Lorena Maria Zvunka

Report Assignment 3

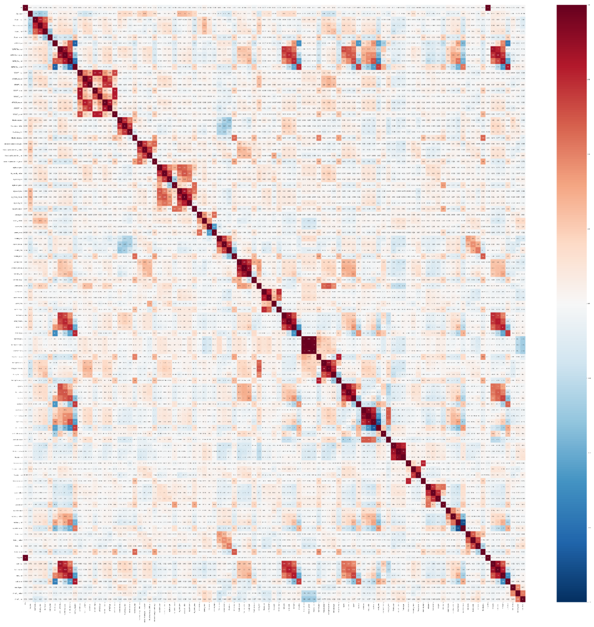
K-means clustering

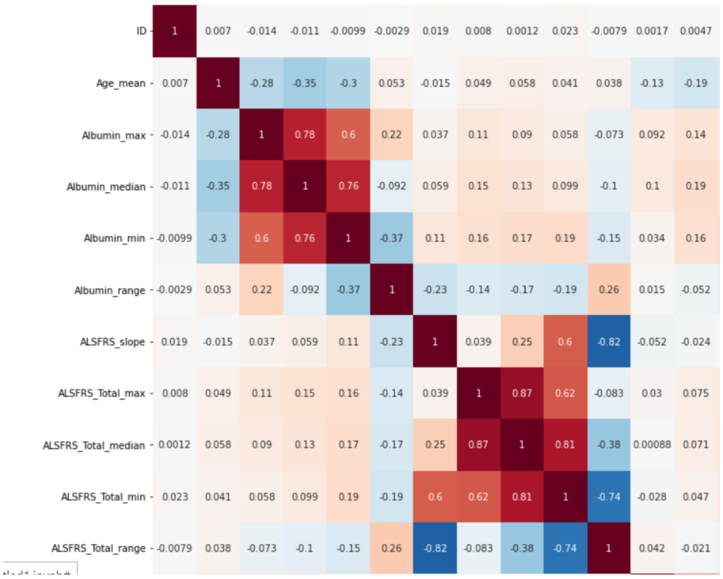
i) Load and prepare the data.

We begin by loading the data using pandas read.csv function. Once the data is read, we can begin to understand what it represents. The given dataset contains 101 columns, representing the features we will later be working with. With the use of the .head function we can take a look at the first rows and notice that the data varies a lot. Some columns have values close to 0 while others have values over 100. This proves the need for this data set to be normalized and adjusted.

ii)Perform summary and preliminary visualization

Seeing as we are dealing with a really large dataset it can be hard to understand or even see all of the features. Therefore, it is understandable that when trying to analyze the data not all of the columns will be equally important. We can visually see the importance of each feature in relation to the rest with the use of a heatmap.



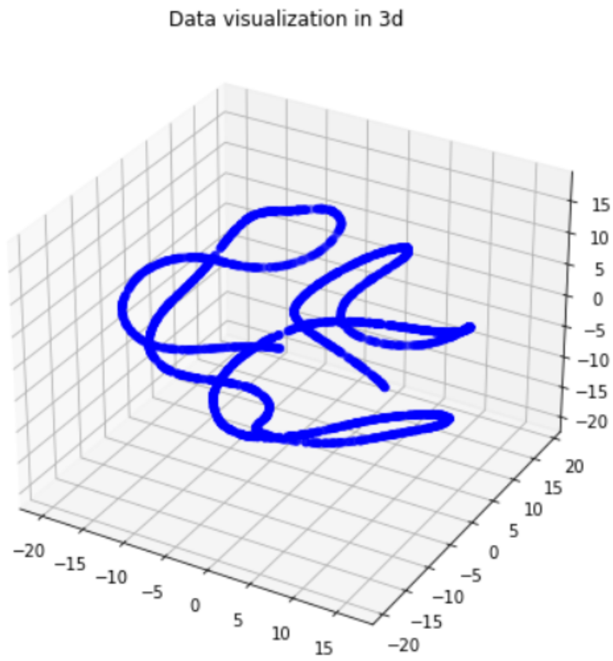


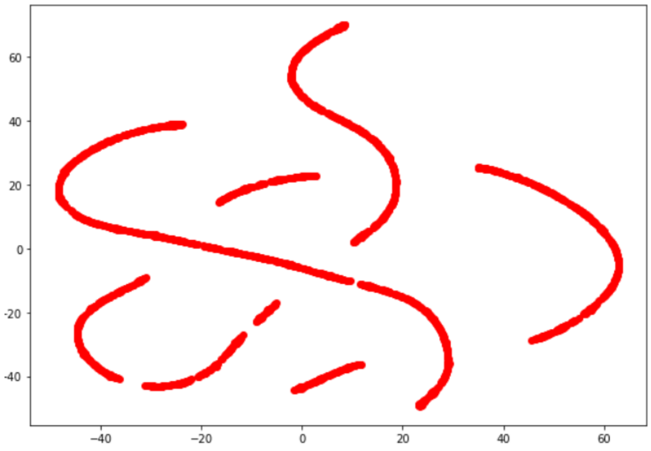
Because of the large number of columns we are dealing with this might be a little confusing, however a simple zoom in makes it easy to read and allows for a better understanding of our dataset. This can be a very useful representation when deciding which features to keep and which to delete.

iii) Normalize the data and analyze the result of normalization

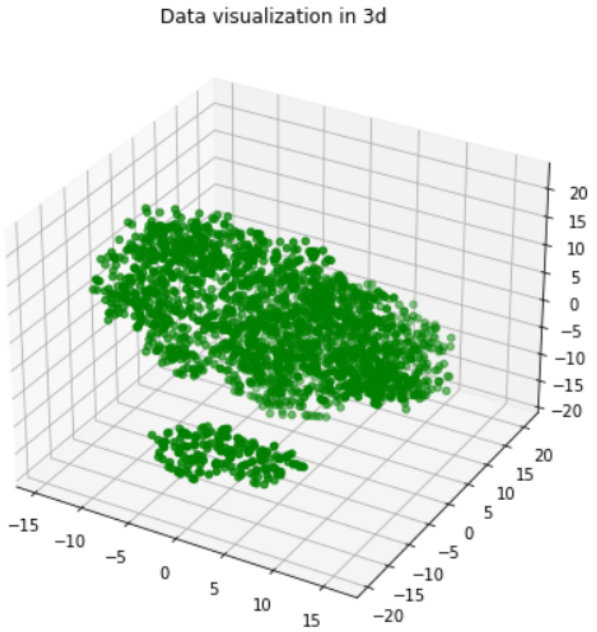
In order to actually visualize such a highly dimensional dataset I chose to use t-SNE to reduce its dimensionality to 2d and 3d. I started by making a function, prepare_tsne take would take as arguments an integer number, representing the dimension to which we want the data converted, and as the second parameter, the actual data we want transformed. This function returns the transformed data.

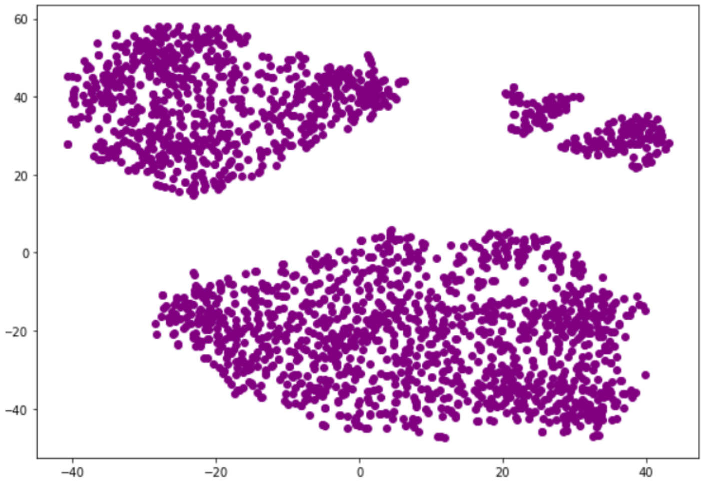
So that we can better see the major difference caused by normalizing data, we start first by representing the raw dataset.





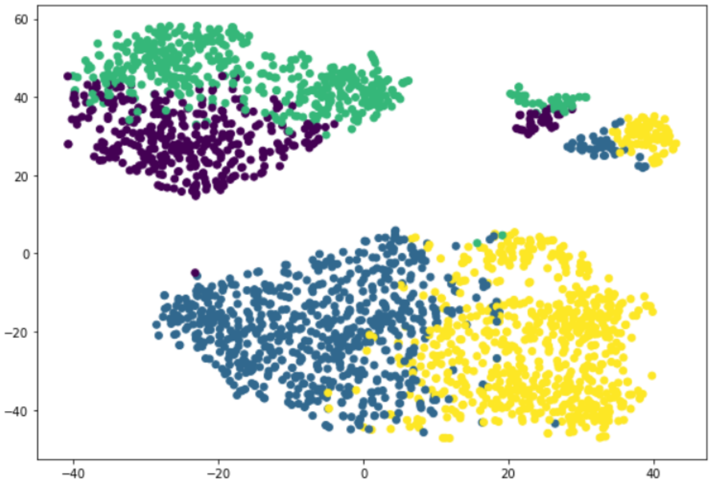
The method chosen to normalize the data was a MinMax scaler. By using this we ensure the data will be more evenly distributed and reduce the risk of dealing with unbalanced information. This brings all the values of the features to be between 0 and 1, thus eliminating the huge differences that existed before. Once this process is done, our data, represented visually, looks like this.

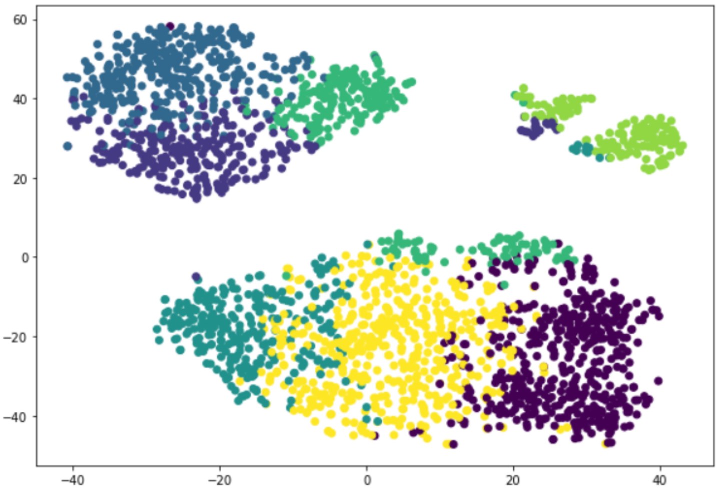




iv) K-means

In the following we trained a kmeans model. The difference between the result of the algorithm when changing the number k are pretty obvious just by taking a look at some plots.

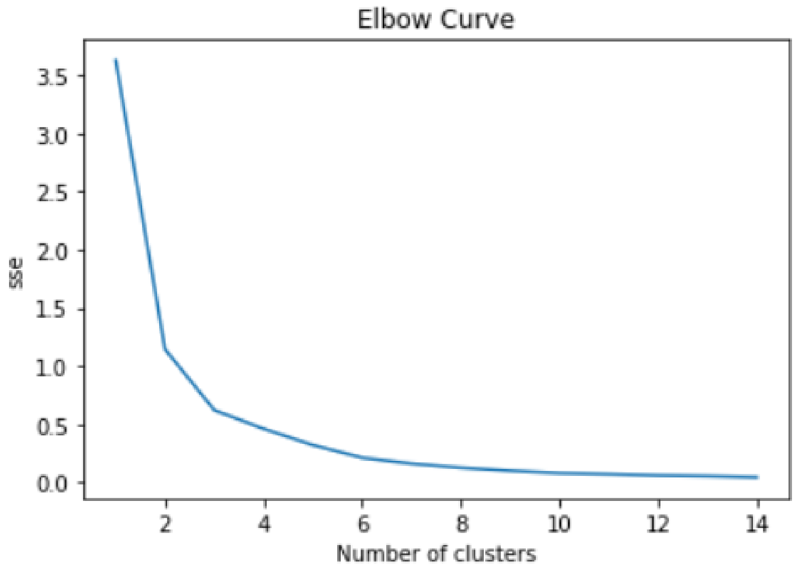




The figure on the left is a representation of the data in 4 clusters while the one on the right was assigned into 7 clusters. Besides the visual difference we can see, there is also one in the performance of the algorithm itself. Inertia measures how well a dataset was clustered and the inertia for having 4 clusters is 0.462, bigger than the one we get for 7.

However, the changing of the other variables do not seem to affect the outcome.

In order to best determine the number of clusters we need, I decided to use the Elbow Curve method. The range I considered was from 1 to 15 and after fitting the algorithms for each of those the resulting graph was this.



From this, we can determine that the best choice for k would be 3.