

Machine Learning Engineer Nanodegree

Capstone Project

Job Quiroz

March 16th, 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).

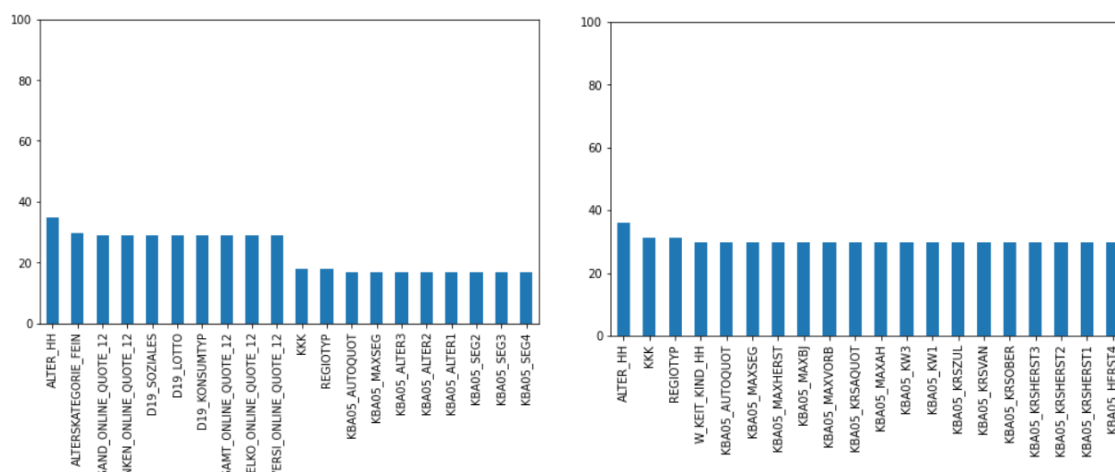
II. Analysis

Data Exploration

The data provided was in tabular format (csv files). The dataset of the general population (*azdias*) contained 891,221 rows and 365 columns, around 75% of which had at least one missing value, and with 16 columns having more than 25 % values missing.

On the other hand, the dataset of the customers contained 191,652 rows and 369 columns. These four additional columns included information about whether or not the purchase was online, an ID for each customer (LNR), and a two labels to identify to which customer and product group a subject is part of.

Around 75% of the columns in the *customers* dataset contained at least 1 missing value, but unlike the previous dataset, 228 columns had more than 25% missing values. The following figures show this difference:



Top 20 features with missing values from the *customers* dataset. Around 9% of the values in the *azdias* were missing, while the *customers* dataset had 18% of missing values.

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

For each single column an analysis of the value distribution was performed. In this analysis we:

- Printed the percentage of missing values
 - o For columns with a few missing values (<5%) we filled with the most common value for the column or filled at random
 - o For columns with more than 5% missing values we tried to impute values from other attributes, tried to fill with the most common value, or we dropped the column.
- Check if the variable was categorical or ordinal
 - o If it was categorical, we decided if it was worth it to assign dummy variables, but if the number of categories was too large we dropped the column
 - o If it was ordinal we look for outliers that could affect later computations
- Check the minimum and maximum values and realize whether or not they made sense
- Created new useful features. An example is the attribute "ALTER" (age) which was created by combining GEBURTSJAHR (estimated year of birth) and ALTERSKATEGORIE_GROB (age group).

```
|: 1 general_attribute_info('ANZ_TITEL')
Percentage of missing values in customers 24.31282 %
Percentage of missing values in azdias 8.24700 %

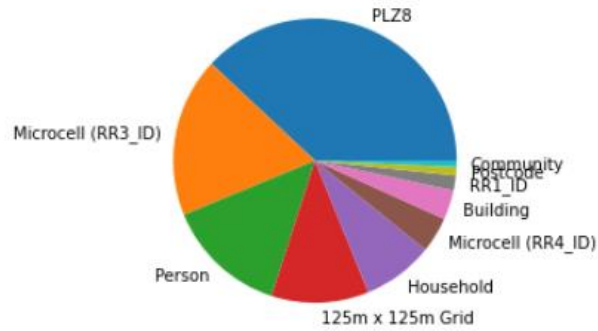
Frequencies in azdias:
0.0      814542
NaN       73499
1.0        2970
2.0         202
3.0           5
4.0           2
6.0           1
Name: ANZ_TITEL, dtype: int64

Frequencies in customers:
0.0      142316
NaN      46596
1.0       2533
2.0        198
3.0          8
5.0          1
Name: ANZ_TITEL, dtype: int64
```

An output example of the function used to analyze features, due to the higher number of features in the dataset numerical output was preferred over histograms.

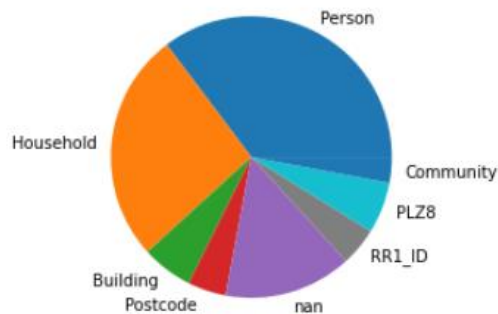
Exploratory Visualization

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:



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. A meta estimator that fits a number of decision tree classifiers on various subsamples of the datasets.² This algorithm is very used when learning tabular data, one disadvantage is that it can easily overfit the training data, but by choosing the right hyperparameters this effect can be mitigated.

XGBoost. An optimized distributed gradient boosting algorithm that implements parallel tree boosting techniques to solve learning problems³. This algorithm is also highly efficient running on major distributed environments.

AdaBoost. This algorithm is also 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; subsequent classifiers focus more on difficult cases.⁴

Boruta feature selection. The boruta feature selection algorithm implements statistical methods to reject features that do not seem to contribute with the label for a given example. Since the number of features remains to be high after cleaning, we propose to use the Boruta technique to find the most important features. After performing these selection we could use the same three algorithms as before focusing on less features.

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.

² <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

³ <https://xgboost.readthedocs.io/en/latest/>

⁴ <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>

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:

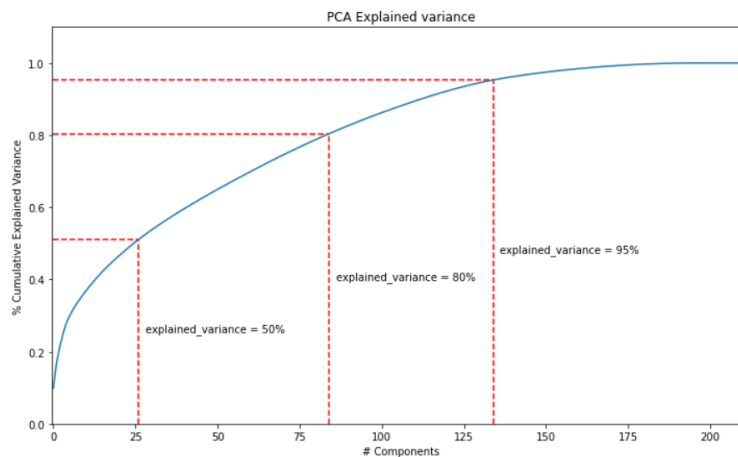
1. Reading cleaned data and scaling
2. Principal Component Analysis
 - Component interpretation
3. Clustering
 - K selection (elbow method)
4. Clustering analysis
 - Comparing distributions of relevant features

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.

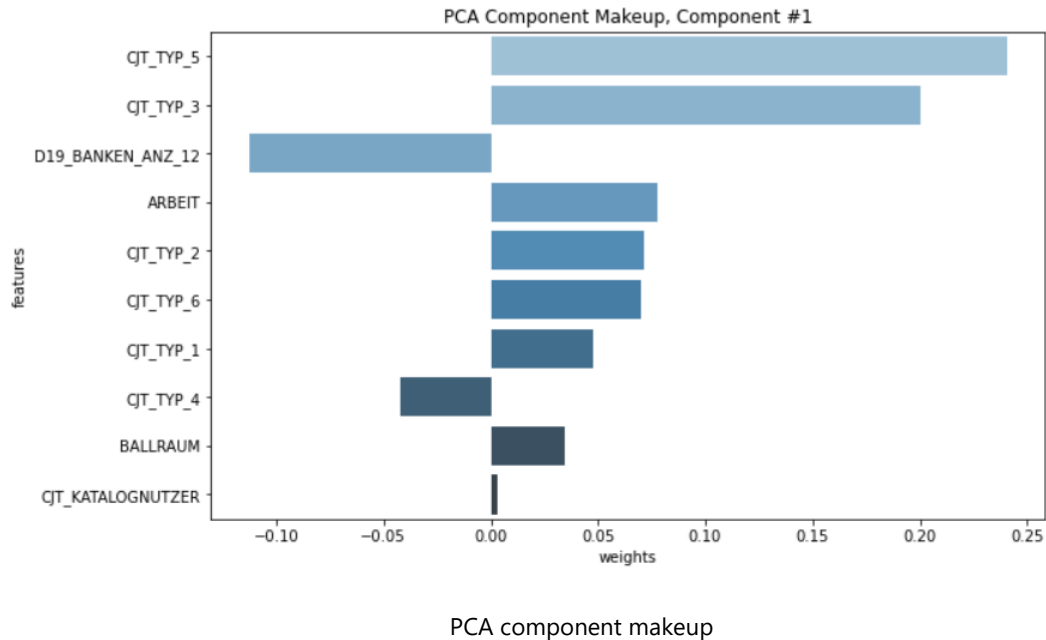
2. Principal Component Analysis

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

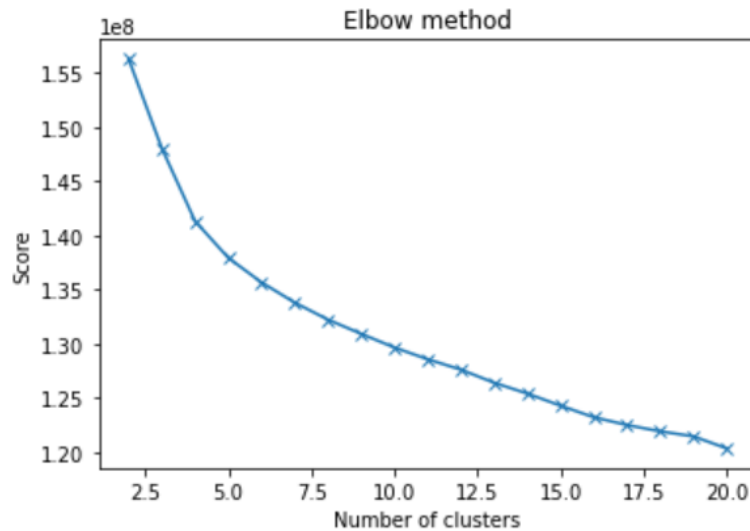
Another interesting experiment was to limit the number of components of the PCA estimator to 10, and we then displayed how some of such components were constituted of. For example, the following images show which attributes from *azdias* make up the component #1 of the resulting PCA:



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.

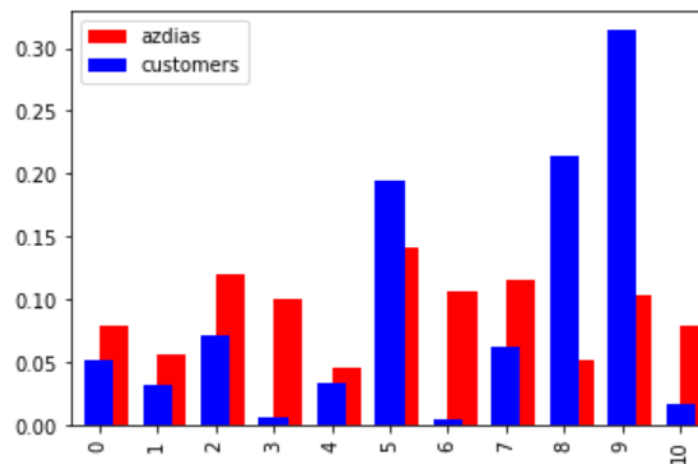
3. Clustering

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.

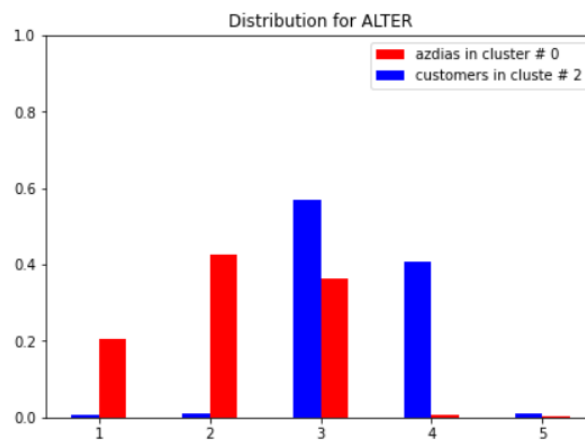
Some attributes where we can observe some differences are:

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

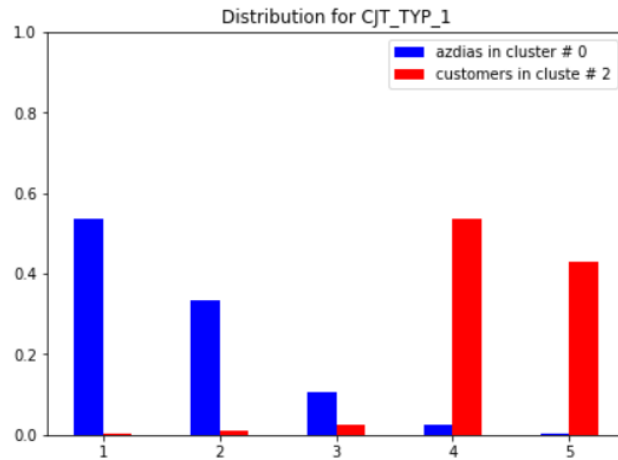
Customers tend to be middle-age people, young people do not seem to be potential customers



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.*

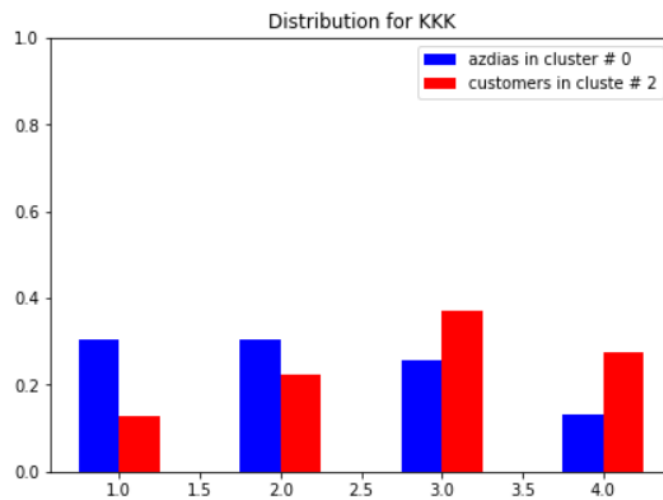
They also appeared in the component makeup above.



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

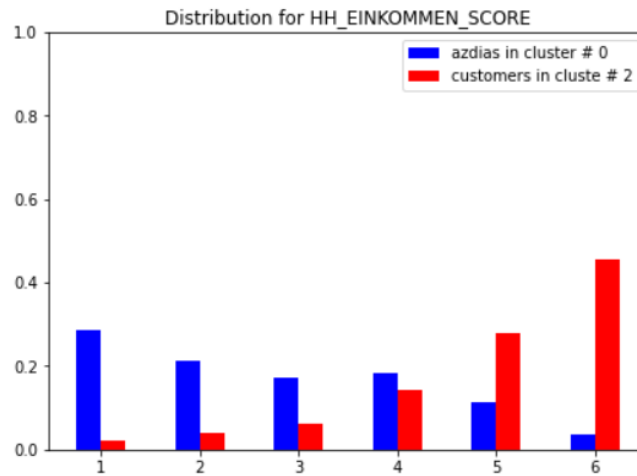


Estimated household net income

- 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

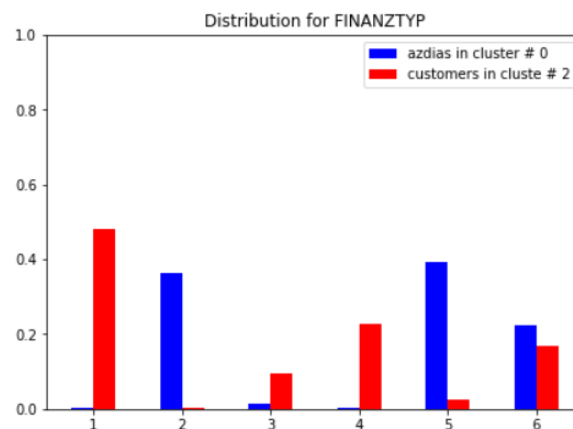


Finance typification

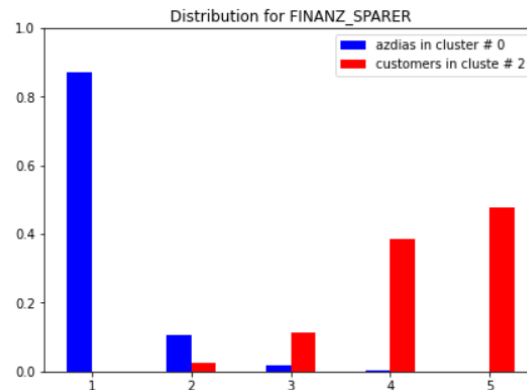
Best describing financial type of person.

- 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

Training and testing was done following these steps:

- Apply a cleaning function to the training set (it implements the steps developed in section II)
- Normalize and scale the cleaned training dataset using scikit-learn "StandardScaler"
- Define a cross-validation searching pipeline over a pre-defined set of hyperparameters
 - For the RandomForest and AdaBoost classifiers we selected a GridSearchCV, which performs an exhaustive search over all the parameters.
 - For XGBoost we selected a RandomSearchCV, which randomly chooses values of the hyperparameter set (our trained XGBoost model has a higher number of hyperparameters than the other two models, so a RandomSearch was a better option)
 - In both cases we selected the scoring parameter as 'roc_auc'.
- After the previous definitions we called the *fit* method, plotted the highest scores and stored the best estimator.
- Finally, we used the best estimator to predict the classes of the testing set.

Boruta feature selection

As mentioned in the Algorithms and Techniques section we proposed to apply a feature selection algorithm to the training and testing sets to identify which features seem to be more relevant. In a final phase of the project we applied the Boruta feature selection technique to the training set and train new models using only the following six features:

- D19_KONSUMTYP_MAX
- D19_SOZIALES
- HH_EINKOMMEN_SCORE
- RT_SCHNAEPPCHEN
- CJT_GESAMTTYP_6.0
- D19_KONSUMTYP_6.0

The final results showed a slight increase in the AUROC score for the three algorithms tested, (see the following section).

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