# Customer Segmentation – Arvato Financial Solutions

Edward Amankwah

November 5, 2020

Capstone Report

Udacity Machine Learning Engineer Nanodegree

Table of Contents

# Definition

## Project Overview

### Project Domain Background

Bertelsmann is a leading publishing house since 1835 (Wikipedia, 2020), which over the years has expanded from media, services and education (Bertelsmann, 2020) into software and hardware distribution especially in the 1980s (Computerwoche, 1983). The company obtained its current name, Arvato Bertelsmann in 1999 and migrated more into high tech, information technology, and e-commerce services in 2012 (Wikipedia, 2020). Some of the services they offer include supply chain, financial solutions, e-commerce, insurance companies, energy providers, IT consulting, Information technology and Internet providers (Arvato-Bertelsmann, 2020).

Arvato is using artificial intelligence to derive data driven insights for their customers to implement business decisions. Customer services such as financial solutions, portfolio risk management, payment processing and customer targeting for marketing campaigns benefit immensely from artificial intelligence derived solutions. It is in this field that this project will be developed, as machine learning and data science practices are immensely used to attain business goals and fulfil customer satisfaction.

Since customer centric marketing is a key component for modern companies to thrive, this project will focus on using available data from Arvato to help a mail-order company selling organic products in Germany, to better understand its customer purchasing decisions. The available customers will be segmented to help identify probable future customers that should be targeted for marketing campaigns. The project explores the existing customer and demographic data of population in Germany, which is crucial to understand the behavior of the different customer segments, and then build a trained model to predict who will be a potential future customer based on the demographic data.

### Problem Statement

The problem statement for this capstone project is “How can a mail-order client selling organic products efficiently attract new customers, given the demographic information of a person?” My approach to this challenge includes firstly, I will compare the customer segments derived from both the demographic data of the current customers and the general population data by performing an unsupervised machine learning modelling. Secondly, I will conduct a supervised learning modelling based on the resultant customer segments using a dataset with demographic features that corresponds to a client being a customer for marketing campaign. Lastly, I will predict probable individuals that are good candidates to be converted to the company customers

## Dataset and Inputs

The following datasets were provided by Arvato for the sole purpose of the Udacity Machine Engineer Nanodegree, capstone project topic on customer segmentation and targeted marketing campaign.

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).

Moreover, two metadata files have been provided to give 2017 attribute information:

- DIAS Information Levels - Attributes 2017.xlsx: top-level list of attributes and descriptions, organized by informational category and
- DIAS Attributes - Values 2017.xlsx: detailed mapping of data values for each feature in alphabetical order.

The four demographic comma separated values (CSV) data files contain rows that represent the demographics of a single individual. Additionally, each row contains extra information about their household, building and neighborhood. The Customers data includes three extra columns containing specific mail order company for each individual. Also, train and test data have been provided for the evaluation of the supervised learning models.

## Metric

The main bulk of the analysis is divided into two parts:

### Part 1: Customer Segmentation using unsupervised learning algorithm

The linear PCA technique was employed to reduce the number of dimensions in the dataset. The explained variance ratio was used to determine the important features with the most variance that can be retained for the modelling exercise. In deciding the number of clusters to group customers, an unsupervised K-Means algorithm was used, and the distance metric between data points and nearest cluster center was based on squared errors. The Elbow method was used to determine the right number of clusters.

### Part 2: Customer conversion using supervised learning algorithms

A training dataset (MAILOUT_052018_TRAIN.csv) with response variable containing a binary class was provided for this section of the project. The task is therefore a supervised classification

problem to predict whether the mail-order company should target a particular potential customer or not for marketing campaigns. Several supervised algorithms were used to accomplish the data classification. This dataset was split into train and validation datasets for model training and evaluation, respectively. The evaluation metrics for the binary classification included Accuracy, classification report and Area Under the Receiver Operating Characteristic Curve (ROC AUC) score. The figure below shows the data class balance or distribution.

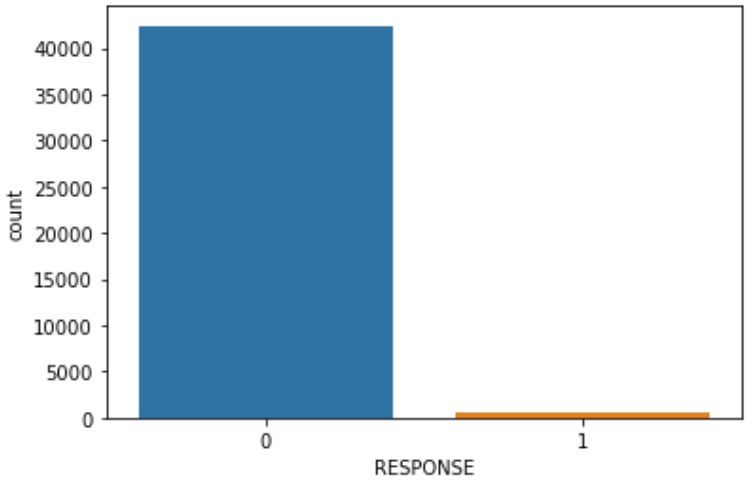


*Figure 1: Response Class Imbalance*

The figure above indicated that out of the 42,430 mailouts, there were 98.76 % negative responses with class label "0" and 1.24% positive responses with class label "1". Initially in the supervised modelling process, the dataset will be balanced using resampling techniques with Accuracy being used as the main metric. However, the main challenge is the ability to target potential future customer (minority class) for marketing campaigns as opposed to general mail out to the entire population of Germany. The Area Under the Receiver Operating Characteristic Curve (ROC AUC), which is indifference to imbalance data was finally used on the imbalanced training dataset to compare different models. The AUC also has the ability to distinguish between the minority and majority classes based on inputs without selecting a threshold.

Moreover, the AUC score the required evaluation metric for the associated Kaggle competition.

**Analysis**

**Data Exploration and Visualization**

An exploratory data analysis (EDA) was performed on the general population (Azdias) dataset to find some potential issues that need to be addressed before any modeling activity. A cleaning function was developed to help simply the data wrangling process of the given customer dataset. Below is a summary of how the features sets (independent variables) were cleaned:

**Data Profiling**

The following profiling were conducted:
Summary statistics were obtained to understand the range and mean of the numerical features. The object types, dimensions of the dataset and non-null values were all checked to ensure data integrity.

**Warnings, Mixing types and Extra Feature Columns**

A mixed type noise appeared at the beginning, during the loading of the datasets (Azdias and customers) on columns 18 and 19. A cursory look at the provided Attribute-values excel sheet indicated that the warnings were due to mixed features (CAMEO_DEUG_2015 and CAMEO_INTL_2015) and mis-recorded values. The mix-recorded values ('X' and 'XX') were later replaced with NAN (not a number)s in a dataframe.

Using Pandas head() method, it was discovered that the customers data set had three extra columns that were absent in the general dataset (Azdias). The three extra columns, including PRODUCT_GROUP, CUSTOMER_GROUP and ONLINE_PURCHASE were dropped in order to have consistent dataset. Also, all extra duplicated columns (Unnamed: 0) were dropped.

**Verifying Common Features**

Based on the analysis of the datasets and the attribute-values, 272 features were found to be common among the general population, costumers and attribute information data with descriptions. Forty-two features were found to be specific to attribute information data with no descriptions.

**Missing Values**

Using Pandas missing value function (isna().sum().sum()), it was discovered that a total of 7,930,552 and 850,610 missing values were present in the general population and customer dataset, respectively.

**I. Column Missing Values**

The analysis and distribution of missing data in each column in the general population and customer datasets appeared to be similar as indicated in figure 2 below:

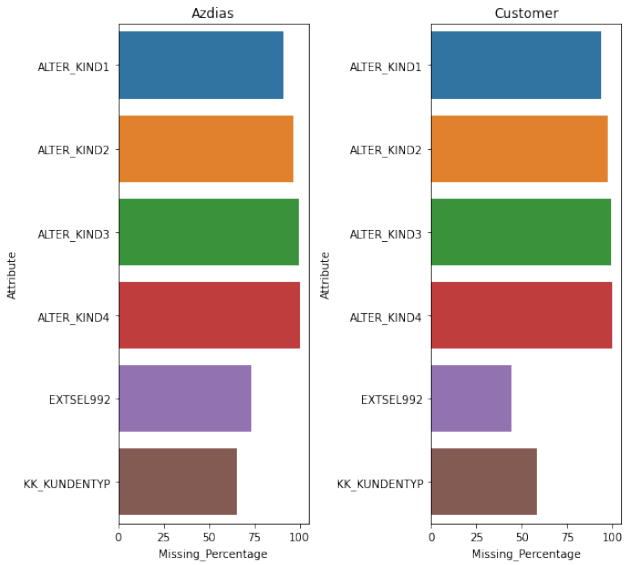


*Figure 2: General population (*Azdias) vs Customer columns missing more than 30% Values

As a general rule, columns with more than 30% missing values were dropped. This reduced the number of features in both the general population and customers date by 6 columns.

## II. Row Missing Values

As a general rule, all observations with more than 50 missing row values were dropped. The figure 3 below indicates the distribution of the row missing observations.

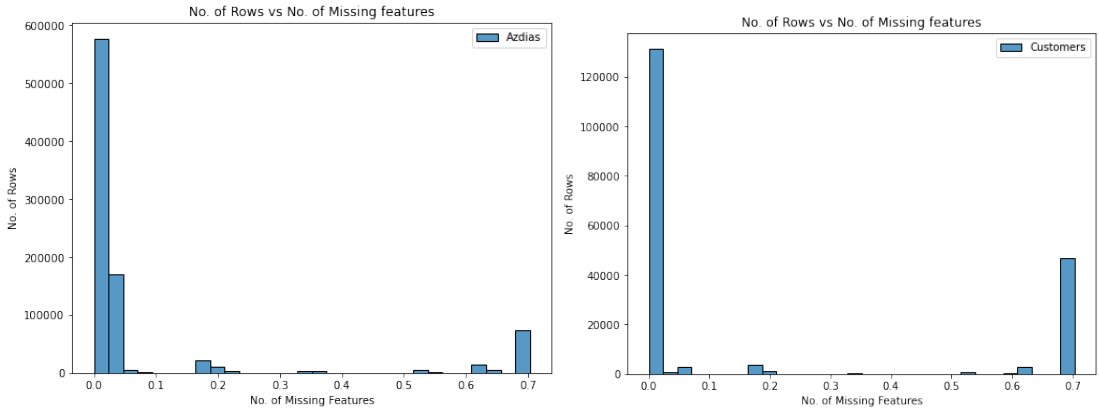


*Figure3: Distribution of Missing Row Values*

A total of 99,194 (891,221 - 792,027) and 50,749 (191,652 - 140, 903) observations were dropped from the general population and customers dataset, respectively.

**Unknown Values**

Based on the analysis of the datasets and the attribute-values, total of 232 unknown values, usually represented as '-1, 0 or 9' were recoded as NANs in the dataframe

**Feature Encoding and Engineering**

The features below were encoded as follows:

- EINGEFUEGT_AM is a date-time related feature showing the date an entry was made. The year part was extracted to represent the feature.
- WOHNLAGE gives the quality of the neighborhood area ranging from poor to very good. About 5382 rows containing mix values were coded as NANs in the dataframe.
- LP* columns describe an individual's family, life and financial status. There were six LP columns; LP_FAMILIE_FEIN, LP_FAMILIE_GROB, LP_STATUS_FEIN, 'LP_STATUS_GROB, 'LP_LEBENSPHASE_FEIN and LP_LEBENSPHASE_GROB representing small or large family, life and status. In order to reduce the level of granularity in the feature sets, the large (grob) feature columns were dropped.
- LP_FAMILIE_FEIN column was encoded into five family levels (single, couple, single parent, family and multifamily)
- LP_STATUS_FEIN column was encoded into five income status levels (low income, mid income, independent income, owner and top earner.
- LP_LEBENSPHASE_FEIN was subdivided into age level and wealth level columns. The age level was encoded into four age sublevels (younger age, middle age, advanced age, and retirement age). The wealth column was encoded into 4 sub wealth levels (low, middle, wealthy and top.
- CAMEO_INTL_2015 column was encoded into two columns; family and wealth status according to international standards on wealth and family.
- ANREDE_KZ column describing the lifestyle characteristics was reencoded as gender, columns with values 0 and 1 representing male and female, respectively.
- Ost_West_KZ column which had binary labels ('W', 'O') were relabeled as 0 and 1, respectively.
- CAMEO_DEU_2015 contained the groupings an individual may belong but was dropped because of the multilevel nature of the data, which can increase significantly the number of features if included.
- D19_LETZTER_KAUF_BRANCHE contained information about the latest branch a purchase was made and was also dropped due to similar multilevel nature as the CAMEO_DEU_2015 feature.

**Imputing NAN Values**

All other missing values encoded as NAN values were replaced with the most frequent values in each feature for the general population dataset.

A data wrangling function (clean_func) which combines all the data clean steps above was developed to simply the wrangling of the customer dataset.

**Feature Scaling**

After dropping the LNR column which uniquely identifies each individual for later analysis, the range in the general population dataset was plotted as shown below:

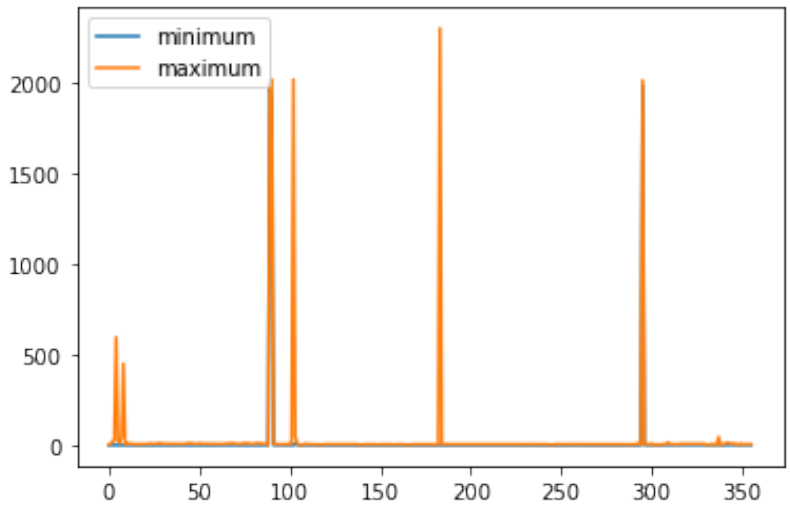


*Figure4: Distribution of Missing Row Values*

The features were scaled to ensure that all the features can have similar weight inputs during data modelling and dimensionality reduction.

**Methodology**

**Part 1: Customer Segmentation Report**

Customer segmentation is the process of breaking customers into groups of common characteristics so that each sub group could be targeted for specific purposes such as marketing or advertising campaigns. This section aims at segmenting the customers and the general population datasets into groups for comparison purposes. Groupings in the general population dataset that are similar in characteristics to the customer dataset can then be targeted as potential future customers. This will save money and also help the mailing company to avoid targeting the entire population for future campaigns. The main algorithms employed in this exercise includes principal component analysis and Kmeans clustering.

**Dimensionality Reduction**

The Principal Component Analysis (PCA), a linear technique was used to trim down a number of the features that may not contribute much to explain the variability in the dataset.
The variance explained by the number of available features after the data cleaning is shown in the figure 5 below:

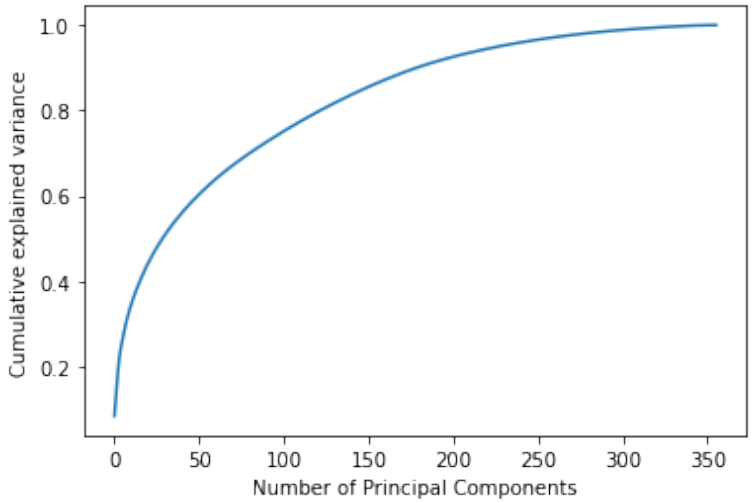


*Figure 5: The Variance Graph*

From the plot, it can be seen that the first 200 principal components explain more than 90% of the variance. Based on this graph, it was decided that 200 principle components would be used for model fitting without losing more than 10% of the variability in the dataset.

**Kmeans Clustering**

Kmeans, a clustering algorithm was used to group the similarities in both the general population and the customer dataset into a predetermined number of clusters. These groupings (clusters) are selected by minimizing the intra-cluster variations and maximizing the inter cluster distances. The number of clusters changes every time the clustering algorithm is run. The elbow method was used to optimally define the right number of clusters for the datasets. This method assesses the closeness of clusters, the objective being to minimize a value known as inertia. The inertia determines how far from each other or how close to each other they are for a group of data points. The figure 6 below indicate how the number of clusters varies with the inertia values.

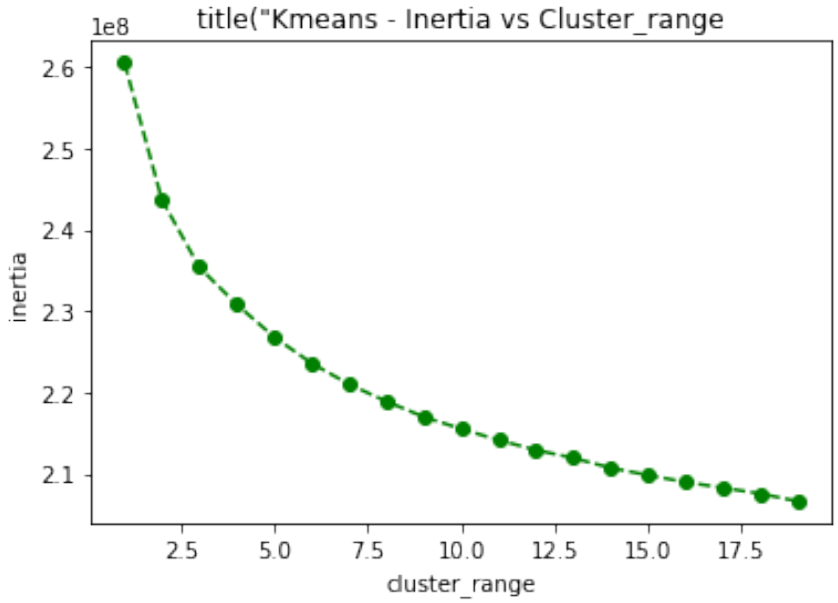


*Figure 6: Plotting the Elbow Method*

The is no outright inflection point (the elbow) where the plot changes sharply. However, the variations or the sum of squares of the inertia starts to decrease at a very low rate after cluster number 8. Therefore, 8 was chosen as the optimal number of clusters.

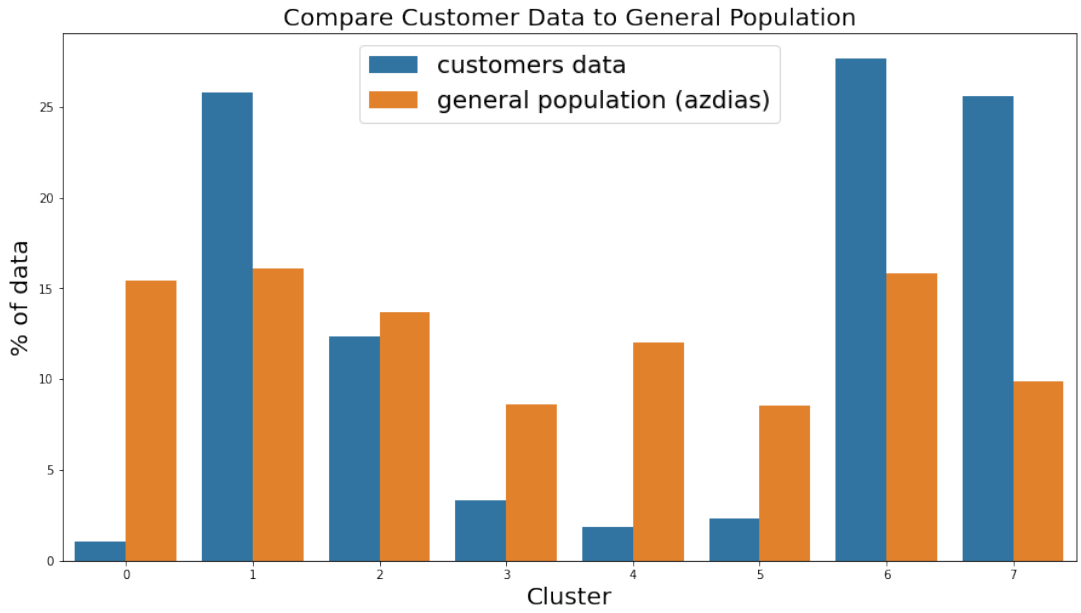


*Figure 7: Distribution of general population and customers across clusters*

The figure 7 above indicate that the general population data is fairly distributed evenly among the 8 clusters. However, the customers data is more grouped in cluster numbers 1, 6 and 7 and are more likely to be customers of the mail-order company.

## Part 2: Supervised Learning Model

In this section, supervised learning methods were used to predict against a response variable indicating whether an individual will be a customer or not based on the demographic dataset. The "MAILOUT" data files represent individuals that were targeted for a mailout campaign. Almost half of the MAILOUT data, including an extra target column, "RESPONSE", were used for training the supervised models. The other half of the MAILOUT data were used to test the predictive performance of the models. Similar cleaning and scaling functions used to clean the general population and customers dataset were used to clean and scale the "mailout-train" and "mailout_test" sets.

### Baseline Modelling

The mailout_train was further split into train and validation sets by 70 to 30 ratio. In order to find an optimal model for predicting the potential customers, a simple logistic regression with default values was fitted on the training data and predicted on the validation data to provide benchmark metrics. Logistic regression predicts the probability of occurrence of an event by fitting data to a logistic function to predict a binary outcome. The classification report (see Jupyter notebook for details) indicated an accuracy of 99 % and 0 % for the negative (majority) and positive (minority) responses, respectively. The right metric would have been recall, which indicates the ability of the classifier to correctly identify the respective classes. However, the recall result was 100 % and 0 % for the negative class and positive class, respectively indicating a bias toward the majority class. The actual class proportions in the training dataset were 98.7 % and 1.27 % for the majority (0) and minority classes, respectively. The recall results prompted the lookout for better strategies to mitigate such biases. The AUC which indicates the probability of distinguishing between the negative class from the positive class was computed to be 0.34

### Balanced Dataset Modelling

Resampling techniques were adopted to make the dataset evenly balanced. This involved taking samples from the available dataset to create new dataset in order to make the new dataset balanced. Model default parameters with a threshold value of 0.5, which determines whether a class will be categorized as positive (1) or negative (0) were used. The following techniques were implemented:

- Randomly under sampling the majority class (response "0") to make the dataset balanced. This reduces the size of the dataset and may have adverse effect on the predictive power of the model.
- Synthetic Minority Oversampling Technique (SMOTE) to oversample the minority class. New synthetic data points were generated from the neighborhood of the minority class to augment the minority class, thereby balancing the dataset.

- Modified Synthetic Minority Oversampling Technique (MSMOTE) to oversample the minority class. New groups of synthetic data points known as security samples, border samples and latent noise samples were generated from the neighborhood of the minority class to augment the minority class, thereby balancing the dataset.

The table 1 below summarizes the results of the metrics:

*Table 1: Logistic Regression Classification Report*

| Sampling Technique | Accuracy | Precision | Recall | AUC |
|---|---|---|---|---|
| | | | **Negative Class (0)** | |
| Baseline | 0.99 | 0.99 | 1.00 | 0.34 |
| Undersampling | 0.64 | 0.99 | 0.64 | 0.36 |
| SMOTE | 0.80 | 0.99 | 0.80 | 0.34 |
| MSMOTE | 0.96 | 0.99 | 0.97 | 0.36 |
| | | | **Positive Class (1)** | |
| Baseline | 0.99 | 0.00 | 0.00 | 0.34 |
| Undersampling | 0.64 | 0.02 | 0.53 | 0.36 |
| SMOTE | 0.80 | 0.02 | 0.33 | 0.34 |
| MSMOTE | 0.96 | 0.03 | 0.07 | 0.36 |

Under the current business scenario, the best sampling method which has the highest ability to classify the minority class is the undersampling technique with a recall of 0.53 and an AUC of 36%. These results are unsatisfactory because an intuitive guest trial method would expect an AUC of about 50%. Therefore, new learning algorithms capable of performing classification task with hyperparameter tuned techniques were employed.

**Supervised Modelling with Full Feature sets and PCA Reduced Data**

The following algorithms were used:

- Support vector machine (SVM) - analyzes data and recognize patterns, and that can be used for both classification and regression tasks. It simply generates hyperplanes or simply putting lines, to separate and classify the data in some feature space into different regions.
- Random Forest (rf) - an ensemble approach to finding the decision tree that best fits the training data by creating many decision trees and then determining the “average” one.
- Gradient Boosting (gb) – like the rf, it makes predictions by forming an ensemble of weak prediction models on decision trees. It uses the loss function of the base model to minimize the error of the overall model.

- Ada Boosting (ab) - the algorithm fits a sequence of weak learners on different weighted training data. Being an iterative process, it continues to add learner(s) until a limit is reached in the number of models or accuracy and
- Extreme Gradient Boosting (xgb) - it mostly combines a huge number of regression trees with a small learning rate to optimize boosting tree algorithms. Trees added early are significant and trees added late are unimportant. Xgb is able to utilize more computational power and get more accurate prediction and it is slow due to the randomness introduced during training.

The algorithms were implemented on the training dataset and prediction was done on the validation set to find the future potential customers. At this stage the only matric used was AUC store and the figure below summarizes the results.

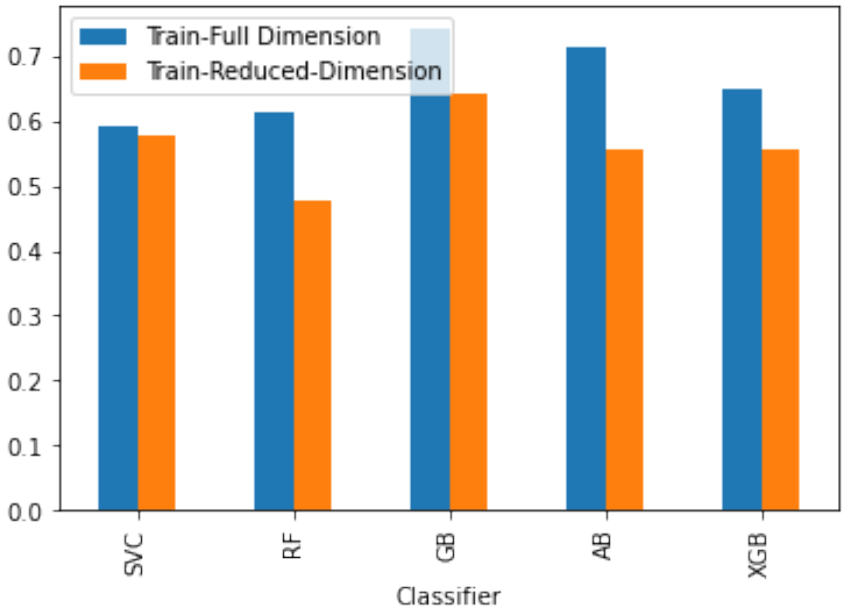


*Figure 8: Distribution of AUC score for Full dimension Vs PCA reduce dimensions*

The models using the full feature sets performed better than the pca reduced dataset models. In the next step, the models were tuned to optimize performance on the validation sets. In general, tuning models take a lot of time and hence the top three models; gradient boosting classifier, Ada boosting classifier and extreme gradient boosting classifier were further analyzed using the full training dataset.

**Model Selection: Random Search versus Automated Hyperparameter Tuning**

In this section, the random search and hyperopt automated techniques were used to tuned the selected models. The search iterations were all limited by time and computer power resources.

- Random search - a set up a hyperparameter were randomly selected over combination of values, fitted on the trained data and scored on the validation data.

- Automated hyperopt tuning - it does optimization using a version of Bayesian Optimization with the Tree Parzen Estimator (TPE). Bayesian optimization finds the value that minimizes an objective function by building a probability model based on past evaluation results of the objective (Dewancker, McCourt & Clark, 2016). The TPE is a sequential model-based optimization (SMBO) method where models are constructed to approximate the performance of hyperparameters based on historical measurements, and then subsequently choose new hyperparameters to test based on this TPE model.

**Results**

A summary of the AUC score is shown in figure 9 below:

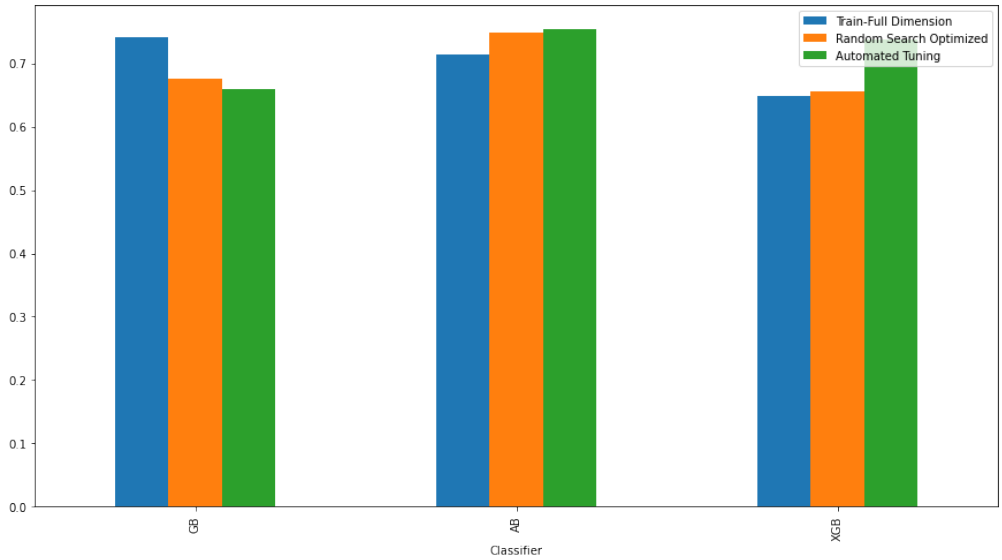


*Figure 9: Distribution of AUC score for Full dimension Vs PCA reduce dimensions*

In general, tuned or optimized models performs better than untuned models. The best performing model; automated Ada boosting and extreme gradient boosting with AUC scores of 0.75 and 0.74, respectively were selected for further prediction on the unseen mailout_test. The final AUC scores on the test data were submitted for Kaggle competition.

**Ada Boosting Feature Importance**

The figure 10 below indicate the order of importance of the features in predicting whether or not it will be worth to include an individual in the marketing campaign.

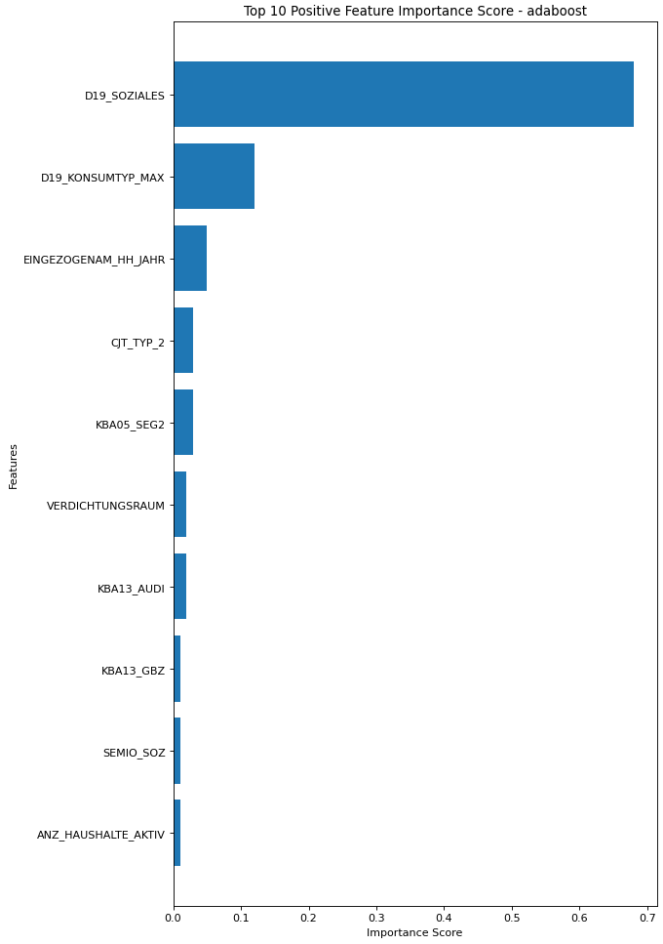


*Figure 10: Adaboost Feature Importance*

The feature importance plot shows that "D19_SOZIALES" is the most important predictor of whether or not it will be worth to include an individual in the marketing campaign.

### Extreme Gradient Boosting Feature Importance

The figure 11 below indicate the order of importance of the features for the xgb model:

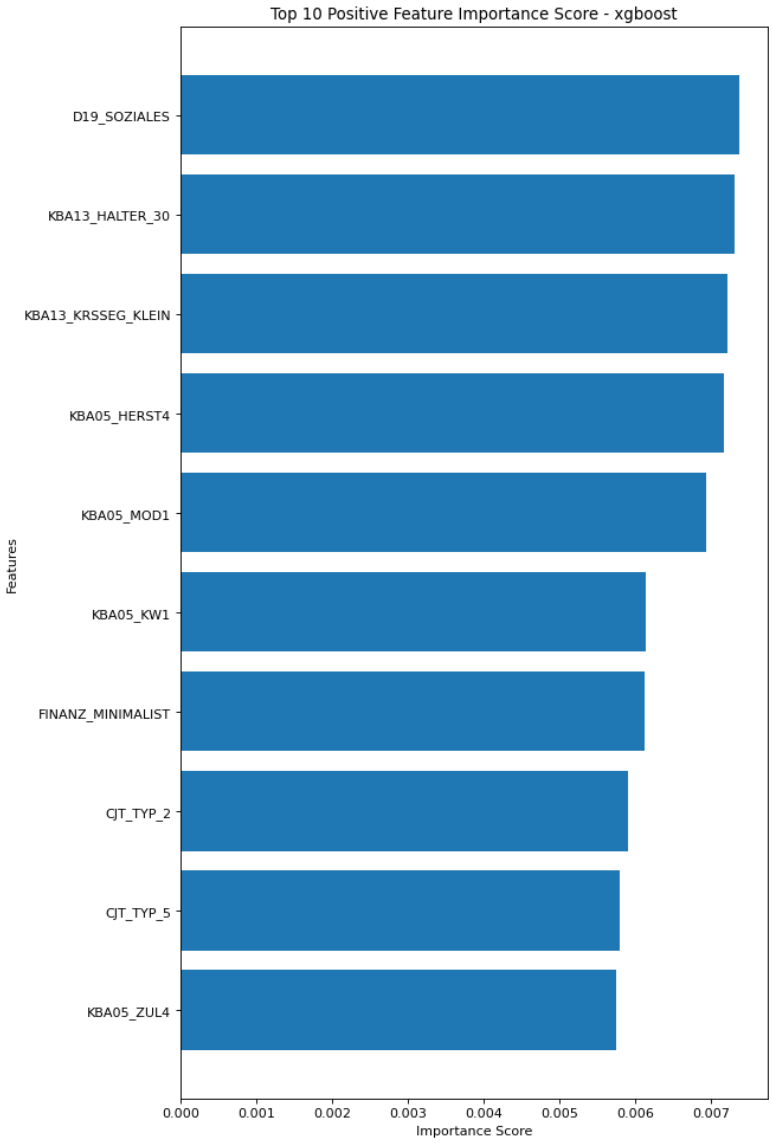


*Figure 11: Extreme Gradient Boosting Feature Importance*

Again, the feature importance plot above shows that "D19_SOZIALES" is the most important predictor of whether or not it will be worth to include an individual in the marketing campaign.

**Prediction on Test Data**

The best two models (Adaboost and XGBoost) were used to make predictions on the provided test dataset, 'Udacity_MAILOUT_052018_TEST.csv' which were held out for the Kaggle competition. The mailout_test data was cleaned and scaled using similar cleaning and scaling function like the mailout_train data.

**Kaggle Competition**

My position on Kaggle leadership board after submitting my predictions on the mailout_test is shown on figure 12 below:

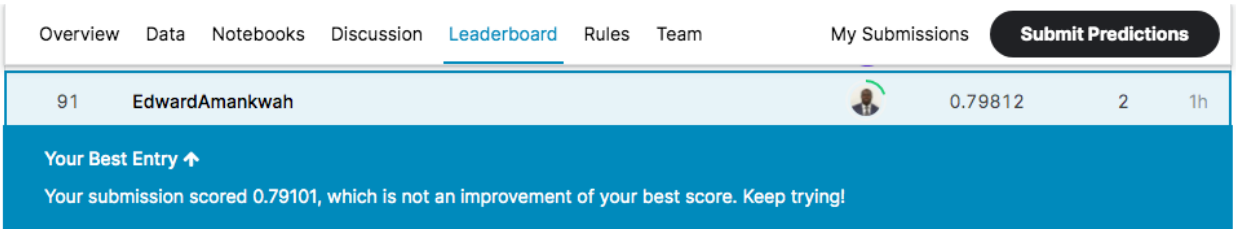


*Figure 12: Kaggle Leadership Score*

The performance of the Adaboost model using the AUC score was 0.79812 leaving more room for improvement of the feature engineering, hyperparameter tuning and model learning iterations.

**Conclusion**

In this project, I analyzed demographics data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population.

The first section entails using an unsupervised Kmeans clustering algorithm to determine which parts of the population are more likely to be customers of the mail order company. The results indicated that clusters 1, 6 and 7 which constitute 42% general population can be target as customers. The second section included the development of several supervised machine learning algorithms to predict whether or not it will be worth to include an individual in Germany in a mail order campaign. The provided training data was divided into train and validation sets; a baseline logistic regression model was fitted on the train set and evaluated on the validation set. The baseline probability that an individual responding to the mail order campaign was predicted with an area under the receiver operating curve score of 34%.

Out of the three sampling techniques that were implement, the AUC score marginally increased by 2% above the baseline model. Several models were tuned using random search and automated hyperopt optimization techniques which resulted in a significant improvement in the AUC score. The best two models, Adaboost and XGBoost classifiers scoring an AUC values of 0.75 and 0.74, respectively were finally used to make predictions on the provided test dataset. The best models were submitted for Kaggle competition yielding and AUC score of 0.79812 on the leadership board.

**Reflection**

The process used for this project can be summarized using the following steps:

I. An initial problem and relevant, private datasets were provided for the general population (Azdias) and customers (company customers).
II. The datasets were wrangled by fixing missing values, unknown values and data imputation.
III. A cleaning function was developed to simply the data cleaning process.

IV. The datasets were exploited, visualized where possible and preprocessed via feature scaling and feature engineering.
V. A linear principal component analysis was performed to reduce the data dimensionality
VI. The datasets were clustered into segments.
VII. A benchmark and resampling modelling techniques were created for the binary classifier.
VIII. The classifiers were trained using the full feature sets as well as the pca reduced datasets.
IX. The hyperparameters of the classifiers were tuned until a good set of parameters were obtained.
X. The best two trained models were used to make predictions on the provided unseen test dataset.
XI. The predictions on the test sets were submitted on Kaggle competition board

The most difficult and interesting part is found in steps III and IX where a lot of time was needed to ensure the datasets are ready for modelling. Hyperparameter tuning takes a lot of computing resources and time and it may need adjustments or restart after a long wait for iteration convergence.

**Improvement**

Good strategies are needed when dealing with datasets that are imbalance. The datasets were balanced using sampling techniques with the default threshold value of 0.5, which determines whether a class will be categorized as positive (1) or negative (0). A better threshold needs to be investigated so that the underlying models can be greatly improved.

Most of the model parameters were either partially tuned or not tuned as this would require higher computer time. A Python code was used to determine my hardware and system information in Python (Pythoncode, 2020).

A higher scale of random search needs to be conducted. The only hyperopt auto tuning method considered was the Tree-structured Parzen Estimator (TPE). Other hyperopt tuning models such as Simulated Anneal should be implement to see how it compares with the TPE model.

**References**

Bertelsmann (2020). Arvato-Bertelsmann. Retrieved from https://www.bertelsmann.com/divisions/arvato/#st-1

Bertelsmann (2020). Bertelsmann Company. Retrieved from https://www.bertelsmann.com/company/

Computerwoche (1983). Bertelsmann vertreibt Rechner von TI. Retrieved from https://www.computerwoche.de/a/bertelsmann-vertreibt-rechner-von-ti,1180755

Wikipedia (2020). Arvato. Retrieved from https://en.wikipedia.org/wiki/Arvato

Dewancker, I, McCourt, M & Clark, S (2016). Bayesian Optimization for Machine Learning A Practical Guidebook.

Pythoncode (2020). How to Get Hardware and System Information in Python. Retrieved from https://www.thepythoncode.com/article/get-hardware-system-information-python