Machine Learning Engineer Nanodegree

Capstone Proposal

Matteo Giuliani
October 10th, 2024

Domain Background

Starbucks is a renowned global brand specializing in coffee and other beverages. Starbucks offers a mobile application that enables registered users to order coffee for pickup, pay directly via the app in-store, and earn rewards points. Additionally, the app features promotional offers, which can range from simple advertisements about a product to actual deals like discounts or BOGO (buy one, get one free) offers. This project aims to customize these promotional offers for each user based on their historical responses to similar offers, identifying those most likely to engage.

Problem Statement

The primary goal is to identify the most suitable offers for each user, based on their reactions to previously received promotions. The challenge is that not all users receive the same offers, and this variability must be addressed using a dataset provided by Starbucks, covering customer interactions over a 30-day period. I will also develop a machine learning model to predict how customers will respond to these offers.

Datasets and Inputs

The dataset consists of simulated data that mirrors user behavior on the Starbucks rewards mobile app. Starbucks sends offers to users of the app every few days, and this data is captured across three JSON files:

- **portfolio.json**: Contains details of offers, including their IDs and metadata (e.g., type, duration).
- **profile.json**: Contains demographic data for each customer.
- **transcript.json**: Contains records of transactions, offers received, offers viewed, and offers completed.

Here is an overview of each file's schema and the variables they include:

portfolio.json

- **id** (string): Unique identifier for the offer.
- **offer_type** (string): Type of offer, such as BOGO, discount, or informational.
- **difficulty** (int): Minimum spending required to qualify for the offer.
- **reward** (int): Reward provided for completing the offer.
- **duration** (int): Duration of the offer's validity, measured in days.
- **channels** (list of strings): Channels through which the offer is delivered.

profile.json

- **age** (int): Age of the customer.

- **became_member_on** (int): Date when the customer joined the app.
- **gender** (str): Gender of the customer (with 'O' indicating non-binary/other options).
- **id** (str): Unique customer ID.
- **income** (float): Customer's income level.

transcript.json

- **event** (str): Description of the activity (e.g., transaction, offer received, offer viewed).
- **person** (str): Customer ID.
- **time** (int): Time elapsed in hours since the start of the test, beginning at t=0.
- **value** (dict): Depending on the record, this may include an offer ID or transaction amount.

The **portfolio.json** file identifies three primary types of offers Starbucks may distribute:

1. **BOGO (Buy-One-Get-One)**: Provides a second, identical product at no extra charge when a specific spending threshold is met.
2. **Informational**: Provides information about a product with no direct reward but encourages purchases if a spending threshold is met.
3. **Discount**: Offers a discount on a product, reducing its original price when certain conditions are met.

```
import pandas as pd
import matplotlib.pyplot as plt

portfolio = pd.read_json('../datasets/portfolio.json', lines=True)
transcript = pd.read_json('../datasets/transcript.json', lines=True)
profile = pd.read_json('../datasets/profile.json', lines=True)
```

```
portfolio.head()
```

	reward	channels	difficulty	duration	offer_type	id
0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd
1	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0
2	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed
3	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9
4	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7

```
portfolio.shape
```

```
(10, 6)
```

```
transcript.head()
```

	person	event	value	time
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}	0
1	a03223e636434f42ac4c3df47e8bac43	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0
2	e2127556f4f64592b11af22de27a7932	offer received	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}	0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}	0
4	68617ca6246f4fbc85e91a2a49552598	offer received	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}	0

```
transcript.shape
```

```
(306534, 4)
```

```
profile.head()
```

	gender	age	id	became_member_on	income
0	None	118	68be06ca386d4c31939f3a4f0e3dd783	20170212	NaN
1	F	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0
2	None	118	38fe809add3b4fcf9315a9694bb96ff5	20180712	NaN
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0
4	None	118	a03223e636434f42ac4c3df47e8bac43	20170804	NaN

```
profile.shape
```

```
(17000, 5)
```

Solution Statement

To determine which offers to send to customers, I will focus on identifying the offers that are most appealing to them. Through Exploratory Data Analysis (EDA), I will explore key aspects, including:

1. The most frequently accepted offers.
2. Customer responses to different offers.
3. Analysis of age and gender groups that show the highest interest in offers.

These analyses will be conducted in a local Jupyter notebook environment. For predicting how a customer might respond to an offer, I plan to use models such as RandomForestClassifier and DecisionTreeClassifier from sklearn, assessing which model best captures the patterns in the data. The Jupyter notebook environment will facilitate interactive analysis and model development, allowing for efficient visualization of results, parameter tuning, and performance evaluation.

Benchmark Model

For a quick and effective benchmark, I will use the KNeighborsClassifier. This method is commonly used for binary classification problems due to its simplicity and speed. The model's performance will be evaluated using the F1 score as the primary metric.

Evaluation Metrics

The F1 score will be used as the evaluation metric for this project, providing a balance between precision and recall. This metric is particularly useful for determining how well the model handles both false positives and false negatives. The F1 score ranges from 0 to 1, where a score of 1 represents perfect precision and recall, while 0 indicates the poorest performance.

Project Design

The overall approach for this project will follow these steps:

1. Set up the working environment using Jupyter.
2. Clean and preprocess the data to prepare it for modeling.
3. Perform detailed exploratory analysis to understand the data.
4. Build and compare various models to identify the best fit for the data.
5. Apply the benchmark model and evaluate results using the chosen metric.
6. Compile findings and insights into a comprehensive blog post.

References

- Liaw, A., & Wiener, M. (2002). "Classification and Regression by randomForest." R News, 2(3), 18-22. (https://journal.r-project.org/articles/RN-2002-022/RN-2002-022.pdf)
- Bhavani, K., & Rao, K. V. (2019). "An Efficient Recommendation System using Random Forest in Machine Learning." International Journal of Recent Technology and Engineering (IJRTE), 8(2S3), 343-348. (https://www.jetir.org/papers/JETIR1907B10.pdf)