Capstone Project

Data Science Nanodegree

Juan Camilo
Jiménez Rodríguez

18 January, 2023

Definition

Project Overview

As part of Arvato Financial's business strategy, they want to use Machine Learning techniques that allow them to generate insights from the analysis of information from their clients. It is a real life exercise, where they shared the following data sets:

- 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).

In the same way they shared the metadata, contained in the following files:

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

The project apart from seeking to demonstrate our skills as data scientists, the reality is to carry out an analysis of the data that was shared with us where the clients of the Arvato company are and in the same way a population group from Germany, in such a way that we can Obtain a data set ready to be used in the modeling phase. One of the first modeling tasks consists of comparing these two data sets given a segmentation, which allows me to identify which individuals of the German population best describe the company's customers, likewise, which individuals of the German population do not describe anything to the company's customers.

As a second task within the modeling, it consisted of using the best Supervised Machine Learning technique for the third data set (training) and fourth data set (testing), which contains demographic information for the company's marketing campaign objectives, predicting who of them are more likely to become customers of the company.

Problem Statement

In this project the problem we will be working with is finally:

With the information of the German population and the company's customers, determine how many and which individuals can become new target customers of the company?

Metrics

As there are two modeling tasks, the metrics used in both models are:

- **Segmentation**: No emphasis is placed on the selection of the metric, simply for the reduction of variables using PCA, the minimum is that these new variables explain 90% of the accumulated variance. Subsequently, choose the number of clusters with the elbow method. It would be possible to use the cohesion and separability metrics, and as a metric criterion to use the Silhouette index.

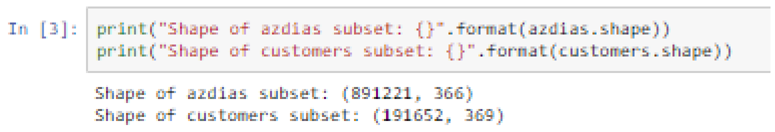
- **Prediction**: Since we already have our clean, pre-processed and scaled data, we must now carry out the Modeling phase. But before that, since we must choose the best model that is capable of predicting whether an individual will become a customer or not, it is necessary to evaluate it and based on this, the one with the best evaluation would be selected. And how do we evaluate it? Machine learning models have evaluation metrics, in the specific case of classification models we find metrics such as accuracy, precision and recall. However, if we were predicting, for example, a patient whether or not he has cancer, precision would be selected because where it is serious where a client who had cancer but the model says no, the impact is serious by not treating someone who did need it. that is, the True Positives (TP). So Precision TP/(TP+FP) would be helping me determine how accurate our results are in predicting whether or not a client will have cancer. For this case, I have decided to use the AUC-ROC curves since they consider the true positive rate (TPR) and the false positive rate (FPR). As such, there is no minimum to determine from more than one model, which is the best, but simply the one with the highest score, although there are those who recommend a minimum of 0.8, but in my case, I will be interested in the one with the best score.

The evaluation metric we will use is AUC for the ROC curve. A receiver operating characteristic (ROC) curve is a graphical plot used to show the true positive rate (TPR, known as recall, the proportion of real customers correctly labeled) versus the false positive rate (FPR, proportion of not customers tagged as customers)

Analysis

Data Exploration

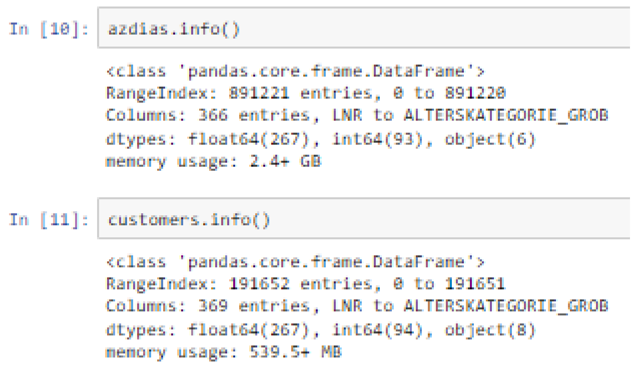
When we load the data, we see a warning message, indicating that there are columns with mixed data types, so it is the first call, to clean the data in later steps. Meanwhile, we host in a first dataframe called AZDIAS and in a second dataframe CUSTOMERS, the first being the group of individuals from Germany and the second the set of company customers. I decide to validate that the data was loaded by applying .head(). In the same way I want to see the size of the data as explained below:



```
In [3]: print("Shape of azdias subset: {}".format(azdias.shape))
        print("Shape of customers subset: {}".format(customers.shape))

        Shape of azdias subset: (891221, 366)
        Shape of customers subset: (191652, 369)
```

Likewise, also see the information of these two data sets:



```
In [10]: azdias.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891221 entries, 0 to 891220
Columns: 366 entries, LNR to ALTERSKATEGORIE_GROB
dtypes: float64(267), int64(93), object(6)
memory usage: 2.4+ GB

In [11]: customers.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 191652 entries, 0 to 191651
Columns: 369 entries, LNR to ALTERSKATEGORIE_GROB
dtypes: float64(267), int64(94), object(8)
memory usage: 539.5+ MB
```

If we compare both data sets, we observe a difference in the number of columns, this, since the customers data set has the following columns: ('CUSTOMER_GROUP', 'ONLINE_PURCHASE', and 'PRODUCT_GROUP')

Additionally, I wanted to determine that the two support files related to the metadata could be useful for me as an understanding of the two data sets, and I discovered that the set that best informs is attributes_values, since it contains the description column of the other metadata file, y contains the possible values that each dataframe would contain and what each unique value means, so with this it becomes the official dictionary.

Data Visualization

As part of the understanding of the data, rather than conducting an exploratory data analysis or EDA, what I decided was to focus on the visualization helping me clarify doubts about problems with the data.
To do this, I first started with the number of missing values per column, with this first graph showing the number of missing values per column in an ordered manner:

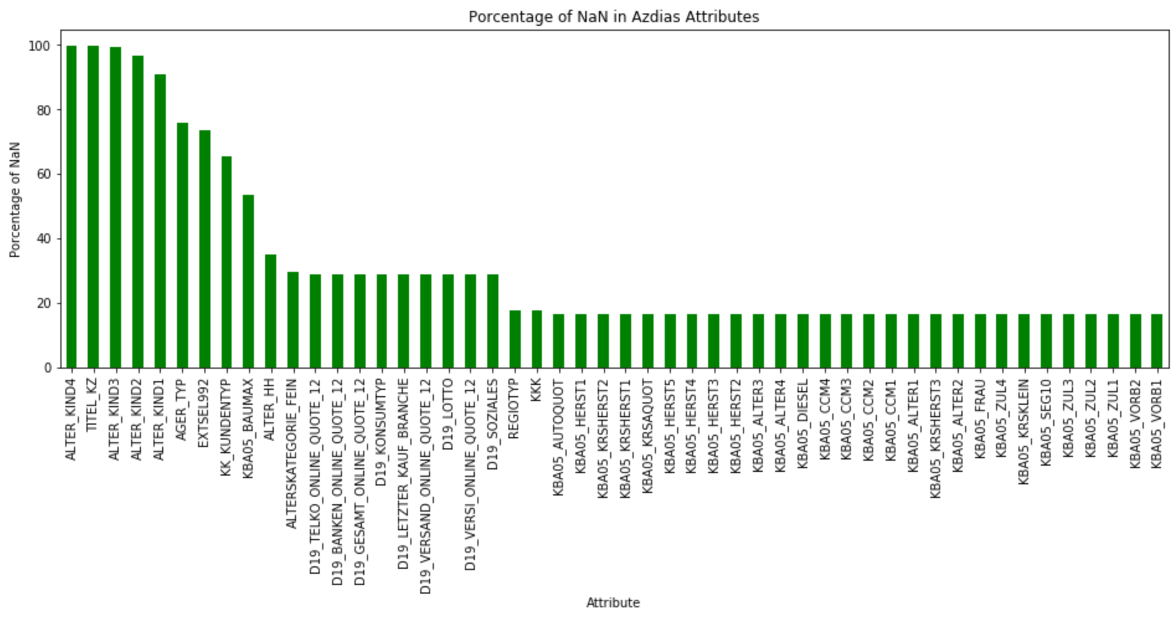


Fig 1. Participation missing values per column for the azdias dataframe.

Similarly, I wanted to know the participation of nulls for each record, as shown in the following image:

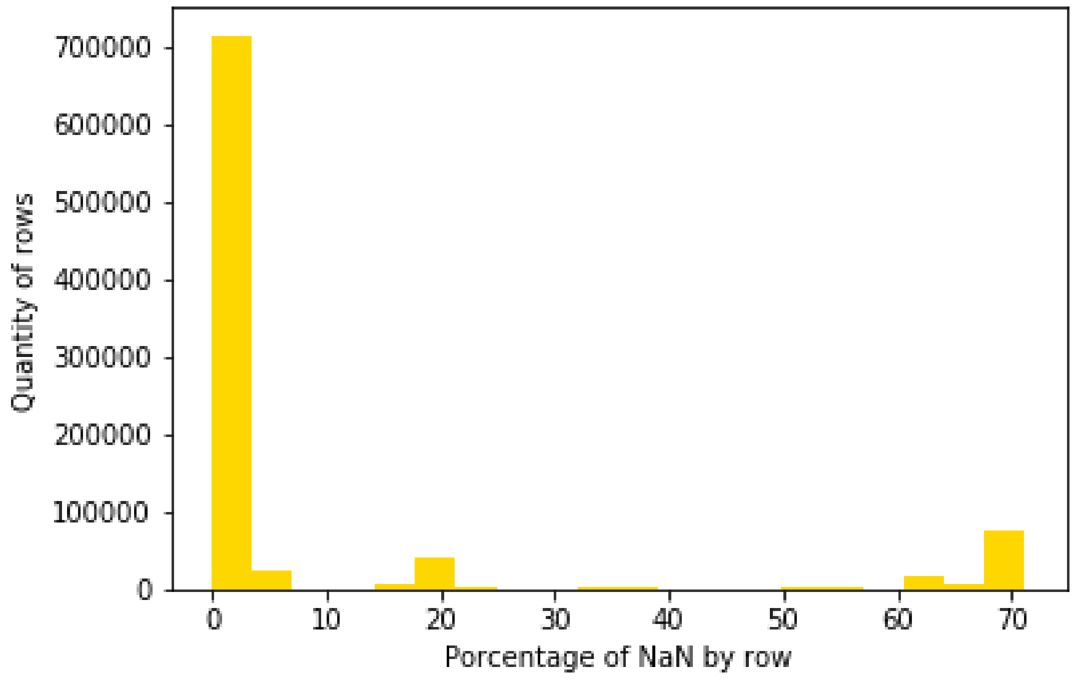


Fig 2. Participation missing values by record for the azdias dataframe.

Methodology

Data Preprocessing

So by analyzing the data, we managed to establish a function that, given a dataframe, generates the clean dataframe. This function includes the following tasks:
- Step 1: Get attribute_values
- Step 2: Replace X and XX in mixed datatype attributes with NAN
- Step 3: Replace Unknown values with NAN
- Step 4: Handle missing values by columns
- Step 5: Handle missing values by rows
- Step 6: Drop attributes are in dataframe but aren't in our dictionary (attribute_values dataframe)
- Step 7: Feature Engineering and Impute NAN
- Step 8: Drop correlated Features
- Step 9: Drop duplicated rows

Once this function is applied, we carry out the scaling process with MinMaxScaler

Implementation

- **ML Unsupervised:**

  I use unsupervised learning techniques to describe the relationship between the demographics of the company's existing customers and the general population of Germany, because the goal is to describe parts of the general population that are more likely to be part of the mail-order company's main customer base, and which parts of the general population are less so.

  Because we have a large number of columns, so that our models are not affected by this aspect, we use dimension reduction using Principal Components Analysis or PCA. In essence, what it does is generate equations where they involve or represent our real variables.To know the number of components, we use the cumulative explained variability

  This code performs dimensionality reduction using the PCA (Principal Component Analysis) method. The objective of this method is to understand the main components that describe the data and eliminate those that do not provide relevant information. The parameter "n_components" indicates the number of main components that you want to keep. If this parameter is None, then the number of principal components that explain at least 95% of the variance in the data will be retained. The random_state parameter is a seed for the random number generator in the PCA algorithm. This allows the results to be reproducible. The "fit" method calculates the principal components from the data. The "transform" method applies the transformation to the data.

If we analyze, for example, the case of the azdias dataframe, we observe the following cumulative variance curve vs. number of principal components:

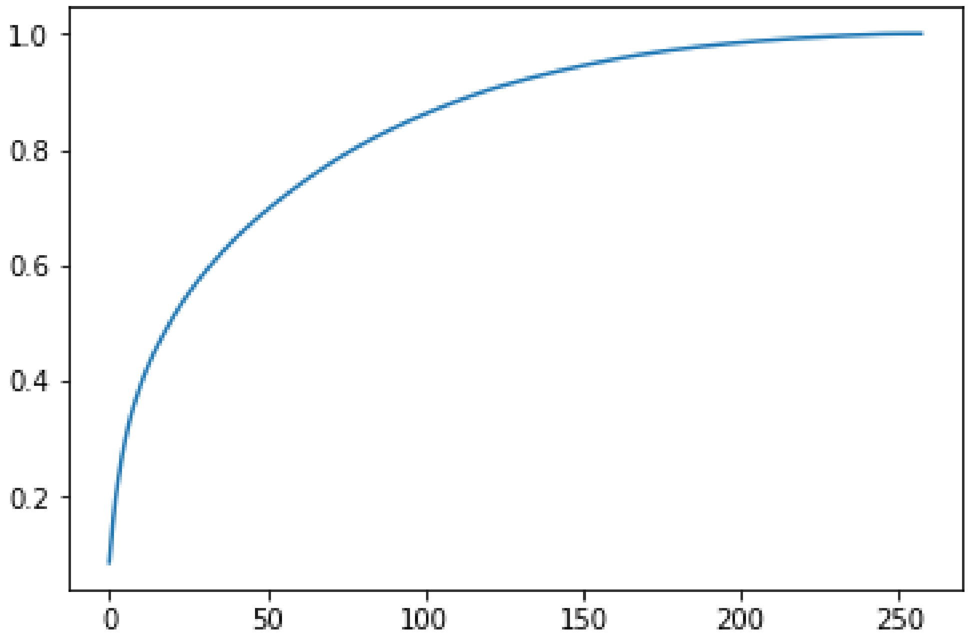
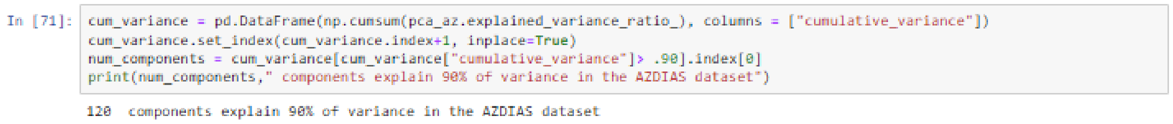


Fig 3. Cumulative variance vs Quantity of principal components - azdias dataframe.

We then determined that the appropriate number of principal components at a minimum was 120 components by explaining 90% of the variance:



```
In [71]: cum_variance = pd.DataFrame(np.cumsum(pca_az.explained_variance_ratio_), columns = ["cumulative_variance"])
         cum_variance.set_index(cum_variance.index+1, inplace=True)
         num_components = cum_variance[cum_variance["cumulative_variance"]> .90].index[0]
         print(num_components," components explain 90% of variance in the AZDIAS dataset")

         120  components explain 90% of variance in the AZDIAS dataset
```

Once we have optimized the number of variables to be used, we determine the number of clusters using the elbow method:

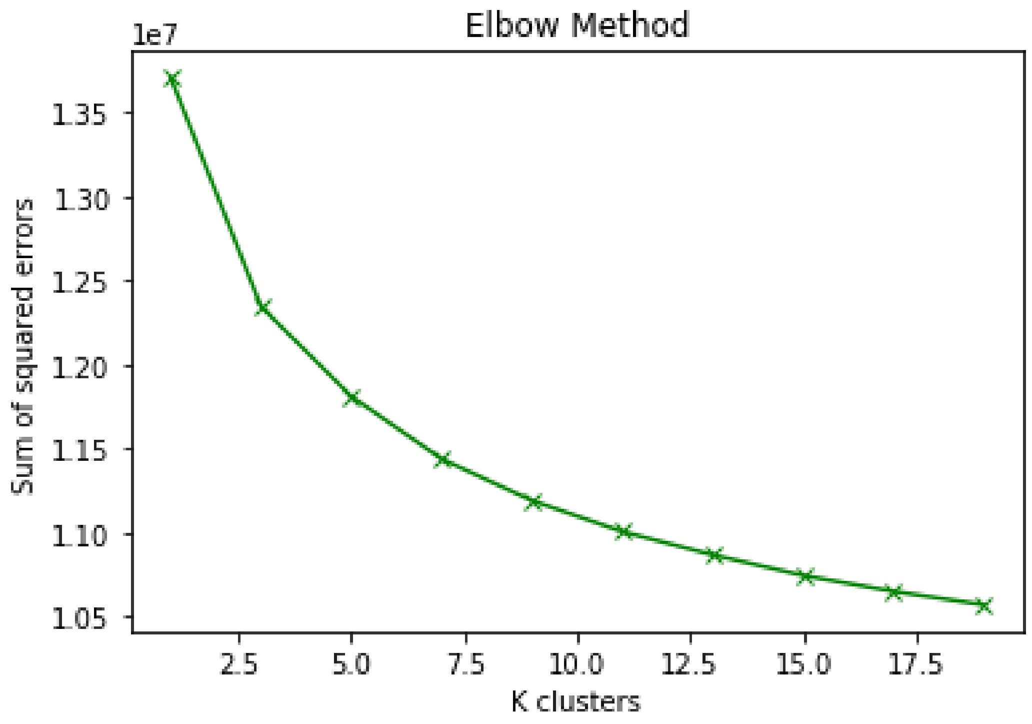


Fig 4. Elbow method - azdias dataframe.

For this case, I could have included in the selection of the optimal cluster the Silhoutte index that combines two important metrics, both cohesion and separability, and allows me to see within a range, if the number of clusters is adequate. For now I will base myself on the previous graph where it is more visual, which is where there is a change in slope in each line (point to point). It is with the above, that given the elbow method, I consider that 8 clusters is the correct amount with these data.

- **ML Supervised:**

I make the decision to balance the data, since there is a large imbalance towards category 0.

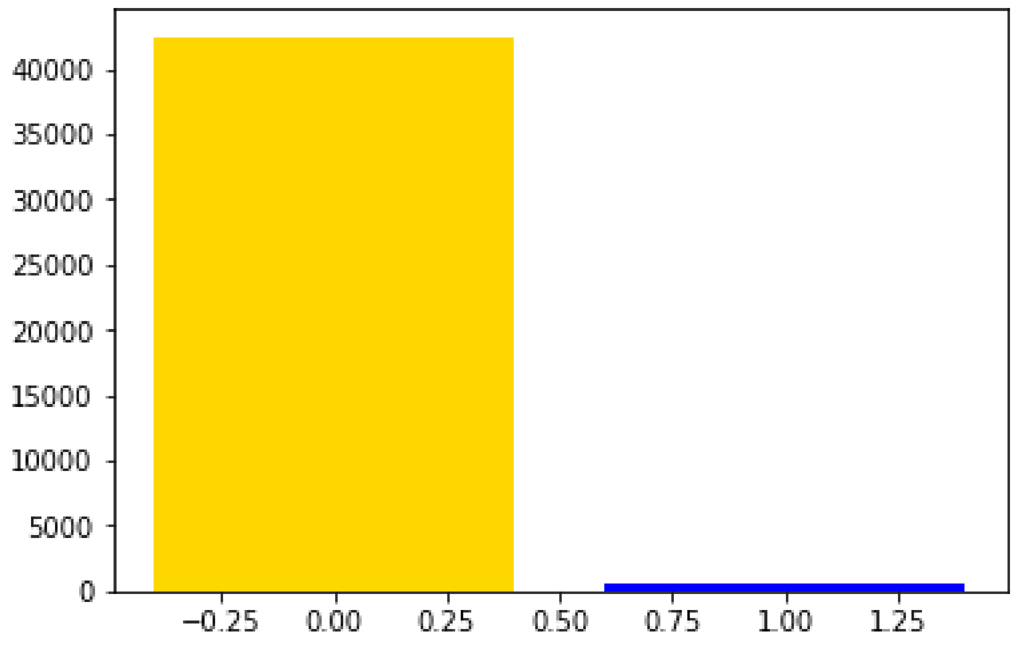


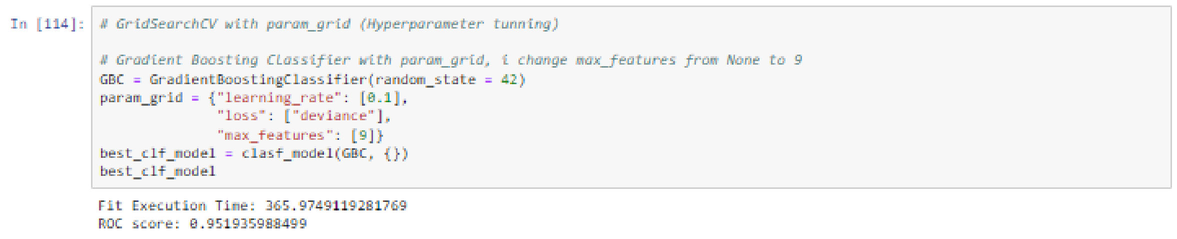
Fig 5. Imbalance RESPONSE column

In order to carry out the modeling phase, I decided to create a function that would only allow me to indicate the name of the classifier, and it would automatically return me: training time, the ROC score and the best parameters in that execution. Use the following classifiers:

- Random Forest Classifier
- Gradient Boosting Classifier
- Ada Boost Classifier
- logistic Regression Classifier

Refinement

Only apply to prediction model. Once we have carried out the training with the four models, I select the Gradient Boosting Classifier because it is the one with the best score among the 4. Since with RandomForestClassifier it obtained 1, that means that it overfitted. As our function to perform the modeling already includes to place the hyperparameters, it is only necessary to call the function and include them. Remember that this clasf_model function cross-validates 5 groups, so we did not do the split that is usually done 70-30 or 80-20. Only change max_features parameter on GradientBoostingClassifier I did not see changes



```
In [114]: # GridSearchCV with param_grid (Hyperparameter tunning)

          # Gradient Boosting Classifier with param_grid, i change max_features from None to 9
          GBC = GradientBoostingClassifier(random_state = 42)
          param_grid = {"learning_rate": [0.1],
                        "loss": ["deviance"],
                        "max_features": [9]}
          best_clf_model = clasf_model(GBC, {})
          best_clf_model

          Fit Execution Time: 365.9749119281769
          ROC score: 0.951935988499
```

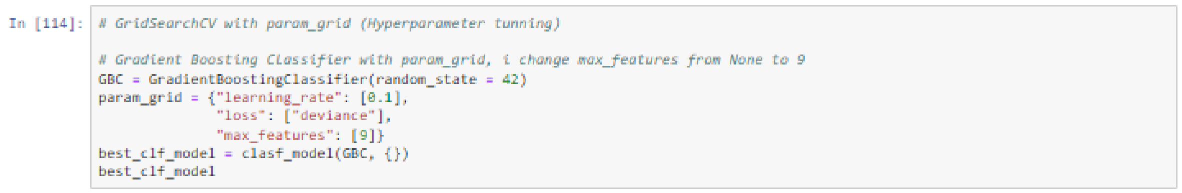
Results

Model Evaluation and Validation

I will talk about the predictive model, where we obtained the following scores for the four models:

Classifier	ROC Score
Random Forest Classifier	1.0
Gradient Boosting Classifier	0.951
Ada Boost Classifier	0.769
logistic Regression Classifier	0.785

As I mentioned before, we had a model with overfitting, so among the 3 it was that I chose Gradient Boosting Classifier. Then, when applying GridSearchCV (before and after tuning we worked with cross validation = 5), these were the parameters:



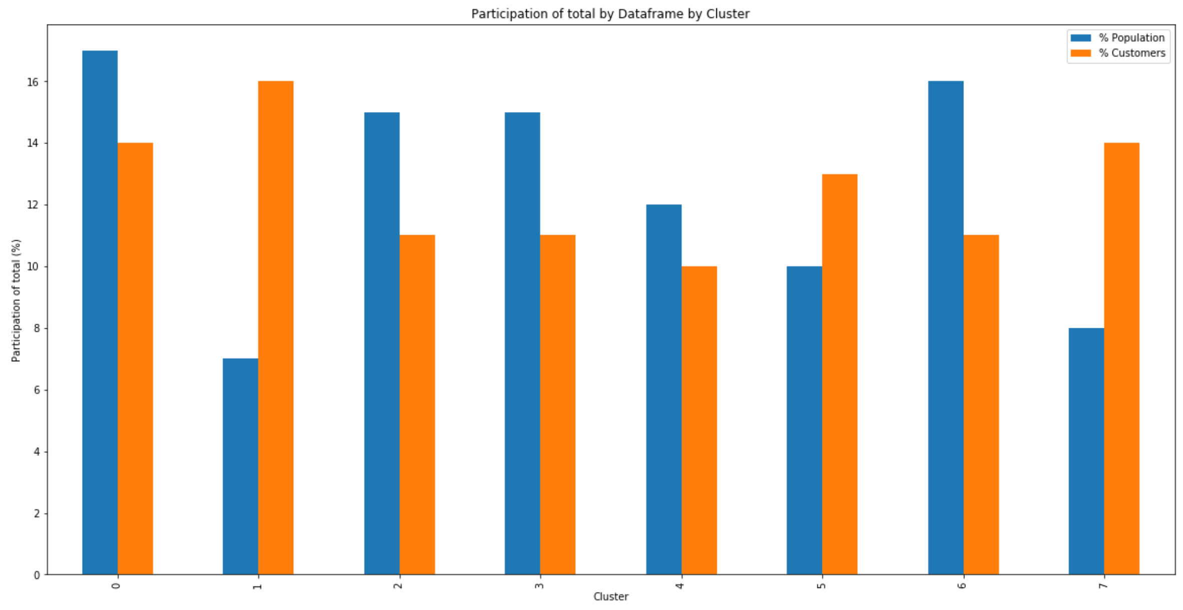
```
In [114]: # GridSearchCV with param_grid (Hyperparameter tunning)

          # Gradient Boosting Classifier with param_grid, i change max_features from None to 9
          GBC = GradientBoostingClassifier(random_state = 42)
          param_grid = {"learning_rate": [0.1],
                        "loss": ["deviance"],
                        "max_features": [9]}
          best_clf_model = clasf_model(GBC, {})
          best_clf_model
```

Where there were no changes in the ROC Score.

Justification

In the case of the segmentation model, by applying k = 8, both in the AZDIAS and CUSTOMERS data sets, and whose objective is to see those customers who could have a certain affinity for being part of the company, I decided to create a graph, where you will place the number of clients for each cluster for each dataframe:



Based on the latest graphs, we can see that with the company's customers where there is more participation, it is the focus on which we must concentrate, since they add up to the first 4 groups 57% that are reflected in the cluster 1 , 0 , 7 and 5 (Ordered). So by being clear about those clusters where the most customers are, it is likewise where we must look for potential customers from the population of Germany. That means, that under the premise of looking for clients from those same clusters, they are summarized in potential clients:

Cluster 0 there are 123.045 people
Cluster 1 there are 54.138 people
Cluster 5 there are 76.101 people
Cluster 7 there are 59.231 people

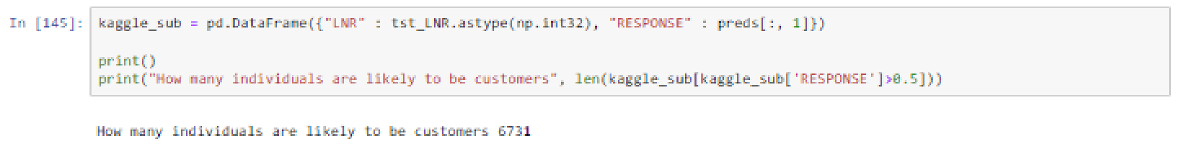
Approximately 312k potential people for the business

In the same way, to avoid reaching the wrong clients with offers or perhaps they were not the first to receive the company's first offer, they are the ones in the clusters with the least participation, which means that transposing it to the population of Germany , these customers who are less likely to be part of the company are:

Cluster 3 there are 111.239 people
Cluster 4 there are 90.326 people

Approximately 200k people who do not represent an opportunity for the business

For the supervised model, when we apply the predictions with the best model including the hyperparameterization, we observe that the number of customers that have a probability of be likely are 6731 people:



```
In [145]: kaggle_sub = pd.DataFrame({"LNR" : tst_LNR.astype(np.int32), "RESPONSE" : preds[:, 1]})

          print()
          print("How many individuals are likely to be customers", len(kaggle_sub[kaggle_sub['RESPONSE']>0.5]))

          How many individuals are likely to be customers 6731
```

Conclusion

Reflection

When navigating through the data, both from the company's customers and from the general population of Germany, it was clear that we were going to encounter a large number of variables, which could impact the performance of our Machine Learning models, and being important to go to the dictionary that explained the definition, and the possible data in them, for a proper understanding of the business and the data. I think it was right to have generated a data cleaning function, since that allowed it to be used in the other datasets. As an appreciation that I commented at the time, I did not want to treat the outliers since I did not consider that they were rare data that were the reason for elimination, but that they are people who in real life have these behaviors, and their elimination could erase knowledge. I am happy to remove columns and records based on the number of NaNs and those not in the dictionary, since we would not know what exactly we have of those unknown columns.

When moving to the unsupervised model, with clean, pre-processed, and scaled data, we began to see the importance of performing PCA since there were many variables, and thus allowed us to general segmentation, with the number of clusters. I found it interesting to be able to discover potential clients for the business, by associating each cluster, and this is how we observe which clusters and therefore which clients would be clients, and who would not.

Lastly, with the prediction model, with the training data, we applied the data cleaning function, the scaling function, as I stated at the time, it was the test decision to balance the data so that there was no bias in the model. Additionally, I wanted to test mostly computationally robust models, so that it would allow me to generate the best model for our problem. Once the best model has been learned, and with the hyperparameterization, we already generated the prediction, where it indicates that there are 6,731 potential individuals to be clients of the company.

Note: Within the structure of the Notebook, the directory or location to make the prediction to the TEST set was not suggested, so understanding that in part 0 they said about this activity especially to be compared to kaggle, it is that I carried out that prediction stage.

Improvement

- I believe that the model could be improved if we applied an analysis to the outliers, given that even though they are clients who really are there and it is not unusual behavior, the model may be impacted by this scenario.
- I would like to have achieved a better understanding of the columns, talking with the stakeholders, in this case members of the company, for a better understanding of the business, and of the data, since I believe there are possibilities of carrying out a stage that I did not want to carry out. and it is featuring engineering
- Regarding scaling, you could change MinMaxScaler to StandardScaler, and observe the changes in the model.
- Unfortunately I couldn't use XGBoosting, since at the time of writing the code, it was not installed, and its installation was not possible. With which I would have liked to use it, and determine if it would have been the model selected to make the predictions.