# *Capstone Project*

*Machine Learning Engineer Nanodegree*

*Keyvan Tajbakhsh*
*Aug 12th, 2019*

# *Customer Segmentation Report for Arvato Financial Services*

## *I. Definition*

### *Project Overview*

Nowaday, all business organizations are adopting datadriven strategies to generate more profits out of their business. Growing startups are investing a lot of funds in data structures to maximize profits of the business group by developing intelligent tools backed by machine learning and artificial intelligence.

Customer segmentation allows a business to precisely reach a consumer with specific needs and wants. In the long term, this benefits the company, because they are able to use their corporate resources more effectively and make better strategic marketing decisions.

In other words this is the practice of dividing a customer base into groups of individuals that are similar in specific ways relevant to marketing, such as age, gender, interests and spending habits.

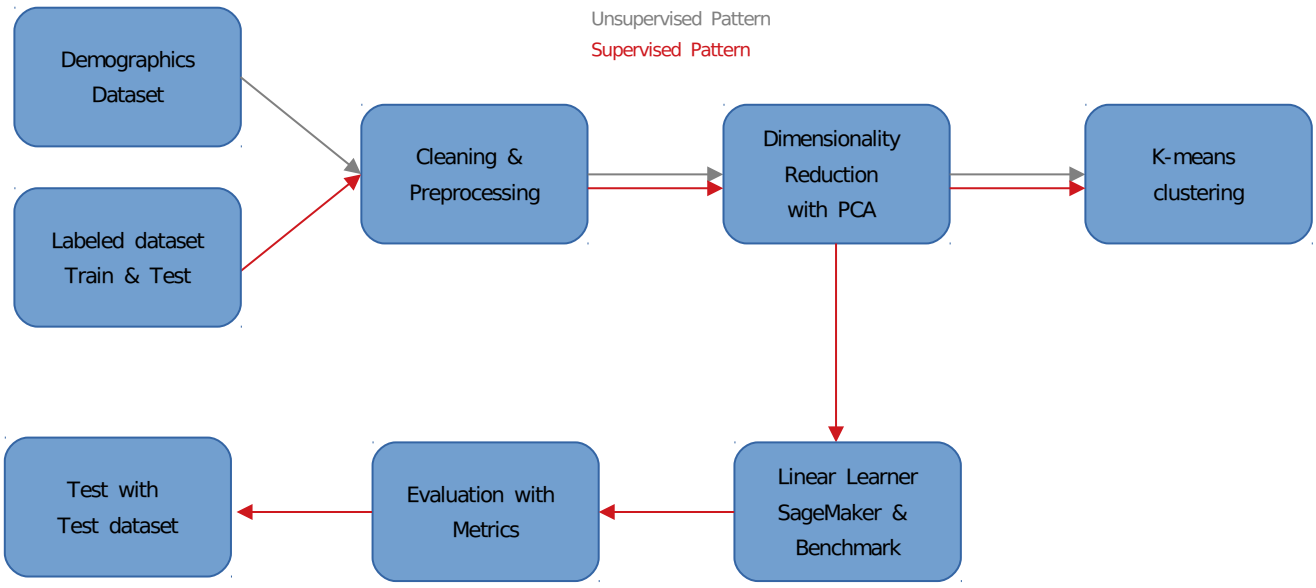
Companies employing customer segmentation operate under the fact that every customer is different and that their marketing efforts would be better served if they target specific, smaller groups with messages that those consumers would find relevant and lead them to buy something. Companies also hope to gain a deeper understanding of their customers' preferences and needs with the idea of discovering what each segment finds most valuable to more accurately tailor marketing materials toward that segment.

This project relies on identifying key differentiators that divide customers into groups that can be targeted. Information such as a customers demographics (age, race, religion, gender, family size, ethnicity, income, education level), geography (where they live and work), psychographic (social class, lifestyle and personality characteristics) and behavioral (spending, consumption, usage and desired benefits) tendencies are taken into account when determining customer segmentation practices.

For this purpose we will use unsupervised learning techniques to describe the relationship between the demographics of the company's existing customers and the general geographical population of Germany. The datasets provided need to be treated and prepared before implementing machine learning algorithms.

Then our cluster analysis will be used to implement our supervised learning algorithm. In this context we will train and implement a supervised algorithm able to predict if a customer will respond positively to the mail-order campaign or not (binary classification problem). Then we will create a benchmark model to compare our final result and test the data.

Here below we have represented the workflow of this project and how we will proceed.



## *Problem Statement*

The goal in this project is to create a model capable of predicting which individual is likely to be in mail-order list of the marketing campaign or not. To do this we break it into two parts as described below:

### *1. Customer Segmentation Report*

In this section two datasets are provided for creating our unsupervised model. We have to note that all datasets provided will be treated (cleaning and preprocessing) in the same manner before implementing our models. We will discuss later about how to preprocess the data. For now we describe this section by dividing it in two parts; 1) Principal Components Analysis (PCA) for dimensionality reduction (simplification of data) and 2) K-means Clustering to create group of individuals and relate these groups to our mail-order marketing campaign.

### *2. Supervised Learning Model*

Once we have created our unsupervised model with K-means Clustering to groups of customers and identify in which cluster customers are more likely to be in mail-order campaign. Now it's time to build a supervised prediction model. The goal in this section is to create a binary classifier model. This model will be able to describe parts of the general population that are more likely to be part of the mail-order company's main customer base (labeled 1), and which parts of the general population are less so (labeled 0). Then we will create another model as a benchmark to comparison to our binary classifier.

## *Metrics*

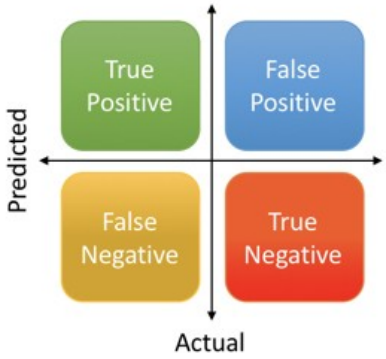
In this project, we'll want to evaluate the performance of our binary classifier and compare it to our benchmark; training it on some training data and testing it on test data that it did not see during the training process.

Once our model is trained, we can see how it performs when applied to the test data. To evaluate our predictor we'll calculate false negatives and positives as well as recall, precision, and accuracy.

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$




True Positive = Correctly predicted part of mail list (1s are 1s)
True Negative = Correctly predicted not part of mail list (0s are 0s)
False Positive = Uncorrectly predicted part of mail list (0s are 1s)
False Negative = Uncorrectly predicted not part of mail list (1s are 0s)

# *II. Analysis*

## *Data Exploration*

The dataset provided is splitted into two part:
1 - The first part ***Customer Segmentation Report*** is composed of two csv file (unsupervised learning):

- 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)

The general population dataset (AZDIAS) will be used to create our unsupervised model (PCA and K-means). Then customers dataset will be mapped into both models in order to identify patterns and relation between customers groups.

2 - The second part ***Supervised Learning Model*** is composed of two csv file:

- 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).

We can see a lot of missing or non value datas that should be cleaned and preprocessed before implementing unsupervised and supervised models. All datasets described above should be treated in the same way. That means the final cleaned and preprocessed datasets should have the same columns length.

## *Cleaning and Preprocessing*

Once our data is described and some of our features processed we are now ready to treat missing values. Arvato Financial Services has provided a real life dataset of demographics characteristics of customers for a general population located in Germany. Cleaning the data is deciding how to treat the missing values, for example if we drop or replace it. This is a key step which will influence our model performance in the next steps. In order to get a cleaned data, we made use of different preprocessing techniques to have a flawless data set. Data cleaning methods attempt to fill in missing values, smooth out noise, and correct inconsistencies in the data. In the last step we will normalize our data to get data values between 0 to 1 and also remove outliers.

It is important to note that all cleaning and preprocessing steps are applied on AZDIAS,Customers, Training and Test datasets in the same way. We will discuss about all steps later in Methodology section.

## *Exploratory Visualization*

After cleaning and normalizing the data we'll use visualization tools such as histogram and correlation matrix heatmap to represent distribution and features correlation. These representations will help us to understand how features are distributed and related to each other.

The histogram plots below show the distribution of our dimensionality reduced dataset. Each plot represent a component out of sixteen components we have.

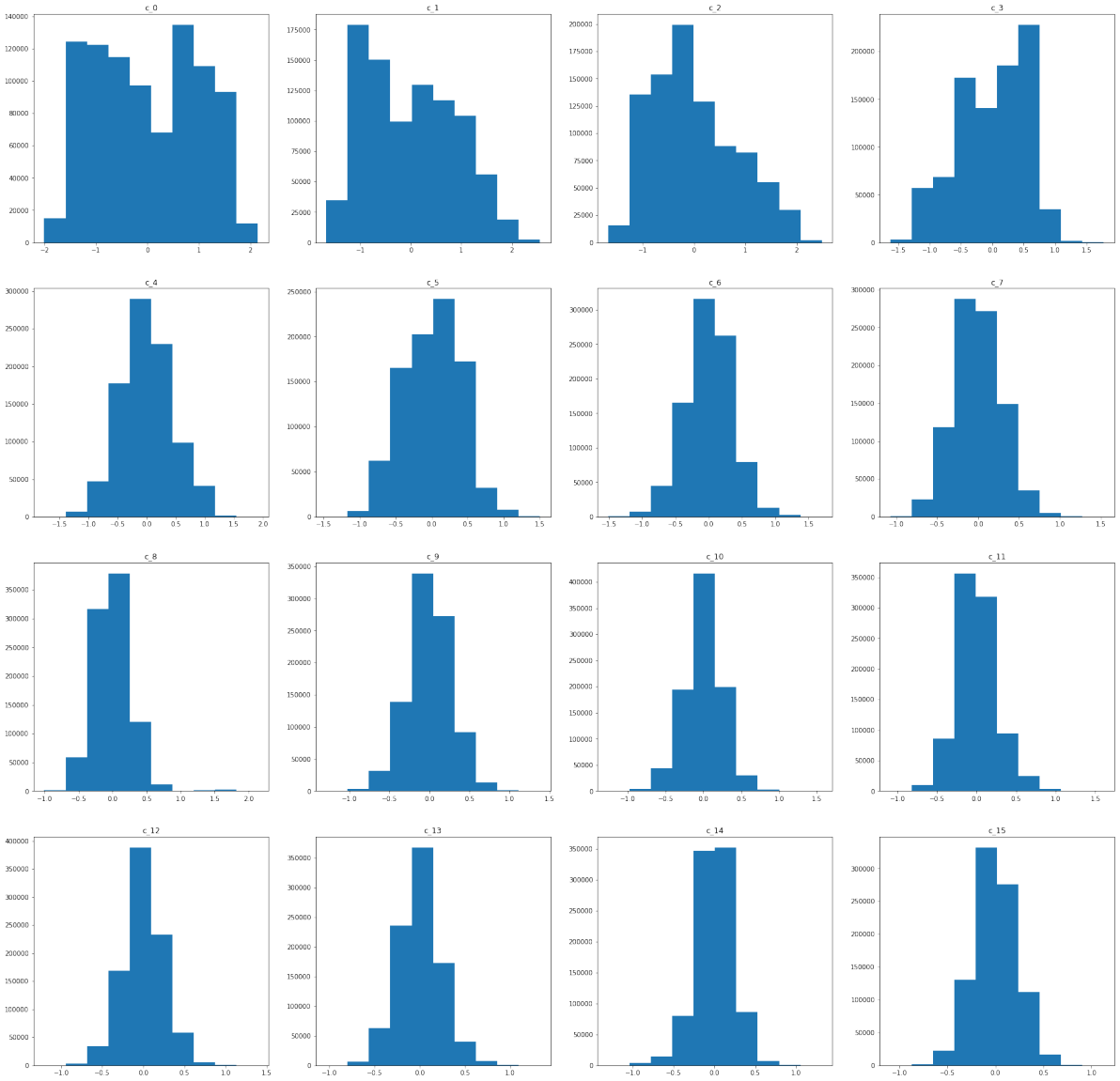



fig.1 – Data distribution

Heatmap is plotted using Seaborn package. Here we have used correlation heatmap to see how our components are connected to each other. However we observe that components are totally independent and it shows that our dimensionnality reduction has played an important role in simplification of the data.

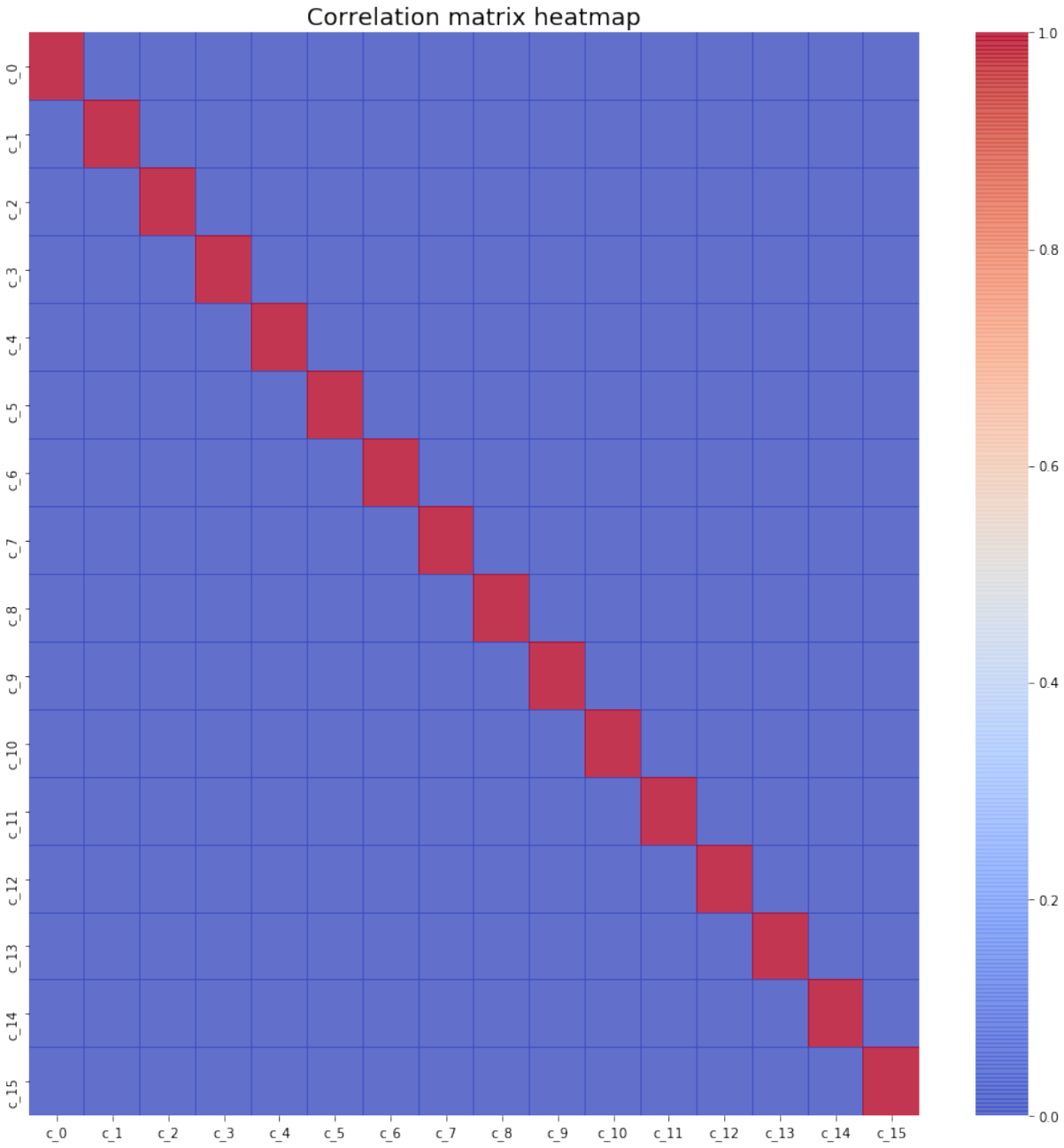



fig.2 – Feature Correlation Matrix

## *Algorithms and Techniques*

One of the most important part of a machine learning work flow is the algorithms that we choose to get an output from our model. For each step of this project we'll implement different

models. **Principal Components Analysis** for dimensionality reduction, **K-means clustering** used to group customers in clusters, **LinearLearner** for binary classification and **Logistic Regression** used as a benchmark, are well known algorithms used in modern case studies that we'll discuss about them here below.

### *1. Principal Components Analysis (PCA)*

Principal Component Analysis (PCA) is a technique which uses sophisticated mathematical principles to transforms a number of possibly correlated variables into a smaller number of variables called principal components. The origins of PCA lie in multivariate data analysis. One of the most important and perhaps its most common use is as to reduce dimensionality of large data sets. The large size of our datasets and features would be difficult to use through creating a clustering model. To prevent this, we'll use Principal Components Algorithms (PCA) to reduce the dimensionality of our preprocessed data. Our model is trained with azdias dataset, thereafter we choose the n_components (the number of components we want to keep) to retain regarding the explained variance (visualizing with elbow graph in figure 3). Once our azdias PCA model is ready we map the customers dataset into the azdias pca model. the PCA model used is imported from Scikitlearn package.

The following graph represent the number of components vs the explained variance. The goal is to reduce dimensionality as much as possible and capture minimum 80% of the explained variance.

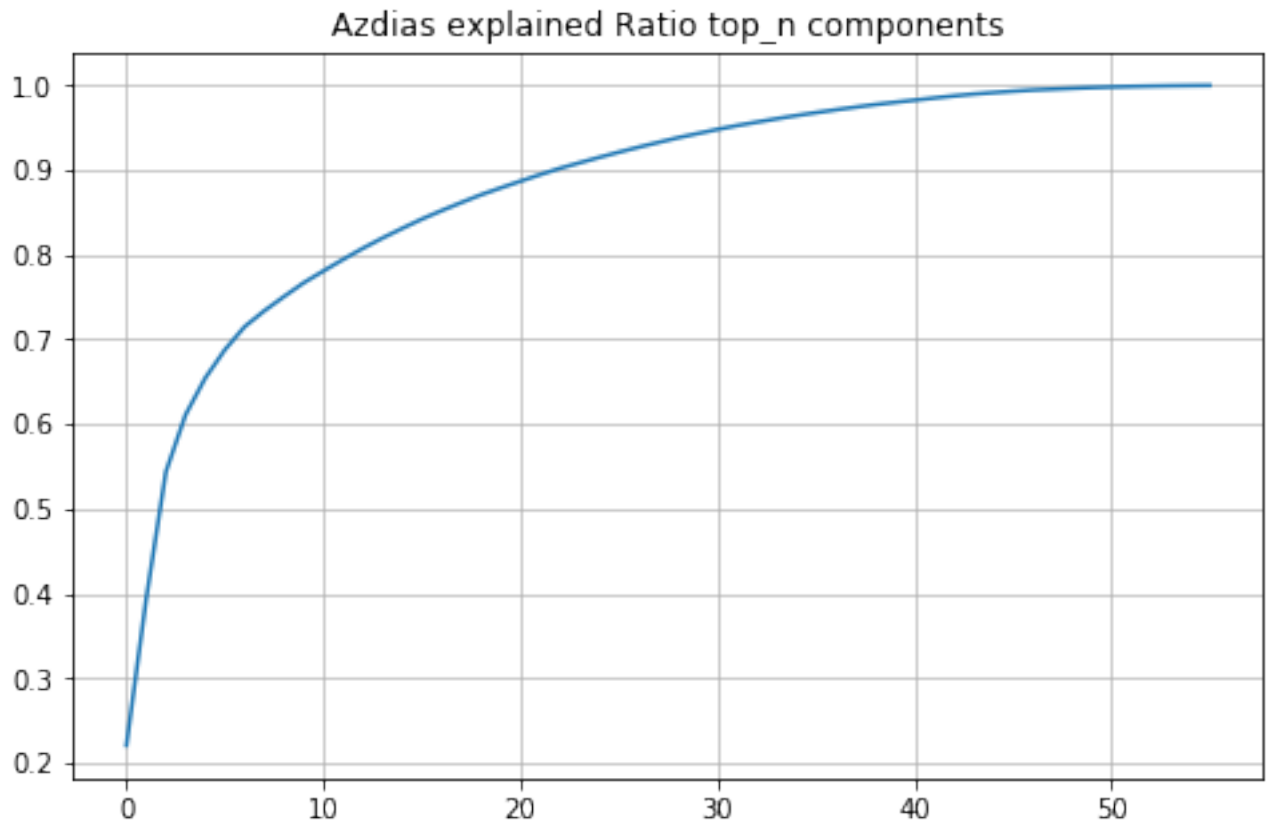



fig.3 – PCA components vs explained variance

In this context we chose n_components = 16 for an explained variance of 84.21%. Once we have our model ready the customers dataset is mapped into the azdias pca model and tranformed and reduced to 16 components.

### *2. K-means Clustering*

In this project, we'll use the unsupervised clustering algorithm, k-means, to segment general population using their PCA components, which are in the transformed DataFrame we just created. Then we will use this model to create segments for our customers dataset. The goal is to identify groups of individuals that have similar demographic characteristics and then relate this analysis to our goal which is to identify individuals which are more likely to respond to the mail-order list marketing campaign.

K-means create clusters, regardless of the actual existence of any structure in the data. When using K-means clustering, we are making an hypothesis of some structure among the objects. We should note that just because clusters can be found does not validate their existence. Only with strong conceptual support and good visualization clusters could potentially be meaningful and relevant. In the following parts we describe how we choose the k value and then how we create  our model. In this context we used different visualization tools to represent clusters and components correlation.

However the first step in this section will be to determine the optimal k value. For this purpose we create a plot representing k value vs the sum of squared distance to clusters centers.

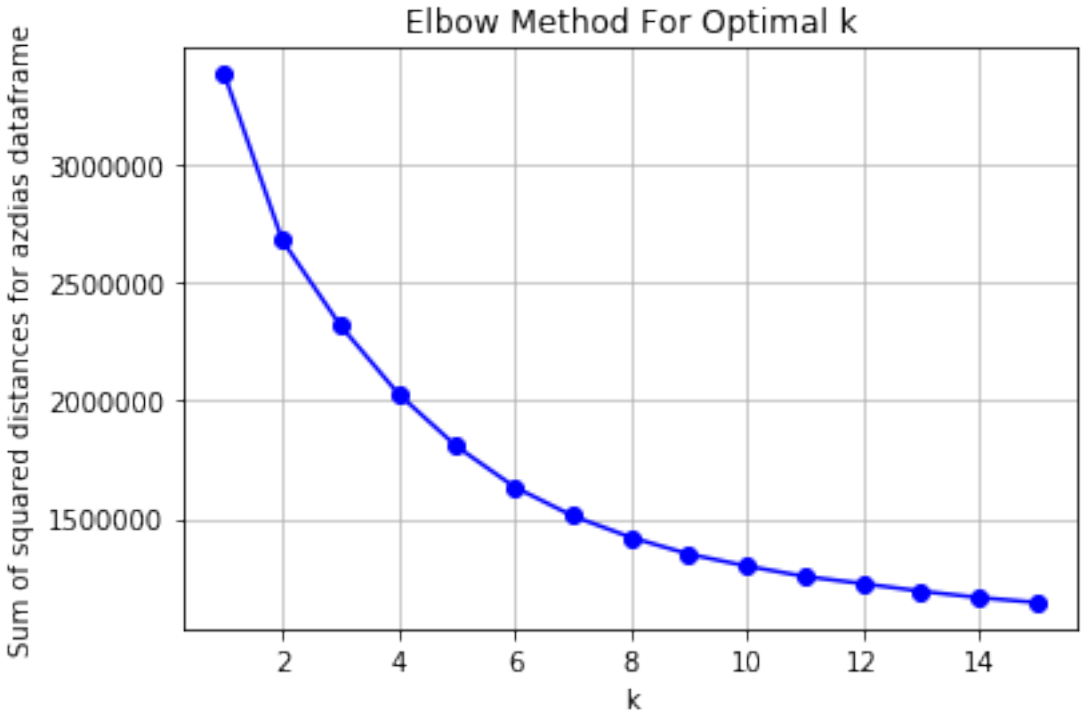



fig.4 – Optimal k elbow graph

In the plot the optimal k for this dataset is between 4 and 5. we chose k=4 for our model.

In the **Supervised Learning** section we will use this model for clustering of our reduced supervised dataset and try to regroup customers that are labeled 1 inside a specific cluster.

### *3. Linear Learner (SageMaker)*

*Linear Learner* is Amazon SageMaker builtin supervised learning algorithm used for solving either classification or regression problems. Here we give a high dimension multiclass dataframe as input and the output is a binary one dimensional dataframe where the label must be either 0 or 1. This algorithm allows us to explore a large number of models and choose the best, which optimizes either continuous objectives such as *mean square error*, *cross entropy loss*, *absolute error*, etc., or discrete objectives suited for classification such as *F1* measure, *precision@recall*, *accuracy*. However, in this project we will focus on optimizing recall and compensating imbalance using defined parameters implemented in this algorithm.

### *4. Logistic Regression*

Logisitc Regression is a statistical method used in machine learning and implemented here as a binary classifier. Here we will use this algorithm to explain the relationship between multiclass variables dataset and our binary label described as 0 for people predicted not to be in mail-order list and 1 for people predicted be in the mail-order list.

# *III. Methodology*

### *Data Preprocessing*

The dataset provided needs to be cleaned and preprocessed before feeding to PCA model.
To make it possible we proceed as follows:

1. Identifying  and dropping non values columns
2. Convert remaining non values data to numeric values

3. Replacing unknown and none data with NaN
4. Analyzing NaN values in dataset
5. Dropping columns with more than 20% of NaN values for azidas
6. Replacing remaining NaNs with -1
7. Checking data type and cleaned values
8. Normalizing the data

***Implementation***

The implementation process can be split into three main stages:

1. Dimensionality reduction with PCA
2. K-means Clustering
3. Linear Learner (Sage Maker)

***1. Dimensionality reduction with PCA***

During the first stage, the PCA was trained on the preprocessed data. This was done in a Jupyter notebook on Udacity work space “Bertelsmann/Arvato Project Workspace” (titled “Arvato Project Workbook.ipynb”), and can be further divided into the following steps:

1. Explore cleaned data attributes
2. Creating PCA model
3. Data Variance
4. Data variance vs dimensionality reduction
5. Component Makeup
6. Components Histogram
7. Correlation matrix heatmap

***2. K-means clustering***

Now we’ll ready to implement our k-means model. Before training we have to determine the k value, then we’ll use the reduced dimension data to train our k-mean model.

This section is break out to the following steps:

1. Determining the optimal number of clusters for k-means clustering
2. Creating K-means model
3. Predicting customers labels
4. Visualization
5. Natural groupings

***3. Linear Learner (Sage Maker)***

We'll have access to a third dataset with attributes from targets of a mail order campaign. We'll use the previous analysis to build a Linear Learner model that predicts whether or not each individual will respond to the campaign.
To build our supervised model using  Amazon SageMaker we will divide this section to the following steps:

1. Load preprocessed Data from S3
2. Splitting the data
3. Imbalanced training data
4. Create a LinearLearner Estimator
5. Convert data into a RecordSet format
6. Evaluating Model

***4. Benchmark***

Once our Linear Learner model is implemented and its performance is measured, it's time to create our benchmark model. The goal is to compare our binary classifier and its metrics to our benchmark model. In this context we will use Logisitc Regression provided by Scikitlearn to create our model and see if it outperforms our  Linear Learner model or not.

1. Preparing the data
2. Model Development and Prediction
3. Metrics
4. Oversample minority class
5. Compare to our model

# *IV. Results*

### *Model Evaluation and Validation*

The LinearLearner is used as a binary classifier. This SageMaker algorithm allows us to focus on the minority class accuracy trying to maximize True Positives and minimize False Negative (Recall). Moreover the Amazon SageMaker platform allows us to deploy the model and create an API in order to put the model in production environment. However instead tuning hyperparameters the result is not satisfying:

```
LinearLearner(role=role,
              train_instance_count=1,
              train_instance_type='ml.c4.xlarge',
              predictor_type='binary_classifier',
              output_path=output_path,
              sagemaker_session=sagemaker_session,
              epochs=20,
              binary_classifier_model_selection_criteria=
              'precision_at_target_recall',
              target_precision=0.8,
              positive_example_weight_mult='balanced')
```

```
rediction (col)   0.0   1.0

actual (row)
0.0               4247  6354
1.0                 23   117

Recall:     0.836
Precision:  0.018
Accuracy:   0.406
f1_score:   0.035
```

### *Justification*

Both models have not show a satisfying results, we have tried to compensate the imbalance of positive label and focus on recall to get the best predicitve result for our minority class. Here below we are comparing results between two models:

| | Recall | Precision | Accuracy | F1 |
|---|---|---|---|---|
| Linear Learner | 0.836 | 0.018 | 0.406 | 0.035 |
| Logisitic Regression | 0.671 | 0.021 | 0.589 | 0.041 |

## *V. Conclusion*

### *Free-Form Visualization*

Visualization helped to get an overview of the data and also to see the improvements that we made during the project. Here below we can observe how the azdias PCA model has simplify the data (figure 5 and 6). In the other figure (figure 7) we have represented a 2D and 3D plots of clusters made it by azdias K-means model where we can differentiate different clusters.

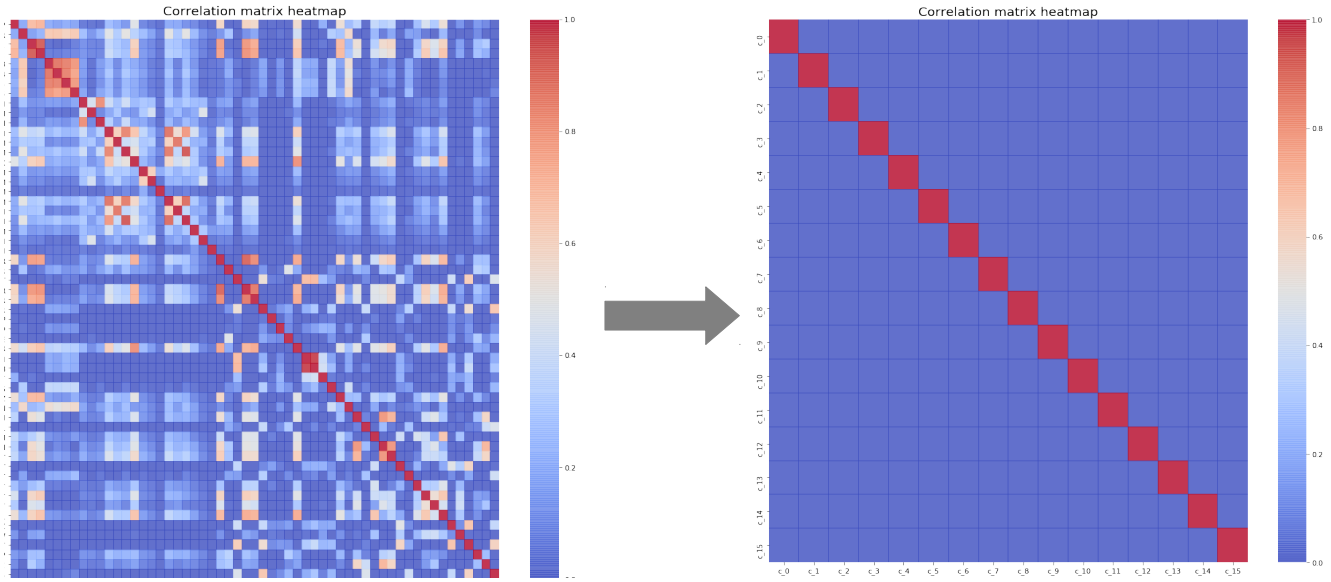


fig.5 – Correlation Heatmap: before and after PCA

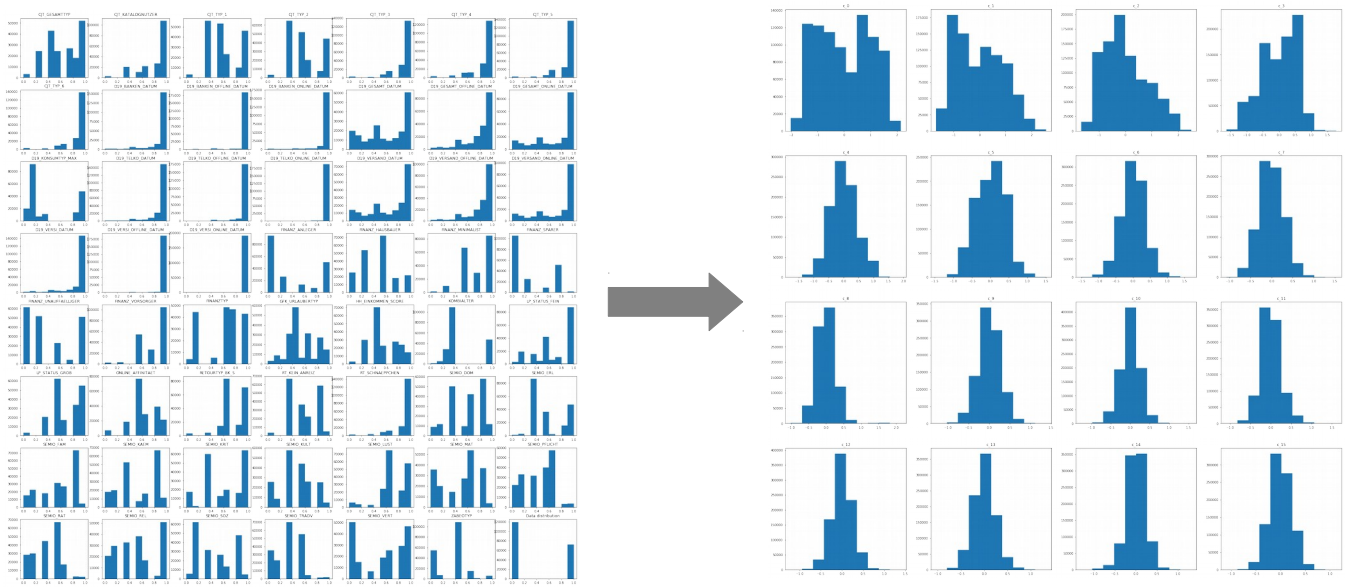
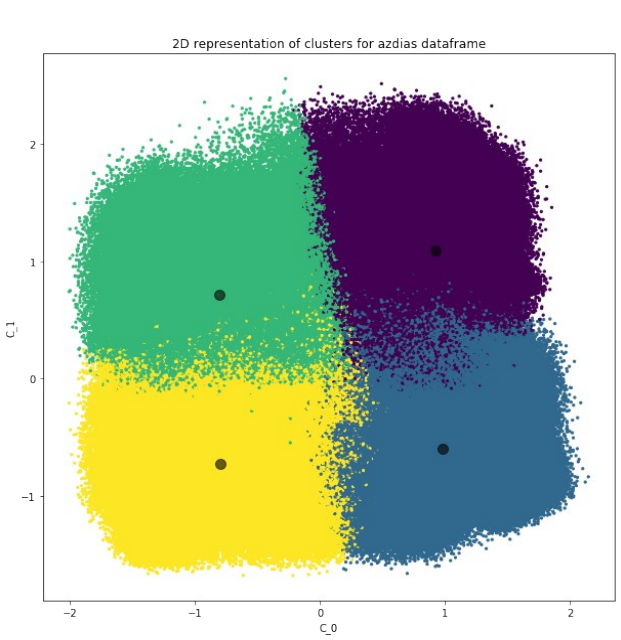


fig.6 – Features Histogram: before and after PCA



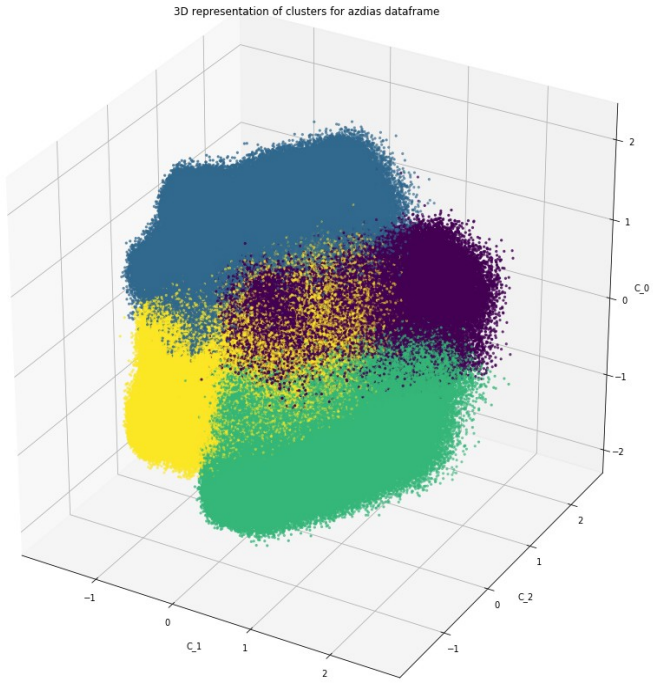



fig.7 – 2D and 3D scatter plot of K-means clusters

### *Reflection*

In this project we tried different machine learning techniques and algorithm to implement a robust model capable to predict if a customer has the potential to respond positively to mail-order marketing campaign or not. However after evaluation we can observe that our models are not performing well and some improvement has to be made. In the other hand we have not be able to establish a relation between our clustering model and the supervised data. In this context, it will be interesting to test other algorithms such as Convolutional Neural Network (CNN) and try to tune this model with different architectures.

### *REFERENCES*


Udacity Machine Learning Case Study (2017). Fraud Detection
Shruti Saxena (2018). Precision Vs Recall, *TowardsDataScience.com*
Franklin, Joel N. (1968). Matrix Theory. Dover Publications.
Siotani, Minoru (1964). Tolerance regions for a multivariate normal population
Czar Robero (2018). K- means Clustering Tutorial
Andrew Aziz (2017). Customer Segmentation based on Behavioural Data in E-marketplace
Joseph F. Hair, Jr. William C. Bl, Barry J. Babin, Rolph E. Anders (2014). Multivariate Data Analysis