Udacity Machine Learning Nanodegree

Capstone Project

Sanjeev Kumar

30th March 2021

Domain Background

In this project Starbucks Capstone problem is chosen to be researched and resolved. Problem at hand is to optimize the app offers with Starbucks. With the purpose of generating maximum revenue Starbucks has different offers for its regular customers. This can be divided into 2 sets, firstly, what offers to send whom. Secondly, which group is mostly likely to go for the offers. Ultimate aim is to increase the offer acceptance rate for the group of customers whom the offers has been sent to.

Problem statement

Find the group of customer and the corresponding offers for which the acceptance rate is highest.

Dataset and Input

The data is provided by Starbucks and 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:

--Edited to address review 1.

portfolio.json : **shape (10 rows x 6 columns)**

- 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 :**shape (17000 rows x 5 columns)**

- 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 **: shape (306534 rows x 4 columns)**

- 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

The datasets would be joined using the offer id and customer id attributes.

--Edited to address review 1.

Data which I am going to use is well Balanced. I am interested in finding

1) **Total completed offers: 33579**
2) **Total number of viewed offers: 57725**
3) **How many viewed the offer and completed it too: 29456**
4) **How many viewed the offer but didn't complete it: 28269**

I will be performing binary classification on this data and since the data points are quite balance the accuracy will be helpful in this case.

```
In [64]: viewed_offers_df[viewed_offers_df['offer_used_label'] == 0].count()
         #29456
         #28269

Out[64]: person              28269
         event               28269
         value               28269
         time                28269
         offer_used_label    28269
         dtype: int64
```

```
In [65]: viewed_offers_df[viewed_offers_df['offer_used_label'] == 1].count()

Out[65]: person              29456
         event               29456
         value               29456
         time                29456
         offer_used_label    29456
         dtype: int64
```

Solution Statement

Aim is to find the offers which have highest rate of acceptance. To achieve this, I will take each kind of offer (bogo, discount and advertisement for a drink) and make separate predictions and find which is best offer for a customer.

To achieve, this I plan to use sklearn's Support Vector Classification(SVC) and compare the results with PyTorch's neural network that performs Binary Classification.

Benchmark Model

Main model to be used will be a deep feedforward neural network having three or more hidden layers and a convolutional neural network to avoid overfitting through regularization. A benchmark model I could use will likely be a model outside the category of neural networks, relatively simple, and one that has been widely used in academia and industry in the past. Another good candidate may be a linear model such as a support vector machine with a linear kernel.

Evaluation Metrics

To evaluate the performance of the model I may look at accuracy, precision and recall.

-- Edited to address review 1.

Note: the data points that I will be using for the binary classification are quite balanced. Hence I think accuracy along with precision and recall will be helpful in this case

Data which I am going to use is well Balanced. I am interested in finding

1) **Total completed offers: 33579**
2) **Total number of viewed offers: 57725**
3) **How many viewed the offer and completed it too: 29456**
4) **How many viewed the offer but didn't complete it: 28269**

I will be performing binary classification on this data and since the data points are quite balance the accuracy will be helpful in this case.

```
In [64]: viewed_offers_df[viewed_offers_df['offer_used_label'] == 0].count()
         #29456
         #28269

Out[64]: person              28269
         event               28269
         value               28269
         time                28269
         offer_used_label    28269
         dtype: int64
```

```
In [65]: viewed_offers_df[viewed_offers_df['offer_used_label'] == 1].count()

Out[65]: person              29456
         event               29456
         value               29456
         time                29456
         offer_used_label    29456
         dtype: int64
```

The company may not be too concerned about accuracy and perhaps expects to experience a fairly large number of incorrect predictions. The initial intuition is that Starbucks would want to minimize lost sales – it is worse for the company to lose out on a potential sale by not providing an offer than to hand out an offer that is not used. Perhaps the revenues from a few sales more than make up for the costs of many offers given to customers. With this primary motivation, Starbucks may place more weight on recall, to reduce the number of false negatives, meaning if the model predicts the customer will not make a purchase after being shown an offer, the customer does not make a purchase. Another way to frame it is that the company would like to correctly identify as many of the people who would buy as possible and is willing to overshoot on identifying customers as potential buyers. So Starbucks may be willing to tolerate the model scoring lower on precision, meaning it would allow for the model to produce more false positives.

Project Design

Create a dataset with features and labels

- The features will be the demographic data and the properties of the Offers
- The labels will be the outcome of the customer making a purchase or not

Split the data into training, validation and test data

The relevant data are the customers who both received and viewed the data, so filter out the customers who did not receive or did not view the offers

- Select the machine learning model to train and make predictions

- Build the model, train using training data, validate and test the model
- Deploy the model to a web API endpoint so that a user can enter demographic data, receive an offer and indicate if he/she likes the offer
- To determine which offer to provide the customer, I may try to hold all test customers' features constant, vary the type of offer and make separate predictions to see which offer has the highest probability of encouraging the user to make a purchase.

-- Edited to address review 1.
--Below questions are raised during review 1

Q: For instance, what are all of the steps that you'll take during pre-processing?

a. **Enrich the transcript with information from profile and portfolio.**
b. **Find out the all the completed offers.**
c. **Find out all the viewed offers.**
d. **Find all the offers which are viewed and completed also which are not.**
e. **Use step 2 data to created train and test data.**
f. **Perform binary classification.**

Q: What approaches do you have in mind for dealing with missing values/outliers/NaN's?

a. **In my case, I still need to analyze the outliers. If found will need to remove them. NaN's will be replaced with 0 if found in number fields.**
Also, As I am targeting for higher recall, I expect the outliers in the data.

Q: Will you need to do any feature engineering?

a. **Yes, feature engineering will be required as explained above.**

Q: Will you need to scale or normalize the data?

a. **For now, I do not see a need to scale or normalize the data. If required I will do the same.**

Q: Can you talk a little bit more about the deep learning model you want to build?

a. **I want to create a binary classification feedforward models having hidden layers in between the input and the output layers. After every hidden layer, an activation function is applied to introduce non-linearity. When you have more than two hidden layers, the model is also called the deep/multilayer feedforward model or multilayer perceptron model (MLP).**

Q: What type of neural network would you want to use?

a. **Choice is between feed forward and convolutional NN (counters overfitting). But right now, the data set looks quite simple so convolutional NN may not be needed**.

Q: Do you have any architecture(s) in mind at this point?

a. **For now, I can say, 3 layers feed forward network with ReLU (non-lenearity) and DroupOut. Use Sigmoid to ouput as binary classification is the need.**
    - **3 input features**
    - **2 hidden layers**
    - **1 output dimention**

Q: How will you tune or refine your deep learning model?

a. **I will experiment with the number of layers that I use. And Epochs.**

-- Edited to address review 1.

References

1. https://www.deeplearningwizard.com/deep_learning/practical_pytorch/pytorch_feedforward_neuralnetwork/
2. https://medium.com/biaslyai/pytorch-introduction-to-neural-network-feedforward-neural-network-model-e7231cff47cb
3. https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28
4. https://www.quora.com/What-is-the-linear-model-in-machine-learning