# Customer Segmentation Report for Arvato Financial Services

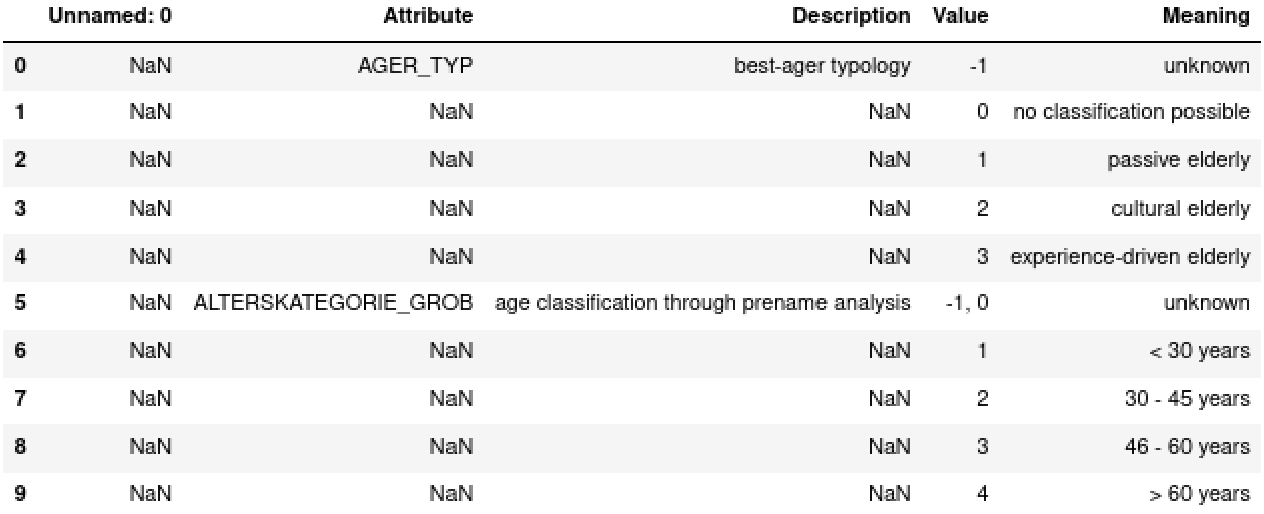
Umar Ul-Hassan

# 1 Data exploration and visualisation

In this section the data outlined in the proposal will be reviewed. Ant abnormalities within the data set will also be rectified and the steps taken to do this will be discussed in this section.

## 1.1 Cleaning the attribute values data set and applications

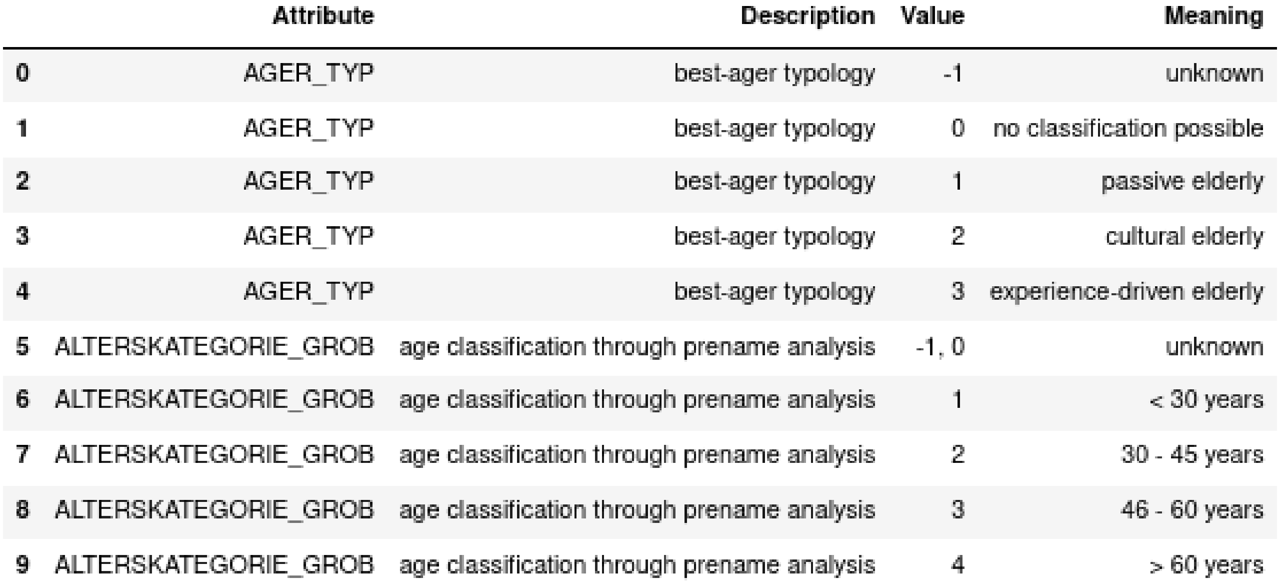
The attribute values table obtained from 'DIAS Attributes - Values 2017.xlsx' has several issues which need fixing. These can be seen in Figure 1. First of all the 'Unnamed: 0' is redundant and can be dropped. Furthermore the NaN in this context should be replaced by the latest entry. This was done by using the pandas 'ffill' method on both the 'Attribute' and 'Description' columns



| | Unnamed: 0 | Attribute | Description | Value | Meaning |
|---|---|---|---|---|---|
| 0 | NaN | AGER_TYP | best-ager typology | -1 | unknown |
| 1 | NaN | NaN | NaN | 0 | no classification possible |
| 2 | NaN | NaN | NaN | 1 | passive elderly |
| 3 | NaN | NaN | NaN | 2 | cultural elderly |
| 4 | NaN | NaN | NaN | 3 | experience-driven elderly |
| 5 | NaN | ALTERSKATEGORIE_GROB | age classification through prename analysis | -1, 0 | unknown |
| 6 | NaN | NaN | NaN | 1 | < 30 years |
| 7 | NaN | NaN | NaN | 2 | 30 - 45 years |
| 8 | NaN | NaN | NaN | 3 | 46 - 60 years |
| 9 | NaN | NaN | NaN | 4 | > 60 years |

Figure 1: The 'DIAS Attributes - Values 2017' dataset in its crude form

The resultant dataframe can be seen below in Figure 2.



| | Attribute | Description | Value | Meaning |
|---|---|---|---|---|
| 0 | AGER_TYP | best-ager typology | -1 | unknown |
| 1 | AGER_TYP | best-ager typology | 0 | no classification possible |
| 2 | AGER_TYP | best-ager typology | 1 | passive elderly |
| 3 | AGER_TYP | best-ager typology | 2 | cultural elderly |
| 4 | AGER_TYP | best-ager typology | 3 | experience-driven elderly |
| 5 | ALTERSKATEGORIE_GROB | age classification through prename analysis | -1, 0 | unknown |
| 6 | ALTERSKATEGORIE_GROB | age classification through prename analysis | 1 | < 30 years |
| 7 | ALTERSKATEGORIE_GROB | age classification through prename analysis | 2 | 30 - 45 years |
| 8 | ALTERSKATEGORIE_GROB | age classification through prename analysis | 3 | 46 - 60 years |
| 9 | ALTERSKATEGORIE_GROB | age classification through prename analysis | 4 | > 60 years |

Figure 2: The 'DIAS Attributes - Values 2017' dataset' after being cleaned

Another issue however arises in that there are some values which have unknown meanings. Therefore this data set will be used to identify values which should be replaced with NaN in the 'Udacity_AZDIAS_052018', 'Udacity_CUSTOMERS_052018', 'Udacity_MAILOUT_052018_TRAIN' and 'Udacity_MAILOUT_052018_TEST' data sets.

## 1.2 Cleansing miscellaneous columns

There are several columns in the data set which have undisclosed values which have no meaning. Examples of columns with these issues will be listed below:

- **CAMEO_DEUG_2015'**: Contains the string 'X' in what appears to be an exclusively integer column. To remedy this any entry with 'X' will be converted to NaN and the column will be converted into integer data type.
- **CAMEO_INTL_2015**: Contains the string 'XX'. This will be replaced with NaN. The data in this column is numerical and hence will also be converted to integer data type.
- **CAMEO_DEU_2015**: Contains the string 'XX'. This will be replaced with NaN. Given that this data in this column is categorical in nature it will remain as an object data type.
- **EINGEFUEGT_AM**: This contains data about the time of entry of the data. This data is redundant since customer segmentation is not a time series problem. Therefore this column will be dropped

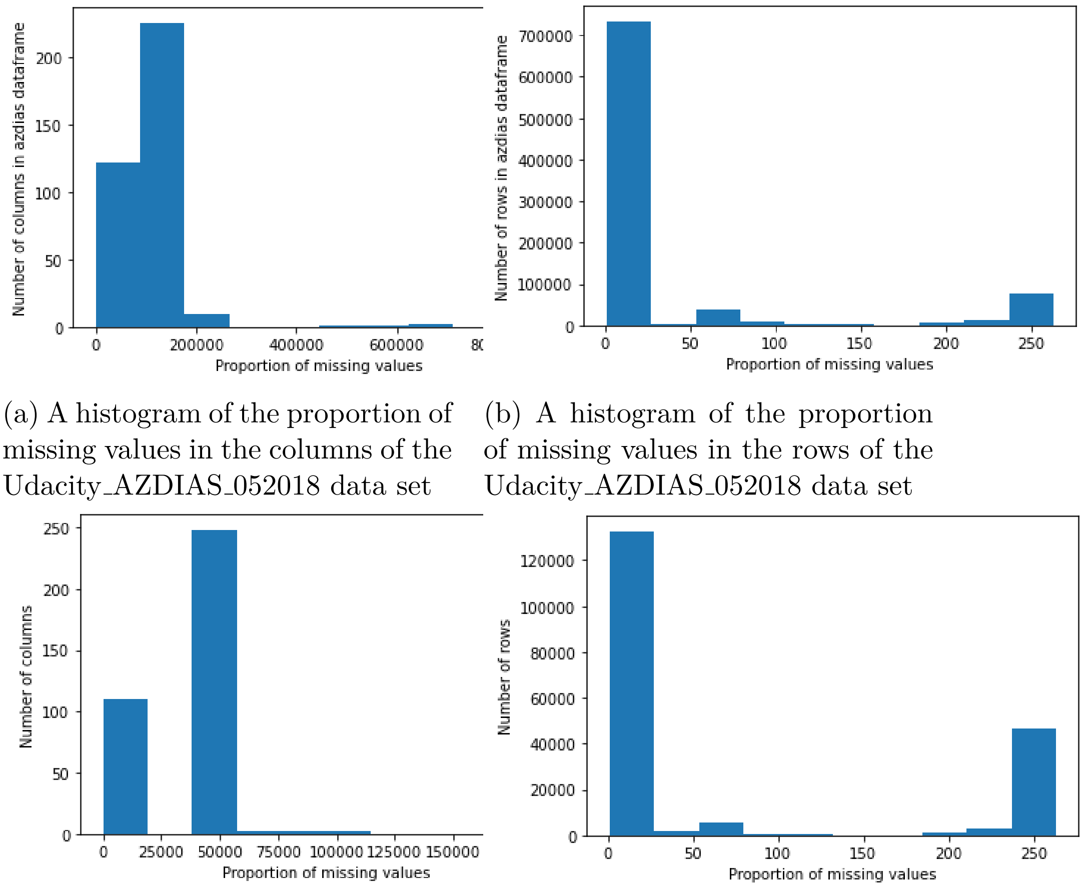
## 1.3 Dropping columns and rows

Prior to imputing missing values within the relevant data sets, it is important to first drop rows and columns with too many missing values. This is because these rows and columns will not offer sufficient information to the machine learning algorithms to provide useful insights and in some instances may deteriorate the performance of the machine learning models. In order to determine the threshold for the number of missing values each row/column should have before dropping them, it is best to plot a histogram in order to ensure not all the data will be dropped and meet a good compromise between retaining enough data and removing features and entries which may impede the performance of the machine learning models used in the later stages of this project.

In this instance, rows with missing values exceeding 50 were dropped for both the Udacity_AZDIAS_052018 and Udacity_CUSTOMERS_052018 data sets. For the Udacity_CUSTOMERS_052018 data set any columns with more than 60,000 missing values were dropped and for the Udacity_AZDIAS_052018 data set columns with more than 300,000 missing values were dropped. A similar procedure for dropping rows and columns for the other data sets were also implemented.

## 1.4 Data Preprocessing

Prior to using the data for training the machine learning models, it is important to first impute the missing values so that more useful information can be used for training the machine learning models. It is also import to perform feature scaling so that the range of values don't make the machine learning model favour features with larger values over others. However each data type must be handled differently and how each data type is dealt with will be outlined below.



(a) A histogram of the proportion of missing values in the columns of the Udacity_AZDIAS_052018 data set

(b) A histogram of the proportion of missing values in the rows of the Udacity_AZDIAS_052018 data set

(c) A histogram of the proportion of missing values in the columns of the Udacity_CUSTOMERS_052018 dataset

(d) A histogram of the proportion of missing values in the rows of the Udacity_CUSTOMERS_052018 dataset

Figure 3: Histograms representing the proporttion of missing values along the rows and columns of the Udacity_CUSTOMERS_052018 and Udacity_AZDIAS_052018 datasets

.

### 1.4.1 Categorical data

First the categorical data will be imputed with the modal value for that specific column as recommended in industry [1]. The categorical data will first be encoded using a ordinal encoder. This is because a lot of the categorical data types have a relationship to one another and this relationship can be exploited through the use of an ordinal encoder to make training the machine learning models more effective. After the ordinal encoding a Min-Max scaling will be used. This is because the maximum and minimum are well defined for categorical data.

### 1.4.2 Integer data

From the insights attained from the DIAS Attributes - Values 2017, it is apparent that a lot of the coumns with integer inputs are actually categorical data. Therefore the same preprocessing steps used in subsubsection 1.4.1 will be used.

### 1.4.3 Floating point data

The values will be imputed using the mean as this makes the most sense in this instance [1]. The Min-Max scaler will also be used because using a standard scaler may lead to floating point data to be treated differently as the majority of the data will have a range between 0 and 1. Therefore the presence of negative values or values with an absolute value exceeding 1 may lead to sub par performance for the machine learning models models in this project.

# 2 Algorithms and techniques

Now that the data has been cleaned, it will now become possible to perform machine learning on the data. The first task to be performed will be segment the customers and the general population into clusters and identify how much of the general population can fit into one of the general population clusters.

After this supervised machine learning will be used on the to classify whether individuals of the 'Udacity_MAILOUT_052018_TRAIN' data set are responsive to mail outs and the best machine learning models will be further analysed and the best model will be used to classify individuals whom respond to the MAILOUT in the 'Udacity_MAILOUT_052018_TRAIN' data set.

## 2.1 Customer segmentation

Prior to performing clustering to segment the populations described in the 'Udacity_MAILOUT_052018_TRAIN' and the 'Udacity_CUSTOMERS_052018' data sets it is important to first perform feature selection in order to perform dimension reduction and identify the more important features for clustering. After this the proportion of customers relative to the general population will be examined to gain additional insights.

### 2.1.1 Dimension reduction

Given that there are no labels for the data sets used for the customer segmentation, an unsupervised feature selection method known as PCA will be utilised for dimension reduction. PCA is used to identify which features are the most responsible for the variability if the dataset. Any variables which have low variability are likely to have no impact on the overall performance of the clustering model and should therefore be removed to speed up the training step and improve interpretability.

From Figure 4 it looks like 100 components largely captures most of the data as an additional 257 components only provides an additional 20% increase in the explained variance ratio. Therefore for the clustering model used later on in the customer segmentation part, the top 100 components from the PCA analysis will be used.

### 2.1.2 PCA Feature Analysis

From Figure 5 it looks like the most important feature in the data set is whether someone's social status is rough. However the remainder of the top features from component 1 looks be a mixed bag with only the feature 'FINANZ_SPARER' complementing the financial situation theme of the most important feature 'LP_STATUS_GROB'.

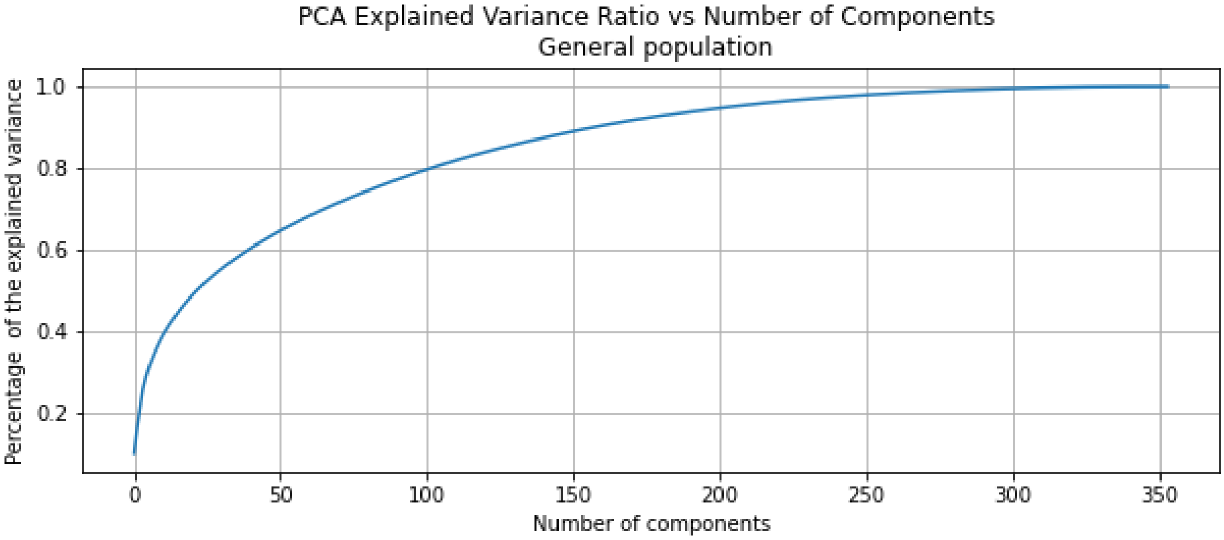
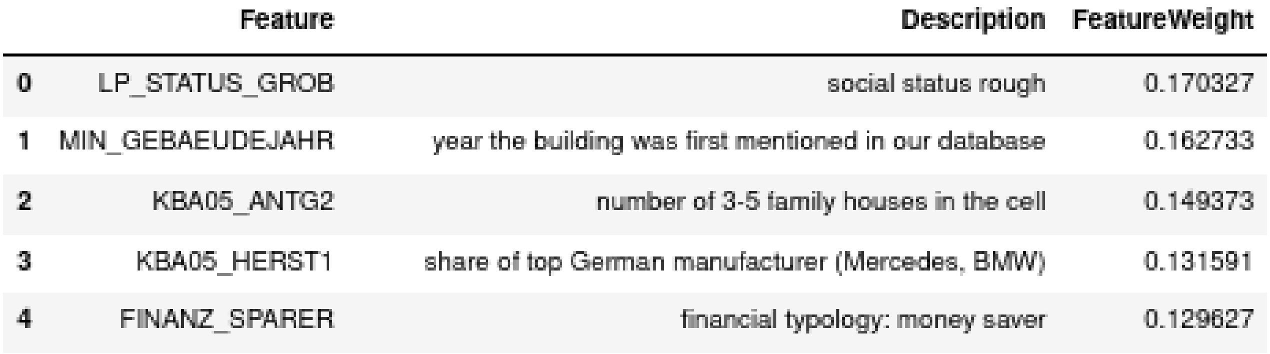


Figure 4: Explained variance ratio plotted against number of components used in PCA

?? however has a very clear theme with factors relating to the PLZ8 region larely dominating the third component from the PCA performed.



| | Feature | Description | FeatureWeight |
|---|---|---|---|
| 0 | LP_STATUS_GROB | social status rough | 0.170327 |
| 1 | MIN_GEBAEUDEJAHR | year the building was first mentioned in our database | 0.162733 |
| 2 | KBA05_ANTG2 | number of 3-5 family houses in the cell | 0.149373 |
| 3 | KBA05_HERST1 | share of top German manufacturer (Mercedes, BMW) | 0.131591 |
| 4 | FINANZ_SPARER | financial typology: money saver | 0.129627 |

Figure 5: Table of the most important features from a PCA of the first component

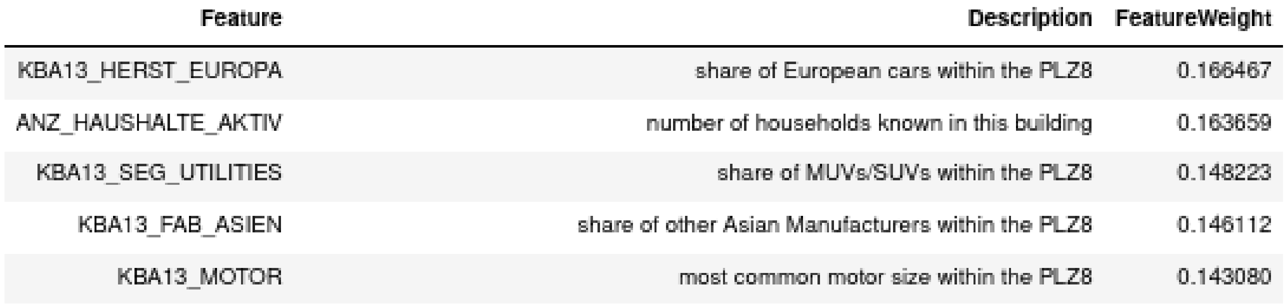
## 2.2 Clustering

The next step after dimension reduction is to try and segment the general population and the customer population. This will be done using K-Means clustering. First the K-Means algorithm will be fitted against the whole of the 'Udacity_AZDIAS_052018' data set. And then predictions will be made on both the 'Udacity_AZDIAS_052018' and 'Udacity_CUSTOMERS_052018' datasets in order to segment both the general population and customer populations.

Prior to segemtning both of the aforementioned populations it is important to first identify what the optimal number of clusters are. This will be done using yellowbrick linrary [2]. This library makes it possible to use an algorithm to determine the elbow point and hence the optimal number of clusters rather than qualitatively.

From the results from Figure 7 it is seen that the optimal number of clusters is 5.

The results from segmenting both the customer populations and general populations can be seen in Figure 8 and it looks like the number of customers in each cluster is roughly



| Feature | Description | FeatureWeight |
|---|---|---|
| KBA13_HERST_EUROPA | share of European cars within the PLZ8 | 0.166467 |
| ANZ_HAUSHALTE_AKTIV | number of households known in this building | 0.163659 |
| KBA13_SEG_UTILITIES | share of MUVs/SUVs within the PLZ8 | 0.148223 |
| KBA13_FAB_ASIEN | share of other Asian Manufacturers within the PLZ8 | 0.146112 |
| KBA13_MOTOR | most common motor size within the PLZ8 | 0.143080 |

Figure 6: Table of the most important features from a PCA of the third component

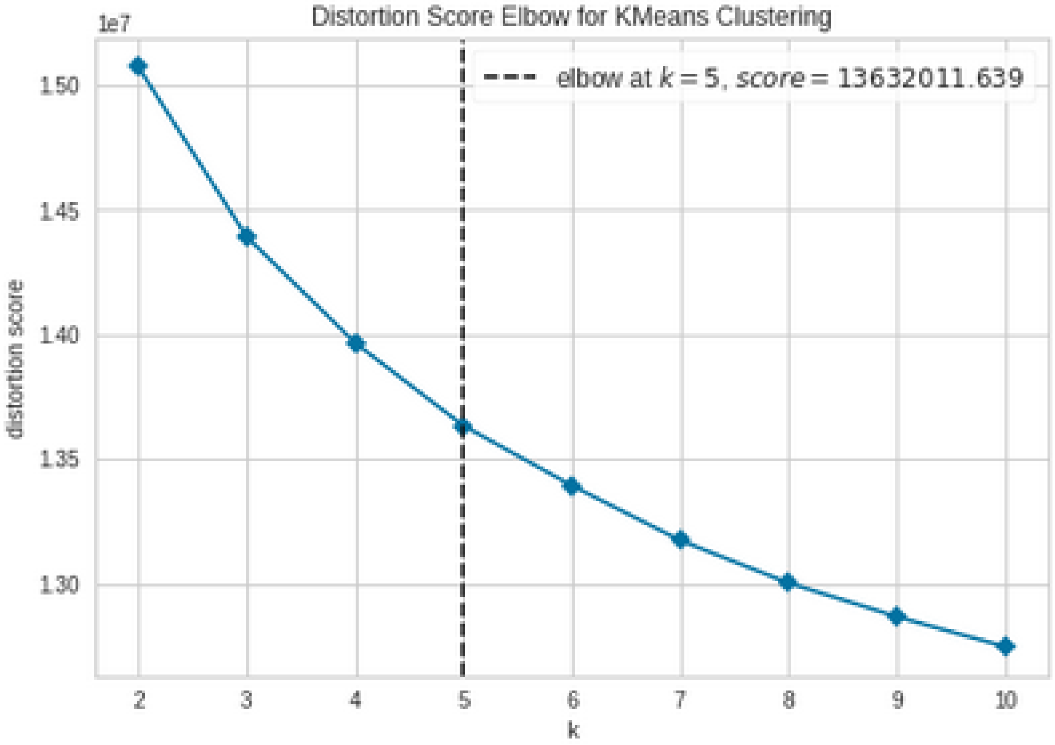


Figure 7: Plot of the elbow curve for the K-Means clustering algorithm using the sum of squared distances from each point to its assigned center

evenly distributed along each of the clusters. This implies that the proportion of customers are evenly distributed along the general population. This can be confirmed by taking the ratio of the proportion of the customer population segments and general population segments as seen in Figure 9

# 3 Customer Acquisition

In this section of the project supervised learning algorithms will be trained using the ’Udacity_MAILOUT_052018_TRAIN’ data set. The end result of the training will be to produce a model which can accurately predict whether or not an individual will respond to a mail out. The labels are included in an additional column called ’RESPONSE’. This column has not been cleansed or preprocessed.

In addition the dataset has been re-balanced so that there is an equal proportion of entries where people have responded as well as people who have not.

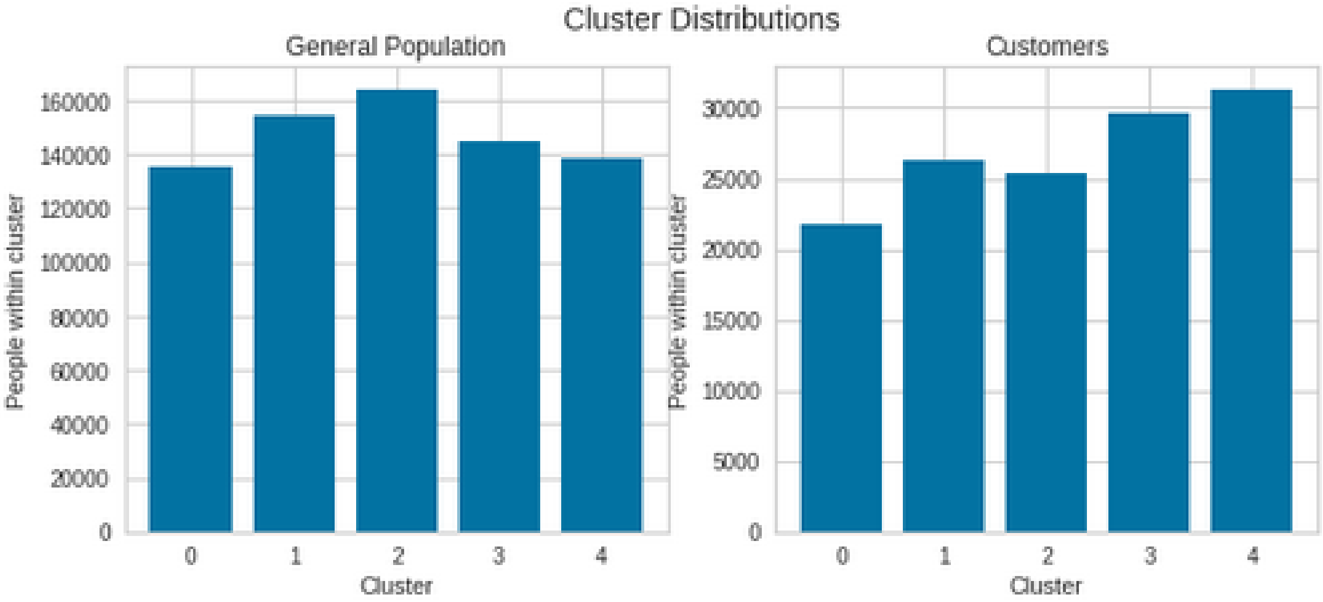


Figure 8: Number of people in the customer population and general population within a given cluster

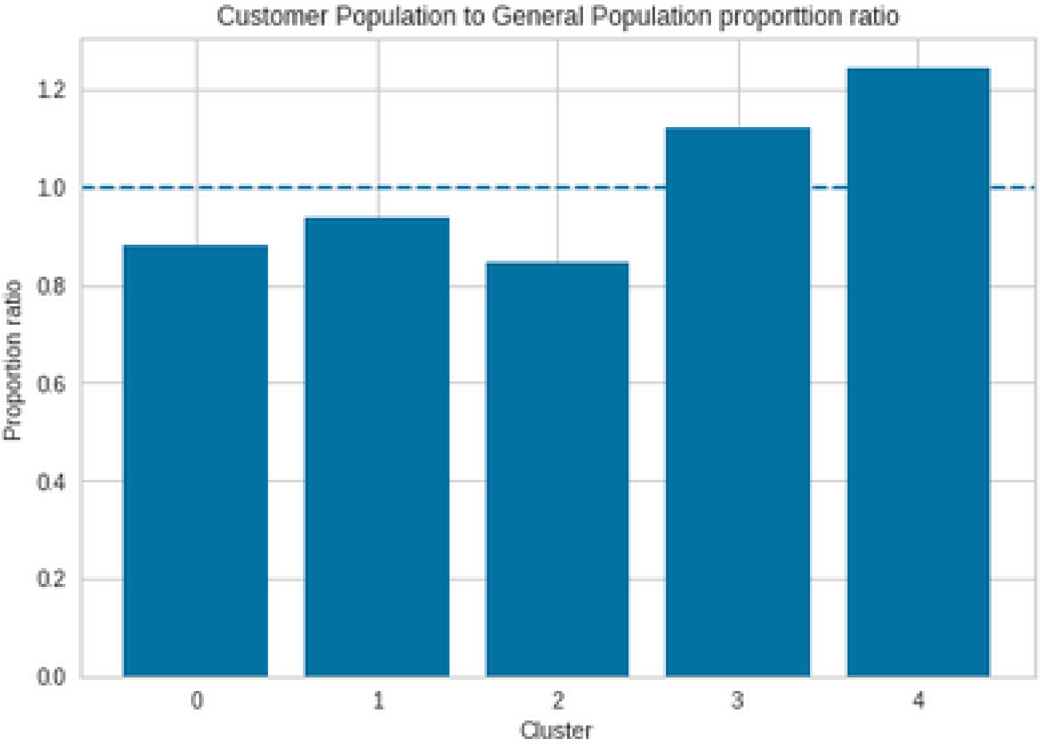
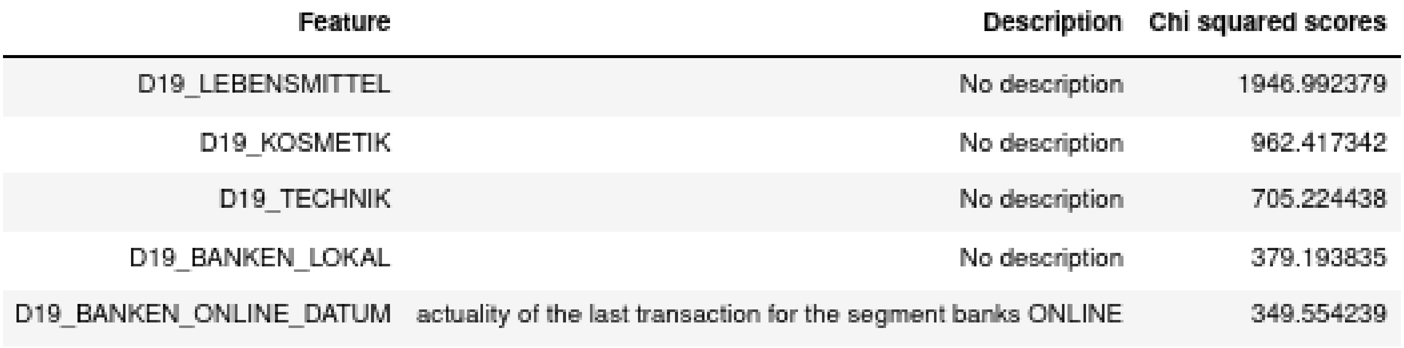


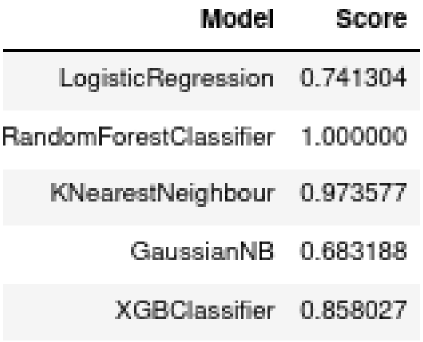
Figure 9: Number of people in the customer population and general population within a given cluster



| Feature | Description | Chi squared scores |
|---|---|---|
| D19_LEBENSMITTEL | No description | 1946.992379 |
| D19_KOSMETIK | No description | 962.417342 |
| D19_TECHNIK | No description | 705.224438 |
| D19_BANKEN_LOKAL | No description | 379.193835 |
| D19_BANKEN_ONLINE_DATUM | actuality of the last transaction for the segment banks ONLINE | 349.554239 |

Figure 10: Table showing the most important features according to the kbest selector along with their descriptions and chi squared score.

Furthermore the scikit-learn class **SelectKBest** has been implmented for supervised feature selection for dimension reduction. Chi squared was used as the scoring metric and the total number of features chosen was 150. What is interesting to note is that the top 5 most important features are associated with the D19 (unfortunately no information was given about what these features mean) as evidenced by Figure 10.



| Model | Score |
|---|---|
| LogisticRegression | 0.741304 |
| RandomForestClassifier | 1.000000 |
| KNearestNeighbour | 0.973577 |
| GaussianNB | 0.683188 |
| XGBClassifier | 0.858027 |

Figure 11: A bar plot of the count of each of the distinct labels in the Udacity_MAILOUT_052018_TRAIN' prior to rebalancing the data set.

## 3.1 Benchmark

Prior to hyper parameter tuning it is best to first compare the performance of the defaults of a variety of machine learning models. Then the models which exceed the performance of the base line model (in this instance logistic regression) will be further optimised using hyper parameter tuning.

From the results shown in Figure 13 it is clear that the xgboost algorithm and the k nearest neighbours algorithm should be further examined.

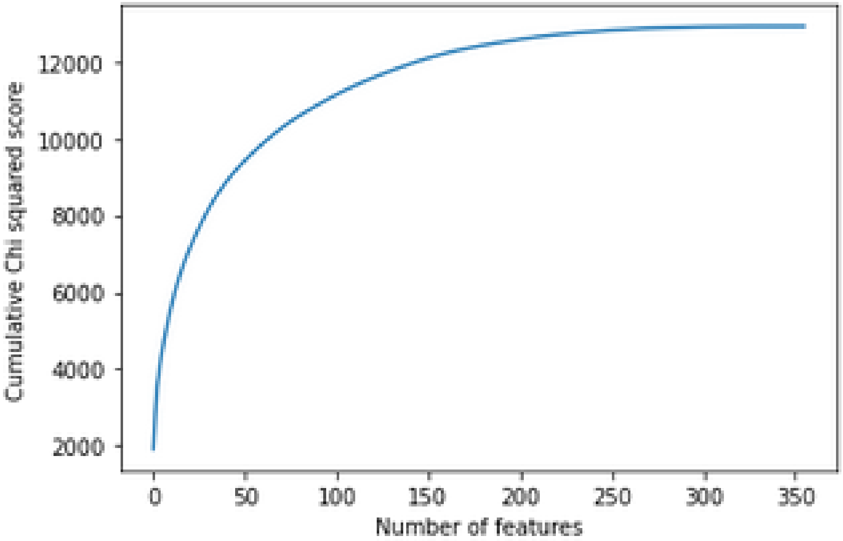
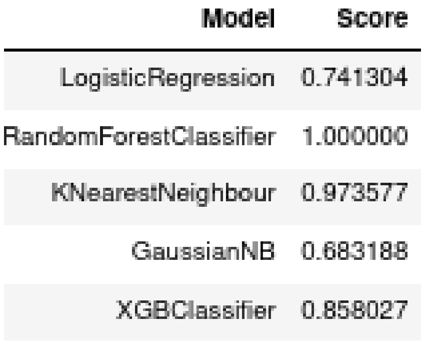


Figure 12: Cumulative Chi squared score plotted against the number of features for the Udacity_MAILOUT_052018_TRAIN data set



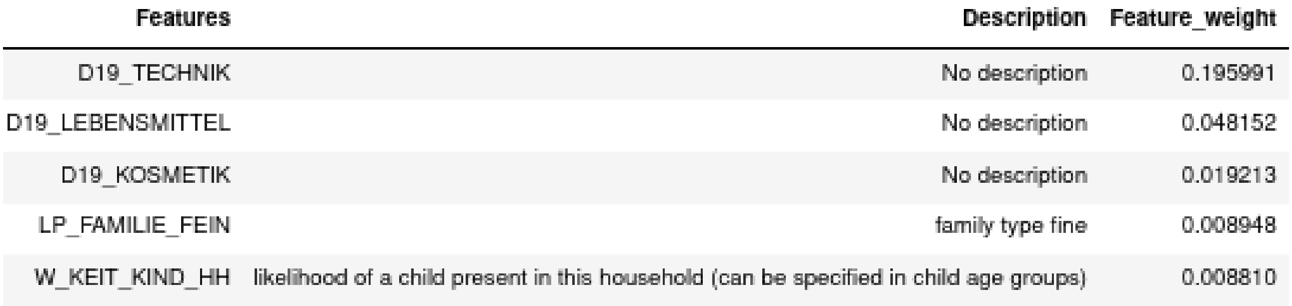
| Model | Score |
|---|---|
| LogisticRegression | 0.741304 |
| RandomForestClassifier | 1.000000 |
| KNearestNeighbour | 0.973577 |
| GaussianNB | 0.683188 |
| XGBClassifier | 0.858027 |

Figure 13: Table of the performance of a variety of machine learning models

### 3.1.1 Xgboost

The first algorithm to analyse will be xgboost. The feature importance's for the xgboost model are shown below:

What is clear from Figure 14 is that the most important feature by far is 'D19_TECHNIK'. All the other appear insignificant in comparison to this feature. The final score for the optimised model is 0.953.



| Features | Description | Feature_weight |
|---|---|---|
| D19_TECHNIK | No description | 0.195991 |
| D19_LEBENSMITTEL | No description | 0.048152 |
| D19_KOSMETIK | No description | 0.019213 |
| LP_FAMILIE_FEIN | family type fine | 0.008948 |
| W_KEIT_KIND_HH | likelihood of a child present in this household (can be specified in child age groups) | 0.008810 |

Figure 14: Table showing the most important features for the xgboost model

### 3.1.2 K Nearest Neighbour

The end result from the k nearest neighbour model was a result of 0.990 which means that this is the model which will be used for making predictions. Unfortunately the run time for creating a permutation importance score was to be used to determine the most important features for the k nearest neighbours algorithm [3].

## 4 improvements and future steps

In the future the following steps should be taken:

- Experiment with more forms of feature encoding such as one hot encoding
- Experiment with the parameters associated with the preprocessing steps
- Experiment with using more or less features in training the machine learning algorithms.
- Increase the search space for the hyperparameters used in creating the machine learning pipeline.
- Determine what the most important features are for the k nearest neighbours by using a permutation feature importance scores.

## References

[1] O. Anita, “Data handling scenarios part 2: Working with missing values in a dataset,” Apr 2021. [Online]. Available: https://heartbeat.fritz.ai/data-handling-scenarios-part-2-working-with-missing-values-in-a-dataset-34b758cfc9fa

[2] B. Bengfort, R. Bilbro, N. Danielsen, L. Gray, K. McIntyre, P. Roman, Z. Poh *et al.*, “Yellowbrick,” 2018. [Online]. Available: http://www.scikit-yb.org/en/latest/

[3] J. Brownlee, “How to calculate feature importance with python,” Aug 2020. [Online]. Available: https://machinelearningmastery.com/calculate-feature-importance-with-python/