

Machine Learning Engineer Nanodegree

Capstone Report

Customer Segmentation and Prediction – Arvato Financial Solutions

Binbin Zhou
May 21st, 2020

Contents

Definition	3
Project Overview.....	3
Domain Background.....	3
Datasets and Inputs	3
Problem Statement.....	3
Evaluation Metrics	4
Analysis	5
Exploratory Data Analysis and Preprocessing.....	5
Check missing rate for each variable	5
Check categorical features and re-encode	6
Combine former steps to a general preprocessing step.....	6
Algorithms and Techniques	7
Customer segmentation:	7
Customer targeting model:.....	7
Results.....	7
Customer Segmentation	7
Principle component analysis	7
KMeans analysis	9
Modelling.....	12
Benchmark Model.....	12
Supervised model building.....	12
Parameter tuning	12
Make prediction and submit data for Kaggle ranking.....	14
Things to try for Improvement.....	14

Definition

Project Overview

Domain Background

Arvato, the second-largest division of Bertelsmann, is a global services company headquartered in Germany, whose service include customer support, information technology, logistics and finance.

In this capstone project, Arvato is trying to help a mail-order sales company in Germany to boost their client population. To achieve this goal, we are going to identify segments of the general population, understand the patterns of current customers, and target the potential customers to send out mail advertisement/invitation.

Datasets and Inputs

There are 4 datasets provided by Arvato, all of which have demographic features:

1. **Udacity_AZDIAS_052018.csv**: Demographics data for the general population of Germany; 891 211 persons (rows) x 366 features (columns)
2. **Udacity_CUSTOMERS_052018.csv**: Demographics data for customers of a mail-order company; 191 652 persons (rows) x 369 features (columns)
3. **Udacity_MAILOUT_052018_TRAIN.csv**: Demographics data for individuals who were targets of a marketing campaign, with a column "RESPONSE" to show if they turn into customers; 42 982 persons (rows) x 367 (columns).
4. **Udacity_MAILOUT_052018_TEST.csv**: Demographics data for individuals who were targets of a marketing campaign; 42 833 persons (rows) x 366 (columns).

Besides, there are another 2 files containing the meta-information for features in these tables:

5. **DIAS Attributes - Values 2017.xlsx**: *A detailed description of each data values for each variable*
6. **DIAS Information Levels - Attributes 2017.xlsx**: *A top-level description for each attributes, grouped by informational category*

Data 1 and data 2 can be used for exploratory, setting up data cleaning procedures, and unsupervised customer segmentation; Data 3 will be used for model training, which will be applied to data 4 to generate the prediction. Data 5 and data 6 can be used to map to all demographic tables, which will be helpful to get a better understanding of the segmentation and model. I found data 5 is especially helpful to map missing/unknown data values.

Problem Statement

The problem statement of this project is: How can this mail-order sales company send out mail advertisement/invitation to those people who are more likely to be converted to their customer?

I am going to solve this problem by the following steps:

1. Exploratory data analysis: data cleaning, categorical value encoding.
2. Use unsupervised learning techniques to cluster the general population and current customers, and look into the difference.
3. Based on the training sample provided, use supervised learning technique to build a model to predict how likely a person can be converted to their customer.
4. As a competition is hosted on the Kaggle, the prediction data will be uploaded to Kaggle so we can see how our model perform on out of time validation sample

Evaluation Metrics

For principle component analysis, there are three types of model attributes:

- Mean: the mean that was subtracted from a component in order to center it.
- Components_: the makeup of the principle components;
- S: The singular values of the components for the PCA transformation.

From s, we can get an approximation of the data variance that is covered in the first n principle components. The approximate explained variance is given by the formula: the sum of squared s values for all top n components over the sum over squared s values for all components:

$$\frac{\sum_n S_n^2}{\sum S^2}$$

For the second part for the predictive model, as the target rate in the dataset is highly imbalanced (1.2%), it will not be appropriate to just use accuracy to evaluate the performance of the model. Instead, I proposed to use ROC-AUC, which is a performance measurement for classification problem at various threshold settings. ROC (Receiver Operating Characteristics) is a probability curve and AUC is the area under this curve. ROC curve is plotted with TPR (true positive rate) on x-axis against the FPR (false positive rate) on y-axis.

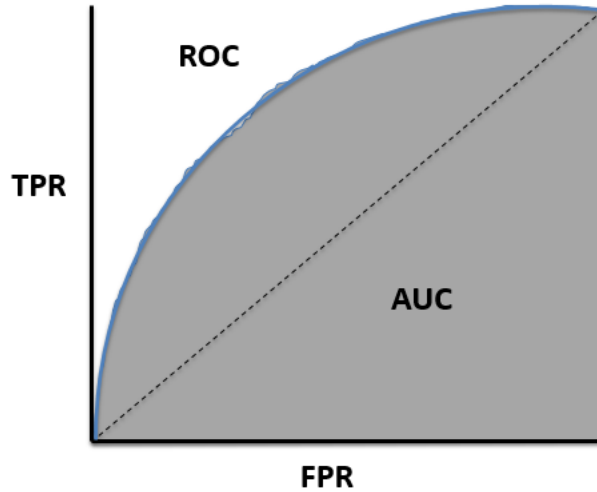


Figure 1: AUC-ROC curve (my own drawing)

As the figure 1 shows, higher area under the curve, model can have higher TPR with controlling FPR under a certain threshold. In our project here, higher ROC-AUC means that the model can identify higher percent of customers that can be converted to real customer, while in this targeted group, the false positive rate can be controlled in a relative small number.

Analysis

Exploratory Data Analysis and Preprocessing

Check missing rate for each variable

First of all, I checked the missing rate for each variable in the whole dataset, and remove those with high missing rate. When I check the meta-information for these features, I found that the data not only contains NaN, for lots of features, 0 or 9 could also represent missing/unknown, so I go through each variables and check the missing rate for NaN, 0, and 9 (depends on the metadata information for value explanation). During this step, 89 features have modified missing rate due to unknown value replacement.

Finally, as figure 2 shows, 7 features in the red box that have missing rate higher than 50% got removed: 'ALTER_KIND4', 'ALTER_KIND3', 'ALTER_KIND2', 'ALTER_KIND1', 'AGER_TYP', 'EXTSEL992', 'KK_KUNDENTYP'.

	Missing Values	% of Total Values
ALTER_KIND4	890016	99.9
ALTER_KIND3	885051	99.3
ALTER_KIND2	861722	96.7
ALTER_KIND1	810163	90.9
AGER_TYP	677503	76.0
EXTSEL992	654153	73.4
KK_KUNDENTYP	584612	65.6
ALTERSKATEGORIE_FEIN	262947	29.5
D19_VERSI_ONLINE_QUOTE_12	257113	28.8
D19_VERSAND_ONLINE_QUOTE_12	257113	28.8

Figure2. Features with high missing rate

Check categorical features and re-encode

Feature type is checked and categorical features will be picked in this step. Then I checked the number of unique levels for each categorical feature, if there are only 2 unique values, replace them with 0 and 1. Otherwise, the feature will be converted to dummy variables.

- CAMEO_DEU_2015 -- number of unique values: 45
- CAMEO_DEUG_2015 -- number of unique values: 19
- CAMEO_INTL_2015 -- number of unique values: 43
- D19_LETZTER_KAUF_BRANCHE -- number of unique values: 35
- EINGEFUEGT_AM -- number of unique values: 5162
- **OST_WEST_KZ -- number of unique values: 2**

So OST_WEST_KZ is encoded to 0 and 1, all other categorical values are encoded as dummy variables.

Combine former steps to a general preprocessing step

A general preprocess function was generated to combine the steps for removing high missing rate features, and categorical data encoding. This function can be applied to the following steps to make sure all dataset have the same preprocess step.

Algorithms and Techniques

Customer segmentation:

During this part, I used unsupervised learning methods to cluster the population into different groups. First of all, non-numerical features are encoded, and feature scaling method is applied to prepare data for principle component analysis. A popular dimension reduction technique PCA is used, to reduce the dimension around half number of total features. After that, KMeans analysis is applied on the PCA processed dataset, to separate the whole population into subgroups.

Segmentation can be applied both to the general population and the customer dataset, which can be used to compare with each other and look into the difference to find any potential pattern for customers.

Customer targeting model:

Supervised learning technique is applied to build a model and predict who is more likely to be converted to a customer. For this particular project, a tree based model would be appropriate concerning that there are around 500 features in total. I choose light GBM as the target model for this project, which is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm.

LightGBM is picked due to these reasons:

- Faster training speed and higher efficiency
- Lower memory usage
- Better accuracy
- Compatibility with large datasets

Also, **GridSearchCV** is used for parameter tuning, for better performance of the model and avoiding overfitting.

Results

Customer Segmentation

Principle component analysis

Data after preprocessing as mentioned above, will go through another two steps to be ready to do segmentation analysis: **missing data imputation and feature scaling**. In this study, missing data will be replaced by the mean value, and a standard scaler is applied to bring all features to the same range.

After that, PCA is applied to reduce the dimension from original number of features (around 500 after encoding) to 300, which in total can explain around 95% of variance, and the top 150 components can explain 75.7% of variance (Figure 3).

- total explained variance: 95.2%
- top 150 components explained variance: 75.7%

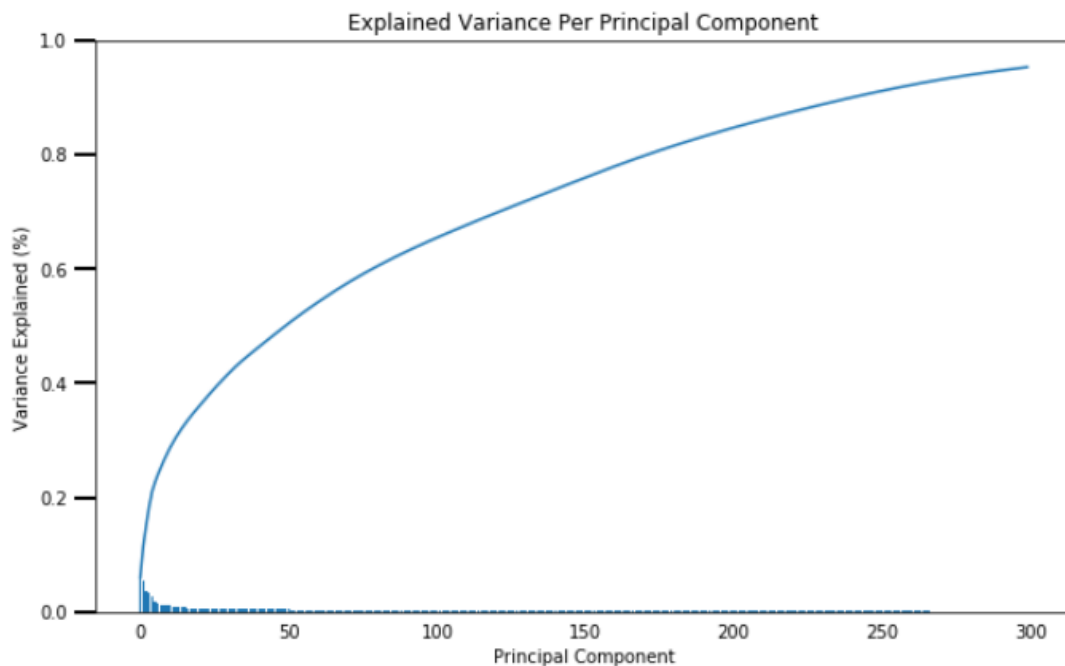


Figure3: Explained variance for each principle component and accumulated explained variance

Top 3 components were checked in detail and for each of these components, the top 3 and bottom 3 features contributed to this components were picked and their definition were checked to get more understanding.

- **first dimension:**
PLZ8_ANTG3 0.1229: number of 6-10 family houses in the PLZ8
KBA13_ANTG3 0.1228: nan
KBA13_ANTG4 0.1203: nan
KBA13_ANTG1 -0.1235: nan
PLZ8_ANTG1 -0.1238: number of 1-2 family houses in the PLZ8
MOBI_REGIO -0.1289: moving patterns

- **second dimension:**
 KBA05_SEG6 0.178: share of upper class cars (BMW 7er etc.) in the microcell
 KBA05_KRSOBER 0.163: share of upper class cars (referred to the county average)
 KBA05_KRSVAN 0.1605: share of vans (referred to the county average)
 PLZ8_ANTG3 -0.0367: number of 6-10 family houses in the PLZ8
 KBA13_ANTG3 -0.0367: nan
 KBA05_ANTG3 -0.0392: number of 6-10 family houses in the cell
- **third dimension:**
 D19_GESAMT_ANZ_24 0.1519 nan
 ONLINE_AFFINITAET 0.1434 online affinity
 D19_GESAMT_ANZ_12 0.1432 nan
 D19_VERSAND_ONLINE_DATUM -0.1427: actuality of the last transaction for the segment mail-order ONLINE
 D19_GESAMT_DATUM -0.1435: actuality of the last transaction with the complete file TOTAL
 D19_GESAMT_ONLINE_DATUM -0.148: actuality of the last transaction with the complete file ONLINE

KMeans analysis

Based on the 300 components identified by the PCA in the upper steps, MiniBatchKMeans were applied to analyze the data further. Based on the elbow method (figure 4), it seems that separate the population into 16 clusters gives the smallest SSE (sum of squared error).

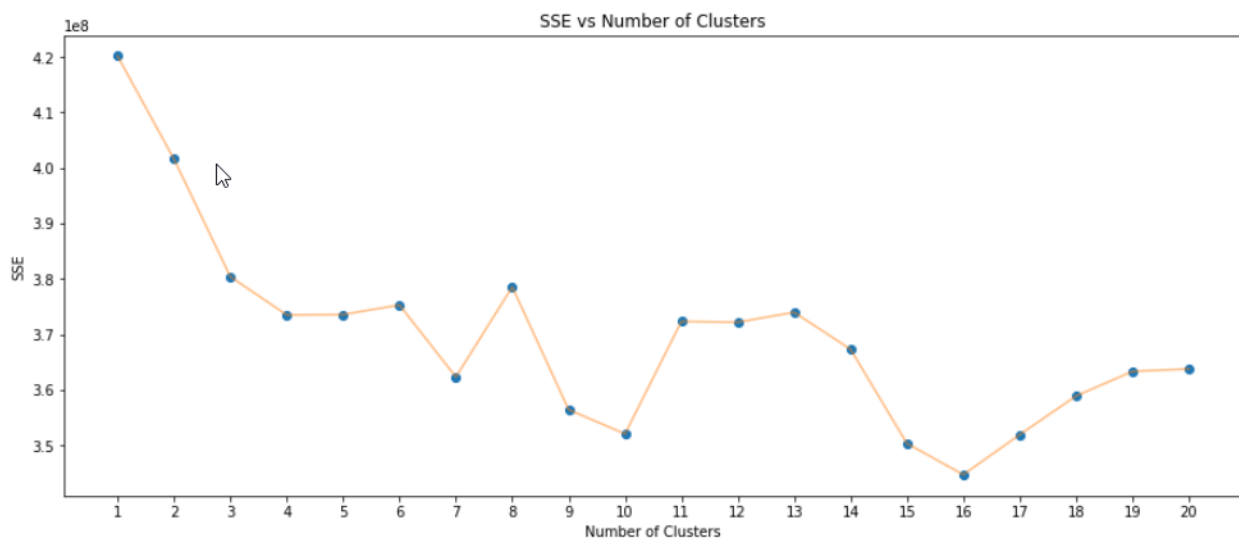


Figure 4: Sum of Squared Error by number of Clusters

The algorithm was applied to both the general population and the customer data, whose clusters are compared to each other to identify the overrepresented cluster and underrepresented cluster compared to the general population (figure 5 and 6).

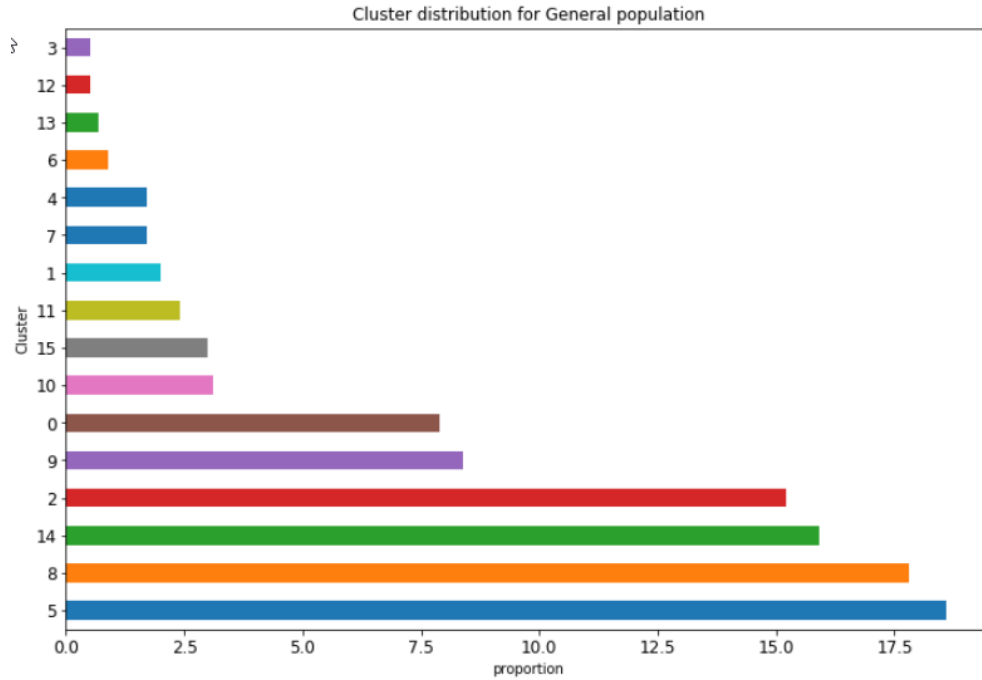


Figure 5: cluster distribution for general population

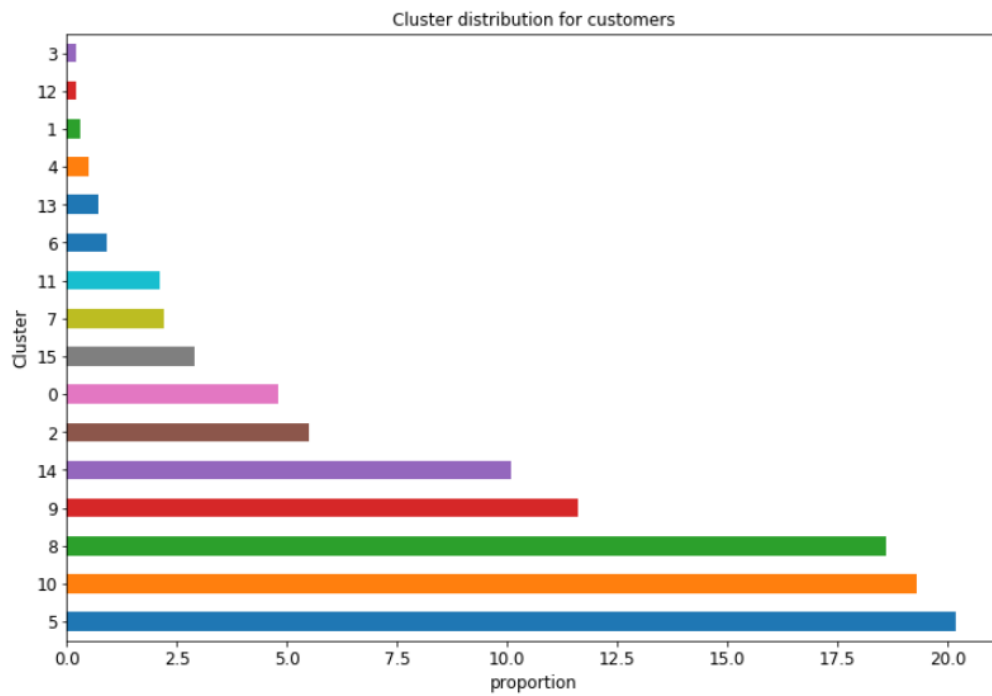


Figure 6: cluster distribution for customer data

After comparing with these two datasets cluster information, clusters that got overrepresented and underrepresented were picked as shown in Figure 7:

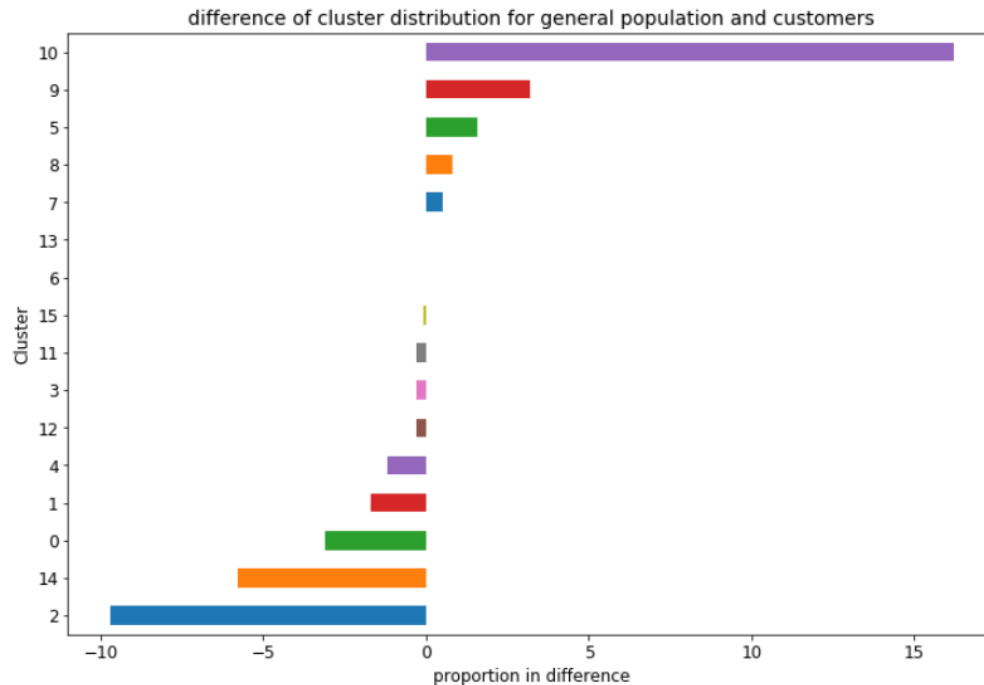


Figure 7: cluster comparison for general population and customers

- In Customer group, cluster 10 has around 16% more and cluster 2 has around 10% less compared to the distribution in general population

For these two clusters, I checked the mean value for the top features picked from the PCA analysis (top 3 and bottom 3 features for top 2 components, 12 features in total). Combining with the metadata information, I summarized some characteristics for the over represented group as shown in Figure 8.

Based on this comparison, the over represented group tends to have:

- higher MOBI_REGION: lower moving activities
- lower PLZ8_ANTG1: lower number of 1-2 family houses in the PLZ8
- higher KBA05_KRSVAN: higher share of vans
- lower PLZ8_ANTG3: lower number of 6-10 family houses in the PLZ8
- lower KBA05_ANTG3: lower number of 6-10 family houses in the cell
- lower KBA05_SEG6: lower share of upper class cars (BMW 7er etc.) in the microcell

So in general, the over represented group tends to have these characteristics:

- less moving activities
- drive vans more often, not upper class cars

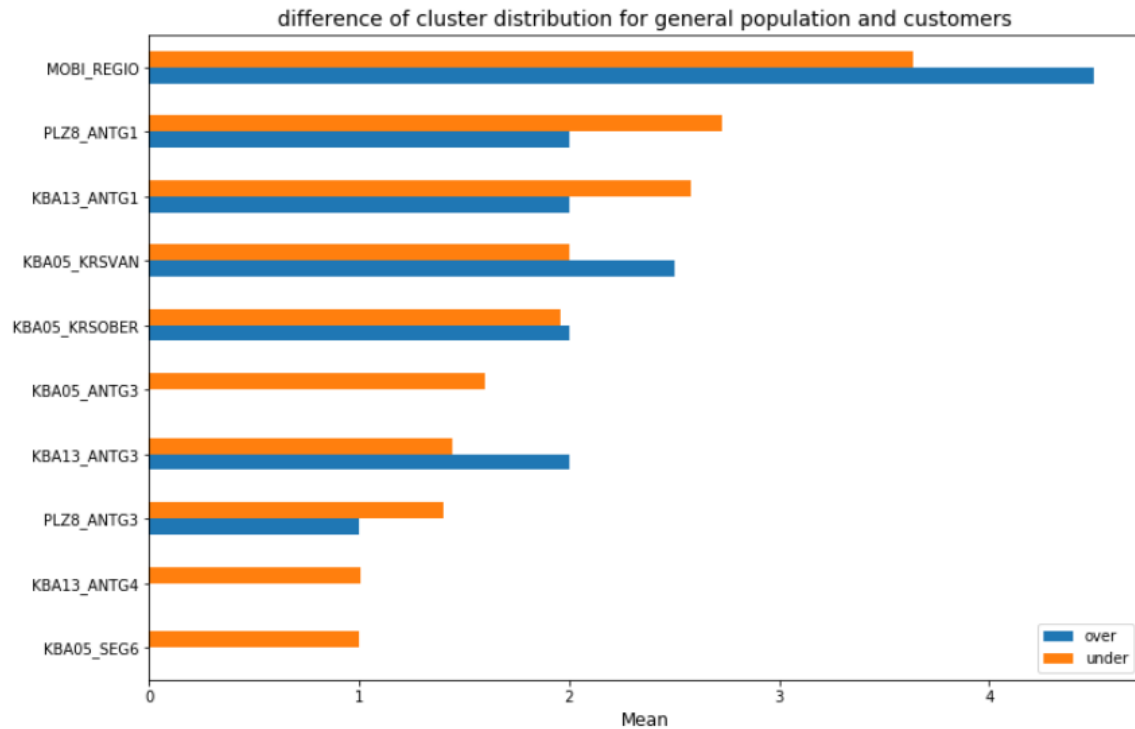


Figure 8: mean value comparison for cluster 10 (over) and cluster 2 (under)

Modelling

Benchmark Model

A simple gradient boosting model without parameter tuning can be used as a benchmark model. The ROC-AUC (explained in the following part 'Evaluation Metrics') for such simple model is around 0.72 based on my test.

Supervised model building

Based on former experience, a decision tree based model would be appropriate for this project. And as I mentioned above, light GBM was picked for model building, as compared to the popular algorithm XGBoost, lightGBM is faster and require smaller computer resource.

Parameter tuning

GridSearchCV is applied to model training and parameter tuning. These parameters are tuned to gain the best ROC-AUC score for the model:

- n_estimators

- colsample_bytree
- max_depth
- num_leaves
- min_data_in_leaf
- reg_alpha
- reg_lambda
- min_split_gain
- subsample

After several rounds of parameter tuning, the best parameter picked by the GridSearchCV is:

{'colsample_bytree': 0.7, 'learning_rate': 0.01, 'max_depth': 3, 'min_data_in_leaf': 10, 'min_split_gain': 0.4, 'n_estimators': 200, 'num_leaves': 80, 'reg_alpha': 1.5, 'reg_lambda': 1.5, 'subsample': 0.8, 'subsample_freq': 20}

With the best model picked, the ROC-AUC on the whole training sample is 0.827. Figure 9 displays the importance feature:

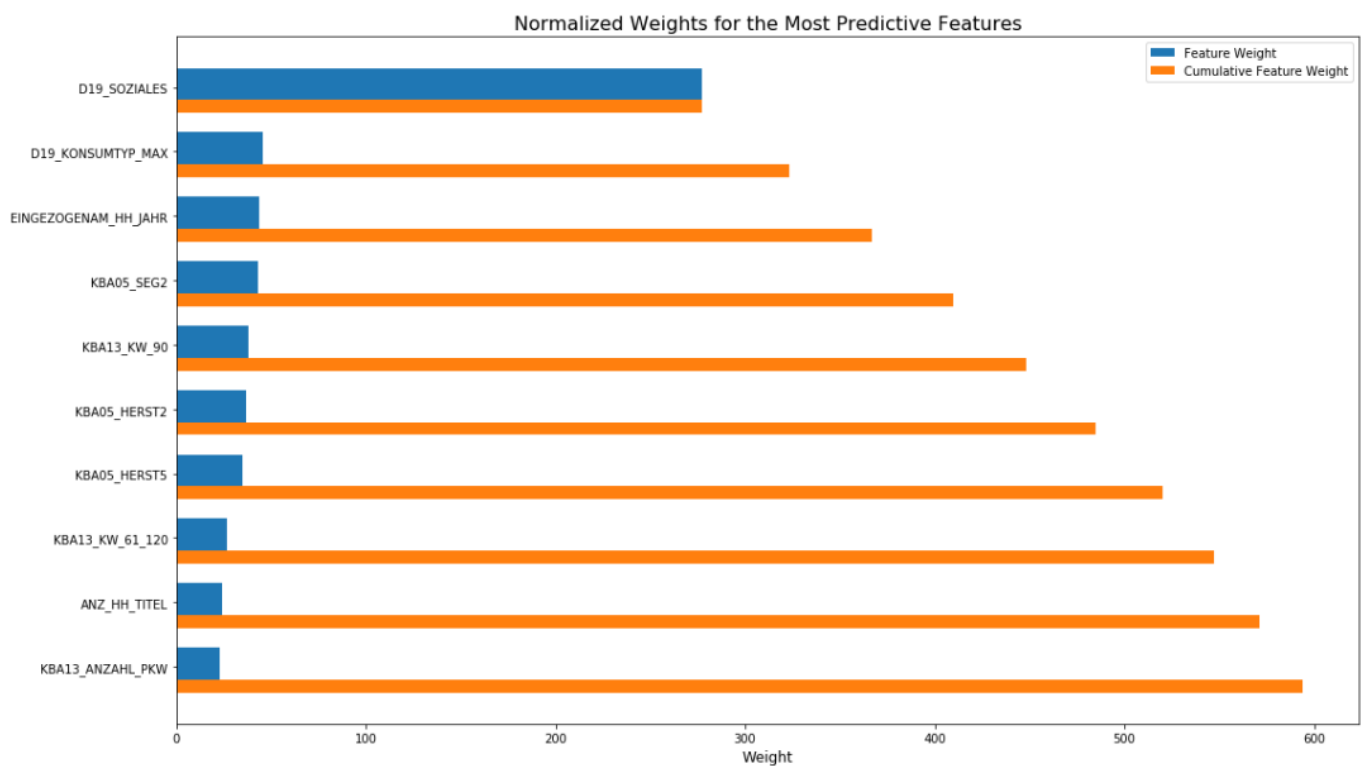
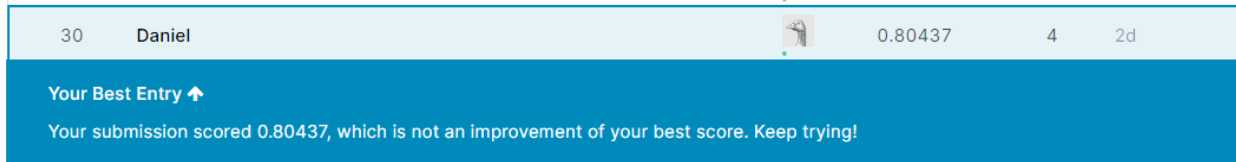


Figure 9: Feature importance

Make prediction and submit data for Kaggle ranking

With the best model picked after parameter tuning, predictions for the test data (after preprocessing same to the training dataset), the submission on Kaggle returned the score 0.80437, which ranks 30th/187.





30	Daniel		0.80437	4	2d
Your Best Entry 					
Your submission scored 0.80437, which is not an improvement of your best score. Keep trying!					

Figure 10: Screenshot for Kaggle submission

Things to try for Improvement

Till the date this report is generated, the best score on Kaggle is **0.81063**, which is around 0.006 higher than my score. Some score of improvements in my plan:

- ✚ Apply the same logic for missing data replacement as I mentioned in the EDA, which is to replace the unknown values (0, 1, 9 depends on each variable) to one same value. As we used tree based model, this improvement might not be tremendous, but if we can group 0 and 9 into the same data could still improve the model.
- ✚ Look into the features in more detail and do more feature engineering
- ✚ Try to run some up-sampling and down-sampling to see if model performance can be improved
- ✚ Even though we picked LightGBM and it seems working fine, it would still worth to try other popular machine learning models like XGBoost/AdaBoost.