

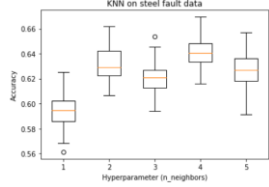
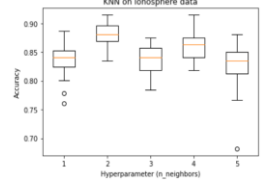
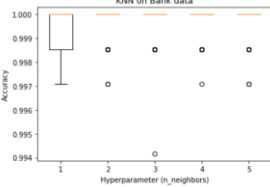
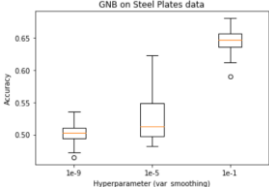
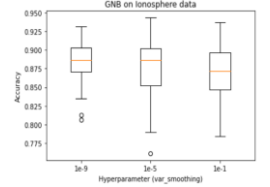
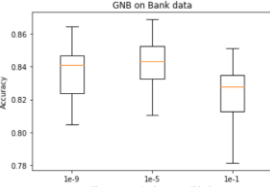
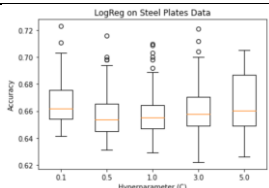
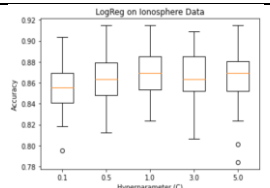
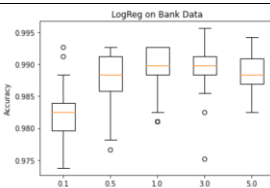
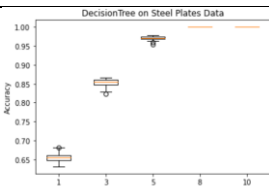
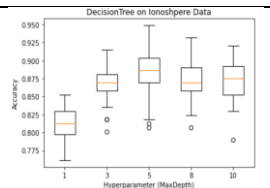
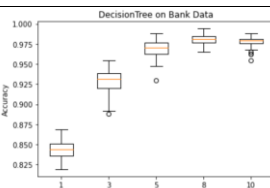
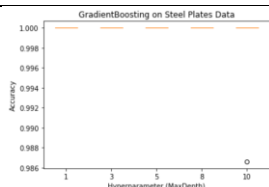
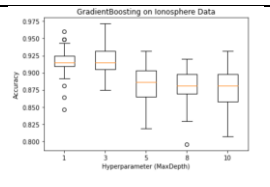
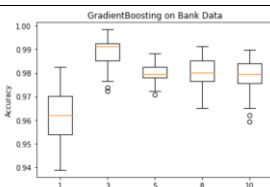
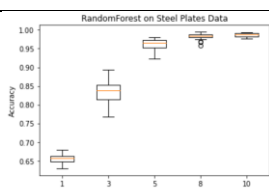
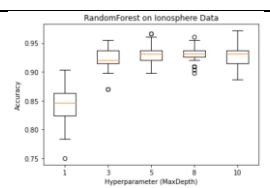
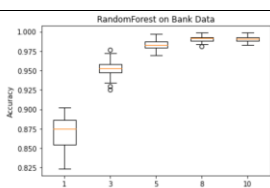
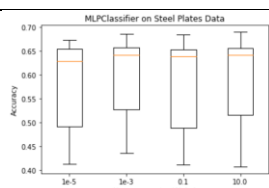
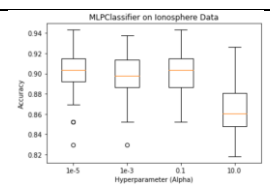
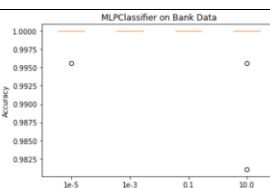
COMP309 Assignment One

Ben Allan

300476366

Part One: Classification

Task One: Boxplots

	Steel-Plates-Fault	Ionosphere	Banknotes
KNeighborsClassifier			
GaussianNB			
LogisticRegression			
DecisionTreeClassifier			
GradientBoostingClassifier			
RandomForestClassifier			
MLPClassifier			

Task Two: Table of Best Mean Test Error

	Steel-Plates-Fault	Ionosphere	Banknotes
KNeighborsClassifier	0.330587	0.085227	0.0
GaussianNB	0.319258	0.056818	0.131195
LogisticRegression	0.277034	0.085227	0.004373
DecisionTreeClassifie	0.0	0.051136	0.005831
GradientBoostingCla	0.0	0.022727	0.001458
RandomForestClassif	0.005149	0.022727	0.001458
MLPClassifier	0.309910	0.034091	0.0

Task Two: Table of Best Hyperparameters

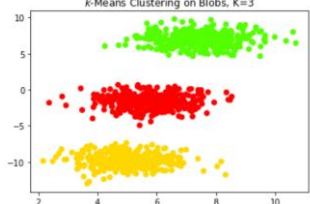
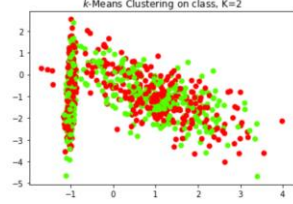
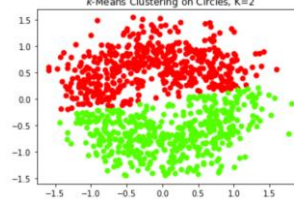
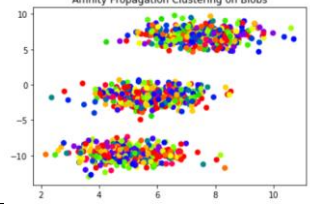
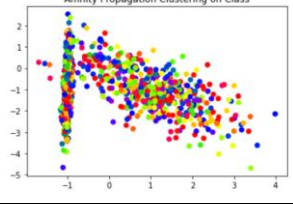
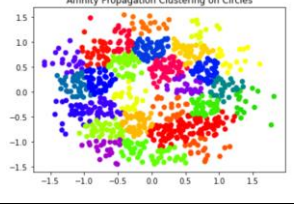
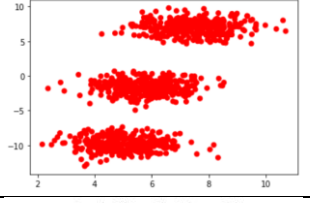
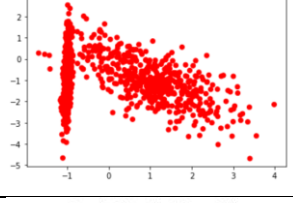
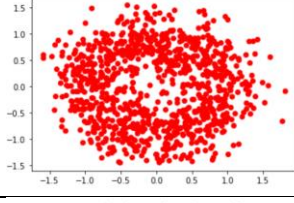
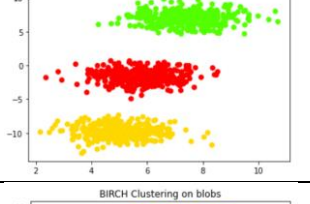
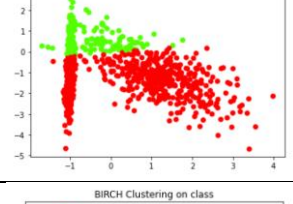
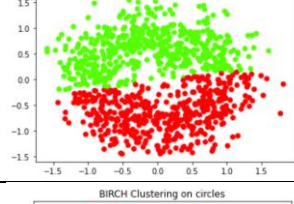
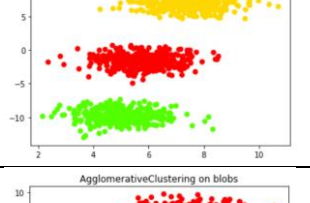
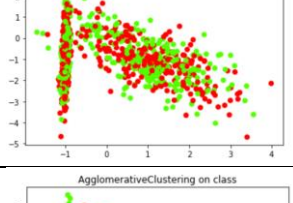
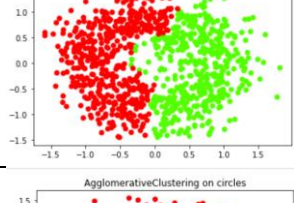
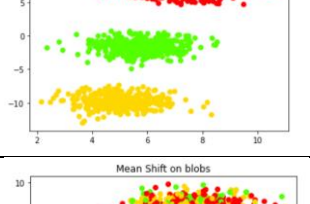
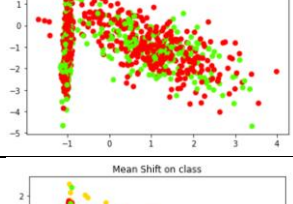
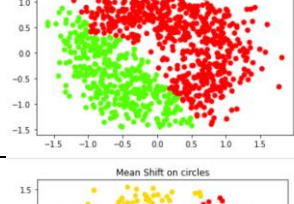
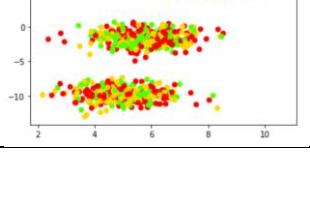
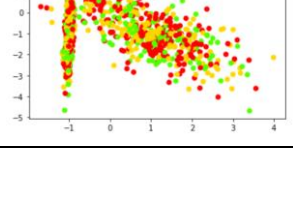
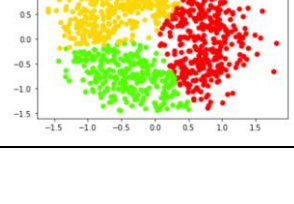
	Steel-Plates-Fault	Ionosphere	Banknotes
KNeighborsClassifier	4	2	4
GaussianNB	1e-1	1e-9	1e-5
LogisticRegression	0.1	1.0	1.0
DecisionTreeClassifier	8	10	8
GradientBoostingClassifier	1	3	3
RandomForestClassifier	10	10	10
MLPClassifier	1e-3	1e-5	1.0

Task Three: write up

The best models are the ones that perform with the highest accuracy as can be see in the boxplots above. We can also look to the mean test errors to determine the best model by taking the lowest error rate. Although error rates with a score of 0.0 or accuracy's with a score of 1.0 shouldn't be considered as they provide a perfect fit to the data, meaning they capture every single point, the model will not be able to achieve generalisation when used on real data and therefore is an invalid model. Therefore, I will not be including these with the best models. We see varied results across the different algorithms and datasets some algorithms perform better than others of different datasets often due to the different ways they fit data, for example model complexity or bias-variance trade off. These and serval other factors influence the ways algorithms fit data and therefore is why we don't see the same result every time. These models are highly sensitive to the complexity control hyperparameter we often see wildly different results across the boxplots, most notably the within the Tree classifiers, the increase in max depth greatly increases the accuracy of the model. The best model across all three datasets is the random forest classifier as it had the best accuracy and mean test error scores over all datasets. Although we should express some concerns about overfitting as the best scores came from forests with much more complexity, having a max depth of 8 and 10 and 8 on each dataset respectively.

Part Two: Clustering

Task (i): Table of scatterplots

	Blobs	Classification	Circles
K-Means			
Affinity Propagation			
DBSCAN			
Gaussian Mixture Model			
BIRCH			
Agglomerative Clustering			
Mean Shift			

Task(ii):

The range of algorithms produced mixed results on the various datasets they were tested on. The Classification dataset seems to produce poor results no matter what algorithm was applied. We may have seen vastly improved results if the parameters of each algorithm were precisely tuned using their respective hyperparameters. The Kmeans algorithm saw good clustering of the 2 datasets, we can note kmeans spherical nature and works well within those datasets but with class and its elliptical cluster. Affinity propagation performed awfully on all datasets probably because of the way it identified the explementary points among the data points are how it formed a cluster around it. DBSCAN performed the worst of all the algorithms it couldn't correctly cluster any of the datasets bringing back one big cluster every time. Gaussian Mixture Model was the best algorithm of all that were tested mostly due to the way it handles these types of datapoints. As both the means and covariances move, the algorithm is able to better generate the data and as it repeats it provides a better fit. BIRCH gave a similar fit to the Kmeans although classifying the blobs dataset in the reverse order to kmeans, this is due to the hierarchical characteristics of this model. Argumentative clustering uses a bottom-up hierarchical approach which is why it was better suited to the blobs data, if a different linkage criterion was used for the merge strategy, we could have seen better results. Finally, the mean shift algorithm performed very poorly on the first 2 datasets but did well on the circles points, the mean shift is supposed to find dense areas of datapoints, but without the tuning of the bandwidth parameter we weren't able to see good clustering on all the datasets.

Overall, we saw mixed results across each data set and algorithms mostly because different algorithms are suited to different types of data points and spread. The Gaussian mixture defiantly performed the best across all three sets, but I believe that if the parameters of each model were tuned carefully a lot more algorithms could correctly cluster each datasets points.