Pradip Goban S S
4 April 2020

Machine Learning Engineer Nano degree Capstone Proposal

Starbucks Project

Domain Background

Starbucks Corporation is an American coffee company. Starbucks was founded in Seattle, Washington, in 1971. As of early 2019, the company operates over 30,000 locations worldwide.

Its quite interesting to see how different people respond to or how they don't respond to an offer, the factors that come into play in making that decisions, giving an insight into the complex working for the human mind, thinking about even the mundane and everyday tasks such as buying coffee.

Problem Statement

The problem can be formulated in to a binary classification problem, where there can be two possible outcome cases:

1. A person will respond to an offer encoded as (1)
2. A person will not respond to an offer encoded as (0)

The metrics that will be used in solving this problem is Accuracy. Based on this metric we will calculate how likely a person is going to respond to this offer making it as 0 or 1.

Datasets and Inputs

The Data is provided by Udacity and Starbucks as three files:

profile.json

Rewards program users (17000 users x 5 fields)

- gender: (categorical) M, F, O, or null
- age: (numeric) missing value encoded as 118
- id: (string/hash)
- became_member_on: (date) format YYYYMMDD
- income: (numeric)

portfolio.json
Offers sent during 30-day test period (10 offers x 6 fields)
- reward: (numeric) money awarded for the amount spent
- channels: (list) web, email, mobile, social
- difficulty: (numeric) money required to be spent to receive reward
- duration: (numeric) time for offer to be open, in days
- offer_type: (string) bogo, discount, informational
- id: (string/hash)

transcript.json
Event log (306648 events x 4 fields)
- person: (string/hash)
- event: (string) offer received, offer viewed, transaction, offer completed
- value: (dictionary) different values depending on event type
    - offer id: (string/hash) not associated with any "transaction"
    - amount: (numeric) money spent in "transaction"
    - reward: (numeric) money gained from "offer completed"
- time: (numeric) hours after start of test

Here we can see some data is in number form and some data is in characters. Since a Machine Learning Model doesn't take input in characters, we need to convert the field which are not numbers to numbers by using **one-hot-encoding** method. Since some of the values are high in number, it will be computationally expensive task. That's why need to reduce these values such as field ***income*** in **profile.json** using **PCA** method.

Solution Statement
The solution for this problem will be to combine the three files and make a single file or table which will have all the demographics of the person and offer details and the predicted output (i.e., a person will or not respond to the offer given).
By using this table we will pass the input to the model and get the output as 0 or 1.
We need to take **transcript.json** file or table first. Here by using the field ***person***, we can extract person details from **profile.json** file using ***id*** field.
By using the field ***value*** in **transcript.json** file which has a json object ***offer_id***, we can extract the details of the offer from **potfolio.json** file using ***id*** field.

Benchmark Model
Since it is a binary classification problem we need to use a Binary Classifier to solve this problem which will get output as 0 or 1. We can also try to use Deep Neural Networks to solve and check on which model it performs better by tuning the hyper-parameters.
We get the output of this model as the percentage of which a person is likely to respond to a given offer. Based on the percentage we can consider that if output is less than 50% then the person is likely to reject an offer.

Evaluation Metrics

Evaluation metrics for this model will be based on **Accuracy percentage**.
`Accuracy = No. of predicted correct / No. of actual correct`

Project Design

The implementation workflow for this project will include three steps.

1. Data Preparation
2. Model Training
3. Model Tuning
4. Model Prediction

1. Data Preparation :

- Building the Dataset : We have 3 datasets namely portfolio.json, profile.json and transcript.json. We need to combine these 3 datasets to make a single dataset using the common field in each dataset. For example we have field ***customer_id*** as common field between profile.json and transcript.json.
- Feature Exploration : Checking the co-relation between different fields in the table and understand the dataset. If any fields are more similar to each other then we can remove these redundant features and if require add more features from them.
- Preparing the Dataset : As the Machine Learning Model doesn't understand categorical data we need to use one-hot-encoding method for the fields with categorial data. For example the field ***gender*** in profile.json has categorial values.So we need to do one-hot_encoding for this field also we need to normalise the data using PCA so that a model can be trained faster.

2. Model Training :

As we determine if the person will respond to an offer or not, it will be binary classification problem.We can use Binary Classifier or Deep Neural Networks to train the model as this will be ideal for solving this type of problem. After the model has been trained we can check the **accuracy** of the model and decide whether to train the model again or to stop training.

3. Model Tuning :

This step will be done if the previous model is not trained enough or have less accuracy after model training step. Things involved when tuning the model is increasing/decreasing the epochs, hidden layers, learning rate etc and check for the accuracy. This step will be trail and error method as to guess what hyper parameters to change and check for the model improvement.

4. Model Prediction :

This will be the final step of the process. If the model is successfully trained will good amount of accuracy then we are now ready to predict the data.