CapstoneProject: BeterIsmann/Arvato Predicting Campaign Response

AWS Machine Learning Engineer Nanodegree - Jean Carlos da Cruz

Domain Background

BeterIsmann/Arvato is a company that provides financial, IT and supply chain management services. It is a global company focused on automation and analytics. Many global renowned companies from a wide variety of industries, from telecommunications providers, energy, banks, insurance, e-commerce, IT and others make use of Arvato's portfolio of solutions. The company is helping businesses to retrieve insights and make better data drive decisions by using data science and machine learning. In this project, we aim to help a mail-order company that sells organic products in Germany to build a solution that can predict which users are more likely to convert into becoming customers for the company. The existing customer data and the demographic data of the population in Germany are to be studied to understand different customer segments, and then build a system to make predictions on whether a person will be a customer or not based on the demographic data.

Problem Statement

The problem statement can be described as: "Given the demographic data of a person, how can a mail order company convert users into new customers?" In order to answer that question we will first identify segments in the general population and segments inside the existing customers, after that discover what demographic features correspond to a person being a customer for the company. Secondly, we will model a supervised learning algorithm to predict whether a person is a probable customer or not, based on the demographic data.

Datasets and Inputs

There are four data files associated with this project: - 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). Additionally, 2 metadata files have been provided to give attribute information: - DIAS information Levels - Attributes 2017.xlsx : Top level list of attributes and descriptions, organized by informational category; - DIAS Attributes - Values 2017.xlsx : Detailed mapping of data values for each feature in alphabetical order. Each row of the demographics files represents a single person, but also includes information outside of individuals, including information about their household, building, and neighborhood. Use the information from the first two files to figure out how customers ("CUSTOMERS") are similar to or differ from the general population at large ("AZDIAS"), then use your 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 "CUSTOMERS" file contains three extra columns ('CUSTOMER_GROUP', 'ONLINE_PURCHASE', and 'PRODUCT_GROUP'), which provide broad information about the customers depicted in the file. The original "MAILOUT" file included one additional column, "RESPONSE", which indicated whether or not 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 your final predictions will be assessed in the Kaggle competition.

Evaluation Metrics

For the customer segmentation part we will use dimensionality reduction technique (PCA) to reduce the number of dimensions, the selection will be based on the explained variance ratio of each feature, the minimum number of dimensions explaining as much variation as possible, the better. To create segments we will use K-Means, in this case the number of clusters will be a hyperparameter and it will be selected based on the squared error i.e. the distance between all the clusters. For the model prediction part the evaluation metrics will be the standard ones when it comes to classification models, such as: F1 Score and AUROC, because the dataset is highly skewed, the accuracy score will not be a good chose to evaluate the models.

Analysis

EDA

To start our analysis step we loaded and check the data for integrity in its size and shape. The first step in EDA was to address mixed type columns since we got warning when loading the data. The columns 18 and 19 contained mixed features (both numerical and categorical) and mis-recorded values. So, using the

Attribute-values spreadsheet as reference to check what the columns really represent and what can be done. The Columns ‘CAMEO_DEUG_2015’ and ‘CAMEO_INTL_2015’ were addressed and the mis-recorded values (‘X’ and ‘XX’) were replaced with NaN values. The second step was to address unknown values using The ‘Attribute-values’ spreadsheet that contains the information about which columns contain unknown values and how they are entered specified in the dataset. With this information all the unknown values are replaced with NaN values in the dataframes. In total, there were 232 columns which contained unknown representations. The third step was to check the similarity among the data frames and what features are having a description in the given attribute information data.The results were that there are 272 columns between general population and customers data with clear description, 3 columns are only contained in the customers dataframe and 42 columns had no description found. The fourth step was to fix the non-existent values in ‘LP’ columns, these columns give the information about a person’s family status, financial status and the life stage they are in, where these columns contained ‘0’ as a value in the recorded data, which does not correspond to any category specified in the Attribute information data. These ‘0’s have been converted to NaN values. The columns ‘LP_LEBENSPHASE_FEIN’ and ‘LP_LEBENSPHASE_GROB’ had too much granular information packed into them. The FEIN data consisted of fine information about life stage and wealth information. This information has been splitted to represent wealth information as one feature and life stage information as one feature and saved into the same two columns.The columns ‘LP_FAMILIE_FEIN’ and ‘LP_STATUS_FEIN’ have been dropped since they contained duplicate information that the corresponding ‘_GROB’ columns consisted.

To start finishing the EDA step we performed re-encoding on the EINGEFUGT_AM, ANREDE_KZ, CAMEO_INTL_2015, WOHNLAGE and LNR, each column had its own characteristics that needed to be addressed.After re-encoding, we worked on the missing values on the column wise where each column were analyzed. We empirically defined a threshold of 30% was decided after analyzing the percentage of missing values, so, the columns that had more than the threshold were dropped from both dataframes (11 columns in total).

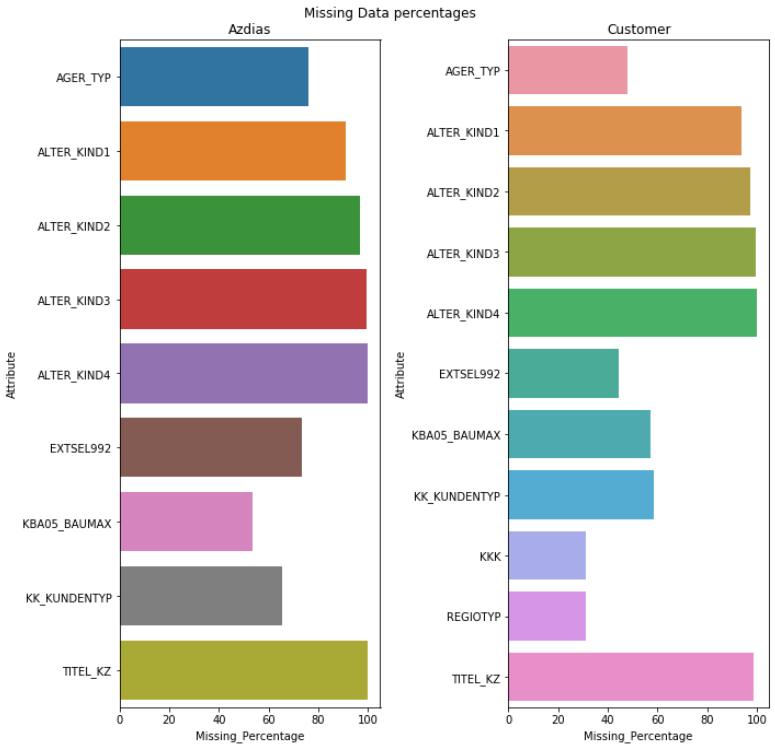


Figure 1: Features with more than 30% of missing values

In the row wise all observations with more than 50 missing features were dropped, that resulted in excluding a total of 1,53,933 rows from the Azdias dataset and a total of 57,406 rows from the the customers dataset.

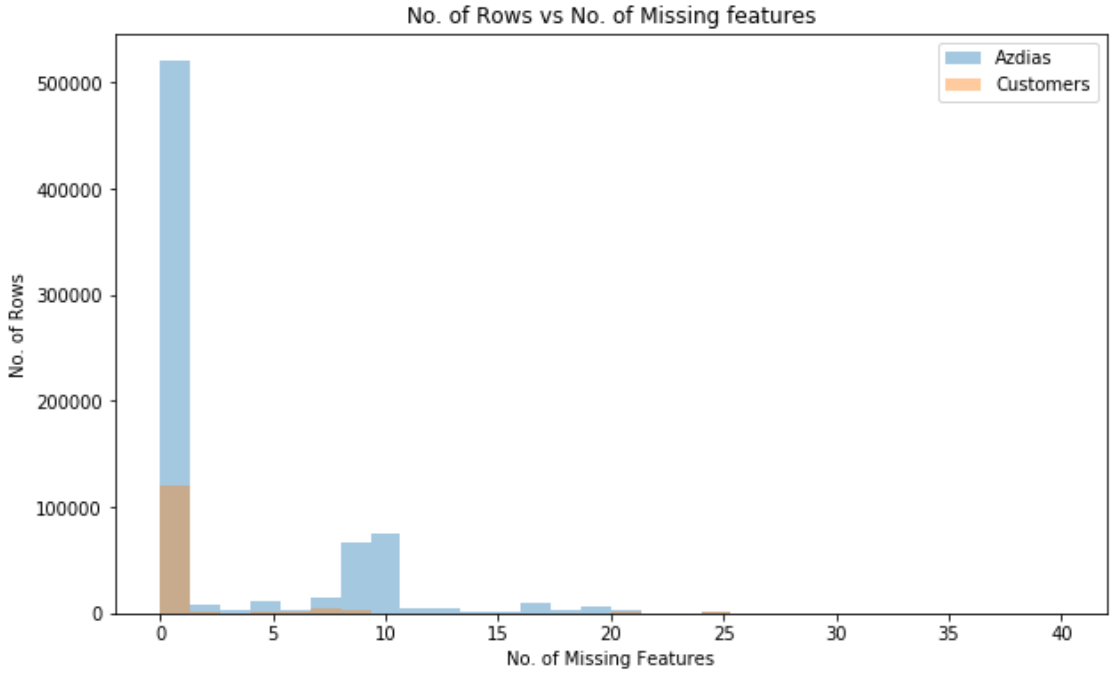


Figure 2: Distribution of missing values

After removing the features and rows which contained missing values based on set thresholds. The data still has some missing values. These missing values

have been replaced with the most frequently occurred observation in each feature. Since the data corresponds to population in general, imputing the missing values with most frequent observations had been selected. And finally, the last step was to use a standard scaler to transform all features into the same range. That was done in order to eliminate feature dominance when applying dimensionality reduction.

Customer Segmentation

The goal of this segmentation part is to create segments of the general population and the customers, in order to compare the results and try to determine future customers. Company's existing customers data was available to understand and compare each feature in the customers data and the general population data, that requires much time spent in analysis since not all features have a significant relevance in defining customer behavior. Not to mention that might exist complex interactions between features resulting in a user being a customer or not.

So, to solve this problem the Principal Component Analysis (PCA) was performed on the given data to reduce the number of dimensions. Since there were 353 features after the EDA step, the results can be seen in the figure 3 where we have 353 features and almost 90% of the variance in the data can be explained with only 150 components of PCA. With this step we will be able to reduce the number of features from 353 to 150, resulting in way less computing resources needed.

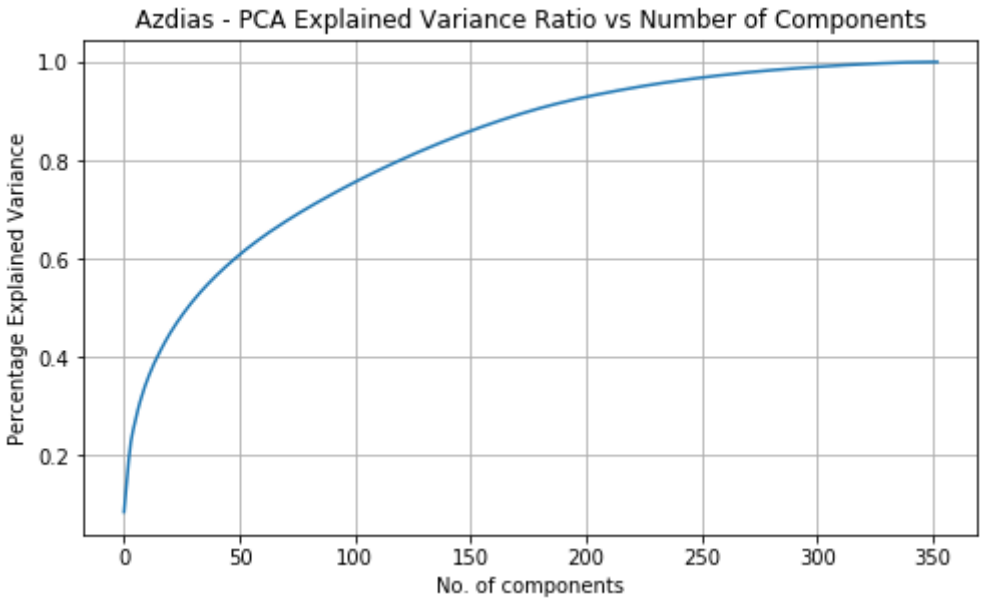
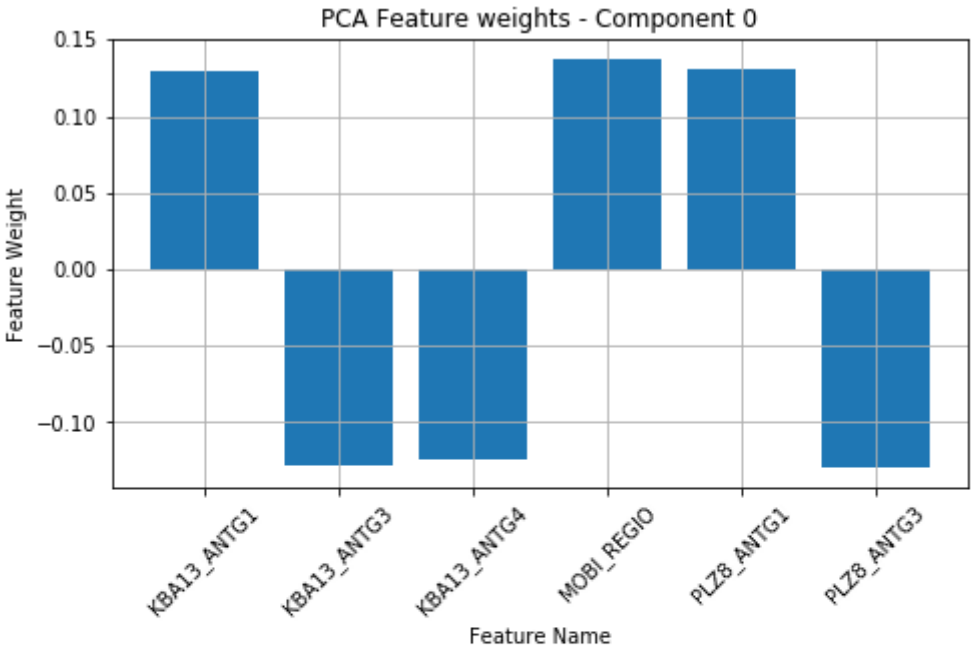


Figure 3: PCA Explained Variance

All the 150 components can be explained by looking at the features weights PCA results considering the original features. As an example, the component 0 results are shown in figure 4.



	Feature	Description	FeatureWeight
2	MOBI_REGIO	moving patterns	0.136978
1	PLZ8_ANTG1	number of 1-2 family houses in the PLZ8	0.130518
0	KBA13_ANTG1	No description given	0.129848
5	KBA13_ANTG4	No description given	-0.124790
4	KBA13_ANTG3	No description given	-0.128478
3	PLZ8_ANTG3	number of 6-10 family houses in the PLZ8	-0.129217

Figure 4: PCA Component 0 Results

The results for component 0 were that it has a high positive weight to moving patterns and 1-2 family houses in the PLZ8 and negative weight on 6-10 family houses in the PLZ8. Interesting KBA13 (which corresponds to share of cars) features with no description given had positive and negative weights

K-Means Clustering

After reducing the number of dimensions, we will now use the K-Means Clustering algorithm to cluster the general population into different segments. We used K-Means because it is simple, it measures the distance between two observations in order to define a cluster. That helped to split the general population into clusters and check for similarities with the customers data. The number of clusters is a hyperparameter when working with clustering algorithms. The basics are to select the number of clusters to minimize the intra-cluster variation, that means, each observation in one cluster is the closest as possible to each other. There is no right recipe to define the optimal number of clusters. The elbow plot in

Figure 5, shows the sum of squared distances in each cluster for the specified list of clusters.

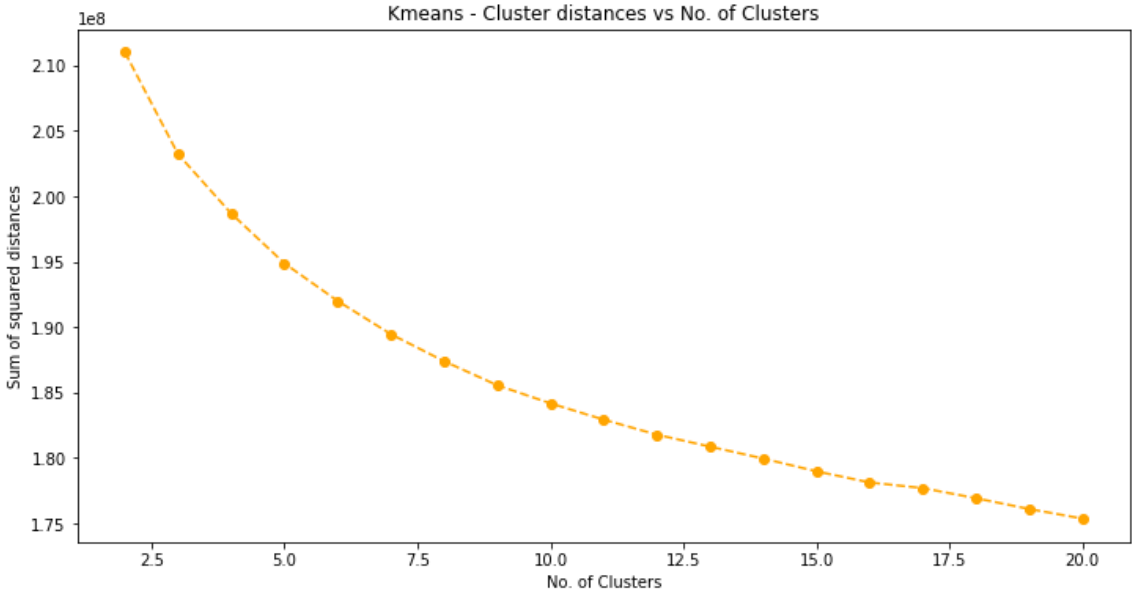


Figure 5: K-Means elbow plot

This plot shows how the number of clusters change de intra-cluster distances. So, the optimal number of clusters can be the number where the sum of squares line smooths. The number of clusters in this analysis were set to 10, since the sum of square distances decreases in a way less rate at this point, so adding one more cluster will not reduce the sum of square distances between the clusters.

Number of clusters

The AZDias and the customer data frames both have been clustered using K-Means, Figure 6 shows the share of population in each cluster.

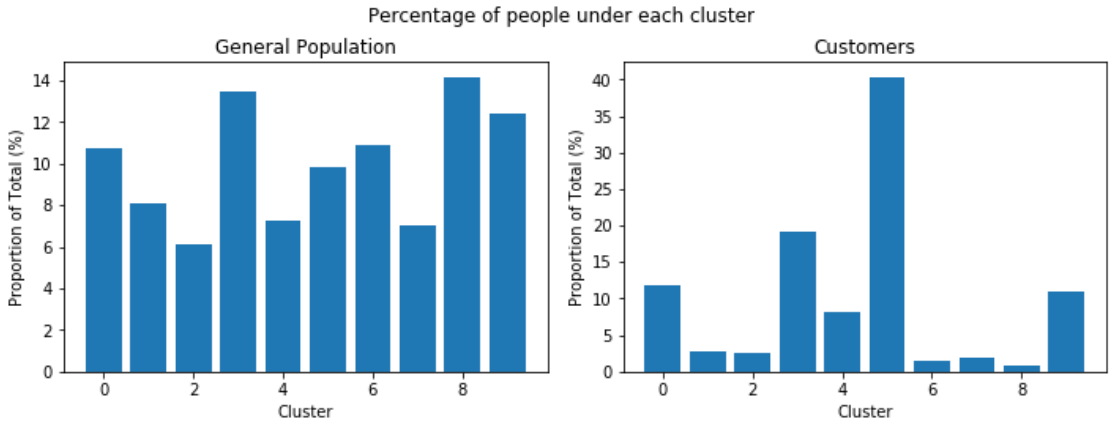


Figure 6: Share of users per cluster in AZDIAS and Customers dataframes

The total number of users have been distributed into clusters segments where the distribution in General Populations is somehow similar between clusters but when it comes to the Customers dataframe it is possible to see that most of the customers are in the cluster number 5,3 and 1, respectively.To confirm this outcome we checked the proportions ratio of customers clusters and general population, the results are shown in Figure 7.

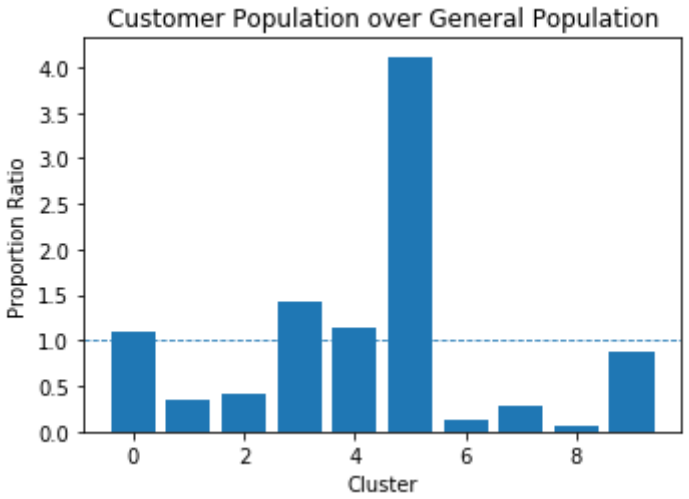


Figure 7: Ratio of customers clusters over general population

Clusters 1,3,4 and 5 have ratios greater than 1, meaning these clusters have more customers in the existing population and have a potential to have more future customers, which is the target of the project.

And if it is needed to describe what components make up each cluster and how each feature represents in each component we can run a simple check on the PCA weights and extract valuable information. An example is shown in Figure 8:

	Component	ComponentWeight	Feature	Description	FeatureWeight
2	0	7.856286	MOBI_REGIO	moving patterns	0.136591
1	0	7.856286	PLZ8_ANTG1	number of 1-2 family houses in the PLZ8	0.129638
0	0	7.856286	KBA13_ANTG1	No description given	0.128969
5	0	7.856286	KBA13_ANTG4	No description given	-0.124159
4	0	7.856286	KBA13_ANTG3	No description given	-0.127560
3	0	7.856286	PLZ8_ANTG3	number of 6-10 family houses in the PLZ8	-0.128308
2	0	7.856286	ONLINE_AFFINITAET	online affinity	0.154768
1	0	7.856286	PRAEGENDE_JUGENDJAHRE	dominating movement in the person's youth (avantgarde or mainstream)	0.143306
0	0	7.856286	D19_GESAMT_ANZ_24	No description given	0.140687
5	0	7.856286	CJT_TYP_5	No description given	-0.134127
4	0	7.856286	D19_GESAMT_ONLINE_DATUM	actuality of the last transaction with the complete file ONLINE	-0.136540
3	0	7.856286	LP_LEBENSPHASE_FEIN	lifestage fine	-0.141184

Figure 8: Cluster 0 - Components described

Predicting Campaign Response

Starting our second part of this project we had used a supervised learning algorithm to predict if a user will probably become a customer or not based on the demographic data that is available. So, in order to train the data we used the Udacity_MAILOUT_052018_TRAIN' file that have 42962 rows with the same columns structure as the previous files with an addition of a extra columns called 'RESPONSE' with is our target field that says if a user became or not a customer. After loading the data, we ran the same steps to clean the dataset as we used for the general and customer data.

Moving further we defined a model to be set as benchmark, the decision was made on the simplest model possible. This was set to comparte the results from more complex models that we trained after. The data was splitted into train and validation and a logistic regression model was trained on unscaled training data and evaluated on the unscaled validation data. So the AUROC score obtained using the logistic regression model was 0.67 and that was our baseline.

After the benchmark result, the train data was worked in order to scale using a standard scaler and again splitted into train and test. Different algorithms were trained on the training data and evaluated on test data. For this project the algorithms that have been used were:

- Logistic Regression;
- Decision Tree Classifier;
- Random Forest Classifier;
- Gradient Boosting Classifier;
- AdaBoost Classifier.

As we can see by the names, all of them are classification models since our problem is a classification problem, using the metrics that we choose earlier the results for each model were compared and can be seen in the Figure 9:

	Model	AUROC_score	Time_in_sec
0	LogisticRegression	0.631793	18.2736
1	DecisionTreeClassifier	0.510325	3.72911
2	RandomForestClassifier	0.530196	1.66718
3	GradientBoostingClassifier	0.749229	48.2555
4	AdaBoostClassifier	0.709368	17.2039

Figure 9: Model performance comparison

The models were trained using the default hyperparameters, comparing the results seen in the Figure 9, GradientBoostingClassifier has the highest score, but it is the slowest to train. Also AdaBoostClassifier have the next highest score (not that far from GradientBoostingClassifier) and also take less time to train.So, considering that the score are similars and AdaBoostClassifier take way less time, AdaBoost was selected for the hyperparameter tuning step.

The hyperparameter step was performed on AdaBoost using Grid Search where we defined a set of hyperparameters (number of estimators, learning rates and the SAMME.R

algorithm) in order to get the best performing model possible, the running time took 55 minutes and gave the score of 0.77 (higher than the 0.67 when using the default parameters) when runned using the learning rate 0.1 and 100 estimators.

After the hyperparameter step we define the features importances for the tuned Adaboost model since the algorithm is a tree-base model. The importances plot can be seen in Figure 10, where it is possible to see that “D19_SOZIALES” has the highest importance to predict campaign response.

Unfortunately, there is no description given in the attribute information files about the “D19_SOZIALES” feature. But making some kind of correlation with other “D19” it seems that the feature has something to do with social transactions, but we can infer for sure.

To finish the project we predicted the best tuned model against a totally new dataset for test purposes, this dataset had the same structure as the previous ones and had 42833 rows. We performed the same cleaning steps as before as well, the result was pretty much the same as using the train data (0.77) which gives us a strong positive result of the usability of the model for new datasets with the same precision.

Results

- Using the PCA and K-means algorithm in order to create cluster of customers was successful, we were able to check where the actual customers of the company intersect the cluster of general segments (Clusters 0,3,4 and 5) which are segments that are more likely to convert on actual customers for the company;
- We were able to predict campaign response using machine learning by training different models using default hyperparameters, getting the best one to tune using GridSearch with a result of 0.77 AUROC_score which is pretty strong.
- With the results of this project the company can be more accurate when working on acquiring new customers both on segmentation and campaign response prediction by addressing marketing budget more accurately aiming the best converting number possible.

References

Arvato-Bertelsmann: https://www.bertelsmann.com/divisions/arvato/#st-1
Sklearn Adaboost docs:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
The 5 Classification Evaluation metrics every Data Scientist must know:
https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226