Clustering of Customer Data

Patrick Brus

27.07.2021

Abstract

Customer clustering is a powerful method to group customers based on given data without requiring labeled data. This report uses customer data from Kaggle and groups the customers based on the given features into four different clusters. The optimal number of clusters is found by computing the inertia value for different unsupervised models and using the elbow method to get the optimal number of clusters. The K-Means algorithm is used as unsupervised clustering algorithm.

1 Introduction

Customer clustering can be very helpful for finding useful information about customers. Clustering can help to gain knowledge of which type of customers are buying which products, which in turn can be very helpful for sending customized advertisement to the correct group of customers. Clustering is very powerful, because no labeled data is required. The Kaggle data set on [1] contains information about 2000 different customers. The goal of this report is to cluster the customers into different groups in order to gain some insights about the different groups of customers. Before applying unsupervised machine learning, the data set is explored and some feature engineering is applied in order to understand the data and to get the best performance of the machine learning algorithm. This report summarizes the findings after performing some Exploratory Data Analysis (EDA). In addition, feature engineering and pre-processing is applied in order to prepare the data set for machine learning. Afterwards, a K-Means model is trained with clusters ranging from 1 to 10 and the elbow method is used in order to find the optimal number of clusters. The best performing is then used as final model. The code can be found on my Github page [1] in Jupyter notebooks.

2 Attribute Information

This chapter gives a quick overview of all available features and a short description of them. Table 1 shows all feature names and a short description of each feature.

3 Exploratory Data Analysis, Feature Engineering and Data Cleaning

This chapter describes the process of analyzing the data, cleaning it and pre-processing the data. These steps are combined in one chapter, because they often go hand in hand. As first step, the

[1] link to Github: `https://github.com/patrickbrus/IBM_Machine_Learning_Professional/tree/master/Clustering`

pandas data frame head function is used to get an initial view into the data. In total, the data set contains 2000 samples with 8 features each. Secondly, the pandas data frame function $info()$ is used in order to quickly check which data types are available and if data is missing. All columns are of data type integer. There are no categorical columns. Therefore, no column needs to be encoded first. When looking at the missing values, then it gets clear that no feature is missing any values. As a next step, a histogram is plotted for all numerical features. This helps to get first insights into the distribution of the features. Figure 1 shows all histograms. The distributions are looking skewed, so a log transformation is used in order to reduce the skewness. The distribution of the transformed features can be seen in the second row of Figure 1. As one can see, the distributions are now more Gaussian and less skewed than before. As a next step, the correlation heat map is plotted in order to check the correlations between the features. Figure 2 shows the resulting correlation heat map. There are some features with a higher correlation. However, they are not high enough in order to remove some of the features. The highest correlation can be found between the features *Income* and *Occupation*, with a correlation value of 0.7. This makes totally sense, because a customer with an occupation level of two is more likely to have a higher income than a customer with an occupation level of one or even zero. Finally, the standard scaler from scikit-learn is used in order to scale the data.

4 Methodology

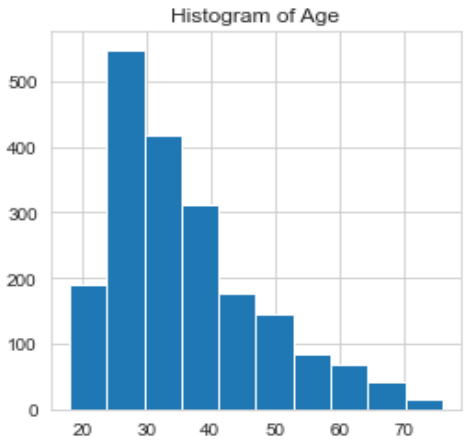
As stated in the introduction, the underlying machine learning task is a clustering task and the K-Means clustering algorithm is used. The K-Means algorithm requires the number of clusters as input parameter. It is not clear here what number of clusters to use. Therefore, the optimal number of clusters needs to be found first. For this purpose, a K-Means model is trained for one to ten clusters and the inertia value is stored for each trained model. In the end, the elbow method is used on the inertia curve in order to find the best suited amount of clusters. Figure 3 shows the resulting inertia curve. The elbow point is not clearly visible, but four clusters seem to be the most promising elbow point. Therefore, the final K-Means model is trained using four clusters.

5 Results

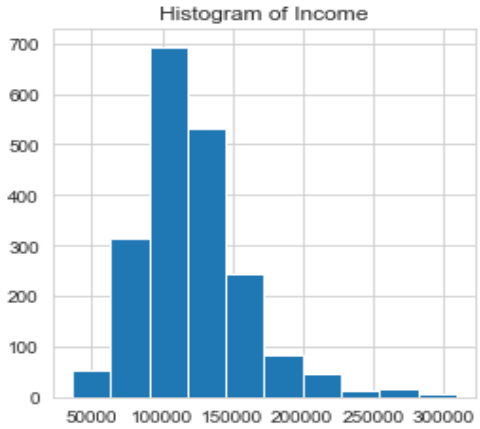
Figure 4 shows the average values for each cluster and each feature. The average age gets higher for a higher cluster number. In addition, the income is the largest for the last cluster, which also seems to include more older customers. This makes sense, because there are less younger customers who are already managers or having their own successful business.

6 Future Work

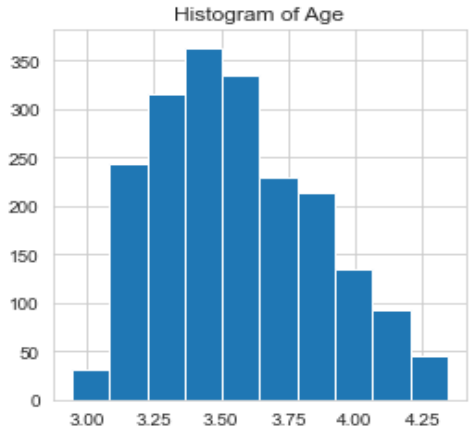
This section briefly discusses some future work that could be done. In this report, only the K-Means algorithm is used. In a future work, also other clustering algorithms could be used in order to check if they achieve better results. In addition, the results could be discussed in more detail with a better and more suited clustering algorithm.



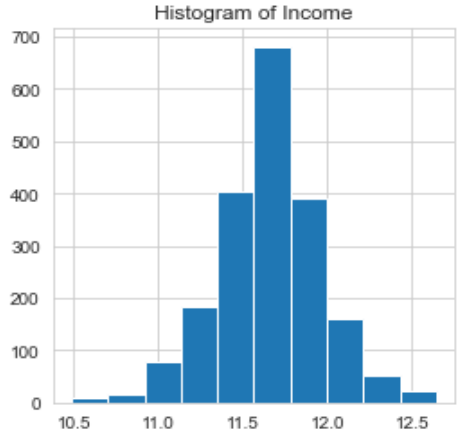
Histogram of age before applying the log-transformation



Histogram of income before applying the log-transformation



Histogram of age after applying the log-transformation



Histogram of income after applying the log-transformation

Figure 1: Histograms of numerical features of customer data set before and after applying the log-transformation.

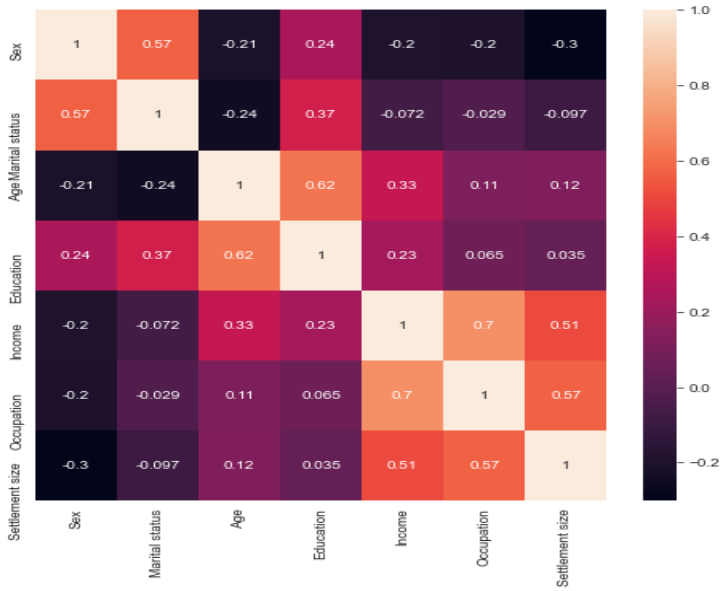


Figure 2: Correlation heat map of all features and the target variable.

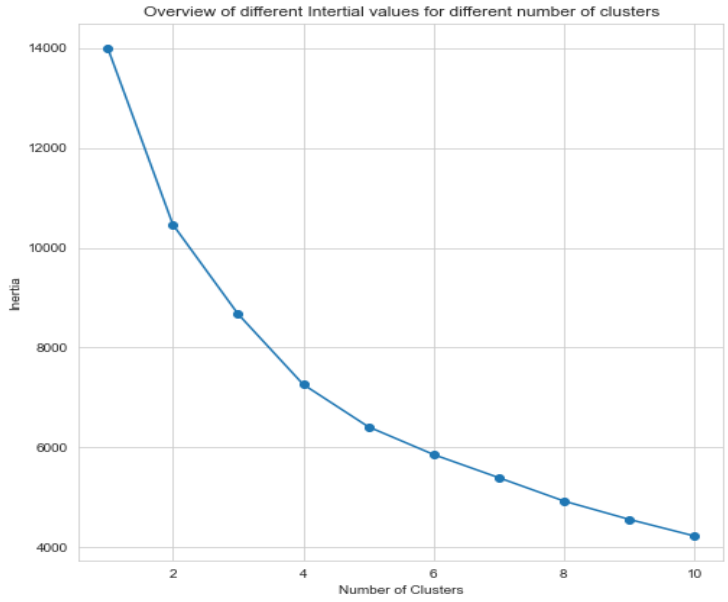


Figure 3: The inertia curve for the different K-Means models trained with different number of clusters.

Cluster Nr	Sex	Marital status	Age	Education	Income	Occupation	Settlement size
0	0.730841	0.988785	27.906542	1.000000	121324.493458	1.071028	0.833645
1	0.617391	0.431304	32.631304	0.885217	87274.460870	0.085217	0.006957
2	0.011785	0.015152	36.792929	0.680135	137826.106061	1.185185	1.244108
3	0.543919	0.699324	54.966216	2.121622	151853.756757	0.996622	0.976351

Figure 4: The average values of all features for each cluster.

References

[1] Dev Sharma. *Customer Clustering*. 2021. URL: `https://www.kaggle.com/dev0914sharma/customer-clustering` (visited on 07/27/2021).

Variable	**Data type**	**Range**	**Description**
ID	numerical	Integer	Shows a unique identificator of a customer.
Sex	categorical	0,1	Biological sex (gender) of a customer. In this dataset there are only 2 different options: • 0 male • 1 female
Marital status	categorical	0,1	Marital status of a customer: • 0 single • 1 non-single (divorced / separated / married / widowed)
Age	numerical	Integer	The age of the customer in years, calculated as current year minus the year of birth of the customer at the time of creation of the dataset.
Education	categorical	0,1,2,3	Level of education of the customer: • 0 other • 1 high school • 2 university • 3 graduate school
Income	numerical	Real	Self-reported annual income in US dollars of the customer.
Occupation	categorical	0,1,2	Category of occupation of the customer: • 0 unemployed / unskilled • 1 skilled employee / official • 2 management / self-employed / highly qualified employee / officer
Settlement size	categorical	0,1,2	The size of the city that the customer lives in: • 0 small city • 1 mid-sized city • 2 big city

Table 1: An overview of all features available in the customer clustering data set.