Analysis of Models

Data sets uses

Steel-plates-fault:
https://www.openml.org/search?type=data&sort=runs&id=1504&status=active
Ionosphere:
https://www.openml.org/search?type=data&sort=runs&id=59&status=active
Banknote-authentication:
https://www.openml.org/search?type=data&sort=runs&id=1462&status=active

A 7-by-3 (7 classifiers and 3 datasets) table to present the boxplots on the classifier accuracy versus parameter values has been made and can be seen in Code&Output.html file.

Two summary tables, with rows being classifiers, and columns being datasets.

Table (1) contains the lowest mean value of the test errors obtained from each classifier with various hyperparameter settings.

Table (2) contains the corresponding hyperparameter values for obtaining the best test errors.

Summary Table 1: Lowest Mean Test Errors

	Ionosphere	Steel Plates Fault	Banknote Auth
KNeighborsClassifier	0.118636	0.359918	0.000204082
GaussianNB	0.111591	0.355201	0.159971
LogisticRegression	0.136591	0.340474	0.0109038
DecisionTreeClassifier	0.1275	0	0.0222741
GradientBoostingClassifier	0.0815909	0	0.011137
RandomForestClassifier	0.0844318	0.0305664	0.0127697
MLPClassifier	0.0947727	0.403337	0

Summary Table 2: Corresponding Hyperparameter Values

	Ionosphere	Steel Plates Fault	Banknote Auth
KNeighborsClassifier	2	4	2
GaussianNB	1e-09	0.1	1e-09
LogisticRegression	2	1	5
DecisionTreeClassifier	3	8	8
GradientBoostingClassifier	3	1	3
RandomForestClassifier	10	10	10
MLPClassifier	1e-05	10	1e-05

Comparing and analysis of the overall results as captured in these two tables.

The results show that different classifiers perform better or worse depending on the dataset. For the Ionosphere dataset, the GradientBoostingClassifier leads with the lowest mean test error of 0.0816 followed by the RandomForestClassifier at 0.0844. This suggests that ensemble methods are good at capturing the complexities in the data. In the Steel Plates Fault dataset, both the DecisionTreeClassifier and GradientBoostingClassifier achieve perfect scores of 0, indicating that the data is highly separable with the right tree depth and iterative boosting can significantly improve accuracy. For the Banknote Authentication dataset the MLPClassifier has a mean test error of 0, which is the best showing the strength of neural networks in identifying patterns. The KNeighborsClassifier also does well here with an error of 0.0002, benefiting from instance-based learning. Classifiers show different sensitivity levels to their hyperparameters. The KNeighborsClassifier's performance changes a lot with different numbers of neighbours, with optimal values being 2 and 4. GaussianNB's var_smoothing parameter varies from 1e-09 to 0.1, indicating the need to control data variance for better performance. LogisticRegression's C parameter, which affects regularisation strength, changes across datasets, suggesting that less regularisation (higher C) is better for complex data, while more regularisation (lower C) prevents overfitting in simpler data. The DecisionTreeClassifier's max_depth influences complexity, with the best depths balancing detail and overfitting. Both the GradientBoostingClassifier and RandomForestClassifier are sensitive to max_depth, preferring depths that strike a balance, though GradientBoosting often needs lower depths due to its iterative nature. The MLPClassifier's alpha parameter, which controls regularisation strength, shows a need for minimal regularisation in some cases like 1e-05 for Ionosphere and Banknote Authentication and higher values in others like 10 for Steel Plates Fault to prevent overfitting. Overfitting happens when a model learns the training data too well, including noise and outliers, making it perform badly on new, unseen data. Underfitting occurs when a model is too simple to capture the underlying patterns in the data, resulting in poor performance on both the training and test sets. Overall, ensemble methods like GradientBoosting and RandomForest tend to perform well across datasets, and fine tuning hyperparameters is key to balancing underfitting and overfitting based on the specific data.