CSE4062

Introduction to Data Science and Analytics

2023-2024 Spring

Group 5

Company Bankruptcy Prediction

Delivery 2 Report

Ahmet Onat Özalan - 150118054 - Computer Engineering

Email: o171141@gmail.com

Selin Zeydan - 150322823 - Industrial Engineering

Email: selinzeydan9@gmail.com

Mediha Ecem Polat - 150820822 - Bioengineering

Email: medihaecempolat@gmail.com

Fırat Bakıcı - 150120029 - Computer Engineering

Email: firat143@gmail.com

Kardelen Kubat - 150118056 - Computer Engineering

Email: kardelenkubatcse@gmail.com

Berfin Ege Yarba - 150321036 - Industrial Engineering

Email: berfinegeyarba@gmail.com

Osman Buğra Göktaş - 150119565 - Computer Engineering

Email: osmanbugrag@gmail.com

Project Description

K-means:

1) Feature Selection

In order to speed our model execution and be able to make better analysis of our results, we used a “feature selection” method to reduce the number of our features. We picked 10 features from the original features. 5 of them are the features that are most positively correlated with the class label in our dataset (named “Bankrupt?”), and the other 5 are the features that are most negatively correlated with the class label. The features we use are given below:

```
Debt ratio %
Current Liability to Assets
Borrowing dependency
Current Liability to Current Assets
Liability to Equity
Net Income to Total Assets
ROA(A) before interest and % after tax
ROA(B) before interest and depreciation after tax
ROA(C) before interest and depreciation before interest
Net worth/Assets
```

Figure 1. The features that we used.

2) Dropping the Class Label from Dataset

After the feature selection, we dropped the class label, “Bankrupt?”, from the dataset to work on unlabeled data.

3) Determining Number of Clusters with Elbow Method

We used the K-Means clustering algorithm in this delivery to cluster the instances of our dataset. In the K-Means algorithm, the number of clusters to be created by the algorithm is given as input to the algorithm. To determine the number of clusters, we used the “elbow method”. We run the K-Means algorithm for the cluster numbers = 1, 2, …, 50. The algorithm is run 10 times for each number and in each run, it picks

initial random points. Then, it returns the best result out of 10. Then, we calculated the graph of scores based on these results and saw that the number 10 is close to the elbow of the curve. So, we selected the number 10 as the number of clusters to be used.

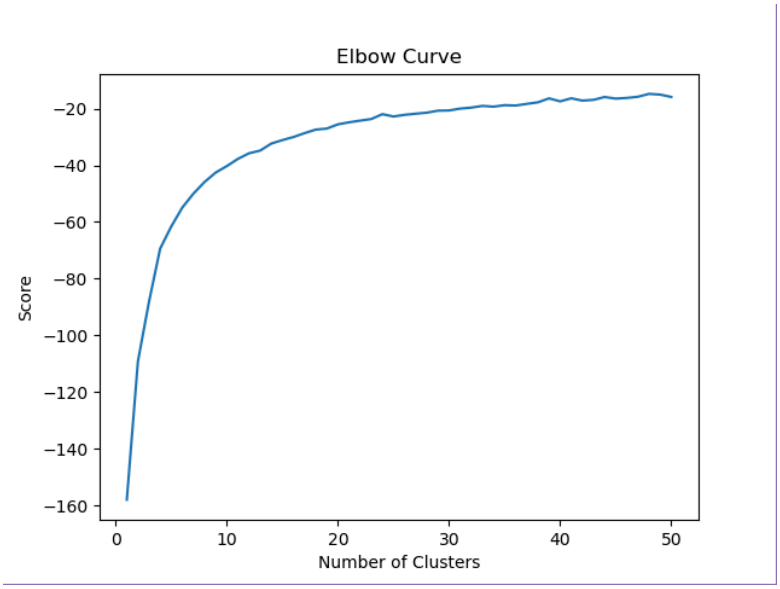

Figure 2. The graph of the "elbow method".

4) Running the K-Means Algorithm

After the selection of the number of clusters, we run the K-Means algorithm. The algorithm is run 100 times, and the initial cluster centroids are picked by random in each run. The result of the algorithm is the best result out of the results of the 100 runs.

5) Visualizing and Evaluating the Results

5.1) Visualizing the Clusters in a 2-Dimensional Space

To be able to visualize the data, we applied Principal Component Analysis (PCA) to our dataset and the set of cluster centroids we got from the K-Means algorithm. Our reason to apply PCA is to reduce the dimensions of the result to 2. That way, we will be able to plot the results in a 2-dimensional graph. The graph we got after applying these steps is given in Figure 3.

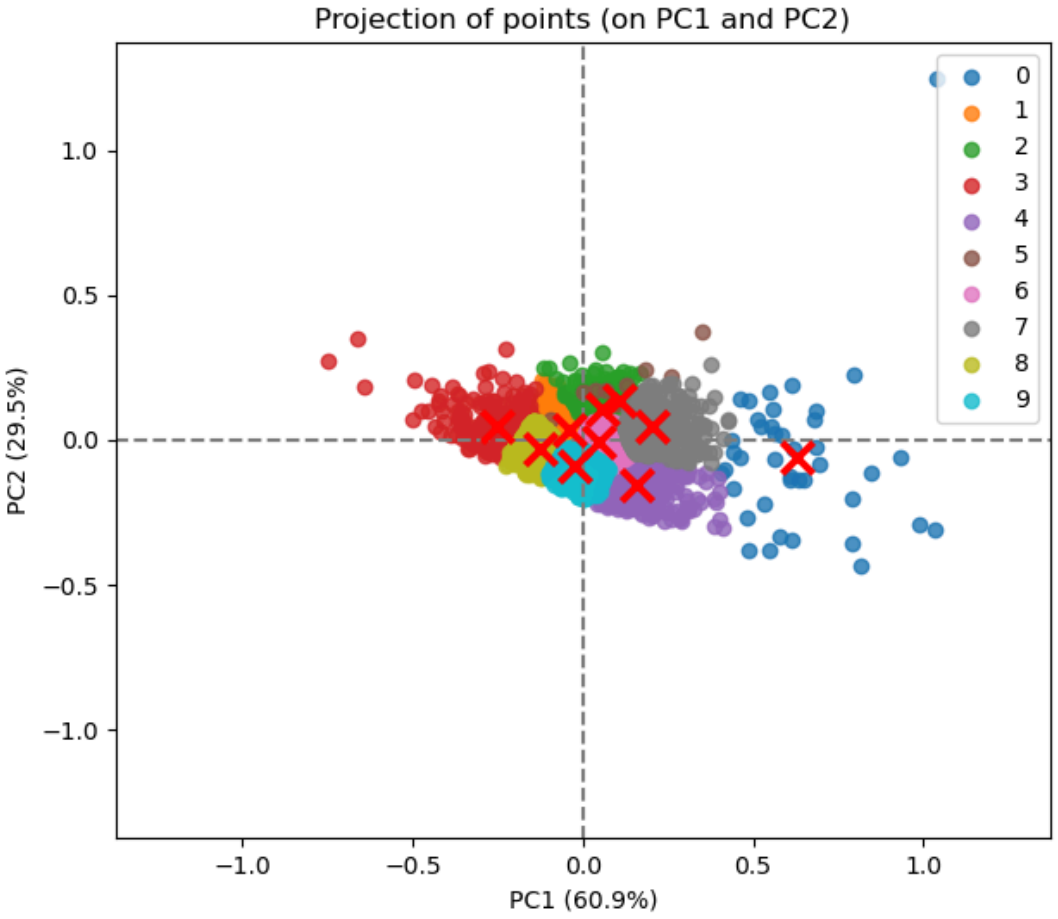

Figure 3. The visualization of the clusters and their centroids after applying PCA and reducing them to 2 dimensions.

5.2) Average Values of Features in Each Cluster

Also for each of the 10 features we used, we calculated the average value (centroid) of that feature in each of the clusters created by the K-Means algorithm. This method provides some good insights about our dataset. The visualization of this method is given in Figure 4.

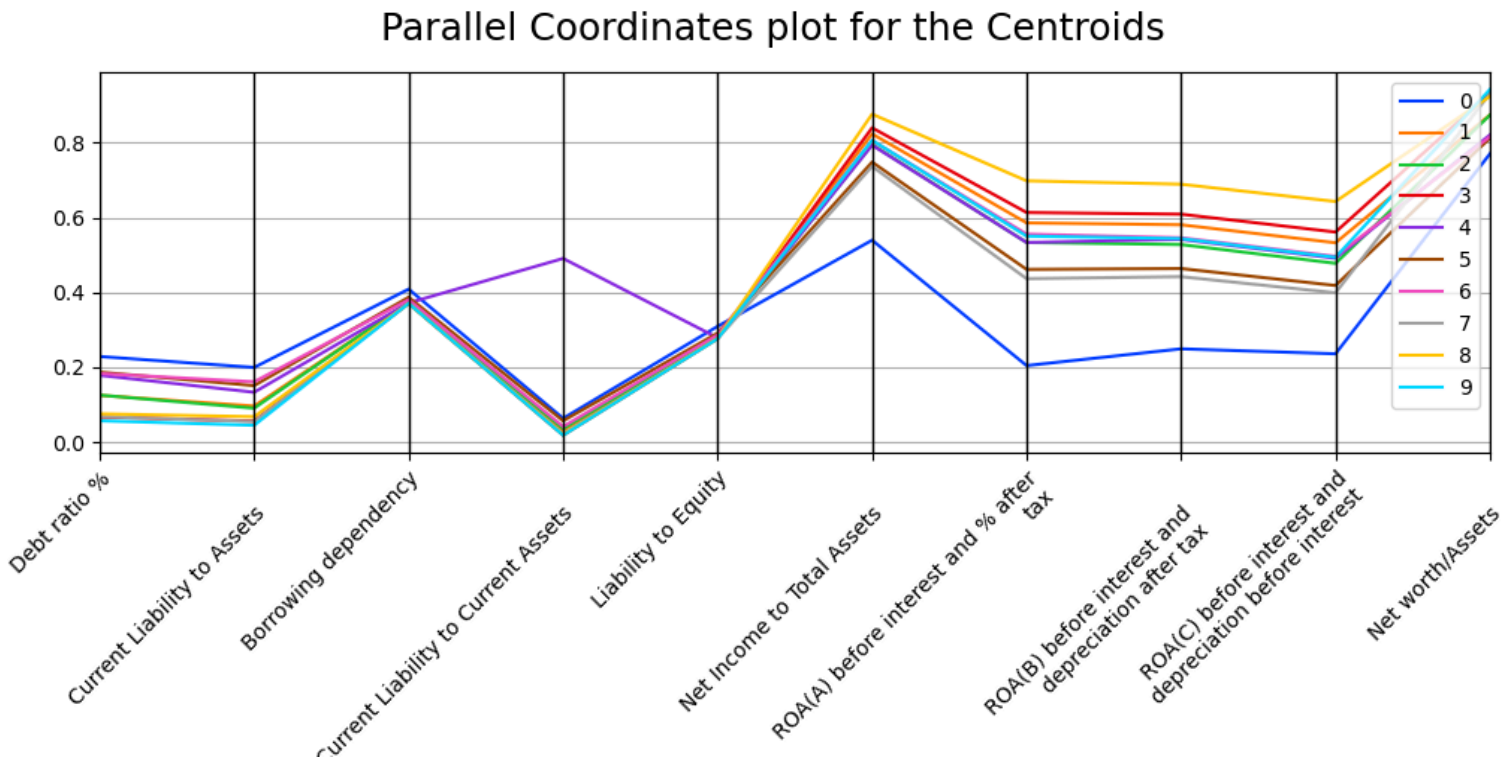

Figure 4. The average values of the features in each cluster.

The insights we gained from the parallel coordinates plot:

- In Cluster 4, the value of the feature “Current Liability to Current Assets” seems to be significantly higher than the other clusters.
- In Cluster 0, the values of the features “Net Income to Total Assets”, “ROA(A) before interest and % after tax”, “ROA(B) before interest and depreciation after tax”, “ROA(C) before interest and depreciation before interest” and “Net worth/Assets” seem to be significantly less than their respective values in the other clusters.
- The values of the features “Borrowing dependency”, “Current Liability to Current Assets” and “Liability to Equity” show very little variance between clusters.
- The first 5 features seem to be positively correlated between themselves. The last 5 features seem to be positively correlated between themselves. The first

5 features and the last 5 features seem to be negatively correlated between each other.

5.3) Number of Instances in Each Cluster

Another metric we used to evaluate the clusters is the number of instances in each cluster. This information is given in Figure 5.

```
Cluster 0: 41
Cluster 1: 1330
Cluster 2: 946
Cluster 3: 331
Cluster 4: 290
Cluster 5: 14
Cluster 6: 1351
Cluster 7: 373
Cluster 8: 944
Cluster 9: 1199
```

Figure 5. The number of instances in each cluster.

Clusters 0 and 5 have very few instances associated with them. The instances in these clusters might be outliers.

5.4) Silhouette Score of K-Means

Our last metric of evaluation is Silhouette score. Its values range between -1 and 1. The Silhouette score provides a measure of how close each point in one cluster is to points in the neighboring clusters. The values near +1 means more well-defined, far-away clusters. The values near 0 means the clusters are very close to each other. The values near -1 means the points might be assigned a wrong cluster. The Silhouette score for the result of our K-Means algorithm is given below:

```
Silhouette score:  0.2656970807116298
```

As it can be seen, it is greater than 0 and not very close to 0. So, it tells us that the result of our K-Means algorithm is decent.

DBSCAN:

1) Feature Selection

In order to enhance the efficiency of our model execution and facilitate more effective analysis of our results, we employed a "feature selection" method to reduce the number of features utilized by the Dbscan algorithm. From the original set, we specifically selected two features for inclusion. This decision was based on our examination of the dataset's characteristics, where these particular features demonstrated a wide range of distributions. By focusing on these, we aimed to improve the clarity of clustering within our dataset, thus enhancing the overall performance and interpretability of our model.

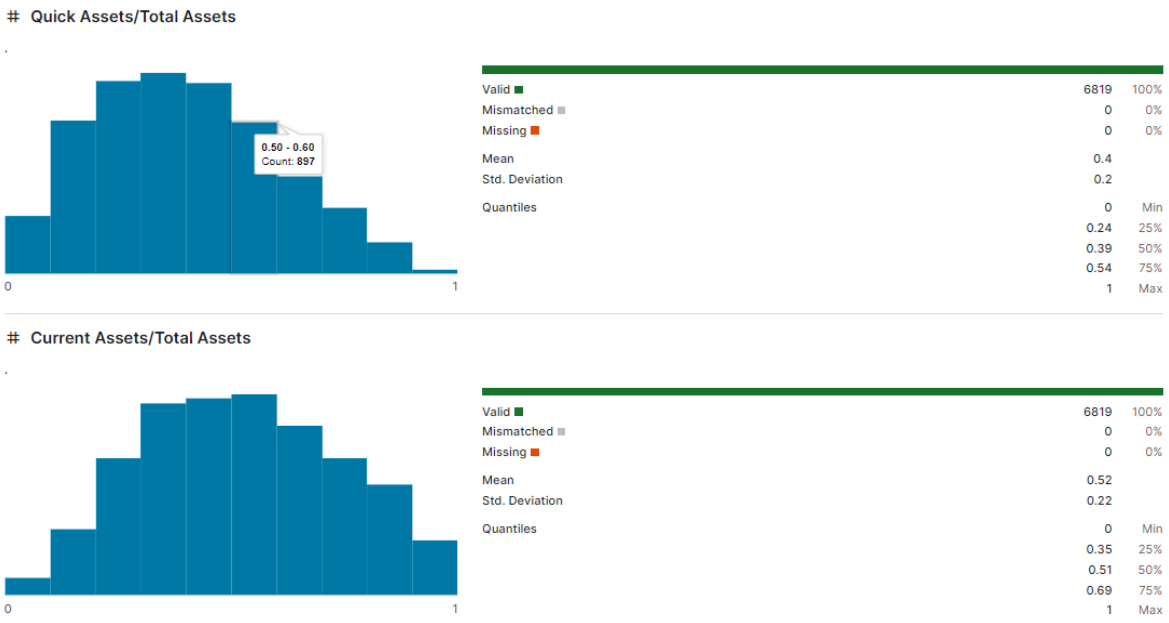

Figure 6. Distribution of selected features

3) Determining eps value with k- nearest neighbors Algorithm

We focus on optimizing the DBSCAN clustering approach by establishing the correct epsilon (eps) value. This optimization process began with extracting and analyzing the k-nearest neighbors for each data point, specifically using k=4, for the two

financial ratios: 'Quick Assets/Total Assets' and 'Current Assets/Total Assets'. After calculating the distances to the fourth nearest neighbor for each data point in these columns, we exported this detailed neighbor information into an Excel sheet for further scrutiny. Simultaneously, we utilized Matplotlib to graphically plot these distances, aiming to identify an elbow, or breaking point, in each plot. This elbow point serves as a crucial indicator for setting the optimal eps value, as it represents a balance point beyond which the increase in distance signifies a diminishing return on cluster homogeneity. This graphical method, complemented by our data export, ensures a robust analysis by visually and quantitatively assessing the distribution of k-nearest distances, thereby guiding our selection of an appropriate eps value for effective data clustering.

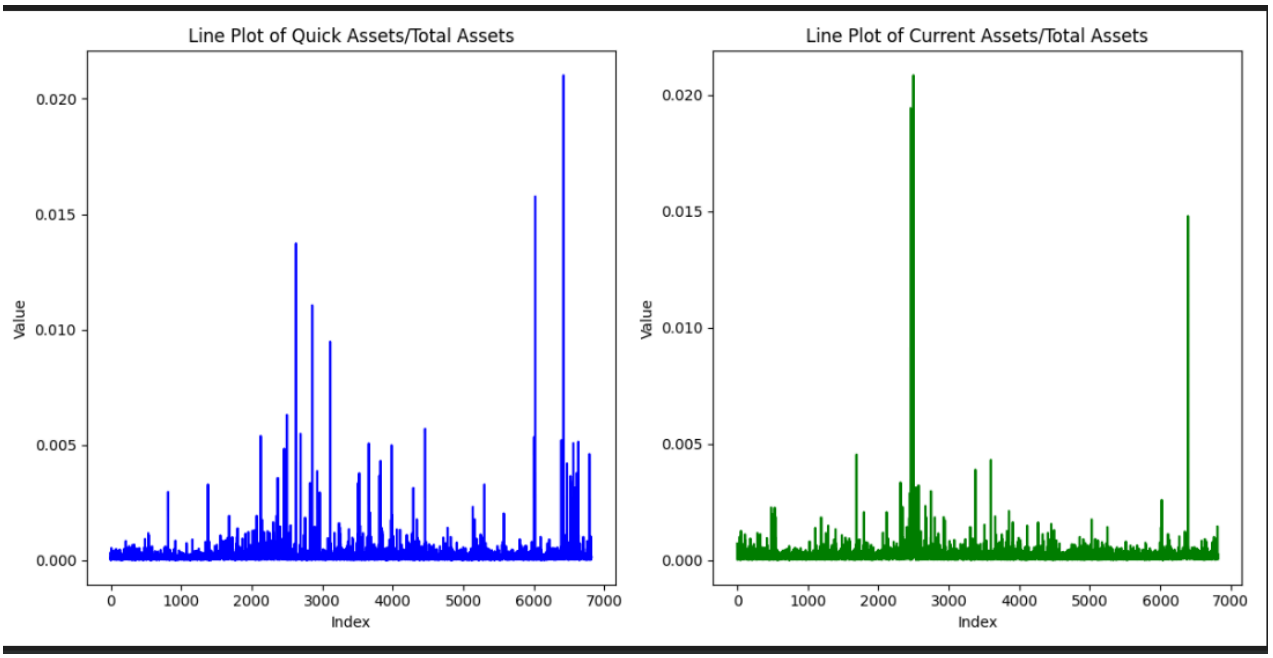

Figure 7. K-neighbor distances for selected features for k=4

We implemented a method to calculate and recommend eps values for DBSCAN clustering using Python. The process began by loading the distances of the k-nearest neighbors from an Excel file named 'combined_k_nearest_neighbors.xlsx'. This dataset contains computed distances to the fourth nearest neighbor for various financial ratios, which were previously identified as crucial for our clustering analysis.

Using the calculated average distances as reference eps values, we conducted several experiments to refine and identify the optimal eps value for DBSCAN clustering.

4) Running the Dbscan Algorithm

Initially, the dataset is loaded from a CSV file, and the recommended eps values for each column are loaded from an Excel file. The script specifically targets two columns. For each column of interest, the code retrieves the corresponding eps value from the Excel file. If the eps value is available and non-zero, the column data is preprocessed by removing any missing values and scaling the data using StandardScaler. The DBSCAN algorithm is then applied to the scaled data, using the eps value and a minimum sample count of 5 to determine clusters. For each identified cluster, statistical calculations such as mean, variance, and z-scores are computed and written into a text file, dbscan_cluster_stats_results.txt, for each cluster within the selected columns. If an eps value is not found or is zero, the script notes that no clustering was performed for that column. The process concludes by outputting the results into a text file, providing a detailed account of the clustering outcomes for each column.

Analysis of Clustering Results on Financial Ratios

In this section of the report, we present an evaluation of the clustering results for two key financial ratios:

"Quick Assets/Total Assets" and "Current Assets/Total Assets"

The analysis involved using the DBSCAN clustering algorithm to categorize the data into multiple clusters based on their similarity.

Quick Assets/Total Assets

The dataset was segmented into 108 distinct clusters.

The clusters' mean values ranged significantly, mostly between 0.05 and 0.87.

This variation suggests a diverse dataset where certain clusters display a higher concentration of values, potentially indicating distinct financial health or operational characteristics among the grouped entities.

The outlier cluster, labeled as Cluster -1, demonstrated a mean value of 0.6609.

This high average could be indicative of atypical financial structures or exceptional cases differing significantly from the mainstream data.

Current Assets/Total Assets

A total of 157 clusters were identified for this financial ratio, reflecting a complex and varied dataset.

The mean values of these clusters were observed between 0.1 and 0.98. Clusters with particularly high mean values suggest the presence of entities with unusually high proportions of current assets, which could reflect specific industry practices or liquidity statuses.

Similarly, Cluster -1 showed a mean value of 0.4087, highlighting it as containing outlier data which may require further scrutiny to understand the underlying business contexts.