Capstone Project – CREATE A CUSTOMER SEGMENTATION REPORT FOR ARVATO FINANCIAL SERVICES

Udacity – MACHINE LEARNING ENGINEER NANODEGREE

SEPTEMBER 3, 2020

HAZEM HAMADA ABDELLATIF
zomahamada.hh@gmail.com

ABSTRACT

In this project, we analyse demographic data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population. EDA is performed to understand and clean the data. Unsupervised learning techniques are used to perform customer segmentation, identifying the parts of the population that best describe the core customer base of the company. Then, we'll apply what we've learned on a third dataset with demographic information for targets of a marketing campaign for the company, and use a model to predict which individuals are most likely to convert into becoming customers for the company.

Keywords: Exploratory Data Analysis(EDA), Unsupervised Learning, Supervised Learning.

1 Definition:

1.1 Project Overview:

In this project, a mail-order sales company in Germany is interested in identifying segments of the general population to target with their marketing, to grow their customer base. Demographics information is available for both the general population as well as for prior customers of the company. We use this information to build a model of the customer base of the company. The target dataset contains demographic information for targets of a mailout marketing campaign. The objective is to identify which individuals are most likely to respond to the campaign and become customers of the mail-order company. The data has been provided by Bertelsmann Arvato Analytics and consists of four data files:

- 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).
- Udacity_MAILOUT_052018_TRAIN.csv: Demographics data for individuals who were targets of a marketing campaign; 42 982 persons (rows) x 367 (columns).
- Udacity_MAILOUT_052018_TEST.csv: Demographics data for individuals who were targets of a marketing campaign; 42 833 persons (rows) x 366 (columns).

The "CUSTOMERS" file contains three extra columns ('CUSTOMER_GROUP', 'ONLINE_PURCHASE', and 'PRODUCT_GROUP'), which provide broad information about the customers depicted in the file. Each row of the demographics files represents a single person, but also includes information outside of individuals, including information about their household, building, and neighbourhood.

1.2 Problem Statement:

The goal is to identify segments of the population that form the core customer base for the company. These segments can then be used to direct marketing campaigns towards audiences that have the highest expected rate of returns.

The information from the first two files is used to figure out how customers ("CUSTOMERS") are like or differ from the general population at large ("AZDIAS"), then use this analysis to make predictions on the other two files ("MAILOUT"), predicting which recipients are most likely to become a customer for the mail-order company.

The original "MAILOUT" file included one additional column, "RESPONSE," which indicated whether each recipient became a customer of the company. For the "TRAIN" subset, this column is present, but in the "TEST" subset it has been removed; it is against that withheld column that we will asses the final predictions. The higher the score obtained, the better the model is at predicting customers.

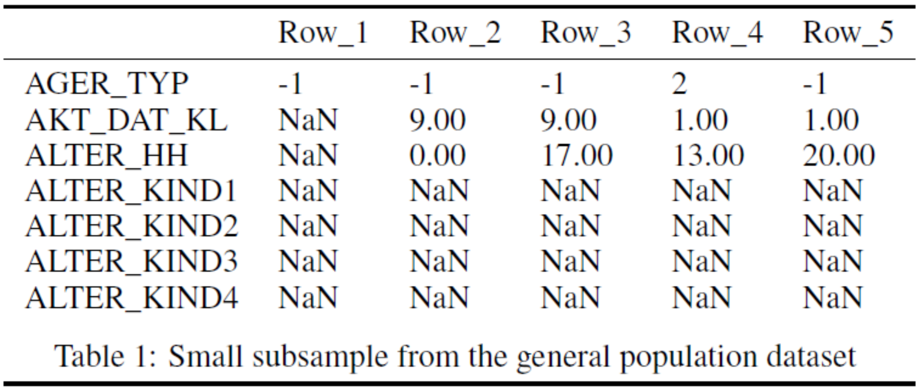
1.3 Metrics:

The evaluation metric for the Kaggle competition is RMSE which gives us a score of 0.74659, this score is provided from the Kaggle submission. The model hyperparameters are adjusted using hyperparameter tuning in SageMaker.

2 Analysis:

2.1 Data Exploration:

The demographic data for the general population of Germany contains 366 features, and to display it here, we must transpose it. Below are the first 5 samples from the first 10 features:



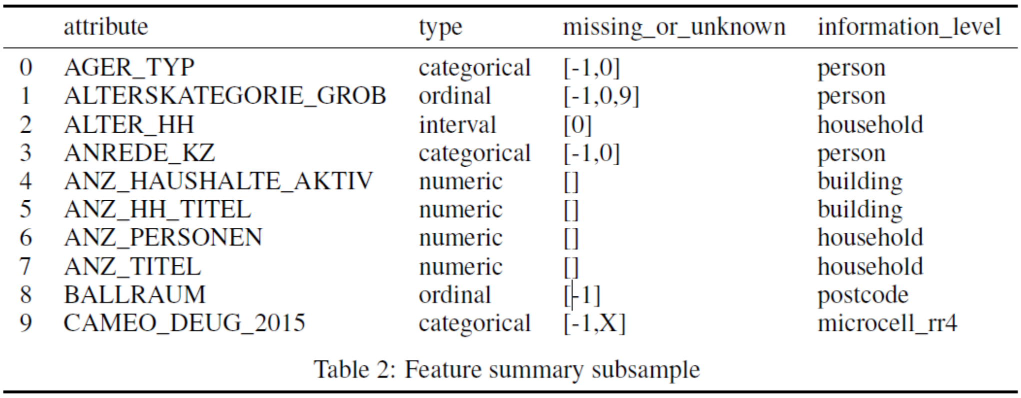
	Row_1	Row_2	Row_3	Row_4	Row_5
AGER_TYP	-1	-1	-1	2	-1
AKT_DAT_KL	NaN	9.00	9.00	1.00	1.00
ALTER_HH	NaN	0.00	17.00	13.00	20.00
ALTER_KIND1	NaN	NaN	NaN	NaN	NaN
ALTER_KIND2	NaN	NaN	NaN	NaN	NaN
ALTER_KIND3	NaN	NaN	NaN	NaN	NaN
ALTER_KIND4	NaN	NaN	NaN	NaN	NaN

Table 1: Small subsample from the general population dataset

The feature names are not very explanatory, but fortunately for us along with the datasets, we have two other files:

- DIAS Information Levels - Attributes 2017.xlsx: Describes each feature.
- DIAS Attributes - Values 2017.xlsx: Describes the type of each feature along with possible values along with values that represent missing or unknown information.

As we can see the dataset contains a lot of missing values and some of the values, like (-1, 0 or 9) also indicate missing or unknown information. Based on the DIAS Attributes - Values 2017.xlsx we've built a new dataset AZDIAS_Feature_Summary.csv that contains a summary of properties for each demographics data column, as follows:



	attribute	type	missing_or_unknown	information_level
0	AGER_TYP	categorical	[-1,0]	person
1	ALTERSKATEGORIE_GROB	ordinal	[-1,0,9]	person
2	ALTER_HH	interval	[0]	household
3	ANREDE_KZ	categorical	[-1,0]	person
4	ANZ_HAUSHALTE_AKTIV	numeric	[]	building
5	ANZ_HH_TITEL	numeric	[]	building
6	ANZ_PERSONEN	numeric	[]	household
7	ANZ_TITEL	numeric	[]	household
8	BALLRAUM	ordinal	[-1]	postcode
9	CAMEO_DEUG_2015	categorical	[-1,X]	microcell_rr4

Table 2: Feature summary subsample

We use this file to help us make cleaning decisions for the project.

Missing Values: The third column (missing_or_unknown of the feature attributes summary, documents the codes from the data dictionary that indicate missing or unknown data. Before converting data that matches a 'missing' or 'unknown' value code into a NaN value, we first have a look how much data takes on a 'missing' or 'unknown' code, and how much data is naturally missing, as a point of interest.

Select and Re-Encode Features: Checking for missing data isn't the only way in which we can prepare a dataset for analysis. Since the unsupervised learning techniques, we use only work on data that is encoded numerically, we need to make a few encoding changes or additional assumptions to be able to make progress. While almost all the values in the dataset are encoded using numbers, not all of them represent numeric values.

- For numeric and interval data, these features can be kept without changes.
- Most of the variables in the dataset are ordinal. While ordinal values may technically be non-linear in spacing, we make the simplifying assumption that the ordinal variables can be treated as being an interval in nature (that is, kept without any changes).
- Special handling may be necessary for the remaining two variable types: categorical, and 'mixed.'

Categorical Features: For categorical features, we encode the levels as dummy variables. Depending on the number of categories, we can perform one of the following:

- For binary (two-level) categorical features that take numeric values, we can keep them without needing to do anything.
- If there are binary variables that take on non-numeric values, we need to re-encode the values as numbers or create a dummy variable.
- For multi-level categorical features (three or more values), we can choose to encode the values using multiple dummy variables

Mixed-Type Features: There are a handful of features that are marked as "mixed" in the feature summary that require special treatment before we can include then in the analysis. There are two that deserve attention:

- PRAEGENDE_JUGENDJAHRE combines information on three dimensions: generation by decade, movement (mainstream vs. avantgarde), and nation (east vs. west). While there aren't enough levels to disentangle east from west, we create two new variables to capture the other two dimensions: an interval-type variable for the decade, and a binary variable for movement.
- CAMEO_INTL_2015 combines information on two axes: wealth and life stage. We break up the two-digit codes by their 'tens'-place and 'ones'-place digits into two new ordinal variables (which, for this project, is equivalent to just treating them as their raw numeric values).

Ordinal and Interval Features: Nothing special here, we need to decide what to do with the missing values.

Numerical Features: Nothing special here, we need to decide what to do with the missing values and perhaps perform
some scaling.

2.2 Exploratory Visualization:

Figure 2, obtained using the missingno package shows the missing values of the provided data and figure 2b same information but after we replaced the unknown or missing value codes with NaN.

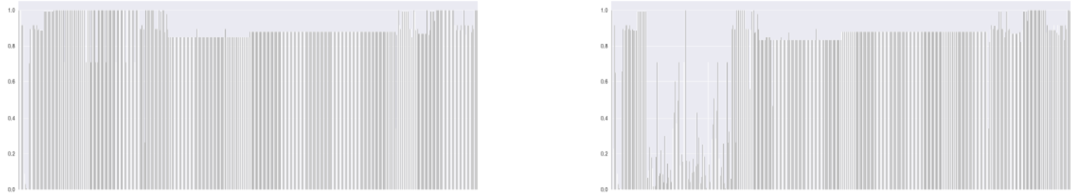
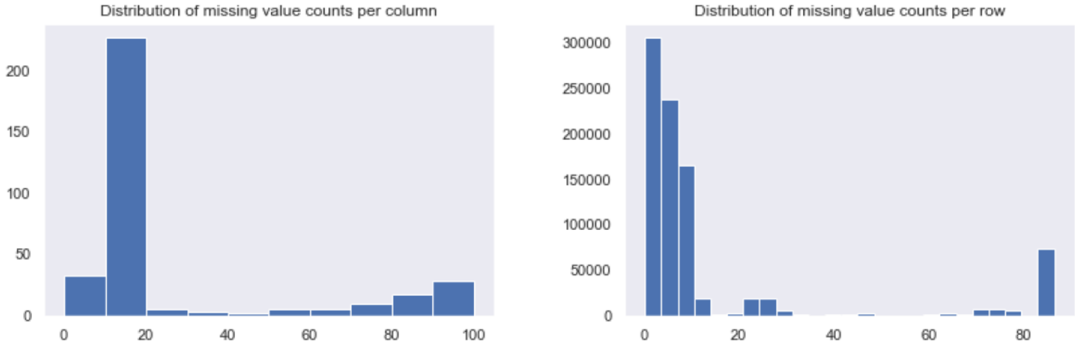


Figure 2: Missing values per column before and after replacing unknown value codes

In figure 3a we have the distribution of missing value counts where we can see that that there are a few columns that are outliers in terms of the proportion of values that are missing. We also perform a similar assessment for the rows of the dataset to asses how much data is missing in each row (see figure 3b). As with the columns, we see some groups of points that have a very different number of missing values.



(a) Distribution of missing value counts per column

(b) Distribution of missing value counts per rows

Figure 3: Distribution of missing value counts per column and row

2.3 Algorithms and Techniques:

There are 2 parts of this project:

Unsupervised modelling: The main bulk of our analysis is in this part of the project. Here, we use unsupervised learning techniques to describe the relationship between the demographics of the company's existing customers and the general population of Germany. By the end of this part, we should be able to describe parts of the general population that are more likely to be part of the mail-order company's primary customer base, and which parts of the general population are less so.

We first apply PCA: (Principal Component Analysis) to reduce the dimensionality of the dataset. We decide on a number of components that explain at least 90% of the variance in data.

After PCA, we create KMeans models with clusters from 2 to 15. Clustering is a method of unsupervised learning, where each data point or cluster is grouped into a subset or a cluster, which contains similar kind of data points. We decide on the best number of clusters to take based on the Elbow method.

Supervised modelling: To predict the probability of a person to reply to the mailing campaign, we create an XGBoost-Classifier model which we use to predict this probability. Before training the model, we start by searching the best hyperparameters for models using all available features by SageMaker hyperparameters tuning.
Once we have a list of optimized hyperparameters, we use them for training a model on the resampled data. After training the model is used to predict on the TEST dataset.

2.4 Benchmark Model:

The original "MAILOUT" file included one additional column, "RESPONSE", which indicated whether each recipient became a customer of the company. For the "TRAIN" subset, this column has been retained, but in the "TEST" subset it has been removed; it is against that withheld column that the final predictions will be assessed in a Kaggle competition. The higher the score obtained in the Kaggle competition, the better the model is at predicting customers.

3 Methodology:

3.1 Data Pre-processing:

Using the analysis and data exploration above, we've built the following pre-processing function (the same cleaning process will also be applied on the training and testing datasets)

The pre-processing pipeline has the following components:

Parsing composed of two steps:

- Parse Missing or Unknown - custom pipeline that uses the feature summary constructed dataset (column missing_or _unknown) to recode values as NaNs.
- Drop columns with T% NaNs - Custom pipeline that drops all features that have more than T% (75% in our case) missing values

Transformation applies the following:

- Categorical Transformation applies the following steps to all categorical features and mixed-type features:
    - Transform the categorical values to numeric values.
    - Fill in the missing values using the most frequent value along each column.
    - Re-engineer the PRAEGENDE_JUGENDJAHRE mixed-type feature into two additional categorical features DECADE and MOVEMENT and then drop the original column.
    - Re-engineer the CAMEO_INTL_2015 mixed-type feature into two additional categorical features WEALTH and LIFE_STAGE and then drop the original column.
    - Re-engineer the CAMEO_DEU_2015 mixed-type feature into two additional categorical features CAMEO_DEU_2015_1 and CAMEO_DEU_2015_2 and then drop the original column.
    - Change data type.

- Ordinal Transformation applies the following steps to all ordinal features:
    - Fill missing values.
    - Change data type.

- Numerical Transformation applies the following steps to all numeric features:
    - Fill missing values.
    - Change data type.

In figure 5, we can see the distribution of feature types before and after dropping features with more than 75% missing values.

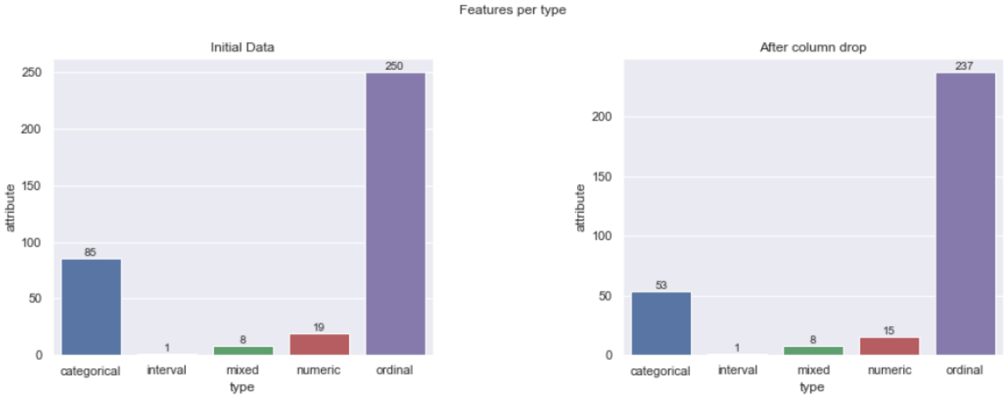


Figure 5: Feature per type

We also perform an analysis for rows with a lot of missing values. As we can see in Figure 3b, there are some rows with more than 85% missing values per row.

To know what to do with the outlier rows, we look the distribution of data values in columns that are not missing data (or are missing very little data) are similar or different between the two groups.

As we can see in Figure 13 in Section 6, the data with many missing values looks very different from the data with few or no missing values. We decide not to remove these rows.

3.2 Implementation:

3.2.1 Perform Dimensionality Reduction:

On our pre-processed data, we are now ready to apply dimensionality reduction techniques.

We use sklearn's PCA class to apply principal component analysis on the data, thus finding the vectors of maximal variance in the data.

We start by fitting a PCA on 685 dimensions (our initial dataset has 366 dimensions but increases to 685 after encoding the categorical features). You can find the results for the PCA in figure 6 below.

We check out the ratio of variance explained by each principal component as well as the cumulative variance explained.

Based on the results from the PCA fitted previously, we decide to keep the first 150 reduced dimensions, that explain 90% cumulative variance in data.

Now that we have our transformed principal components, we check out the weight of each variable on the first few components to see if we can interpret them some fashion.

Each principal component is a unit vector that points in the direction of highest variance (after accounting for the variance captured by earlier principal components). The further a weight is from zero, the more the principal component is in the direction of the corresponding feature. If two features have large weights of the same sign (both positive or both negative), then increases in one tend to expect to be associated with increases in the other. To contrast, features with different signs can be expected to show a negative correlation: increases in one variable should result in a decrease in the other.

To investigate the features, we map each weight to their corresponding feature name, then sort the features according to weight. The most interesting features for each principal component, are those at the beginning and end of the sorted list.

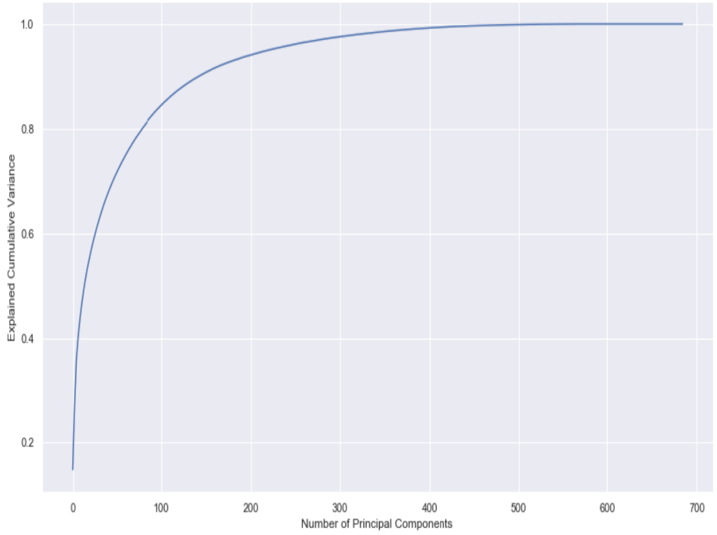


Figure 6: Principal Component Analysis

We present here the investigation of feature associations from the first three principal components:

Top 5 weights for PC1

D19_GESAMT_ONLINE_QUOTE_12	0.4903
D19_VERSAND_ONLINE_QUOTE_12	0.4737
ONLINE_AFFINITAET	0.1393
D19_BANKEN_ONLINE_QUOTE_12	0.0913
D19_GESAMT_ANZ_12	0.0885

//First component is all about online affinity and online transactions in the last 12 months.

Top 5 weights for PC2

ALTER_HH	0.3310
SEMIO_REL	0.2465
SEMIO_PFLICHT	0.2112
FINANZ_SPARER	0.2055
ORTSGR_KLS9	0.1711

// Second component describes the number and age of inhabitants, affinity to religion, being traditional minded and money saver financial topology.

Top 5 weights for PC3

ORTSGR_KLS9	0.3036
EWDICHTE	0.2092
SEMIO_ERL	0.1452
FINANZ_HAUSBAUER	0.1156
SEMIO_LUST	0.1153

// Third component describes the number and density per square kilometre of inhabitants, affinity to events and being sensual minded, as well as having the house as the main financial focus.

Next, we see how the data clusters in the principal components space. We apply k-means clustering to the dataset and use the average within-cluster distance to decide the number of clusters to keep. We use sklearn's KMeans class to perform k-means clustering on the PCA-transformed data.

We fit a KMeans model on the 150 reduced dimensions, and we investigate the change within-cluster distance across for a range of clusters between 2 and 16.

Based on Figure 7, we can see that a good number of clusters is 8. We refit the k-means model with the selected number of clusters and obtain cluster predictions for the general population demographics data.

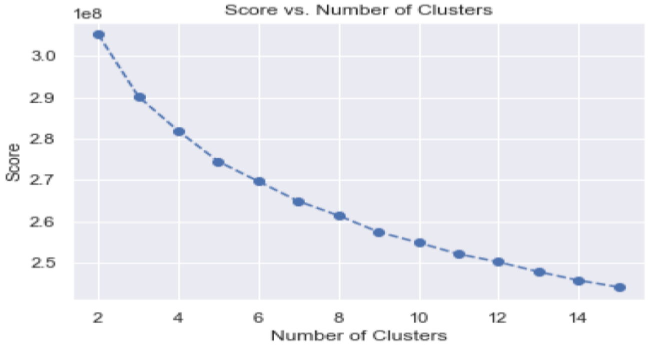


Figure 7: Average within-cluster distances

We compare the proportion of data in each cluster for the customer data to the proportion of data in each cluster for the general population:

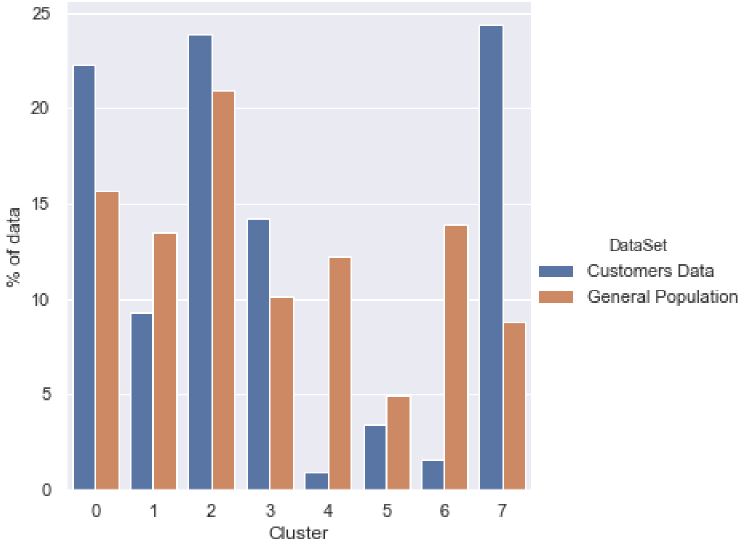
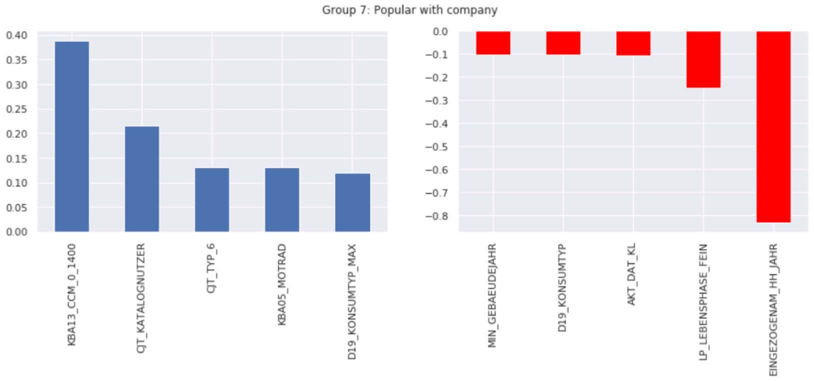


Figure 8: Proportion of data points in each cluster for the general population and the customer data.

We inspect what kind of people are part of a cluster that is over-represented in the customer data compared to the general population (cluster 7):

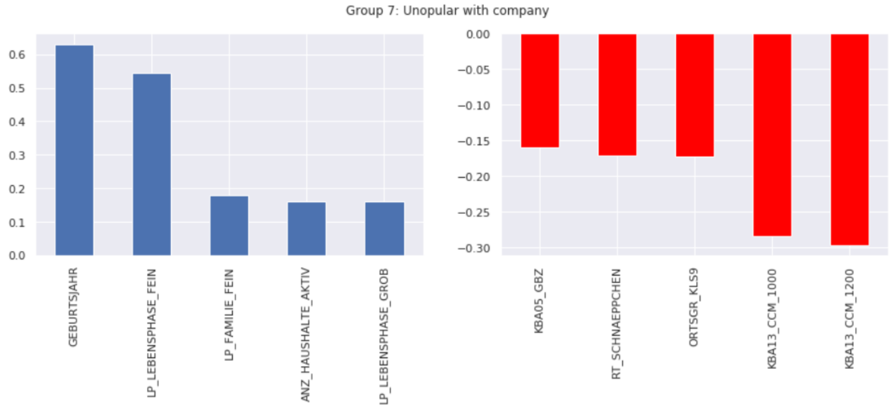
Popular with the company - Cluster 7 By using PCA's inverse transform we obtain the following values:

- S'KBA13_CCM_0_1400'
- 'CJT_KATALOGNUTZER'
- 'CJT_TYP_6'
- 'KBA05_MOTRAD'
- 'D19_KONSUMTYP_MAX'



Unpopular with the company - Cluster 4 By using PCA's inverse transform we obtain the following values:

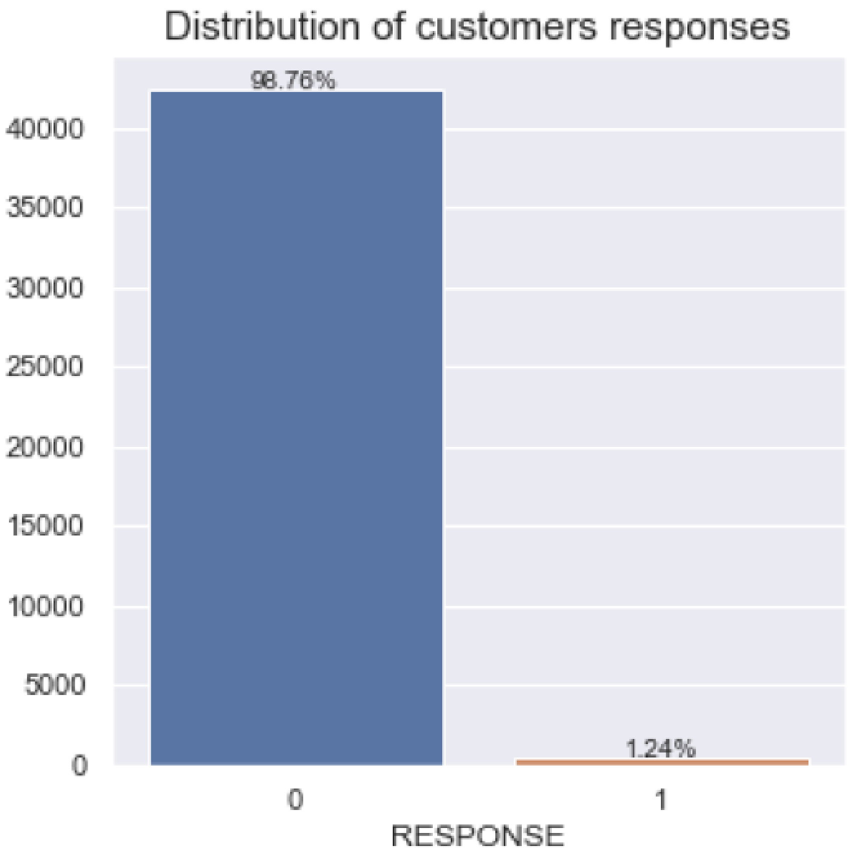
- 'GEBURTSJAHR'
- 'LP_LEBENSPHASE_FEIN'
- 'LP_FAMILIE_FEIN'
- 'ANZ_HAUSHALTE_AKTIV'
- 'LP_LEBENSPHASE_GROB'



3.2.2 Supervised Learning Model:

To implement the supervised model, we start by pre-processing the training dataset using the same pipeline as above.

We start by analysing the training dataset and especially the distribution between the two types of responses: 0-non-customer and 1-customer:



We split the dataset in train and test datasets by specifying that the two datasets are to be stratified using the target and keep the same weight for the classes. The proportion of data after splitting is 80% for training and 20% for validation. We used SageMaker hyperparameters tuning to find the best hyperparameters values. XGBoost Classifier model is used.

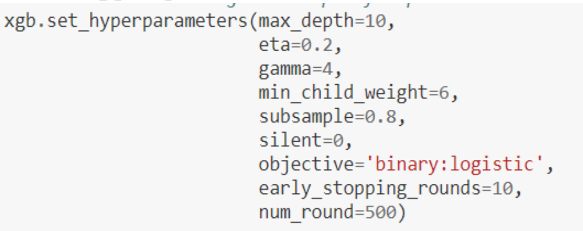
4 Results:

4.1 Model Evaluation and Validation:

We obtain the following results (the best model after tuning the hyperparameters):

```
train-rmse:0.110526
validation-rmse:0.106726
```

the initial hyperparameters were:



```
xgb.set_hyperparameters(max_depth=10,
                        eta=0.2,
                        gamma=4,
                        min_child_weight=6,
                        subsample=0.8,
                        silent=0,
                        objective='binary:logistic',
                        early_stopping_rounds=10,
                        num_round=500)
```

And the tuning job did some changes: tree pruning end, 1 roots, 10 extra nodes, 16 pruned nodes, max_depth=5

4.2 Justification:

The final tuned XGBoost Classifier performs better not only on the validation set but also on the test set held for the Kaggle competition.

5 Conclusion:

5.1 Reflection:

For a direct marketing campaign, it is essential to correctly identify the customers who will respond to a particular campaign.

In this project, we analysed demographic data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population. Exploratory Data Analysis was performed to understand and clean the data. Unsupervised learning techniques were used to perform customer segmentation, identifying the parts of the population that best describe the core customer base of the company. Then, we applied what we've learned on a third dataset with demographic information for targets of a marketing campaign for the company, and use a model to predict which individuals are most likely to convert into becoming customers for the company.

This project was an excellent opportunity to apply and learn new techniques primarily related to imbalanced data problems. Also going beyond simple grid search for hyper-parameter tuning was both a new tool to learn and also a time saver.

5.2 Improvement that can be made:

Reflecting on the steps taken in this project, we can identify some areas where improvements can be made:

- Data pre-processing - Engineer more categorical features: We believe that better results can be obtained if more categorical features are treated like mixed-type features and re-engineered.
- Data pre-processing - Missing Data:
- Analyse if there is data missing at random or there are patterns.
- Try to find correlations between missing values and use PCA to remove some of them.
- Use a supervised model to predict the values for NaN instead of just filling with the median or mode.
- Dimensionality Reduction - Use FAMD (Factor Analysis of Mixed Data) instead of applying PCA on both numerical and Categorical features.