Capstone Project

Report for Arvato Financial Solutions

By: Mohamed Abed

12th Dec 2022

Project Overview

Description:

Arvato is a services company that provides financial services, Information Technology (IT) services and Supply Chain Management (SCM) solutions for business customers on a global scale. It develops and implements innovative solutions with a focus on automation and data analytics. Arvato's customers come from a wide range of industries such as insurance companies, e-commerce, energy providers, IT and Internet providers. Also, Arvato is wholly owned by Bertelsmann, which is a media, services and education company.

Arvato is helping its customers get valuable insights from data in order to make business decisions. Customer centric marketing is one of the growing fields. Identifying hidden patterns and customer behavior from the data is providing valuable insights for the companies operating in customer centric marketing. Data Science and Machine Learning are immensely used now a days to fulfil business goals and to satisfy customers.

In this project, Arvato is helping a Mail-order company, which sells organic products in Germany, to understand its customers segments in order to identify next probable customers. The existing customer data and the demographic data of population in Germany are to be studied to understand different customer segments, and then building a machine learning model to make predictions on whether a person will respond to emails from the company based on their data.

Datasets & 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

Project Steps:

The project has two major steps: the customer segmentation report, and the supervised learning model.

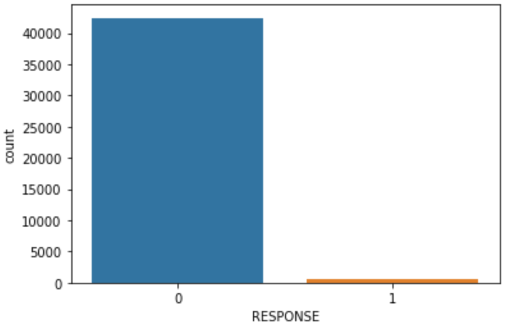
- Customer Segmentation using Unsupervised Learning:
  This part we perform exploratory data analysis and feature engineering steps to prepare the data for further steps. A Principal Component Analysis (PCA) is performed for dimensionality reduction. Then K-Means Clustering is performed on the PCA components to cluster the general population and the customer population into different segments. These clusters are studied to determine what features make a customer with the help of cluster weights and component weights.

- New Customers Acquisition using Supervised Learning, in this part of the project the customers data with defined targets indicating the past responses of the customers has been used to train Supervised Learning algorithms. Then the trained model is used to make predictions on unseen test data to determine whether a person could be a possible customer.

Evaluation Metrics:

For the first step of the project, an unsupervised learning algorithm like K-Means Clustering is proposed, and the number of clusters is selected on the squared mean error (the distance between all the clusters).

The evaluation metric for the supervised learning step of this project is AUC for the ROC curve, relative to the detection of customers from the mail campaign. A ROC, or receiver operating characteristic, is a graphic used to plot the true positive rate (TPR, proportion of actual customers that are labeled as so) against the false positive rate (FPR, proportion of non-customers labeled as customers). Which can be used in cases of labels imbalance as seen in the figure below of our mailout training data.



The line plotted on these axes depicts the performance of an algorithm as we sweep across the entire output value range. We start by accepting no individuals as customers (thus giving a 0.0 TPR and FPR) then gradually increase the threshold for accepting customers until all individuals are accepted (thus giving a 1.0 TPR and FPR). The AUC, or area under the curve, summarizes the performance of the model. If a model does not discriminate between classes at all, its curve should be approximately a diagonal line from (0, 0) to (1, 1), earning a score of 0.5. A model that identifies most of the customers first, before starting to make errors, will see its curve start with a steep upward slope towards the upper-left corner before making a shallow slope towards the upper-right. The maximum score possible is 1.0, if all customers are perfectly captured by the model first. (It should be noted that this particular task is very difficult with a lot of noise, and so you should not expect extremely high scores!)

Data Analysis

Exploratory Data analysis:

Preprocessing:

I. Addressing mixed type columns

When loading the data, A warning appeared for The columns 18 and 19 contained mixed features and some mis-recorded values. The Attribute-values excel sheet was used as a reference to understand what these columns represent and what values can these columns take.

- Affected columns: 'CAMEO_DEUG_2015' & 'CAMEO_INTL_2015'.
- Mis-recorded values: ('X', 'XX'), we replaced them with NaN values in the dataframe.

II. Addressing 'unknown' values

The second step is to replace the unknown representations in all the columns with one representation. The 'DIAS Attributes-values' excel sheet contains the information about which columns contain unknown values and how they are represented in the dataset. With this information we replace all the representations of unknown values with NaN values in the dataframes.

In total, there were 232 columns which contained unknown representations

III. Addressing non-existent values in 'LP_' columns

The columns 'LP_FAMILIE_FEIN', 'LP_FAMILIE_GROB', 'LP_STATUS_FEIN', 'LP_STATUS_GROB', 'LP_LEBENSPHASE_FEIN' and 'LP_LEBENSPHASE_GROB'. These columns give the information about a person's family status, financial status and the life stage they are in.

- 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 were converted to NaN values.
- The 'LP_LEBENSPHASE_FEIN' and 'LP_LEBENSPHASE_GROB' have too much granular information packed into them. The FEIN data consisted fine information about life stage and wealth information. This information has been divided 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.

IV. Imputing Missing Values

After cleaning the data and engineering certain features, I wanted to see the percentage of missing values in our dataset.

The percentage of missing values in each column was visualized in the figure below. The columns which had missing values in customers data also seems to have missing data in the general population data and the distribution of the missing data per column is similar between these two.

- The columns that had more than 30% missing values were dropped from both customers data and general population data. A total of 8 columns have been dropped in this step, the columns that have been dropped are 'ALTER_KIND1', 'ALTER_KIND2', 'ALTER_KIND3', 'ALTER_KIND4', 'EXTSEL992', 'KBA05_BAUMAX', 'KK_KUNDENTYP', 'TITEL_KZ'.
- As for rows with missing values in their features, I decided against removing them because the nature of how missing values for irrelevant columns don’t affect the useful data present and many of the irrelevant data are reduced or dropped anyway.
- Then we go through the process of imputing these missing values with the mean of other values present in each column, since we don’t want to skew the dataset by imputing them with a fixed value or most frequent value, we chose the strategy to be mean value to keep the distribution of our dataset the same.

V. Feature Scaling

The standard scaler from Scikit-learn library is used to bring all the features to the same range. This is done in order to eliminate feature dominance and immensely large values that are considered by pandas as infinite values.

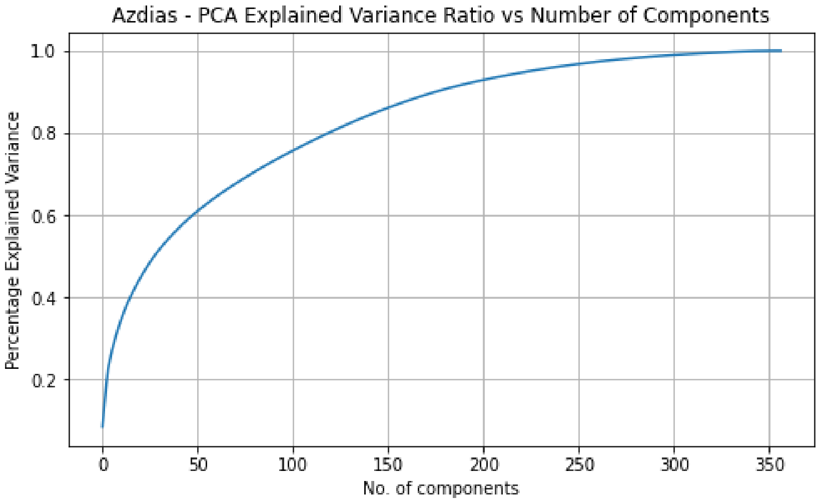
Customer Segmentation:

The aim of the first step of the project is to divide the general population and the customers into different segments, as to compare the general population and customers to determine probable future customers. Here the company’s existing customers data was available to understand and compare each feature in the customers data and the general population data. This requires lot of analysis and this process is time consuming because not all the features will be important in determining the customer behavior. Also, there might exist some complex interactions between these features which resulted in the person being a customer. A hand coded analysis like this would consume a lot of time resulting in no fruitful results.

I. Principal component analysis

For this reason, an approach to segment the customers and general population into different parts using unsupervised learning algorithms was chosen. The Principal Component Analysis (PCA) was performed on the given data to reduce the number of dimensions. Since there were 357 features after the data cleaning step, there is a need to understand which features will be able to explain the variance in the dataset.

This is done with the help of PCA and the resulting explained variance plot is shown in the figure below.



Although we have 357 features, we found that almost 90% of the variance in the data can be explained using around 170 components of PCA.

II. PCA components

These 170 components can be further explained by looking at the feature weights the PCA algorithm has given to the original features. For example, the component 0 explanation is shown in the table below.

The component 0 corresponds to people who have high moving patterns and have a greater number of 1-2 family houses in their neighborhood. Also, these people have a smaller number of houses with 6-10 families. Which shows that these people tend to live in neighborhoods which have small family buildings and not 9 apartments. Other features 'KBA13_*' corresponds to shares of cars.

Component 0		
Feature	Description	Feature Weight
LP_STATUS_FEIN	social status fine	0.129169
MOBI_REGIO	moving patterns	0.126022
KBA13_ANTG1	No description given	0.121915
CAMEO_DEUG_2015	CAMEO classification 2015 - Uppergroup	-0.118559
PLZ8_ANTG3	number of 6-10 family houses in the PLZ8	-0.120843
KBA13_ANTG3	No description given	-0.121457

While another component, component 1 explanation would be people who have an affinity for online interactions, made transactions within the last 24 hours, and these people did a lot of moving/travelling in their teen years.

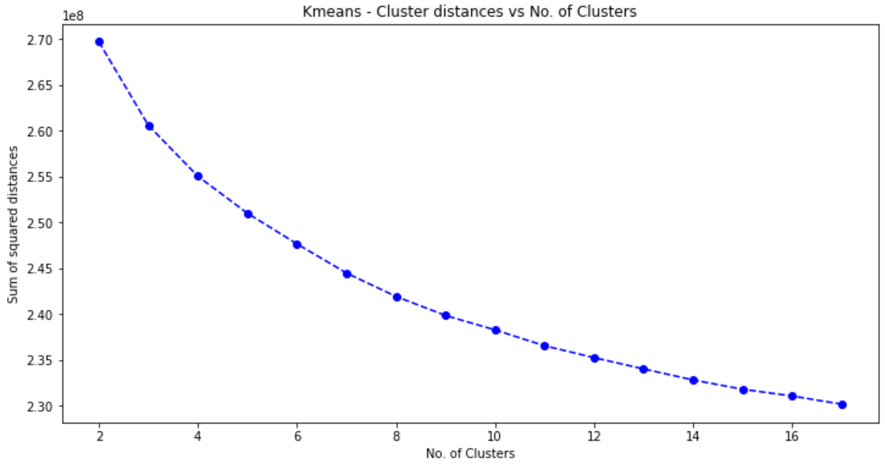
Component 1		
Feature	Description	Feature Weight
ONLINE_AFFINITAET	online affinity	0.151977
PRAEGENDE_JUGENDJAHRE	dominating movement in the person's youth	0.145527
D19_GESAMT_ANZ_24	transaction activity TOTAL POOL in the last 24 hours	0.142673
CJT_TYP_3	No description given	-0.139825
CJT_TYP_4	No description given	-0.141980
CJT_TYP_5	No description given	-0.144396

III. Clustering

After the dimensionality reduction, the next step is to divide the general population and customer population into different segments. K-Means clustering algorithm has been chosen for this task. Since it is simple and is apt for this task, since it measures the distance between two observations to assign a cluster. This algorithm will help us in separating the general population with the help of the reduced features into a specified number of clusters. And use this cluster information to understand the similarities in the general population and customer data.

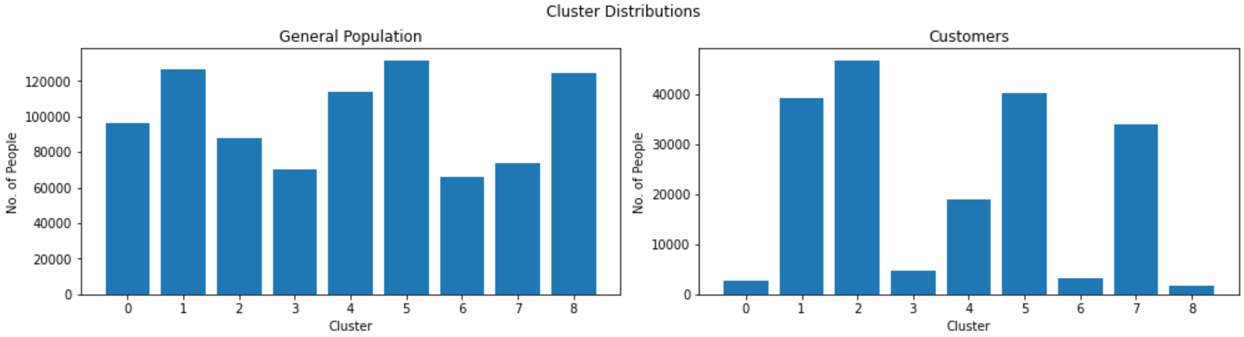
The number of clusters is a hyperparameter when working with clustering algorithms. The basic idea behind the clustering algorithms is to select the number of clusters to minimize the intra-cluster variation. Which means the points in one cluster are as close as possible to each other. There is no definitive way of selecting the number of clusters, we can either intuitively select a specific number of clusters or perform an analysis and then select the number of clusters.

Here, an elbow plot has been used to decide the number of clusters for the K Means algorithm. The elbow plot plots the Sum of Squared distances in each cluster for the specified list of number of clusters



IV. Clusters analysis

The general population and the customer population have been clustered into segments. The figure below represents the proportions of population coming into each cluster. The cluster distribution of the general population is uniform, meaning that the general population has been uniformly clustered into 8 segments. But the customer population seems to be mostly coming from clusters 1, 2, 4, 5, 7.



Each cluster can be explained the same way as we did with PCA components explaining each cluster at a time and what components are present in them.

The customer cluster that has the most percentage of the population is cluster 3 which can be explained by the table below, where people in this cluster correspond to a certain shopping, insurance, and health topology and they owned cars while being between the ages 31 and 45 years old.

Cluster 3				
Component	Component Weight	Feature	Description	Feature Weight
9	3.409547	KOMBIALTER	No description given	0.245940
9	3.409547	SEMIO_REL	affinity indicating in what way the person is ...	0.145954
9	3.409547	CJT_TYP_3	No description given	0.143101
9	3.409547	SHOPPER_TYP	shopping typology	-0.216493
9	3.409547	VERS_TYP	insurance typology	-0.236502
9	3.409547	HEALTH_TYP	health typology	-0.244748
8	3.341706	KOMBIALTER	No description given	0.219636
8	3.341706	KBA13_ALTERHALTER_45	share of car owners between 31 and 45 within t...	0.167184
8	3.341706	KBA13_HALTER_40	share of car owners between 36 and 40 within t...	0.151070
8	3.341706	KBA13_ALTERHALTER_61	share of car owners elder than 61 within the PLZ8	-0.185565
8	3.341706	HEALTH_TYP	health typology	-0.214601
8	3.341706	VERS_TYP	insurance typology	-0.220686

Supervised Learning 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 their 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.

Benchmark Model

The first step in the supervised learning is to set a benchmark, which is the base performance with the simplest model possible. This benchmark is set to compare the results from future steps in order to evaluate the used models. The data is split into train and validation splits and a logistic regression model was trained on unscaled training data and evaluated on the unscaled validation data.

We obtained the score for baseline model – **0.72** (**AUROC** score)

Other Models

This table provides the AUROC score for various models with which we can compare their performance to the benchmark model

Model	AUCROC Score
LogisticRegression	0.702652
DecisionTreeClassifier	0.510095
RandomForestClassifier	0.600276
GradientBoostingClassifier	0.77013
AdaBoostClassifier	0.740869
XGBClassifier	0.686039

As well as using Autgluon tabular predictor to test multiple models that aren't in our list:

Model	score_val	pred_time_val	fit_time
WeightedEnsemble_L2	0.777851	8.189339	163.886536
CatBoost_BAG_L1	0.767140	0.378276	65.714697
LightGBM_BAG_L1	0.762490	0.368800	35.779063
LightGBMXT_BAG_L1	0.758643	0.329957	38.537738
XGBoost_BAG_L1	0.755987	1.660933	25.245924
NeuralNetFastAI_BAG_L1	0.634392	4.021423	106.615764
ExtraTreesEntr_BAG_L1	0.620863	7.10490	13.575696
RandomForestGini_BAG_L1	0.592162	6.973986	17.400885
RandomForestEntr_BAG_L1	0.587040	7.011517	13.889712
ExtraTreesGini_BAG_L1	0.581131	7.059189	11.353179

KNeighborsDist_BAG_L1	0.511265	110.003354	0.536864
KNeighborsUnif_BAG_L1	0.511260	106.618754	0.572390

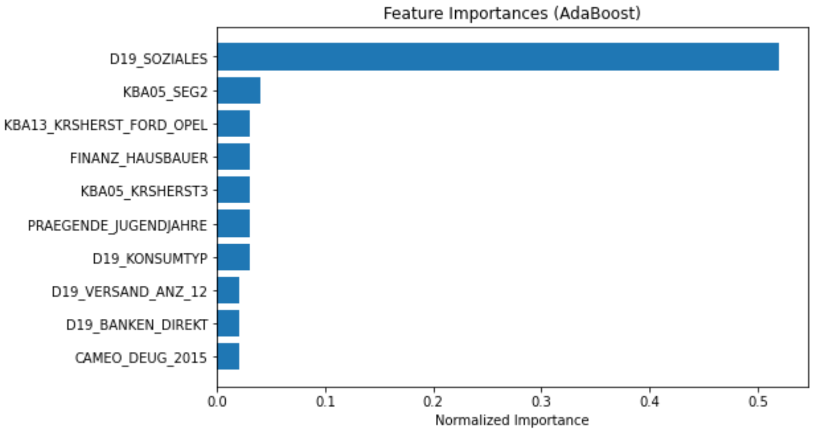
The list of models above and the ones offered by Autogluon are both trained with default hyperparameters, and even though some models offer scores a lot better than our benchmark score, they have worse fit time values/ prediction time values.

So, for these reasons I decided to move forward with AdaBoost classifier and used grid search to find the best performing hyperparameters which resulted in these values:

- *Algorithm: 'SAMME.R'*
- *Learning_rate: 0.1*
- *n_estimators: 100*

Which gave us an AUROC score of **0.7722**

And after analyzing the features importance in our tuned model we found 'D19_SOZIALES' having the highest importance value followed by other features in our dataset as shown in the figure below.



Testing the data

The final predictions were made on the test data which was provided in the file 'Udacity_MAILOUT_052018_TEST.csv'. The same pre-processing steps were performed to clean the data. This data was scaled with the scaler which was fit on the training data

And we saved the prediction output to a csv file names 'Submission.csv'

References

[1] Arvato-Bertelsmann, "Arvato," Bertelsmann, [Online]. Available: https://www.bertelsmann.com/divisions/arvato/#st-1.

[2] Bertelsmann, "Company," Bertelsmann, [Online]. Available: https://www.bertelsmann.com/company/.

[3] S. M. Lador, "What metrics should be used for evaluating a model on an imbalanced data set? (precision + recall or ROC=TPR+FPR)," Towards Data Science, 2017. [Online]. Available: https://towardsdatascience.com/what-metrics-should-we-use-on-imbalanced-data-set-precision-recall-roce2e79252aeba.

[4] A. Kassambara, "Determining the Optimal Number of Clusters: 3 Must Know Methods," DataNovia, [Online]. Available: https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters3-must-know-methods/.