UDACITY MACHINE-LEARNING NANO_DEGREE

CAPSTONE PROJECT : STARBUCKS DATA

Yassine.H
March-2020

Project Overview

Introduction

This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. Once every few days, Starbucks sends out an offer to users of the mobile app. An offer can be merely an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users might not receive any offer during certain weeks.

Not all users receive the same offer, and that is the challenge to solve with this data set.

The data 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

Challenges :

Combine transaction, demographic and offer data to determine which demographic groups respond best to which offer type. This data set is a simplified version of the real Starbucks app because the underlying simulator only has one product whereas Starbucks actually sells dozens of products.

Every offer has a validity period before the offer expires. As an example, a BOGO offer might be valid for only 5 days. You'll see in the data set that informational offers have a validity period even though these ads are merely providing information about a product; for example, if an informational offer has 7 days of validity, you can assume the customer is feeling the influence of the offer for 7 days after receiving the advertisement.

transactional data showing user purchases made on the app including the timestamp of purchase and the amount of money spent on a purchase. This transactional data also has a record for each offer that a user receives as well as a record for when a user actually views the offer. There are also records for when a user completes an offer.

Installation

For running this project, the most important library is Python version of Anaconda Distribution. It installs all necessary packages for analysis and building models:

{math; json; tqdm; pandas; numpy; seaborn; sklearn}

Introducing a Dataset

This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. Once every few days, Starbucks sends out an offer to users of the mobile app. An offer can be merely an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users might not receive any offer during certain weeks.

Not all users receive the same offer, and that is the challenge to solve with this data set.

Project Motivation

I chose this project to understand the success rate of offers being sent and analysis is done through addressing the following questions.

1. **How many customers were provided with a specific offer?**

--> the feature importance given by all 3 models were that the tenure of a member is the biggest predictor of the effectiveness of an offer. Further study would be able to indicate what average tenure days would result in an effective BOGO offer.

2. **What's the performance level of an offer?**

--> My strategy for solving this problem has mainly two steps. First, I combined offer portfolio, customer profile, and transaction data. Second, I assessed the accuracy and F1-score of a naive model that assumes all offers were successful. Third, I compared the performance of logistic regression and random forest models. This analysis suggests that a random forest model has the best training data accuracy and F1-score. Analysis suggests that random forest model has a training data accuracy of 0.762 and an F1-score of 0.753. The test data set accuracy of 0.740 and F1-score of 0.730 suggests that the random forest model I constructed did not over-fit the training data.

Data Preparation

There are three datasets provided and each dataset is cleaned and preprocessed for further analysis. The target features for analysis are offer_success, percent_success.

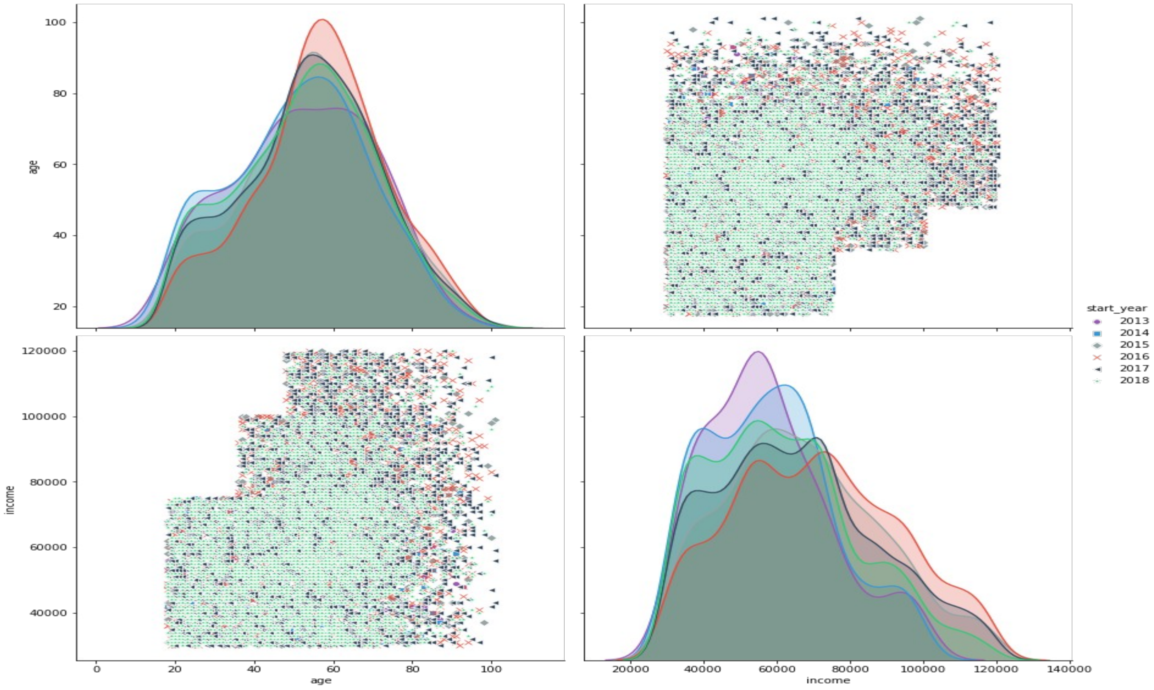
A. Portfolio - renaming id column name to offer_id, one-hot encoding of channels and offer_type columns

B. Profile - profile: renaming id column name to customer_id, replacing age value 118 to nan, creating readable date format in became_member_on column, dropping rows with no gender, income, age data, converting gender values to numeric 0s and 1s, adding start year and start month columns (for further analysis)

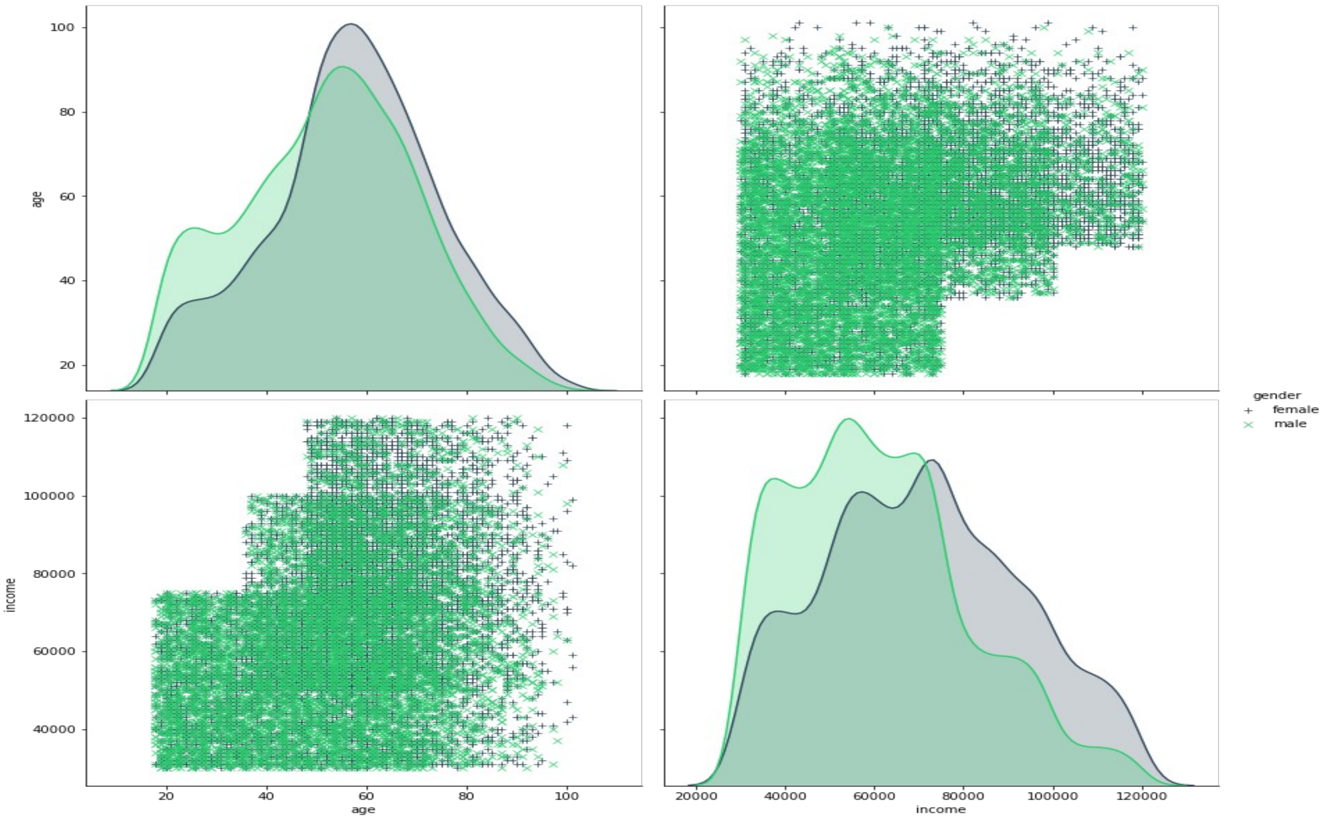
C. Transcript - renaming person column name to customer_id, creating separate columns for amount and offer_id from value col, dropping transaction rows whose customer_id is not in profile:customer_id, converting time in hours to time in days, segregating offer and transaction data, finally dropping duplicates if any

Data Visualization

1- Customers Global Data Pattern Visualization [Age, Income,] per years: Age distribution plot depicts that the median age of a customer is 60 and most of our customers belong to age range between 40 to 70. Income distribution plot shows that the number of customers whose average salary is less than 70K is high than the other side considering 70K to be median of the income distribution. Membership distribution has interesting results - 2017 has the highest registered customers than any starting from 2013. The plot also shows that there is an increasing trend in the number of registrations except for 2017:



2- Customer Income distribution per gender & ages: plots conclude that minimum and maximum income for both male and female are approximately same but the count of male customers in low-income level is slightly higher than that of female customers. concidering all highest income are from age od 40 above:



Machine Leaning Implementation :

Few things to consider while constructing the models - all features were converted to numerical to fit and train above models. Bias and variance are two characteristics of a machine learning model. Bias refers to inherent model assumptions regarding the decision boundary between different classes. On the other hand, variance refers a model's sensitivity to changes in its inputs. These can influence our results sometimes so models have to be tested throughly against bias and variance. Also, while splitting train and test datasets and tuning parameters to fit a model, we will have to make sure that data doesn't overfit the model:

X. Evaluate Naive Predictor Performance:

```
Naive predictor accuracy: 0.471
Naive predictor f1-score: 0.640
```

A. Logistic Regression Model:

```
LogisticRegression model accuracy: 0.693
LogisticRegression model f1-score: 0.805
```

Results

- Results suggest that a logistic regression model's accuracy and f1-score is better than the naive predictor
- Accuracy
    - Naive predictor: 0.471
    - Logistic regression: 0.693
- F1-score
    - Naive predictor: 0.640
    - Logistic regression: 0.805

B. Random Forest Classifier Model:

```
RandomForestClassifier model accuracy: 0.761
RandomForestClassifier model f1-score: 0.751
```

Results

- Results suggest that a random forest model's accuracy and f1-score is better than the naive predictor
- Accuracy
    - Naive predictor: 0.471
    - Random forest: 0.761
- F1-score
    - Naive predictor: 0.640
    - Random forest: 0.751

Results

However, the performance of a random forest model can be still improved by analyzing features which impacts an offer's success rate as a function of offer difficulty, duration, and reward. These additional features should provide a random forest classifier with the opportunity to construct a better decision boundary that separates successful and unsuccessful customer offers.

Also, initially it seemed like we had a lot of data to work, but once NaN values and duplicate columns were dropped and the data were combined into one single data-set, it felt as though the models might have benefited from more data. With more data, the classification models may have been able to produce better accuracy and F1-score results.

Additionally, better predictions may have been deducted if there were more customer metrics. For this analysis, I feel we had limited information about customer available to us—just age, gender, and income. To find optimal customer demographics, it would be nice to have a few more features of a customer. These additional features may aid in providing better classification model results.

File Descriptions

There is a notebook available here to showcase work related to the above questions and wrangling process.

There are 3 data file please Note all json files should be in a folder named [data] before using the Jupyter notebook. [unzip the data file]

A. portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)

B. profile.json - demographic data for each customer

C. transcript.json - records for transactions, offers received, offers viewed, and offers completed

The main observations of the code are published https://github.com/yassineelhallaoui/Machine-Learning-Engineer-Nanodegree-Program-Capstone_Starbucks.