to the customer.

On the other hand, the output for the model will be offer id. For proper Neural network and Binary classifier performance, the data will be split 70% for training and the remaining 30% for testing. The entirety of the data used in the project will be provided by Starbucks through Udacity.

Solution Statement

Each data feature mentioned above should be input in a neural net of the classifier. This supervised method would be trained on data from previous years. The objective of the neural net is to learn certain patterns between customer type and offer details. Initially, I am hoping to create a model with a few hidden layers and a relatively high dropout rate.

29

Evaluation Metrics

A suitable evaluation metric test would be to test the accuracy of our model with new data the model has not seen before. Evaluation is a measure of how accurate the model is in predicting whether or not a specific customer responds to the offer. We can also use a special MSE like qualitative of revenue lost to determine how well a model has performed. For instance, if our model decides not to give a certain person an offer but in actuality, the person does receive an offer and buys an item our model will be deducted the revenue from that particular item. Besides, if our model decided to give a certain person an offer when in actuality that person doesn't receive an offer but still purchases that item the lost revenue will be the discount. If we have enough data this test could be run multiple times to see if the new model consistently makes accurate predictions.

Benchmark Model

In this project, the benchmark would be another SVM or a random offer generator which outputs random offers to people. Using the data provided the created Neural net or Binary Classifier is expected to perform better than SVM and Random offer generators which would indicate that we are making the company more revenue.

##Project Outline

1. Clean the data and make sure no missing values are present. If missing values are present either eliminate them completely or use the median from previous rows depending the the feature type.(We can't take the median of an offer code but we can take the median of time)
2. Create some graphs emphasising the difference between frequent and infrequent customers. Create a reasonable threshold using features to differentiate types of customers. Remove frequent customers from the dataset.
3. Split the remaining dataset 70-30 to training and testing.
4. Using the mentioned data columns as features and use output as weather or not the customer buys items.
5. Evaluate using accuracy as well as revenue wasted though the mentioned method