Capstone Project | Nouf Alduhailan

Data science Nanodegree | 10 June, 2024

Starbucks Capstone Project

1- Project Definition

Project Overview

This project involves analyzing a dataset that simulates customer behavior on the Starbucks rewards mobile app. The data, created by a simulation program, shows how people make purchasing decisions and how these are influenced by different promotional offers. Each simulated customer has hidden traits that affect their buying habits, which are linked to visible traits. The dataset records events like receiving offers, opening offers, and making purchases, but does not track specific products—only the amounts spent or received through offers are noted.

There are three types of offers in the simulation: buy-one-get-one (BOGO), discount, and informational. BOGO offers require spending a certain amount to get an equivalent reward. Discount offers give a reward based on a fraction of the amount spent. Informational offers provide no rewards and have no spending requirements. These offers are delivered through various channels to mimic real-world marketing strategies.

Problem Statement

The primary objective is to analyze the dataset to build a predictive model that can determine whether an individual will respond to a given offer.

Datasets and Inputs

profile.json

This dataset contains information about the users enrolled in the Starbucks rewards program. It includes the following fields for 17,000 users:

- **gender**: (categorical) The gender of the user, which can be 'M' (Male), 'F' (Female), 'O' (Other), or null if not provided.
- **age**: (numeric) The age of the user. Missing values are encoded as 118.
- **id**: (string/hash) A unique identifier for each user.
- **became_member_on**: (date) The date when the user joined the rewards program, formatted as YYYYMMDD.

- income: (numeric) The annual income of the user.

portfolio.json

This dataset describes the promotional offers sent to users during a 30-day test period. It includes 10 different offers, each with the following fields:

- reward: (numeric) The amount of money awarded to the user for meeting the offer conditions.
- channels: (list) The platforms through which the offer was delivered, which can include web, email, mobile, and social.
- difficulty: (numeric) The amount of money the user needs to spend to qualify for the reward.
- duration: (numeric) The number of days the offer remains valid.
- offer_type: (string) The type of offer, which can be 'bogo' (buy-one-get-one), 'discount', or 'informational'.
- id: (string/hash) A unique identifier for each offer.

transcript.json

This dataset logs events related to user interactions with the offers. It contains 306,648 events, each with the following fields:

- person: (string/hash) The unique identifier of the user associated with the event.
- event: (string) The type of event, which can be 'offer received', 'offer viewed', 'transaction', or 'offer completed'.
- value: (dictionary) Additional details about the event. The specific keys and values depend on the event type:
    - For 'offer received', 'offer viewed', and 'offer completed' events, the value includes the offer id.
    - For 'transaction' events, the value includes the amount spent.
- time: (numeric) The time in hours since the start of the test period when the event occurred.

Evaluation Metric

Accuracy is a metric used to measure the proportion of correctly predicted instances among the total number of instances in a classification problem. It reflects the overall effectiveness of a model in classifying both positive and negative instances correctly.

Formula:

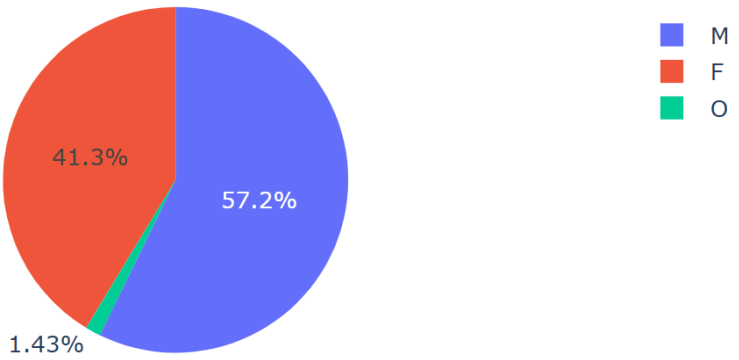
$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Instances}}$$

2- Exploratory Data Analysis

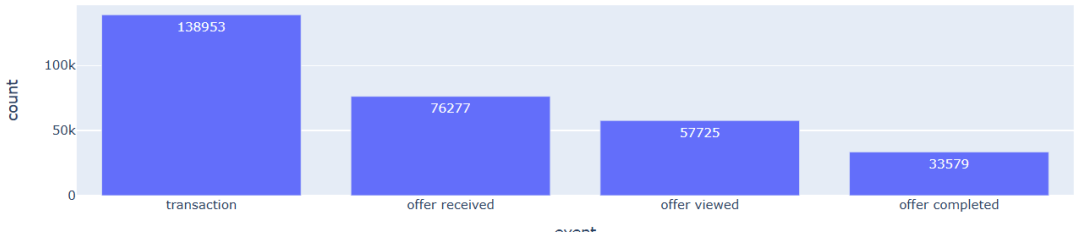
Data Exploration & Visualization

In this section, we will explore the datasets and analyze the customers of Starbucks.



The chart shows that 41.3% females, 57.2% males, and 1.43% others, with males being the predominant gender group (more than 50%).

It appears that individuals aged over 80 use the app less frequently. Hence, I classify individuals above this age as outliers.



This chart shows our target column, "event," which we aim to predict.

3- Data Cleaning

At this stage of our analysis, we dive into the essential task of data preprocessing. This meticulous process is all about refining our raw datasets so they're ready for building models.

1. Portfolio
   - Split the 'Channels' column into four distinct channels using one-hot encoding, assigning a 1 or 0 to each channel based on its presence
   - Apply one-hot encoding to the 'offer_type' column
   - Convert from boolean to binary
   - Rename the 'ID' column to 'offer_id'."

	reward	difficulty	duration	offer_id	email	mobile	social	web	offer_type_bogo	offer_type_discount	offer_type_informational
0	10	10	7	ae264e3637204a6fb9bb56bc8210ddfd	1	1	1	0	1	0	0
1	10	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	1	1	1	1	1	0	0
2	0	0	4	3f207df678b143eea3cee63160fa8bed	1	1	0	1	0	0	1
3	5	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	1	1	0	1	1	0	0
4	5	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	1	0	0	1	0	1	0

2. Profile
   - Rename the 'id' column to 'user_id' for clarity.
   - Drop rows where age is 118
   - Replace missing income values with the mean income.
   - Remove Other from 'gender' (occur less frequently)
   - Replace missing gender values with the mode (most frequent gender).
   - Map gender to binary values

	gender	age	user_id	became_member_on	income
1	0	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0
3	0	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0
5	1	68	e2127556f4f64592b11af22de27a7932	20180426	70000.0
8	1	65	389bc3fa690240e798340f5a15918d5c	20180209	53000.0
12	1	58	2eeac8d8feae4a8cad5a6af0499a211d	20171111	51000.0

3. Transcript
   - Renamed the 'person' column to 'user_id'.

- Replaced spaces in the 'event' column with underscores.
- Extracted 'offer_id' from the 'value' column.
- Extracted and rounded 'amount' from the 'value' column.
- Impute amount column with mean for missing values
- Dropped the original 'value' column from the transcript transcriptFrame.
- Encode event column

	user_id	event	time	offer_id	amount
0	78afa995795e4d85b5d9ceeca43f5fef	0	0	9b98b8c7a33c4b65b9aebfe6a799e6d9	12.777356
1	a03223e636434f42ac4c3df47e8bac43	0	0	0b1e1539f2cc45b7b9fa7c272da2e1d7	12.777356
2	e2127556f4f64592b11af22de27a7932	0	0	2906b810c7d4411798c6938adc9daaa5	12.777356
3	8ec6ce2a7e7949b1bf142def7d0e0586	0	0	fafdcd668e3743c1bb461111dcafc2a4	12.777356
4	68617ca6246f4fbc85e91a2a49552598	0	0	4d5c57ea9a6940dd891ad53e9dbe8da0	12.777356

4- Methodology

For this project, a methodology combining machine learning modeling and web application development was employed to predict user behavior based on input features. The methodology involved several key steps, including data preprocessing, model implementation, refinement, and evaluation.

Data Preprocessing

Data preprocessing involved cleaning and preparing the raw dataset obtained from multiple sources. Missing values were handled, outliers were treated, and features were encoded or transformed as necessary. Feature engineering techniques were applied to create new features and modify existing ones to improve model performance.

Implementation

The machine learning model, specifically a RandomForestClassifier, was implemented using the scikit-learn library in Python. Data was split into training and testing sets, and the model was trained on the training data. Additionally, a Flask web application was developed to integrate the trained model, allowing users to input features and receive predictions.

Refinement

Iterative refinement was performed to enhance the model's performance and the web application's usability. Hyperparameter tuning, feature selection, and other optimization techniques were applied to improve prediction accuracy and user experience.

5- Results

The outcomes of the model predictions were analyzed, including accuracy metrics and visualizations to interpret the results. Insights were provided into the effectiveness of the model in predicting user behavior based on the input features.

Model Evaluation and Validation:

The performance of the trained model was evaluated using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. The model's generalization ability was validated by assessing its performance on unseen data.

- Best Parameters: The hyperparameters of the RandomForestClassifier model were tuned using grid search, resulting in the following best parameters:
    - bootstrap: True
    - max_depth: None
    - min_samples_leaf: 4
    - min_samples_split: 10
    - n_estimators: 300
- Improved Model Evaluation Metrics:
    - Accuracy: 0.6700
    - Precision: 0.6443
    - Recall: 0.6700
    - F1 Score: 0.6489

Justification

The selection of the RandomForestClassifier model and the chosen preprocessing techniques were justified based on their suitability for the project objectives. The approach was aligned with the problem statement and aimed to address the project objectives effectively.

6- Conclusion

In conclusion, the project demonstrated the successful implementation of a machine learning model and a web application for predicting user behavior. Key findings and insights were summarized, highlighting the effectiveness of the implemented solution in addressing the project problem.

Reflection

This project successfully combined machine learning and web development to predict user behavior. Challenges included fine-tuning model parameters and managing data complexity, but collaboration and problem-solving led to a robust solution.

Improvement

- Enhanced Features: Explore more feature engineering techniques.

- Ensemble Models: Investigate combining models for better predictions.
- User Interface: Improve user experience with better visuals and feedback.
- Scalability: Optimize deployment for handling larger datasets.