UDACITY

Machine Learning Nanodegree Program

Capstone Project Report: Customer Segmentation Report for Arvato Financial Services

---

Madhav

JUNE 11th, 2021

Table of Contents

I. Definition

Project Overview

"Customer Segmentation Report for Arvato Financial Solutions" is one of the projects proposed for the Machine Learning Engineer Nanodegree Program by the Udacity. The main objective of the project is to find important features in the available dataset and predict whether the given customer can be in targeted audience or not.

The dataset is already provided by Bertelsmann Arvato Analytics, that have demographic information about the general population of Germany and also the current customers of the company, targeted mailout campaign outcomes, and two files with a description of the demographic features.

The project is divided into several subtasks:

1. Data Analysis and Preprocessing;
2. Customer Segmentation Report;
3. Supervised Learning Predictive Model;
4. Kaggle Competition;

The entire dataset is protected under the Terms and Conditions and is unavailable for public sharing.

Domain Background

"Arvato is an international based service sector company providing it's various services through software and hardware technologies with a focus on innovations in automation and data/analytics. Globally renowned companies from various domains such as telecommunications providers , IT services, Net

Banking, e-commerce– rely on Arvato's portfolio of solutions. Arvato is fully owned by Bertelsmann. [1]"

Arvato is looking all the possible ways to support a client-base(mail-order company selling organic products) with available datasets to find the target audience for their campaign. To achieve this, I will use the available dataset to segment the customers based on their interest with attributes and demographic

feature that are more suitable. With the help of Data Analysis and Machine Learning concepts many underlying patterns can be discovered and also handle very high volumes of data efficiently.

arvato
BERTELSMANN

Problem Statement

The Problem Statement for this project is pretty straight forward "How can an existing mail order company selling organic products can acquire more number of clients in order to expand their business?" The proposed solution is divided into 2 subparts. The first part would deal with the segmentation of the customers using unsupervised machine learning techniques based on the intersection of the data with current customers and population provided. Secondly, a supervised model will be used to predict the outcome whether or not a customer is likely to be in the targeted audience or not.

Metrics

For the dimensionality reduction algorithm PCA, a plot between experienced variance and number of components can be taken into consideration to choose number of components.

The supervised learning belongs to the binary classification problem and it has highly imbalanced data.

```
: sns.countplot(x ='RESPONSE', data = mailout_train)
: <matplotlib.axes._subplots.AxesSubplot at 0x7fcf16074f50>
```

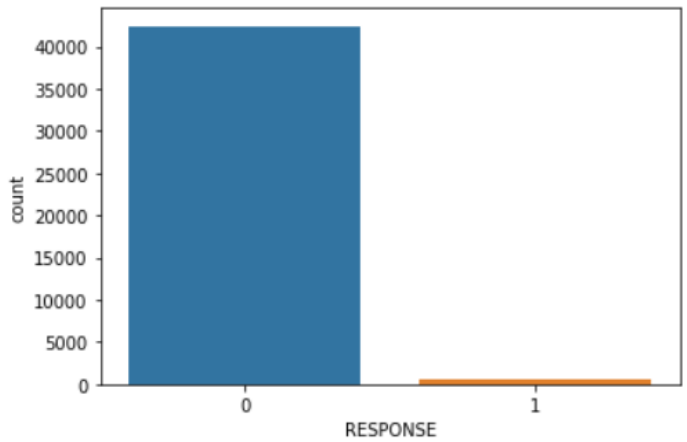


Fig 1 Response balance in the training dataset

Due to high imbalanced data, the evaluation metrics chosen is Area Under the Curve Receiver Operating Characteristics (AUC-ROC).

The AUC-ROC is used to visualize the True Positive Rate against False Positive Rate. Then AUC equals to 1, meaning that True Positives and True Negatives are disjoined and perfectly distinguishable, while AUC equals 0 means that the models makes exact opposite classification (all true negatives are classified as positives and vice versa).

II. Analysis

Datasets and Inputs

The dataset is already provided by Bertelsmann Arvato Analytics, following are the files along with their description provided for 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, there were 2 more files for describing attributes:

- **DIAS Attributes - Values 2017.xlsx**: Explains values encoding.
- **DIAS Information Levels - Attributes 2017.xlsx**: Explains column names meanings.

MAILOUT_TRAIN and MAILOU_TEST are provided for the building training and testing a supervised model.

Data Exploration

Warning Message

When dataset gets loaded a warning message is been displayed

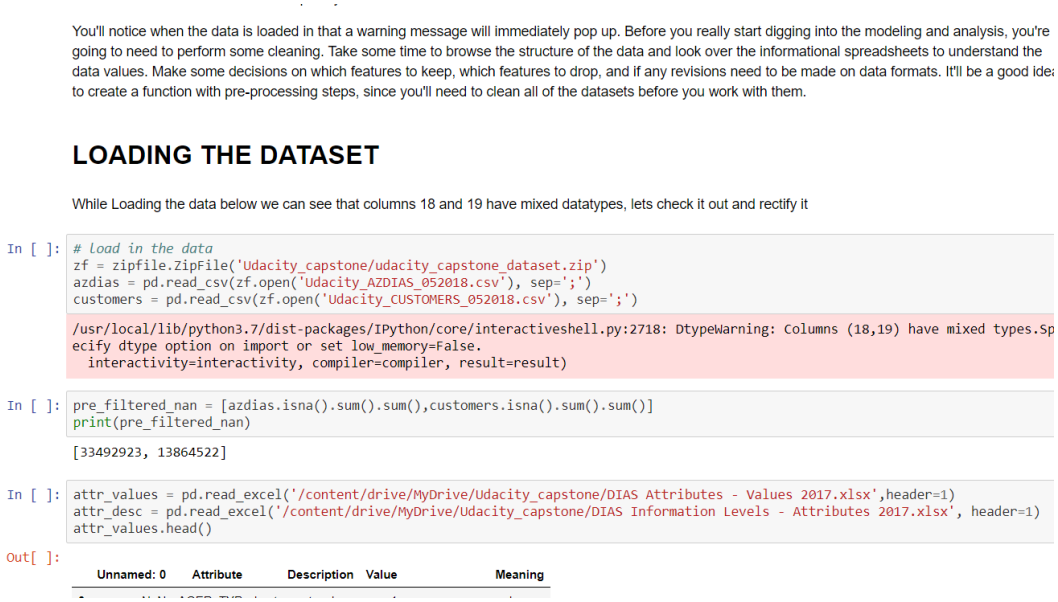


Fig. 2 The warning message

This warning occurred due to occurrence of combinations of various types.

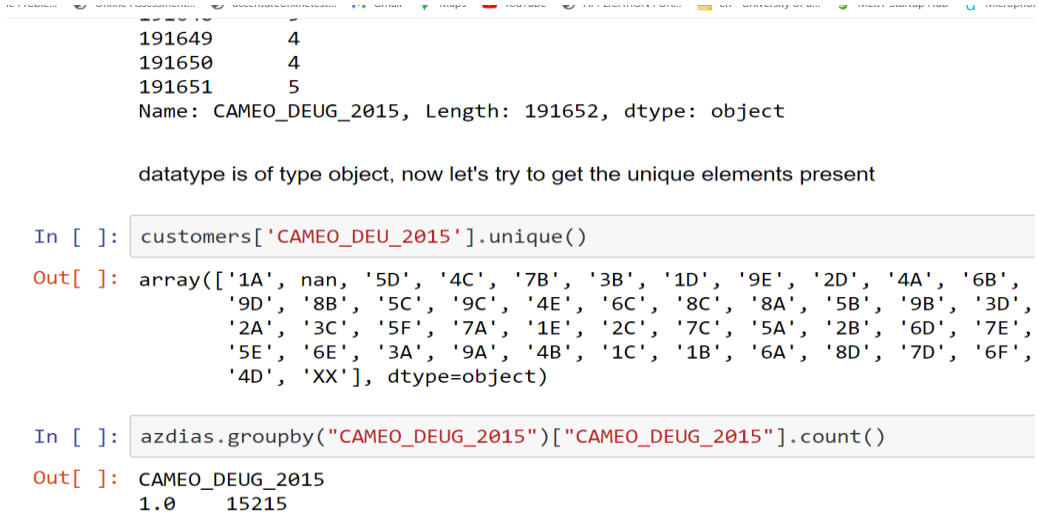


Fig. 3 Groups of the unique values from the CAMEO_DEUG_2015 column

Above figure clearly indicates that X/XX are two different types which need to be replaced by numpy.nan .

Value types for encoding

Because of the warning message about mixed types in columns 18 and 19, other columns of 'Object' type have been verified and listed down.

D19_LETZTER_KAUF_BRANCHE	EINGEFUEGT_AM	OST_WEST_KZ
NaN	NaN	NaN
NaN	1992-02-10 00:00:00	W
D19_UNBEKANNT	1992-02-12 00:00:00	W
D19_UNBEKANNT	1997-04-21 00:00:00	W
D19_SCHUHE	1992-02-12 00:00:00	W

Fig. 4 The snapshot of the table with columns that store object-type values

Since Machine cannot understand categorical input following columns need to be label encoded:

```
CAMEO_INTL_2015, LP_LEBENSPHASE_FEIN, LP_FAMILIE_GROB, LP_STATUS_GROB, EINGEFUE
GT_AM, D19_LETZTER_KAUF_BRANCHE, OST_WEST_KZ, PRODUCT_GROUP, CUSTOMER_GROUP
```

Determine Missing Data

The next immediate step is to collect labels for unknown values from the "DIAS Attributes - Values 2017.xlsx" and iterate them over the entire dataset and replacing all unknowns with np.Nan.

```
Pre-filtered Cusmomer df, no of NaN values:  13864900
Post-filtered Cusmomer df, no of actual NaN values:  14488847
Pre-filtered Azdias df, no of NaN values:  33494042
Post-filtered Azdias df, no of actual NaN values:  37088636
```

Fig. 5 Results of the replacement unknown values with NaN

Determine Outliers

In DIAS Attributes - Values 2017.xlsx, there are descriptions for each column values, anything beyond those range is definitely a mistake. A dictionary is defined which stores the info on the proper ranges with key as feature and value as array with encodings.

After comparing the azdias dataset with dictionary many columns with potential outiers exist.

GEBURTSJAHR has a birth year 0 which seems senseless. In case of KBA05_MODTEMP, LP_FAMILIE_FEIN, LP_FAMILIE_GROB, LP_LEBENSPHASE_FEIN, LP_LEBENSPHASE_GROB, ORTSGR_KLS9 all other values except than nan are supposed to be exist

The above mentioned column values were converted into no.NaN values, whilst others were crapped up by IQR method

Overcategorized data

Surprisingly, when looking for ranges in the dictionary I came across some columns that are over categorised

```
CAMEO_DEUG_2015, => 10
CAMEO_DEU_2015, => 44
CAMEO_DEUINTL_2015, => 26
CJT_GESAMTTYP, => 7
D19_BANKEN_ANZ_12, => 7
D19_BANKEN_ANZ_24, => 7
D19_BANKEN_DATUM, => 10
D19_BANKEN_DIREKT_RZ, => 8
D19_BANKEN_GROSS_RZ, => 8
D19_BANKEN_LOKAL_RZ, => 8
D19_BANKEN_OFFLINE_DATUM, => 10
```

Fig 6 A snapshot from the categories calculations

Following is the list of columns that are supposed to be overwritten unless they will not be removed:

```
ALTER_HH, CAMEO_DEU_2015, CAMEO_DEUINTL_2015, LP_FAMILIE_FEIN, LP_FAMILIE_GROB,
LP_LEBENSPHASE_FEIN, LP_LEBENSPHASE_GROB, LP_STATUS_FEIN, LP_STATUS_GROB, PRAEG
ENDE_JUGENDJAHRE.
```

Algorithms and Techniques

There are a couple of steps to be performed towards the solution.

1. Segmentation

   Here, the dimensionality reduction technique known as Principle Component Analysis will be used to reduce the dimensions because original data contains 366 features not all of them play a crucial role. Afterwards elbow method will be used to find appropriate number of clusters to be used in the K-Means technique.
2. Predictive modeling

   In last section a predictive model using supervised machine learning models will be build such as XGBoost, ADABoost, RandomForests etc

   KNeighborsClassifier

   “Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point [5]”.

   RandomForestClassifier

   “A diverse set of classifiers is created by introducing randomness in the classifier construction. The prediction of the ensemble is given as the averaged prediction of the individual classifiers [7]”.

   AdaBoostClassifier

   “The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessings, such as small decision trees) on repeatedly

modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction [8]".

GradientBoostingClassifier

GradientBoosting "is an accurate and effective off-the-shelf procedure that can be used for both regression and classification problems in a variety of areas [9]". ✞ XGBoost Classifier

Benchmark

The first base model that strikes for any binary classification model is Logistic Regression.

After fitting the data into the Logistic Regression model the following result was obtained:

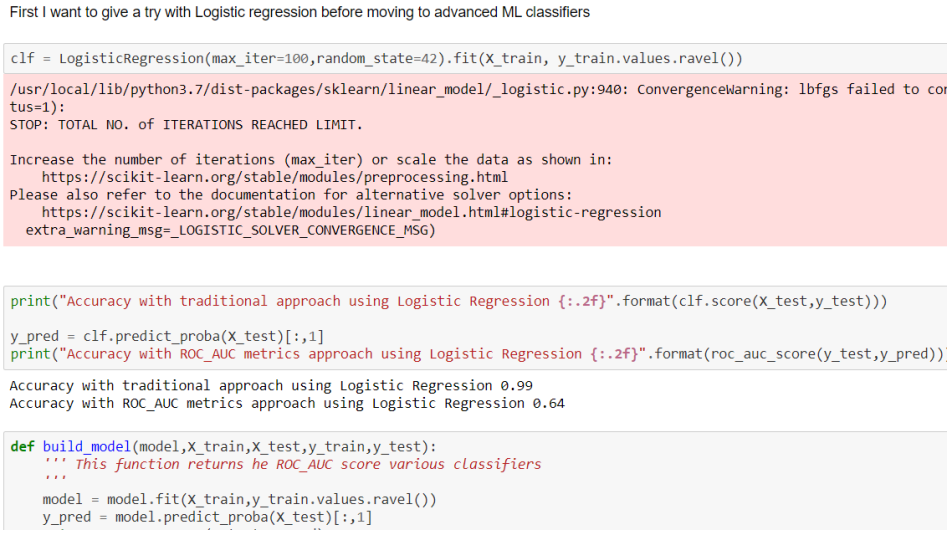


Fig 7 The Logistic Regression evaluation

From above it is concluded that traditional accuracy metrics leads to inconsistent results, therefore roc_auc metric is used for evaluation.

The next step is to calculate roc_auc score for various advanced classification models with default parameters those include

- ✞ KNeighborsClassifier
- ✞ RandomForestClassifier
- ✞ AdaBoostClassifier
- ✞ GradientBoostingClassifier
- ✞ XGBoost Classifier

The outcome of the training was as follows:

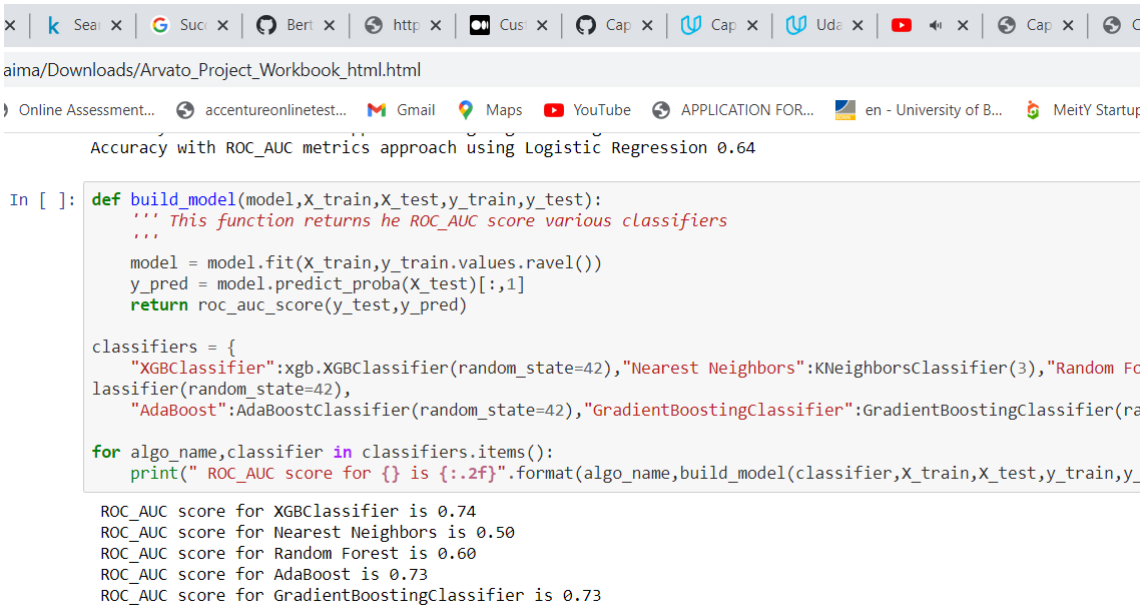


Fig 8 The table of Benchmark model with scores and training time values

RandomForest and Nearest Neighbours can easily be ruled out from the list. Well after receiving an roc_auc score of 0.74 by XGBClassifer and 0.73 with AdaBoost and GradientBoostingClassifier I had directly applied this to test data and uploaded to kaggle competitions but kagggle score was around 0.45. Therefore, hyper parameter tuning needs to be done for above three algos.

III. Methodology

Data Preprocessing

Following are the various data pre-processing techniques that have been followed

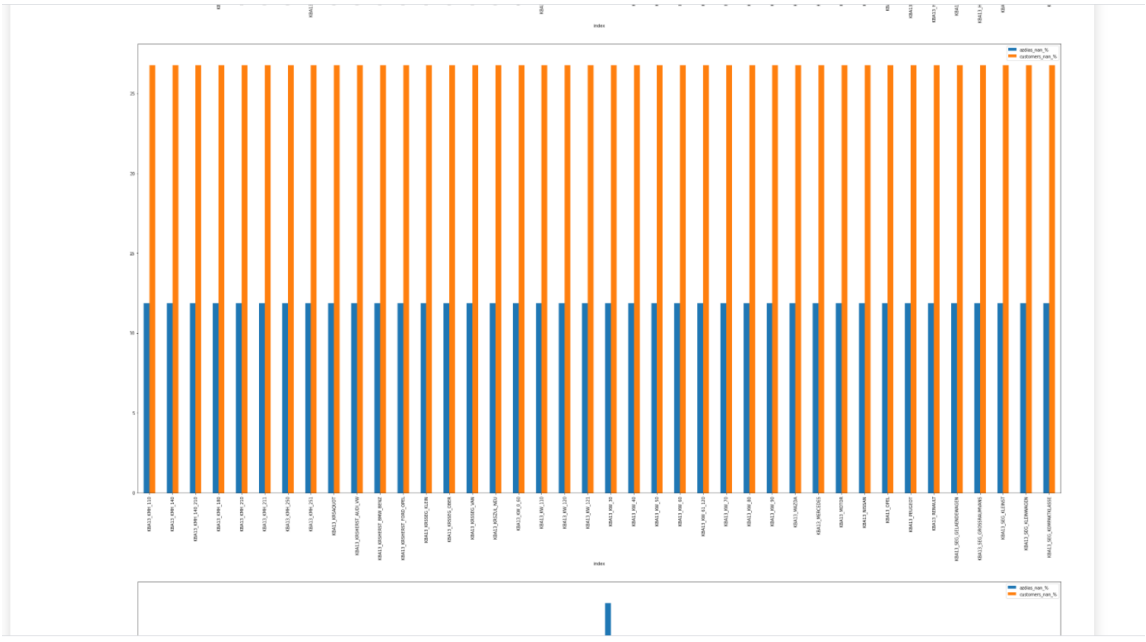
1. Handling mixed types for columns 18 and 19;
2. Replace encoded unknown values with np.nan;
3. Verify whether `ANZ_HAUSHALTE_AKTIV, ANZ_HH_TITEL, ANZ_PERSONEN, ANZ_TITEL, KBA13_ANZAHL_PKW` columns have outliers, if so, then remove.
4. Calculate the percentage of NaN values by columns and rows. Drop columns where the percentage is higher than 30, drop row where the percentage is higher than 50.
5. Calculate the correlation of the features, drop columns where the correlation percentage is higher than 90.
6. Feature engineering;
7. Feature scaling;

Cleaning the missing data

Fig. 9 The representation of the missing entries for the AZDIAS dataset

The process of cleaning the missing data was separated into several steps:

1. Calculate the percentage of the missing values by columns and rows for AZDIAS and CUSTOMERS datasets, then merge those results.



```
merged_dataframe = azdias_missing_cols.reset_index()
merged_dataframe = merged_dataframe.drop('count',axis=1)
merged_dataframe.rename(columns={'percentage':'azdias_nan_%'}, inplace=True)
temp = customer_missing_cols.reset_index()
temp = temp.drop('count',axis=1)
temp.rename(columns={'percentage':'customers_nan_%'}, inplace=True)
merged_dataframe = merged_dataframe.merge(temp,on='index')
```

Fig 11 The Bar plot for the missing data from both AZDIAS and CUSTOMERS

2. Find columns where the percentage of missing data was higher than 30% in both the datasets and delete them.

3. The next step was to delete columns which had no description because in later part these columns makes no sense at all.

4. There are also many rows which have missing values more than 50%, such rows were identified and dropped.

Feature Engineering and Imputation

As mentioned above, now it is the time to deal with the overcategorized data and missing values.

1) Map CAMEO_INTL_2015 into CAMEO_INTL_2015_wealth_type and CAMEO_INTL_2015_family_type

2) Map LP_LEBENSPHASE_FEIN into LP_LEBENSPHASE_FEIN_family_type, LP_LEBENSPHASE_FEIN_earner_type, LP_LEBENSPHASE_FEIN_age_group

3) PRAEGENDE_JUGENDJAHRE using movement(mainstream or avangard) and decade.

4) The rest of the missing column values can undergo imputation with the 'most frequent' strategy if values in columns are categorical and if they are numerical then imputation with the column mean is used.

Remove highly correlated features

Highly correlated columns crossing the threshold 90% have been dropped, following is the list

```
['CAMEO_DEUG_2015', 'D19_VERSAND_DATUM', 'D19_VERSAND_ONLINE_DATUM', 'D19_VERSA
ND_ONLINE_QUOTE_12', 'KBA13_HALTER_66', 'KBA13_HERST_SONST', 'KBA13_KMH_250', '
LP_LEBENSPHASE_GROB', 'LP_STATUS_GROB_earner_type', 'PRAEGENDE_JUGENDJAHRE_move
ment']
```

Feature Scaling

Before moving to the dimensionality reduction, it is recommended to apply feature scaling to the dataset.

"This transformation does not change the distribution of the feature and due to the decreased standard deviations, the effects of the outliers increases. Therefore, before normalization, it is recommended to handle the outliers [10]". The scaling technique used here is MinMax scalar which maps all values between 0 and 1.

Implementation

Principal Component Analysis (PCA)

Since there are 255 features the first step is to understand what number could represent the variance more efficiently.

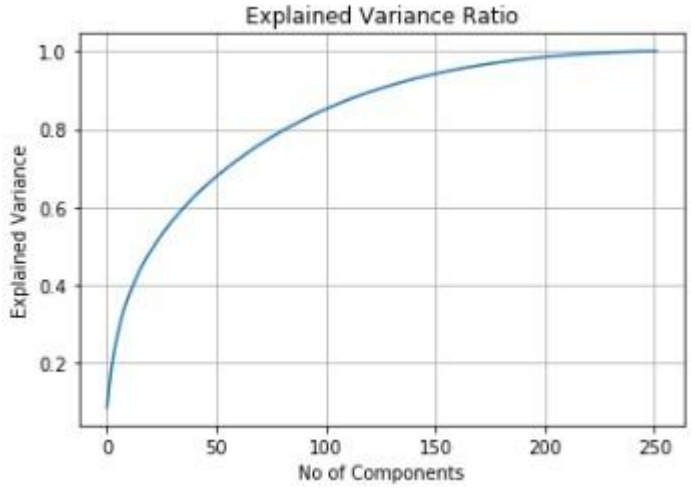


Fig. 13 Explained Variance Ration

So, from above plot, It can be observed that if I can take number of components as 125 I tend to cover around 90% of the data

Elbow Method

The elbow method is used to find the number of clusters. "The elbow method is a heuristic used in determining the number of clusters in a data set. The method consists of plotting the explained variation as a function of the number of clusters and picking the elbow of the curve as the number of clusters to use [11]."

"The elbow method runs k-means clustering on the dataset for a range of values for k (say from 1-10) and then for each value of k computes an average score for all clusters. By default, the distortion score is computed, the sum of square distances from each point to its assigned center [12]."

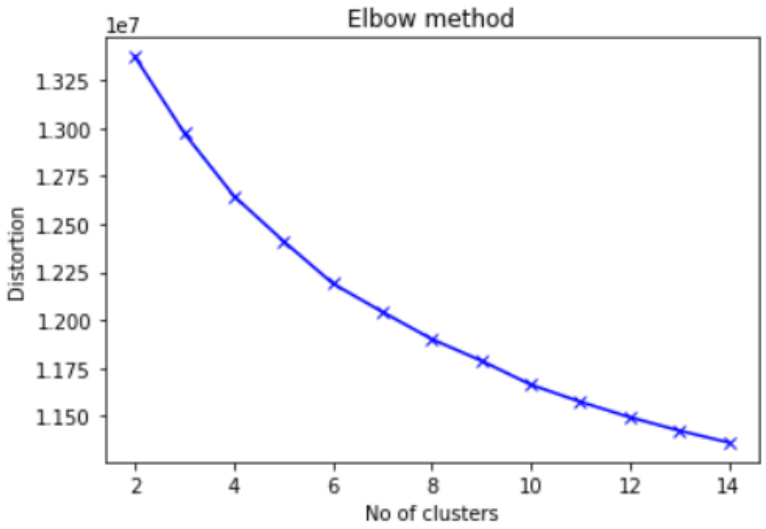


Fig. 14 Elbow graph for 150 components

For the given data, I conclude that the optimal number of clusters for the data is 9., as the metric stops to rapidly decrease after this point.

K-Means

"The KMeans 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 (see below). This algorithm requires the number of clusters to be specified. It scales well to a large number of samples and has been used across a large range of application areas in many different fields [13]."

The following results were obtained by applying k=9 in K-Means algorithm to transformed data

```
In [ ]: clusters_dataframe[['percentage_of_azdias','percentage_of_customers']].plot(kind='bar',figsize=(15,8))
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0b09955e90>
```

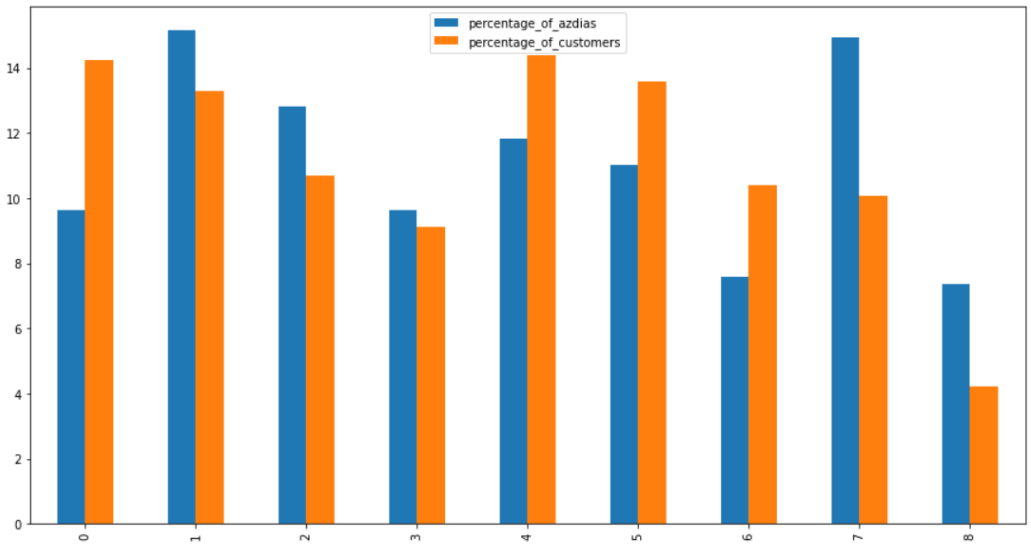


Fig. 15 9 clusters for customers segmentation

I am only interested in clusters 0,4,5,6 where percentage of customers is higher than azdias which indirectly represents features for targeted audience. While 7,8 represent features which exclude core customers.

Refinement

Hyperparameters tunning

For AdaBoost classifier, As an optimization option GridSearchCv was implemented. "The grid search provided by GridSearchCV exhaustively generates candidates from a grid of parameter values specified [14]."

```
from sklearn.model_selection import GridSearchCV
params = {
     'n_estimators': np.arange(25,200,25),
     'learning_rate': [0.01, 0.05, 0.1, 1],
 }
grid_cv = GridSearchCV(AdaBoostClassifier(random_state=42), param_grid= params,
 cv=5, n_jobs=-1,scoring = "roc_auc")
```

The results were as follows:

```
grid_cv.best_score_,grid_cv.best_params_
(0.7603690617450454, {'learning_rate': 0.01, 'n_estimators': 50})
```

```
The same was applied to xgb.XGBClassifier and the results was
xgb.XGBClassifier(eta= 0.05, max_depth= 5, min_child_weight= 10, n_estimators=
300) ➔ Best parameters
```

The roc_auc score for xgb is lower than other two so it can be ruled out.

For Ada it was 0.75 but for Xgb it is around 0.71

The same technique was applied to GradientBoosting algorithms, but the results were not convincing. Well GradientBoosting dominated AdaBoost in terms of overall prediction but for test data AdaBoost out traced GradientBoostingClassifier. Since, this is a competition I have send both the submissions to kaggle GradientBoostingClassifier score was around 0.57 while AdaBoost score was around 0.78. GradientBoosting migh have overfitted with the data leading to score on MailOut test data

IV. Results

Out of those three, AdaBoostClassifier probabilities derived from MailOutTest dataset were submitted to Kaggle competition

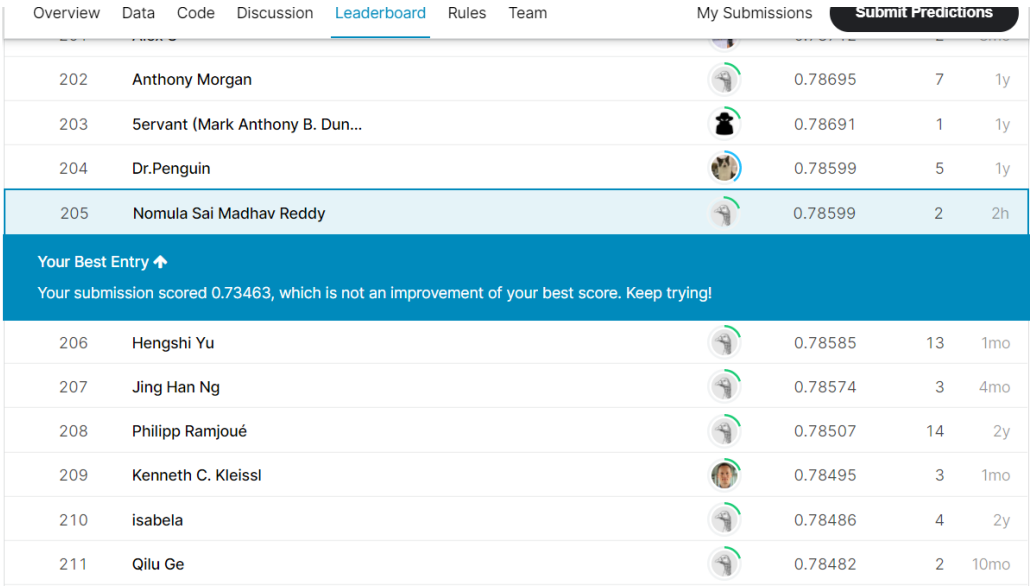


Fig. 20 Kaggle's score

The private score of the final performance of my model was 0.78, on Kaggle competition it managed to hold 0.78 of accuracy as well according to the AUC-ROC evaluation metric, which is a good result. However, there is plenty of room for improvement.

Justification

When compared to baseline model Logistic Regression, the final private score on Kaggle is quite better. But, that's not the best as said above there is a plenty of room for improvement since the expected score should be more than 0.90. But even, my solution adequately handled the proposed problem statement.

V. Conclusion

Customer-centric marketing is a field where more business are looking to make it automate with the use of machine learning instead of traditional approach.

Customer Segmentation Report for Arvato Financial Services project has brought me with mnay different experiences. Firstly, I have never handled seen such dataset before and never performed these any pre-processing methods before. Secondly, this is a representation of real life Data Science and Data Analytics project which itself makes me feel unique from others.

Coming to the result, there is still scope for modification and improvement, instead of these classifiers we can go with Neural Networks concept. And, it is also good to again re-think about Data pre-processing techniques.

References

[1] Arvato-Bertelsmann, "Arvato", Bertelsmann. [Online].

Available: https://www.bertelsmann.com/divisions/arvato/#st-1 [Accessed JUNE 9TH, 2021]

[5] Scikit-learn, "Nearest Neighbors Classification". [Online].
Available: https://scikit-learn.org/stable/modules/neighbors.html#classification/ [Accessed June 5th. 2021]

[6] Scikit-learn, "Forests of randomized trees". [Online].
Available: https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees [Accessed June 9th, 2021]

[7] Scikit-learn, "AdaBoost". [Online].
Available: https://scikit-learn.org/stable/modules/ensemble.html#adaboost [Accessed June 7th, 2021]

[8] Scikit-learn, "Gradient Tree Boosting". [Online].
Available: https://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting [Accessed June 8th, 2021]

[9] Wikipedia. "Elbow Method". [Online].
Available: https://en.wikipedia.org/wiki/Elbow_method_(clustering) [Accessed June 8th, 2021]

[10] Scikit-Yellowbrick, "Elbow Method". [Online].
Available: https://www.scikit-yb.org/en/latest/api/cluster/elbow.html#/ [Accessed JUNE 7th, 2021]

[11] Scikit-learn, "Clustering". [Online].
Available: https://scikit-learn.org/stable/modules/clustering.html#k-means/ [Accessed June 8th, 2021]

[12] Scikit-learn, "Tuning the hyper-parameters of an estimator". [Online].
Available: https://scikit-learn.org/stable/modules/grid_search.html#grid-search/ [Accessed June 8th , 2021]

https://towardsdatascience.com/how-customer-centric-marketers-use-machine-learning387df1a33850 [Online]. Available: https://towardsdatascience.com/how-customer-centric-marketers-use-machinelearning-387df1a33850 [Accessed June 5th, 2021]