Capstone Project
Data Scientist Nanodegree

Tianxiang Ma
April 13th, 2019

Customer Segmentation Report
for Arvato Financial Services

Definition

Project Overview

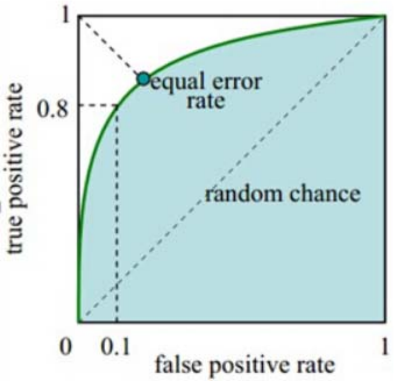
In the project, a mail-order sales company in Germany is interested in identifying segments of the general population to target with their marketing in order to grow. The objective is to describe the core customer base of the company, and to identify which individuals are most likely to respond to the campaign.

Problem Statement

- Analyze demographics data (provided by Arvato Financial Solutions, a Bertelsmann subsidiary) for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population.
- Use unsupervised learning techniques to perform customer segmentation, identifying the parts of the population that best describe the core customer base of the company.
- Use supervised learning techniques to predict customer responses in a marketing campaign for the mail-order company, which is also a Kaggle competition.

Scoring

The evaluation metric for this competition is AUC for the ROC curve, relative to the detection of customers from the mail campaign. A ROC, or receiver operating characteristic, is a graphic used to plot the true positive rate (TPR, proportion of actual customers that are labeled as so) against the false positive rate (FPR, proportion of non-customers labeled as customers).

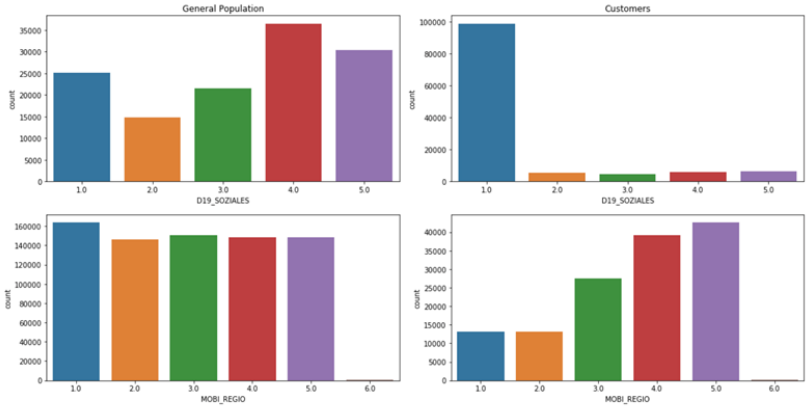


Analysis

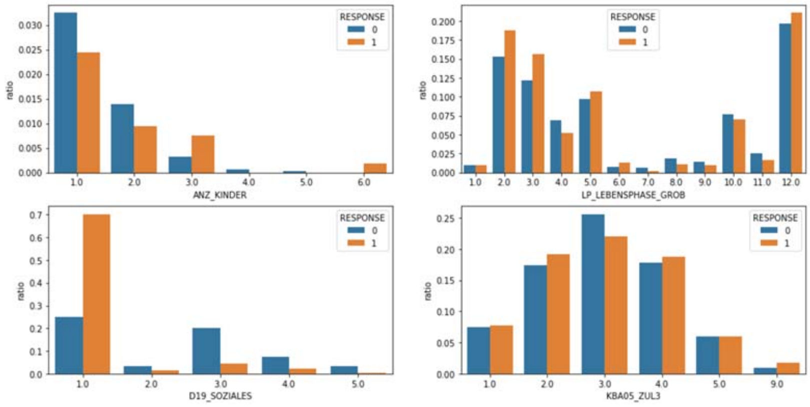
Data Exploration and Visualization

In this project we have 4 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).
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).

Comparing data between general population and company customers, there are some features which show a different distribution, such as D19_SOZIALES, MOBI_REGIO.

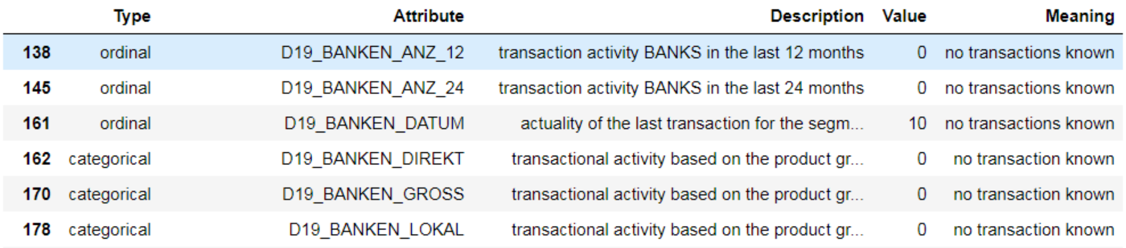


Also, distribution of some features from training dataset, like ANZ_KINDER, LP_LEBENSPHASE_GROB, D19_SOZIALES, KBA05_ZUL3, shows different distribution between individuals who respond to the campaign and who did not.



The two reference I have:
1. *AZDIAS_Feature_Summary.csv*: dictionary about feature Types and Unknown Values, however it covers only 85 features;
2. *DIAS Attributes - Values 2017.xlsx*: provides value meanings but no Type info.

To access missing data, I noticed there are quite a few features that have values (i.e.: -1, 0, 9, 10, etc.), meaning 'missing or unknown' **other than NaN**, which needs to be cleaned before statistics.



	Type	Attribute	Description	Value	Meaning
138	ordinal	D19_BANKEN_ANZ_12	transaction activity BANKS in the last 12 months	0	no transactions known
145	ordinal	D19_BANKEN_ANZ_24	transaction activity BANKS in the last 24 months	0	no transactions known
161	ordinal	D19_BANKEN_DATUM	actuality of the last transaction for the segm...	10	no transactions known
162	categorical	D19_BANKEN_DIREKT	transactional activity based on the product gr...	0	no transaction known
170	categorical	D19_BANKEN_GROSS	transactional activity based on the product gr...	0	no transaction known
178	categorical	D19_BANKEN_LOKAL	transactional activity based on the product gr...	0	no transaction known

In addition, we are facing some serious challenges:
1. Undocumented features: 90 still unknown after combining above two references;
2. Unknown data types: no idea whether given feature is numerical or categorical, except those 85 documented in *AZDIAS_Feature_Summary.csv*;

The incomplete dictionary requires manual editing later in implementation.

Algorithm and Techniques

PCA

Since there are hundreds of variables in the demographics dataset, it is necessary to reduce the dimension of the feature space before clustering, in order to reduce the complexity of the problem, and to draw conclusions about the underlying structure of the dataset.

Principal component analysis (PCA) will be applied in the project. It is used to decompose a multivariate dataset in a set of successive orthogonal components that explain a maximum amount of the variance.

Since using PCA on a dataset calculates the dimensions which best maximize variance, we will find which compound combinations of features best describe customers.

K-Means

To explore unlabeled dataset, we need to define groups in a way that objects in the same group are more similar to each other than those in other groups.

The K-Means algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares.

K-Means clustering algorithm is easy to understand and fast. Though it assumes clusters as convex and isotropic, and not stable in high dimensional space, we can alleviate this problem and speed up the computations from previous PCA dimension reduction algorithm.

XGBoost

For predictive model, I select XGBoost as a solution for this supervised learning problems.

XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data. The objective function of XGBoost is consist of two parts:

$$\text{obj} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i)$$

L is the training loss function, and Ω is the regularization term. The training loss measures how predictive our model is with respect to the training data. Regularization term controls the complexity of the model, and helps us to avoid overfitting.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

The specialty xgboost differs from other gradient boosting algorithm is that xgboost used a more regularized model formalization to control over-fitting, which gives it better performance.

Methodology

Data Pre-processing

1. Build a complete data dictionary

(1) Record all features needed:

Part1	features from ***AZDIAS_Feature_Summary.csv***
Part2	***DIAS Attributes - Values 2017.xlsx*** features that have specific Null values;
Part3	***DIAS Attributes - Values 2017.xlsx*** features with no specific Null values;
Part4	features in dataset but undocumented.

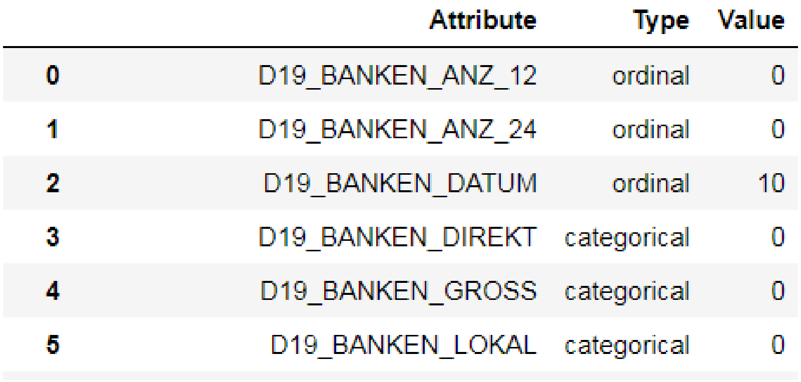
(2) Label data types for each feature:

i) **Part1** features are documented well.

ii) Also, I was able to manually add data types for **Part2** and **Part3** features according to reference ***DIAS Attributes - Values 2017.xlsx***.

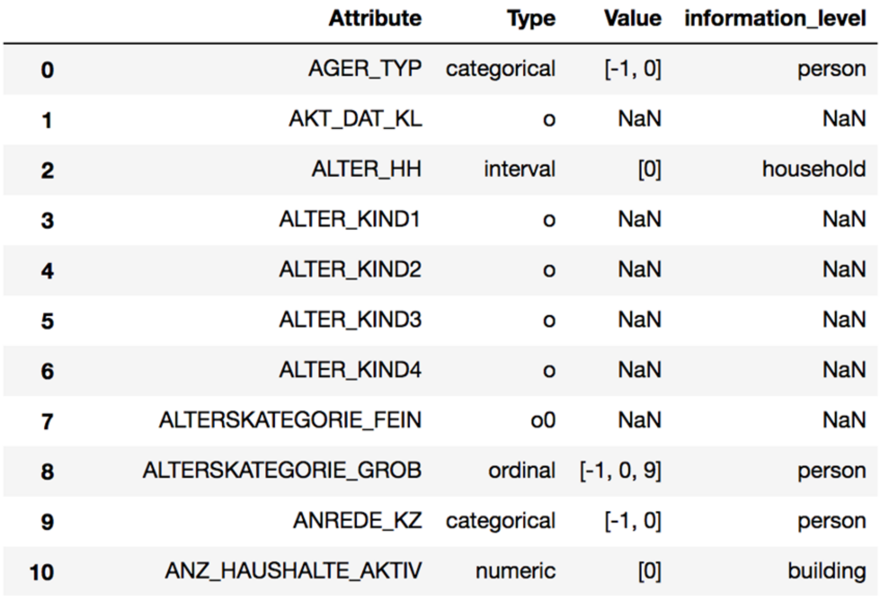
iii) For **Part4** features I had to guess and label them; Finally, I set almost all other features to be numerical but two: D19_LETZTER_KAUF_BRANCHE, D19_SOZIALES;

(3) document 'missing or unknown' values to **Part2** features:



	Attribute	Type	Value
0	D19_BANKEN_ANZ_12	ordinal	0
1	D19_BANKEN_ANZ_24	ordinal	0
2	D19_BANKEN_DATUM	ordinal	10
3	D19_BANKEN_DIREKT	categorical	0
4	D19_BANKEN_GROSS	categorical	0
5	D19_BANKEN_LOKAL	categorical	0

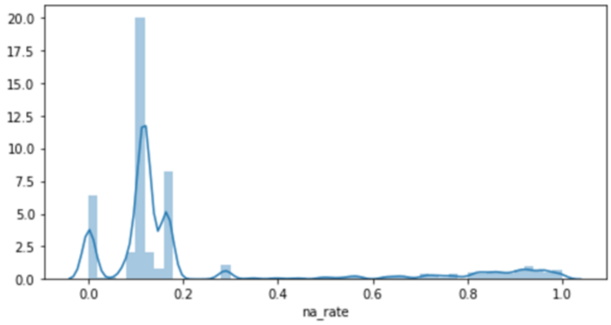
- Final dictionary file is displayed as follows:



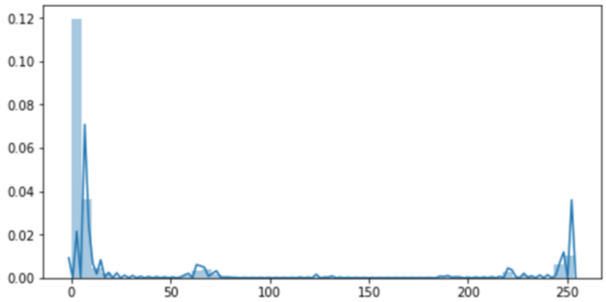
	Attribute	Type	Value	information_level
0	AGER_TYP	categorical	[-1, 0]	person
1	AKT_DAT_KL	o	NaN	NaN
2	ALTER_HH	interval	[0]	household
3	ALTER_KIND1	o	NaN	NaN
4	ALTER_KIND2	o	NaN	NaN
5	ALTER_KIND3	o	NaN	NaN
6	ALTER_KIND4	o	NaN	NaN
7	ALTERSKATEGORIE_FEIN	o0	NaN	NaN
8	ALTERSKATEGORIE_GROB	ordinal	[-1, 0, 9]	person
9	ANREDE_KZ	categorical	[-1, 0]	person
10	ANZ_HAUSHALTE_AKTIV	numeric	[0]	building

2. Accessing NULL values

(1) Replace 'missing or unknown' values with np.nan;



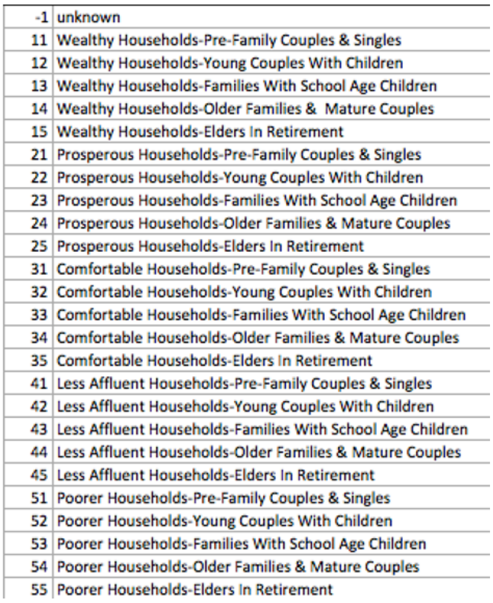
(2) For columns, we observe from above histogram that most features have less than 30% null values. So I dropped columns with null value rate greater than 30%.



(3) For rows, with more than 20 null values is reasonable to be considered as outliers. So I dropped rows with more than 20 null values.

3. Specific Feature Engineering

There is one feature that is neither categorical nor numerical: ***CAMEO_INTL_2015***, which can be divided into two ordinal features, wealthy level and age level.



-1	unknown
11	Wealthy Households-Pre-Family Couples & Singles
12	Wealthy Households-Young Couples With Children
13	Wealthy Households-Families With School Age Children
14	Wealthy Households-Older Families & Mature Couples
15	Wealthy Households-Elders In Retirement
21	Prosperous Households-Pre-Family Couples & Singles
22	Prosperous Households-Young Couples With Children
23	Prosperous Households-Families With School Age Children
24	Prosperous Households-Older Families & Mature Couples
25	Prosperous Households-Elders In Retirement
31	Comfortable Households-Pre-Family Couples & Singles
32	Comfortable Households-Young Couples With Children
33	Comfortable Households-Families With School Age Children
34	Comfortable Households-Older Families & Mature Couples
35	Comfortable Households-Elders In Retirement
41	Less Affluent Households-Pre-Family Couples & Singles
42	Less Affluent Households-Young Couples With Children
43	Less Affluent Households-Families With School Age Children
44	Less Affluent Households-Older Families & Mature Couples
45	Less Affluent Households-Elders In Retirement
51	Poorer Households-Pre-Family Couples & Singles
52	Poorer Households-Young Couples With Children
53	Poorer Households-Families With School Age Children
54	Poorer Households-Older Families & Mature Couples
55	Poorer Households-Elders In Retirement

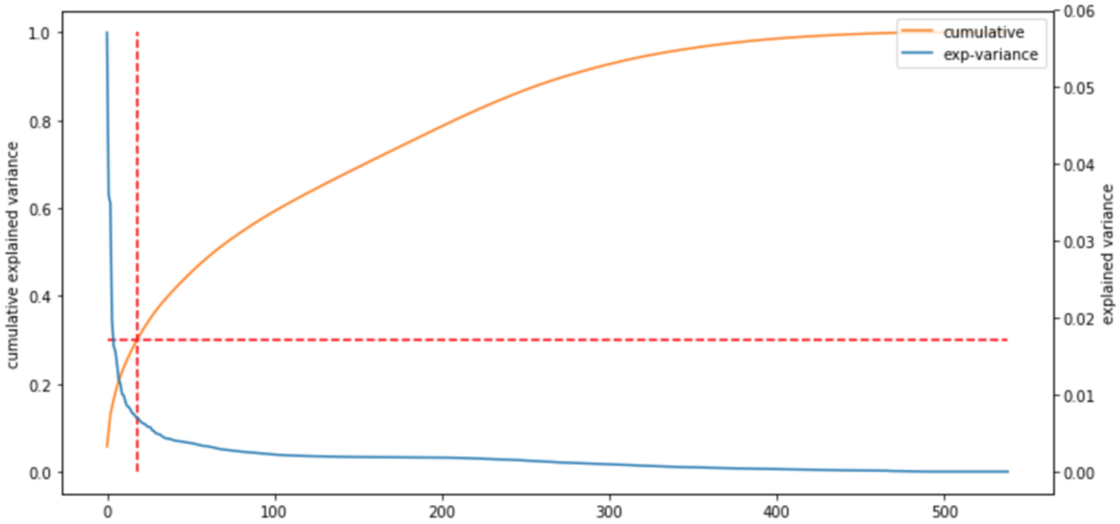
4. One-Hot Encoding for categorical features; (No ordering relations)

5. Fill Null values: replace remain null cells with the most frequent value in each column; (PCA and K-Means not able to deal with NaN values)

6. Scale the dataset by StandardScaler() from scikit-learn package; (K-means assumes spherical shapes of clusters)

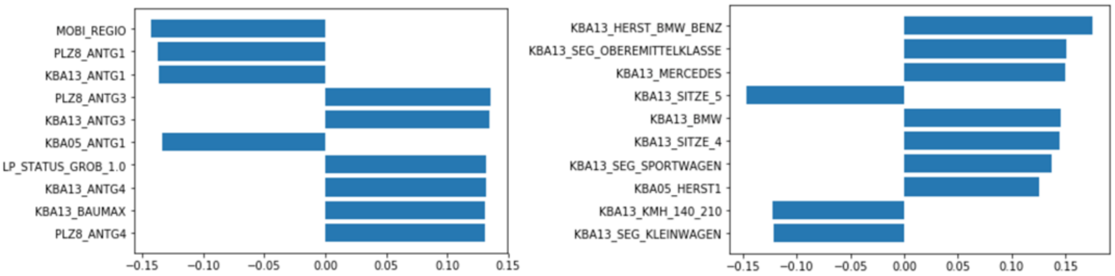
Implementation

1. Dimension Reduction with PCA



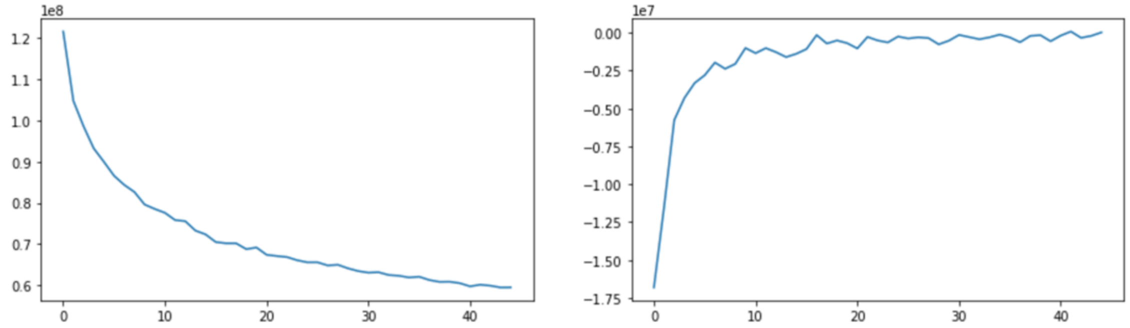
From above curve plot, we see that 18 or more components are able to explain Top 30% of the total variance and the explained variance drops significantly after about 20. So I picked 20 as the parameter for further analysis.

Top 2 PCA features and their corresponding are displayed as below:

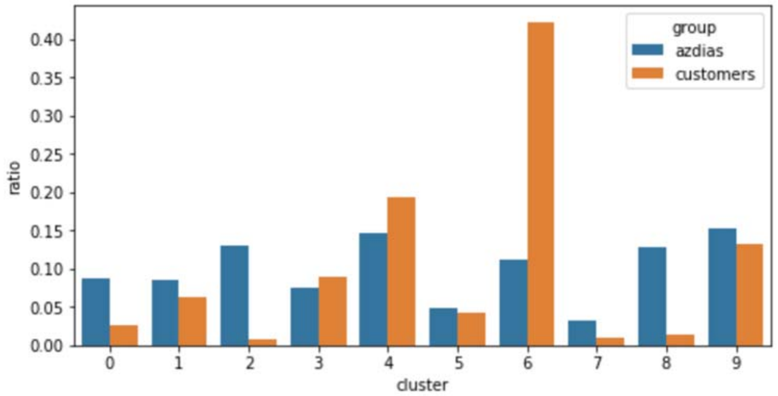


2. K-Means Clustering

From below plots, we see a limited score increase after 10 clusters. So 10 is selected as a reasonable cluster number for further analysis.

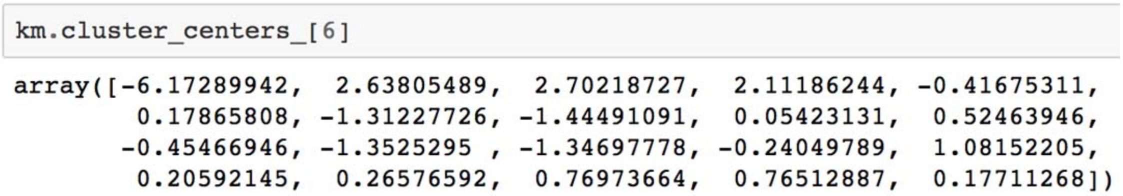


After training, we apply 10 cluster K-Means model on both demographics and customers dataset.



Based on cluster distribution histogram on both datasets, it is clear to see that Cluster 6 is outstanding for customers, which means people in this group is more likely to be part of the mail-order company's main customer base than other groups.

The coefficients of Cluster 6 is displayed below:



```
km.cluster_centers_[6]
```

```
array([-6.17289942,  2.63805489,  2.70218727,  2.11186244, -0.41675311,
        0.17865808, -1.31227726, -1.44491091,  0.05423131,  0.52463946,
       -0.45466946, -1.3525295 , -1.34697778, -0.24049789,  1.08152205,
        0.20592145,  0.26576592,  0.76973664,  0.76512887,  0.17711268])
```

The starting 4 PCA features have the largest absolute value of coefficient, with negative impact on the first one (-6.1728).

3. Supervised Learning model

(1) Model selection: I picked XGBoost, which is explained in previous section;

(2) Feature Engineering:

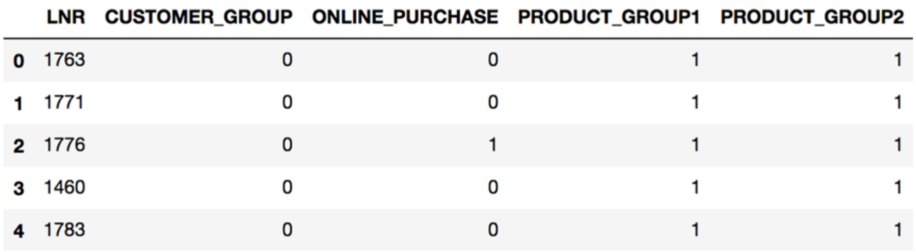
i) For prediction purpose, I am not going to drop any features columns or rows, in order to maintain information as much as possible.

ii) The main work for preprocessing, is transforming missing values into np.nan, labelling numerical and categorical columns, and one-hot encoding the categorical ones.

iii) Also, as far as Xgboost is a non-linear model, I did not scale the data. In addition, I left all Null cells alone, because Xgboost is able to dealing with them by default.

iv) Add extra features: By checking LNR column, I noticed that people in both train and test set are included in CUSTOMERS dataset. In this case, CUSTOMER_GROUP, ONLINE_PURCHASE, PRODUCT_GROUP can be used in the predictive model.

* Note that PRODUCT_GROUP is related to two elements: FOOD or COSMETIC, so that it can be engineered into two 0-1 features: PRODUCT_GROUP1 (whether it contains FOOD) and PRODUCT_GROUP2 (whether it contains COSMETIC).

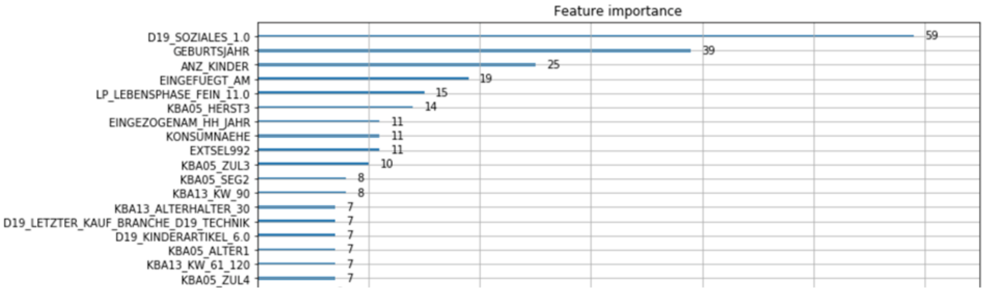


	LNR	CUSTOMER_GROUP	ONLINE_PURCHASE	PRODUCT_GROUP1	PRODUCT_GROUP2
0	1763	0	0	1	1
1	1771	0	0	1	1
2	1776	0	1	1	1
3	1460	0	0	1	1
4	1783	0	0	1	1

(3) Data split into train and validation set;

(4) Train on training set and evaluate on validation set, with AUC score as evaluation metric and 50 as early stopping rounds.

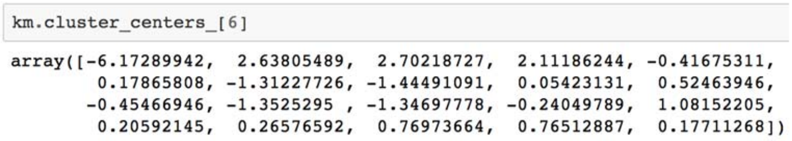
* Finally Xgboost model got a validation score of 0.76322 on the 48th iteration, and jumped to the 3rd place of Kaggle's Public Leaderboard on score of 0.80762;
* Refer to below graph for most important features:



Results

Customer Segmentation Analysis

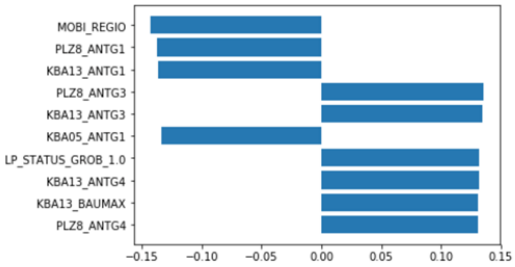
From above implementation section, K-Means Cluster 6 is considered to be the best representative of company customers. Since Cluster 6 is most affected by the first 4 PCA features, let's look inside for detail.



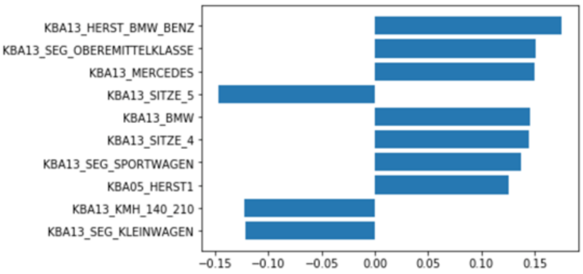
```
km.cluster_centers_[6]
```

```
array([-6.17289942,  2.63805489,  2.70218727,  2.11186244, -0.41675311,
        0.17865808, -1.31227726, -1.44491091,  0.05423131,  0.52463946,
       -0.45466946, -1.3525295 , -1.34697778, -0.24049789,  1.08152205,
        0.20592145,  0.26576592,  0.76973664,  0.76512887,  0.17711268])
```

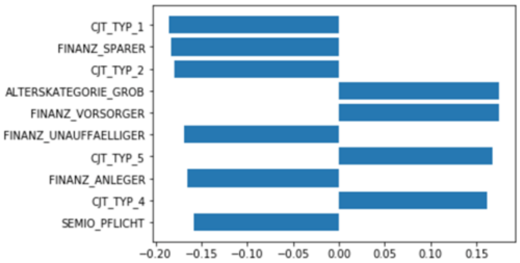
1st PCA feature



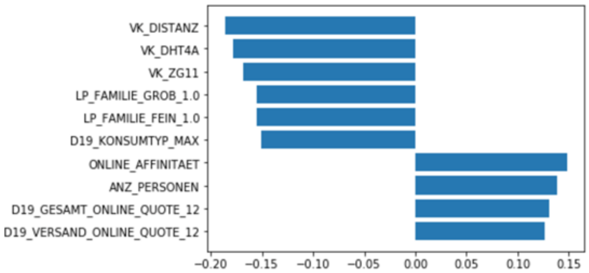
2nd PCA feature



3rd PCA feature



4th PCA feature



Following are the observation for each PCA feature:

1st: Low-income, High mobility, areas with more family houses or business buildings; (should be interpreted on reverse meaning due to negative coefficients {-6.17289942})

2nd: Cars: Upper-class, German, Fast, Sports;

3rd: Not young, relatively wealthy, Interested in advertising and will to shop online;

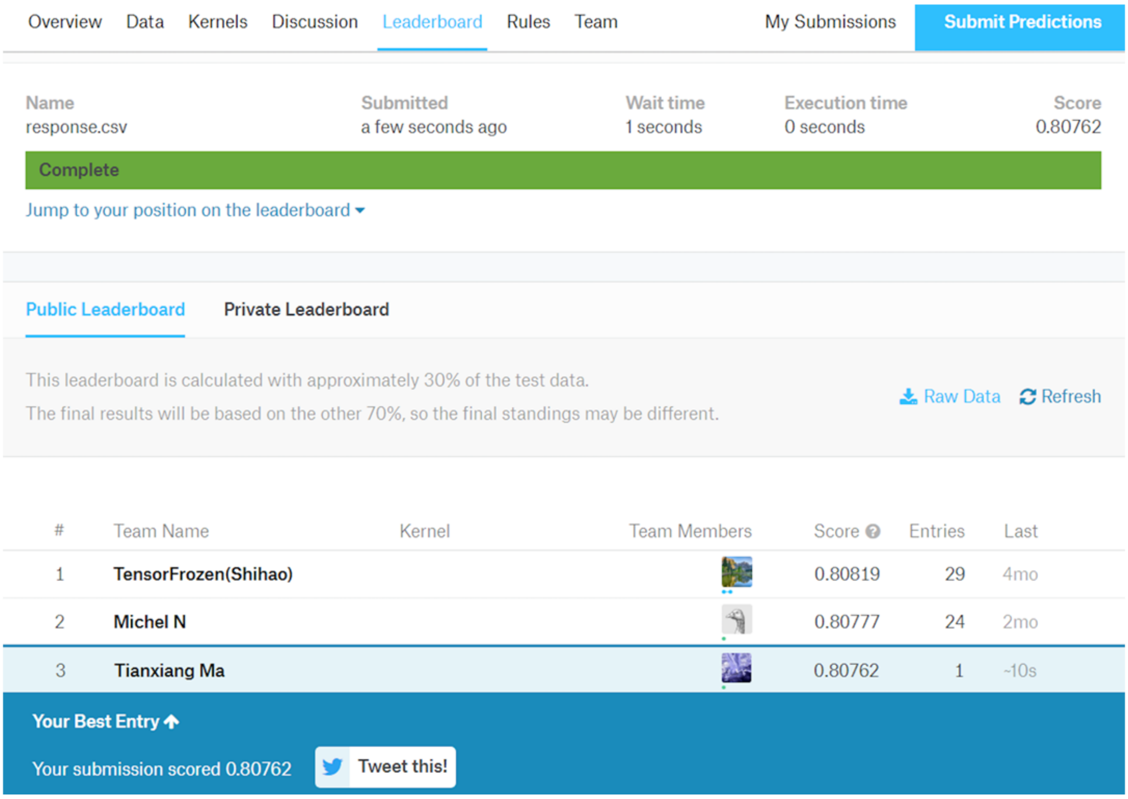
4th: Many online transactions, Not single, Has big family;

In conclusion, for the entire population, the target customer for this mail-order company is most likely to be:

- Life Stage: in the middle age and has big family;
- Financial Ability: with good income level and have upper class cars;
- Behavior Pattern: comfortable with advertising and online shopping;

Model Score on Kaggle

The final prediction (response.csv) has received a score of 0.80762 on Kaggle's Public Leaderboard, which ranked top 3 at the time of submission and very close to the Top Score 0.80819.



Name	Submitted	Wait time	Execution time	Score
response.csv	a few seconds ago	1 seconds	0 seconds	0.80762

Complete

Jump to your position on the leaderboard ▾

This leaderboard is calculated with approximately 30% of the test data.
The final results will be based on the other 70%, so the final standings may be different.

#	Team Name	Kernel	Team Members	Score	Entries	Last
1	TensorFrozen(Shihao)			0.80819	29	4mo
2	Michel N			0.80777	24	2mo
3	Tianxiang Ma			0.80762	1	~10s

Your Best Entry ↑

Your submission scored 0.80762

Conclusion

Reflection

In this project, demographics data of Germany was analyzed. Various Machine Learning techniques, in both Unsupervised Learning and Supervised Learning, were used to answer different questions.

While undertaking the research work, I did learn a lot not only in manipulating ML techniques, but also in finding insights of real-life business from unfamiliar domains. I would like to thank Udacity and Arvato Financial Services for setting this up.

The largest challenge for me is Data Understanding. There are hundreds of variables, but not all of them are documented. So I did a thorough study of all reference I can get to and engineered all features into a synthetic data dictionary with some manual work on data types. It ensures me on complete following project.

The second thoughts are about data engineering for predictive model. In this project, based on the limited understanding of data features, I tried to maintain as many features as possible in the training set, in order to preserve more underlying characteristics of original data.

It turns out that the single model I have trained performs well on Kaggle's Public Leaderboard, and some undocumented feature are very important in the model, such as 'D19_SOZIALES', 'ANZ_KINDER', 'EXTSEL992', and so on.

Improvement

In this project, even though the score looks ok on Kaggle's Public Leaderboard, there is still a lot to be improved.

1. Feature Engineering: Better understanding of data will definitely help in feature selection, and additional features extracted from outside resources for Germany population may improve clustering and prediction.

2. Other algorithms: lightgbm, catboost, and many more algorithms can be put into experiment in this project.

3. Fusion Model: for the time reason, I only trained one model in this project. However, Fusion Model is a necessary part in predictive problems. By combining multiple model which trained in different scheme, will probably further push the score to another level.