# Section 1: Project Definition

## Project Overview:

In this project, we will analyze demographics data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population. we'll use unsupervised learning techniques to perform customer segmentation, identifying the parts of the population that best describe the core customer base of the company. Then, we'll apply what we've learned on a third dataset with demographics information for targets of a marketing campaign for the company, and use a model to predict which individuals are most likely to convert into becoming customers for the company. The data that we will use has been provided by Udacity partners at Bertelsmann Arvato Analytics, and represents a real-life data science task.

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).

Each row of the demographics files represents a single person, but also includes information outside of individuals, including information about their household, building, and neighborhood.

The "CUSTOMERS" file contains three extra columns ('CUSTOMER\_GROUP', 'ONLINE\_PURCHASE', and 'PRODUCT\_GROUP'), which provide broad information about the customers depicted in the file. The original "MAILOUT" file included one additional column, "RESPONSE", which indicated whether or not each recipient became a customer of the company. For the "TRAIN" subset, this column has been retained, but in the "TEST" subset it has been removed; it is against that withheld column that our final predictions will be assessed in the Kaggle competition.

Otherwise, all of the remaining columns are the same between the three data files. For more information about the columns depicted in the files, you can refer to two Excel spreadsheets provided in the repo. [One of them](./DIAS Information Levels - Attributes 2017.xlsx) is a top-level list of attributes and descriptions, organized by informational category. [The other](./DIAS Attributes - Values 2017.xlsx) is a detailed mapping of data values for each feature in alphabetical order.

Problem Statement:

We will use the information from the first two files to figure out how customers ("CUSTOMERS") are similar to or differ from the general population at large ("AZDIAS"), then use our analysis to make predictions on the other two files ("MAILOUT"), predicting which recipients are most likely to become a customer for the mail-order company.

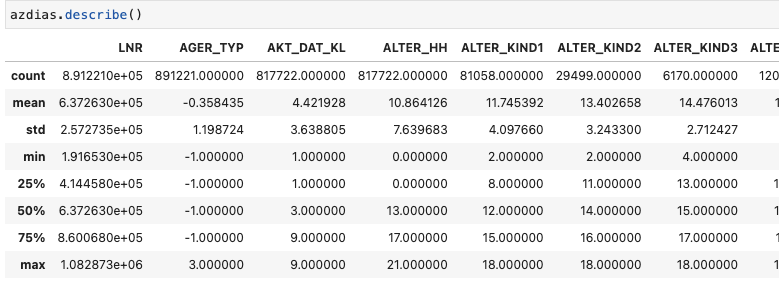
Metrics:

An AUC-ROC curve from predicted probabilities will be used to evaluate the performance of the classification models. AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AU C, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between classes. Details can be read from [this](https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5).

# Section 2: Analysis

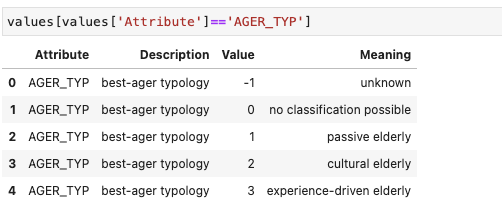
## Data Exploration/Visualization

I have started with azdias(german population data) because it is the largest and contains nearly all features.



As it can be seen there are so many features to analyze. To gain some time, we may want to eliminate less important features while looking at missing ratio. Higher missing ratio leads to low information gain so may consider to drop these features.

Let’s look at missing value distribution in terms of row basis. We have 93 feature with completely filled values. So I have checked the excel file, it seems that there are features with special values which means that some of missing values are assigned to spesific value. So we need to do conversion to calculate missing ratios correctly.



As it can be seen that the dataset has special values to indicate the missing values. Therefore, we need to convert values to null to evaluate missing values correctly. I have prepared a script that fixes that issue.



After conversion, it seems that we don't have that many missing columns. We can drop if they are higher than %20. So, we will drop 17 attributes. Still, we have so many missing features. We may want to drop highly cardinal variables. Because, most of ML algorithm requires one-hot encoding to use categorical variables. So that high cardinal variable creates issue in that sense. The following variables assigned as categorical; actions are also shown in the below.

CAMEO\_DEUG\_2015: CAMEO classification 2015 - Upper group. It is like some kind of classification we may want to keep it.

CAMEO\_DEU\_2015: CAMEO classification 2015 - detailed classification. We can drop it because we already have upper group segmentation

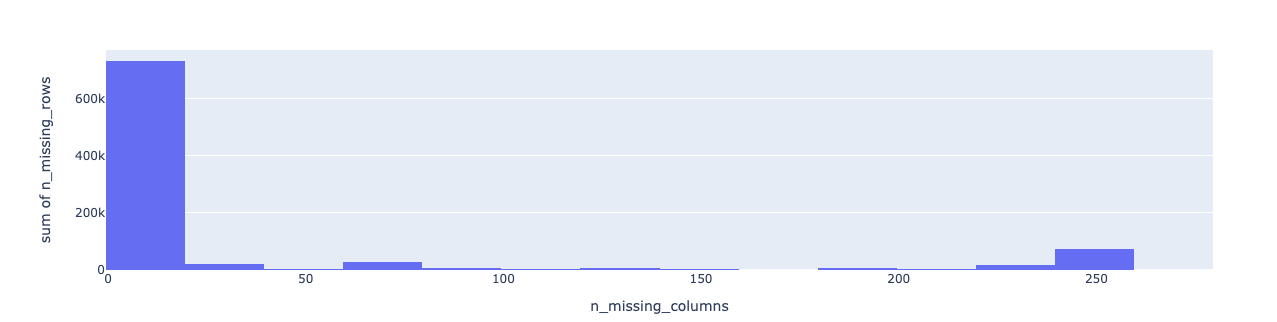
CAMEO\_INTL\_2015: we don’t' have info about this feature we can drop it.

D19\_LETZTER\_KAUF\_BRANCHE: we don’t' have info about this feature we can drop it.

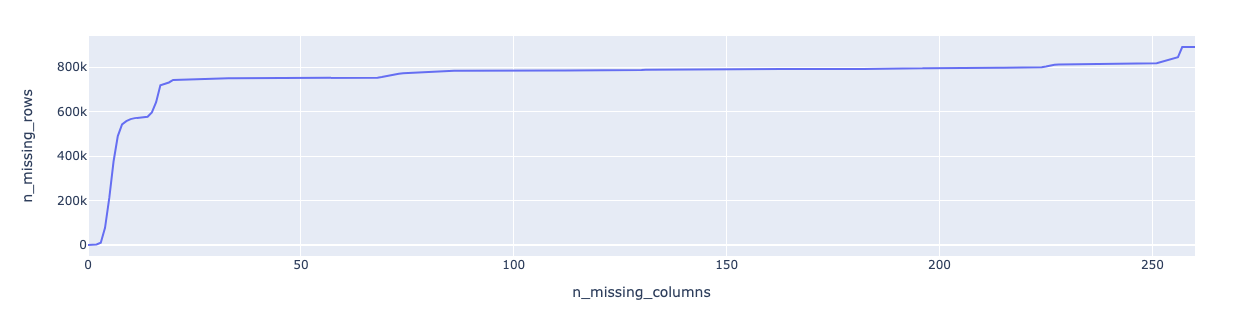
EINGEFUEGT\_AM we have so many features so we can drop time related attributes.

OST\_WEST\_KZ: flag indicating the former GDR/FRG. We can convert it to numerical attribute

Apart from that, we have 4 data frames, so we need to drop non-common features. After dropping all necessary columns, we can do missing row investigation for each row. AS a result, we can check whether we need to drop observations due to high missing columns.



It seems that majority of the population have 0-19 missing columns. To be sure on this let’s do it by cumulatively.



After 21, trend normalizes so we can choose 21 for missing rows elimination.

# Section 3: Methodology

## Data Preprocessing:

There are 2 different problems in our cases first one is identifying how much companies’ customer segments similar to the German population which is unsupervised methodology because we don’t know the labels. Second one is that selected groups are going to response to our campaigns which is supervised because we have already labeled data. For unsupervised learning, we have used k-means clustering algorithm and PCA to decrease the size of the features. Clustering algorithms are very sensitive to the data, so we needed to do missing imputations, outlier elimination, standardizations and dimension reduction. We have decided to following transformations for each data types.

Binary: we need to determine 2 value features. We can assign missing values as most popular value.

Categorical: we need to do one-hot encoding to convert categorical to binary features

Numerical: we can use median value to impute missing values to overcome outliers’ effects.

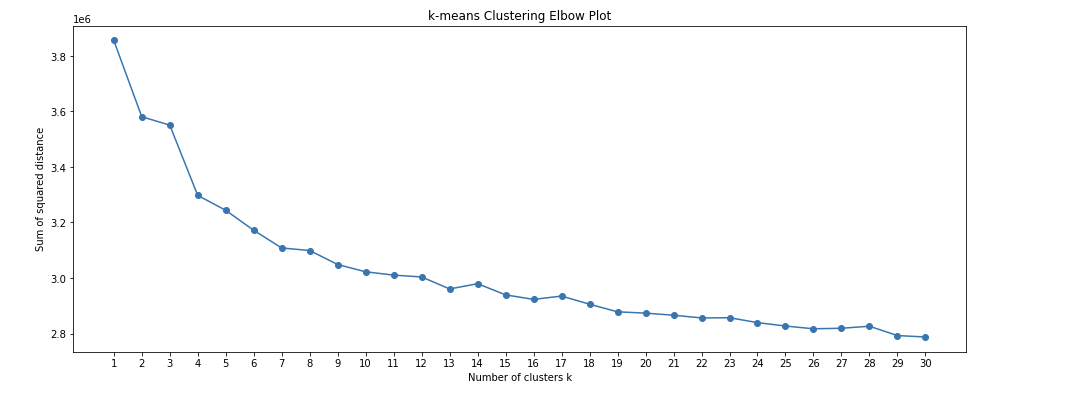
For the supervised learning side, I have only applied XGBoost model due to time constraint and memory issues. We have invested time on hyperparameter tunning. For the XGBoost there is no need for feature transformation because decision tree algorithms handle all missing values, outliers etc.

## Implementation:

While doing transformation for clustering algorithm, we have faced with memory issue because we have many features. So, we have decided to drop correlated features which means that they have similar meaning. Looking at correlation also took so long but it is important that we did that only once however for transformation or PCA or other algorithms we had needed to deal with these problem multiple times. Therefore, dropping correlated attributes improved our performance on the overall. In addition to that, we had to random resampling while doing clustering because again we hit the memory issue.

## Refinement:

As we have stated before, we have used sklearn's KMeans clustering algorithm to identify different segments of the data. In KMeans clustering we had to choose the number of clusters that the algorithm will find. This can be difficult since we don't know how many clusters exist. So we choose number of clusters with elbow method.



There was no certain elbow in the graph however it seemed that there is slow decrease after 13th cluster.

For the XGBoost algorithm, we have split train dataset as test and train then used GridSearchCV to automate the tuning of hyperparameters. Also, we have unbalanced dataset. For imbalanced dataset, we needed to arrange scale\_pos\_weight parameter. It is documented that it is better to use if there is imbalanced data.

scale\_pos\_weight = total\_negative\_examples / total\_positive\_examples

As a result, the following parameters are chosen as best.

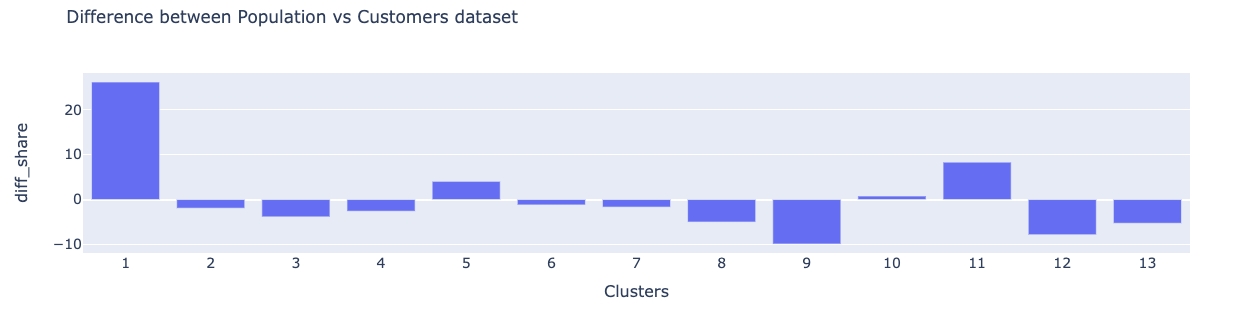
{'gamma': 1.0, 'learning\_rate': 0.05, 'max\_depth': 2, 'reg\_lambda': 0, 'scale\_pos\_weight': 80}

# Section 4: Results

## Model Evaluation, Validation, Justification:

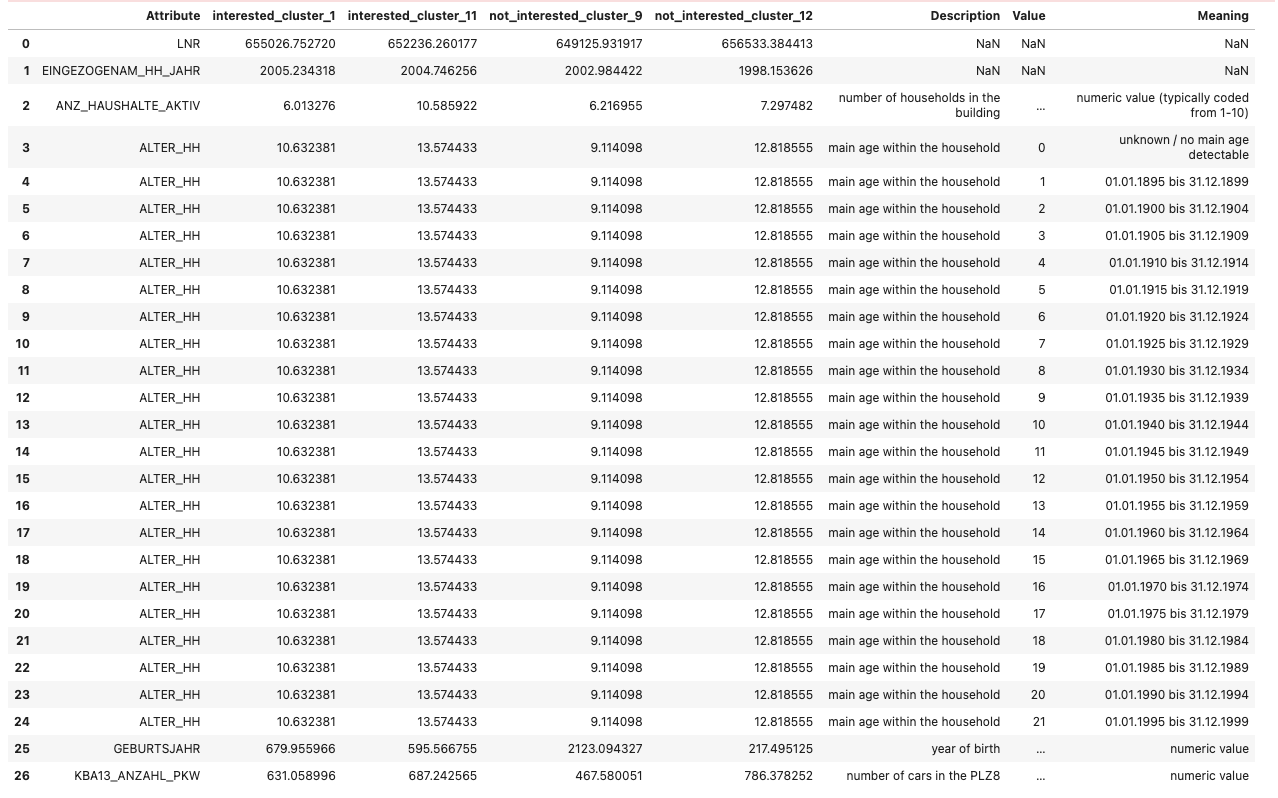
As a result, we have found out 13 clusters like in the following.





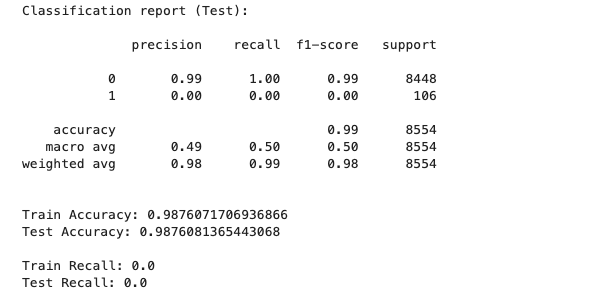
It seems that 1st group targeted most. 11th, 5th and 10th clusters follow it accordingly. 9th cluster customers are outside of the company focus group because they are dominating population but not in companies customer segment. 12th and 13th groups follow it accordingly.

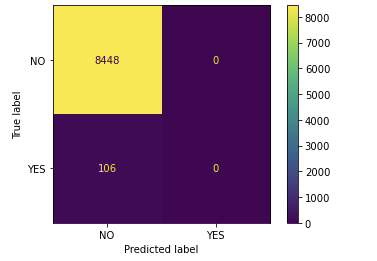
When we look at clusters’ details to understand the groups behavior, the following tabular data show us features values on specified clusters.



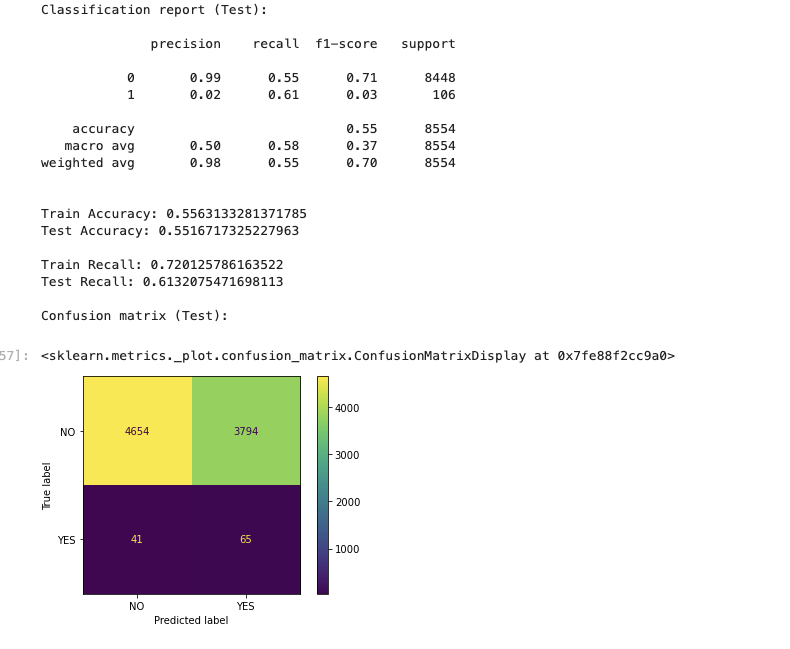
As a result, clusters are very close to each other so to be able to interpret correctly we can focus on most focused and unfocussed groups which are cluster 1 and cluster 9.It seems that marketing team mostly focus on younger groups and who have cars.

For the supervised learning, we have firstly developed preliminary XGBoost model, however the performance metrics were really bad like the following.





It seems that the model tends to say customer will not answer. So we have played around parameters a little bit. The biggest reason might be having unbalanced dataset. So to overcome this, we have calculated scale\_pos\_weight as 80.



Final result is much better. However still we don’t have well performing model. We have discussed what can be done to improve model in the Conclusion section.

# Section 5: Conclusion

My biggest challenge was related about size of the data. I had to restart kernel multiple times. To overcome this problem, unfortunately I had to drop correlated features mostly they were personal data which would help me get interesting clusters. Without them, I came across with similar clusters. The maximum result I could get from clustering was that marketing team mostly focus on younger groups and who have cars. For the supervised learning side there are plenty of room to improve such as trying other classification algorithms like Gradient Boost, Ada boost, SVM, etc. Apart from that, we can try to segment the data and fit supervised algorithm accordingly. This might be another solution to develop specific solutions to specific customer segments.