PRA3021 Assignment 1

Tom Kores Lesjak

September 2024

1 Plotting the Data

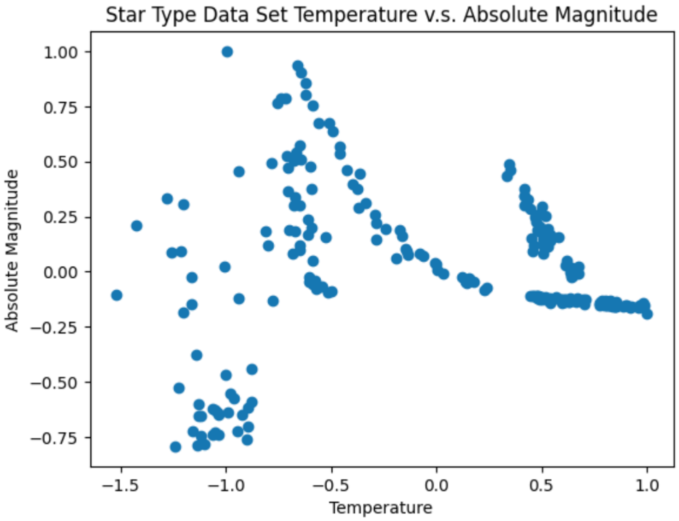

Figure 1: Star type data set plotted temperature against absolute magnitude

The plot of the star-type dataset is shown in Figure 1. It can be seen that the data forms distinct clusters. Each point in the plot represents a star, with its temperature on the x-axis and absolute magnitude on the y-axis, illustrating key stellar properties. These clusters likely correspond to different star types, such as main-sequence stars, giants, and white dwarfs, which exhibit unique temperature and brightness characteristics, and therefore are sorted into distinct clusters. The mean temperature and absolute magnitude is $-7.113 \cdot 10^{-7}$ and $1.4833 \cdot 10^{-7}$ respectively. If the dataset is spread out across multiple clusters, the mean will typically fall somewhere between the clusters, often in an area where there might be very few or no actual data points. In data sets with clear clusters the mean becomes less useful as a descriptive statistic, as it looks at the global measure not providing an insight into the local structure of the data.

Yet if the data is separated into clusters, then the mean is useful when only considering individual clusters at a time.

2 Function and Algorithms

2.1 Square Distances Function

To create this function the definition of Euclidean Distance was used and is defined as follows;

$$r(U, v) = \Sigma_{i=1}^{n}(U_i - v_i)^2 \tag{1}$$

It takes as input a matrix U representing multiple data points and a matrix v representing the centers for clustering. The function computes the squared distance between each point in U and each point in v and returns a distance matrix.

2.2 K-Means Algorithm

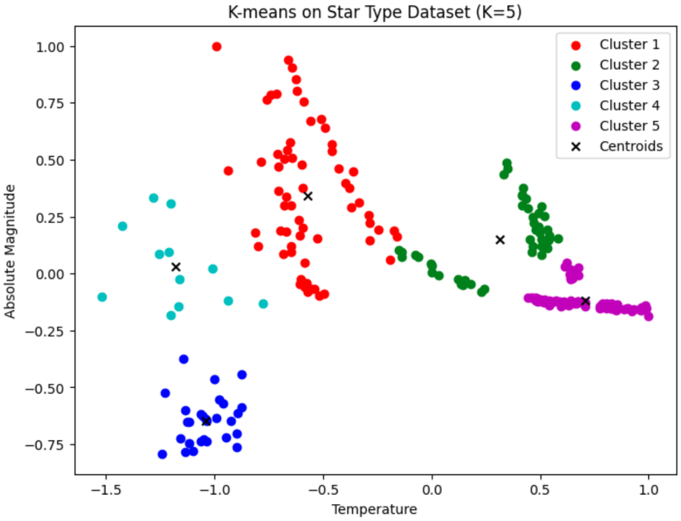

Figure 2: K-Means clustering preformed on star type dataset with K=5

The K-means algorithm is an iterative clustering method used to partition a dataset into K distinct clusters. This algorithm begins by calculating the distance between data points and an initial set of cluster centers. Each data point is then assigned to the nearest center, grouping the points into clusters. Afterward, the center are updated by calculating the mean of the points within each cluster, and the centers are moved to these new mean positions. This process is repeated: calculating the distances, reassigning points to the closest center, and updating the centers. The algorithm continues iterating until the centers no longer move, indicating that the clusters have stabilized and the optimal grouping has been reached. An example clustering can be seen in figure 2, where 5 centers are chosen at random.

2.3 KNN Function

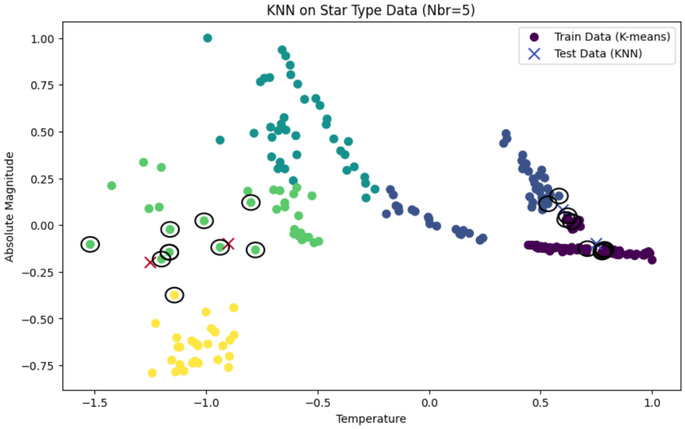

Figure 3: Plot KNN preformed on star type data with Nbr=5, where nearest neighbor points are circled

In the context of clustering with the K-means algorithm, the K-Nearest Neighbors (KNN) algorithm plays a crucial role in classifying new data points based on the clusters identified from the training data. Assuming that we do not have prior knowledge of the cluster centers, but only have labeled points from the training data, KNN can help by assigning test data points to the appropriate cluster. It does this by identifying the K nearest neighbors among the labeled training points, evaluating their assigned clusters, and then assigning the test point to the most common cluster among these neighbors. This approach not only helps in classifying new data based on the learned clusters but also provides a means to validate the clustering results. By leveraging local patterns and the proximity of neighbors, KNN assesses how well the clusters generalize to unseen data, offering valuable insights into the effectiveness and accuracy of the K-means clustering algorithm. Figure 3 gives an example of assigning new points to clusters.

2.3.1 Impact of Nbr on the Results

A small value of Nbr makes the model sensitive to local variations and noise, which can lead to overfitting. For example, if a point lies near an outlier, the model may assign it the wrong label due to the small neighborhood size focusing on nearby but misleading points. Conversely, a larger Nbr smooths the decision boundary by considering a broader set of neighbors, but it risks underfitting. This is because the classifier might overlook local patterns, leading to less accurate classification for points that are near the boundaries of different clusters.

2.3.2 Inferring a Point Near the Cluster Centre

Inferring the label of a point near the center of a cluster generally increases classification accuracy. This is because points near the center are more likely to be surrounded by data points from the same cluster. In such cases, the majority voting mechanism of KNN works effectively, as most of the nearest neighbors will have the correct label. On the other hand, if a point lies near the edge of a cluster, it becomes more susceptible to misclassification, especially if it is close to another cluster, where neighbors from both clusters could have similar distances.

2.3.3 Determining the Quality of the Classifier with True Labels

If true labels for the test data are available, the classifier's quality can be assessed in several ways. One approach is to calculate the percentage of correct predictions out of the total, providing a general measure of how often the model makes accurate classifications. However, this alone may not capture the full picture, particularly if certain groups or clusters are more difficult to classify. A more detailed analysis would involve identifying the types of mistakes the classifier makes. This includes tracking how often different groups are confused with one another, highlighting whether the model struggles with specific clusters or simply makes random errors. This analysis can reveal potential biases, such as a tendency to over-predict a particular group or consistently misclassify certain data points.

3 Decision Boundary Function

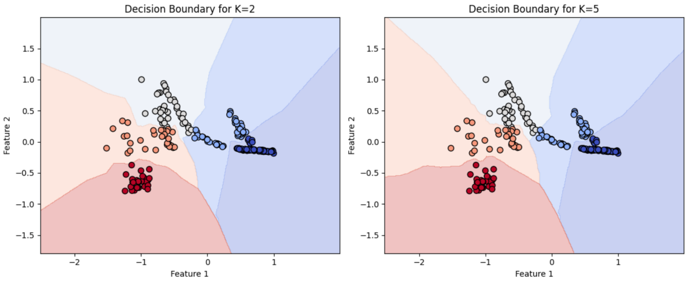

Figure 4: Enter Caption

Decision boundaries represent the regions where the classifier's decision changes, delineating which class a particular point is assigned to based on its position in the feature space. For example, in KNN classification, the decision boundary is influenced by the number of nearest neighbors considered (K). When K is small, such as 2, the decision boundary can become highly sensitive to local variations and noise in the data. This often results in a more complex and jagged boundary that closely follows the data points. Conversely, with a larger K, such as 5, the decision boundary smooths out and generalizes more, potentially overlooking small variations but providing a broader and less sensitive classification.