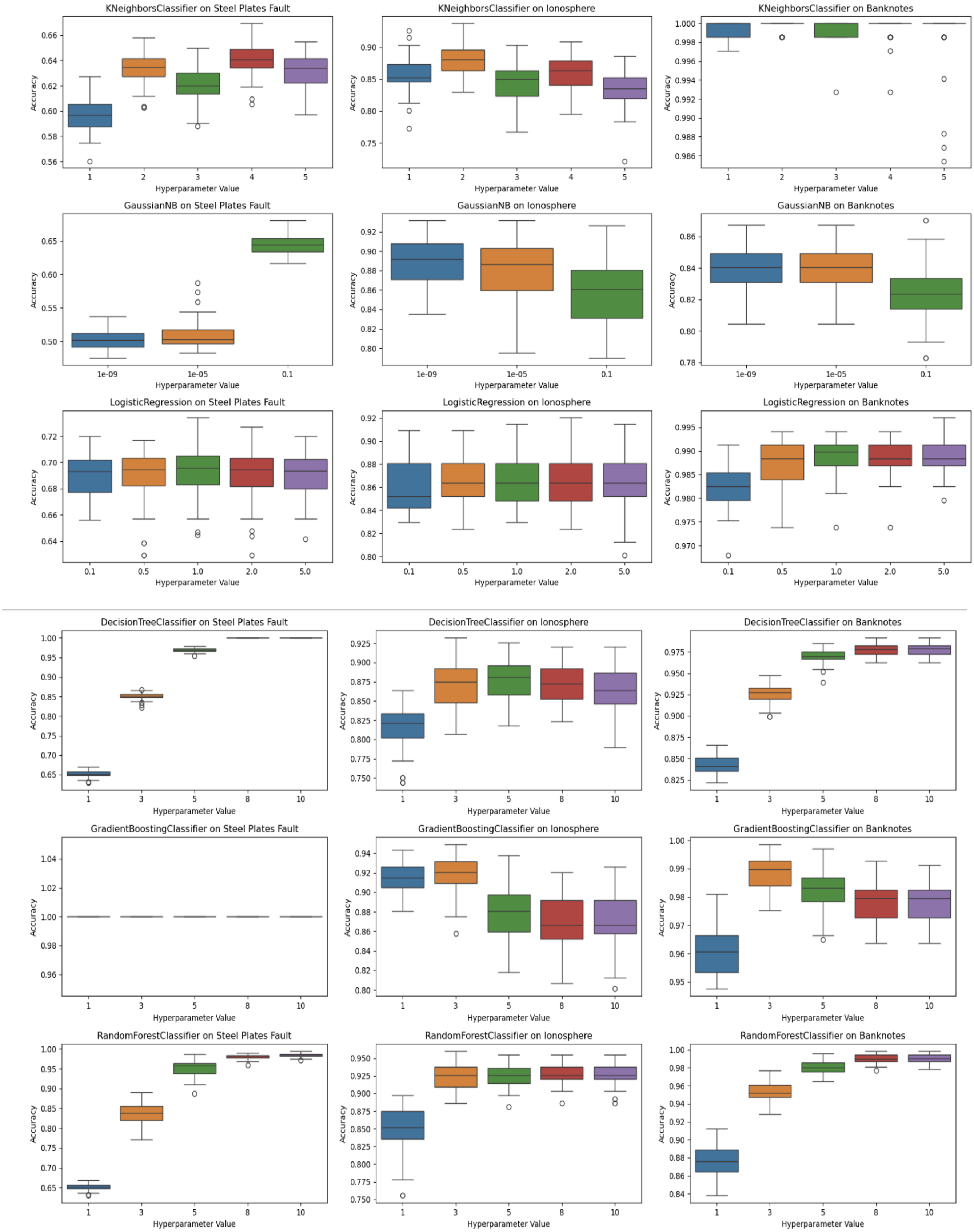
Assignment 1

1. Classification Tasks

(i) Figure 1: Box plots comparing each classifier’s accuracy across three datasets:



KNeighborsClassifier on Steel Plates Fault

KNeighborsClassifier on Ionosphere

KNeighborsClassifier on Banknotes

GaussianNB on Steel Plates Fault

GaussianNB on Ionosphere

GaussianNB on Banknotes

LogisticRegression on Steel Plates Fault

LogisticRegression on Ionosphere

LogisticRegression on Banknotes

DecisionTreeClassifier on Steel Plates Fault

DecisionTreeClassifier on Ionosphere

DecisionTreeClassifier on Banknotes

GradientBoostingClassifier on Steel Plates Fault

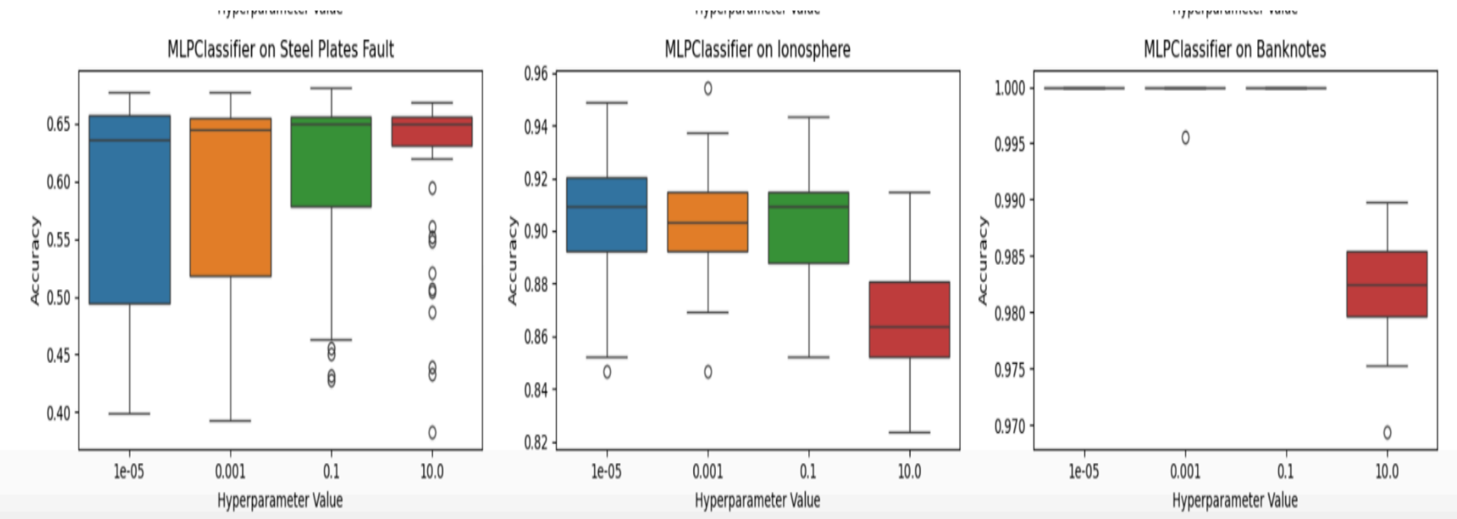
GradientBoostingClassifier on Ionosphere

GradientBoostingClassifier on Banknotes

RandomForestClassifier on Steel Plates Fault

RandomForestClassifier on Ionosphere

RandomForestClassifier on Banknotes



(ii)

Table 1: Lowest Mean Values of Test Errors for each Classifer

Classifier	Steel Plates Fault	Ionosphere	Banknotes
KNN	0.359918	0.118636	0.000204
GNB	0.355201	0.111591	0.159971
LR	0.306962	0.133523	0.010991
DTC	0.000000	0.124773	0.022478
GBC	0.000000	0.082045	0.011079
RFC	0.015283	0.071818	0.009650
MLP	0.381607	0.096136	0.000000

Table 2: Corresponding Hyperparameters for the values in Table 1

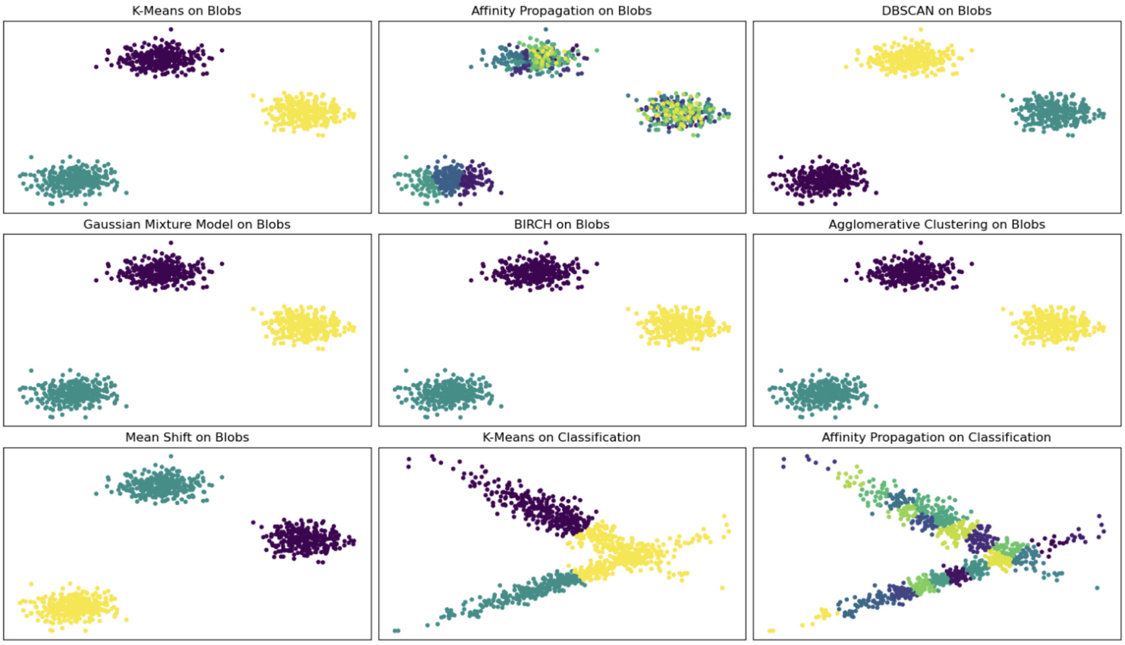
Classifier	Steel Plates Fault	Ionosphere	Banknotes
KNN	4	2	2
GNB	1.000000e-01	1.000000e-09	1.000000e-09

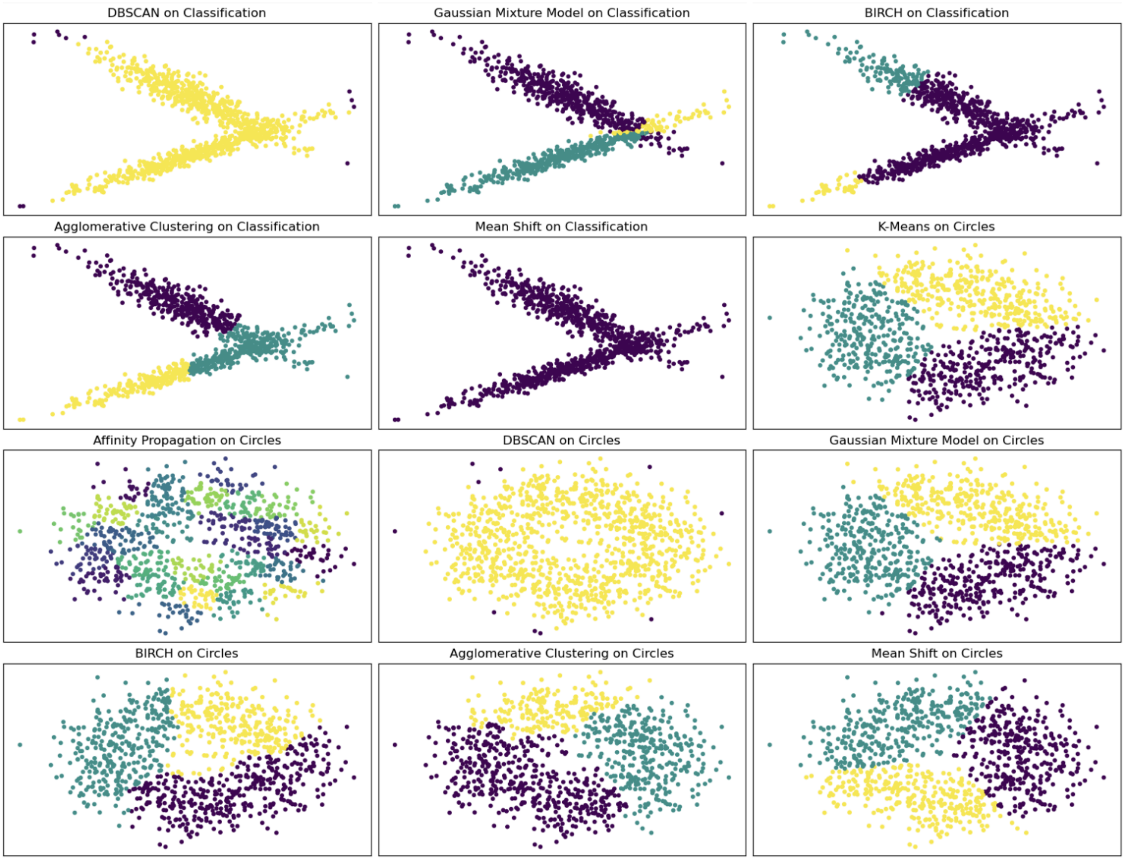
LR	1.0	2.0	1.0
DTC	8	5	10
GBC	1	3	3
RFC	10	8	8
MLP	10.00000	0.10000	0.00001

(iii) The results from the summary tables indicate that the Random Forest Classifier (RFC) and Gradient Boosting Classifier (GBC) generally achieve the best performance across the three datasets, with RFC having the lowest mean test errors for both the Ionosphere and Banknotes datasets, and GBC having the lowest mean test error for the Steel Plates Fault dataset. The Decision Tree Classifier (DTC) also performed exceptionally well on the Steel Plates Fault dataset. The sensitivity of the models to hyperparameter tuning is evident, especially in MLP, where the choice of alpha significantly impacts performance. Similarly, the hyperparameter “max_depth” for tree-based models (DTC, GBC, RFC) also has significant influence in determining accuracy. This suggests that while certain classifiers inherently perform better on specific datasets, hyperparameter optimisation is still important to achieve optimal results. Overall, RFC shows robust performance with the lowest mean errors across two datasets, indicating its effectiveness in handling complex patterns and its relative insensitivity to overfitting due to its ensemble nature.

2. Clustering Tasks

(i) ***Figure 2: Scatterplots Showing Clusters Generated by each Algorithm:***





(ii) When applying different clustering algorithms to the datasets: blobs, classification, and circles — we observe various strengths and weaknesses. K-Means works well on the blobs dataset with distinct, well-separated clusters but struggles with non-linear shapes like circles. It handles linearly separable data in the classification dataset but still misses complex patterns.

Affinity Propagation tends to over-segment, creating more clusters than exist, especially with complex datasets like classification and circles. This approach can capture subtle patterns but often misaligns with the true cluster count. DBSCAN works well with irregular shapes and noise, which makes it a great choice for the circles dataset. It can accurately identify non-convex patterns and it handles outliers effectively. However, the plots show that its performance on blobs can vary depending on parameter settings, sometimes misclassifying points as noise.

Gaussian Mixture Models perform similarly to K-Means, assuming data follows a Gaussian distribution. They do well on datasets with clear, convex clusters like blobs and classification but struggle with more complex shapes such as circles. BIRCH efficiently handles large datasets and performs similarly to K-Means, identifying clusters in blobs and classification reasonably well. However, it struggles with non-convex data shapes, leading to less effective clustering in circles. Agglomerative Clustering builds clusters hierarchically and does well on blobs and classification but doesn't handle circular patterns effectively. Mean Shift adapts well to data density, helping it detect clusters in blobs. However, it can over-segment or under-segment data. It manages non-uniform clusters in circles better than K-Means but not as effectively as DBSCAN.

In summary, the choice of clustering algorithm depends on the dataset's shape and characteristics. DBSCAN is ideal for non-convex shapes and handling noise, while K-Means and Gaussian Mixture Models are suitable for simpler, well-separated clusters. Affinity Propagation and Mean Shift offer adaptive clustering but require careful parameter tuning. Understanding the dataset's structure is the key to selecting the most effective clustering technique for meaningful insights.