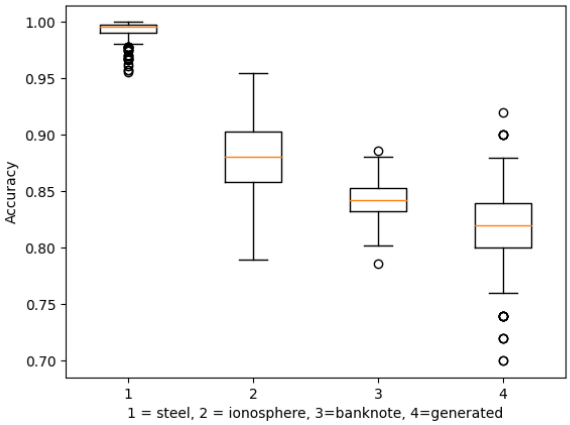
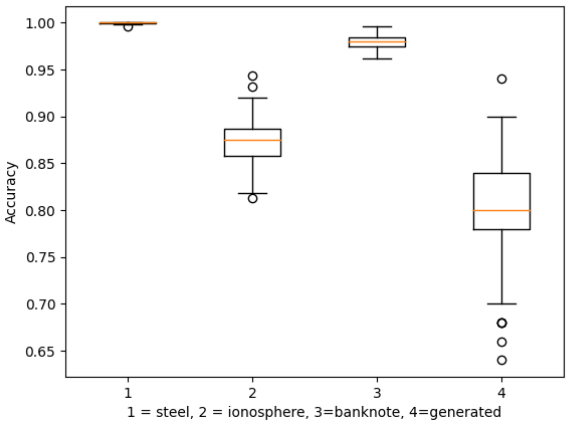
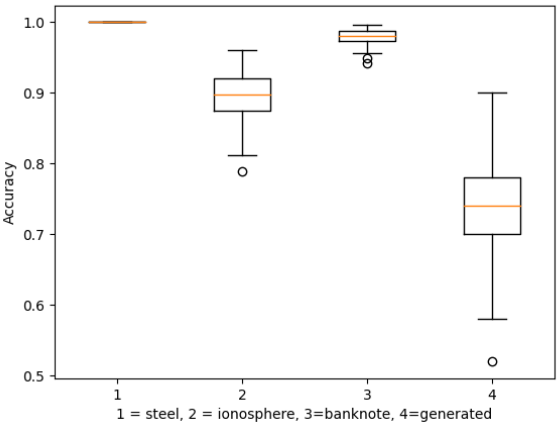
GaussianNB



LogisticRegression

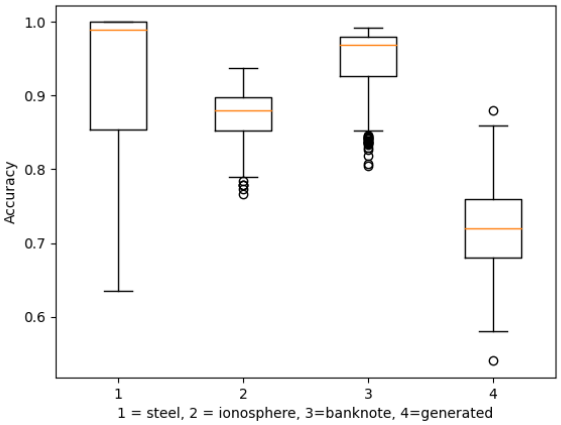


GradientBoostingClassifier

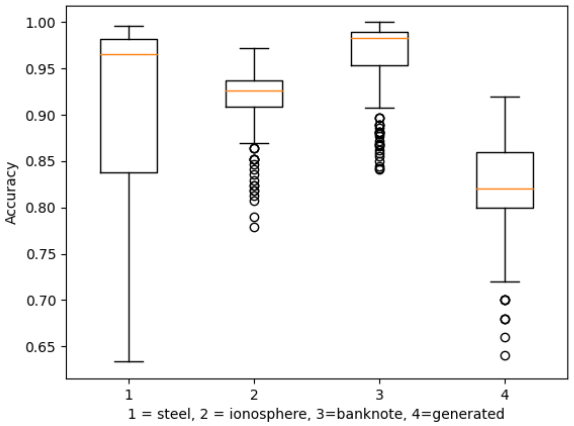


MLPClassifier

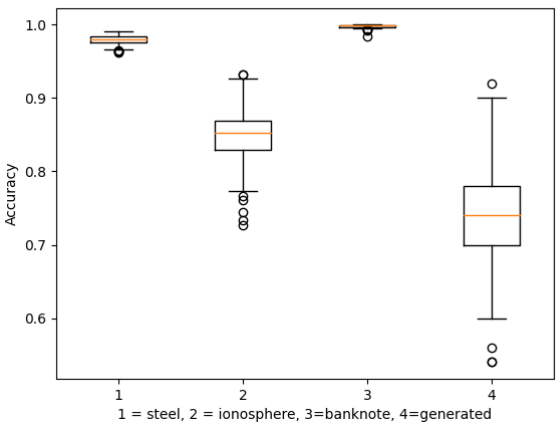
DecisionTreeClassifier



RandomForestClassifier



KNeighborsClassifier



```
+------------Table 1-------------+
+-------------------------+---------------------+---------------------+---------------------+---------------------------+---------------------------+---------------------+
|           +             | KNeighborsClassifier |     GaussianNB      | LogisticRegression | DecisionTreeClassifier | GradientBoostingClassifier | RandomForestClassifier |   MLPClassifier     |
+-------------------------+---------------------+---------------------+---------------------+---------------------------+---------------------------+---------------------+
|    Steel-plate-faults   |  0.9816683831101956 | 0.9972659506645087 | 0.9996910401647786 |          1.0              |           1.0             |  0.9868177136972193 | 0.9991336046948817 |
|       Ionosphere        |  0.8880681818181817 | 0.8809523809523809 | 0.8776136363636364 |  0.8913636363636362       |     0.9247727272727272    |  0.9363636363636364 | 0.901605339105339  |
| Banknote-authentication |  0.9982798833819241 | 0.8420796890184644 | 0.9847813411078719 |  0.979475218658892        |     0.9900291545189505    |  0.9926530612244899 | 0.9985885510666851 |
|    Generated Dataset    |  0.7360000000000001 | 0.8364285714285715 |       0.838        |          0.7376           |     0.7839999999999999    |         0.8008      | 0.8101587301587303 |
+-------------------------+---------------------+---------------------+---------------------+---------------------------+---------------------------+---------------------+
+------------Table 2-------------+
+-------------------------+---------------------+---------------------+---------------------+---------------------------+---------------------------+---------------------+--------------+
|           +             | KNeighborsClassifier |     GaussianNB      | LogisticRegression | DecisionTreeClassifier | GradientBoostingClassifier | RandomForestClassifier | MLPClassifier |
+-------------------------+---------------------+---------------------+---------------------+---------------------------+---------------------------+---------------------+--------------+
|    Steel-plate-faults   |       k = 1         | var_smoothing = 1e-9 |      C = 5.0       |     max_depth = 6         |       max_depth = 1       |     max_depth = 10     |  alpha = 0.1  |
|       Ionosphere        |       k = 2         | var_smoothing = 1e-9 |      C = 1.0       |     max_depth = 2         |       max_depth = 1       |     max_depth = 8      |  alpha = 0.1  |
| Banknote-authentication |       k = 2         | var_smoothing = 1e-9 |      C = 5.0       |     max_depth = 10        |       max_depth = 4       |     max_depth = 9      |  alpha = 1e-3 |
|    Generated Dataset    |       k = 5         | var_smoothing = 1e-1 |      C = 1.0       |     max_depth = 8         |       max_depth = 2       |     max_depth = 7      |  alpha = 10.0 |
+-------------------------+---------------------+---------------------+---------------------+---------------------------+---------------------------+---------------------+--------------+
+------------Overall Averages per algorithm-------------+
+-------------------------+---------------------+---------------------+---------------------+---------------------------+---------------------------+---------------------+---------------------+
|           +             | KNeighborsClassifier |     GaussianNB      | LogisticRegression | DecisionTreeClassifier | GradientBoostingClassifier | RandomForestClassifier |   MLPClassifier     |
+-------------------------+---------------------+---------------------+---------------------+---------------------------+---------------------------+---------------------+---------------------+
|    Steel-plate-faults   |  0.9782780638516992 | 0.9927500694750953 | 0.9994109165808445 |  0.9163501544799176       |          1.0              |  0.8970010298661174 | 0.9987657953672371 |
|       Ionosphere        |  0.8513636363636364 | 0.874774531024531  | 0.8737954545454545 |  0.8677954545454546       |     0.8943181818181818    |  0.9176363636363635 | 0.8964420995670995 |
| Banknote-authentication |  0.9979883381924198 | 0.8406451015780462 |  0.98000583090379  |  0.9474577259475219       |     0.9812128279883382    |  0.9655801749271137 | 0.9909007820815401 |
|    Generated Dataset    |  0.7031999999999999 | 0.8321428571428571 | 0.8228800000000001 |        0.71896            |     0.7296800000000001    |         0.78096     | 0.7747619047619048 |
+-------------------------+---------------------+---------------------+---------------------+---------------------------+---------------------------+---------------------+---------------------+
time elapsed:
774.0852284431458
```

Write a paragraph summarising the overall results, as captured in these two tables.

From table one we can observe that for

Steel plate faults: Decision tree classifier using a value of 6 and Gradient Boosting Classifer using a value of 1 produced the most accurate result with 100% rate of accuracy.
KNeighbour Classifier produced the least accurate result with a 98.1% rate of accuracy using a value of 1.

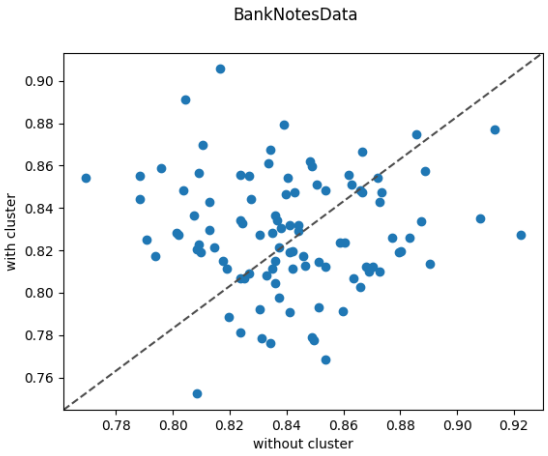
Ionosphere: Random Forest Classifier using a value of 8 produced the most accurate result with 93.63% rate of accuracy.
LogRegression produced the least accurate result with a 87.76% rate of accuracy using a value of 1.0

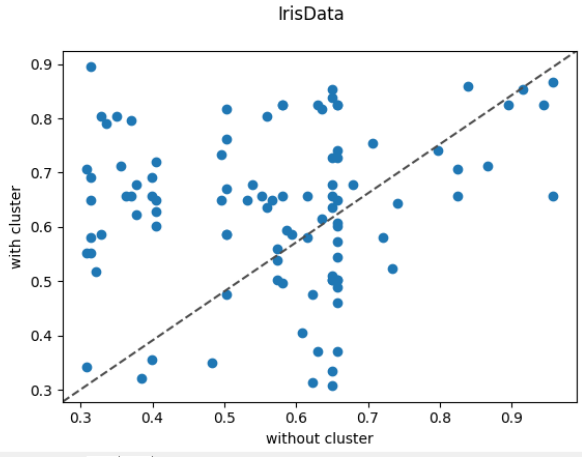
Banknote: MLP Classifier using a value of 1e-3 produced the most accurate result with a 99.85% accuracy
GaussianNB produced the least accurate result with 84.2% rate of accuracy using a value of 1e-9

Generated dataset: LogRegression using the value of 1.0 produced the most accurate result with 83.8% rate of accuracy.
KNeighbour Classifier produced the least accurate result with a 73.6% rate of accuracy using a value of 5.

If you notice something unexpected, point it out, explaining why you think it is worth mentioning.

It is unusual for the accuracy of any algorithm to be 100% accurate, Decision tree classifier using a value of 6 and Gradient Boosting Classifer using a value of 1 produced a 100% rate of accuracy for the data set Steel plate faults.
I found this result unexpected as the results for all values of Gradient Boosting Classifer produce a 100% rate of accuracy (observed from table 3). However the accuracy of the prediction of this dataset is high for all algorithms with all accuracy rates being over 98% hence it may be justifiable.





Iris data appears to be more accurate than bank notes, from the graph.
Iris has better results with semi-supervised learning by clustering the data we are able to observed this from more points plotted above the diagonal axis.
Whereas with the banknote data set produces similar result with and without the use of semi-supervised learning with a majority of the points plotted around the diagonal axis.

Why:
A semi-supervised machine-learning algorithm uses a limited set of labelled sample data to train itself, resulting in a 'partially trained' model.
The brief required the use of 3 clusters (KMeans(n_clusters=3)) for iris that has 3 labels (Iris-virginica, Iris-versicolor, Iris-setosa) this dataset may benefit more from Semi-supervised learning.
However, banknotes data only has 2 labels (1, 2) so a cluster of 3 doesn't help the algorithm learn as much about the data.