

Machine Learning Engineer Nanodegree

Capstone Project

Job Quiroz

March 15th, 2021

I. Definition

Project Overview

Marketing is essential for any sales-related business; however, marketing campaigns are expensive and sometimes even risky. In this project we contribute to create an effective marketing campaign for a mail-order company that uses machine learning techniques to estimate which individuals can be potential clients. The final solution can help to reduce the cost of marketing while increasing the chance of having new customers.

This project was proposed by Arvato Financial services. They have provided four datasets with attributes and demographic information from existing clients, and from a sample of the general population in Germany. Based on these datasets we have to gain a better understanding of the problem, provide insights about the difference between customers and the general population sets, and train a model able to predict potential customers.

Problem Statement

From a business perspective, the problem statement of this project is to answer the following question: "How can our client (a mail-order company of organic products) acquire new clients more efficiently?"

From a technical perspective, we would like to have a way to identify which attributes the customers have that the general population do not have. Since the number of attributes is so vast, first we have to find a way to keep the best attributes. This will allow us to organize subjects in different groups (unsupervised learning) and also to develop a predictive model (using supervised learning).

Metrics

For the supervised learning section of the project we will use the AUROC metric (Area Under the Receiver Operating Characteristics). AUROC is a performance metric for the classification

problems at various threshold settings. It indicates "how much the model is capable of distinguishing between classes"¹.

The main reason to choose AUROC over accuracy is the high imbalance in the training set (it has much more negative examples than positive ones).

II. Analysis

Data Exploration

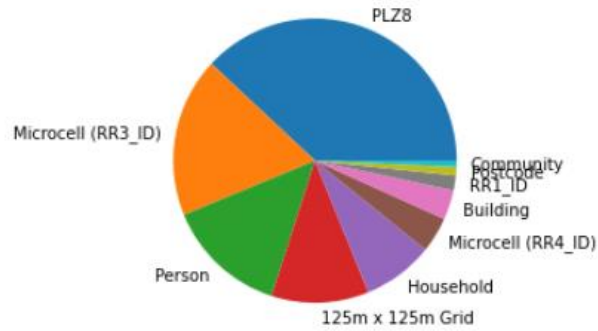
The data provided was in tabular format (csv files). After an initial analysis we found the following:

- most columns have missing values
- most attributes are numerical but in fact they are categorical variables, while some other are ordinal variables.
- most attributes do not contain outliers
- there are some attributes for which we don't have information (the *DIAS Attributes* and *DIAS Information levels* tables do not contain information about all the columns in the actual datasets)
- the *azdias* (general population) dataset has nearly 900,000 rows, the data processing has to be done using very efficient methods to avoid time processing problems.
- the *customers* dataset has three additional columns with respect to *azdias*. These attributes contain information about the type of customer, whether he/she buy online and what type of products they buy.
-

Exploratory Visualization

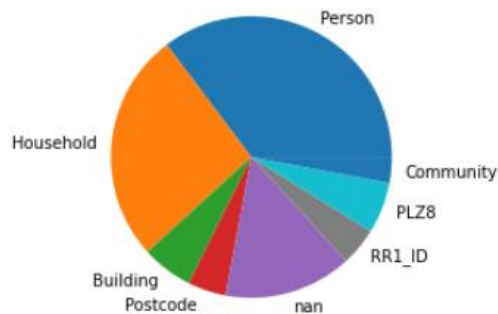
One of the auxiliar files contained the *information level* of each attribute. These information levels were in fact like attribute classes. The next figure shows the attribute distribution before cleaning:

¹ <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>



Attribute distribution before cleaning

After reading the descriptions of attributes in each class we discover that the PLZ8, Microcell (RR3_ID), and 125x125 m Grid, four of the most populated information levels had mostly irrelevant attributes like very specific characteristics of cars. After removing some of these attributes we got the following distribution:



Attribute distribution after cleaning

This final distribution shows that according to our understanding of the relation between the attributes and the problem to solve, the most numerous attributes will be those related with personal and household information.

Algorithms and Techniques

For the unsupervised section of the problem, we proposed to use Principal Component Analysis (PCA) to reduce the dimensionality of the features. To cluster similar samples we used the K-Means algorithm. These algorithms were chosen because they are efficient for the size of our datasets and they have been proven to offer good results for these tasks.

For the supervised section of this problem we used the following algorithms:

- RandomForestClassifier
- XGBoost
- AdaBoost

These three algorithms have good performances on tabular data and their programming implementation includes parameters to deal with imbalanced data.

Benchmark

There is not an specific benchmark for this project. However, to consider this problem solved we have to reach a performance metric higher than 0.5 (the value corresponding to a random classifier).

We can also try to reach a high rank in the Kaggle leaderboard. At the moment of writing this report the top 100 submission have an AUROC above 0.79.

III. Methodology

Data Preprocessing

The analysis of each feature was stated in the notebook *Arvato Project Workbook 1*, for the purpose of clarity we will not include the analysis of each attribute in this report. However, we can mention some key steps for this phase:

1. Attributes that were extremely specific like the properties of cars within a neighborhood were discarded.
2. Attributes in the *Person* information level class were treated carefully, including the age, gender, financial and family typologies and consumer behavior. We kept as much information as possible.
3. Attributes in *Finanz_...* were also treated carefully, however, most of these attributes needed the same kind of processing because they were in similar formats.
4. Attributes referring to transactional data (*D19...*) were also processed similarly, because its format was also similar.
5. The *household* attributes had a high number of missing values, and we did not have information about many attributes because they did not appear in the *AZDIAS Attributes* file.

6. Attributes for which we did not have information were kept if they had a few missing values (less than 10%) and they had ordinal values (rather than categorical ones).

Implementation and Refinement: Population segmentation

We can divide this analysis in four stages:

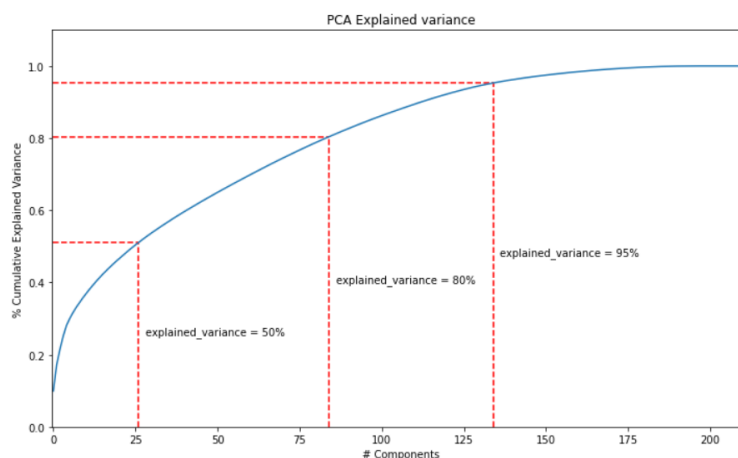
1. Reading cleaned data and scaling
2. Principal Component Analysis
 - o Component interpretation
3. Clustering
 - o K selection (elbow method)
4. Clustering analysis
 - o Comparing distributions of relevant features

1. Reading cleaned data and scaling

This step was straightforward: we read the csv files generated in the previous notebook and use the scikit-learn's `StandardScaler` estimator on the *azdias dataset* (fit and transform) and apply it to the *customers* dataset (only transform). We remove the column LNR as well as the last three columns of customers, because it has specific information about the customers.

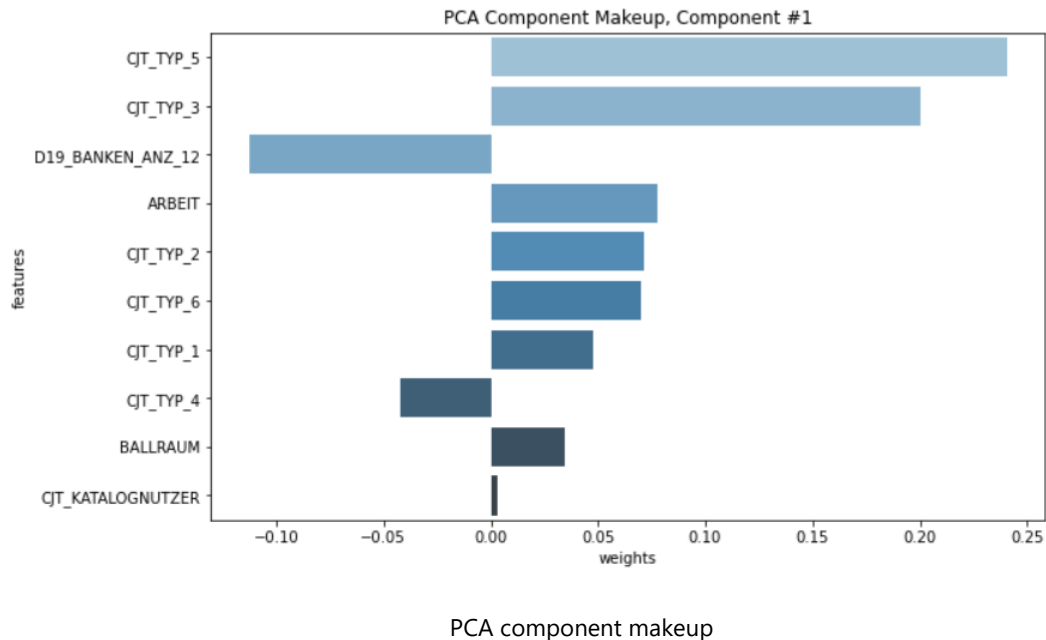
2. Principal Component Analysis

In this step we used the PCA estimator provided by scikit-learn and fit the scaled *azdias* dataset with the default parameters, and started experimenting with the different options to perform PCA. After this, we generated a graph of the *number of components vs the explained variance* of a PCA processing:



Number of PCA components vs cumulative explained variance

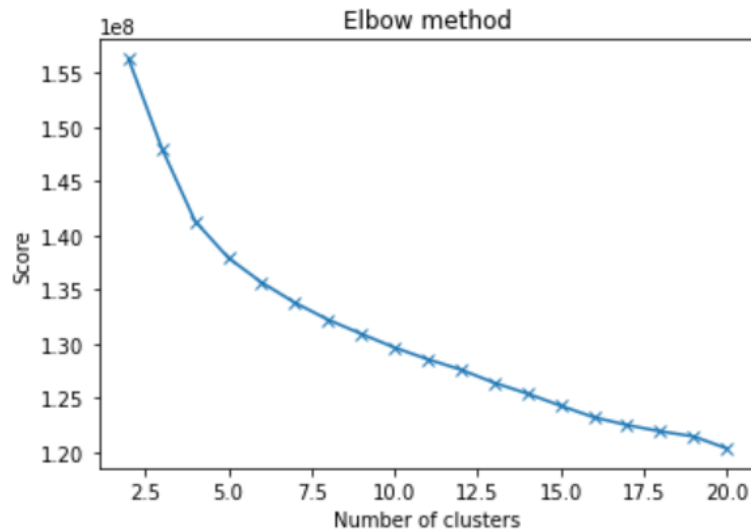
Another interesting experiment was to limit the number of components of the PCA estimator to 10, and we then displayed how some of such components were constituted of. For example, the following images show which attributes from *azdias* make up the component #1 of the resulting PCA:



We can recognize some features such as *CJT_TYP_#* from which we do not have information, but we can assume that are related to the typology of a consumer. We can also see attributes related to financial and job information, the distance of the subject to an urban center. The fact that these features are considered important makes sense with the problem we are trying to solve.

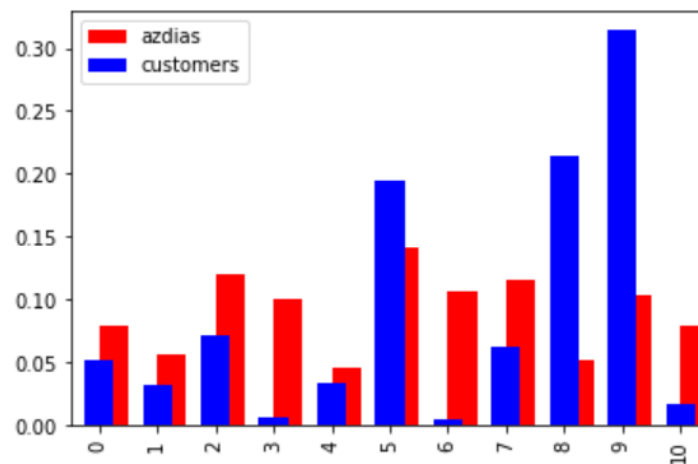
3. Clustering

In workbook 2 we realize a apply the K-Means algorithm to the scaled datasets. We proposed two different methods to select k , one was the elbow method, but we also performed a loop to see how the distribution of the datasets are different for different values of k and *explained variance*. The following figures show the curve obtained from the elbow method.



Elbow method for k selection

After performing several tests we decided to have a PCA estimator with 90% explained variance and a k value of 11. The following figure shows the distribution of subjects in different clusters for the azdias and customers datasets:



Cluster distribution

4. Clustering analysis

The main goal of the component analysis and the clustering algorithm was to **provide insights** about the difference between the customers and the general population. To clearly observe this differences, we chose two clusters: one in which the customers are **under-represented** and one where they are **over-represented** and then we plot the distributions for different value attributes. We will focus on clusters #3 and #9.

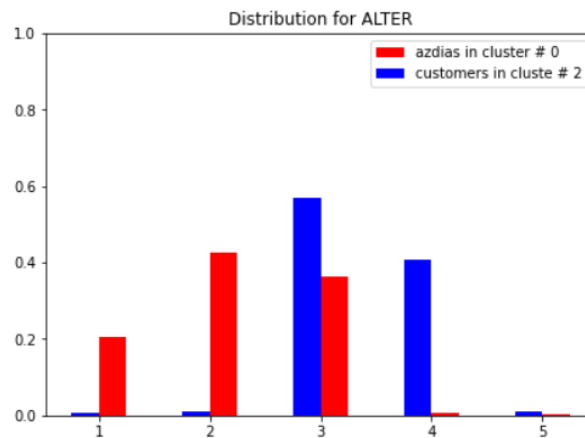
Some attributes where we can observe some differences are:

Age

This feature was introduced in workbook #1. It is a mixed between ALTERSKATEGORIE_GROB and GEBURTSJAHR:

- 1: Less than 30 years old
- 2: 30 – 45
- 3: 46 – 60
- 4: 60 – 75
- 5: > 75

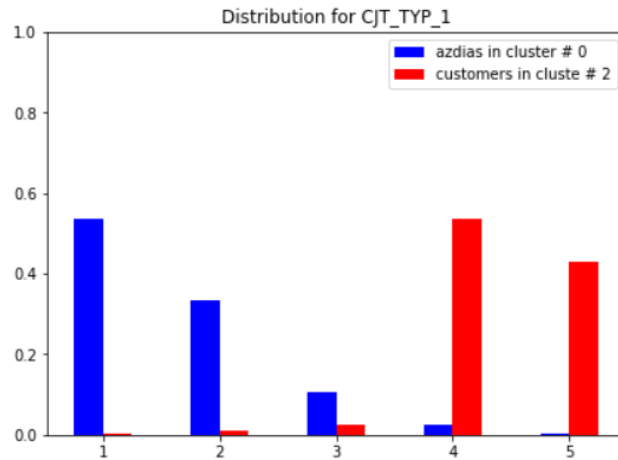
Customers tend to be middle-age people, young people do not seem to be potential customers



CJT_TYP_#

We don't have information about these features in the dictionary files (*probably* refers to a consumer typification), but they have been clearly engineered to differentiate between different type of consumers. *We cannot make a sound conclusion from these attributes because we don't know its details but they are very useful to detect customers.*

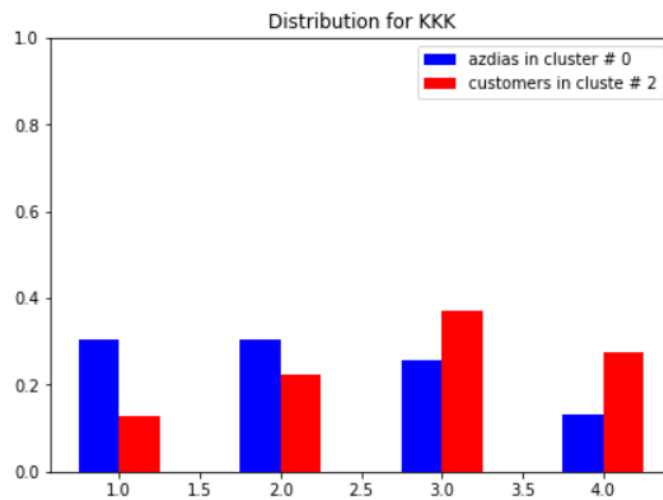
They also appeared in the component makeup above.



Purchasing power

- 1: Very high
- 2: high
- 3: average
- 4: low

There is a small tendency for customers to have from average to low purchasing power

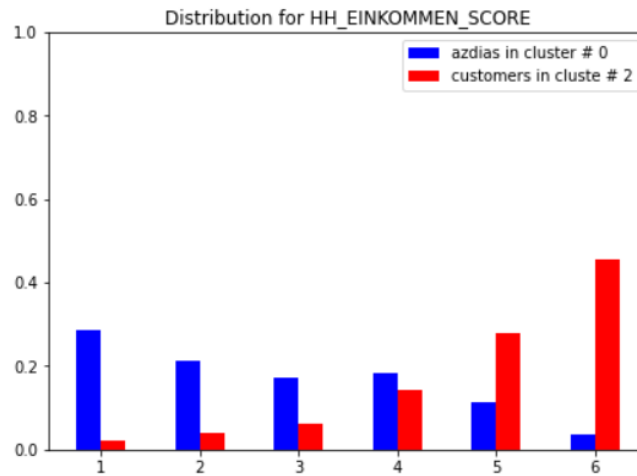


Estimated household net income

- 1: Highest income
- 2: Very high income
- 3: High income
- 4: Average income

- 5: lower income
- 6: very low income

This graph reinforce the previous plot, customers tend to be average and lower income persons

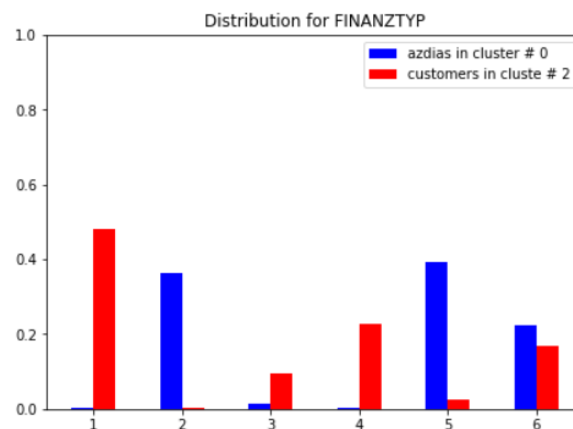


Finance typification

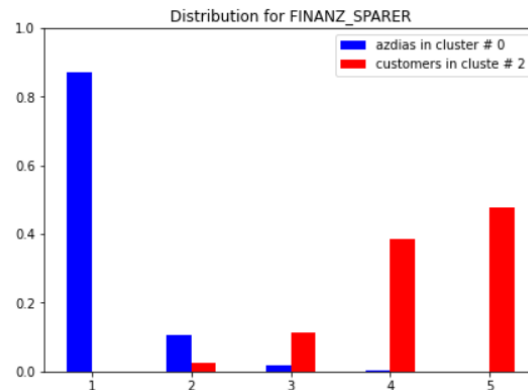
Best describing financial type of person.

- 1: low financial interest
- 2: money saver
- 3: main focus is the own house
- 4: be prepared
- 5: investor
- 6: unremarkable

Customers have low financial interest, but they are not money savers



Only for money savers:



*Customers of the mail order company are somewhat identified from the general population, our analysis shows that they tend to be **middle age people with average to low incomes** but with a tendency to buy, because they are **not money savers**.*

Implementation and refinement: Supervised learning algorithm

RandomForest Classifier

According to scikit-learn:

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

RandomForest is a very famous algorithm for tabular data, it can easily overfit the training data but by choosing the correct parameters it can have nice performance in classification tasks.

To tune this model we use a grid search estimator performing a 3-fold cross validation.

XGBoost

According to its [documentation](#):

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

This is also a very famous algorithm on Kaggle. It performs very well on tabular data, making it an obvious option for this problem.

Since it has many hyperparameters to tune, we used a random search estimator with 400 iterations.

AdaBoost

According to scikit-learn:

An AdaBoost [1] classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

To tune this model we use a grid search estimator performing a 3-fold cross validation.

Boruta feature selection

The unsupervised method used before allows to understand which features can be more important when it comes to clustering similar subjects. However, when we have a labeled dataset we can use a feature selection algorithm, such as Boruta, to decide which features have more statistical significance.

The boruta feature selection algorithm implements statistical methods to reject features that do not seem to contribute with the label for a given example.

After applying boruta to our cleaned and scaled training set, it only returned the following **6 features as relevant**:

- D19_KONSUMTYP_MAX
- D19_SOZIALES
- HH_EINKOMMEN_SCORE

- RT_SCHNAEPPCHEN
- CJT_GESAMTTYP_6.0
- D19_KONSUMTYP_6.0

We retrained the previous three algorithms with these features and obtained different results showed in section IV.




IV. Results

Model Evaluation and Validation

The following table shows the AUROC score for the three algorithms used, including the models trained after selecting features with Boruta.

Algorithm	Without feature selection		With feature selection	
	Training	Test	Training	Test
RandomForest	0.861	0.767	0.804	0.7987
XGBoost	0.786	0.789	0.799	0.798
AdaBoost	0.786	0.793	0.786	0.795

The RandomForest classifier using the Boruta feature selection achieve the best results: **0.7987**. At the time of writing this report this model was in the 105 position of the leaderboard.

105	JobQuiroz		0.79873	6	now
Your Best Entry 					
Your submission scored 0.79873, which is an improvement of your previous score of 0.79743. Great job!				 Tweet this!	

Justification

The final models reach a scores very close to 0.8, meaning that our work was definitely better than assigning labels at random (auroc = 0.5). The position in the leaderboard was also a good benchmark, we were near the top 100 best performing models.

Our solution is significant to solve the problem of focused marketing because even though our model won't be 100% accurate, the recommendations it provides can help the marketing team to take better decisions.

V. Conclusion

Reflection

This was a very interesting problem, it required the use of several machine learning techniques as well as data wrangling skills. The high amount of noisy data was a challenge, but data cleaning is always hard work.

Improvement

As the leaderboard shows there is still room for improvement, we could try to implement a different set of algorithms or maybe use more iterations in the hyperparameter tuning process.

Also, it would be interesting to know the meaning of the CJT_TYP_# attributes, because they surely have important information, probably coming from another previous analysis.