Capstone Project

Machine Learning Engineer Nanodegree

Keyvan Tajbakhsh
July 14th, 2019

Customer Segmentation Report for Arvato Financial Services

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.

In this project we'll use unsupervised learning techniques to describe the relationship between the demographics of the company's existing customers and the general geographical population of Germany.

After dimensionality reduction and clustering we'll build our model regarding supervised data using PyTorch. Our 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, and which parts of the general population are less so.

Problem Statement

To overcome this project we'll break it into four notebook files. Each section is also divided in different parts that we will discuss later.

1. Data exploration

One of the most important step in machine learning work flow is data exploration and visualization. In order to understand the problem, we'll need to clean and prepare the data to feed our machine learning models in the next parts. In this project, our dataset needs to be treated since it contains NaN values and data types that are not suitable for this project.

When the data is prepared it will be very useful to visualize and identify different variables and correlations.

2. Feature engineering

Feature engineering is also a data preparation process. One modifies the data such that machine learning algorithms identify more patterns. This is done by combining and transforming existing features into new features.

In this project we'll use principal components analysis (PCA) to reduce dimensionality and get the maximum variance at the same time. This technique helps us to simplify the data and get better view of different groups of features.

3. Training segmentation model

Clustering is commonly used in population segmentation specially for marketing purposes. Clustering is an unsupervised machine learning technique, where there are no defined dependent and independent variables. The patterns in the data are used to identify group with similar observations.

We'll use k-means clustering for creating customer segments based on the new components extracted from pca model. k-means clustering is an iterative algorithm where the number of clusters k is predetermined and the algorithm iteratively assigns each data point to one of the k based on the feature similarity.

4. Training supervised model

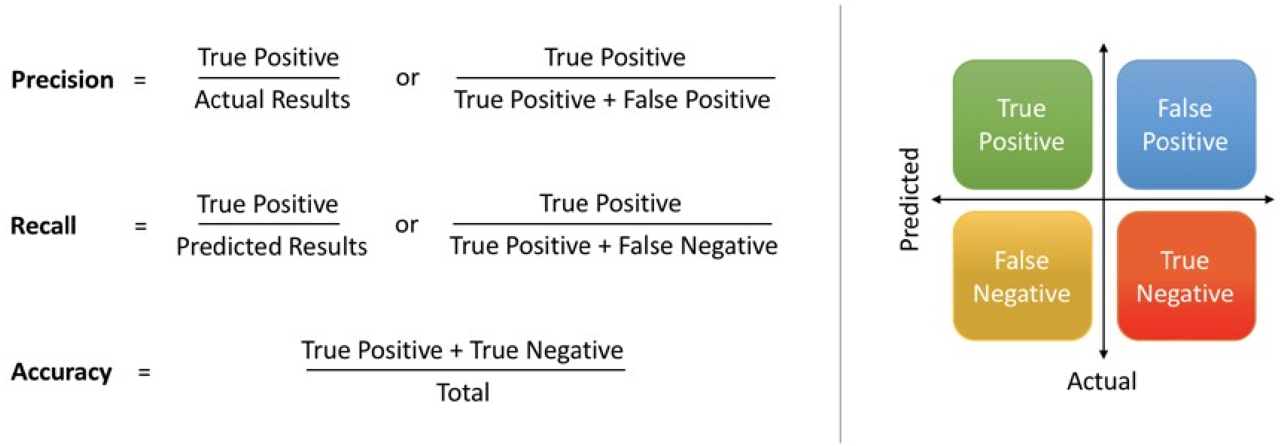
The final step we'll be to train a supervised model using data previously prepared/grouped to create a recommendation mail-order for different company regarding geographic data of people in Germany. For this purpose, we'll implement a convolutional neural network (CNN) using PyTorch module to build a machine learning model that predicts whether or not each individual will respond to the mailing campaign.

The CNN gets its name from the process of Convolution, which is the first filter applied as part of the feature-engineering step. Convolution is similar to applying a filter to our data. We pass over our reduced grouped data, usually called a kernel, and output the resulting, filtered subset of our data.

Metrics

In this project, we'll want to evaluate the performance of our CNN classifier; 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 and deployed, 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 [1, 2].



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

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

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

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)

Analysis

Data Exploration

The dataset provided is spitted to two part:
1 - The first part ***Customer Segmentation Report*** 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)

2 - The second part ***Supervised Learning Model*** composed also 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).

The first part is composed of two csv data files. We can see a lot of missing data or non value data which it should be treated in order to work on PCA and clustering models.

Exploratory Visualization

After cleaning and preprocessing the data we'll use visualization tools such as histogram and correlation matrix heatmap to represent distribution and features correlation.

The histogram plots below show different distribution of features and as we can see there is some similarities between features.

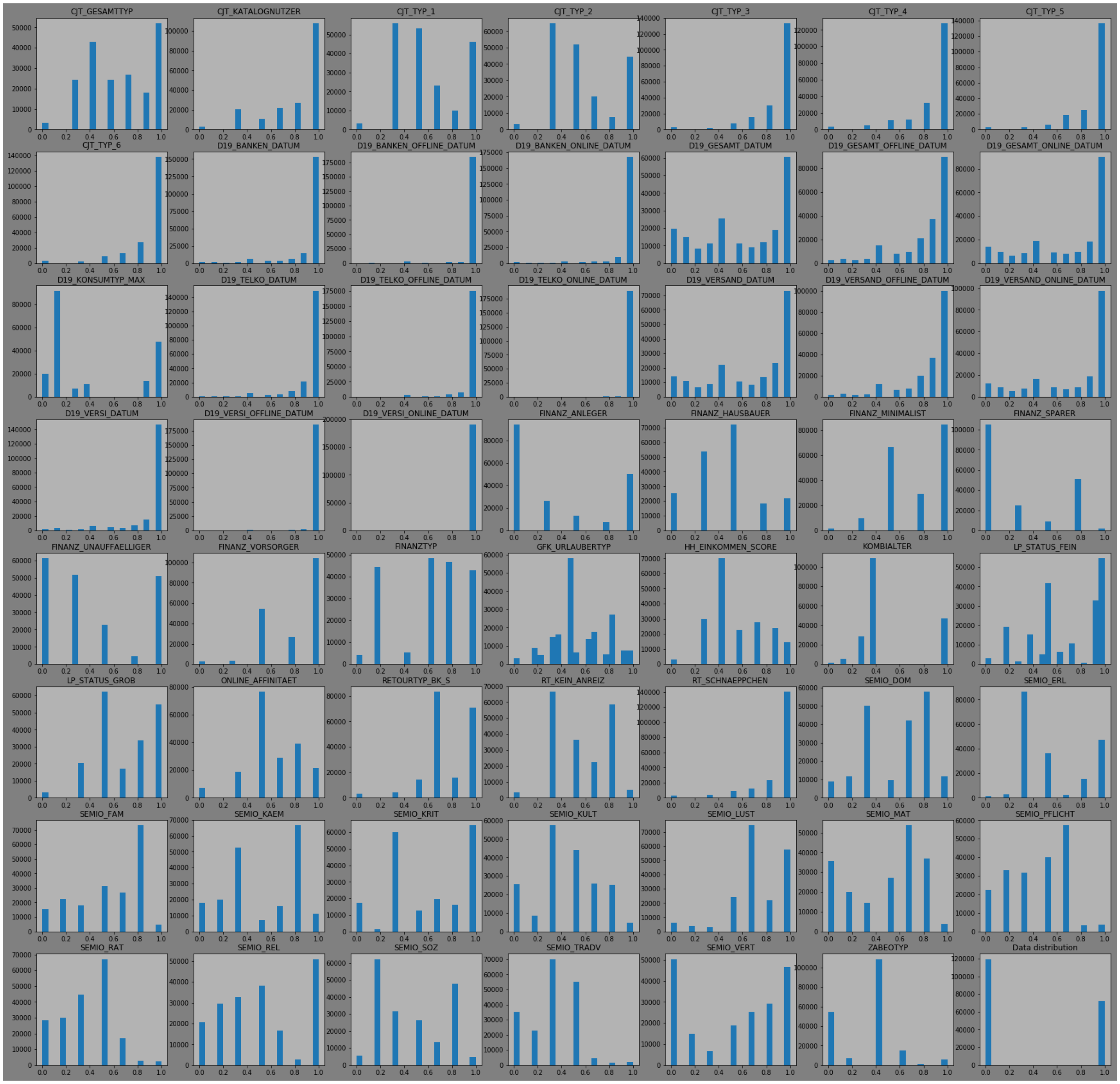


fig.1 – Data distribution

In the correlation features heatmap below, we can distinguish different correlation zones in red where it is important to note the relations between groups of features.

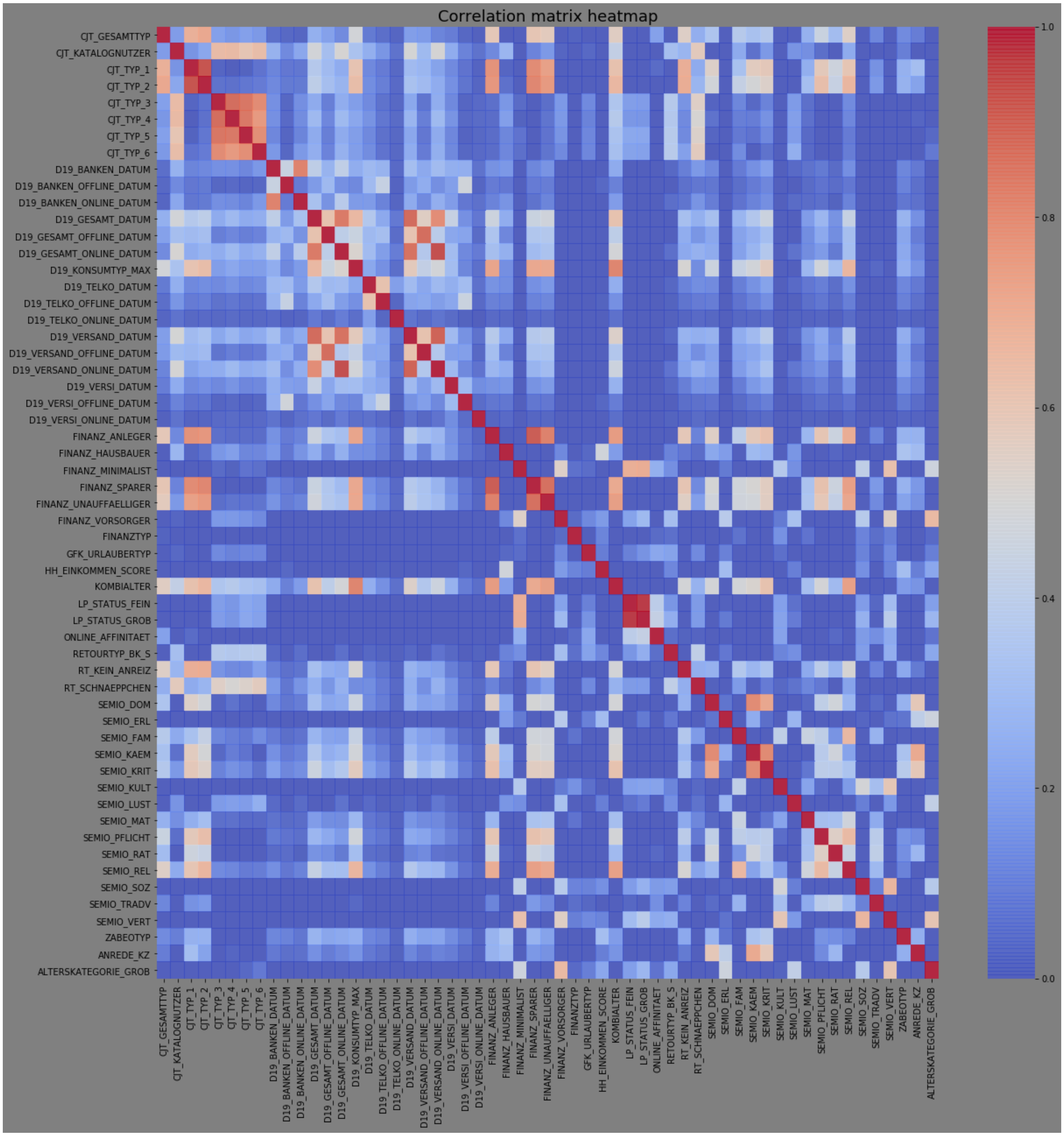


fig.2 – Feature Correlation Matrix

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 techniques. Principal Components Analysis, K-means clustering and Convolution Neural Network are three well known algorithms used in modern case studies that we'll discuss about them in this section.

1. Principal Components Analysis (PCA)

PCA is mostly used as a tool in exploratory data analysis and for making predictive models. It is often used to visualize genetic distance and relatedness between populations. PCA can be done by *eigenvalue decomposition*[2] of a data covariance (or correlation) matrix or singular value decomposition of a data matrix, usually after a normalization step of the initial data. The normalization of each attribute consists of mean centering – subtracting each data value from its variable's measured mean so that its empirical mean (average) is zero – and, possibly, normalizing each variable's variance to make it equal to 1. The results of a PCA are usually discussed in terms of component scores, sometimes called factor scores (the transformed variable values corresponding to a particular data point), and loadings (the weight by which each standardized original variable should be multiplied to get the component score). If component scores are standardized to unit variance, loadings must contain the data variance in them.

The following example figure represent a scatter plot of Gaussian Distribution multidimensional data set and how PCA could reduce conditionality which represented by two vectors.

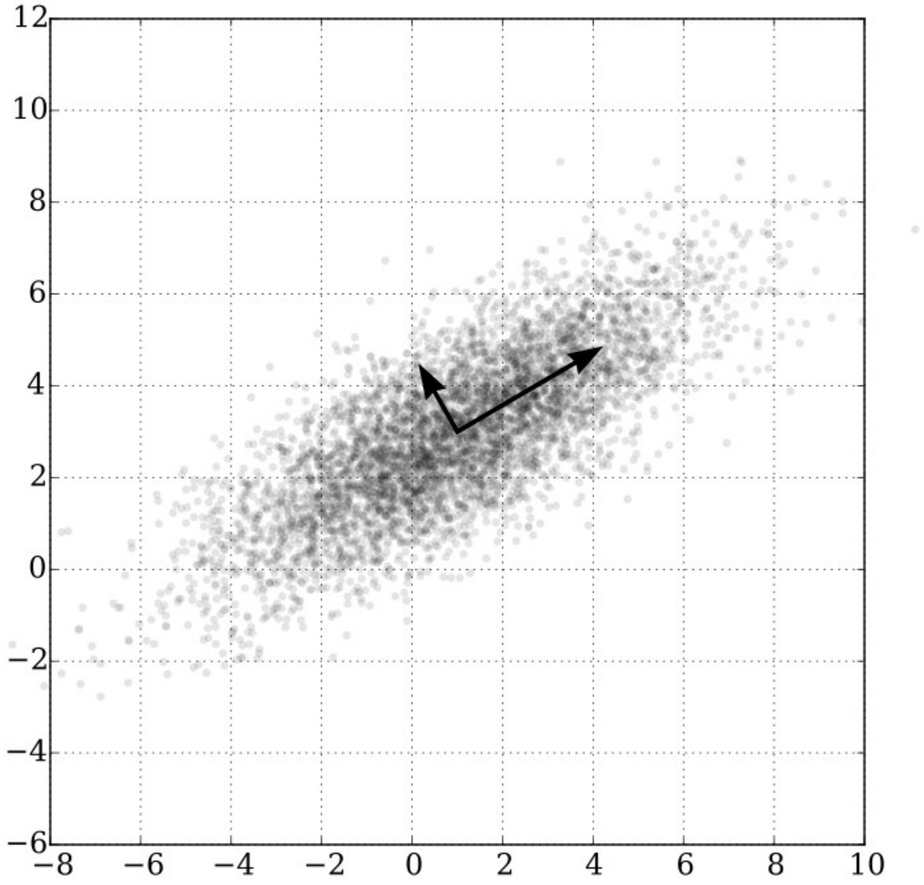


fig.3 – PCA of a multivariate Gaussian distribution centered at (1,3). The vectors shown are the eigenvectors of the covariance matrix scaled by the square root of the corresponding eigenvalue, and shifted so their tails are at the mean [3].

The Large size of data and feature components 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 the preprocessed data.

2. K-means Clustering

In the second part, we'll use k-means clustering to assign each customer to a particular cluster based on where a group lies in component space. How each cluster is arranged in component space can tell us which customers are most similar and what demographic traits define that similarity, this information is used to create a targeted mail list.

The following figure is an example illustrating the K-means algorithm on a 2-dimensional data set. The black dots represent centroids of different class (1,2 and 3).

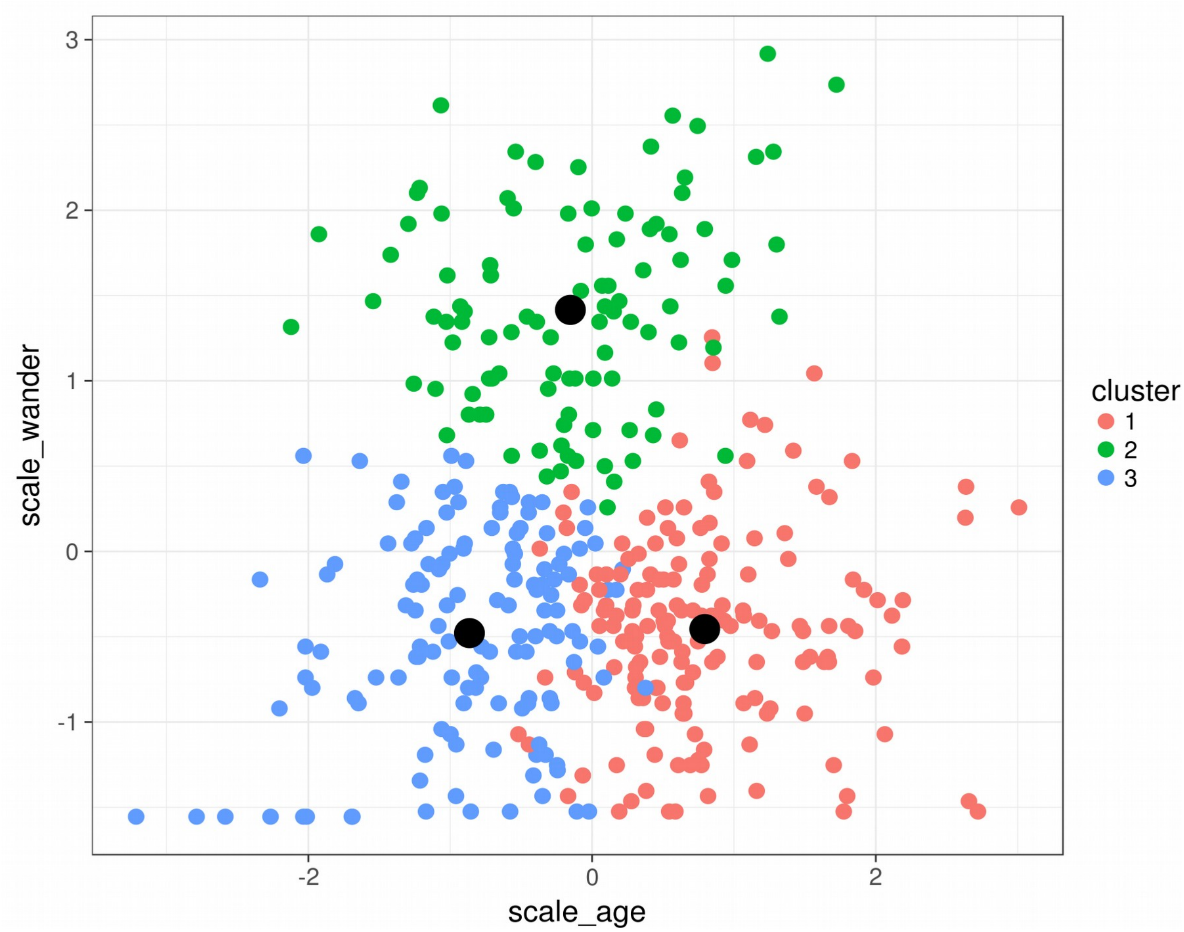


fig.4 – Graphical representation of the clusters that could be formed using the algorithm [4]

3. Convolutional Neural Network (CNN)

The classifier we'll use for supervised learning is a Convolutional Neural Network , which is commonly used for classification problems.

Benchmark

- 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).

Demographics data for individuals who were targets of a marketing campaign will be used as a benchmark and comparison to our model.

Our metrics (accuracy, recall and precision) threshold are set to 0.8 for this project. We consider these values "reasonable" regarding different customers segmentation projects norms [5].

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 10% of NaN values
6. Dropping the rows only if all of the values in the row are missing
7. Replacing NaNs with -1 value
8. Checking data type and cleaned values
9. Normalizing the data

Implementation

The implementation process can be split into three main stages:

1. Dimensionality reduction with PCA
2. K-means clustering
3. Multi-classification with CNN

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. Data Modeling
2. Define PCA model
3. Train the model
4. Accessing the PCA Model Attributes
5. Component makeup

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. Define a k-means model
2. Choosing k value
3. Create training data
4. Deploy the k-means model
5. Pass the training data to predictor and evaluate the results
6. Exploring the result clusters and visualize the distribution
7. Model attributes & explainability
8. Visualizing centroids in components space
9. Natural groupings with geographical position in Germany

3. Binary classification with PyTorch

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 machine learning model that predicts whether or not each individual will respond to the campaign.
To build our supervised model using Convolutional Neural Network (CNN) model and deploy it on a API gateway we’ll divide this section to the following steps:

1. Create model
2. Define a train script
3. Create a PyTorch estimator
4. Train the estimator
5. Create a trained model
6. Deploy the trained model
7. Evaluate the model
8. Test the model
9. Create an API gateway

Improvement

To achieve the optimal evaluation scores (accuracy, recall and precision) we should optimize our model.

In this case, we want to build a model that has as many true positives (1s are 1s) and as few false negatives, as possible. This corresponds to a model with a high **recall:** true positives / (true positives + false negatives).

Our labeled benchmark dataframe shows low number of 1s and this imbalance could affect our model. After implementing a model that is tuned to get a higher recall, which aims to reduce the number of false negatives, we’ll focus on class imbalance that may actually bias our model towards predicting if a customer is in the mail list or not.

Therefore, It will be important to tune hyperparameters to achieve our goals mentions above.

INDEX

[1] Udacity Machine Learning Case Study (2017). Fraud Detection

[2] Shruti Saxena (2018). Precision Vs Recall, TowardsDataScience.com

[3] Franklin, Joel N. (1968). Matrix Theory. Dover Publications.

[4] Siotani, Minoru (1964). Tolerance regions for a multivariate normal population

[5] Czar Robero (2018). K- means Clustering Tutorial

[6] Andrew Aziz (2017). Customer Segmentation based on Behavioural Data in E-marketplace