Udacity Data Science Nanodegree

Capstone Project

Arvato Financial Solutions – Customer Segmentation

Author:

David Kimani

Course:

Data Science Nanodegree - Udacity

Date: 13 October 2020

Contents

1 Overview

This report completes part of the requirements for the Data Science Nanodegree – Capstone Project which explores methods for customer segmentation for the Arvato Financial Solutions. We have been asked to test a model to population data to pinpoint the people who Arvato Financial Solutions should contact that have a higher likelihood of responding to the next mailout promotions.

To support this work Arvato has provided the data of the general population of Germany, their customer data, client response to previous mailouts which has been divided into a train and test dataset.

The defined steps to complete this project are:

1. Customer segmentation report
2. Supervised leaning model
3. Kaggle competition

Supporting documentation and analysis can found at (Github), this will not include the dataset provided as they are protected under terms and conditions.

2 Background

Arvato's serviced spans customer support, information technology, logistics and finance and is based in Germany.

This project will analyze the demographic data for customers of a mail-order sales company in Germany to the general population using unsupervised learning methods to perform customer segmentation, thus identifying part of the population that best describes the core customers for this company.

What was learnt in the unsupervised leaning segmentation will be will be implemented onto a third dataset with demographic information for labels of a marketing campaign, and using supervised leaning methods predict which individuals are most likely to convert to become customer of this company.

Marketing that is customer centric is a rapidly growing field that benefits greatly from precise customer segmentation, with the developments in machine learning pattern can be found in large volumes of data that would otherwise have gone missing.

3 Datasets

There are four datasets to be used in this project:

1. Udacity_AZDIAS_052018.csv: This is Germany demographic data for the general population which consists of 891,221 rows and 366 columns.

2. Udacity_CUSTOMERS_052018.csv: which contains the customer demographic data for the mail-order company. This has 191,652 rows and 369 columns.

3. Udacity_MAILOUT_052018_TRAIN.csv: The demographic data for individuals who were targeted at a marketing campaign and their responses included. Having 42,982 rows and 367 columns.

4. Udacity_MAILOUT_052018_TEST.csv: this is the demographic data for individuals who were targets of a marketing campaign who we will later want to know the likelihood they will become customers.

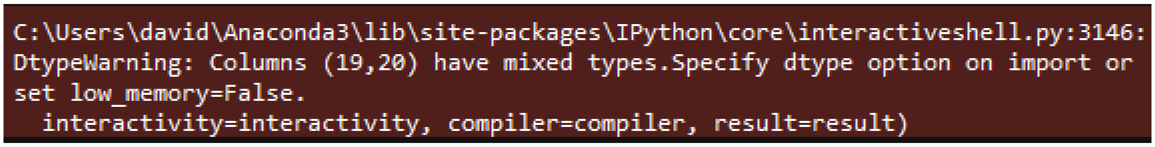
Every row of the each of the demographic files represents a person and includes information about their neighborhoods, households and building.

We have been provided two other files 'DIAS Information Level – Attributes 2017.xlsx' is a top-level list of attributes and descriptions. The other is 'DIAS Attributes – Values 2017.xlsx' this is a detailed mapping of the data values for each feature in alphabetical order.

6 Preprocessing

Warnings

Figure 1 shows mixed type warnings that was displayed when the data was loaded.



```
C:\Users\david\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3146:
DtypeWarning: Columns (19,20) have mixed types.Specify dtype option on import or
set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Figure 1 mixed types warning.

This warning was explored and was found to be due to some string entries that were included in these two columns both in the azdias (population dataset), customers (customer datasets) and mailouts datasets. This was found to be "X" and "XX" in these columns while the remaining inputs were numerical. Thus, the mixed type warning received.

A function (content_fix(df, column)) was created to convert the "X", "XX", " " and "" into null values as these data represents instances where they did not have this information.

Once this warning was delt with the next step is to explore the population and customer data checking the columns for the number of missing values they have, and also look at the 'DIAS Attributes – Values 2017.xlsx' (Attributes data) to see what insight this can gain about the data.

Feature reduction using attributes dataset

A review of the Attributes dataset reviled that it did not have descriptions for all the features of the all the datasets and thus could be used to reduce the number of features. To accomplish this a unique list of the Attributes was generated and the datasets were filtered with this list to leave only the columns we have descriptions for. I would not be useful to use data that we cannot define. This took the features to work with to 272.

Data Exploration

Following the reduction using the Attributes dataset we looked to explore what missing data we had in each feature to see some could be dropped. The following histograms were plotted for the population and customer datasets:

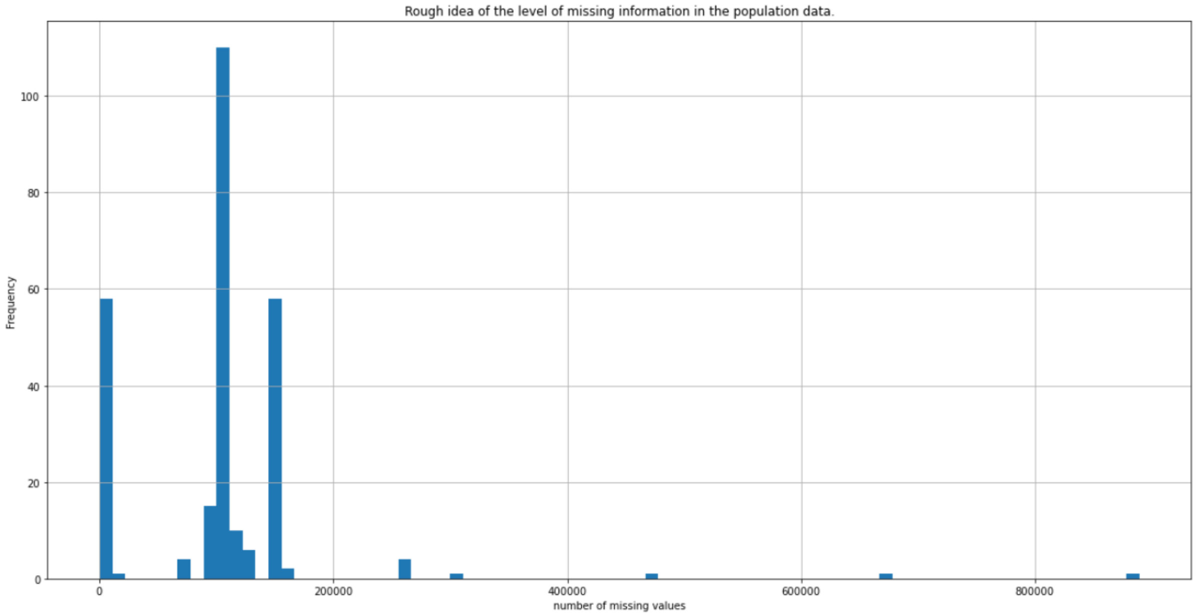


Figure 2 missing data in the population dataset.

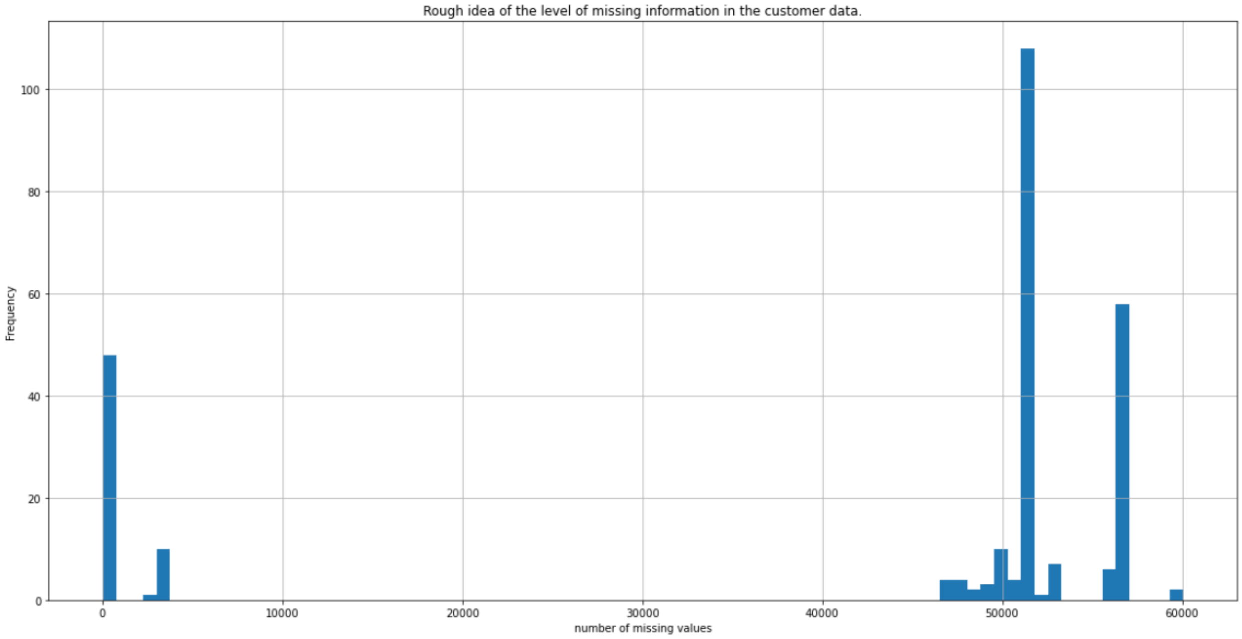
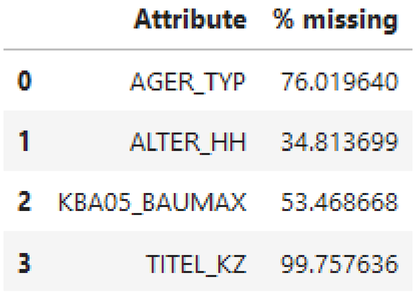


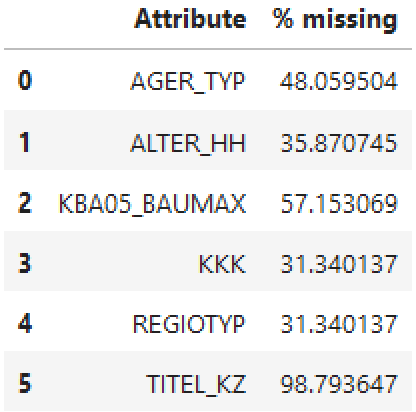
Figure 3 missing data in the customer dataset.

It was deemed that if more than 30% of the data was missing in a column that column would be dropped and the remainder imputed with the most frequent values when we move to the machine learning segments of this study. Below are the columns from the population and customer dataset which have more than 30% data missing.



	Attribute	% missing
0	AGER_TYP	76.019640
1	ALTER_HH	34.813699
2	KBA05_BAUMAX	53.468668
3	TITEL_KZ	99.757636

Table 1 Columns with more than 30% null values in the population dataset.



	Attribute	% missing
0	AGER_TYP	48.059504
1	ALTER_HH	35.870745
2	KBA05_BAUMAX	57.153069
3	KKK	31.340137
4	REGIOTYP	31.340137
5	TITEL_KZ	98.793647

Table 2 Columns with more than 30% null values in the customer dataset.

Decision was made to keep the KKK which stands for purchasing power and REGIOTYP which is the social class of the neighborhood the person lives.

Dealing with missing values

Now that we only have columns with less than 30% missing values with the exception of KKK and REGIOTYP, the missing values should be imputed before we progress to the machine learning methods to segment the population to those most likely to respond to the mailout campaign and become customers.

The method selected uses sklearn's SimpleImputer module to impute the missing values with the most frequent values in that column.

It is also prudent to check the missing values along the rows. The rows have been deleted if the row has more than 50% of the data missing.

Feature Exploration

The following was an attempt to reduce the features using the SelectKBest from sklearn's feature selection. This required the data to be scaled, in this instance, sklearn's preprocessing module normalizer was used. The categorical features where be passed through a function (dummy_variables(df, cat_cols, dummy_na=False) to convert the data into 1 or 0.

The customer data was used for the exploration of the best features as we have the labeled data. The MULTI_BUYER column was used, converted into dummy variables so that we can assess the columns that are the best predicting who would be potential customer.

Now with the scaled features data and the labels for the customer dataset, we can put this data through the SelectKBest module with the score function using the chi2 module.

This scaled features and the labels were fitted to the SelectKBest module and the result was plotted, this can be seen in Appendix A. This allows for a list of the best columns greater than a chosen score can be selected and one can create a list of the best features and the worst features.

An alternative method using PCA model for feature selection resulted in 151 features selected in comparison to the 167 from the SelectKBest method. Tweaking of the features selected should be able to yield desirable results when modeling and clustering the data as features that would otherwise have adverse effect have been removed.

Unsupervised Learning

The next section discusses the unsupervised learning method for population and customer segmentation for the mailout campaign.

This will end with a selection from the clusters generate for the population that best aligns to the customer demographic.

Supervised Leaning

After the unsupervised learning modeling an exploration of the supervised learning method will be used to predict the which people in the mailout test dataset are likely to respond to the mail-order and become customer.

This will leave us with an output of the customer ID (LRN) and the likelihood that that person will be a customer.

7 Unsupervised Learning

At the point we can get the categorical data in the datasets converted into dummy variables, cleaned so there is no missing data, and scaled.

The scaling used for this section is he RobustSclaer (Scikit learn, 2020) module which gave the best performance to the model.

Dimensionality Reduction using PCA

PCA stands for Principle Component Analysis, which is a linear dimensionality reduction method that uses Singular Value Decomposition of the data to project lower dimensional space. (Scikit learn, 2020)

The main benefits to use PCA in this task are:

- It reduces the training time,
- Removes noise – keeping only what's relevant,
- Makes visualizations possible.

The cleaned, scaled and imputed population dataset was pass through the PCA function to create a PCA model which was plotted using the pca_plot function which takes the model and plots the eigenvalues factors of the principle components.

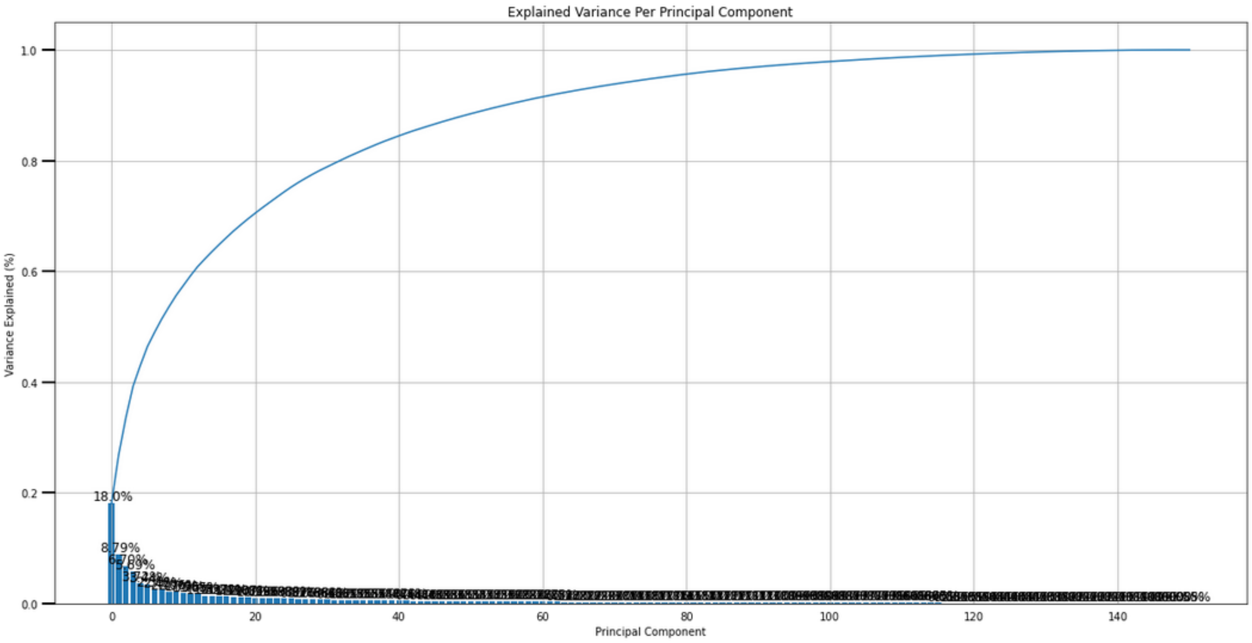


Figure 4 pca_plot using RobustScaler.

Selection of 80 principle components was used for the PCA model as they explain approximately 97% of the data.

The fit_transform method was used on the population data and we could move to the clustering of the data to gain some insights about which segment of the population we should target.

For this sklearn's KMeans module was used as stated in the KMeans documentation:

"The KMeans algorithm clusters data by trying to separate samples in n groups of equal variances, minimizing a criterion known as the inertia or within-cluster sum-of-squares." (SciKit learn, 2020)

Frist one must determine the number of centroids, these are the imaginary centers of each of the clusters. To do this the get_kmeans_score functions was written to run through a range of centroid values and plot the output. This output is then used to support the elbow method (Alade, T, 2018) which in the plot below shows that the optimal k is 5 for this dataset.

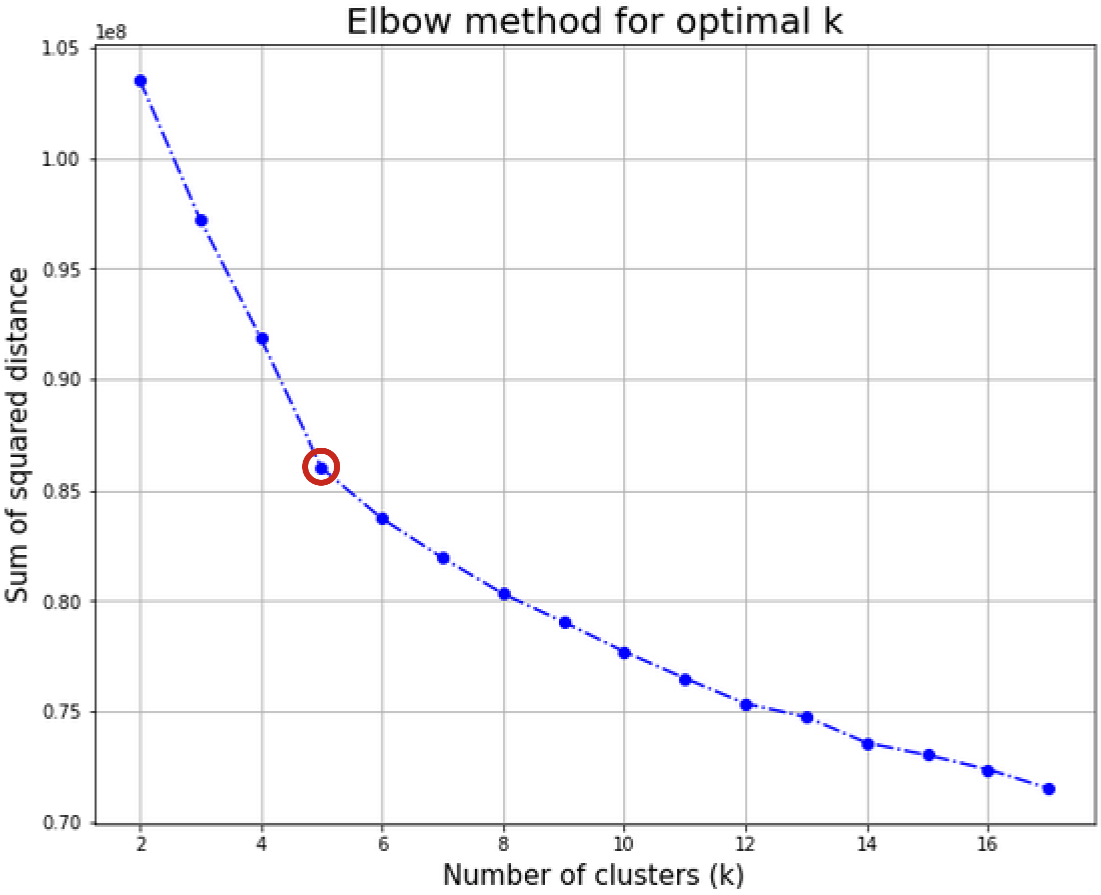


Figure 5 Elbow method for optimal k.

Armed with the optimal k the KMeans model is generated with this value then we fit_predict the pca transformed population data and the pca transformed customer data to obtain the following plot which aligns the clusters.

Using k = 5 produces the following:

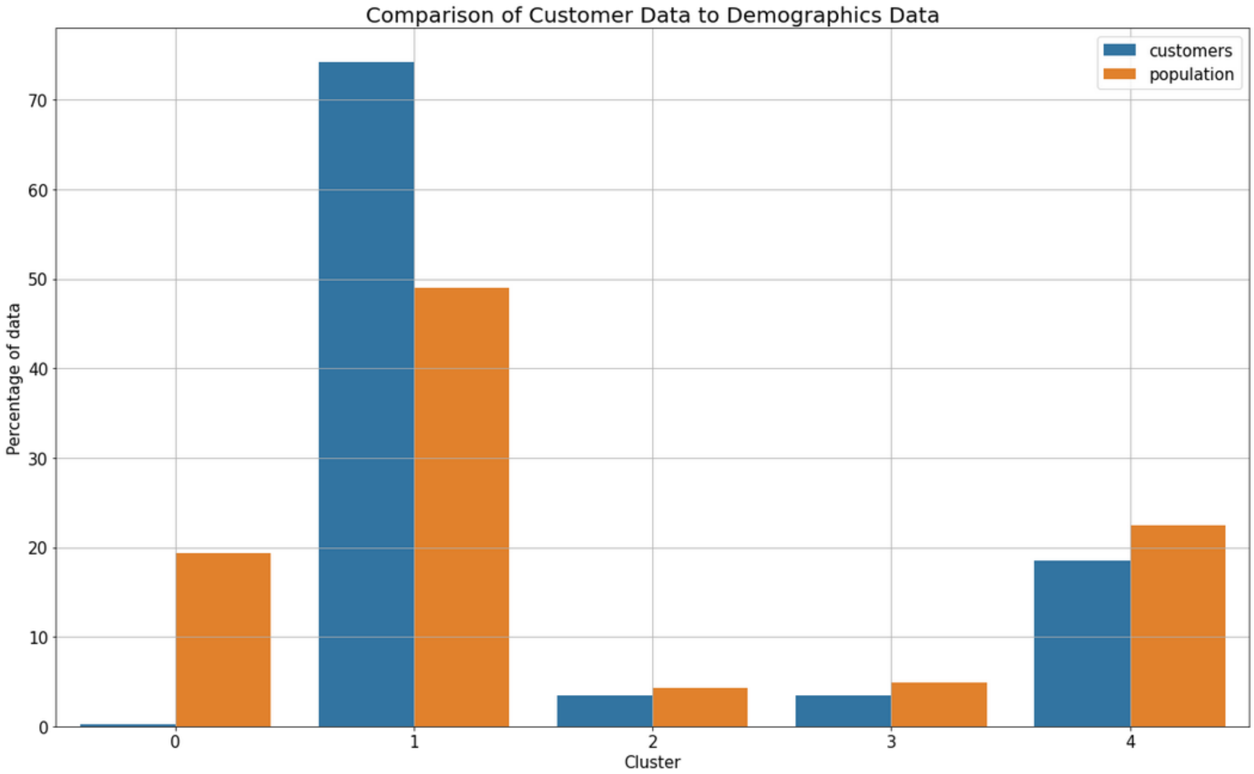


Figure 6 Population and customer clusters at k=5.

This indicates that to save some money and time we should focus on the people in clusters 1 and 4 explain 92% of the current customer, where 74.28% is explained by cluster 1. This gives us good indication to target cluster 1 for the first phase of the mailout campaign.

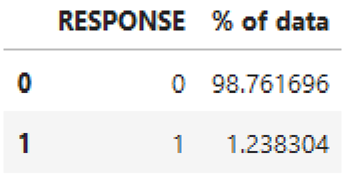
The accuracy of this prediction can be tested using previous population data and customer data prior to a previous mailout campaign and using the outcome of that campaign see what accuracy the model had against people who responded.

8 Supervised Learning

This section of the task is to build a prediction model using the Udacity_MAILOUT_052018_TRAIN.csv (training data) and Udacity_MAILOUT_052018_TEST.csv (test data) provided. Once this data was loaded a quick exploration of the training and test data was carried out.

The shape of the training dataset was 42962 rows and 367 columns. For the test dataset 42833 rows and 366 columns. The difference is the label data of the training dataset.

Taking a look at how the training labels (mail response) is split shows that there is a very small number of people who responded to the mail order campaign.



	RESPONSE	% of data
0	0	98.761696
1	1	1.238304

Table 3 percentage of responders.

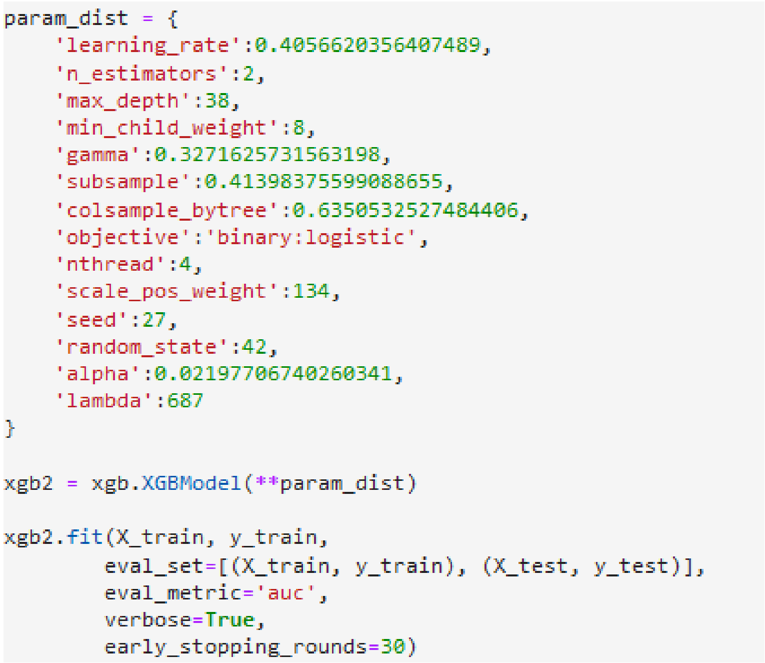
1.24% of people responded to the mailout. This can reduce the accuracy measurements used in machine learning algorithms which aim to maximize the accuracy and reduce the error.

Here the Synthetic Minority Over-sampling Technique (SMOTE) was used to reduce the effects of the oversampling (Chawla, 2002).

The training and test datasets were cleaned as with the population and customer data and missing values delt with leaving those with 30% or less missing values. These were imputed and categorical turned into dummy variables to turn them into numeric inputs.

The first model attempted was sklearn's RamdomForestRegressor which did very badly and too far to long to process the data. Further exploration on algorithms to use led to the XGBRegressor. XGBRegressor uses XGBoost which stands for Extreme Gradient Boosting (Pathak, M. 2019).

The data was split into the train_test_split for the balanced data with a test size of 0.3 and random_state 42. The parameters for the model were:



```
param_dist = {
    'learning_rate':0.4056620356407489,
    'n_estimators':2,
    'max_depth':38,
    'min_child_weight':8,
    'gamma':0.3271625731563198,
    'subsample':0.41398375599088655,
    'colsample_bytree':0.6350532527484406,
    'objective':'binary:logistic',
    'nthread':4,
    'scale_pos_weight':134,
    'seed':27,
    'random_state':42,
    'alpha':0.02197706740260341,
    'lambda':687
}

xgb2 = xgb.XGBModel(**param_dist)

xgb2.fit(X_train, y_train,
        eval_set=[(X_train, y_train), (X_test, y_test)],
        eval_metric='auc',
        verbose=True,
        early_stopping_rounds=30)
```

Figure 7 the hyperparameters to tweak the model.

These were the hyperparameters used for the model last created for this document.

The fit method was used and called for the eval_metric 'auc', which was specified for the Kaggle completion. The accuracy from this model got a best score for validation_0-auc (training accuracy): 0.99640 and validation_1=auc (test accuracy): 0.99593. This indicated that the model is most likely overfitting the data and corrections need to be made to improve the model.

One of the benefits of using XGBoost is that we can plot the feature importance of the data and remove the less useful features.

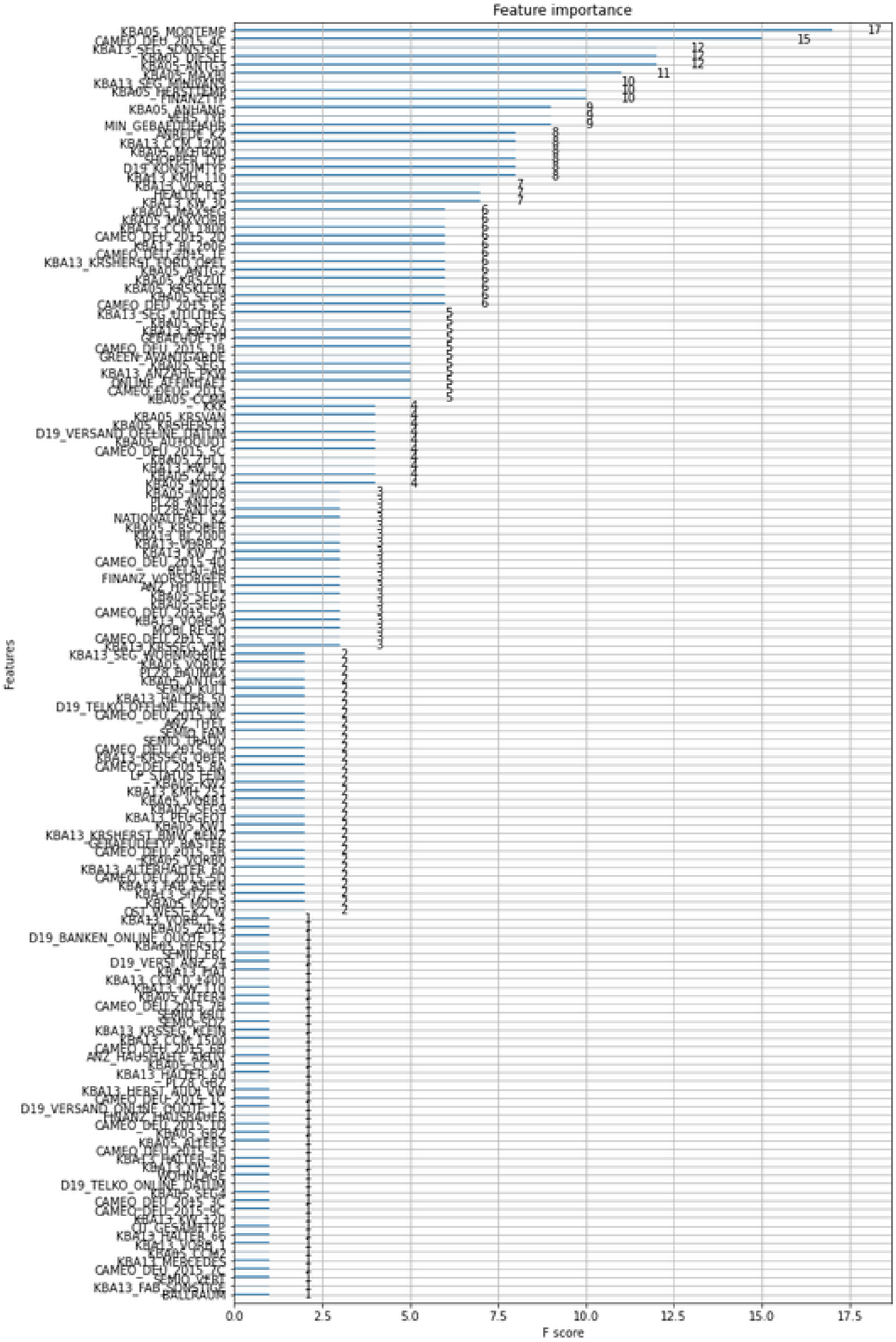


Figure 8 Feature Importance plot.

We then predict the test labels from the test features and use this for out competition submission.

9 Conclusions

To many data science and machine learning problem have multiple ways that they can be addressed. In this project we have explored how a marketing mailout strategy can be made more accurate using machine learning methods to contact people from the general population that fit the customers we currently have.

Unsupervised learning methods were used to segment the population against our current customers to pinpoint the people we should focus out mailouts.

The next task was to create a model that would predict the likelihood that people in a test dataset would respond to the mailout and become customers but fitting a model using the training dataset.

Using the Kaggle competition we are able to test the results of our supervised model predictions but we do not currently have a way to validate the unsupervised predictions.

This first submission for the Kaggle completion saw me at position 225 under the name Dave. The scoreboard can be found here. This was without SMOTE and the hyperparameter tuning using the skopt package.

10 Improvements

This project has been a challenge and presented a large learning curve and thus feel that there are a few improvements yet to be tried. Some things to aim toward when following this on are:

- Reduce the number of features using the SelectKBest and XGBoost to start with to see what improvement I get with the model.
- Improve the sample techniques to reduce the effects of imbalanced data.
- Change the missing values threshold for columns.

References

Scikit learn. 2020. SklearnpreprocessingRobustScaler. [Online]. [16 October 2020]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

Scikit learn. 2020. Principal component analysis (PCA). [Online]. [16 October 2020]. Available from: https://scikit-learn.org/stable/modules/decomposition.html#pca

Scikit learn. 2020. SklearnclusterKMeans. [Online]. [16 October 2020]. Available from: https://scikit-learn.org/stable/modules/clustering.html#k-means

Alade, T. 2018. Tutorial: How to determine the optimal number of clusters for k-means clustering. [Online]. [3 October 2020]. Available from: https://blog.cambridgespark.com/how-to-determine-the-optimal-number-of-clusters-for-k-means-clustering-14f27070048f

Chawla, N.K. 2002. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16 (2002), pp. 321-357.

Pathak, M. 2019. Using XGBoost in Python. [Online]. [19 October 2020]. Available from: https://www.datacamp.com/community/tutorials/xgboost-in-python

Appendix A

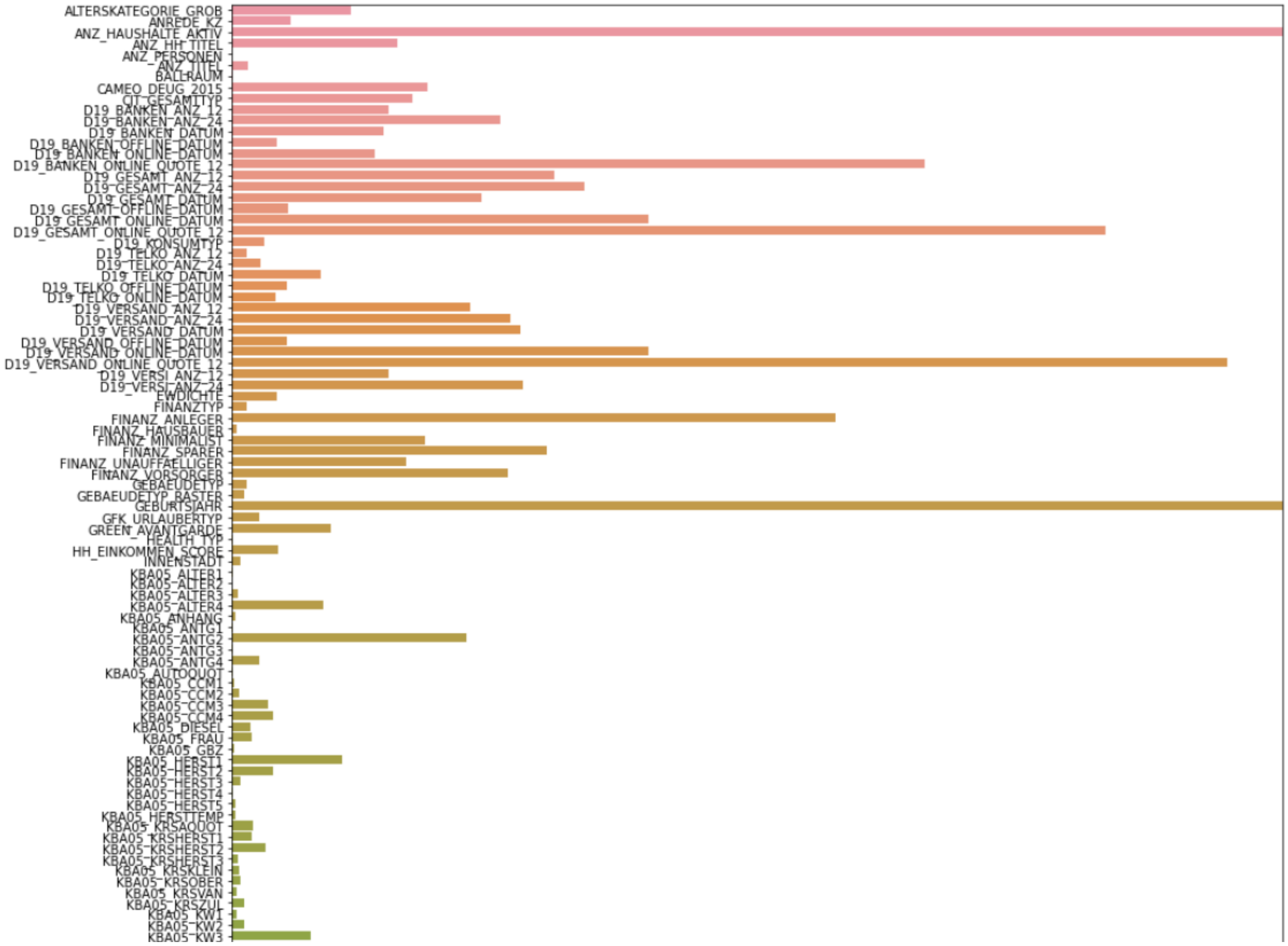


Table 4 shows a sample of the output from the SelectKBest module which highlights columns that are most aligned with multi-buyer customers.

Appendix B

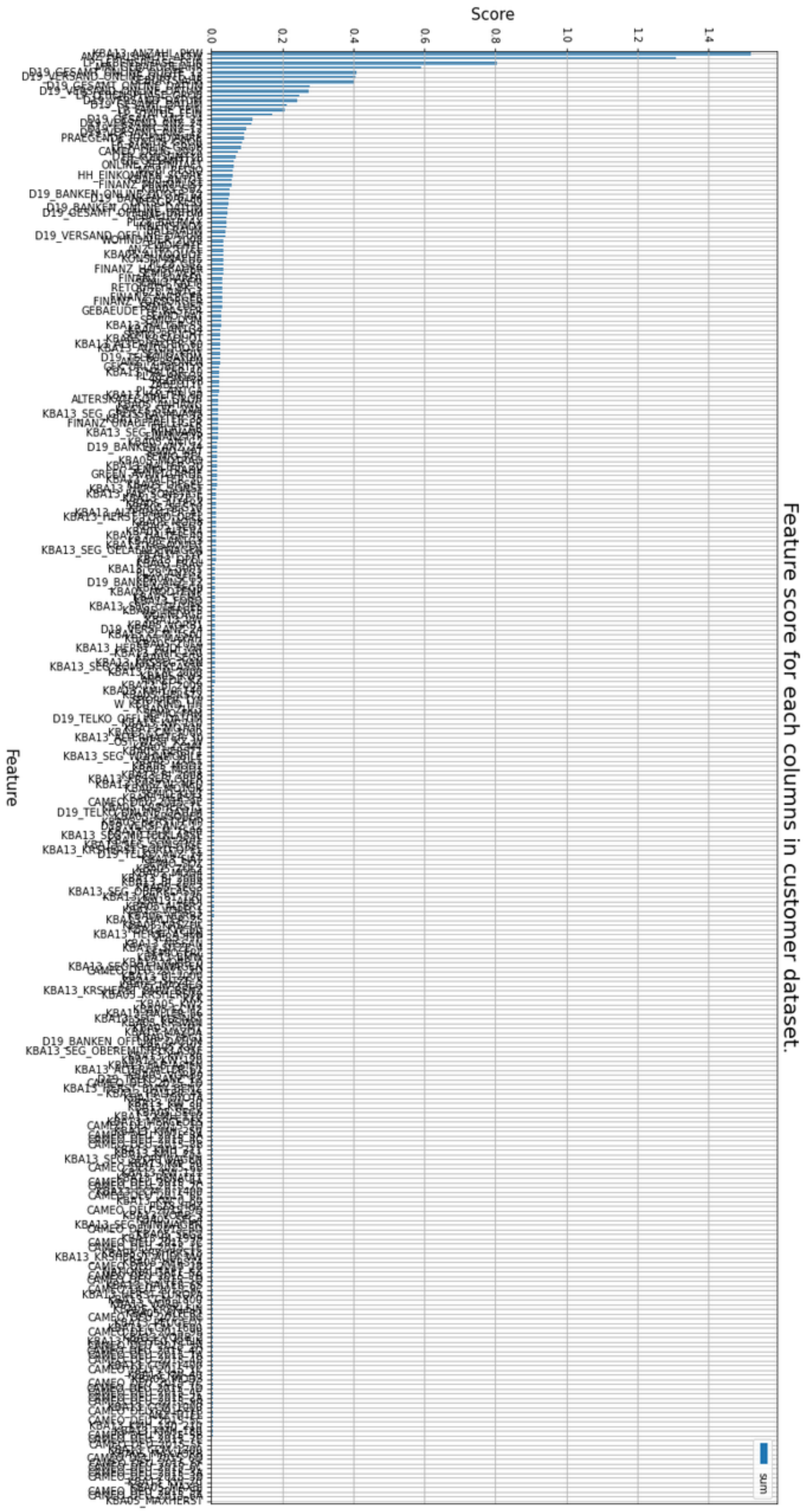


Figure 9 PCA method used for feature selection.