Machine Learning Engineer Nanodegree

Capstone Proposal
Customer Segmentation Report for Arvato Financial Solutions
Youssof Yassin
January 21st, 2021

Domain Background

Arvato is one of the many divisions that make up Bertelmann, a media, services and education group. It is headquartered in Gütersloh, Germany and deals in customer reports, information technology, logistics and finance. Since Arvato is an international services company and has been around for many years, it has acquired a significant amount of data, which it means to make use of as much as possible. With the expanding release of data science and machine learning tools nowadays, it has become the norm for every major company to start using data-related techniques to target customers, instead of just pure intuition.

Problem Statement

Arvato wants first to analyze the attributes of existing clients and match them against a bigger dataset full of people in Germany. This way, we can sort out the customers that have similar behaviour and statistics into specific groups, making it easier to analyse and later perform predictions.

Secondly, we need to find out which people in Germany are most likely to be new customers willing to buy organic products through a specific mail order. So, instead of reaching out to the entire population of Germany, we can just target those specific people that would be interested.

Datasets and Inputs

The datasets are provided primarily by Arvato company to Udacity students for this specific type of capstone project. Datasets used would be:

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).

Demographics data, that was for individuals who were targets of a marketing campaign, was split into 2 datasets:

Udacity_MAILOUT_052018_TRAIN.csv: Contains the data we will be training our algorithms on; 42 982 persons (rows) x 367 (columns).

Udacity_MAILOUT_052018_TEST.csv: Contains the data we will be trying to predict; 42 833 persons (rows) x 366 (columns).

The **CUSTOMERS** dataset will be used to compare with the general population **AZDIAS**, in analyse and study the data and find interesting relations. This should help us make predictions using the later **MAILOUT** files. Each row contains information about a single person, including their household, building and neighbourhood.

In the case of the two **MAILOUT** files, the **TRAIN** file contains a column called “RESPONSE”, which indicates whether that specific person is a good target for our mail order or not. The **TEST,** however, has that column removed and we would need to send to the following kaggle link in order to determine the final accuracy of the model:

https://www.kaggle.com/c/udacity-arvato-identify-customers/data

There are also two more files, outside the four that were originally mentioned, that help explain the columns available in the previous files:

DIAS Information Levels - Attributes 2017.xlsx: is an overview of the attributes and descriptions, sorted by the informational category.
DIAS Attributes - Values 2017.xlsx: provides a bit more detailed explanation of each feature’s data values.

Solution Statement

First, we will spend a bit of time getting to know the data, finding any interesting patterns and visualizations that would help us gain some simple domain knowledge on the subject. We also need to find ways to deal with any missing data, by either removing columns completely, or find ways to fill them in (mean, mode, etc). We might need to use dimensionality reduction techniques, such as principal component analysis (PCA), if the columns still prove to be too many. The main goal of this step is to make sure the data is thoroughly cleaned and ready enough so as to not to confuse any machine learning algorithms we would use in the future.

Next, we will use the classic clustering analysis algorithm in order to group up related or similar customers into a specific number of groups. This number will be determined by visualizing different cluster numbers and finding the optimum number using the ‘elbow’ method.

Finally, we will use the Train data already provided to train our machine learning algorithms. We might need to further split up the train into a train and validation set, to help later optimize our algorithms. The optimization would be done using a means such as Grid Search to tune the many different hyperparameters, possibly using the full train data, as grid search does its own data split into validation.

Finally, after the model is ready and trained, we will use it to make predictions on the Test data provided and send the result to kaggle in order to determine how well our model performed.

Benchmark Model

We will be using **AWS Sagemaker** for its ease of use, speed, as well as good practice. Multiple models will be used:

PCA: To reduce the dimensionality of the many columns we have.
KMEANS: For clustering our customer data into groups.
Random Forest Classifier: As a base model to make predictions.
ANN through PyTorch: Neural Networks are better designed to deal with large amounts of data. So, it might be possible to produce better results using this.
GridSearch: This will be used to find the optimum hyperparameters to use.

Evaluation Metrics

For PCA, we will calculate the correlations and choose the columns that are least correlated with each other.

For clustering, we will plot the different numbers of clusters onto a graph and find the 'elbow' point that tells us the appropriate number of clusters to use.

For machine learning (Random Forest and ANN), we will use AUC; **A**rea **U**nder the **R**eceiver operating characteristic curve. This is the metric that the Kaggle competition will be using to evaluate our model over the test dataset.

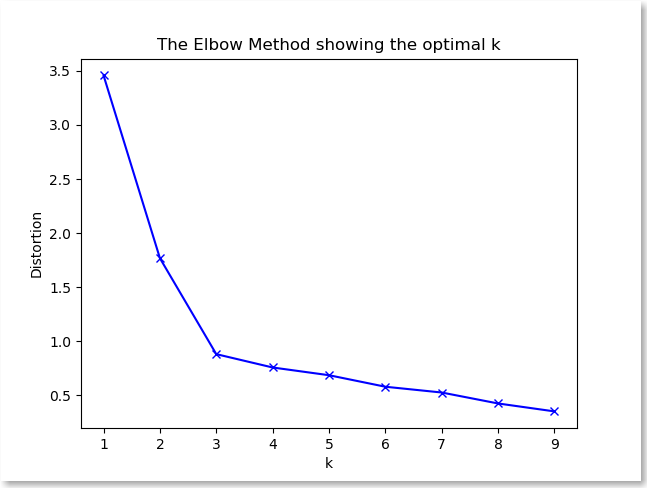
It would also be useful to try to average out the Precision and Recall values (so that we don't get, for example, all precision and zero recall).

Project Design

To Summarise all that was previously mentioned, first, we will collect all six of our datasets and put them onto Amazon Sagemaker notebook and eventually store them in our S3 bucket. We will perform some quick initial visualizations and exploration of the data.

Then, we will perform some basic cleaning, removing empty values, etc. We'll use PCA in order to reduce the number of columns we are working with, choosing the columns that are least correlated with each other.

Next, we will solve the first problem, by clustering the customer data into groups using KMeans algorithm. We will find the appropriate number of using the 'elbow' method as shown below (using the number 3 as the elbow in that example):



Further on, we will build our Random Forest Classifier model in order to get a base number for our predictions. After that, we'll move on to our Pytorch model and start experimenting with the number of layers and other hyperparameters to find a model that would beat our Random Forest Classifier's predictions.

Finally, we'll use GridSearch to go through a lot of different hyperparameters to create a final optimum model to be used to predict the test data. The final predictions will be sent to Kaggle to find out how well we did compared to the rest of the leaderboard there. Typically, we are aiming for a score of 0.8.