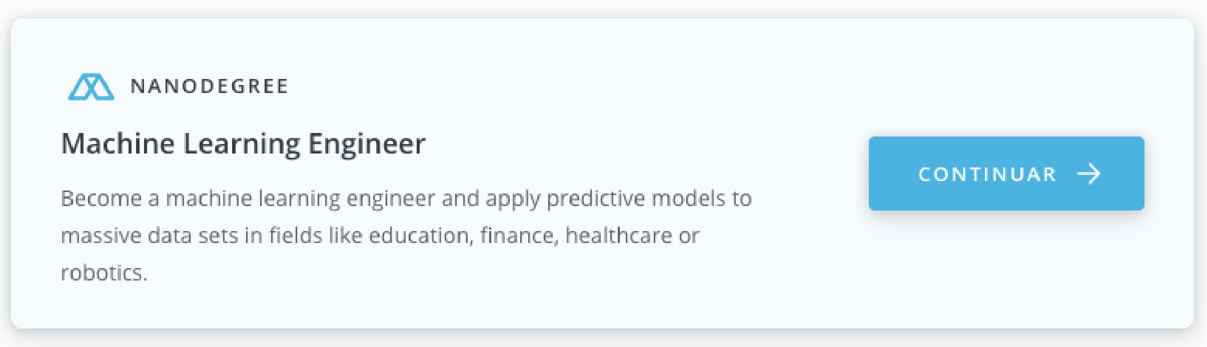
Machine Learning Engineer Nanodegree



NANODEGREE

Machine Learning Engineer

Become a machine learning engineer and apply predictive models to massive data sets in fields like education, finance, healthcare or robotics.

Capstone Project Report

Vagner A Silva

September 16st, 2020

Customer Segmentation – Arvato Financial Solutions

1. Project Overview

This project was one propose capstone project using Kaggle competition for the Udacity Machine Learning Nanodegree. The goal of the project is to determine how one of Arvato's clients can acquire new customers for their mail-order organic products.
To help us with this problem statement Arvato provides data on general population demographics, on their customers, and on client response to the previous campaign. This data is protected under terms and conditions and not shareable. The objective is to use these datasets, we are proposed to predict which characteristics from individuals from the general population can be used to selectively target as good responders to this marketing campaign.

The project has two principal parts :

- Unsupervised Learning to identify segments of the German population that match the existing customer segments;

- Supervised Learning to identify the likelihood of customer conversion from the general population

The results of this analysis and submission of the Kaggle competition present in the final part of the project.

2.Domain Background

Bertelsmann found its origins as a publishing house in 1835 (Schuler, 2010), and through steady growth and development made its way to the software and hardware distribution market in the ’80s (Computerwoche, 1983). By 1999 the company received its current name Arvato Bertelsmann (Name, 1999) and over the next decade fully entered the domain of high-tech, information technology, and e-commerce services (Paperlein, 2012).
Arvato offers financial solutions in the form of diverse segments, from payment processing to risk management activities. It is in this domain that this capstone project will be developed. Arvato is looking to use its available datasets to support a client (mail-order company selling organic products) in identifying the best data founded a way to acquire a new client base. This project explores Arvato’s existing datasets to identify attributes and demographic features that can help segment customers of interest for this particular client.

The segment costumers help a to market is a growing field that benefits greatly from accurate segmentation, with the help of machine learning hidden patterns can be found in volumes that could easily be missed without computational help, requiring very little maintenance or human intervention, leading to an improved experience from customer seekers and customers alike.

3.Problem Statement

To resolve this problem is necessary for the first to see the customer's dataset using unsupervised learning algorithms.
Whit this they propose is to identify segments in general in the existing customers
The next step is to apply one supervised learning algorithm will be used to make predictions on whether a person is a probable customer or not, based on the demographic data.

4.Datasets and Inputs

All the datasets were provided by Arvato in the context of the Udacity Machine Learning Engineer Nanodegree.

There are 4 datasets to be explored in 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).

And 2 metadata files associated with these datasets:

- DIAS Information Levels — Attributes 2017.xlsx: a 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.

Which can help to map the attributes to datasets.

5. Evaluation Metrics

This problem is a multi-class classification problem, and one of the most valuable metrics to measure model performance is the Area Under the Curve Receiver Operating Characteristics (ROC-AUC). The curve represents a degree or measure of separability and, the higher the score the better the model is performing.
A great advantage of using RIOC-AUC is the immunity to class imbalance, which is the case for this problem. The number of people that are positive responders to an ad campaign is on average far lesser than those that respond negatively.

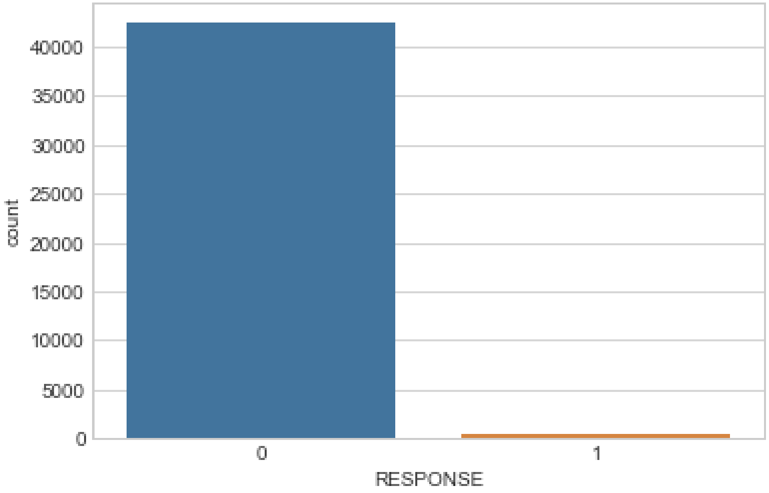


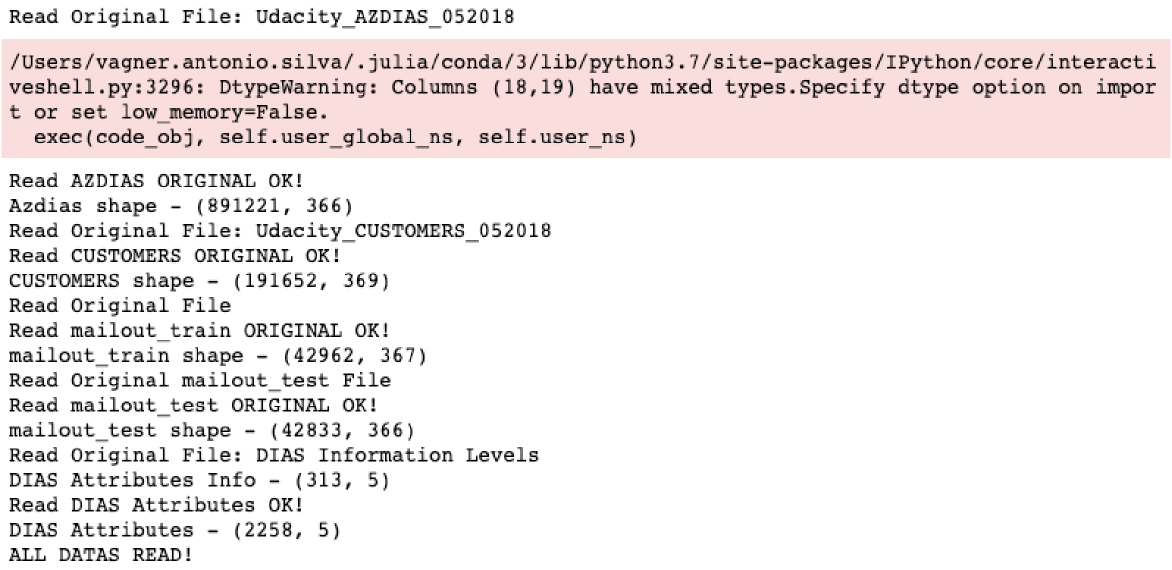
Fig1: dataset Udacity_MAILOUT_052018_TRAIN.csv with responses.

6. Preprocessing

To preprocessing data and prepare to apply models, we need to many steps the first is load data and exploring this.

6.1 Loading dataset

When you read the data receive one warning to differents types in two columns, in the next steps, the warnings are to trated.



```
Read Original File: Udacity_AZDIAS_052018
```

```
/Users/vagner.antonio.silva/.julia/conda/3/lib/python3.7/site-packages/IPython/core/interacti
veshell.py:3296: DtypeWarning: Columns (18,19) have mixed types.Specify dtype option on impor
t or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
Read AZDIAS ORIGINAL OK!
Azdias shape - (891221, 366)
Read Original File: Udacity_CUSTOMERS_052018
Read CUSTOMERS ORIGINAL OK!
CUSTOMERS shape - (191652, 369)
Read Original File
Read mailout_train ORIGINAL OK!
mailout_train shape - (42962, 367)
Read Original mailout_test File
Read mailout_test ORIGINAL OK!
mailout_test shape - (42833, 366)
Read Original File: DIAS Information Levels
DIAS Attributes Info - (313, 5)
Read DIAS Attributes OK!
DIAS Attributes - (2258, 5)
ALL DATAS READ!
```

Fig2: warning and load datasets.

6.2 Exploratory dataset

As we can see there are a lot of nulls, at first sight, let's use the missing library, very useful for this kind of task.

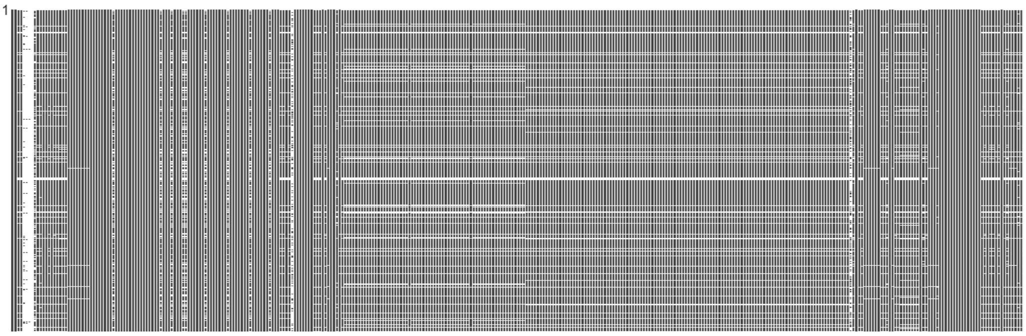
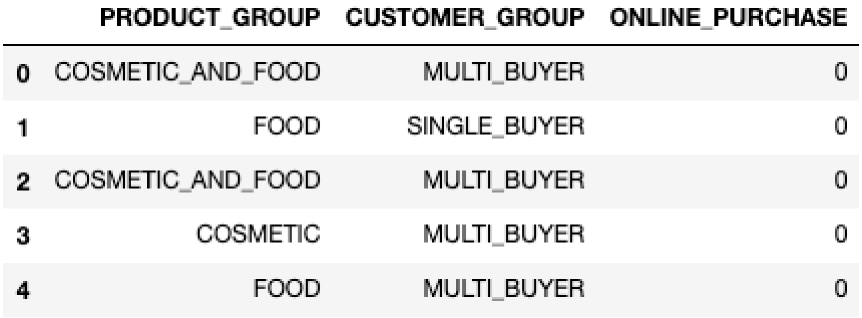


Fig 3: missing values in azdias Dataset.

The dataset Azdias and Costumers have differences in the number of columns



	PRODUCT_GROUP	CUSTOMER_GROUP	ONLINE_PURCHASE
0	COSMETIC_AND_FOOD	MULTI_BUYER	0
1	FOOD	SINGLE_BUYER	0
2	COSMETIC_AND_FOOD	MULTI_BUYER	0
3	COSMETIC	MULTI_BUYER	0
4	FOOD	MULTI_BUYER	0

Fig 4: Columns extra betwen two dataframes.

Interestingly, it appears that the distributions are almost similar. It is possible to observe that the customers in these data have greater activities before 1995, according to the graph.

The EINGEFUEGT_AM column appears the date the entry was made

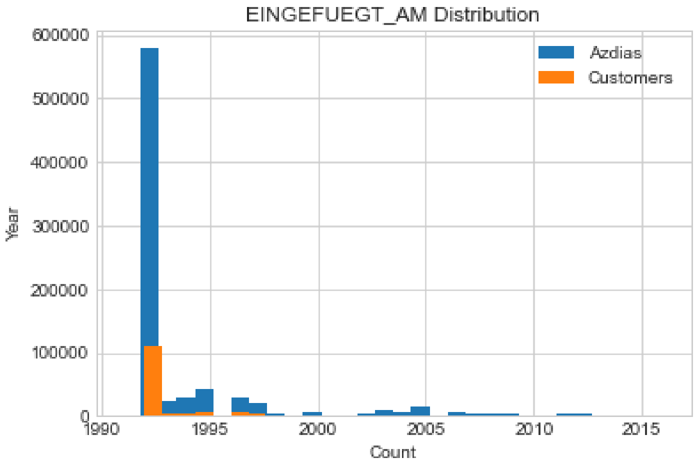
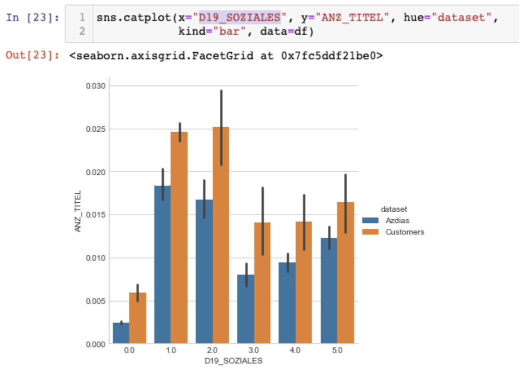
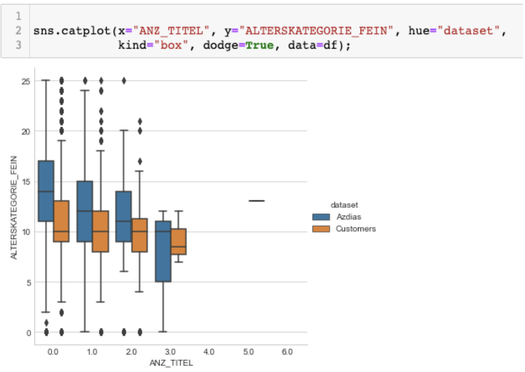


Fig5 : EINGEFUEGT_AM datetime distribuition

when you see in two categories is possible to observe the distribution the dataset



```
In [23]: sns.catplot(x="D19_SOZIALES", y="ANZ_TITEL", hue="dataset",
                     kind="bar", data=df)
Out[23]: <seaborn.axisgrid.FacetGrid at 0x7fc5ddf21be0>
```



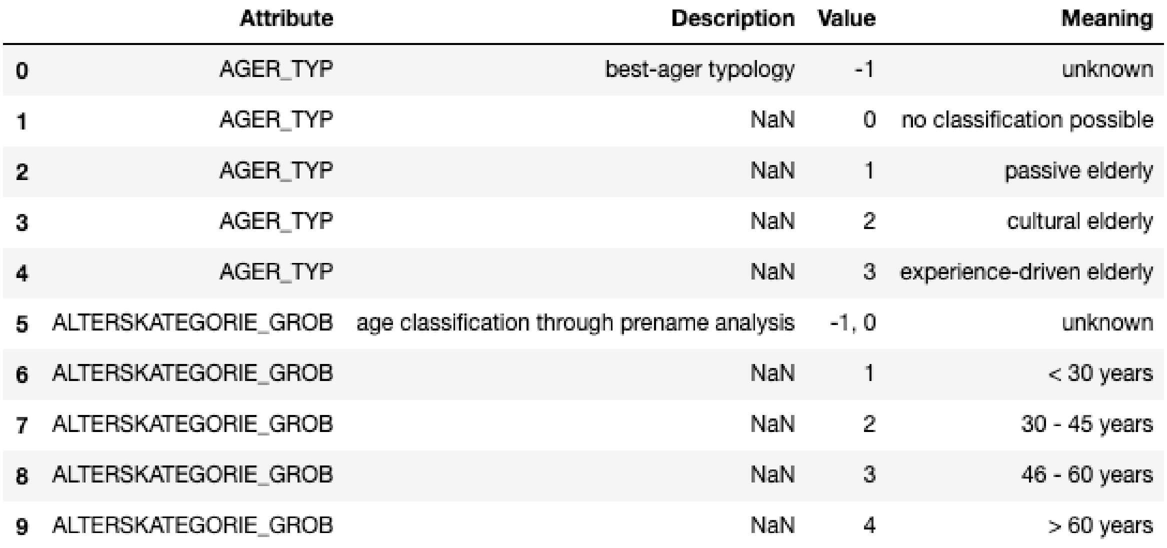
```

sns.catplot(x="ANZ_TITEL", y="ALTERSKATEGORIE_FEIN", hue="dataset",
            kind="box", dodge=True, data=df);
```

Fig6: Distribution the dataset

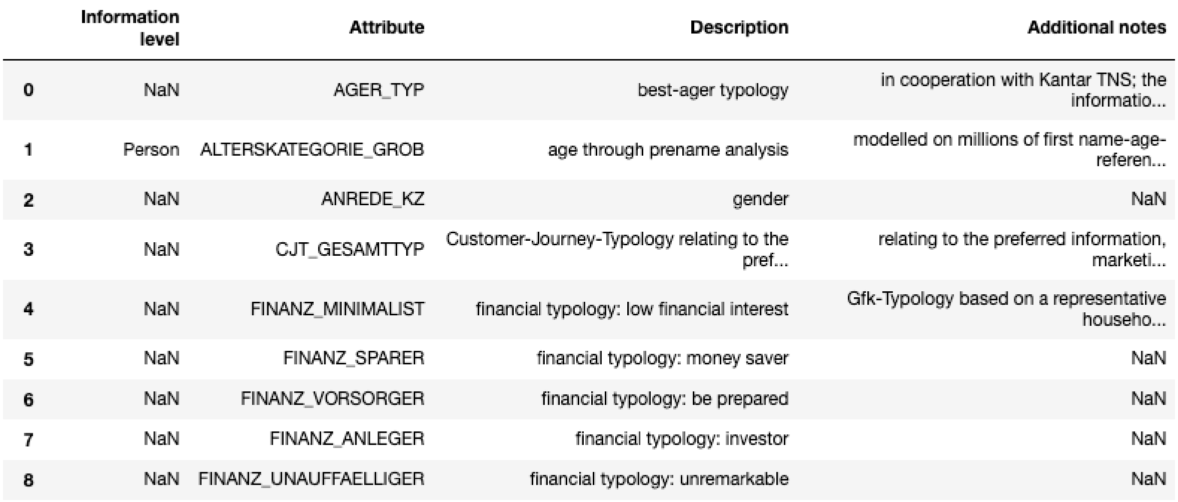
6.3 Reading features and atributes

Let's explore what each column represents and understand a little more about the data.

For this, we will use the excel files to complement the information about the project, as it contains all the column names and their descriptions with the corresponding values. This information is essential to the next step when nan values and others analysis is necessary to clean dataset



	Attribute	Description	Value	Meaning
0	AGER_TYP	best-ager typology	-1	unknown
1	AGER_TYP	NaN	0	no classification possible
2	AGER_TYP	NaN	1	passive elderly
3	AGER_TYP	NaN	2	cultural elderly
4	AGER_TYP	NaN	3	experience-driven elderly
5	ALTERSKATEGORIE_GROB	age classification through prename analysis	-1, 0	unknown
6	ALTERSKATEGORIE_GROB	NaN	1	< 30 years
7	ALTERSKATEGORIE_GROB	NaN	2	30 - 45 years
8	ALTERSKATEGORIE_GROB	NaN	3	46 - 60 years
9	ALTERSKATEGORIE_GROB	NaN	4	> 60 years



	Information level	Attribute	Description	Additional notes
0	NaN	AGER_TYP	best-ager typology	in cooperation with Kantar TNS; the informatio...
1	Person	ALTERSKATEGORIE_GROB	age through prename analysis	modelled on millions of first name-age-referen...
2	NaN	ANREDE_KZ	gender	NaN
3	NaN	CJT_GESAMTTYP	Customer-Journey-Typology relating to the pref...	relating to the preferred information, marketi...
4	NaN	FINANZ_MINIMALIST	financial typology: low financial interest	Gfk-Typology based on a representative househo...
5	NaN	FINANZ_SPARER	financial typology: money saver	NaN
6	NaN	FINANZ_VORSORGER	financial typology: be prepared	NaN
7	NaN	FINANZ_ANLEGER	financial typology: investor	NaN
8	NaN	FINANZ_UNAUFFAELLIGER	financial typology: unremarkable	NaN

Before to start cleaned and another analysis is very important to extract information about the dataset:

- We have 42 exclusive columns provided in the attribute_values that are not present in the customer's data or data.

- We have 3 specific values in the costumers that have already been mentioned in the analysis in the previous steps.

- we have 272 attributes commons between two datasets.

6.4 Data cleaning and resource engineering

All values if no have information in attributes have a categorical type, unknown value, and another inconsistency we change to nan and in the next step cleaned then. To help this work is essential the information extract to attributes and features mentioned before.

The principal columns in the list :
LP_FAMILIE_ *,
LP_FAMILIE_ *,
LP_LEBENSPHASE_ *,
LP_STATUS_GROB
LP_LEBENSPHASE_GROB
LP_LEBENSPHASE_FEIN
LP_STATUS_FEIN
LP_FAMILIE_FEIN,
LP_FAMILIE_GROB,
LP_LEBENSPHASE_GROB
LP_FAMILIE_FEIN
LP_STATUS_FEIN
LP_LEBENSPHASE_GROB
LP_LEBENSPHASE_FEIN
LP_LEBENSPHASE_GRO
LP_LEBENSPHASE_FEIN
LP_LEBENSPHASE_FEIN
AGER_TYP
CAMEO_DEU_2015
D19_LETZTER_KAUF_BRANCHE
OST_WEST_KZ
PRODUCT_GROUP
CUSTOMER_GROUP
CAMEO_INTL_2015
CAMEO_INTL_2015_WEALTH
CAMEO_INTL_2015_FAMILY
WOHNLAGE

the objective is no have more anyone value stay categorical or different the null value, in this case, the process is applied to the Azdias dataset and Costumers dataset.

6.5 Missing values

Now that we've replaced all the unknowns with np.nans, we can see how many missing values each column contains and decide whether to keep a column for later analysis, to start that we initiate with columns and the next to rows.

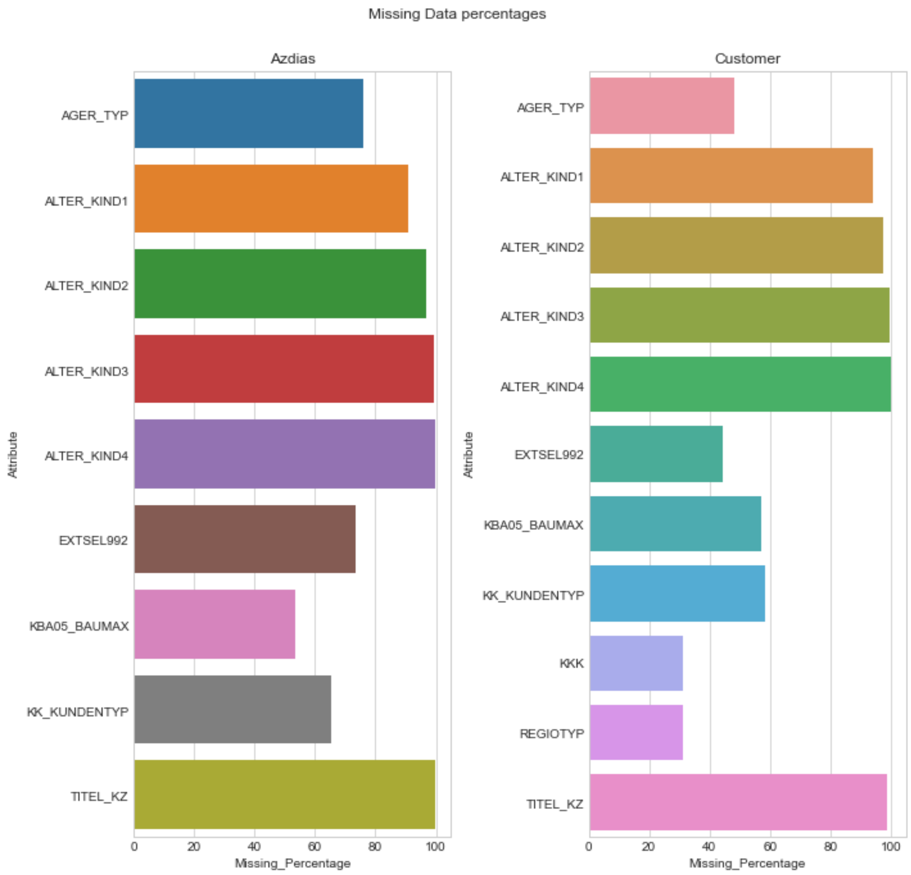


Fig 7 : Percentage missing values

There are 11 resources with more than 30% of missing values in the customer data, while in Azdias data we have only 9 resources.

Considering that, in total, we had 279 columns with missing values in both dataframes.

We will remove the resources with more than 30% of the data from the customer data, also the same resources have to be removed from the Azdias data.

After removing columns that have more than 30% of missing values, we can now examine dataframes with remaining resources for any missing values in the rows

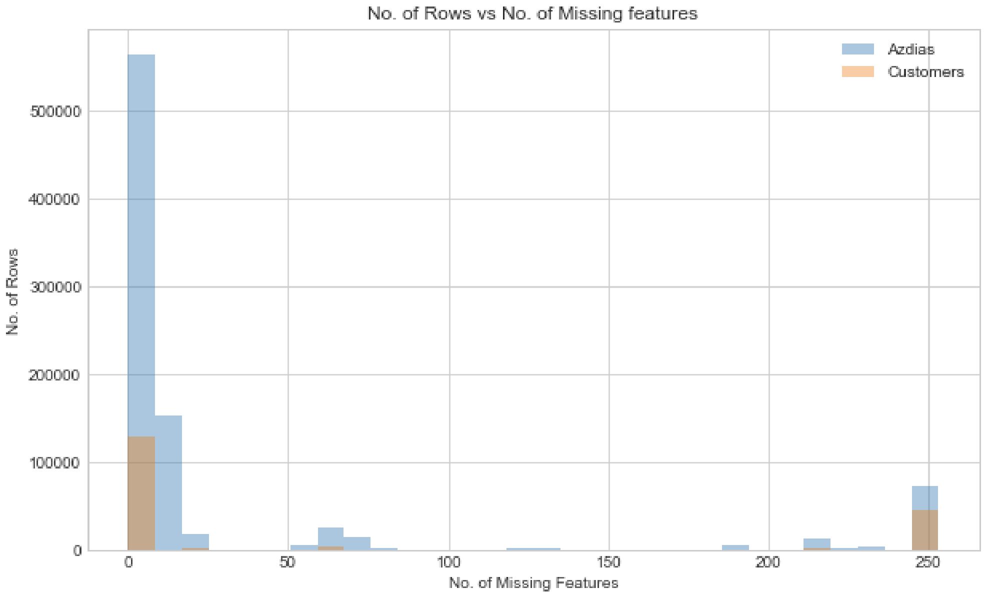


Fig 8: Missing values in rows

We have 250/355 missing resources on approximately 70,000 lines on Azdias and approximately 50,000 lines on customer data.

Most rows have less than 50 missing values in both dataframes. Customer data has comparatively more rows with errors than Azdias.

Observing this information and it is prudent to discard the lines with more than 50 missing values

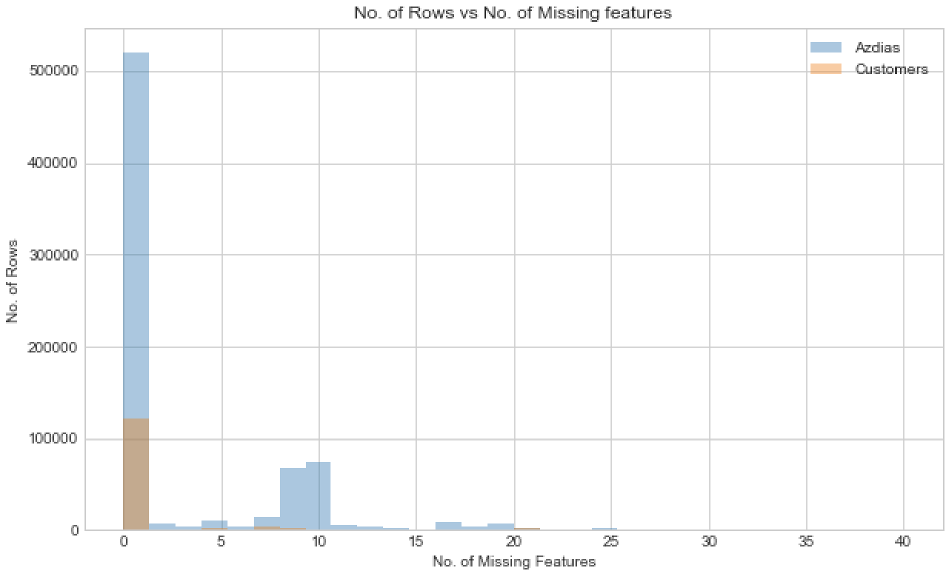
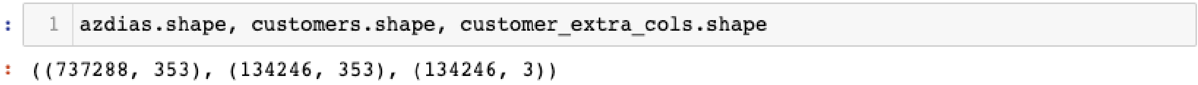


Fig 9: Dataset cleaned values.

The graph above shows the result of the removed lines but can still be observed for some missing values in the range 1 to 50.



```
:  azdias.shape, customers.shape, customer_extra_cols.shape
: ((737288, 353), (134246, 353), (134246, 3))
```

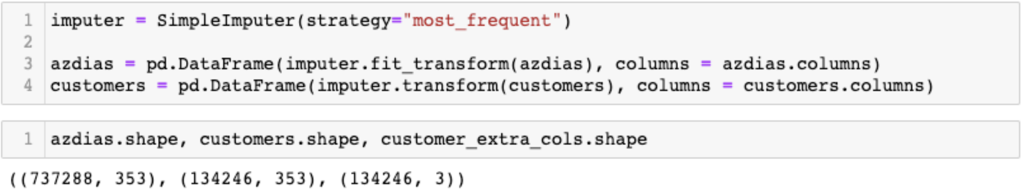
Finally checking the shapes of all the dataframes.

6.6 Imputing missing values

We can input the most common values of the corresponding resources in these lines since the data represent the population demographics. These values considering the average can help in the realization and complement of the model.

Although we have eliminated columns and rows with missing values based on some limit. We still have some columns with missing values. We can now resolve these missing values by filling them with the average of the values or with the most common values.

In this case, imputing missing values with the most common values will make sense, as it is demographic data and the most common values represent the population.



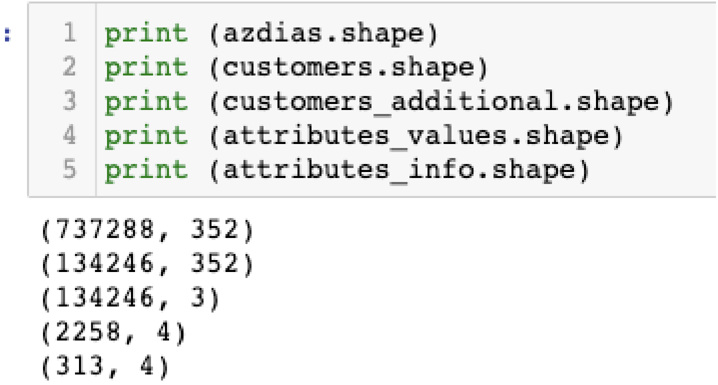
```
imputer = SimpleImputer(strategy="most_frequent")

azdias = pd.DataFrame(imputer.fit_transform(azdias), columns = azdias.columns)
customers = pd.DataFrame(imputer.transform(customers), columns = customers.columns)
```

```
azdias.shape, customers.shape, customer_extra_cols.shape
```

```
((737288, 353), (134246, 353), (134246, 3))
```

6.7 Featuring scaling

Finally, after clearing the data, we can now scale the data to ensure that all resources have the same range. We will use sklearn's StandardScaler to scale the data. However one last column need to drop the "LNR", they mean about this column represent a unique value to each row



```
print (azdias.shape)
print (customers.shape)
print (customers_additional.shape)
print (attributes_values.shape)
print (attributes_info.shape)
```

```
(737288, 352)
(134246, 352)
(134246, 3)
(2258, 4)
(313, 4)
```

7. Dimensionality Reduction PCA

As the number of resources in the data is relatively high, we can see the variation explained by each resource in the data set. Using a dimensionality reduction technique, we can effectively reduce the number of features that do not vary much in the data. Since we cannot analyze each resource on its own to decide if the resource is varying, we can use a statistical approach to find out how much variation is explained by each resource. One of these algorithms is Principal Component Analysis (PCA).

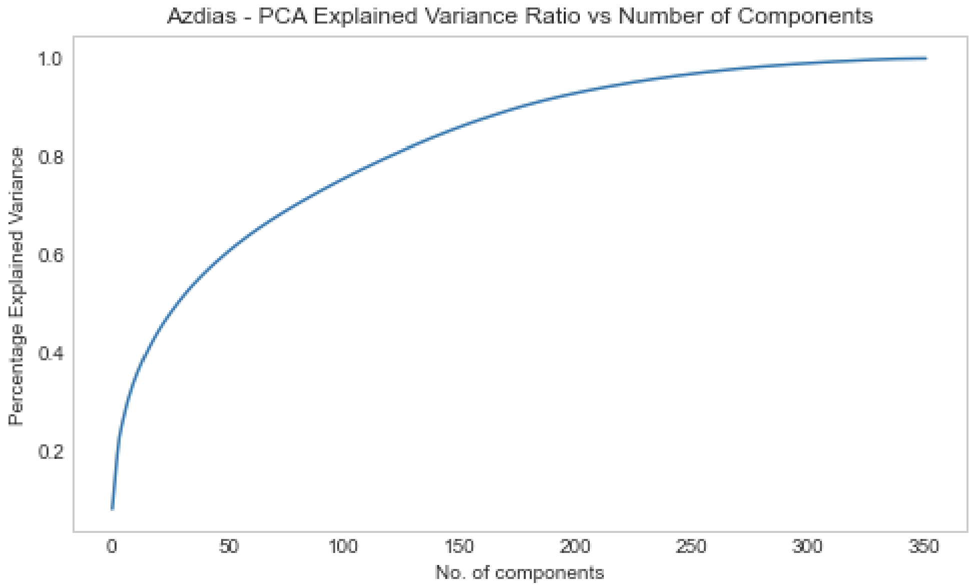


Figura 9: PCA Number of components

From the explained variance graph, it is observed that about 150 components explain 90% of the variance of the data set. We can set the number of components to 150 and perform the PCA analysis to have 150 components. Then, we can see the importance of resources for each component, to understand what each component of the PCA represents.

8. Algorithms and Techniques

After reducing the number of dimensions, we will now use the K-Means Clustering algorithm to group the general population into different segments.

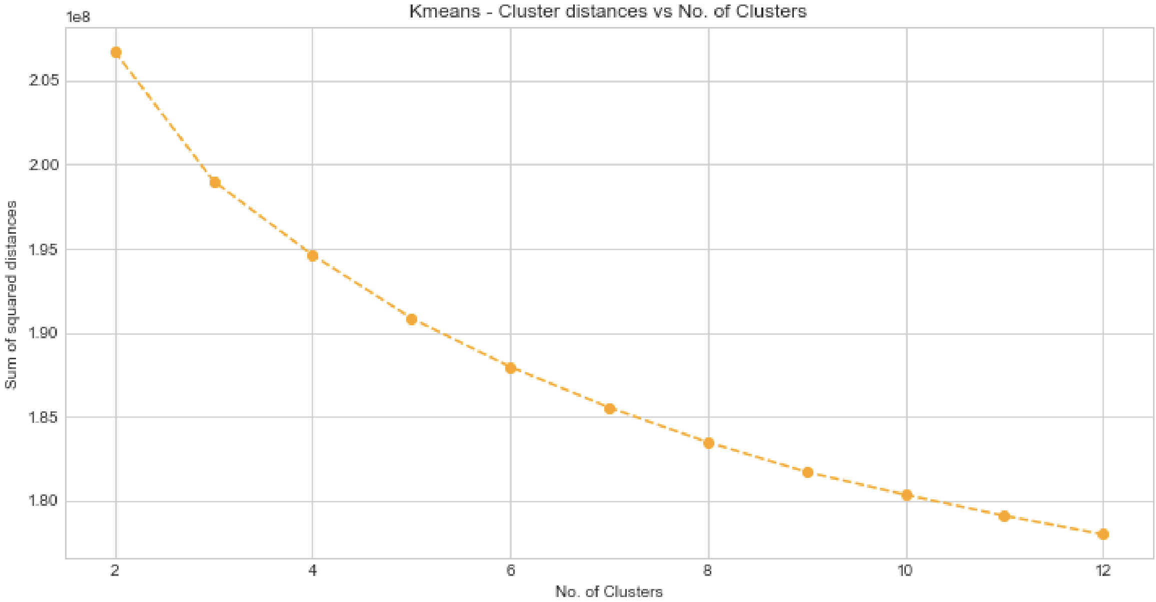
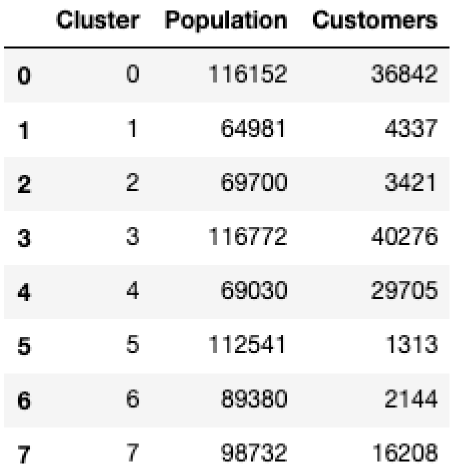


Fig 10: Elbow Graph

The basic idea behind clustering algorithms is to select the number of clusters in order to minimize intracluster variation.

In this process, the elbow method is chosen to select the ideal number of clusters.

From the elbow above, we can see that the sum of the quadratic error decreases with a high slope up to about 8 clusters, and then the slope decreases.



	Cluster	Population	Customers
0	0	116152	36842
1	1	64981	4337
2	2	69700	3421
3	3	116772	40276
4	4	69030	29705
5	5	112541	1313
6	6	89380	2144
7	7	98732	16208

The distribution of the general population is almost uniform (although not perfectly uniform). Clients are mainly from clusters 0, 3, 4, 7.

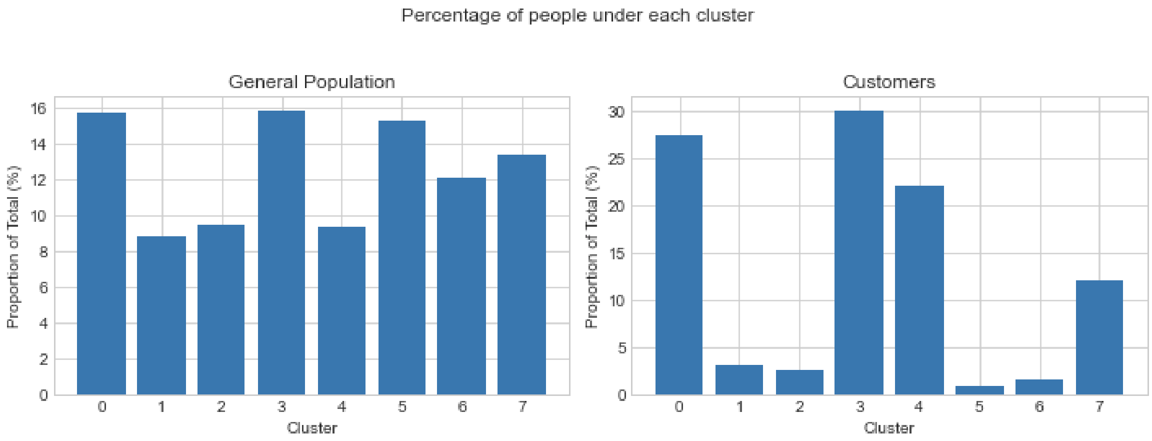
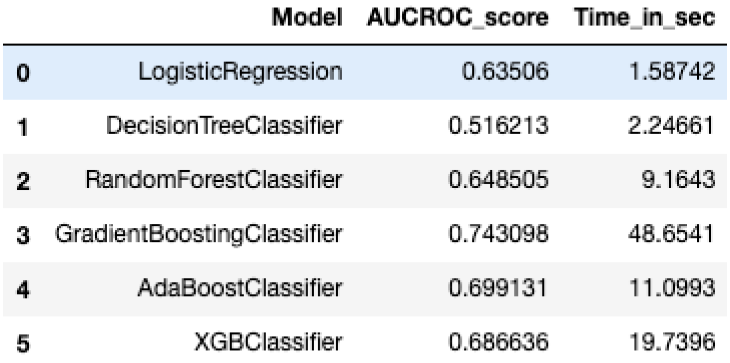


Fig11: People each clusters

9. Supervised Learning Model

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.



	Model	AUCROC_score	Time_in_sec
0	LogisticRegression	0.63506	1.58742
1	DecisionTreeClassifier	0.516213	2.24661
2	RandomForestClassifier	0.648505	9.1643
3	GradientBoostingClassifier	0.743098	48.6541
4	AdaBoostClassifier	0.699131	11.0993
5	XGBClassifier	0.686636	19.7396

In the proposal we suggest one benchmark to see the evolution and determine the best model.
The list show for models and the Logistic Regression with the base to comparate.

10. Results and conclusion

The tuned Gradient Boosting model is finally applied to the provided test data. The results are submitted to the Kaggle competition, where the final roc AUC score is 0.7477. Cross-Validation and Grid Search helped to improve the model quality. However, with better domain knowledge and feature engineering, higher results can be achieved.

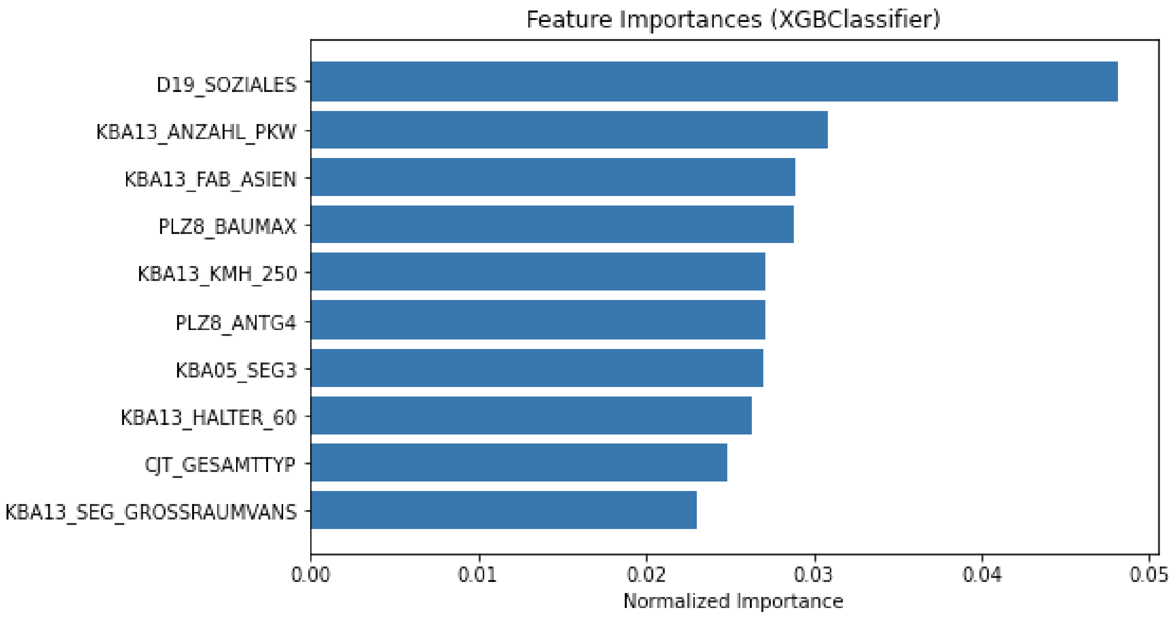


Fig12: Features the model XGBoost.

The feature D19_SOZIALES have importance in the two models Adaboost and XGBoost , but in the second the distribution follow the others features too, and the AUC values show for us the best model is XGBoost

Adaboost: 0.7430
XGBoost: 0.7477

11. LINKS

[1] [https://www.kaggle.com/c/udacity-arvato-identify-customers/overview/description](https://www.kaggle.com/c/udacity-arvato-identify-customers/overview/description]

[2] [https://machinelearningmastery.com/assessing-comparing-classifier-performance-roc-curves-2/]