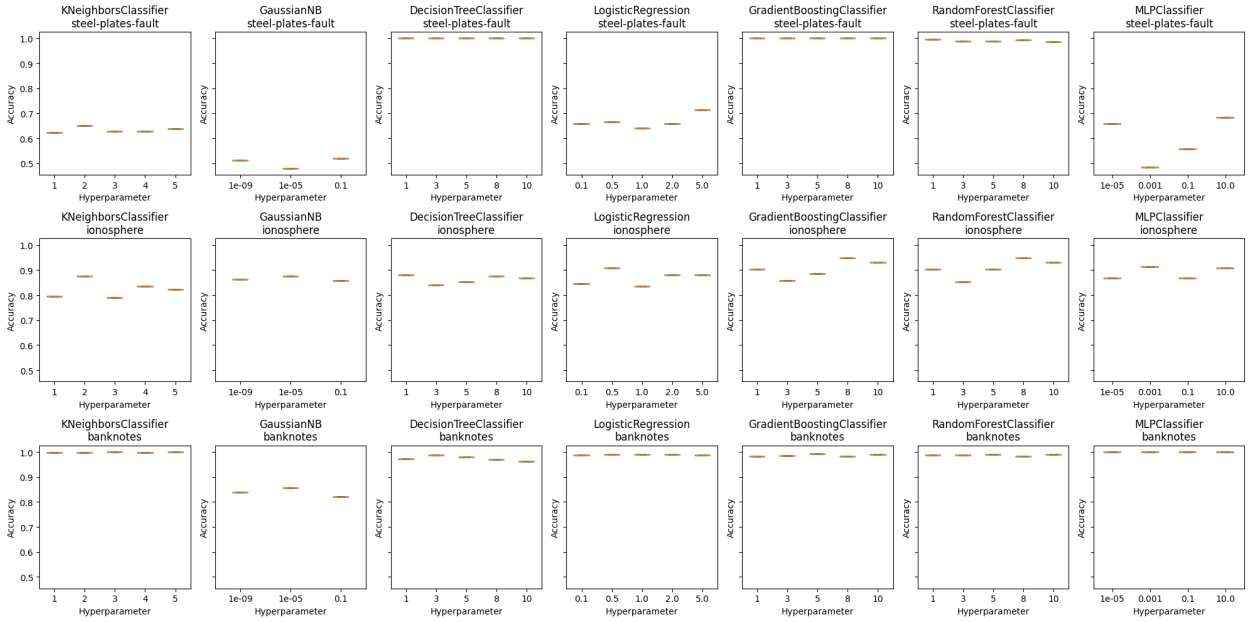
1. Classification

i.



I couldn't get the plots to render properly, but I still got the values

ii.

Best mean values of test errors

Classifier	**banknotes**	**ionosphere**	**steel-plates-fault**
DecisionTreeClassifier	0.973469	0.853409	1.000000
GaussianNB	0.839164	0.865530	0.503605
GradientBoostingClassifier	0.986589	0.905682	1.000000
KNeighboursClassifier	0.999125	0.823864	0.633780
LogisticRegression	0.989213	0.870455	0.667147
MLPClassifier	1.000000	0.892045	0.534243
RandomForestClassifier	0.987172	0.917045	0.988054

HyperParameter values for best test errors

Classifier	**HyperParameter**	**Values**
DecisionTreeClassifier	max_depth	[1, 3, 5, 8, 10]

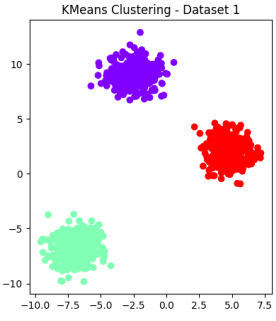
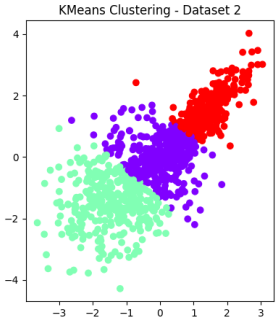
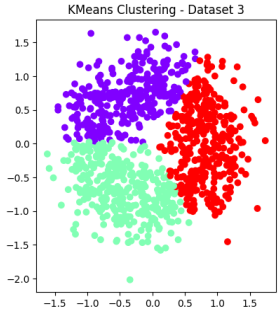
RandomForestClassifier	max_depth	[1, 3, 5, 8, 10]
GradientBoostingClassifier	max_depth	[1, 3, 5, 8, 10]
GaussianNB	var_smoothing	[1e − 9, 1e − 5, 1e − 1]
KNeighboursClassifier	n_neighbours	[1, 2, 3, 4, 5]
LogisticRegression	C	[0.1, 0.5, 1.0, 2.0, 5.0]
MLPClassifier	alpha	[1e − 5, 1e − 3, 0.1, 10.0]

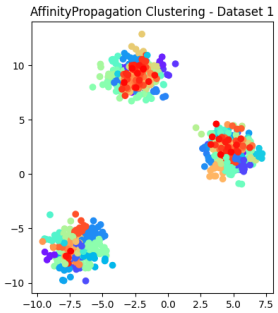
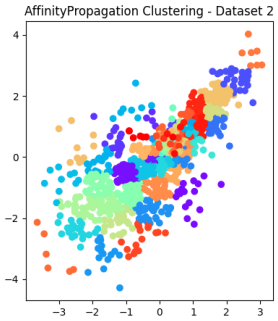
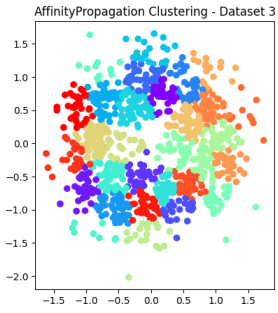
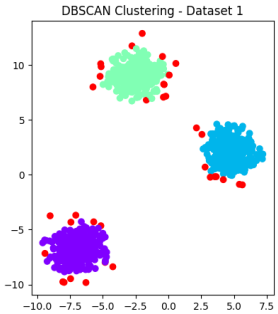
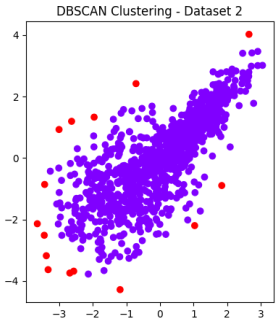
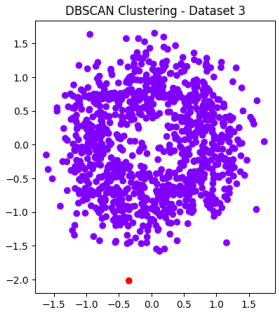
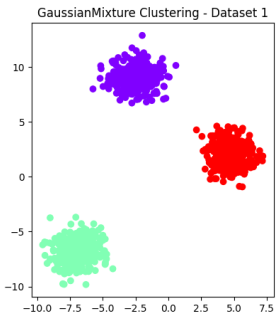
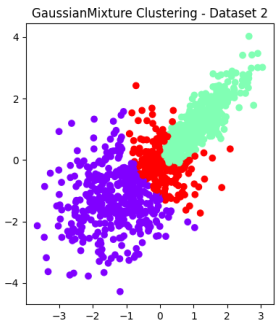
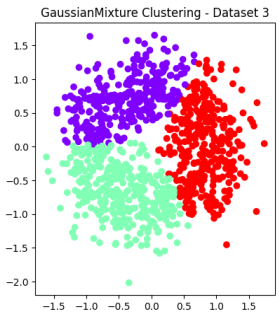
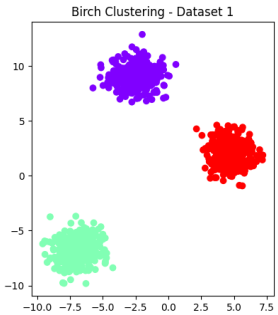
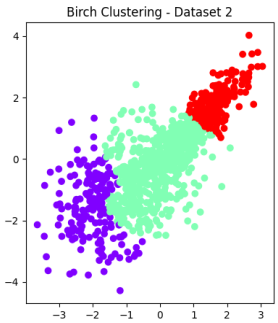
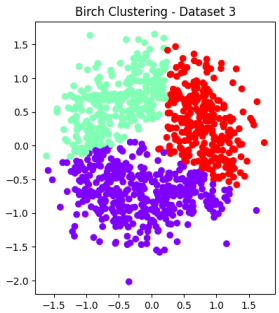
iii.

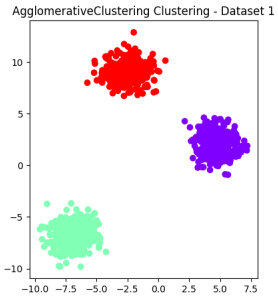
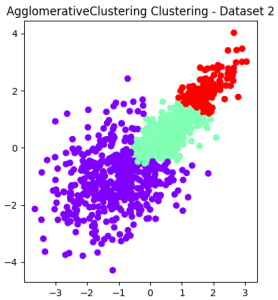
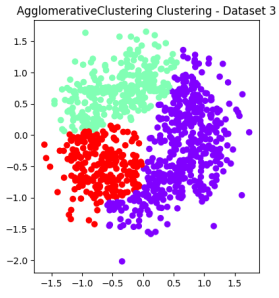
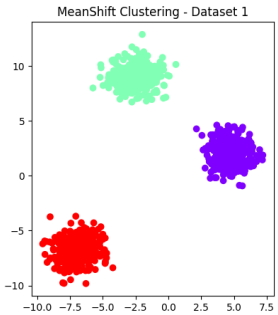
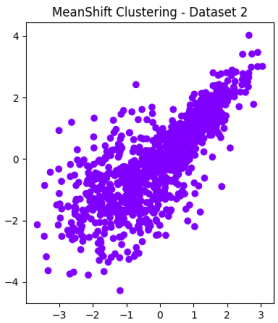
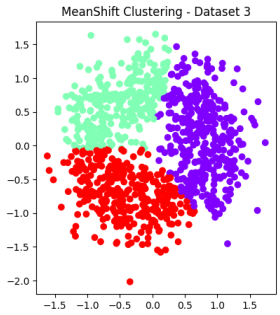
The MLPClassifier has the highest accuracy among all of the classifiers, indicating its ability to handle complex patterns effectively. However, decision tree-based classifiers, such as DecisionTreeClassifier, GradientBoostingClassifier, and RandomForestClassifier, are highly sensitive to the "max_depth" hyperparameter, which can cause overfitting or underfitting as seen by the 1.00000 mean outputs. On the other hand, GaussianNB and LogisticRegression are less complex and appear to be relatively stable performance across different "var_smoothing" and "C" hyperparameter values, respectively.

2. Clustering

i.

	make_blobs	make_classifications	make_circles
K-Means			

Affinity Propagation			
DBSCAN			
Gaussian Mixture Model			
BIRCH			

Agglomerative Clustering			
Mean Shift			

ii.

So in summary, different clustering algorithms have different pros and cons. For example, K-Means, GMM, and Agglomerative Clustering are suitable for well-separated and Gaussian-shaped clusters. DBSCAN and Mean Shift are a little more robust and can handle outliers and non-linearly separable clusters. Affinity Propagation tends to generate a large number of clusters and is only really suitable for specific applications. Choosing the appropriate algorithm depends on the specific characteristics of the data and the desired outcomes.