Arvato Bertelsmann Customer Analysis

Udacity Data Scientist Nanodegree Capstone Project

By Mei Eisenbach

Introduction

In this project, demographics data for customers of a mail-order sales company in Germany, was analyzed and compared against demographics information for the general population. The data was provided Bertelsmann Arvato Analytics.

Customer segmentation was used to identify the parts of the population that best describe the core customer base of the company. In the second part of the analysis, a dataset with demographics information for targets of a marketing campaign for the company was used to model and predict which individuals are most likely to convert into becoming customers for the company.

For the customer segmentation, average within-cluster distance was used to help determine the number of clusters to produce. The clusters were then analyzed and evaluated on how informative they are on the customer base. For the marketing campaign, the predictions were tuned locally using roc-auc and final scoring was done by uploading to the Kaggle competition.

Customer Segmentation - Analysis

Data files:

- 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).

Along with the data, two files describing the data were provided:

- DIAS Attributes - Values 2017.xlsx: a data dictionary of the columns including description, possible values, and what the values mean.
- DIAS Information Levels - Attributes 2017.xlsx: indicates the information level for each column (Person, Household, Building, etc)

Using the two description files, a features.csv file was manually created containing the attribute, information level, data type, missing or unknown values.

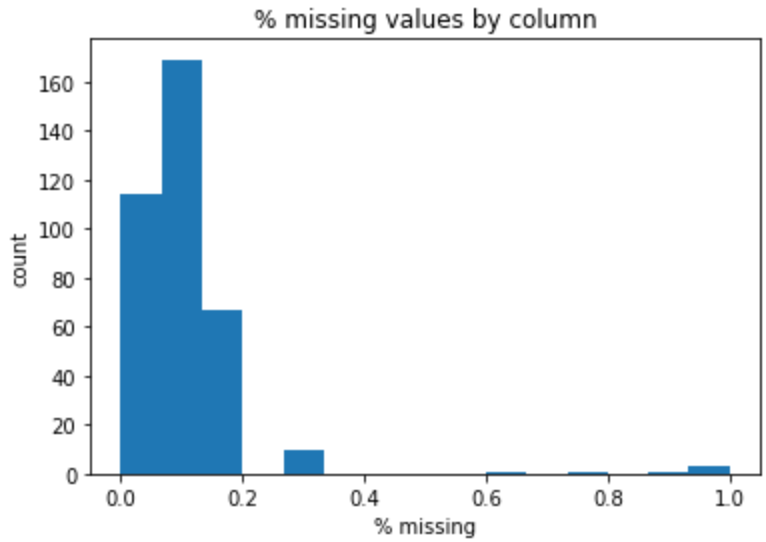
Missing Features

39 features were in the demographics data that were not described in the data dictionary. These were dropped from the customer segmentation datasets. 5 features were in the data dictionary but were not in the data. Ideally, we would have gone back to the company and asked for clarification.

Missing Values

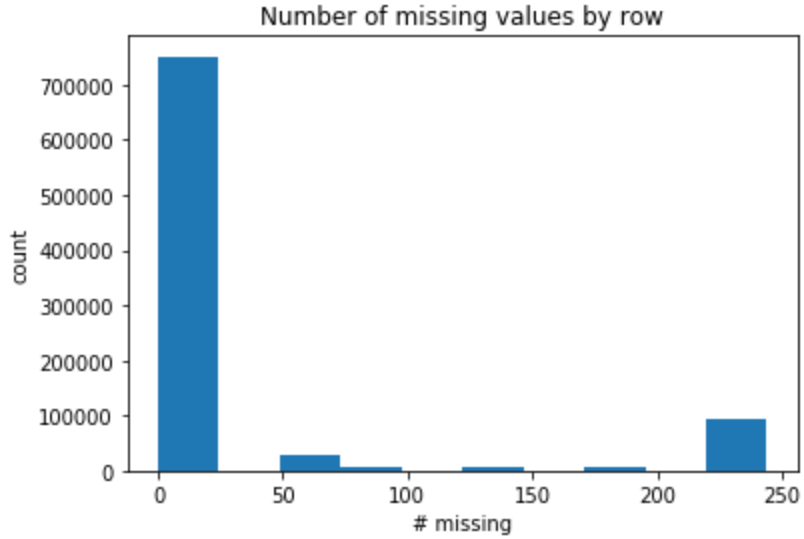
The population data was analyzed for missing values. The first step was to convert missing values to Nan's using the feature information data.

Missing values by column

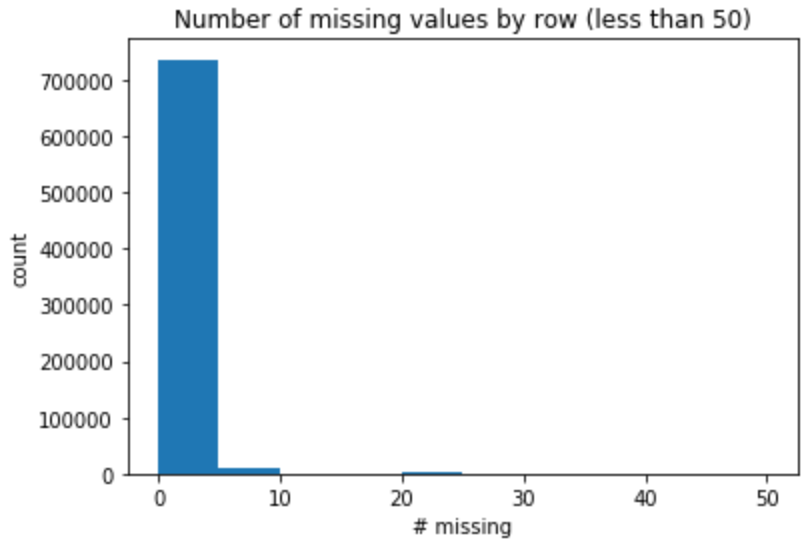


Most columns have less than 20% missing values so this was set as the threshold. The names of the columns to be dropped were saved to a list and the columns were dropped from the population data.

Missing values by row



Most rows have less than 25 missing values.



Zooming in to see the counts for less than 50 missing, we see that 10 is a good threshold.

Customer Segmentation - Methodology

Data Cleaning

A data cleaning script was created which performed the following items.

Missing Values

- Columns with greater than 20% missing were dropped
- Rows with greater than 10 missing values were dropped

Re-encoding and Engineered Features

- Columns starting with D19_: 10's (no transactions) were re-coded as 0's (no transactions). Only some of the D19 columns had 10's. Re-coding was done to be consistent with the other D19 columns.
- OST_WEST_KZ was dummy encoded
- PRAEGENDE_JUGENDJAHRE was split in decade and movement
- CAMEO_INTL_2015 was split into wealth and life_stage

Dropped features

- Features not found in the data dictionary
- CAMEO_DEU_2015, LP_FAMILIE_FEIN, LP_STATUS_FEIN were dropped since the information was also in the _GROB (rough) versions of the the feature.
- KBA_ variables were dropped because they caused problems with PCA

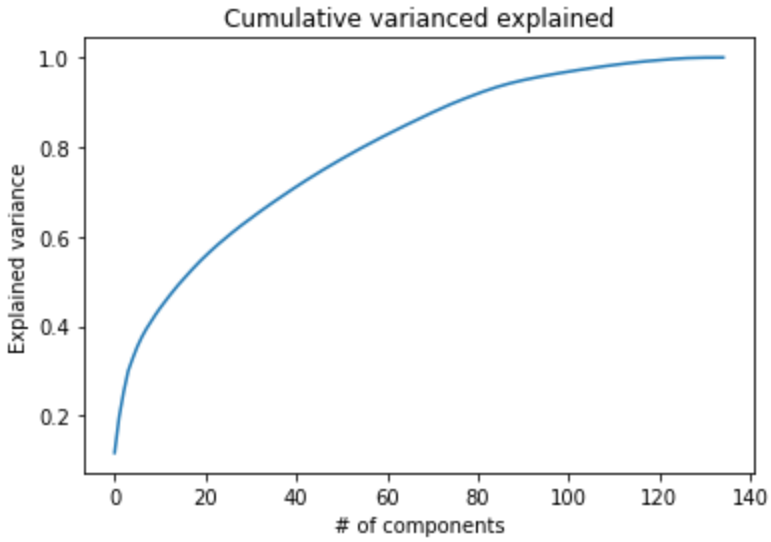
For the clustering task, it was import that all the features were well understood so that the clusters could be interpreted. For this reason columns not found in the given data dictionary were dropped.

Data Preprocessing

The data was imputed using the median to remove any remaining missing values and scaled.

PCA

Principal Component Analysis (PCA) was performed because of the large number of features and the fact that many of them provide almost the same information. For the customer segmentation task, a clustering algorithm will be used and they run very slowly with large numbers of features.



From the cumulative variance explained graph, it was determined that at 90% the gain in explained variance levels off. This occurs at 76 principal components.

PCA Interpretation

First principal component
The first principal component accounts for 11.67% of the variance explained.

Highest weighted features:

- D19_VERSAND_ANZ_24 is transaction activity MAIL-ORDER in the last 12 months (6 = very high activity)
- ONLINE_AFFINITAET is online affinity (5 = highest)

- D19_GESAMT_ANZ_24 is mail-order transaction activity TOTAL POOL in the last 24 months (6 = very high activity)

Lowest weighted features:

- HH_EINKOMMEN_SCORE is estimated household net income (1 = highest income group)
- CAMEO_DEUG_2015 is CAMEO major classification 2015 (1 = upper class)
- wealth was derived from CAMEO_INTL_2015 (1 = wealthy)

This component is an indicator of purchasing activity and wealth. Note, the indicators for wealth are all inverted with 1 indicating highest wealth.

Second principal component
The second principal component accounts for 7.97% of the variance explained.

Highest weighted features:

- SEMIO_REL is affinity indicating in what way the person is religious (1 = highest affinity)
- decade was derived from PRAEGENDE_JUGENDJAHRE (1 = 40's, 15 = 90's)
- FINANZ_SPARER is financial typology: money saver (1 = very high)

Lowest weighted features:

- ALTERSKATEGORIE_GROB is age major category (4 = > 60 years)
- FINANZ_VORSORGER is financial typology: be prepared (1 = very high)
- FINANZ_MINIMALIST is financial typology: low financial interest (1 = very high)

This component is a an indicator of lack of religiousness, youth, free spending.

Third principal component
The Third principal component accounts for 5.65% of the variance explained.

Highest:

- EWDICHTE is population density (6 = densest)
- PLZ8_ANTG3 is number of 6-10 family houses in the zipcode (3 = high share)
- ORTSGR_KLS9 is size of the community, classified number of inhabitants (9 = > 700.000 inhabitants)

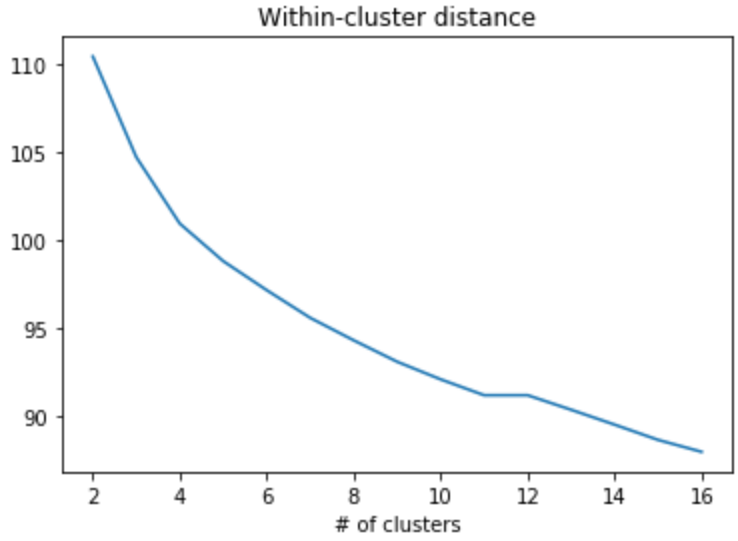
Lowest:

- SEMIO_KULT is affinity indicating in what way the person is cultural minded (1 = highest affinity)
- SEMIO_REL is affinity indicating in what way the person is religious (1 = highest affinity)
- PLZ8_ANTG1 is number of 1-2 family houses in the zipcode (3 = high share)

This component is an indicator of population density and cultural minded/religiousness.

Implementation

Clustering using k-means

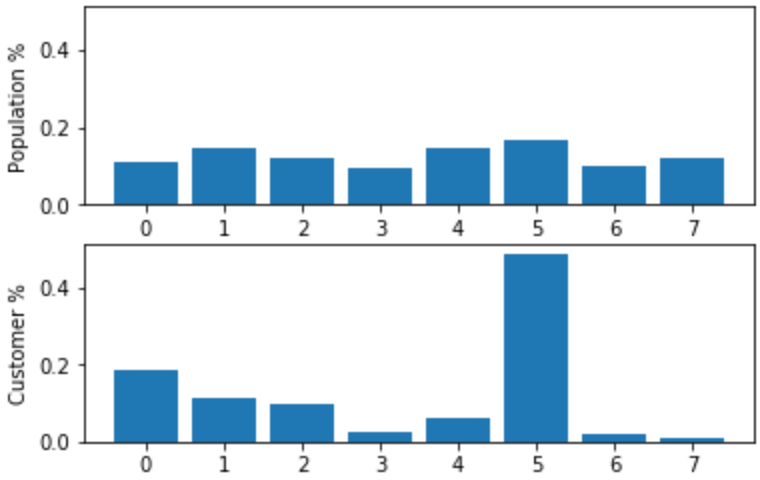


To determine the optimal number of clusters, the elbow method was used. The average within-cluster distance across variance k clusters were calculated and plotted. It was determined that 8 clusters would be a good starting point.

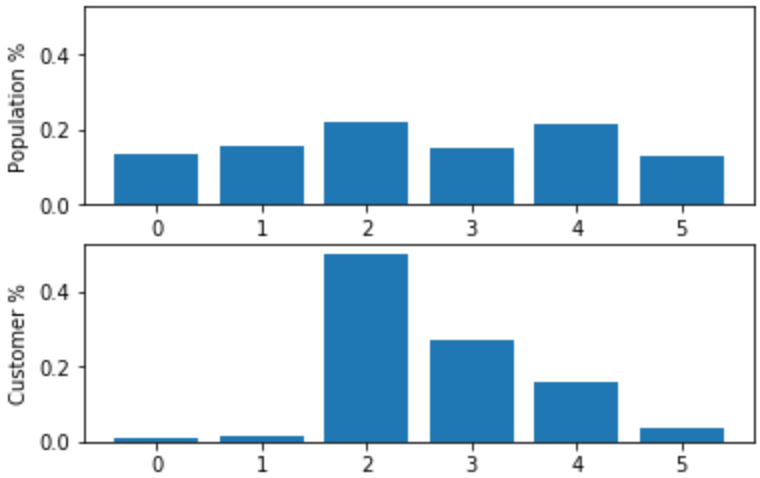
Refinement

At this point, the preprocessing (imputation and scaling), pca, and estimator were combined into a pipeline. The benefit of a pipeline is that it saves the preprocessing objects with the model.

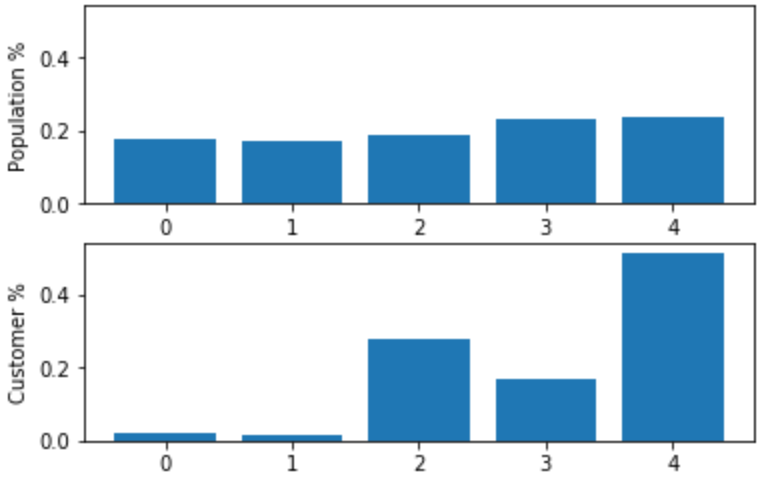
The clustering pipeline was fitted on the population data and then used to cluster the population and customer data.



Starting with 8 clusters, it was found that one cluster was overrepresented in the customer data. It was suspected that some of the clusters could be combined without losing information. Unfortunately, the dataset is too large to use a hierarchical cluster algorithm. With that type of algorithm, you are able to see which groups merge as the number of clusters decreases.



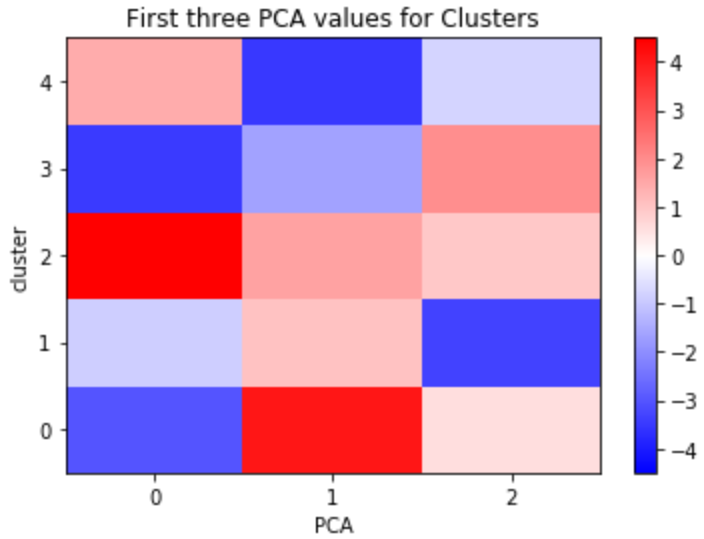
At 6 clusters, there were also one overrepresented cluster. There are three under-represented clusters.



At 5 clusters, you can see that there is one overrepresented group and two underrepresented groups in the customer data.

Customer Segmentation - Results

Cluster Analysis



A heatmap of the clusters and their PCA values shows that at five clusters the clusters are easily distinguishable from one another using the first three PCA components.

PCA summary of first three values

- PCA 0 is an indicator of purchasing activity and wealth
- PCA 1 is a indicator of lack of religiousness, youth, free spending
- PCA 2 is an indicator of population density and urban-ness

Cluster 4: overrepresented
PCA values: 1.46352879, -3.49229666, -0.74227797

The most distinguishing feature of Cluster 4 is the very low value for pca 1. This indicates this cluster is more religious, older and savers. It also as a positive value for pca 0 which is an indicator of purchasing activity and wealth.

Cluster 0: underrepresented
PCA values: -3.01992766, 4.10241541, 0.57541029

Cluster 0 has a very low value for pca0 and a very high value for pca1. This cluster has low purchasing activity and wealth. They are also not very religious but are young and free spending.

Cluster 1: also underrepresented
PCA values: -0.84571908, 1.06199565, -3.32130468

For cluster 1, the most distinguishing feature is the very low pca2 value. This cluster has low population density and cultural minded/religiousness.

Evaluation

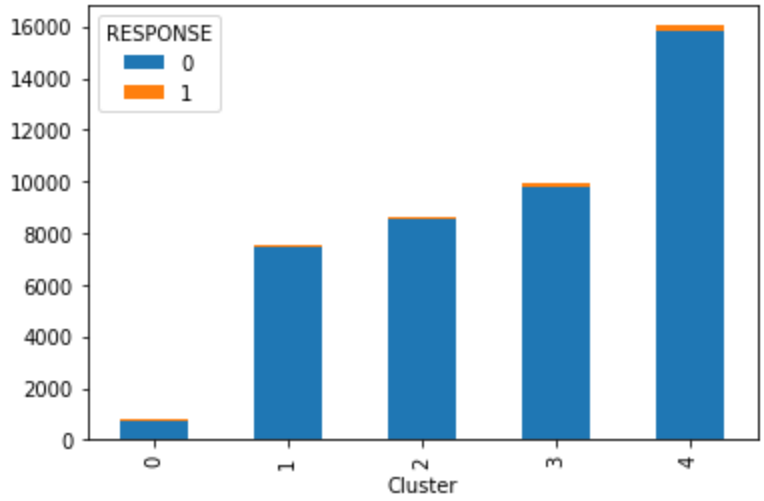
Metrics for unsupervised learning algorithms are not as clear-cut as for supervised learning model and are dependent on the goal of the analysis.

Here we are attempting to separate groups of individuals which are more likely to be customers so it's important for the clusters to show differences in the clusters between the population data and customer data. From the cluster comparison charts, it was clear which clusters were overrepresented and which ones were underrepresented.

Additionally, the clusters should be easily distinguishable from one another. Using only the first three principal components, each of the five clusters can be identified based on whether the values are higher or lower the average.

Clustering Marketing Data

The clustering model developed in the previous section was applied to the mailout data. The thought was that perhaps the response rate would be higher in the overrepresented group. The clustering pipeline which includes pca and creates 5 clusters was used to cluster the training data with the response variable removed.



The response rate for cluster 4 was 1.4% which is higher than all the other clusters and higher than the overall response rate of 1.2%. The difference in response rate would not enough to predict marketing responses. This strategy was then abandoned.

It's interesting to note the distribution of individuals chosen for the marketing campaign seem to agree mostly with the results of the segmentation analysis. 37.3% of the marketing campaign individuals were Cluster 4. This cluster is overrepresented in the customer data as compared to the population and are more likely to become customers. Only 1.8% of the campaign individuals were from Cluster 0, one of the underrepresented clusters.

Marketing Predictions - Analysis

Here is the second part of the project. Supervised training techniques were used to predict whether a individual would respond positively to a marketing campaign.

Data files:

- 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).

Imbalanced classes

The training data for the classification task consisted of 42,982 rows.

Response	*Count*	*% Total*

0	42430	98.8
1	532	1.2

The response column has 42,430 negative responses and only 532 positive. This is a incredibly imbalanced dataset and will affect the training of the classification model. This will also affect the choice of metrics.

Marketing Predictions - Methodology

Data Cleaning

Initially, the same cleaning function was used on the marketing data that was developed based on the the population demographics data but this led to too many of the positive response rows being dropped.

As a result, a separate analysis of the marketing training data was performed and a separate cleaning script was created with the following:

- The column threshold was found to be best at 22%
- Because of the dearth of positive responses, no rows were dropped.
- Re-encoding and Engineered features were the same as described above for the segmentation task
- Columns without descriptions were not dropped
- EINGEFUEGT_AM' (a datetime) was dropped
- ‘D19_LETZTER_KAUF_BRANCHE’ was dummied

The lack of description for certain columns is not a problem since the chosen algorithm, a neural network, is black box. The additional columns resulted in better classifier performance.

LNR is an identifier that should be dropped before classification but could not be included in the cleaning script because the values would be lost.

Data Preprocessing

A preprocessor pipeline was created which comprised of a numerical pipeline and categorical pipeline. For numerical columns, missing values were imputed with the median and then scaled. For categorical columns, missing values were imputed with the most frequent category (no scaling).

PCA was not performed as it's not necessary to reduce the number of features and might negatively impact performance.

Implementation

Classification Models

The best classifier was a LightGBM model. LightGBM is a newer algorithm. It is a gradient boosting framework that uses tree based learning. It differs from other tree based algorithms by growing trees vertically instead of horizontally. It's also faster than traditional tree based algorithms.

The other classifiers trained were: Keras sequential model, Gradient Boost, and AdaBoost. All of the models including LightGBM were chosen because they are suited for handling imbalanced data.

All models were evaluated using 5 fold cross validation using roc_auc. The final scoring done by Kaggle is also using roc_auc. This metric is appropriate for imbalanced data because it divides accuracy into sensitivity (true positive rate) and specificity (true negative rate). If only accuracy were to be considered, with a very imbalanced dataset, a classifier that predict the majority class would be score very well but not be useful.

Refinement

Hyperparameter Optimization was used to determine the best optimizer, learning rate, batch size, and class weights.

A preprocessing pipeline in a python file was created and shared between Jupyter notebooks to ensure consistency. Final versions of each model were saved to a pickle file.

Marketing Predictions - Results

Kaggle Competition

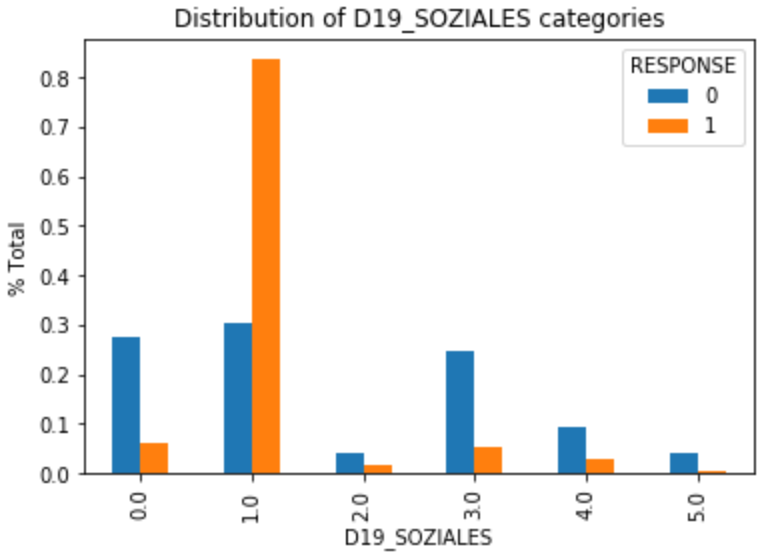
Scores were obtained after uploading submissions to the Kaggle competition (https://www.kaggle.com/c/udacity-arvato-identify-customers/leaderboard).

Model	Local score	Kaggle Score
Keras	0.65836	0.65842
Gradient Boost	0.76524	0.79327
AdaBoost	0.76238	0.79791
LightGBM	0.76294	0.80143

The LightGBM model achieved 9th place on the leaderboard out of 18 teams. All of the top ten scores were very close (within 0.007 of the top score).

Analysis

One of the benefits of using a tree-based model is that the model can give insight into the most import features. All three tree-based models (Gradient Boost, AdaBoost, LightGBM) had D19_SOZIALES as the most import feature.



D19_SOZIALES is not in the provided data dictionary. It appears to be categorical with values from 0-5. "sozial" means social so it's possible that the feature refers to social groups. 84% of individuals with response=1 are in category 1 of D19_SOZIALES as compared to 30% with response=0. This means that category 1 individuals are much more likely to respond positively to the marketing campaign.

Conclusion

For the customer segmentation, we trained a k-means model on the general population data and then used the model to cluster the customer data. The distributions of clusters was compared between the population data and the customer data. The hardest part of the analysis was handling the large amounts of data; both number of features and observations. PCA was useful and it was nice to see that the first three principal components were able to distinguish the five clusters effectively.

Technically speaking, additional analysis, for example silhouette graphs, could be done on the clusters. Practically speaking, quality of the clusters is very subjective. We would need to present our findings to the customer and get feedback.

In the marketing analysis, we used trained LightGBM classifier on the training dataset and used it to make predictions on the test data. Finding the best classification algorithm was the most challenging part of the analysis. It was a mistake to spending a lot of time tuning the first model, a neural network, before trying other algorithms.

The LightGBM mode could be improved by running more extensive hyper-parameter tuning. Given how close the top models were to one another, there isn't much room for additional improvement.

Code for this analysis can be found in its Github repo.

References

8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset
https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

Porto Seguro: balancing samples in mini-batches with Keras
https://imbalanced-learn.readthedocs.io/en/stable/auto_examples/applications/porto_seguro_keras_under_sampling.html#sphx-glr-auto-examples-applications-porto-seguro-keras-under-sampling-py

What is LightGBM, How to implement it? How to fine tune the parameters?
https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc

Customer Segmentation Report for Arvato Financial Solutions
https://towardsdatascience.com/customer-segmentation-report-for-arvato-financial-solutions-b08a01ac7bc0

Investigating Customer Segmentation for Arvato Financial Services
https://medium.com/@shihaowen/investigating-customer-segmentation-for-arvato-financial-services-52ebcfc8501