Ibrahim Munir
CS484-001

HW4 – Clustering

Miner ID: imunir2

Rank/V-Scores: Part 1 – 36/0.95   Part 2 – 335/0.46

Introduction

To accurately classify the iris and image datasets, k-means clustering was implemented to detect similar datapoints such that they could be grouped together. When conducting k-means clustering, silhouette score was used as an internal evaluation metric to judge cohesion and separation of the clusters and their points. In addition to the silhouette score, the data itself was plotted. Such a plot allowed for easy visualization of the data to check the correctness of the clustering.

Initial implementation

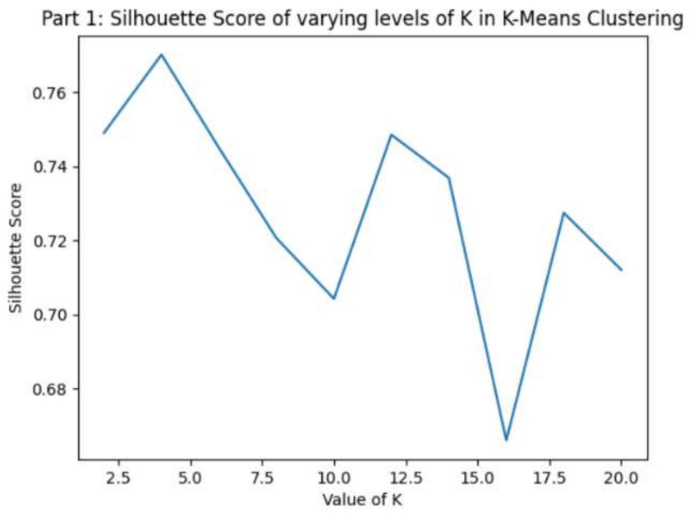
To begin the process of implementing k-means clustering, a simple implementation was tested to see the initial results of the clustering approach. Starting centroids were selected randomly, leading to varying results at the end of each run. Furthermore, centroids were chosen as the arithmetic means of their respective clusters, not as an actual point in the cluster. With the arithmetic mean approach, k-means clustering was able to capture a better picture of each cluster.

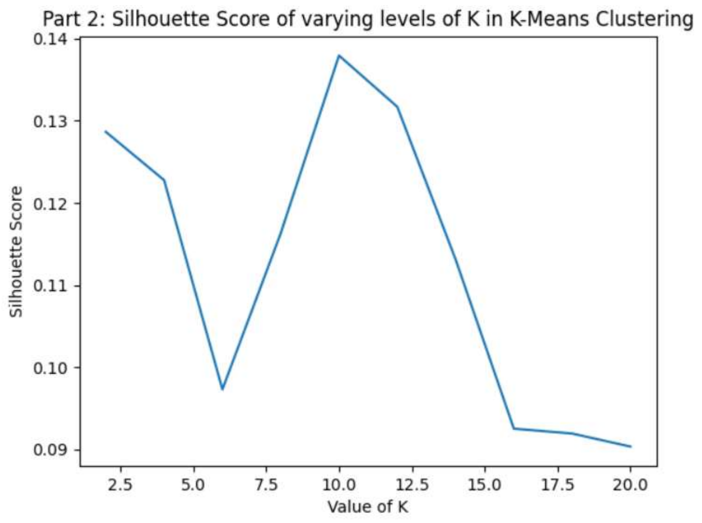
The remaining part of the k-mean clustering algorithm was left unchanged up until this point. The actual algorithm was implemented as a do-while loop, ensuring that k-means was ran at least once. The terminating condition for the loop was also carefully put in place to terminate the algorithm only when centroids remained completely the same as the previous run. Alternatives, such as including a tolerance for change within this condition, were considered to prevent algorithm from being stuck within an infinite loop in the case that centroids continued to undergo minor changes. However, in the end, these approaches never went into practice since such cases were not encountered.

Optimizing the algorithm

After going through the initial implementation of the k-means clustering algorithm, certain areas of improvement were identified. Specifically, the manner in which the starting centroids were initialized and how the distance between points was calculated when deciding the cluster for each point. Last but not least, an optimal k had to be found through the use of an internal evaluation metric.

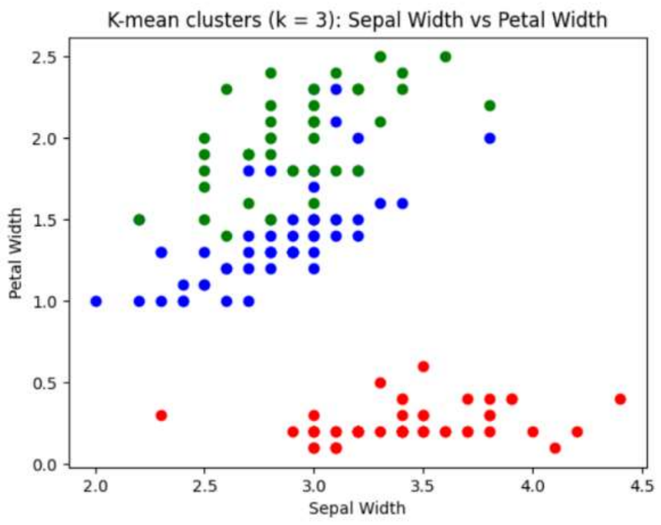
Although, it was known that the k values of 3 and 10 were best suited for the iris and image datasets, respectively, an internal evaluation measure was used to examine the success of other k values and to confirm the belief that the k values of 3 and 10 were indeed, the best hyperparameters in these certain situations. Namely, the silhouette score was used for this purpose. Silhouette score was chosen because of its holistic approach when looking at a clustering. The ability of the silhouette score to take into account the cohesion and separation of the clusters made it seem like the most effective and versatile metric that could be used to grade my clusters produced by k-means clustering. K-values of 2 to 20 were tested in steps of 2 to determine the best k-value to use when clustering the iris and image datasets. The results are shown below:

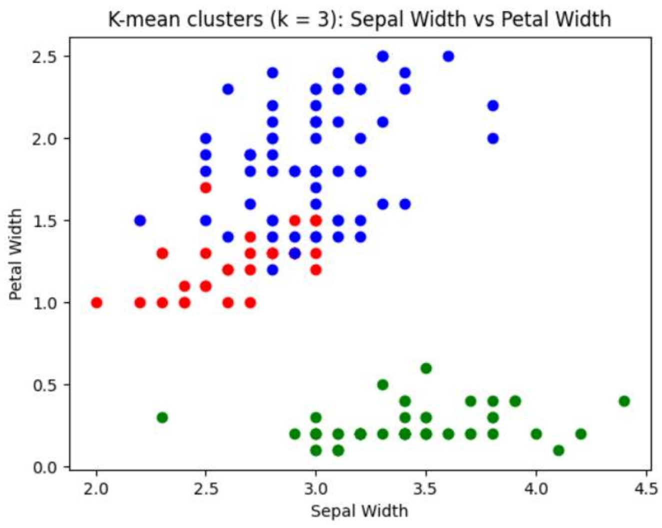


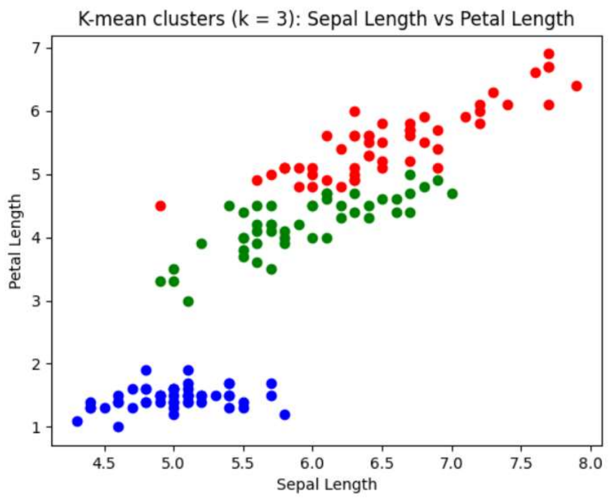


The graphs constructed when evaluating various values of k with the silhouette score verified the belief that a k of 3 and 10 were best suited for the iris and image datasets. In the first graph, the silhouette score showed a gradual increase until after the k value of 3 was passed, after which, the silhouette score began to sharply decrease with occasional spikes. In the case of image dataset, there was another increase as the x approached the k value of 10. Similarly, the silhouette score began to decline once the k value of 10 was passed. In both scenarios, the peak of the graphs occurred at the k values of 3 and 10, respectively.

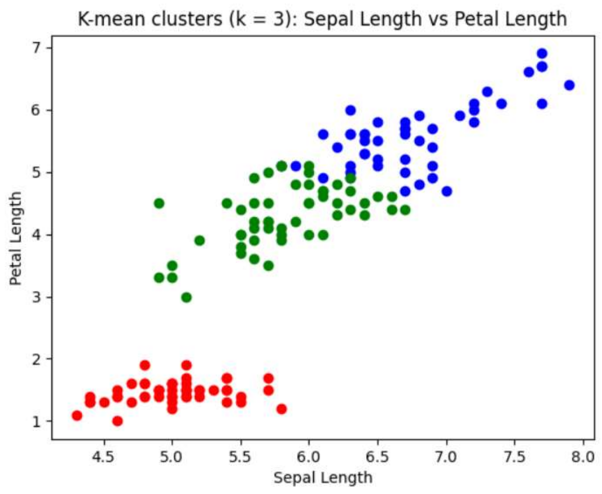
After the k values were decided upon, the distance metric used to calculate distance between points was the next to be evaluated. Two main metrics were looked at: euclidean distance and cosine similarity. To begin the evaluation of the distance metrics, cosine similarity and euclidean distance were first tested on the iris dataset. For each distance metric, two plots were made to model sepal length against petal length and sepal width against petal width.





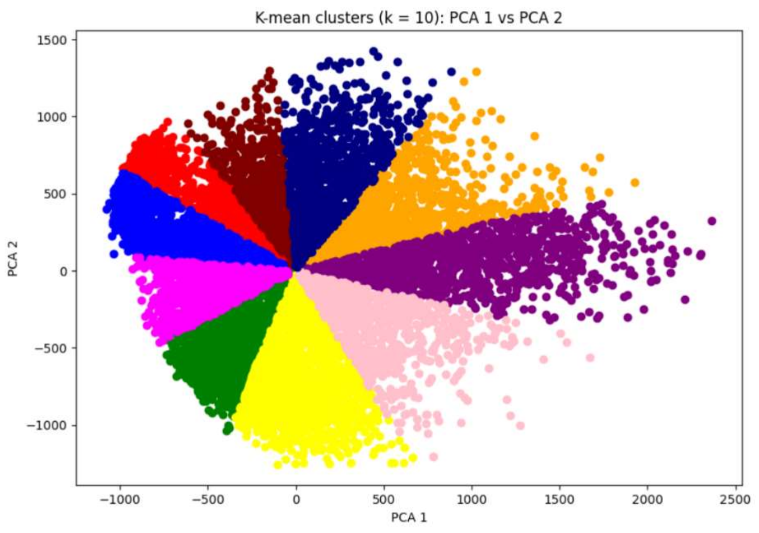


Cosine similarity

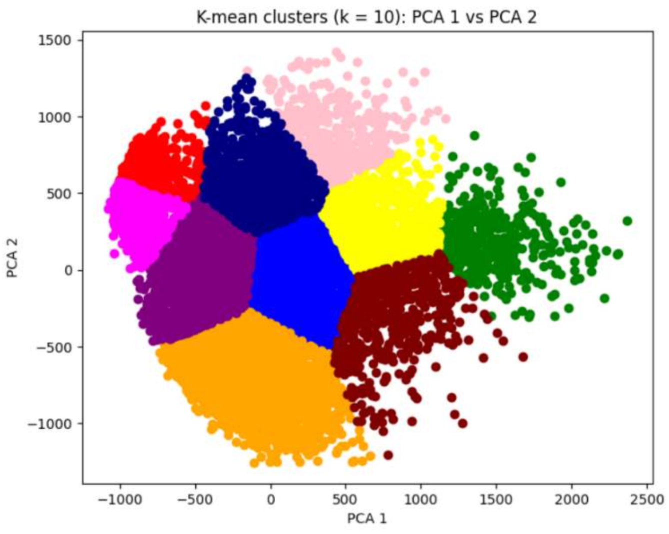


Euclidean distance

After examination of each of the plots, the clusters formed using cosine similarity showed less overlap when compared to the clusters formed using euclidean distance. To take a step further, when tested with V-measure, the euclidean distance clusters ended up with a score of 0.73 compared to the score of 0.95 with cosine similarity. These experiments quickly demonstrated that cosine similarity was probably the best distance metric to use with the k-means clustering algorithm. In order to further cement this belief, the clusters formed with the image data were plotted with euclidean distance and cosine similarity.



Cosine similarity



Euclidean distance

While going through the process of constructing the plots for the image data, it became obvious that dimensionality reduction was needed. The image vectors were too large to be able to plot, and as a result, PCA had to be utilized to shrink the dimensionality of the data. Through the use of PCA, the data was put into such a form allowing it to be plotted without losing the essence of the data. With the data now visualized, it became obvious once again that cosine similarity was the better distance metric to use with the k-means clustering algorithm. The image data was sorted into much more organized and cleaner clusters with the cosine similarity metric versus the jumbled clusters created when using euclidean distance.

Finally, the manner in which the centroids were initialized with k-means clustering proved to be another crucial part of the design and implementation process. As mentioned earlier, centroids were first initialized at random. However, such a process was ineffective and had to be replaced because of its volatility. The first idea that came to mind was K-means++. Through K-means++, centroids would most likely start off by being far away from each other. Such an arrangement would give way to better separation between the clusters. If centroids started out close to each other, overlap between clusters would be more likely, causing an unfavorable outcome. After using K-means++ in my algorithm, my highest V-scores with the iris and image datasets were observed since more consistent centroid initializations occurred through the use of K-means++.

Conclusion

While certain improvements were made to the algorithm throughout its implementation and experimentation with the iris and image datasets, the original k-means clustering algorithm stayed intact for the most part. The different optimizations allowed the algorithm to better adapt to the various outliers in both datasets and the high dimensionality present in the image data. The silhouette score served as a good measuring stick for the cluster correctness along with the scatterplots backing up the numbers seen in the evaluation process. After all the deliberation, I was able to conclude that the best clusters were formed when a combination of K-means++, cosine similarity, and an optimal K were used in K-means clustering.