Machine Learning Engineer Nanodegree

Capstone Project Report: Bertelsmann-Arvato-customer-segmentation

by Zefu Chen

Definition

Project Overview

Domain Background

Arvato is a services company that provides financial services, Information Technology (IT) services and Supply Chain Management (SCM) solutions for business customers on a global scale. Arvato's customers come from a wide range of industries such as insurance companies, e-commerce, energy providers, IT and Internet providers [1]. Also, Arvato is wholly owned by Bertelsmann, which is a media, services and education company [2].

Arvato is helping its customers get valuable insights from data in order to make business decisions. Customer centric marketing is one of the growing fields. Identifying hidden patterns and customer behavior from the data is providing valuable insights for the companies operating in customer centric marketing. As a result, Machine Learning is the right way to go.

In this project, Arvato is helping a Mail-order company, which sells organic products in Germany, to understand its customers segments in order to identify next latent customers. The existing customer data and the demographic data of population in Germany are to be studied to understand different customer segments, and then building a system to make predictions on whether a person will be a customer or not based on the demographic data.

Dataset and Inputs

There are four data files associated with this project:

- 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).

Additionally, 2 metadata files have been provided to give attribute information:

- DIAS Information Levels Attributes 2017.xlsx: top-level list of attributes and descriptions, organized by informational category
- DIAS Attributes Values 2017.xlsx: detailed mapping of data values for each feature in alphabetical order

All the files associated with the project have been provided by Arvato in the context of MLND Program for analysis and customer segmentation purposes.

Problem Statement

Given the demographic data of an individual, how can a mail order company acquire new customers in an efficient way?

Firstly, the demographic data of the general population and the customers will be researched through unsupervised learning algorithms. The goal is to identify segments in general population and segments in the existing customers, and then discovering what demographic features correspond to an individual being a customer for the mail-order company.

Then, supervised learning algorithms will be used to make predictions on whether an individual is a latent customer or not, based on the demographic data.

Evaluation Metrics

The project is divided into two parts:

Customer Segmentation using unsupervised learning algorithms

This part of the project uses a dimensionality reduction technique PCA to reduce the number of dimensions. The explained variance ratio of each feature could be the reference in selecting the number of dimensions for the later steps. The minimum number of dimensions explaining as much variation as possible in the dataset can be chosen in this step. Also, in case of segmenting the customers into different clusters, an unsupervised learning algorithm like K-Means Clustering is proposed. Also, in this case the number of clusters is selected on the squared error i.e. the distance between all the clusters with the help of an elbow plot.

Customer Acquisition using supervised learning algorithms

In the second part of the project, the task is to predict whether or not the mail-order company should approach a customer. Here the given training data will be split into train and evaluation sets, the model will be trained on the training split and will be evaluated on the evaluation split. In this step evaluation metrics for classification can be used.

The class label distribution is highly imbalanced, in this particular binary classification problem there are 42,430 observations with label ‘0’ and only 532 observations with label ‘1’, as shown in Figure 1. For this problem, we need to be able to tell whether a person will be a future possible customer. AUROC metric which considers both true positive rate and false positive rate seem to be a good choice for this problem, since we want to be able to correctly predict both cases i.e. whether a person becomes a customer or not. Since, both these predictions are important for us [3].

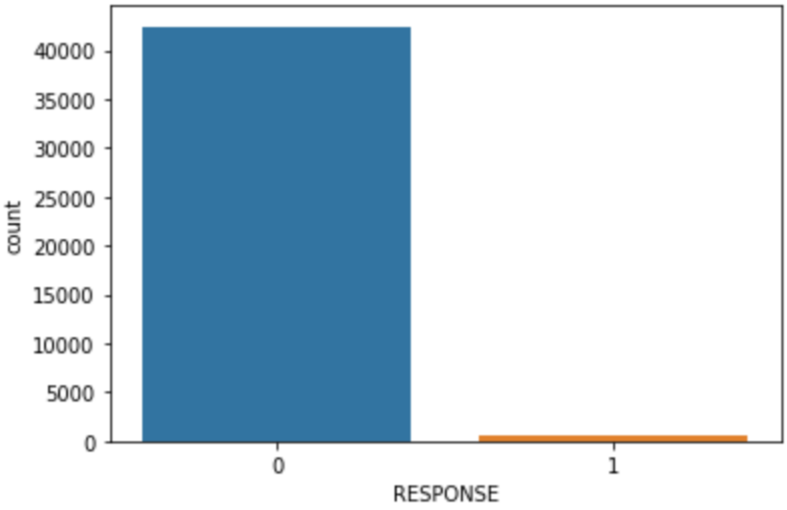


Fig. 1 Class imbalance

For this reason, Area Under Receiver Operating Characteristic (AUROC), has been selected as an evaluation metric. The AUROC gives an idea about overall performance of the model, where the curve is created by plotting True positive rate and False positive rate under different threshold settings. A good performing model will have an AUROC of 1. So higher the AUROC better the performance of the model.

Exploration Data Analysis

i. Addressing mixed type columns

As a first step, the warnings that appeared while loading the data were studied. The columns 18 and 19 contained mixed features and some mis-recorded values. The Attribute-values excel sheet was used as a reference to understand what these columns represent and what values can these columns take.

a) Addressed columns ‘CAMEO_DEUG_2015’ and ‘CAMEO_INTL_2015’.

b) Mis-recorded values – ‘X’, ‘XX’, are replaced with NaN values in the dataframe.

ii. Addressing ‘unkown’ values

The second step is to fix the unknown representations in all the columns. The ‘Attribute-values’ excel sheet contains the information about which columns contain unknown values and how they are entered specified in the dataset. With this information all the unknown values are replaced with NaN values in the dataframes. In total, there were 232 columns which contained unknown representations.

iii. Addressing non-existent values in ‘LP_*’ columns

Another problem with the given data lies in the values in the columns ‘LP_FAMILIE_FEIN’, ‘LP_FAMILIE_GROB’, ’LP_STATUS_FEIN’, ‘LP_STATUS_GROB’, ‘LP_LEBENSPHASE_FEIN’ and ‘LP_LEBENSPHASE_GROB’. These columns give the information about a person’s family status, financial status and the life stage they are in.

a) These columns contained ‘0’ as a value in the recorded data, which does not correspond to any category specified in the Attribute information data. These ‘0’s have been converted to NaN values.

b) The ‘LP_LEBENSPHASE_FEIN’ and ‘LP_LEBENSPHASE_GROB’ have too much granular information packed into them. The FEIN data consisted fine information about life stage and wealth information. This information has been divided to represent wealth information as one feature and life stage information as one feature and saved into the same two columns.

c) The columns ‘LP_FAMILIE_FEIN’ and ‘LP_STATUS_FEIN’ have been dropped since they contained duplicate information that the corresponding ‘_GROB’ columns consisted.

iv. Re-encoding features

The below specified features have been re-encoded,

a) EINGEFUGT_AM: This column represents the date on which the person has joined or the date the entry was made. This column has been converted to datetime column and only year has been extracted as a feature.

b) ANREDE_KZ: This represents the Gender, which was encoded with values 1,2 for male and female, is reencoded to contain 0-male and 1-female.

c) CAMEO_INTL_2015: This column contained information about the status of a person according to international standards. This column has been divided into two different columns to consist information about International Family status, International Wealth status.

d) WOHNLAGE: This column also has mis recorded values. These values were replaced with NaNs.

e) LNR: This column corresponds to an ID given to each person and this feature has been neglected for the analysis.

v. Missing Values

After cleaning the data and engineering certain features, the missing values were studied.

➢ Column wise: The percentage of missing values in each column is analysed. The columns which had missing values in customers data also seems to have missing data in the general population data and the distribution of the missing data per column is similar between these two. A threshold of 30% was decided after analysing the percentage missing value distribution. The columns that had more than 30% missing values were dropped from both customers data and general population data. A total of 11 columns have been dropped in this step, the columns that have been dropped are shown in Figure 2.

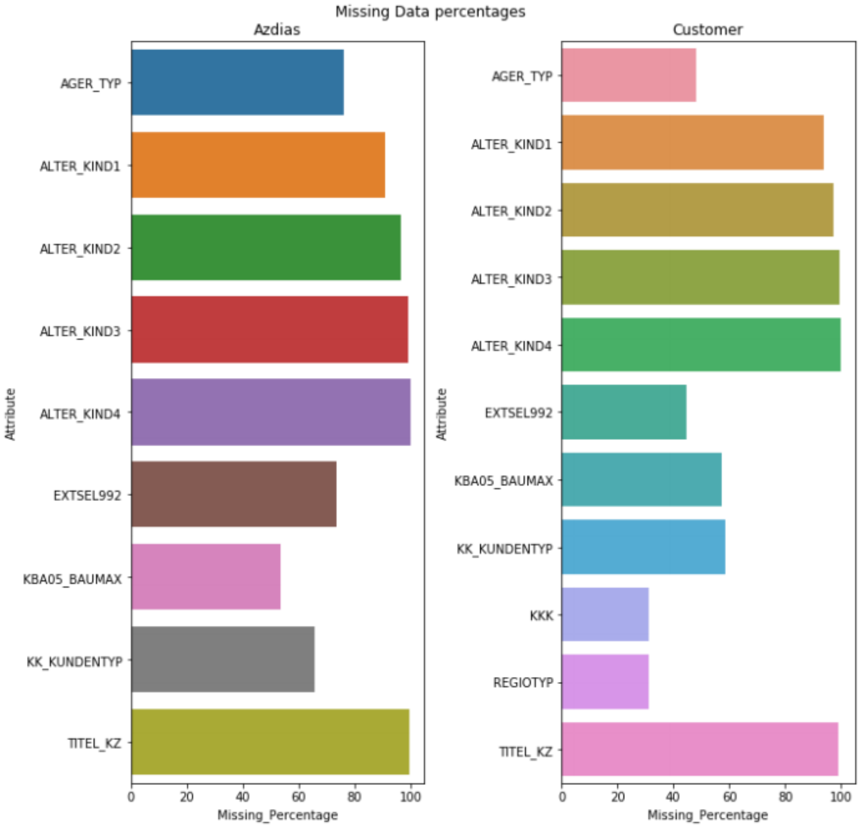


Fig. 2 Columns with more than 30% missing values

- Row wise: The number of missing values per row is analysed in this step. All the observations with more than 50 missing features are dropped in this step. This resulted in dropping a total of 1,53,933 observations from general population data which originally contained 8,91,211 observations. And a total of 57,406 observations were dropped from customers data which originally contained 1,91,652 observations.

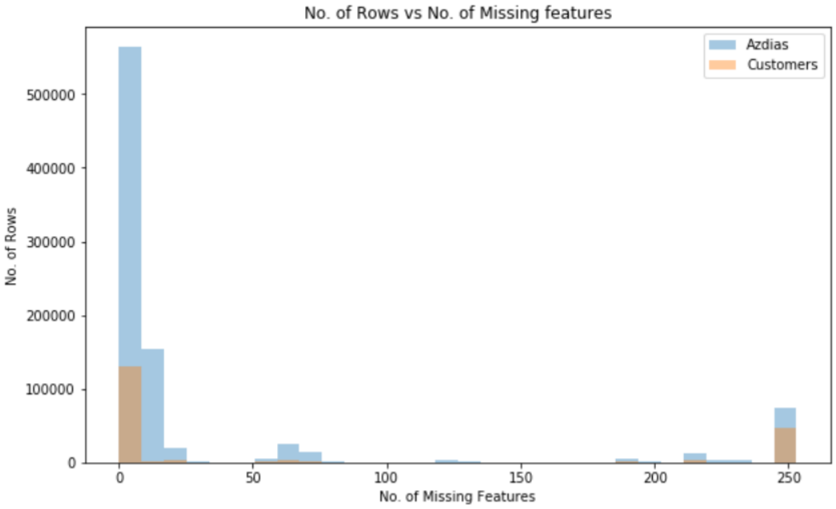


Fig. 3 Distribution of missing values

vi. Imputing Missing Values

After removing the features and rows which contained missing values based on set thresholds. The data still has some missing values. These missing values have been replaced with the most frequently occurred observation in each feature. Since the data corresponds to population in general, imputing the missing values with most frequent observations has been selected.

vii. Feature Scaling

A standard scaler is used to bring all the features to the same range. This is done in order to eliminate feature dominance when applying dimensionality reduction.

Methodology

Customer Segmentation

The aim of the first part of the project is to divide the general population and the customers into different segments, in order to compare the general population and customers to determine future customers. Here the company's existing customers data was available to understand and compare each feature in the customers data and the general population data. This requires lot of analysis and this process is time consuming because not all the features will be important in determining the customer behaviour. Also, there might exist some complex interactions between these features which resulted in the person being a customer. A hand coded analysis like this would consume a lot of time resulting in no fruitful results.

Dimensionality Reduction

For this reason, an approach to segment the customers and general population into different parts using unsupervised learning algorithms was chosen. The Principal Component Analysis (PCA) was performed on the given data to reduce the number of dimensions. Since there were 353 features after the data cleaning and feature engineering step, there is a need to understand which features will be able to explain the variance in the dataset. This is done with the help of PCA and the resulting explained variance plot is shown in Figure 4.

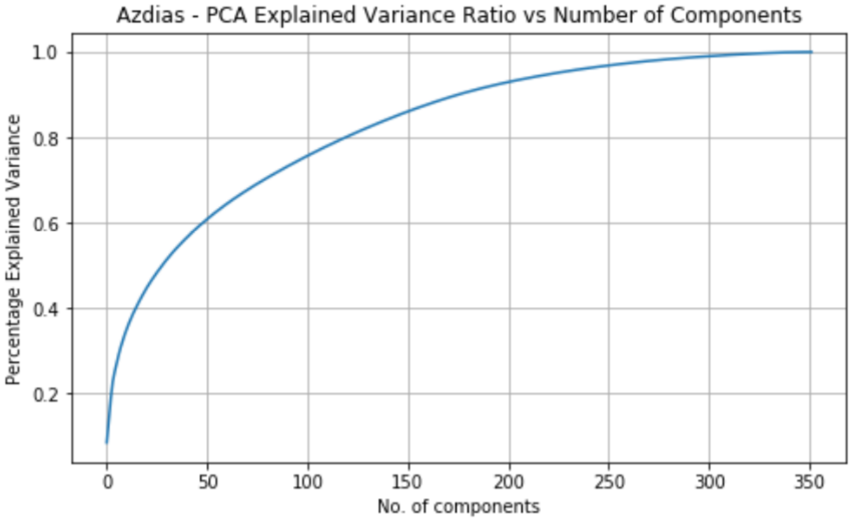


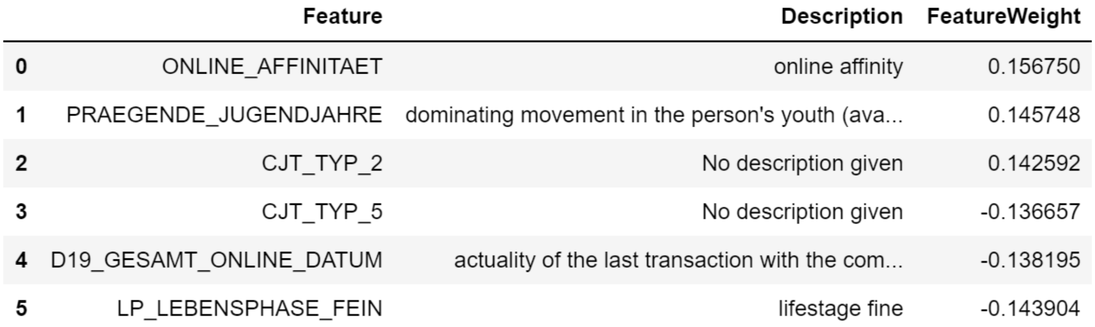
Fig. 4 PCA Explained Variance plot

As seen in Figure 4, although we have 353 features almost 90% of the variance in the data can be explained with the help of 150 components of PCA. With this step we will be able to reduce the number of features from 353 to 150.

PCA Component Analysis

These 150 components can be further explained by looking at the feature weights the PCA algorithm has given to the original features. For example, the component ‘0’ explanation is shown in Table 1. The component ‘0’ corresponds to people who have high moving patterns and have a greater number of 1-2 family houses in their neighbourhood. Also, these people have a smaller number of houses with 6-10 families. Which shows that these people tend to live in neighbourhoods which have small family buildings and not apartments. Other features ‘KBA14_*’ corresponds to shares of cars, which have a certain specification. (The description for these columns is not given but this conclusion is made based on the description of other given feature descriptions.)

Table 1 PCA – Component 1



	Feature	Description	FeatureWeight
0	ONLINE_AFFINITAET	online affinity	0.156750
1	PRAEGENDE_JUGENDJAHRE	dominating movement in the person's youth (ava...	0.145748
2	CJT_TYP_2	No description given	0.142592
3	CJT_TYP_5	No description given	-0.136657
4	D19_GESAMT_ONLINE_DATUM	actuality of the last transaction with the com...	-0.138195
5	LP_LEBENSPHASE_FEIN	lifestage fine	-0.143904

Clustering

After the dimensionality reduction, the next step is to divide the general population and customer population into different segments. K-Means clustering algorithm has been chosen for this task. Since it is simple and is apt for this task, since it measures the distance between two observations to assign a cluster. This algorithm will help us in separating the general population with the help of the reduced features into a specified number of clusters. And use this cluster information to understand the similarities in the general population and customer data.

The number of clusters is a hyperparameter when working with clustering algorithms. The basic idea behind the clustering algorithms is to select the number of clusters to minimize the intra-cluster variation. Which means the points in one cluster are as close as possible to each other. There is no definitive way of selecting the number of clusters, we can either intuitively select a specific number of clusters or perform an analysis and then select the number of clusters. Here, an elbow plot has been used to decide the number of clusters for the K Means algorithm. The elbow plot plots the Sum of Squared distances in each cluster for the specified list of number of clusters. [4]

This plot helps in understanding how the number of clusters affect the intra-cluster distances. The optimal number of clusters can be the number where the sum of squares of distances starts to plateau. The number of clusters in this case is chosen to be ‘8’, since the sum of squares of distances stops decreasing at a higher rate at this point as shown in Figure 5.

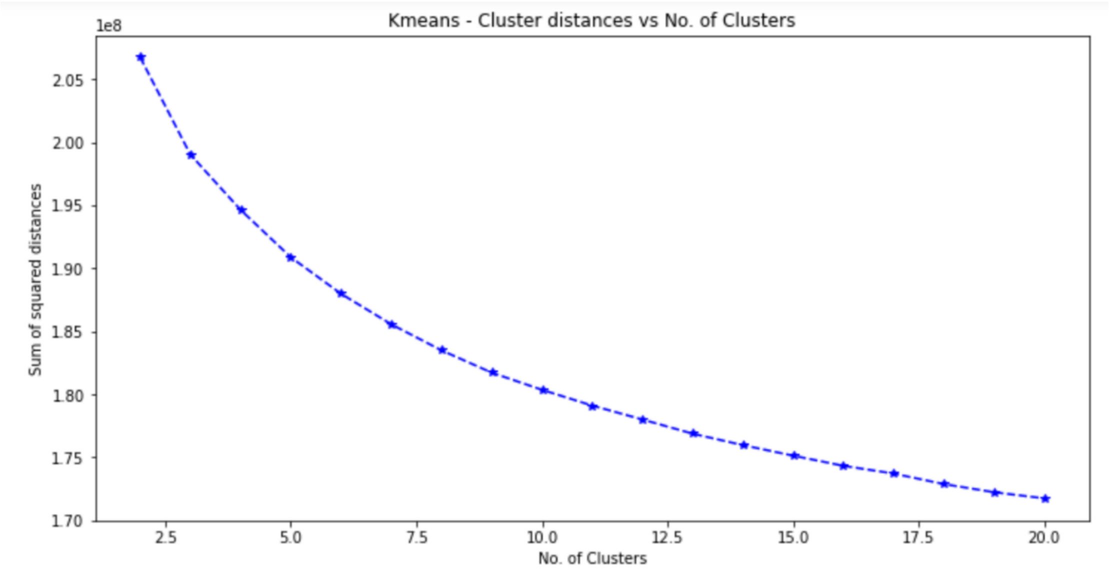
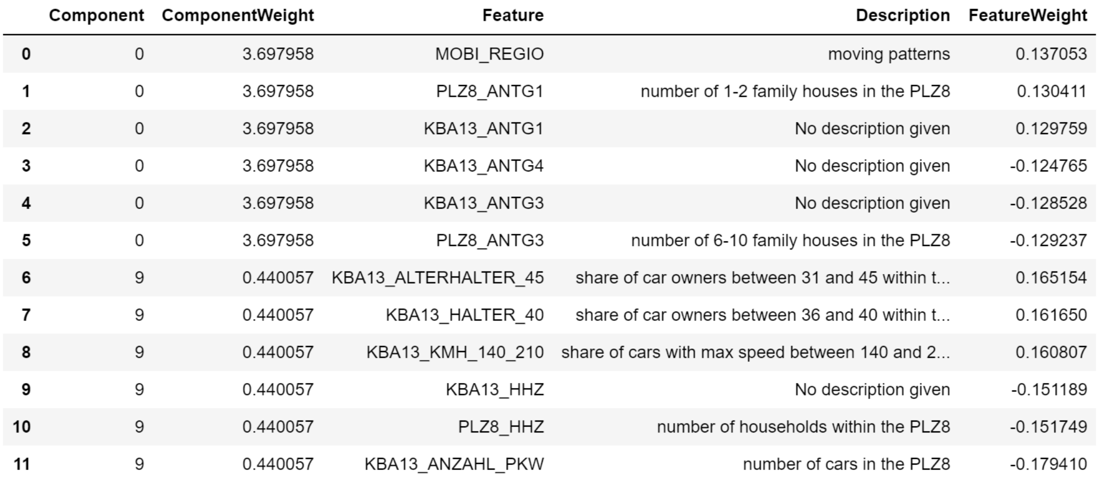


Fig. 5 K-Means elbow plot

K-Means Cluster Analysis

Similar to what we have done with PCA components, we can understand each cluster by analysing what components make up each cluster and what main features make up these components. An example is shown in Table 2.

Table 2 Cluster 3 – Component Analysis



	Component	ComponentWeight	Feature	Description	FeatureWeight
0	0	3.697958	MOBI_REGIO	moving patterns	0.137053
1	0	3.697958	PLZ8_ANTG1	number of 1-2 family houses in the PLZ8	0.130411
2	0	3.697958	KBA13_ANTG1	No description given	0.129759
3	0	3.697958	KBA13_ANTG4	No description given	-0.124765
4	0	3.697958	KBA13_ANTG3	No description given	-0.128528
5	0	3.697958	PLZ8_ANTG3	number of 6-10 family houses in the PLZ8	-0.129237
6	9	0.440057	KBA13_ALTERHALTER_45	share of car owners between 31 and 45 within t...	0.165154
7	9	0.440057	KBA13_HALTER_40	share of car owners between 36 and 40 within t...	0.161650
8	9	0.440057	KBA13_KMH_140_210	share of cars with max speed between 140 and 2...	0.160807
9	9	0.440057	KBA13_HHZ	No description given	-0.151189
10	9	0.440057	PLZ8_HHZ	number of households within the PLZ8	-0.151749
11	9	0.440057	KBA13_ANZAHL_PKW	number of cars in the PLZ8	-0.179410

As seen in Table 3, the to two components that make up this cluster ‘0’ and ‘9’. That means this cluster corresponds to people who like to live in neighborhoods having a smaller number of houses and the houses with a smaller number of families, which can be seen from the feature weights given to the corresponding feature in each component. Also, these people tend to live in neighborhoods which have 31-40 car owners (seen from component 9 feature weights) and like to live where there are a smaller number of cars (seen from last element in the table).

Customer Acquisition

The second part of the project is to use supervised learning algorithms to predict whether a person will be a customer or not based on the demographic data. The file ‘Udacity_MAILOUT_052018_TRAIN.csv’ is provided with the same features as the general population and customers demographic data. An extra column ‘RESPONSE’ has been provided with this data. The response column indicates whether this person was a customer or not. This data has been cleaned by following similar cleaning and processing steps that were followed for general population and customer data.

Benchmark

The first step in the supervised learning is to set a benchmark, which is the base performance with the simplest model possible. This benchmark is set to compare the results from future steps in order to evaluate the used models. The data is split into train and validation splits and a logistic regression model was trained on unscaled training data and evaluated on the unscaled validation data. The benchmark score obtained with the logistic regression model – 0.63 (AUROC score).

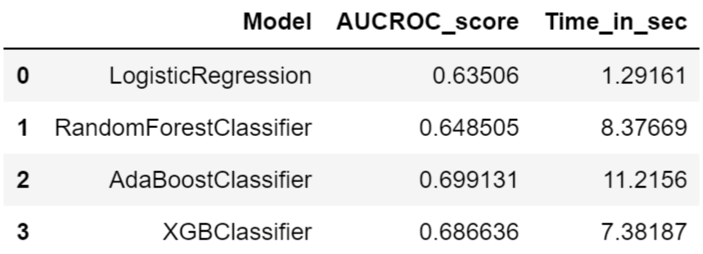
Baseline Performance

After setting the benchmark, the data has been scaled with the standard scaler and is split into training and validation split. Different algorithms have been trained on the training split and have been evaluated on validation split. The algorithms that have been selected for this step are:

- Logistic Regression
- Random Forest Classifier
- AdaBoost Classifier
- XGBoost Classifier

All the selected algorithms can be used for classification tasks. The performance of all the algorithms have been compared with each other and with the benchmark set in the previous step.

Table 3 Performance comparison (scaled data)



	Model	AUCROC_score	Time_in_sec
0	LogisticRegression	0.63506	1.29161
1	RandomForestClassifier	0.648505	8.37669
2	AdaBoostClassifier	0.699131	11.2156
3	XGBClassifier	0.686636	7.38187

The models used here are trained with the default hyperparameters. As seen in the Table 3, the ensemble algorithms outperform all the other models in this case. Random Forest Classifier has a good score. The AdaBoost Classifier and XG Boost classifier have almost similar performance and high score, also can be trained in less amount of time. So, these two algorithms have been selected for the hyperparameter tuning step.

Hyperparameter Tuning

The selected algorithms, Adaboost and XGBoost classifiers have been tuned with the help of a Grid Search. A set of hyperparameters for both the algorithms have been selected for tuning and a grid search has been performed for both the algorithms to determine the best performing models.

Feature Importances

Since the algorithms used here are tree-based models, these algorithms can be analysed futher for the importance these models have given to each feature.

- Adaboost

The feature importances for Adaboost model is shown in Figure 6. The feature ‘D19_SOZIALES’ is having the highest importance which follows by other features.

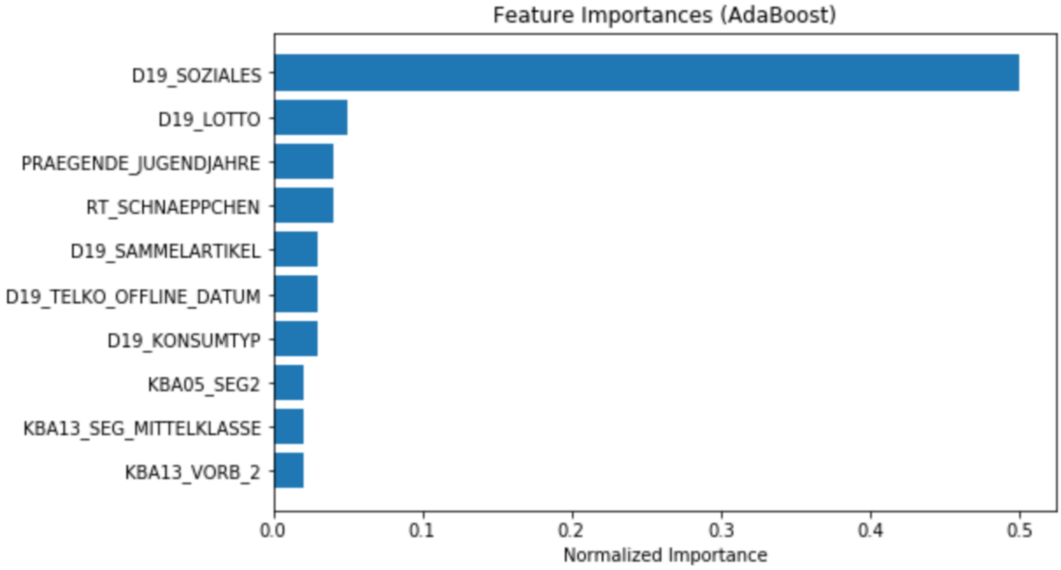


Fig. 6 Adaboost Features Importance

- XGBoost

The feature importances for XGBoost model is shown in Figure 13. The feature ‘D19_SOZIALES’ is having the highest importance which follows by other features.

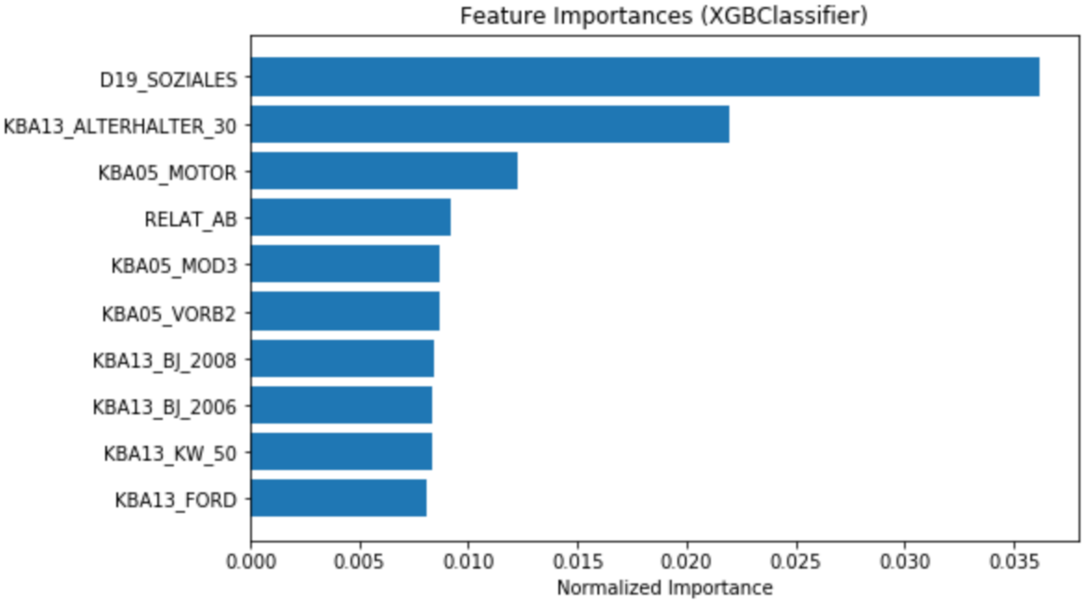


Fig. 7 XGBoost Feature Importance

Both the algorithms have given the highest importance to ‘D19_SOZIALES’ feature. There is no description given in the attribute information files. But as the name suggest and looking at other features with ‘D19_’ as a start, it seems that the feature has something to do with social transactions (this is an assumption and might differ from actual feature description).

The feature importances with the XGboost model seem to be well distributed when compared to AdaBoost model. This might be due to the way these algorithms are designed, Adaboost improves upon weak learners by identifying short comings in the highly weighted data points, where as the XGBoost algorithm improves upon the weak learners with the help of gradients coming from an objective function.

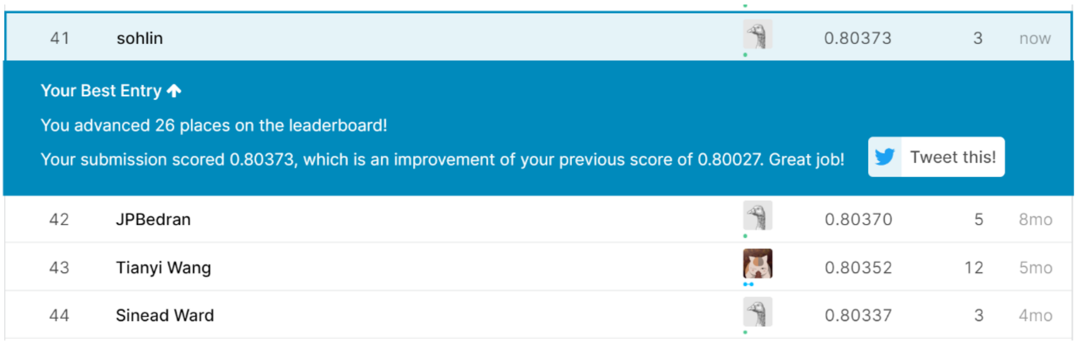
Results

After the hyperparameter tuning, the performance on the validation data with best models resulted in an improvement. The score is shown in the table 4.

Table 4 Hyperparameter Tuning Result

Model	AUROC score
AdaBoost Classifier	0.7431
XG Boost Classifier	0.7478

Also, as to submit the test file to Kaggle, the final result is as Figure 8, the test score is **0.80373**, as it is only 30% of test data being used.



41	sohlin	0.80373	3	now

Your Best Entry ↑

You advanced 26 places on the leaderboard!

Your submission scored 0.80373, which is an improvement of your previous score of 0.80027. Great job!

42	JPBedran	0.80370	5	8mo
43	Tianyi Wang	0.80352	12	5mo
44	Sinead Ward	0.80337	3	4mo

Fig. 8 Kaggle Leaderboard

Improvements and Discussion

With the performed pre-processing steps and modelling, a top score has been achieved. There is still scope for improvement. Future steps include:

- ➢ Dealing with more categorical features and one-hot encoding them
- ➢ Understanding more features and selecting relevant ones

References

[1] Arvato-Bertelsmann, "Arvato," Bertelsmann, [Online]. Available: https://www.bertelsmann.com/divisions/arvato/#st-1. [Accessed April 2020].

[2] Bertelsmann, "Company," Bertelsmann, [Online]. Available: https://www.bertelsmann.com/company/. [Accessed April 2020].

[3] S. M. Lador, "What metrics should be used for evaluating a model on an imbalanced data set? (precision + recall or ROC=TPR+FPR)," Towards Data Science, 2017. [Online]. Available: https://towardsdatascience.com/what-metrics-should-we-use-on-imbalanced-data-set-precision-recall-roce2e79252aeba. [Accessed April 2020].

[4] A. Kassambara, "Determining The Optimal Number Of Clusters: 3 Must Know Methods," DataNovia, [Online]. Available: https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters3-must-know-methods/. [Accessed April 2020].