Machine Learning Engineer Nanodegree

Capstone Project- Starbucks app data

Devesh Katiyar
Feb, 2020

I. Definition

Project Overview

Machine learning has become an increasingly important part of IT today and usage of ML is increasing daily in almost every domain. However, the power of ML is not utilised to its fullest yet and can be useful for society and market in unimaginable ways. ML can also be useful for various companies in improving their products, sales to offer better customer service and earning a higher growth rate.
STARBUCKS is one of the flagship Worldwide companies which has been established since Mar-1971 and worldwide coffeehouse chain, and has a tremendous database of users. Starbucks offers their free app to make orders online, predict the waiting time and offer better service. That's why analysis on app usage is more crucial to leverage the business and understand the customer's behavior.

Problem Statement

We want to find a way to give each customer of STARBUCKS the right in-app special offer. We have different kinds of offers: *Buy One Get One* (BOGO), classic *Discount* or *Informational* on a product.
We will analyze historical data which is made available to us by the company in the json format. We will develop a model on the basis of this dataset and train our algorithm so that we can make a model which can predict the type of offer that if made to a customer will turn into a sale.

Metrics

Based on the past data, we will compare the models prediction with the help of evaluation metrics

$$\text{Accuracy} = \frac{\text{True Positives + True Negatives}}{\text{All Samples}}$$

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F - Score(Balanced) = 2.\frac{Precision\ .\ Recall}{Precision + Recall}$$

II. Analysis

Data Exploration

For this project we have 3 available data sources.

Portfolio

Containing offer ids and meta data about each offer.

- 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)

Profile

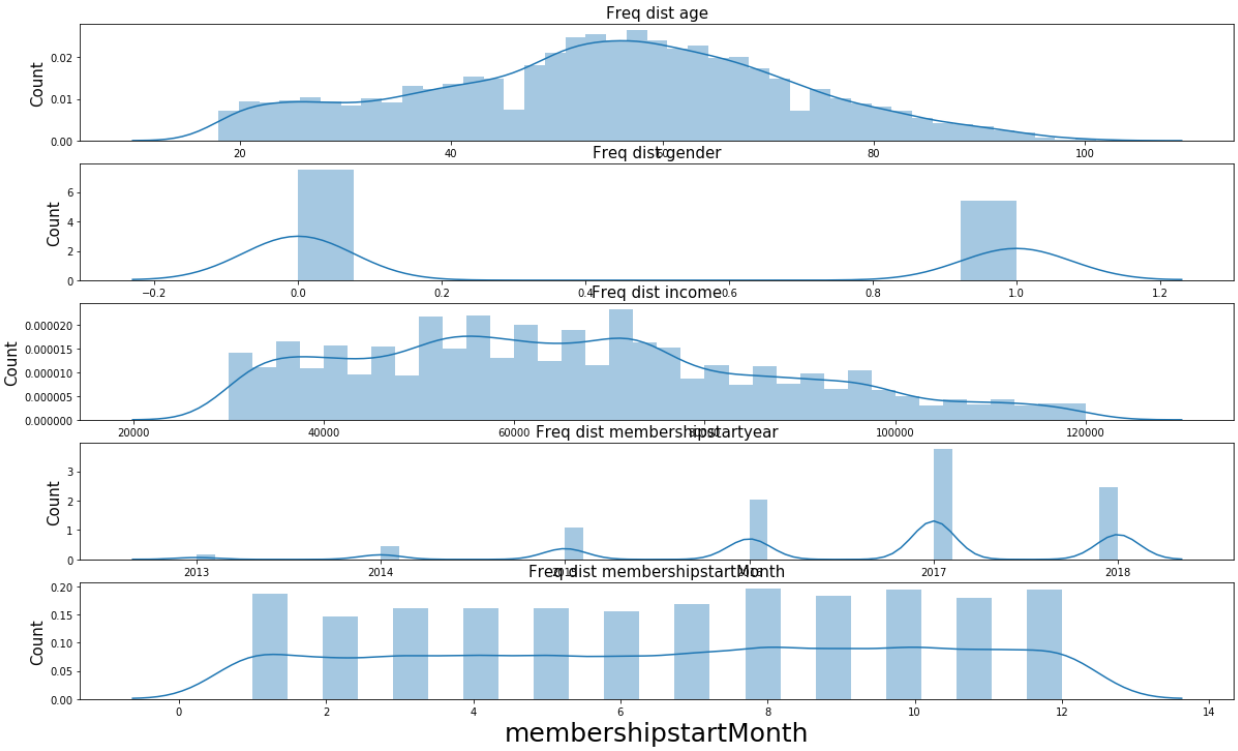
Demographic data for each customer.

- 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

Transcript

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 the start of the test. The data begins at time t=0
- value - (dict of strings) - either an offer id or transaction amount depending on the record.

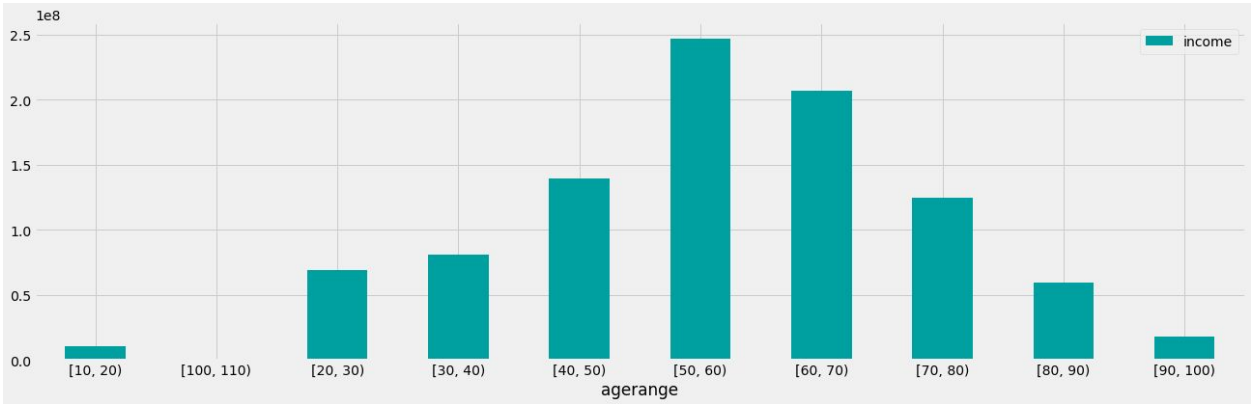


In the above image we have shown the relation of dataset with respect to different features available. We have compared it with features like age, gender, income, membership start month and membership start year. This graph will help us in gaining the insight of the data for solving the problem.

Exploratory Visualization

We divided the income column into two parts separate for male and female so that we can we explore better insights of the dataset and here is what we have found.

In the next graph we are trying to find out the relation between age and income.



Algorithms and Techniques

For the better understanding of the problem we tried three different algorithms for the dataset and made the made the final model with the algorithm giving the best result for our problem.

Decision Tree Classifier

A Decision Tree Classifier is a simple representation for classifying examples. It is a Supervised Machine learning algorithm where the data is continuously split according to certain parameters.

RandomForest Classifier

It is an Ensemble learning method for classification and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean of the individual trees.

Adaboost Classifier

It is a classifier that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

Benchmark

As benchmark we have decided to train our model on all the three classification algorithms but use only the best algorithm out of it to make our final model.

While training and looking at the results we find out that Decision Tree Classifier was giving a accuracy of more than 99 percent and also the F1 score of 99.0 but we didn’t used in our final model as it might have the risk of overfitting the dataset instead we chose the RandomForestClassifier with a accuracy of 92 percent and F1 score 0f 93.

III. Methodology

Data Preprocessing

First we arranged all the columns in an order and modified column names where required so that it will be easy to preprocess.

Categorical Encoding

We know that Machine Learning can only use numerical data, so we must transform the categorical feature into numerical feature. One Hot Encoding can be helpful at this place.

Gender	Gender-M	Gender-F	Gender-O
M	1	0	0
F	0	1	0
O	0	0	1

Dropping Features

We need to drop less important features which are not playing any crucial role in the model. This will help us in two ways, First it will reduce the size of dataset, Second it will reduce the probability of the model to fit over noise which will eventually improve the accuracy of the model.

Data Normalization

Some algorithms are sensitive to unbalanced distribution of the data. For this reason we need to normalize the data so that every feature contributes equally to the model. We used the MinMaxScaler here in our model.

Model Development

We have used RandomForest Classifier, Adaboost Classifier, DecisionTree Classifier for making our model. We have divided the dataset into three different parts

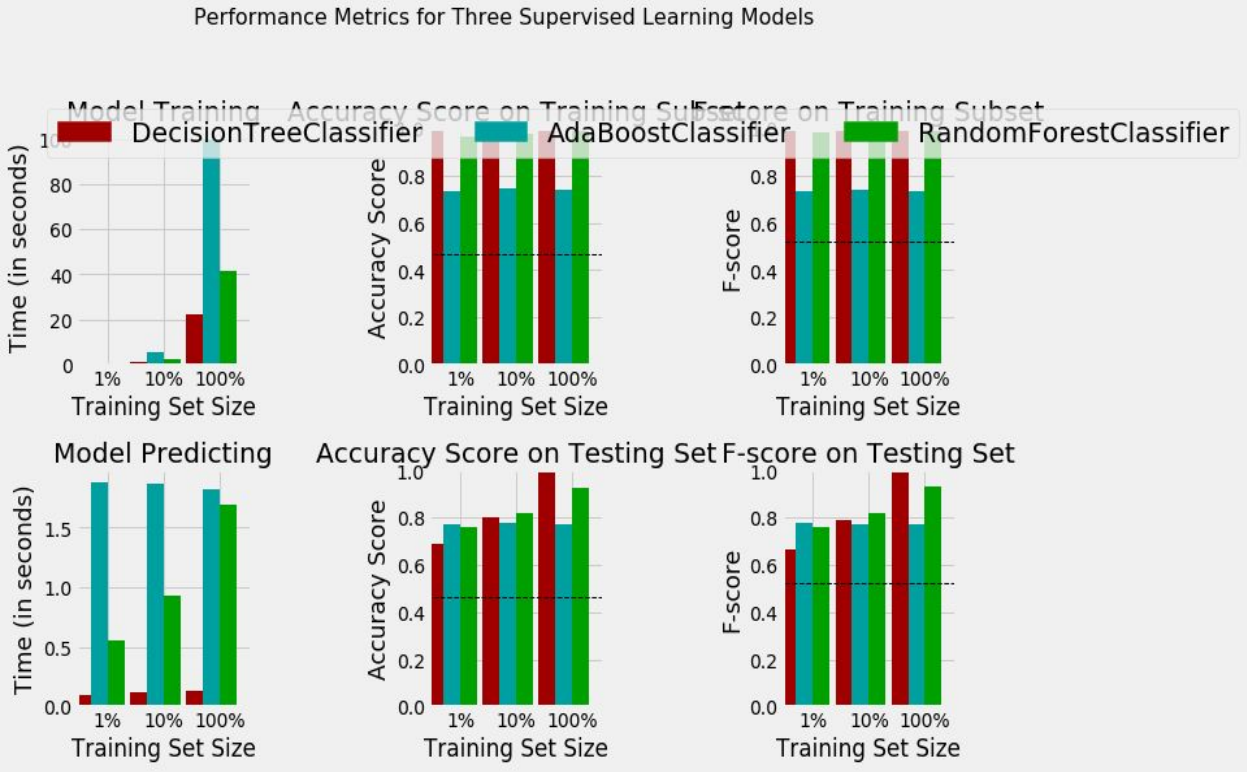
1 percent of the dataset.

10 percent of the dataset.

100 percent of the dataset.

Where we checked the performance of all three algorithms on different samples which we had created. We checked the accuracy score and F1 score for all three samples to choose the best possible algorithm for our model.

We found that Decision Tree Classifier was the best algorithm for us on the training set on the basis of the accuracy score and f1 score but we didn't chose it for our model as it may be very sensitive to noise or may overfit the data instead of Decision Tree we choose RandomForest Classifier for our model as this algorithm is very less prone to noise and works better with all kind of problem.



Hyperparameter Tuning

This is a very crucial part of model making as every Machine Learning algorithm has a set of hyperparameters which need to be set before training the model so that the model can perform to its best. Here in our model we chose RandomForestClassifier as the final algorithm on the basis of our previous model evaluation.

IV. Results

Model Evaluation and Validation

For DecisionTree Classifier

	Train Accuracy	Test Accuracy
1 percent data	1.0	0.6885
10 percent data	1.0	0.8042
100 percent data	1.0	0.9912

	Train f1-score	Test f1-score
1 percent data	1.0	0.6653
10 percent data	1.0	0.7898
100 percent data	1.0	0.9904

For Adaboost Classifier

	Train Accuracy	Test Accuracy
1 percent data	0.7333	0.7714
10 percent data	0.7433	0.7750
100 percent data	0.7399	0.7738

	Train f1-score	Test f1-score
1 percent data	0.7330	0.7760
10 percent data	0.7396	0.7732
100 percent data	0.7366	0.7722

For RandomForest Classifier

	Train Accuracy	Test Accuracy
1 percent data	0.9666	0.7596
10 percent data	0.9766	0.8162
100 percent data	1.0	0.9281

	Train f1-score	Test f1-score
1 percent data	0.9852	0.7620
10 percent data	0.9816	0.8192
100 percent data	1.0	0.9302

V. Conclusion

Let’s summarize all the different steps followed in this process.

- We analyze the data from 3 different datasets containing information about offers in Starbucks app.
- Then we remove the improper data from the dataset.
- Then we create new features with one hot encoding to understand the dataset.
- We develop Machine Learning model with three different algorithms on different sizes of dataset.

  Data cleaning is the most important part of the machine learning model. I personally did several iterations to make the data fit for modelling.

  I tried to implement a goodș model for solving the problem however no model could be a best model in the domain of machine learning there is always a option of improvement remains in any model.