Customer Segmentation Report for Arvato Financial Solutions

Project Overview

The objective of this project is to predict whether targets of a marketing campaign by a mail-order company will respond positively and become a customer. The project consists of two parts: unsupervised learning and supervised learning. In the unsupervised learning part, principal component analysis and KMeans clustering are used to determine the clusters that would likely describe the attributes of a likely customer. In the supervised learning part, a model utilising ensemble classifiers would be trained on the response of a marketing campaign to predict the response of a target customer.

Problem Statement

In this project, we will analyse the demographics data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population. The project is divided into 2 parts - unsupervised learning and supervised learning.

In the first part, unsupervised learning techniques shall be used to perform customer segmentation to identify the parts of the population that best describe the core customer base of the company.
In the second part, the attributes that best identifies the customer of the company as determined in the first part of the project will be applied to a third dataset with demographics information for targets of a marketing campaign for the company. A model will be used to predict which individuals are most likely to convert into becoming customers for the company.

Metrics

The metrics that will be used to evaluate our model is the 'area under the receiver operating characteristics curve' or known as the 'ROC_AUC' score. This metric is appropriate to evaluate our model as our dataset is highly imbalanced - there is a very low occurrence of true positives. The 'ROC_AUC' score represents the

probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. It is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve while AUC represents the degree of separability. It implies the ability of the model in distinguishing between classes. Basically, the higher the value of ROC_AUC, the better the model is at predicting 0s as 0s and 1s as 1s. Applied to our problem, this means the 'ROC_AUC' score represents the probability that a classifier will rank a random target that will respond as customer (1) than a random target that will not respond as non-customer(0).

Results, analysis & discussion

Data Provided

There are 4 datasets and 2 informational files.
The datasets are:
1.) Udacity_AZDIAS_052018.csv
2.) Udacity_CUSTOMERS_052018.csv
3.) Udacity_MAILOUT_052018_TEST.csv
4.) Udacity_MAILOUT_052018_TRAIN.csv

The information files are:
1.) DIAS Information Levels - Attributes 2017
2.) DIAS Attributes - Values 2017

The datasets generally represents attributes of each individual for each row.

Data Pre-processing

1.) Assess missing data in columns and rows
We start by assessing the dataset of Udacity_AZDIAS_052018.csv. To start assessing the missing data in columns, columns with missing values are identified. There are 2 types of missing data in the columns of the azdias dataset. The first type of missing values are values in columns with key codes that corresponds to 'unknown' or 'no_transaction_known'. The second type of missing values in columns occur simply because no values has been entered at all, it has been left blank.

The first type of missing values in columns is addressed first. We iterate through the 'DIAS Attributes - Values 2017' file to identify attributes with key codes that corresponds to 'unknown' or 'no_transaction_known' description. These attributes and its key codes are then extracted into a dictionary named missing_keys_dict. Additionally, some column attributes in the azdias dataset, not described in the 'DIAS_Attributes_Values' file but having similar attribute properties to other columns in the azdias dataset are engineered to have similar missing value key codes. Next, the missing values are converted to numpy nan.

The second type of missing value is addressed next. These missing values are defined as the initial missing value. The azdias dataset is iterated through to return a list of initial missing values for every column.

Finally a data frame showing the initial missing, final missing and percentage of missing values for every column is created. The figure below shows the top 30 columns with the highest percentage of missing values sorted in descending order:

	Attribute	initial_missing	final_missing	%_missing
7	ALTER_KIND4	890016	890016	99.9
349	TITEL_KZ	73499	889061	99.8
6	ALTER_KIND3	885051	885051	99.3
33	D19_BANKEN_LOKAL	0	874745	98.2
5	ALTER_KIND2	861722	861722	96.7
71	D19_TELKO_ANZ_12	0	857990	96.3
43	D19_DIGIT_SERV	0	857661	96.2
41	D19_BIO_OEKO	0	854074	95.8
79	D19_TIERARTIKEL	0	852220	95.6
63	D19_NAHRUNGSERGAENZUNG	0	852176	95.6
47	D19_GARTEN	0	851626	95.6
60	D19_LEBENSMITTEL	0	837914	94.0
95	D19_WEIN_FEINKOST	0	836142	93.8
28	D19_BANKEN_ANZ_12	0	831734	93.3
45	D19_ENERGIE	0	829857	93.1
72	D19_TELKO_ANZ_24	0	826208	92.7
37	D19_BANKEN_REST	0	821760	92.2
87	D19_VERSI_ANZ_12	0	821289	92.2
40	D19_BILDUNG	0	813156	91.2
4	ALTER_KIND1	810163	810163	90.9
38	D19_BEKLEIDUNG_GEH	0	809304	90.8
64	D19_RATGEBER	0	805071	90.3
66	D19_SAMMELARTIKEL	0	802085	90.0
29	D19_BANKEN_ANZ_24	0	794100	89.1
46	D19_FREIZEIT	0	790748	88.7
32	D19_BANKEN_GROSS	0	785351	88.1
88	D19_VERSI_ANZ_24	0	777037	87.2
67	D19_SCHUHE	0	773024	86.7
54	D19_HANDWERK	0	768381	86.2
78	D19_TELKO_REST	0	765973	85.9

Figure 1: Attributes with percentage of missing values

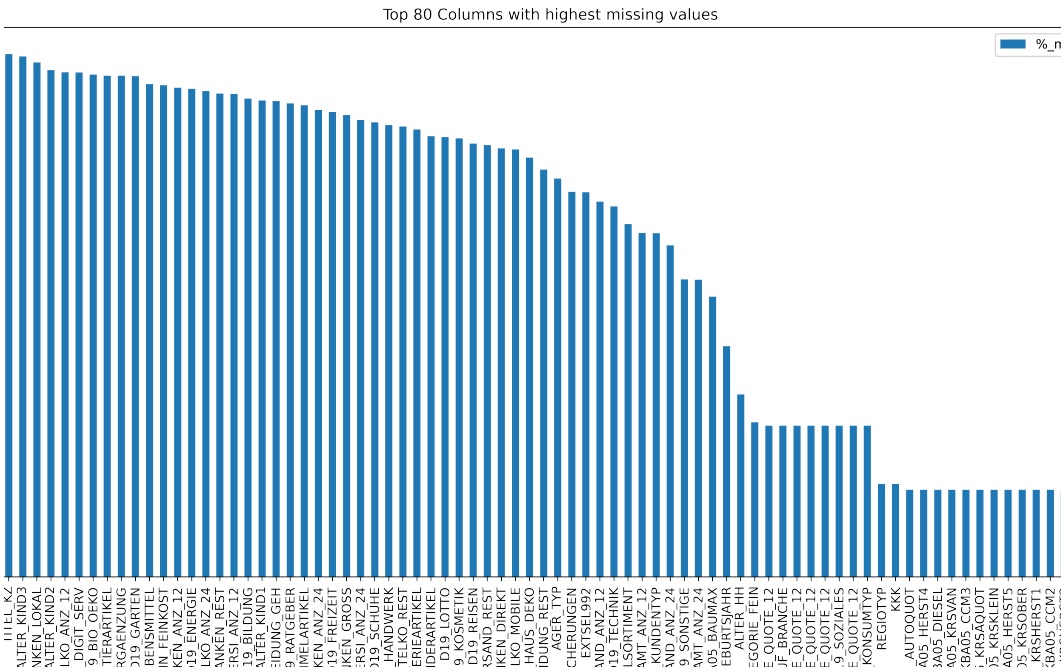

Figure 2: Top 80 columns with highest missing values

Based on Figure 2, it is noted that only several columns have missing values accounting for more than 30% of their data. A total of 54 columns have more than 30% missing values in their data. These 54 columns/attributes are removed from the data frame.

Besides this other columns/attributes are also checked and removed for the following reasons:

1.) identifier that does not contribute to analysis
2.) too many zero values
3.) too many categories
4.) repeat of other attributes
5.) attribute is unknown

The next step is then to assess the number of missing values in rows. In Figure 3, the majority of rows have less than ~20 attributes with missing values.

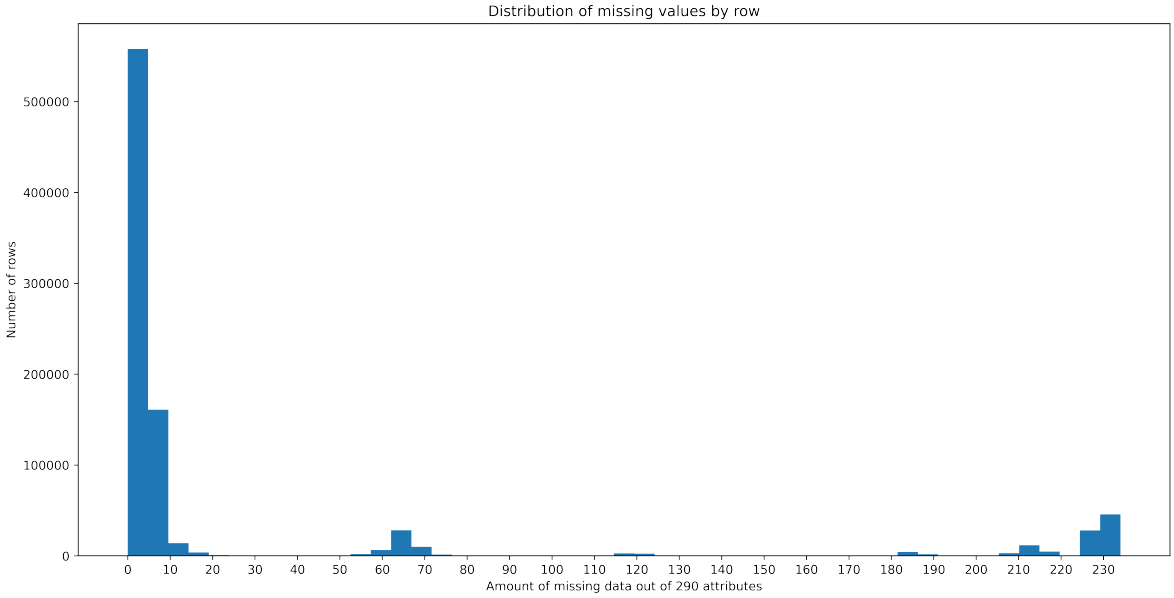

Figure 3: Distribution of missing values by row

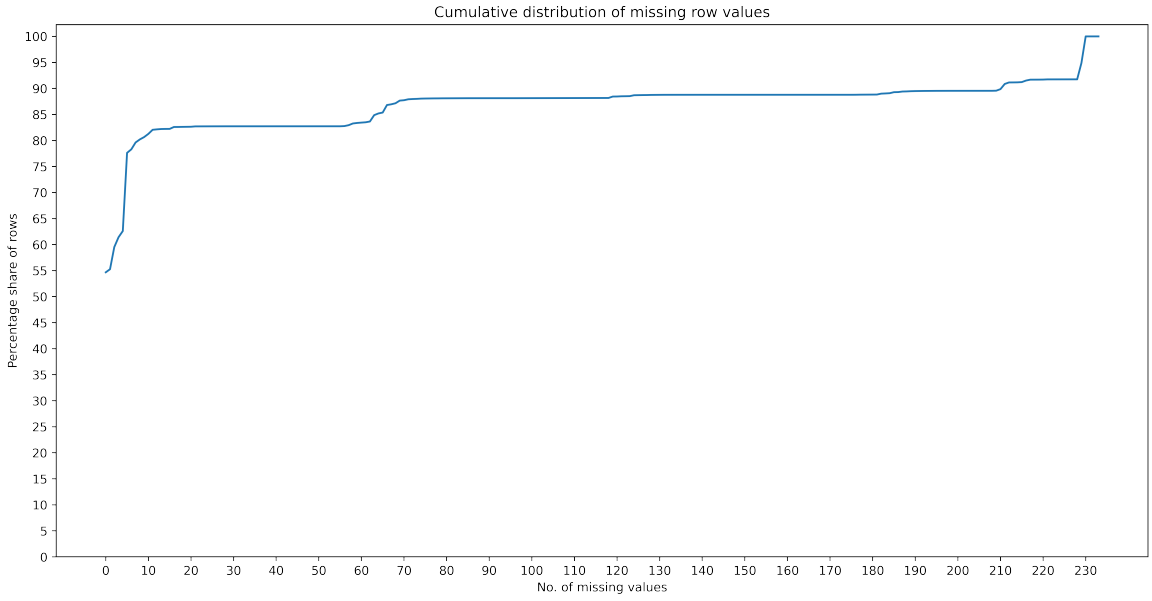

Figure 4: Cumulative distribution of missing row values

```
Percentage of data kept:
 0     54.673869
1     55.268446
2     59.520478
3     61.413836
4     62.604337
5     77.622722
6     78.302576
7     79.619197
8     80.206144
9     80.653059
10    81.269741
11    82.056639
12    82.139447
13    82.203067
14    82.220684
15    82.227753
16    82.594665
17    82.599378
18    82.617218
19    82.629561
20    82.640782
21    82.709564
22    82.713491
23    82.715286
24    82.718652
25    82.722131
26    82.724711
27    82.727741
28    82.727853
55    82.731668
Name: n_missing, dtype: float64
```

Figure 5: Percentage of row data kept

Based on Figure 4 and 5, up to ~82% of the row data is kept if only rows with less than 26 missing rows are retained. This is the threshold by which row data will be retained.

Upon completing column and row data drops, we perform feature engineering of the azdias and customer dataset.

Feature engineering is then performed on 6 attributes - CAMEO_INTL_2015, EINGEFUEGT_AM, OST_WEST_KZ, PLZ8_BAUMAX, PRAEGENDE_JUGENDJAHRE and WOHNLAGE.

The remaining rows with missing values is then imputed in the next section.

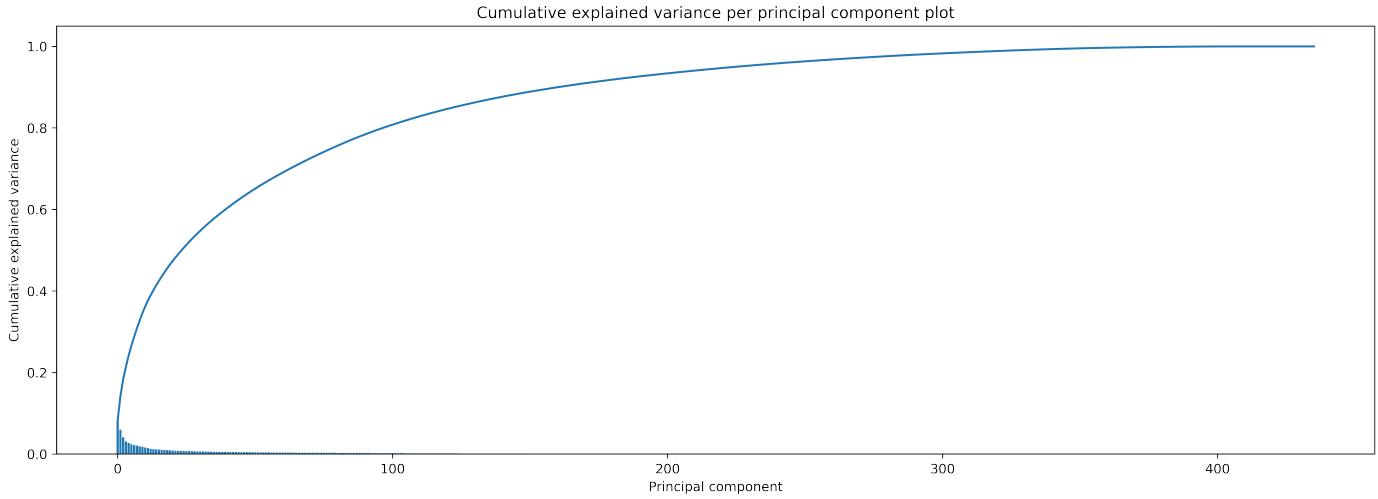

Figure 6: Cumulative explained variance per principal component plot

2.) Impute missing data. Transform and standardise features.

For the remaining missing values, value imputation are performed according to the type of attribute. The imputation performed for the types of attributes are as follows:

a.) skewed continuous - Attributes that are of continuous nature are checked for its skew value. The pandas 'skew' function is used to identify attributes with skew value of more than 1. A pipeline is then created where these skewed continuous attributes are undergo processing, first being log transformed, then median imputed and finally scaled with standard scaling.

b.) binary - Attributes that are of binary nature are imputed based on the most-frequent strategy.

c.) categorical - Attributes that are of categorical nature are processed with a custom transformer.

d.) numerical - Attributes that are of numerical nature are median imputed and then scaled with standard scaling.

A transformer object is used to combine all the feature transformation pipelines above.

Customer Segmentation

1.) Principal Component Analysis

Principal Component Analysis is used for dimensionality reduction. A plot of cumulative explained variance per principal component plot is produced. From

the plot, it is possible to determine that the explained variance value drops off to almost 0 around the 170th component. Here, the cumulative explained variance accounts for 91% of the data variance from the total of 436 features.

By further mapping the PCA components to attributes, we can determine that the top 3 components explains around 18.3% of the data variance.

The first component explains 8.0% of the data variance and is influenced positively by the attributes CAMEO_DEUG_2015, WEALTH, HH_EINKOMMEN_SCORE, KBA13_ANTG3 and ANZ_HAUSHALTE_AKTIV. Thus the first component is associated with status, class in society, household income, number of people in a household building.

The negative attributes influencing this component is KBA05_AUTOQUOT, KBA05_GBZ, KBA05_ANTG1, KBA13_ANTG1 and MOBI_REGIO. These attributes are associated with this component. Hence, targets with a few cars per household, high household units per building (e.g. apartments & flats), low or zero home ownership, high moving patterns (because of rental) are negative attributes influencing component 1.

The second component explains 6.0% of the data variance and is associated with KBA13_HERST_BMW_BENZ, KBA13_SEG_OBEREMITTELKLASSE, KBA13_MERCEDES, KBA13_BMW, KBA13_SITZE_4. These attributes relates to expensive and luxury cars owned. Hence, customers with expensive cars are more likely to be a customer.

The attributes negatively associated with the second component are KBA13_SEG_KLEINWAGEN, KBA13_HALTER_20, KBA13_ALTERHALTER_60, KBA13_KMH_140_210 and KBA13_SITZE_5. The second component is negatively influenced by attributes of small car ownership, the share of car owners is below 21 years old, share of car owners is between 46 and 60 years old, the car owned is a low speed vehicle and number of vehicle with 5 seats.

The third component explains 4.0% of the data variance and is positively associated with ALTERSKATEGORIE_GROB, FINANZ_VORSORGER, KOMBIALTER, SEMIO_ERL and SEMIO_LUST. These attributes are associated

with age, financial preparation, inclination to attend events and inclination to be sensual minded.

The features negatively associated with this component are FINANZ_UNAUFFAELLIGER, SEMIO_REL, SEMIO_PFLICHT, FINANZ_SPARER and GENERATION. These attributes are associated with unremarkable financial state, inclination to be religious, inclination to be dutiful, money saver habits and generation of the customer.

2.) KMeans Clustering

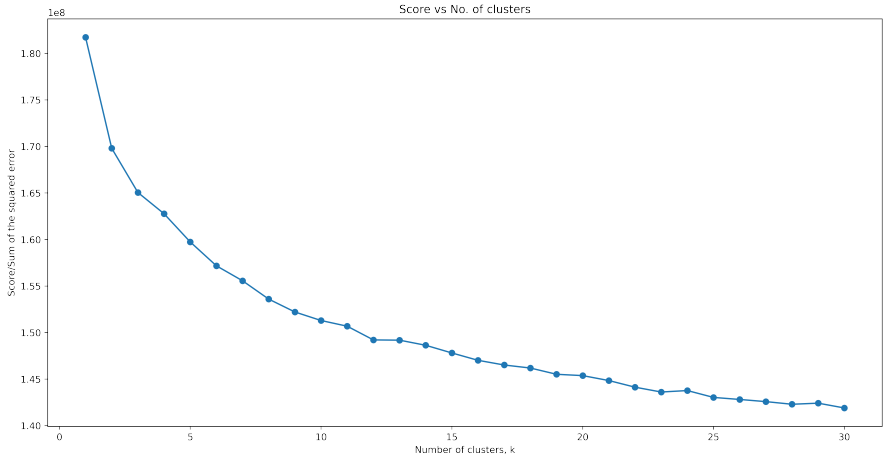

Figure 7: Sum of squared error vs number of clusters

To perform KMeans clustering, an optimal value of 'k' clusters is required. To determine this k-value, an elbow plot will be created. The sum of the squared value roughly plateaus when an optimal number of clusters have been added. This is determined to be at k=12 cluster.

After running the azdias and customers dataset through the clustering pipeline, a plot of percentage share for each cluster is produced for both the azdias (general population) and customers dataset. The resulting plot is as follows:

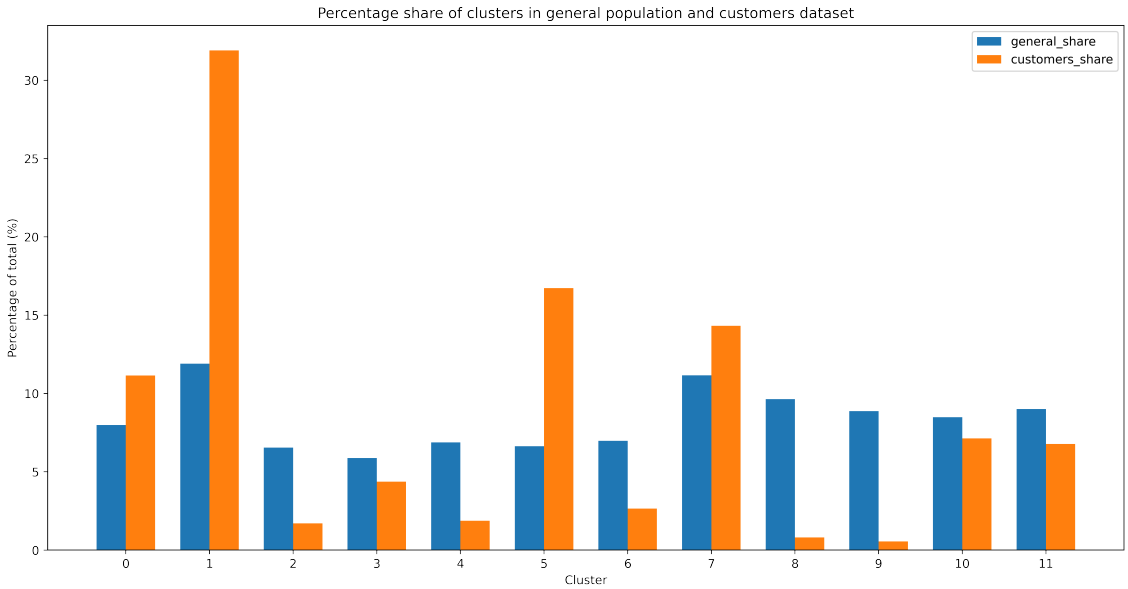

Figure 8: Percentage share of clusters in general population and customers dataset.

From Figure 8, the clusters are represented at a different percentage for the general population and customers dataset. This indicates that a comparison of the percentage representation between the general population and customers dataset can yield insight into the cluster that positively and negatively represents each dataset. Hence, a plot of the percentage difference in total share for general population and customers dataset is created in Figure 9.

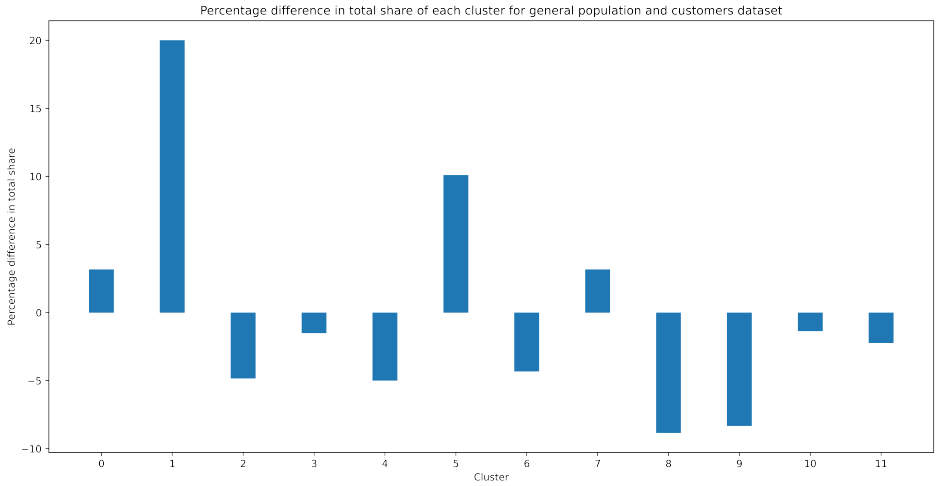

Figure 9: Percentage difference in total share of each cluster between general population and customers dataset

From Figure 9, a positive percentage difference means an over-representation of the customers dataset (compared to the general population dataset), while a negative percentage difference means an under-representation of the customers dataset. The diagram shows prominently that cluster 1 and 5 are over-represented while cluster 8 and 9 are under-represented. The over-represented clusters forms the target group of the marketing campaign and the under-represented clusters is inversely true.

By mapping the clusters back to the attributes of the dataset, an attribute list contributing to the clusters can be studied.

	positive_cluster1	positive_cluster2	negative_cluster1	negative_cluster2
SEMIO_KULT	3.390980	3.708336	5.169783	5.455969
D19_GESAMT_ONLINE_QUOTE_12	1.188808	3.179372	0.700759	2.889749
SEMIO_LUST	5.487079	4.868135	3.194223	2.848390
FINANZ_SPARER	1.296134	2.154606	4.098786	4.425177
FINANZ_MINIMALIST	4.428936	3.663659	2.276622	1.527996
FINANZ_UNAUFFAELLIGER	1.845222	2.661640	4.043443	3.758124
GENERATION	6.014793	7.011037	8.612618	8.632926
SEMIO_TRADV	2.666127	3.619860	5.305816	5.333789
SEMIO_PFLICHT	2.915480	3.680781	5.880629	6.066300
EINGEZOGENAM_HH_JAHR	1999.942163	2002.819713	2005.588961	2006.930659
FINANZ_ANLEGER	1.706023	1.953474	4.086049	3.982268
HH_EINKOMMEN_SCORE	3.159838	2.922414	4.394139	5.692894
D19_VERSAND_ONLINE_QUOTE_12	0.928531	2.932432	0.561418	2.597041
WEALTH	2.248932	2.121247	2.701643	4.465122
SEMIO_ERL	5.409487	4.694577	3.471336	3.299261
ORTSGR_KLS9	4.541447	6.223346	4.154017	6.544858
SEMIO_REL	2.817785	3.462382	5.560400	5.873922
FINANZ_VORSORGER	4.636497	3.805388	2.258766	2.228924
CAMEO_DEUG_2015	3.776982	3.495023	4.588467	7.587349
SEMIO_MAT	2.835536	3.681739	4.987129	5.056515
SEMIO_RAT	2.878243	3.638527	5.084416	5.262348

Table 1: Attributes for over-represented and under-represented clusters

From the table, the attributes of a target customer for the mail order company can be deduced. Attributes of target to be avoided by the company can also be deduced the same way.

Based on inference of the above data, the positively interested clusters (over-represented) are those that have the following attributes:

- High affinity to be culturally minded
- Low affinity to be sensual minded
- A money saver
- Has low finance interest/debt
- Has a good/remarkable financial standing
- Coming of age in the 1970s and 1960s
- Traditional minded
- Dutiful
- Is an investor or have investments
- Has a high household net income
- Wealthy
- Less inclined to events
- Religiously inclined
- Has financial planning
- Are from the established middle-class and consumption middle-class
- Highly material minded
- Very rational-minded

The negatively interested clusters (under-represented) are those that have the following attributes:

- Not culturally minded
- High affinity to be sensual minded
- Does not have money saving habits
- Has high finance interest/debt
- Has low financial standing
- Coming of age in the 1980s
- Not traditional minded
- Not dutiful
- Does not invest or have investments
- Has a low household net income
- Poor
- Inclined to events
- Not religious
- No financial planning
- Are from the lower middle-class and working-class
- Less material minded
- Not rational-minded

Supervised Learning Model

Through unsupervised learning, the attributes of a most likely customer has been identified. Now supervised learning is applied by building a prediction model that predicts whether a target will be a customer. The MAILOUT_TRAIN dataset will be trained on the model.

With the supervised learning, the pipeline consists of a column transformer and a classifier. The benchmark model will be based on a Logistic Regression classifier. The other classifiers chosen are the RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier and XGBClassifier. Ensemble classifiers build on decision-tree model such as RandomForest, AdaBoost and GradientBoosting often improves predictive accuracy by reducing the variation error of unstable classifiers. These classifiers are run on default parameters and the results are then compared to obtain the best classifier.

To evaluate the classifier, we plot the learning curve of the model based on the 'ROC_AUC' score as more data is added to training. This would enable us to determine the classifier that would improve in performance as more data is added and whether which classifier has a high bias or high variance problem.

We will also evaluate our classifiers of choice against the Logistic Regression classifier, which is our benchmark classifier.

The plots of the learning curve for the classifiers are as follows:

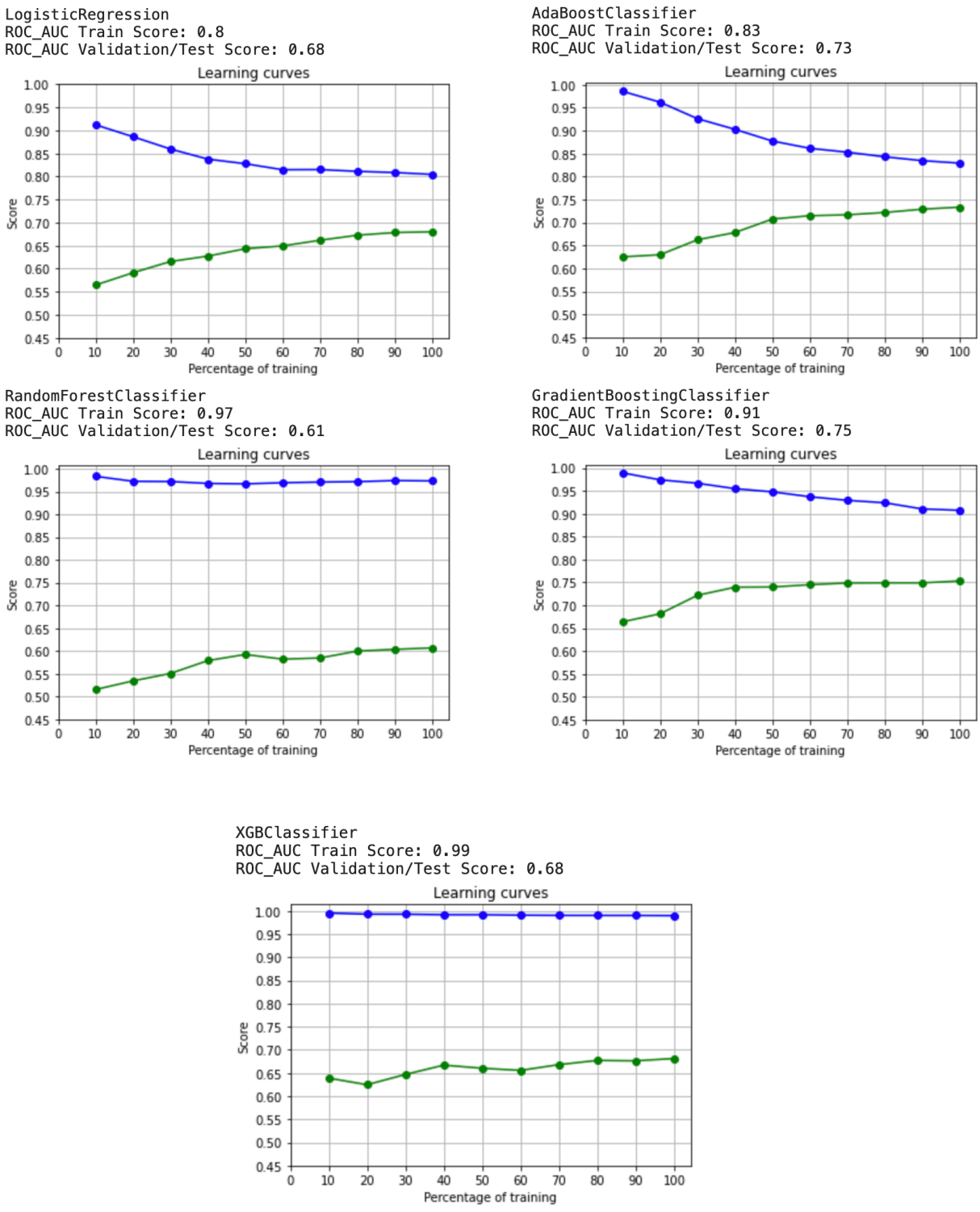

Figure 10: Plots of Learning Curve by Classifier

From these plots, we can see that the RandomForest Classifier and XGB Classifier are a high bias classifier as they both overfit the data.

The AdaBoost Classifier and GradientBoosting Classifier does slightly better as the training and validation curve converges with the addition of data. Both classifiers shows signs that they will converge even more with additional training data. In this case, we select the GradientBoosting Classifier as it has the best validation score.

Evaluating the Gradient Boosting Classifier against the Logistic Regression Classifier, the initial train and test score of the Gradient Boosting Classifier is significantly higher than the benchmark train and test score for the Logistic Regression Classifier.

The GradientBoosting classifier is then parameterised with Grid Search to determine the optimal parameters that would return the best validation score. The optimal parameters for the GradientBoosting classifier is learning_rate: 0.1, max_depth: 3, min_samples_split: 4, n_estimators: 100, with an ROC score of 0.8981. The resulting final validation score of 0.8981 by the Gradient Boosting Classifier shows an even bigger improvement compared to the benchmark test score of 0.68 by the Logistic Regression Classifier.

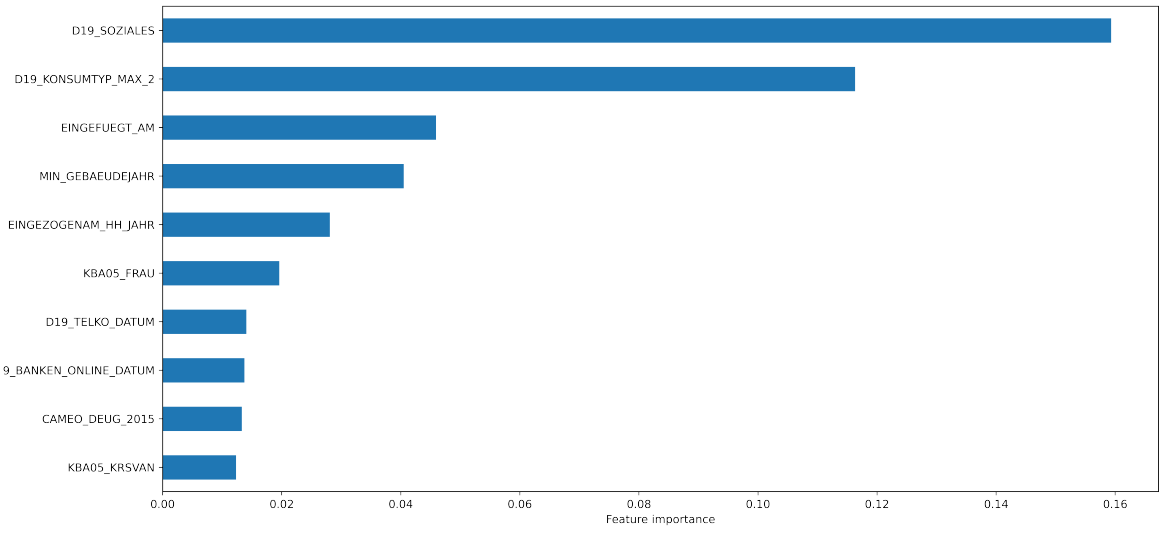

Figure 11: Feature importance plot

Lastly, the estimator is run to return the 'feature_importances_' attribute, it is determined that the estimator relies largely on the attribute of 'D19_SOZIALES' and 'D19_KONSUMTYP_MAX_2'. The first attribute is not explained in data provided while the second attribute indicates consumption type. However, with

an educated guess, it is possible that the first attribute indicates whether the individual is socially active.

Kaggle Competition

Finally, the supervised model is used to predict the response of customers based on the MAILOUT_TEST dataset. The resulting score is submitted to the Kaggle competition. The score obtained was 0.78494.

Conclusion

To summarise, the dataset was first pre-processed to handle missing values and a host of other problems, such as attributes not having a description in the data provided. The dataset was then imputed, transformed and scaled with the Column Transformer pipeline. Then unsupervised learning was performed by principal component analysis to reduce the dimensionality of the dataset. Next, Means clustering was used to determine clusters that explain the attributes of a potentially positive target customer. In the supervised learning section, the GradientBoosting classifier was selected as the best classifier. The classifier was then parameterised to obtain the optimal parameters with Grid Search. The supervised model is then applied to the test data to obtain the final score, which was submitted to Kaggle.
Several improvements can be made to the analysis of this project. Among them includes dropping more columns with missing values, perform feature engineering for more columns, testing different imputation and scaling methods in the Column Transformer pipeline, selecting different classifiers and possibly using over-represented and under-represented cluster attributes datasets only for supervised learning.