Machine Learning Engineer Nanodegree

Capstone Report

Customer Segmentation – Arvato Financial Solutions

Rubén Cruz García

22nd November 2020

1. Definition

Project overview

Arvato Financial Solutions is a German company that offers financial services, Information Technology services and Supply Chain Management solutions for business customers on a global scale. Their services also include the case we will tackle here: client profiles and acquisition.

More specifically, the business matter we will be focusing on here is the following:
a client mail-order company seeks our help/services to acquire new clients more efficiently. To accomplish our project, we will employ customer segmentation: this refers to the practice of dividing a customer base into groups of individuals, depending on well-defined specific features, such as age, gender, family, interests or spending habits.

Dividing the current customer base of the company into smaller meaningful groups will enable us to gain insight into their different types of customers, which will then permit the company to target the German population at large in an informed way: for example, marketing teams would highly benefit from such grouping information, as they would be able to determine which promotional campaign would most appeal to which demographic group before even launching these campaigns. Adding to customer segmentation, we wish to develop a supervised ML model capable to predict whether a person will be a new customer.

Problem statement

The main problem could be formulated as: "Given the demographic data of a person, how can a mail order company acquire new customers in an efficient way".

Machine Learning techniques can be employed in the two main sections of the project:

- Unsupervised learning methods on the data of established customers and the German general population's demographics data, we can create customer segments.

- Supervised learning methods on a third dataset, we can train a model to classify
If a person will become a new customer, and use this model for future predictions.

Datasets and inputs

The project makes use of four datasets:

- **Udacity_AZDIAS_052018.csv**: Demographics data for the general population of Germany; 891 211 people (rows) x 366 features (columns).
- **Udacity_CUSTOMERS_052018.csv**: Demographics data for customers of a mail order company; 191 652 people (rows) x 369 features (columns).
- **Udacity_MAILOUT_052018_TRAIN.csv**: Demographics data for individuals who were targets of a marketing campaign; 42 982 people (rows) x 367 (columns). This is the only dataset with the target variable, which is highly unbalanced (people hiring the services are the minority class).
- **Udacity_MAILOUT_052018_TEST.csv**: Demographics data for individuals who were targets of a marketing campaign; 42 833 people (rows) x 366 (columns).

Evaluation metrics

In the unsupervised learning part, in order to choose the ideal number of clusters for the K-means algorithm, we will use the "elbow method": we select the value of k at the "elbow" i.e. the point after which the distortion/inertia start decreasing in a linear fashion.

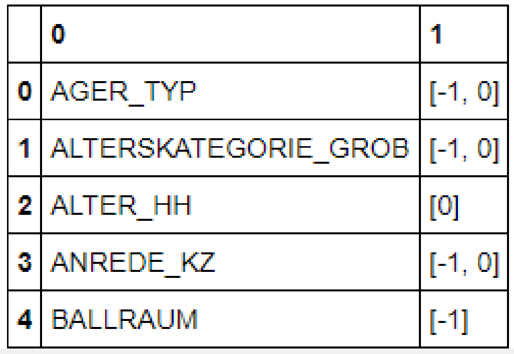
Regarding the supervised learning models, where we predict the outcome of a given customer, we will use the Area Under Curve (AUC) metric, since the target labels are highly imbalanced and the accuracy would be a bad choice to evaluate the model. Also, the AUC is the metric chosen to evaluate the final predictions in the Kaggle submissions section.

2. Analysis

Data exploration

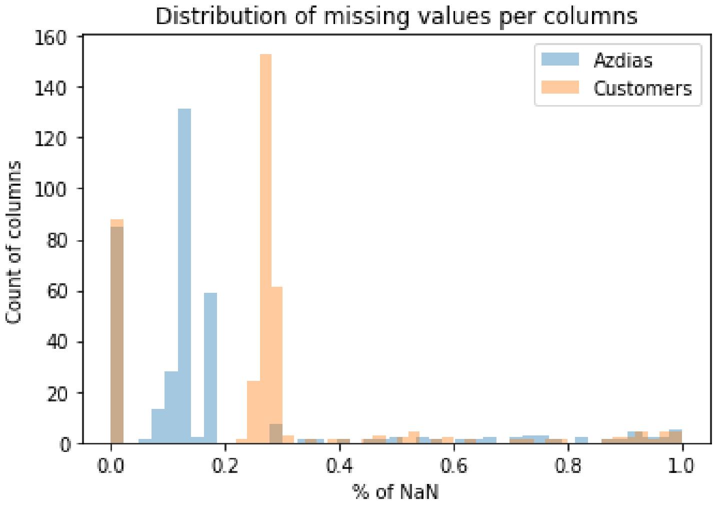
We first explore the AZDIAS and CUSTOMERS datasets, which enables us to gain insight into the data (features/columns present, value ranges, missing values…). Each row of both demographics datasets refers to one single person. The MAILOUT train and test datasets are similar, but the train includes a column with the target to predict (RESPONSE).

The first step is to check the attribute information dataset (DIAS Attributes - Values 2017.xlsx), in order to identify values that should actually be encoded as missing values. For example:



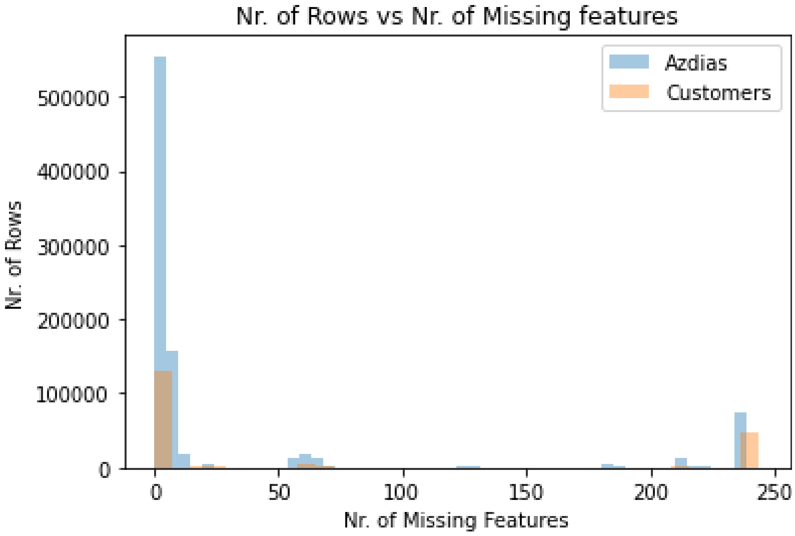
	0	1
0	AGER_TYP	[-1, 0]
1	ALTERSKATEGORIE_GROB	[-1, 0]
2	ALTER_HH	[0]
3	ANREDE_KZ	[-1, 0]
4	BALLRAUM	[-1]

Due to the high amount of missing values in all datasets, we have to decide whether removing or not certain columns for which it does not make sense to impute the values:

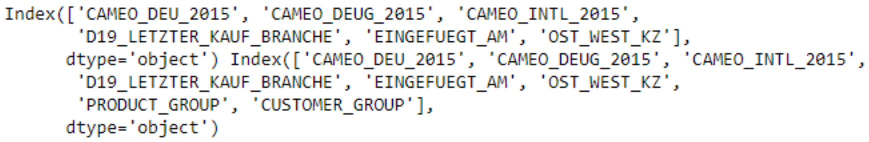


We decide drop the columns where more than 30% of rows are missing. We take as reference the customers' dataset (which has a larger percentage of missing values in its columns). Note that the same columns will be removed for the MAILOUT datasets.

Now we focus on the remaining columns. We decide to remove the rows for where there are more than 50 features missing. This step will not be applied to MAILOUT test, because the submission for Kaggle has to contain the same number of rows as the original dataset.



There are some columns with data type as object:



```
Index(['CAMEO_DEU_2015', 'CAMEO_DEUG_2015', 'CAMEO_INTL_2015',
       'D19_LETZTER_KAUF_BRANCHE', 'EINGEFUEGT_AM', 'OST_WEST_KZ'],
      dtype='object') Index(['CAMEO_DEU_2015', 'CAMEO_DEUG_2015', 'CAMEO_INTL_2015',
       'D19_LETZTER_KAUF_BRANCHE', 'EINGEFUEGT_AM', 'OST_WEST_KZ',
       'PRODUCT_GROUP', 'CUSTOMER_GROUP'],
      dtype='object')
```

We explore these features, replacing values which are not properly encoded and removing some of them. After this, we impute the missing values with the most frequent value, which makes the most sense since we are dealing with population data.

Finally, we normalize the data using “MinMaxScaler()”, which is a necessary step for continuing with the Principal Component Analysis.

3. Results: Algorithms and techniques

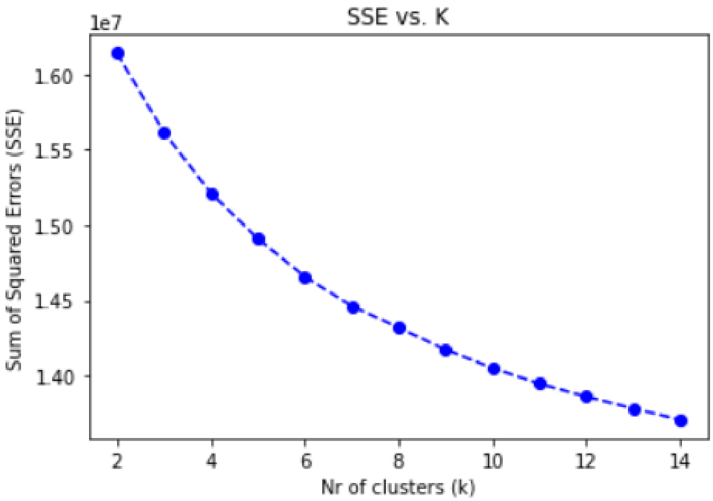
Customer segmentation

The aim of this first part is to divide the general population and the customers into different segments, in order to compare the general population and customers to determine future customers.

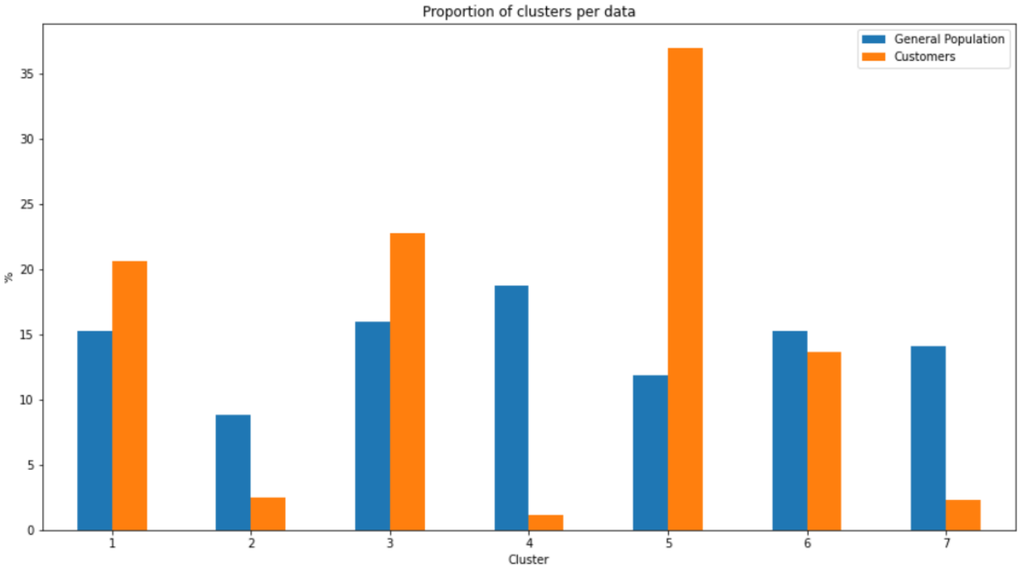
Before employing clustering with K-means, it is best practice to undergo dimensionality reducing, for which we used Principal Component Analysis (PCA). We decided to retain the number of Principal Components that represent at least 90% of variance of the dataset.

Once PCA is done on *azdias* and *customers* and both DataFrames have reduced dimensions (with more meaningful features), we will use K-means clustering to actually create the clusters for customer segmentation, based on demographics data.

In order to choose the ideal number of clusters for the K-means algorithm, we will use the “elbow method”: we select the value of k at the “elbow” i.e. the point after which the distortion/inertia start decreasing in a linear fashion.



The general population and the customer population have been clustered into segments. The next figure represents the proportions of population coming into each cluster.

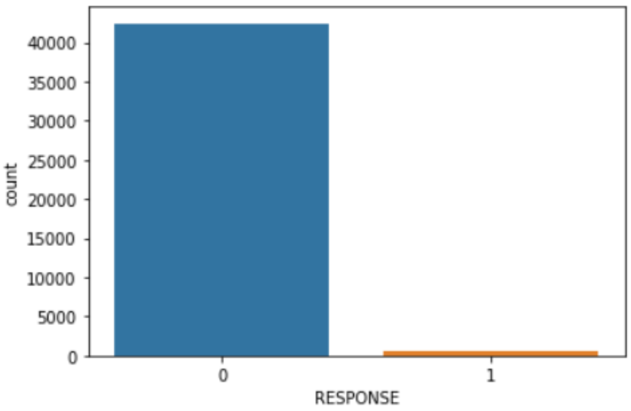


We also produced again the original values (undoing the scaling and the PCA), to check what characterizes the clusters where customers are in a larger proportion (i.e. 1, 3 and 5).

Customer acquisition

The second part of the project is to use supervised learning algorithms to predict whether a person will be a customer or not based on the demographic data. The file 'Udacity_MAILOUT_052018_TRAIN.csv' is provided with the same features as the general population and customers demographic data. An extra column 'RESPONSE' has been provided with this data. The response column indicates whether this person was a customer or not. This data has been cleaned by following similar cleaning and processing steps that were followed for general population and customer data.

First thing we see is how imbalanced the sample is:

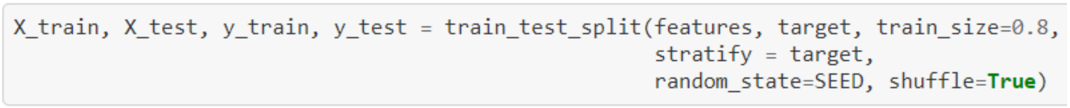


Benchmark model

The benchmark model will be a Logistic Regression algorithm since it is easy to train and test in a short time. This model gives us an AUC of **0.75**.

Train-test split

The final model will be chosen according to how it performs on a validation dataset, so we divide the training MAILOUT dataset into train and test, with a proportion of 0.8-0.2, respectively. We additionally shuffle and stratify the data, to ensure that the order does not lead to a different result and that the proportion of converted customers is the same in the train and test samples:



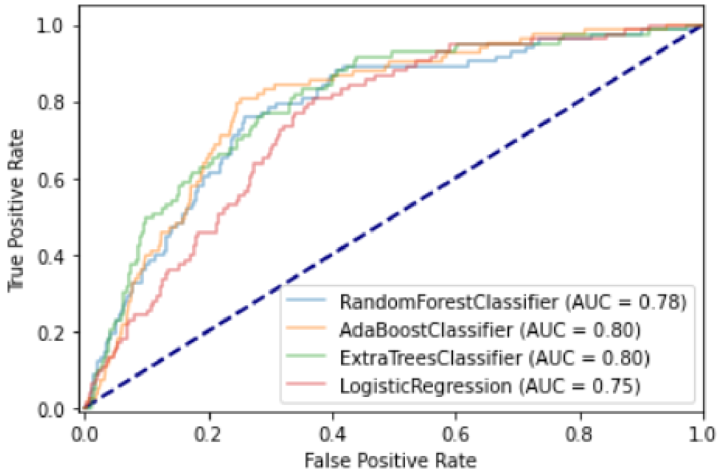
```
X_train, X_test, y_train, y_test = train_test_split(features, target, train_size=0.8,
                                                    stratify = target,
                                                    random_state=SEED, shuffle=True)
```

Before passing the dataset to the models, we will balance the representation of the target classes using a Random Sampler, first by reducing the majority class and then by increasing the minority class. Although we end up having a smaller sample, this will allow the models have a better idea of the characteristics of the target class (converted customers).

Proposed models

We propose 3 tree-based algorithms: Random Forest, Ada Boosting and Extra Trees. We will tune them using search grid with a small set of hyperparameters. For that we will also use a k-fold cross validation, to get a more robust estimate of the quality of our models.

To **evaluate the performance of the models** we will use the Area Under Curve (AUC) metric, since the target labels are highly imbalanced and the accuracy would be a bad choice to evaluate the model. Also, the AUC is the metric chosen to evaluate the final predictions in the Kaggle submissions section.



We can see that the 3 models outperform the benchmark logistic regression, with the Ada Boosting and the Extra Trees giving similar AUC values. We will choose the **Extra Trees** model, since it took only 1 minute to run, and it could be faster for future parameter tuning.

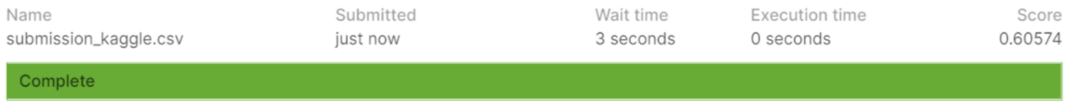
The model was previously saved using "dump" from joblib, so that it can be used in the future. It is always a good idea to do this, because it can prevent you from re-training the model in the future.

Model future improvements:

- More exhaustive grid search for hyper-parameter tuning.
- Tuning with Feature Selection. More features does not always mean better model, and sometimes selecting the adequate features we can obtain a more generalized model.
- Test more models.
- A more in-deep pre-processing (remove possible outliers, other scaling methods...)

Predictions

The data preprocessing applied to the general population and customers' datasets is applied to the MAIOUT train and test datasets, with the only exception that for the test no rows have been removed, since for the Kaggle submission, the predictions must have the same number of rows (I experienced that problem and had to re-do it).



Name	Submitted	Wait time	Execution time	Score
submission_kaggle.csv	just now	3 seconds	0 seconds	0.60574

Complete

The score of the submission could be improved following the stated model future improvements.