

# Machine Learning Engineer Nanodegree

## Capstone Project

John Anisere

January 26th, 2020

## I. Definition

### Project Overview

Most Businesses around the world oftentimes have to do some marketing to gain new customers.

In project I used the data provided by Arvato Financial Solutions, a Bertelsman subsidiary company, to analyse the demographics data for customers of a mail-order sales company in Germany, comparing it against demographics information of the general population.

I also performed customer segmentation by applying unsupervised learning technique to uncovered the demographic characteristics of the core customers. Identifying the part of the population that best describe the customer base of the company.

In addition to that, I will also built a model by applying supervised learning technique using another dataset from a marketing campaign of the company, to predict individuals that are most likely to be the potential customer of the company.

Finally, I participated in a Kaggle competition to evaluate the supervised learning model I created with evaluation metric of AUC

for the ROC curve, where the ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

## Problem Statement

The goal of this project is to predict which individuals are most likely to convert into becoming customers for a mail-order sales company in Germany. The tasks involved are the following:

1. Download and preprocess the Customer and general population demographics data
2. Train a classifier by applying unsupervised learning technique
3. Train a classifier by applying supervised learning technique
4. Evaluate the supervised learning model by participating in a Kaggle competition

## Metrics

The evaluation metric is **AUC for the ROC curve**, relative to the detection of customers from the mail campaign. A ROC, or receiver operating characteristic, is a graphic used to plot the true positive rate (TPR, proportion of actual customers that are labeled as so) against the false positive rate (FPR, proportion of non-customers labeled as customers).

$$A = \int_{x=0}^1 \text{TPR}(\text{FPR}^{-1}(x)) dx = \int_{-\infty}^{\infty} \text{TPR}(T) \text{FPR}'(T) dT = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(T' > T) f_1(T') f_0(T) dT' dT = P(X_1 > X_0)$$

\*[https://en.wikipedia.org/wiki/Receiver\\_operating\\_characteristic#Area\\_under\\_the\\_curve](https://en.wikipedia.org/wiki/Receiver_operating_characteristic#Area_under_the_curve)

## II. Analysis

### Data Exploration

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). In this dataset, 273 columns have naturally missing data. With columns `KK_KUNDENTYP`, `EXTSEL992`, `ALTER_KIND1`, `ALTER_KIND2`, `ALTER_KIND3` & `ALTER_KIND4` having more than 65% of missing values. Once I mapped the values(values coded as `missing_or_unknown`) to `NaN`, the total number of missing values increased from 33,492,923(natural missing observations) to 37,827,227(natural missing observations and observations with `NaN`) which corresponds to increase of 11.46% or we can say that 4,334,304 observations were coded as missing observations apart from naturally missing observations.
- `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).

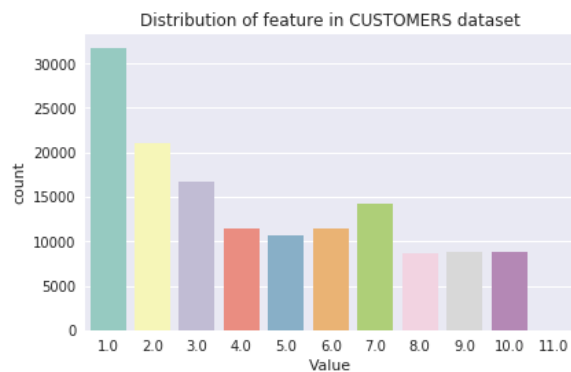
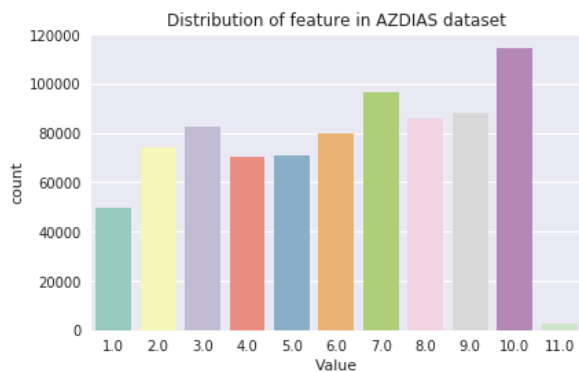
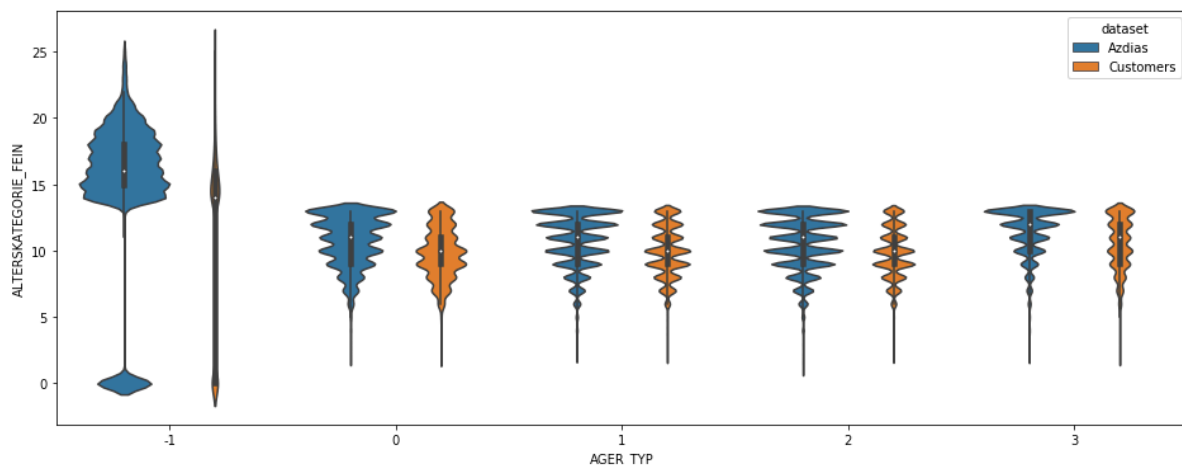
# Exploratory Visualization

Analyszing the AZDIAS dataset we have:

	LNR	AGER_TYP	AKT_DAT_KL	ALTER_HH	ALTER_KIND1	ALTER_KIND2	ALTER_KIND3	ALTER_KIND4	ALTERSKATEGORIE_FEIN	ANZ
count	8.912210e+05	891221.000000	817722.000000	817722.000000	81058.000000	29499.000000	6170.000000	1205.000000	628274.000000	
mean	6.372630e+05	-0.358435	4.421928	10.864126	11.745392	13.402658	14.476013	15.089627	13.700717	
std	2.572735e+05	1.198724	3.638805	7.639683	4.097660	3.243300	2.712427	2.452932	5.079849	
min	1.916530e+05	-1.000000	1.000000	0.000000	2.000000	2.000000	4.000000	7.000000	0.000000	
25%	4.144580e+05	-1.000000	1.000000	0.000000	8.000000	11.000000	13.000000	14.000000	11.000000	
50%	6.372630e+05	-1.000000	3.000000	13.000000	12.000000	14.000000	15.000000	15.000000	14.000000	
75%	8.600680e+05	-1.000000	9.000000	17.000000	15.000000	16.000000	17.000000	17.000000	17.000000	
max	1.082873e+06	3.000000	9.000000	21.000000	18.000000	18.000000	18.000000	18.000000	25.000000	

8 rows x 360 columns

And then taking a look at the distribution comparison of some random features between AZDIAS and CUSTOMERS dataset:



The Description files gave a detailed information of the meaning of each label including how missing value are labeled. I mapped the label of missing value back to NaN and investigate the number of missing value in each columns. In the end, columns with missing value higher than 20% were dropped where most of them were higher than 50%.

## **Algorithms and Techniques**

**Unsupervised Learning:** Principal Component Analysis (PCA) is one of the most useful techniques in Exploratory Data Analysis to understand the data, reduce dimensions of data and for unsupervised learning. In the end, I kept 82 principal components and the cumulative variance is more than 50%. I applied k-means clustering to the dataset and used the average within-cluster distances from each point to their assigned cluster's centroid to decide the number of clusters to keep.

Finally, I clustered the data based on demographics of the general population of Germany, and seen how the customer data for a mail-order sales company maps onto those demographic clusters. And then, compare the two cluster distributions to see where the strongest customer base for the company is.

**Supervised Learning:** I applied supervised learning to investigate MAILOUT\_TRAIN and MAILOUT\_TEST dataset to predict whether or not a person became a customer of the company following the campaign. After some quick investigation against MAILOUT\_TRAIN dataset, I can find among 43 000 individuals, only 532 people response to the mail-out campaign. Which means the training data is highly unbalanced. Based on this discovery, we have to split the data based on the distribution of target.

After testing the performance of several models based on cross-validation result, I used 'XGBClassifier', and trained the model by using a 5 folds validation method. Finally, I submit the result to Kaggle competition and achieved a 0.80124 roc\_auc\_score.

## Benchmark

Because this is a classification problem, the benchmark model is the random forest model. According to this article <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2264-5> , the random forest model performs better for large datasets

## III. Methodology

### Data Preprocessing

The pre-processing done consist of the following steps:

1. Check for extra columns in customers dataset : Three extra columns in customers dataset does not appear to be demographic data like all the columns in general population dataset. Therefore, drop these 3 extra columns from our analysis.
2. Assess Missing Data before re-encoding missing value codes
3. Convert Missing Value Codes to NaNs
4. Assess Missing Data in Each Column after re-encoding missing value code
5. Drop outliers columns i.e. columns with more than 20% of missing values
6. Assess Missing Data in Each Row
7. Drop rows with more than 8 missing values. With 8 or less missing data in rows, we still have 81.81% of data retained
8. Check columns which didn't had a description in feature summary file
9. Drop the columns which didn't had a description in feature summary file
10. Take a sample representative of a large dataset

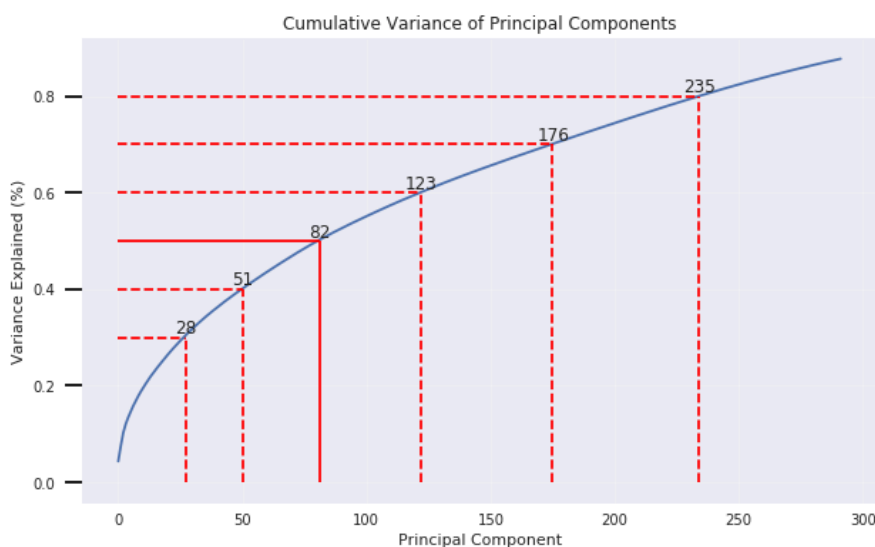
11. Re-Encode binary categorical features & create dummy variables for multi-categorical features.
12. Engineer Mixed-Type Features
13. Complete Feature Selection
14. Create a Cleaning Function `clean_data()` that performs all the above steps so that it can be run on the customers dataset as well

## Implementation

The implementation process can be split into two stages:

1. Unsupervised learning stage
2. Supervised learning stage

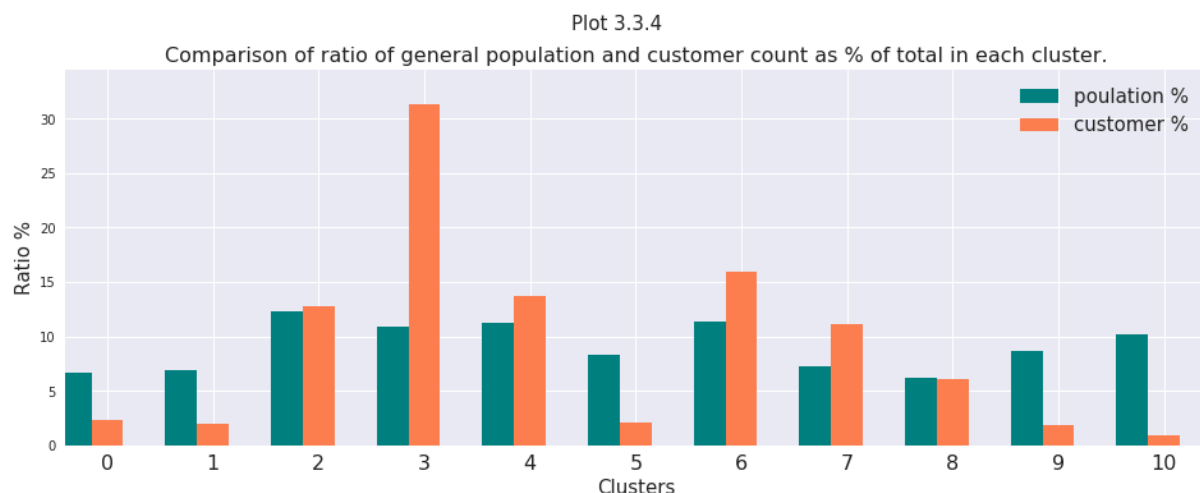
During the unsupervised learning stage, I applied PCA (principal component analysis) to reduce the dimensionality of the data. From the plot of cumulative variance of principal components, I observed that with every 10% increase of variance, the number of components required for the same increase kept on increasing i.e. to get 10 % increase from 30% to 40% to 50% to 60% to 70% to 80% of the total variance, it required to increase components from 28 to 51 to 82 to 123 to 176 to 235 which corresponds to increase of 23 to 31 to 41 to



53 to 59 components respectively.

At 82, 123, 176 and 235 principal components, we had 50%, 60%, 70% and 80% of the total variance of the dataset. So, proceeding with 82 principal components which explains 50% of the total variance in the dataset and should be helpful to much extent in generalizing the customer segments during cluster analysis process.

I used sklearn's KMeans class to perform k-means clustering on the PCA-transformed data. Then, computed the average difference from each point to its assigned cluster's center. I used this fact to select a final number of clusters in which to group the data. Then I selected a final number of clusters to use to re-fit a KMeans instance i.e. 11 to perform the clustering operation and obtain cluster predictions for the general population demographics data and then compare the customer data to the general population demographics data.



During the supervised learning stage, I discovered that the dataset was highly imbalanced because only 532 individuals responded to the mail-out campaign. Because of that, before the imbalanced dataset can be used as input for machine learning algorithms, it must be cleaned, formatted, and restructured. I performed feature transformation by Imputing the missing value in the cleaned train dataset with with median value along the columns by using the Imputer instance and then feature scaling the imputed dataset using StandardScaler making our training dataset to be ready to fed into various machine learning algorithms.

I trained different classifiers which includes the following classifiers ( LogisticRegression, RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier, SVC, LinearSVC, LGBMRegressor, XGBClassifier, GaussianNB , BernoulliNB and LGBMClassifier) on training



dataset to find the best performing classifier algorithm using the GridSearchCV object which used 5 StratifiedKFold and roc\_auc score as a scorer. The XGBClassifier performed best with an roc\_auc score of 0.7587.

## Refinement








After Tuning the best classifier(XGBClassifier) with the help of param grid in GridSearchCV the roc\_auc score improved slightly from 0.7587 to 0.7685. The score could be improved further by trying further again with different combinations of hyper parameters but at the cost of more computing time.


## IV. Results

### Model Evaluation and Validation

To evaluate the model's performance, I used the model to predict which individuals are most likely to respond to a mail-out campaign through the Kaggle competition.

After submission, a kaggle score of 0.80124 was achieved.

25	Ahmed		0.80226	30	17h
26	Sebastian Koenig		0.80225	4	19d
27	DeepVen		0.80198	1	10mo
28	Enyue Jia		0.80176	56	1y
29	Mei Eisenbach		0.80143	21	1y
30	wyy123		0.80135	6	1y
31	John Anisere		0.80124	3	3d

Your Best Entry 

Your submission scored 0.79853, which is not an improvement of your best score. Keep trying!

Given that the model is highly unbalanced, I believe the model is doing very fine.

## **Justification**

Comparing the performance of the RandomForestClassifier to the XGBClassifier, it's clear that the later performed better. Also XGBClassifier which is a gradient boosted classifier, is a very good fit for this problem given that we are dealing with a very large dataset for a classification problem.

## **V. Conclusion**

### **Reflection**

This was my first time working on real life dataset and it has been a great learning experience on how to approach the problem in a methodical approach. Most challenging part for me was mainly on getting the data cleansed and processed without losing key information.

In the Customer Segmentation part, we have performed data pre-processing and used PCA method combined with k-NN algorithm to get the clusters within the general population and customers population. The clusters obtained were compared and we were able to observe the differences in the clusters allocation. The differences were discussed and I was able to figure out which segment of the general population would be the biggest customer segment for mail order company and which segment would be the least relevant to new customer base. With the understanding of this difference, a company could focus more on biggest customer segment within the general population and then perform target mail-out campaign. This would increase the customer conversion rate and lower down the marketing cost. The impact of using machine learning for such kind of target mail-out campaign is large.

In the supervised learning part, I was able to predict the class probabilities of each individual in the testing set to become a customer. I did a decent job in

this step and got the decent roc\_auc score of 0.80124 in Kaggle in-class competition.

## Improvement

I think there are few areas like feature engineering, further fine tuning the model where more focus could be concentrated on which I believe will help to improve prediction performance further.

*The roc\_auc score could further be improved by trying out the following —*

- Performing more feature engineering either manually or automated feature engineering by using **Featuretools**, an open-source Python library for automated feature engineering.
- Trying out more hyper parameter tuning of the best classifier so far and using the best optimized classifier model.