Machine learning Engineer Nanodegree

Capstone Proposal

Dheeraj P

Feb 03, 2020

Domain background

Starbucks Corporation is an American coffee company and coffeehouse chain. In reality, the Starbucks app sends out various types of promotional offers to customers, either discounts (BOGO or 50% off during happy hours) or Star Dash challenges (completing required purchases to earn star rewards). Sometimes it also informs customers about limited-time drinks. Starbucks sends out three types of offers (BOGO, discount and informational) via multiple channels. Customers' responses to offers and transactions are recorded. In either setting, it is important to send the right offer to the right customer.

Problem statement

I chose to build a model that predicts whether or not a customer will complete an offer, meaning that the customer will receive a offer, view the offer and finish the offer before expire day. Thus, to solve this problem, I will establish a binary classifier.

Datasets and inputs

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

The data is provided by Starbucks and Udacity.

Solution statement

Recall the problem, "if a customer will response and complete a given offer?". In dataset, we have features profiling customers and other features about offer descriptions. But we didn't include any information from transaction because the transaction may hint the answer. The label is 1 (complete) or 0 (incomplete). Thus, this is a binary classification problem.
Some common methods are decision tree, random forests, support vector machines(svm) and neural networks(nn). In this project, we will use boosting random forests (XGBoost) model as benchmark model and use neural networks(pytorch) as solution model.
ROC-AUC is used as metric for model evaluations. It computes area under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores. Higher is better.
One potential solution has 3 steps:
1. Preprocess combined transaction, demographic and offer data. This dataset describes an offer’s attributes, the user's demographic data and whether the offer was successful.
Also, split datasets into train, valuation and test datasets.
2. Training XGBoost Model as baseline.
3. Training PyTorch deep learning model and improve the results.

Benchmark model

We use XGBoost model as benchmark. Hyperparameters are
max_depth=5,
eta=0.2,
gamma=4,
min_child_weight=6,
subsample=0.8,
objective='binary:logistic',
The ROC-AUC score is 0.71594. This result will be used as the threshold to determine the improvement of solution model of which the ROC-AUC score is higher.

Evaluation metrics

ROC-AUC is used as evaluation metric. It computes area under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores. This will quantify the performance of both the benchmark model and the solution model. The reason is that the datasets are balanced and also, we only care about the final class predictions and we don't want to tune threshold.

Project design

The project will follow this workflow for approaching a solution:

1. Data Processing1. Data Cleaning

- Profile: figure out how to deal with null values for age, gender and income. Simple way may discard these values. But if they make up a big portion of data, I may use another category to represent them.
- Portfolio: split channels into its own columns.
- Transcript: make an offer completion column or dataset based on customer's behaviors(events).

2. Data Transformation: Join these tables into a data set. The first column is whether the offer is completed, and the rest are features from profile and portfolio.

2. Feature Engineering

1. Determine features by exploratory analyze features to choose a small number of uncorrelated features. If features are too many, I may consider reducing dimensionality by performing a PCA.

2. Prepare final datasets: split dataset into train, valuation and test datasets.

3. Train XGBoost Model on SageMaker

4. Train Deep Learning Model on SageMaker

5. Hyperparameter Tuning Deep Learning Model on SageMaker

6. Conclusion

References

1. Starbucks Wiki
2. Udacity Starbucks Project Overview
3. Accuracy
4. F1 score
5. Starbuck's Capstone Challenge