Machine Learning Engineer Nanodegree
Capstone Proposal
Hector Garcia
October 10, 2021

Create a Customer Segmentation Report for Arvato Financial Solutions

Domain Background

Customer Acquisition is the process of bringing new customers to a business, a key ingredient of business growth. Traditionally, customer acquisition focuses on identifying high-performing channels, such as radio, TV, social media, etc. where potential customers may be reached. Considerations of "who to target" are mixed with "what message or offer would resonate more". And success is often measured using cost-based metrics such as cost per acquisition, cost per click, and the like. But the goal remains the same: to develop an efficient acquisition strategy that targets true potential customers, rather than marketing to the general population.

Machine Learning (ML) provides a unique approach to the problem of targeted marketing in that instead of relying on instinctive heuristics that make marketing an 'art', it leverages on insights gleaned from a mathematical analysis of the customer's data, making marketing a 'science'. Machine Learning can help traditional marketing strategies in at least two ways: First, it can reveal latent or nuanced characteristics of the population that may make them ideal for targeted marketing campaigns, regardless of the channels used to reach them. Second, it can be used to refine heuristic strategies by quantifying their assumptions and measuring their validity; hence, making such customer acquisition strategies more efficient.

Problem Statement

Can the customer acquisition strategy of a Mail-Order company be made more efficient using Machine Learning? Using demographic data of the general population and the customers, as well as results of a recent mailout campaign, can we use Machine Learning techniques to let the data speak directly and tell us what constitutes a likely customer? We want to conceive a targeted advertising campaign based on recommendations from the data itself, as exposed through Machine Learning techniques in order to achieve a higher hit-ratio in marketing.

A trivial solution would be to put the mailout campaign data through a binary classifier such as a *Linear Learner*, hoping that the mailout sample data is representative of the general population; A more sensible approach, free of assumptions about the data, would take into consideration the entirety of the data available. That is the aim of this project. First, we'll sift through the general population and customer datasets using unsupervised ML tools to learn who is a likely customer. Then, we'll use these same tools on the mailout training dataset, together with a binary classifier (supervised learning) to validate and refine our model by measuring its predictive power. Finally, we'll submit our model predictions on the test data to a *Kaggle* competition for final assessment.

Datasets and Inputs

The givens in this problem are three datasets in four files:

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

We'll use the first two files to establish how customers are similar to or different from the general population at large. Then we'll use our analysis on the other two files to predict which recipients are most likely to become customers. The "TRAIN" data will be used to validate and refine our analysis, and the "TEST" data to compete in Kaggle.

The "CUSTOMERS" file contains three extra columns (*'CUSTOMER_GROUP'*, *'ONLINE_PURCHASE'*, and *'PRODUCT_GROUP'*), with information about the customers. The original "MAILOUT" file, before splitting into "TRAIN" and "TEST" included one additional column, *'RESPONSE'*, to indicate if a recipient became a customer. For the "TRAIN" subset, this column has been retained, but in the "TEST" subset it has been removed; it is against this withheld column that our final predictions will be assessed in the Kaggle competition. The other columns are the same between the files.

Additionally, two Excel spreadsheets: `./DIAS Information Levels - Attributes 2017.xlsx`, and `./DIAS Attributes - Values 2017.xlsx` provide a list of attributes and descriptions, as well as a detailed mapping of data values for each feature in alphabetical order.

Solution Statement

Any solution would require a *Customer Segmentation Report*, and a *Supervised Learning Model.* Before doing any analysis, however, the data will be examined, cleaned, visualized to identify promising features, and wrangled enough to make it amenable to analysis with ML models.

The Customer Segmentation part will be preceded by a reduction in dimensionality using *Principal Component Analysis* (PCA). The actual segmentation will be carried out with a *K-means Clustering* algorithm. These three steps — namely, data wrangling, PCA and K-Means will be implemented on the general population and on the customer datasets. These steps should facilitate a meaningful comparison between both datasets, enabling us to make educated guesses as to how the customers could be spotted within the general population. After this, it should be a straightforward task to apply the same steps to the "MAILOUT_TRAIN" dataset, and then to train a binary classification model on the result, using the provided *'RESPONSE'* labels.

Benchmark Model

As a benchmark model, the combination of data wrangling, PCA, K-Means and a Linear Learner binary classifier will be used. This benchmark should suffice as a preliminary simplistic solution that trains fast and can provide a baseline upon which other solutions could be gauged. Other solutions being variations of the supervised learning piece, e.g., XGBoost, PyTorch MLPs, etc.

Evaluation Metrics

In spite of the fact that most of the work in this project will occur during the data wrangling and customer segmentation parts, it is the results of the binary classifier that are judged in the end. Without looking at the data, it is difficult to speak intelligently about appropriate metrics. However, we can venture to make two simplifying assumptions: (1) the mailout data is imbalanced, i.e., it is likely to contain less customers than non-customers, and (2) false positives don't hurt the business as much as false negatives.

The first assumption takes *Accuracy* out of the list of useful metrics; with an overwhelming amount of non-customers in the data, a model that mostly cries "non-customer" would be highly accurate, yet of little use. This leaves us with *Precision*, and *Recall* as potential candidates to evaluate our binary classifier. The second assumption crystalizes the superiority of using Recall over Precision. Yes this is a business-driven decision, but it is a justifiable choice nonetheless. Marketing to a non-customer may mean marginal costs to the company and a minor annoyance to the individual. A missed customer, on the other hand, is bad for business not only because of the lost potential revenue, but also because of the detrimental effect on the quality of the data collection, by giving less relevance to that customer's segment in future runs. For these reasons, it is legitimate to use Recall as a sensible metric with which to judge the goodness of fit of our binary classifier models, both the benchmark, and the solution.

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Project Design

For the Customer Segmentation part, two unsupervised learning algorithms will be used, both on the general population, and on the customer files. First, *Principal Component Analysis* (PCA) will be employed to reduce the dimensionality across features. Then, *K-means Clustering* will be performed to assign each person to a cluster based on centroid distances. These two algorithms will turn a ***891,211 persons x 366 features*** problem into a more tractable ***M x N*** abstraction consisting of *M* clusters and *N* principal components.

The "principal components" extracted from the PCA algorithm are linear combinations of the linearly independent features that account for the largest amounts of data dispersion. The resulting eigenvectors and eigenvalues will guide our decision to establish a tolerable level of retained variance in the data. Essentially, with PCA, we will be able to get rid of correlation redundancy in the features, thus trading off dimensionality (number of features), vs variance (information in the data); to the extent that two features are strongly correlated, one of them makes the other redundant and thus fairly superfluous.

K-Means Clustering will perform the actual segmentation. We'll have some leeway in deciding on an optimal number of clusters, guided by the average centroid distance. Here is another tradeoff, this time between treating all rows as single data points, or the entire dataset as one big cluster. How the clusters are arranged in PC space, tells us which persons are similar and what feature traits define that similarity.

After reducing the dimensionality within acceptable loss of variance, and after segmenting to an optimal number of clusters, we should be able to describe parts of the general population that are more or less likely to become customers. Armed with this information, we'll build a prediction model using the `"MAILOUT"` data files after putting them through the same data wrangling, dimensionality reduction and segmentation procedures. Ideally, we should be able to use the demographic information from each individual to decide whether or not to include that person in the campaign. For this Supervised Learning part, an XGBoost model will be used as a classifier on the `"MAILOUT_TRAIN"` and later on the `"MAILOUT_TEST"` pieces. XGBoost is a widely used algorithm that has been tested for production on large-scale problems that include, but are not limited to, classification.

It is envisioned that throughout these procedures, several opportunities will arise where decisions in the guise of compromises or trade offs will have to be made affecting the performance of the eventual solution. How to clean or complete missing data, which features to drop, how to scale, how many principal components, how many clusters, and other tunable parameters of the different models. These are opportunities to tune our solution that will be revisited with the aim of improving performance using the appropriate metrics dictated by the data as well as the models.