Starbucks Capstone Project Proposal

Stephen Blystone

Udacity ML Engineer Nanodegree | December 2019

Introduction

This Starbucks Capstone project is part of the Udacity Machine Learning Engineer Nanodegree. Udacity partnered with Starbucks to provide a real-world business problem and simulated data mimicking their customer behavior.

Domain Background

As a father of twin toddlers, I am a frequent Starbucks customer. When they wake up in the middle of the night, I need caffeine to function the next day. Starbucks has a Rewards program that allows me to earn points for purchases. There is also a phone app for their Rewards program where they send me exclusive personalized offers based on my spending habits.

This project is focused on tailoring those personalized offers to the customers who are most likely to use them. The Machine Learning terminology for this is "propensity modeling". Propensity models are "often used to identify the customers most likely to respond to an offer" (Childs, 2002).

Logistic regression, Support Vector Machines (SVMs), and neural networks are common methods used to perform propensity modeling (HG Insights, 2018).

Problem Statement

We want to determine which kind of offer, if any, to send to each customer based on their purchases and interaction with the previously sent offers. Some customers do not want to receive offers and might be turned off by them, so we want to avoid sending offers to those customers.

Datasets and Inputs

For this project I will be using the dataset provided by Starbucks. The data consists of 3 files containing simulated data that mimics customer behavior on the Starbucks Rewards mobile app.

Portfolio.json contains info about the offers, profile.json contains info about the customers, and transcript.json contains info about customer purchases and interaction with the offers.

We have information about 10 offers: 4 BOGO, 4 discount, and 2 informational. We also have information about 17,000 customers and a transcript containing 306,534 purchases and offer interactions.

A customer can interact with an offer by receiving it, viewing it, or completing it. It is possible for a customer to complete some offers without viewing them.

To split the customer data into training/validation/testing sets I will use a 60/20/20 split for the customers. 10.2k customers will be for training, 3.4k will be for validation and 3.4k for testing. The

reason for these numbers is to ensure a large enough dataset to validate hyperparameters and to perform testing.

The dataset is balanced. To determine this I looked at the value counts for all events listed in transcript.json.

```
transaction        138953
offer received      76277
offer viewed        57725
offer completed     33579
```

We are primarily concerned with whether a customer completes the offers they have received. To determine what percentage completed their offers, I performed the following calculation:

$$\frac{76{,}277 - 33{,}579}{76{,}277} = \frac{42{,}698}{76{,}277} = 55.97\%$$

That means that 55.79% of the people completed their offers, while 44.03% received offers but did not complete. These percentages are close enough to consider this a balanced dataset.

Below are details about each file:

1) portfolio.json

   This contains information about the offers sent over a 30-day trial period, each having a unique id.

   (10 offers x 6 features defined below)

   - reward (int) – reward given in dollars for completing an offer.
   - channels (list of strings) – platform the offer was sent on. Combination of email, mobile, social, and web.
   - difficulty (int) – minimum dollar amount customer must spend to complete an offer.
   - duration (int) – time in days that the offer is valid.
   - offer_type (string) – type of offer. Either BOGO, discount, or informational.
   - id (string) – unique hash identifying the specific offer.

2) profile.json

   This contains customer information assigning each a unique id.

   (17,000 customers x 5 features defined below)

   - gender (string) – 'M', 'F', or 'O' for other. If none provided, default value is null.
   - age (int) – age of the customer. If none provided, default value is 118 (birth year is 1900).
   - id (string) – unique hash identifying the specific customer.
   - became_member_on (int) – date customer became a member. YYYYMMDD format.
   - income (float) – customer's income. If none provided, default is null.

3) transcript.json

This contains each customer's purchases and whether they received, viewed, or completed an offer.

(306,534 events x 4 features defined below)

- person (string) – unique customer id.
- event (string) – Type of event that occurred. Either "transaction", "offer received", "offer viewed", or "offer completed".
- value (dict of strings) – either an offer id or transaction amount, depending on the event.
- time (int) – time in hours since start of test. Range is between t=0 and t=714

Solution Statement

To solve this, I will use supervised learning models to determine the propensity for a customer to complete an offer.

I plan to implement Logistic Regression (benchmark model), Support Vector Machines, and Neural Network algorithms for my models.

Logistic Regression is a common technique used for binary classification problems. Propensity models are a form of binary classification, since they are concerned whether a customer is likely to respond to an offer or not.

Support Vector Machines (SVMs) attempt to find the best dividing hyperplanes in the data to determine whether to send an offer or not. I will use the kernel method to add non-linearity to the SVM model.

Neural Networks are universal function approximators. This means they can approximate any smooth polynomial function, allowing them to better find the boundaries in high-dimensional data. I plan to use a neural network with 3 layers. The first layer is the input layer and will contain nodes for each feature. The second layer is the hidden layer and I will test different numbers of hidden nodes during my hyperparameter tuning. The final layer is the output layer and will consist of a single node using a sigmoid activation function. This will output a value between 0 and 1, where 1 if the customer is likely to use the offer, and 0 otherwise. I will determine during my testing where the threshold should be on determining whether to send the offer or not.

I plan to use a grid search for my hyperparameter tuning.

Benchmark Model

For a benchmark model, I will first build a logistic regression model and compare all other models against that. This should give me a solid benchmark to compare against, since logistic regression is "quite efficient in dealing with the majority of business scenarios [for propensity modeling]" (HG Insights, 2018).

Evaluation Metrics

False negatives are the worst kind of error we can make for this project. To better understand why, I have provided the possible scenarios below:

- **True Positive:** Send offer and user will likely use it.
- **False Positive:** Send offer but user doesn't want it or doesn't use it. ←Not as serious of an error.
- **True Negative:** Do not send offer and user doesn't want it or doesn't use it.
- **False Negative:** Do not send offer but user would have likely used it if we sent it. ←Worst error.

The two possible errors are False Positives and False Negatives. If we produce a False Positive, the user will likely just ignore our marketing effort and result in possibly some wasted effort on our part. In extreme cases, the user could view False Positives as harassment and be turned off by our brand. Because of this extreme case, False Positives are still important but not as important as False Negatives.

False Negatives result in a missed opportunity to market to a receptive customer. This can result in the business not making sales that they would have otherwise made.

Precision is used when the cost of False Positives is high. Recall is used when the cost of False Negatives is high. In our case we want to consider the cost of both but focus more on False Negatives. To do this we can use the F_2 score, which puts more emphasis on recall.

The F_2 score is defined as:

$$F_2 = (1 + 2^2) \cdot \frac{precision \cdot recall}{(2^2 \cdot precision) + recall}$$

Project Design

Below is my high-level workflow:

1) Create a conda environment on my local computer and install necessary packages.
2) Download the data from the Udacity Workspace to my computer.
3) Perform high-level exploration of the data.
    a. Graph distribution of variables to better understand the data.
        i. Examples include: Gender distribution, Age distribution, Income distribution, event distribution, etc.
    b. Graph relationships between data.
        i. Examples include: total transactions by age, total offers completed by age, total transactions by income, total offers completed by income, etc.
4) Clean the data. Below are the current items I see that need cleaned.
    a. Replace age 118 with NaN, since that is the default value.
    b. Other cleaning may occur as I explore the data and discover more.
5) Perform feature engineering to prepare for modeling. Below are the current features I plan to add:
    a. profile.json
        i. Convert "became_member_on" column to number of days they have been a member.
    b. portfolio.json
        i. Convert channel list into separate columns, with 1 indicating in the list and 0 otherwise.
    c. transcript.json
        i. Convert "value" into separate columns for "offer_id" and "amt_spent" depending on if value is an offer id or part of a transaction.
    d. Other features will be added as I explore the data and discover more.

6) Build our benchmark logistic regression model and use the evaluation metric on it.
    a. This will be the benchmark used when comparing with the other models.
7) Build other models using different supervised-learning algorithms (Support Vector Machines, Neural Networks, and possibly others).
8) Perform hyperparameter optimization using grid search for each model.
    a. This will use the validation set.
9) Run the model against the test set.
10) Use the evaluation metrics on each model and compare with our benchmark.
11) Compile results into a report and blog post.
12) Upload files to GitHub with README.md file.

References

Childs, Gary. "What is…a propensity model?". *Campaign Live*. 26 November 2002, https://www.campaignlive.co.uk/article/propensity-model/165289

HG Insights. "Propensity Modelling for Business". *Data Science Foundation.* 2 April 2018, https://datascience.foundation/sciencewhitepaper/propensity-modelling-for-business