Machine Learning Capstone

Rajashekar Chintalapati
Jul 5, 2020

Domain Background

To increase sales and customer satisfaction, companies often give promotions (like BOGO - Buy one Get one, rebate & reward programs) to customers, so that customer come back more often. By identifying customer purchase patterns, companies more precisely can target the users who purchase less frequently. By doing analysis like given an offer, we can find out user conversion rate. In the project, I plan to do such type of analysis using the Starbucks data which contains simulated data of customer transactions collected using Starbuck rewards mobile app. The data is a simplified version of real Starbucks app.

Problem Statement

The problem I choose to solve given an offer and demographic data, whether user will complete the offer or not. By using given data, we can find the buyer pattern, what factors are impacting the sales. This data was provided by Udacity and is a simplified version of real Starbucks app.

Datasets and Inputs

There are 3 datasets provided.

1. **Portfolio Dataset -** which contains offer ids and meta data about each offer (duration, type, etc.)
    - 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)

2. **Profile Dataset -** which contains demographic data for each customer
    - 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
3. **Transcript Dataset** - which contains records for transactions, offers received, offers viewed, and offers completed.
    - 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

Solution Statement

By going through data, first analyze data, process and prepare the data for training like removing NaN records and removing any unnecessary features, do one-hot encoding where ever necessary, once the data is ready with all required features, identify which feature is the output in this case it would be offer completed. And then split the data into train and test, and use supervised machine learning algorithms like Naive Bayes classifer, sklearn ensemble methods like AdaBoost, Bagging, Random Forest algorithms to train the data and will use evaluation metrics like accuracy, precision, recall and f1 score to find out each model score.

Benchmark Model

Will use Logistic regression which is extremely efficient mechanism for calculating probabilities especially incases like binary classifications.

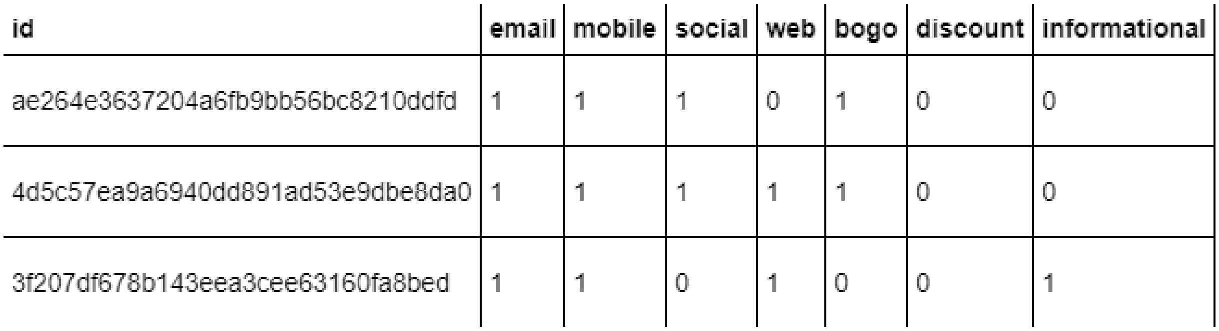
Evaluation Metrics

Will use accuracy, precision, recall and f1 score to find out each model score. And will use feature importance using Random Forest model to define which features were important to decide whether will user will accept the offer or not.

Project Design

From the given data, I see that portfolio data has promotion details, profile data has customer details and transcript dataset has transactional details. First step would be preparing data.

Portfolio data mainly has 3 types of offers - bogo, discount and informational, since these are categorical, will use one-hot encoding so these will become some think like below. And same needs to be on channels



id	email	mobile	social	web	bogo	discount	informational
ae264e3637204a6fb9bb56bc8210ddfd	1	1	1	0	1	0	0
4d5c57ea9a6940dd891ad53e9dbe8da0	1	1	1	1	1	0	0
3f207df678b143eea3cee63160fa8bed	1	1	0	1	0	0	1

Profile data has some NaN values on income, gender & age. So will drop all NaN values. **became_member_on** has the date, will convert this to number of days to find the tenure, how long user was with Starbucks. **Gender** has Female, Male & Others, will convert that to 0, 1, 2 respectively.

Transcript data has event which has 4 types, offer received, offer reviewed, offer completed and transaction. I see that informational offers does not have offer completed so will remove all informational offers since that will not help in solving whether user will complete the offer or not. Records with event type transaction has amount which was spent for that transaction and time which user has spent in this test, using these values will find how many transactions user made, how much amount spent till now and how much time user was on this test. Since transcript data has many records with of same person, first will find how many offers was received and out of them how many of them was viewed and out of them how many of them was completed. Basically I will group by all the records in transcript, so final transcript data would be for each person how many offers was received, how many are viewed, how many are completed, how many emails, mobile, social, web channels was used, how many bogo or discount was offered, what was the total duration, difficulty and rewards earned for all offers.

Final step would be merging all above dataset into one and **offer completed** would be the output which has the value whether user has finally completed the offer or not.

This data will be split into 70% training & 30% testing and will Naive Bayes and sklearn ensemble methods like AdaBoost, Bagging & Random Forest to train the data.

Use Logistic regression to find the benchmark model score. Using test data get the predictions of the model and find accuracy, precision, recall and F1 scores. And get the list of feature importance using Random Forest model to find out which features has more weight.

References

https://scikit-learn.org/stable/modules/ensemble.html

https://scikit-learn.org/stable/modules/naive_bayes.html

https://developers.google.com/machine-learning/crash-course/logistic-regression/calculating-a-probability

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html