capstone_proposal.md

Machine Learning Engineer Nanodegree

Capstone Proposal

Adeshola Afolabi
October 1st, 2020

Proposal (The Arvato Project)

Domain Background

The idea of acquiring new customers via digital channels such as mobile, web applications, emails, etc has been thriving and has become an important communication mode for creating brand longetivity in marketing and advertising. Companies have evolved from using domain knowledge to drive campaigns to using data informed decisions. This challenge employs a well analyzed and effective means of acquiring new users using machine learning techniques (both supervised and unsupervised). Worldwide e-commerce statistics indicate that retail e-commerce sales reached US$ 3.53 trillion in 2019 and e-commerce revenue is projected to reach US$ 6.54 trillion in 2022 indicating the number of digital buyers is on the rise and has become a common practice.

I see an opportunity in this regard, and building effective solutions to help brands acquire a chunk of this revenue is what motivates me for this project.

Problem Statement

The ultimate problem to be solved in this project is:

- Identify how the mail order company can acquire new users/clients efficiently. Upon identifying these users, a marketing campaign is subsequently sent to them.

After identifying the right users, the following problems will be solved:

- Customer segments will be created from the general populace based on the analysis of attributes of already established customers
- The output of the customer segmentation will be fed into a machine learning algorithm (supervised) which will predict whether or not an individual will respond to a marketing campaign.

Datasets and Inputs

The datasets used for this project is provided on the Udacity workspace and it is divided into four namely:

- Udacity_AZDIAS_052018.csv: Demographics data for the general population of Germany; 891 211 persons (rows) x 366 features (columns).
- Udacity_CUSTOMERS_052018.csv: Demographics data for customers of a mail-order company; 191 652 persons (rows) x 369 features (columns).
- Udacity_MAILOUT_052018_TRAIN.csv: Demographics data for individuals who were targets of a marketing campaign; 42 982 persons (rows) x 367 (columns).
- Udacity_MAILOUT_052018_TEST.csv: Demographics data for individuals who were targets of a marketing campaign; 42 833 persons (rows) x 366 (columns).

Each datapoint represents a user and the corresponding attributes such as age, gender, location, employment classification, social status, range of transactions across different channels, etc. The already established customers(Udacity_CUSTOMERS_052018.csv) of the mail order company will be used to identify lookalikes from the general population(Udacity_AZDIAS_052018.csv) of people in Germany and this analysis will be used to make predictions on which user ranks best for a marketing campaign. This will be done with the train and test csv files provided.

Solution Statement

The approach to solving this problem is divided into two:

- The unsupervised learning approach
- The supervised learning approach

Unsupervised Learning

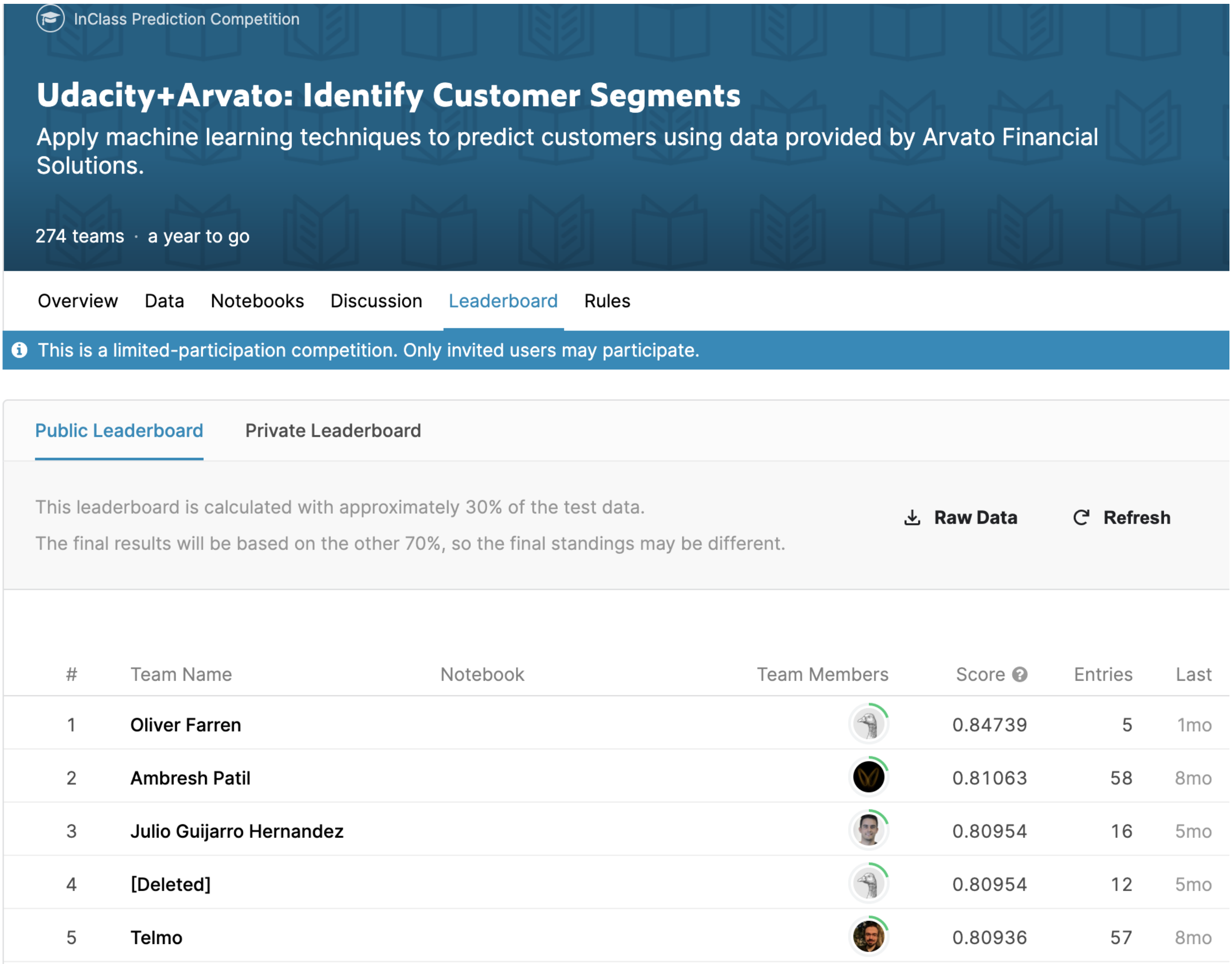
This approach will be used to identify segments within the dataset. To begin with, a principal component analysis (PCA) will be done; this helps with dimensionality reduction and also, identifying attributes that are most important in identifying the segments. The KMeans algorithm will use the output of the PCA to identify the segments within the dataset by clustering similar attributes together. The number of clusters (k) will be determined using the elbow method and each of the cluster centroid determines how far a datapoint(user) is to the cluster.

Supervised Learning

Upon clustering the dataset into different segments, and identifying the most important attributes using PCA, a supervised algorithm can be trained. Using the train dataset provided which has the extra field 'RESPONSE', we try a host of supervised machine learning algorithms or deep learning algorithms and determine which will perform best on the test dataset. A validation dataset will also be created from the train dataset to give an idea of the model performance. The ROC-AUC, accuracy and f1 score will be the evaluation metric to look out for.

Benchmark Model

A way to benchmark the resulting model will be to use the top ranked models on Kaggle to compare the performance of my model. Right now, the team on the leaderboard has a score of 84.7% (using the AUC-ROC final score).



Udacity+Arvato: Identify Customer Segments

Apply machine learning techniques to predict customers using data provided by Arvato Financial Solutions.

274 teams · a year to go

This is a limited-participation competition. Only invited users may participate.

This leaderboard is calculated with approximately 30% of the test data.

The final results will be based on the other 70%, so the final standings may be different.

#	Team Name	Notebook	Team Members	Score	Entries	Last
1	Oliver Farren			0.84739	5	1mo
2	Ambresh Patil			0.81063	58	8mo
3	Julio Guijarro Hernandez			0.80954	16	5mo
4	[Deleted]			0.80954	12	5mo
5	Telmo			0.80936	57	8mo

Evaluation Metrics

The evaluation metric for this project is AUC for the ROC curve, relative to the detection of customers from the mail campaign. A ROC, or receiver operating characteristic, is a graphic used to plot the true positive rate (TPR, proportion of actual customers that are labeled as so) against the false positive rate (FPR, proportion of non-customers labeled as customers).

The line plotted on these axes depicts the performance of the algorithm as we sweep across the entire output value range. We start by accepting no individuals as customers (thus giving a 0.0 TPR and FPR) then gradually increase the threshold for accepting customers until all individuals are accepted (thus giving a 1.0 TPR and FPR). The AUC, or area under the curve, summarizes the performance of the model. If a model does not discriminate between classes at all, its curve should be approximately a diagonal line from (0, 0) to (1, 1), earning a score of 0.5. A model that identifies most of the customers first, before starting to make errors, will see its curve start with a steep upward slope towards the upper-left corner before making a shallow slope towards the upper-right. The maximum score possible is 1.0, if all customers are perfectly captured by the model first.

Project Design

The project is broken down into 5 sections:

- Exploratory data analysis
- Data cleaning and visualization
- Modeling (Unsupervised learning)
- Modeling (Supervised learning)
- Model evaluation and tuning

Exploratory Data Analysis (EDA)

This step is an entry point into what the world of solving this particular problem looks like. An in-depth analysis of the demographic(e.g location, age, education etc) and behavioral data(e.g interest, transactions, response to call to actions, etc) is done. A sense of data distribution is also gotten from this step. An example is to identify the gender distribution in the data analysis(X% of users are male, Y% are female and Z% are unknown). The findings from this step will help in the data cleaning and visualization phase (e.g how do I handle the unknown or missing values in the gender field?).

Data cleaning and visualization

A module will be used here. Several methods will be defined within the module to help with the cleaning and visualization. Some of the methods are:

- An automatic numeric/categorical/hash attributes detector (which will help seperate the data into numerical, categorical, high cardinal attributes)
- An outlier detector (for the numeric attributes)
- A NaN cleaning method (use of mean/median/mode for the numerical attributes, custom filler/mode for the categorical attributes)
- Data visualizer

Modeling (Unsupervised Learning)

This section is sub-divided into two:

- A Principal Component Analysis (PCA)
- Clustering (KMeans)

The Principal Component Analysis

PCA attempts to reduce the number of features within a dataset while retaining the “principal components”, which are defined as weighted, linear combinations of existing features that are designed to be linearly independent and account for the largest possible variability in the data! You can think of this method as taking many features and combining similar or redundant features together to form a new, smaller feature set.

K-Means Clustering

The feedback from the PCA attributes is used to build a k-means model. K-means is a clustering algorithm that identifies clusters of similar data points based on their component makeup. Since we would have reduced the feature space to 'n' PCA components, we can then cluster on the PCA transformed dataset.

One method for choosing a "good" k is to choose based on empirical data. A bad k would be one so high that only one or two very close data points are near it, and another bad k would be one so low that data points are really far away from the centers.

We want to select a k such that data points in a single cluster are close together but that there are enough clusters to effectively separate the data. You can approximate this separation by measuring how close your data points are to each cluster center; the average centroid distance between cluster points and a centroid. After trying several values for k, the centroid distance typically reaches some "elbow"; it stops decreasing at a sharp rate and this indicates a good value of k.

Modeling (Supervised Learning)

To train a supervised learning algorithm, two things are paramount;

1. The target variable ('RESPONSE' in this case)
2. The features to be used for training

Upon clustering the dataset into different segments, and identifying the most important attributes using PCA, a supervised algorithm can be trained. Using the train dataset provided which has the extra field 'RESPONSE', we try a host of supervised machine learning algorithms or deep learning algorithms to determine which will perform best on the test dataset. A train/test split is done on the "Udacity_MAILOUT_052018_TRAIN.csv" dataset to have a validation dataset. The performance of these models are monitored and the best model is deployed.

The final model is uploaded on Kaggle to see where it ranks with the leaderboard.

Model Evaluation and tuning

Several evaluation metrics such as accuracy, f1, etc will be tried with the ultimate metric being the AUC for the ROC curve. If scores are not satisfactory enough, hyperparameter tuning will be done to optimise for a better performance.