Algorithm 1: K-Means Clustering

K-Means is an unsupervised clustering algorithm that attempts to organize datapoints into K clusters. It does this by instantiating centroids (randomly in case of the naïve approach), then assigning the datapoints with the lowest Euclidean distance to itself to that centroid, forming a cluster. When a cluster is formed, the centroids position is set to the mean of its cluster. The process is repeated until no more reassignments are made (convergence). This type of machine learning algorithm fits well for problems dealing with large amounts of similar but somewhat distinct unlabeled data.

An important bias in this machine learning algorithm is that datapoints near a given cluster are more like each other than those farther away. Another important bias in K-Means (and in unsupervised learning more broadly) is that all features have equal importance, this can be ensured with normalization. In this assignment, the naive algorithm was implemented.

The first dataset was already normalized between 0 and 1, it was also easily handled by the algorithm, which mean no preprocessing was needed.

The second dataset only needed to be normalized between 0 and 1 to work properly with the algorithm.

The results from the notebook were the successful identification of K centroids (which defines the clusters) in the given tasks.

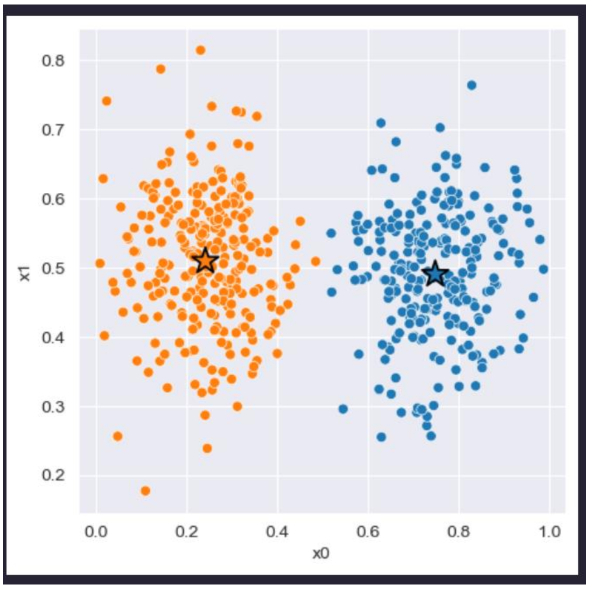

Figure 1: Result from dataset 1

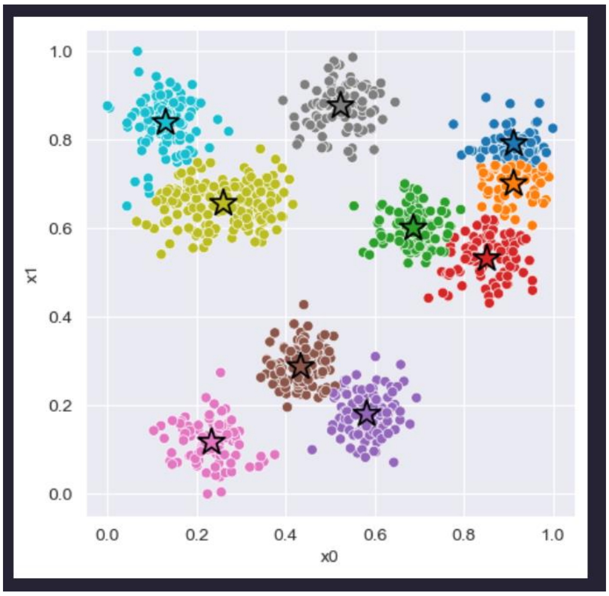

Figure 2: Result from dataset 2 (Preprocessed)

Algorithm 2: Logistic Regression

Logistic regression is a supervised classification algorithm (despite regression being in its name) used to fit inputs into a binary class (0 or 1). It does this by creating a linear model (like linear regression), then passing the result into a sigmoid function. When the error is calculated, it is applied to a gradient descent, which modifies theta and the bias to iteratively improve the approximation of the line/hyperplane that separates the two classes. This is a very important bias in this algorithm, as there must exist a clear separation between the two classes to be identified. As the problem in this specific task was an example of multiple logistic regression (more than one independent variable) a bias variable was needed, which represents the average prediction error. This type of machine learning algorithm is well suited for binary problems where the data has a somewhat clear separation between them. In such cases it might be better suited to evaluating the edge cases, more than the obvious clear cases.

Dataset 1 did not require any preprocessing, as the datapoints in each class are well separated. Dataset 2 required preprocessing, as datapoints of one class was sandwiched between datapoints belonging to the other class. To resolve this, the entire dataset was preprocessed and feature engineered by summing both features of a datapoint and taking their absolute value, then normalizing the resulting data. This created a clear separation between the classes. This was done because the dataset (when plotted) was nearly symmetrical across the positive and negative side of either axis. Another approach could be to rotate all data 45 degrees counterclockwise then taking the absolute values of all features.

The results of the notebook were the successful identification of the separatory line between the classes in each dataset, which in turn allows us to predict upon similar data.

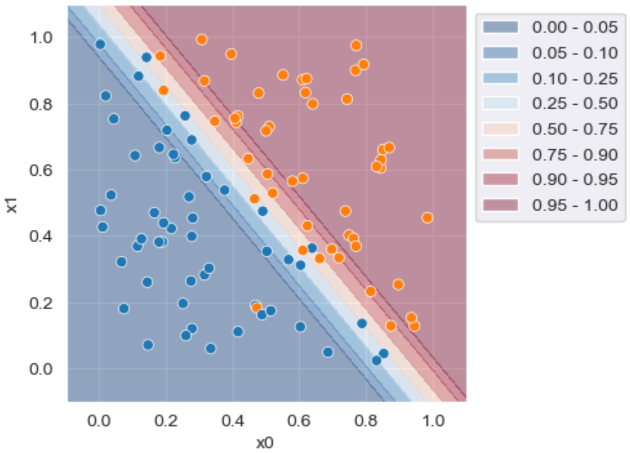

Figure 3: Result from dataset 1

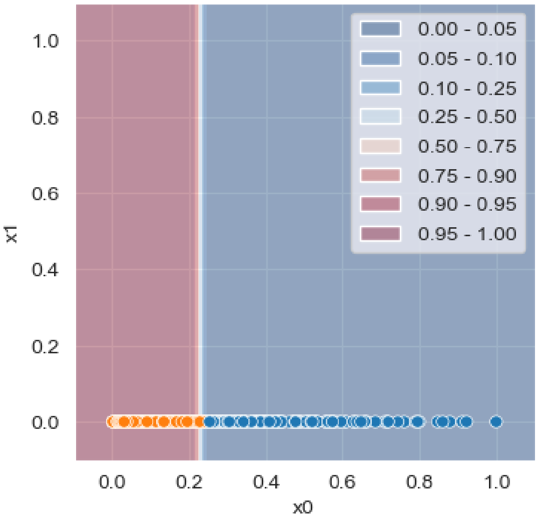

Figure 4: Result from dataset 2 (Preprocessed)