There are 2 notebooks in the zip file. The codes should work without any change as the base datasets are included in the zip. If they are not, paths in both notebooks should be changed to match dataset and image paths.

Part 1:

RandomisedSearchCV was used

Decision Tree Classifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C | D | Best Parameters | Accuracy | F1 Score |
| 300 | 100 | 'criterion': 'gini', 'max\_depth': 3, 'max\_features': 8, 'min\_samples\_leaf': 2, 'splitter': 'random' | 0.5577889447236181 | 0.5686274509803921 |
| 300 | 1000 | 'criterion': 'entropy', 'max\_depth': 5, 'max\_features': 7, 'min\_samples\_leaf': 2, 'splitter': 'best' | 0.5907953976988495 | 0.6101048617731173 |
| 300 | 5000 | 'criterion': 'entropy', 'max\_depth': 5, 'max\_features': 7, 'min\_samples\_leaf': 2, 'splitter': 'best' | 0.6105610561056105 | 0.6506996770721205 |
| 500 | 100 | 'criterion': 'entropy', 'max\_depth': 5, 'max\_features': 4, 'min\_samples\_leaf': 1, 'splitter': 'best | 0.592964824120603 | 0.6161137440758294 |
| 500 | 1000 | 'criterion': 'entropy', 'max\_depth': 5, 'max\_features': 7, 'min\_samples\_leaf': 2, 'splitter': 'best' | 0.6358179089544772 | 0.6448780487804878 |
| 500 | 5000 | 'criterion': 'gini', 'max\_depth': None, 'max\_features': 7, 'min\_samples\_leaf': 6, 'splitter': 'random' | 0.6541654165416542 | 0.6491477272727272 |
| 1000 | 100 | 'criterion': 'gini', 'max\_depth': 3, 'max\_features': 8, 'min\_samples\_leaf': 2, 'splitter': 'random' | 0.7788944723618091 | 0.8035714285714286 |
| 1000 | 1000 | 'criterion': 'gini', 'max\_depth': None, 'max\_features': 7, 'min\_samples\_leaf': 6, 'splitter': 'random' | 0.7338669334667334 | 0.7304964539007092 |
| 1000 | 5000 | 'criterion': 'gini', 'max\_depth': None, 'max\_features': 7, 'min\_samples\_leaf': 6, 'splitter': 'random' | 0.7634763476347635 | 0.7609903991915108 |
| 1500 | 100 | 'criterion': 'entropy', 'max\_depth': 5, 'max\_features': 7, 'min\_samples\_leaf': 2, 'splitter': 'best' | 0.7989949748743719 | 0.7959183673469388 |
| 1500 | 1000 | 'criterion': 'gini', 'max\_depth': None, 'max\_features': 7, 'min\_samples\_leaf': 6, 'splitter': 'random' | 0.8574287143571786 | 0.8558421851289832 |
| 1500 | 5000 | 'criterion': 'gini', 'max\_depth': None, 'max\_features': 7, 'min\_samples\_leaf': 6, 'splitter': 'random'} | 0.8845884588458846 | 0.8847613341322148 |
| 1800 | 100 | 'criterion': 'entropy', 'max\_depth': 5, 'max\_features': 7, 'min\_samples\_leaf': 2, 'splitter': 'best' | 0.8844221105527639 | 0.8866995073891625 |
| 1800 | 1000 | 'criterion': 'gini', 'max\_depth': None, 'max\_features': 7, 'min\_samples\_leaf': 6, 'splitter': 'random' | 0.903951975987994 | 0.9046673286991062 |
| 1800 | 5000 | 'criterion': 'gini', 'max\_depth': None, 'max\_features': 7, 'min\_samples\_leaf': 2, 'splitter': 'random' | 0.9461946194619462 | 0.945744251714401 |

Bagging Classifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C | D | Best Parameters | Accuracy | F1 Score |
| 300 | 100 | 'n\_estimators': 10, 'max\_samples': 50, 'max\_features': 10, 'bootstrap\_features': False, 'bootstrap': True} | 0.592964824120603 | 0.64 |
| 300 | 1000 | 'n\_estimators': 20, 'max\_samples': 10, 'max\_features': 10, 'bootstrap\_features': False, 'bootstrap': False} | 0.5012506253126563 | 0.5236502627806976 |
| 300 | 5000 | 'n\_estimators': 20, 'max\_samples': 10, 'max\_features': 10, 'bootstrap\_features': False, 'bootstrap': False | 0.523952395239524 | 0.3527332064182757 |
| 500 | 100 | 'n\_estimators': 15, 'max\_samples': 25, 'max\_features': 2, 'bootstrap\_features': False, 'bootstrap': False} | 0.5376884422110553 | 0.6 |
| 500 | 1000 | 'n\_estimators': 20, 'max\_samples': 25, 'max\_features': 13, 'bootstrap\_features': True, 'bootstrap': False} | 0.5782891445722862 | 0.5803882528621204 |
| 500 | 5000 | 'n\_estimators': 20, 'max\_samples': 25, 'max\_features': 13, 'bootstrap\_features': True, 'bootstrap': False} | 0.5944594459445944 | 0.637266302889346 |
| 1000 | 100 | 'n\_estimators': 20, 'max\_samples': 50, 'max\_features': 5, 'bootstrap\_features': True, 'bootstrap': True} | 0.7236180904522613 | 0.6994535519125683 |
| 1000 | 1000 | 'n\_estimators': 15, 'max\_samples': 50, 'max\_features': 10, 'bootstrap\_features': True, 'bootstrap': False} | 0.8244122061030515 | 0.8242363545317977 |
| 1000 | 5000 | 'n\_estimators': 20, 'max\_samples': 25, 'max\_features': 13, 'bootstrap\_features': True, 'bootstrap': False} | 0.7255725572557256 | 0.7463955637707947 |
| 1500 | 100 | 'n\_estimators': 20, 'max\_samples': 25, 'max\_features': 13, 'bootstrap\_features': True, 'bootstrap': False} | 0.914572864321608 | 0.919431279620853 |
| 1500 | 1000 | 'n\_estimators': 20, 'max\_samples': 25, 'max\_features': 13, 'bootstrap\_features': True, 'bootstrap': False} | 0.9389694847423712 | 0.9393638170974156 |
| 1500 | 5000 | 'n\_estimators': 20, 'max\_samples': 25, 'max\_features': 13, 'bootstrap\_features': True, 'bootstrap': False} | 0.9244924492449245 | 0.9268198119608413 |
| 1800 | 100 | 'n\_estimators': 20, 'max\_samples': 25, 'max\_features': 13, 'bootstrap\_features': True, 'bootstrap': False} | 0.949748743718593 | 0.9519230769230769 |
| 1800 | 1000 | 'n\_estimators': 20, 'max\_samples': 25, 'max\_features': 13, 'bootstrap\_features': True, 'bootstrap': False} | 0.9734867433716858 | 0.9738271604938272 |
| 1800 | 5000 | n\_estimators': 20, 'max\_samples': 25, 'max\_features': 13, 'bootstrap\_features': True, 'bootstrap': False} | 0.9317931793179318 | 0.9357936358501223 |

Random Forest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C | D | Best Parameters | Accuracy | F1 Score |
| 300 | 100 | 'n\_estimators': 100, 'min\_samples\_split': 2, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 100, 'criterion': 'entropy', 'bootstrap': False | 0.8040201005025126 | 0.7958115183246074 |
| 300 | 1000 | 'n\_estimators': 100, 'min\_samples\_split': 10, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 90, 'criterion': 'entropy', 'bootstrap': True | 0.8579289644822411 | 0.8555442522889115 |
| 300 | 5000 | 'n\_estimators': 100, 'min\_samples\_split': 2, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 100, 'criterion': 'entropy', 'bootstrap': False | 0.916891689168917 | 0.9182649749188551 |
| 500 | 100 | 'n\_estimators': 100, 'min\_samples\_split': 5, 'min\_samples\_leaf': 4, 'max\_features': 'sqrt', 'max\_depth': 50, 'criterion': 'gini', 'bootstrap': True | 0.8743718592964824 | 0.8780487804878049 |
| 500 | 1000 | 'n\_estimators': 100, 'min\_samples\_split': 2, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 100, 'criterion': 'entropy', 'bootstrap': False | 0.9429714857428715 | 0.9434523809523809 |
| 500 | 5000 | 'n\_estimators': 100, 'min\_samples\_split': 10, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 90, 'criterion': 'entropy', 'bootstrap': True | 0.9534953495349535 | 0.95370831259333 |
| 1000 | 100 | 'n\_estimators': 100, 'min\_samples\_split': 10, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 90, 'criterion': 'entropy', 'bootstrap': True | 0.9949748743718593 | 0.9949238578680203 |
| 1000 | 1000 | 'n\_estimators': 100, 'min\_samples\_split': 10, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 90, 'criterion': 'entropy', 'bootstrap': True | 0.9894947473736868 | 0.9895052473763118 |
| 1000 | 5000 | 'n\_estimators': 100, 'min\_samples\_split': 10, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 90, 'criterion': 'entropy', 'bootstrap': True | 0.9938993899389938 | 0.9938920596775809 |
| 1500 | 100 | 'n\_estimators': 50, 'min\_samples\_split': 2, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 90, 'criterion': 'gini', 'bootstrap': False | 1 | 1 |
| 1500 | 1000 | 'n\_estimators': 100, 'min\_samples\_split': 10, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 90, 'criterion': 'entropy', 'bootstrap': True | 0.9989994997498749 | 0.998997995991984 |
| 1500 | 5000 | 'n\_estimators': 75, 'min\_samples\_split': 2, 'min\_samples\_leaf': 1, 'max\_features': 'auto', 'max\_depth': 40, 'criterion': 'entropy', 'bootstrap': True | 0.9995999599959996 | 0.9995999199839968 |
| 1800 | 100 | 'n\_estimators': 50, 'min\_samples\_split': 2, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 90, 'criterion': 'gini', 'bootstrap': False | 1 | 1 |
| 1800 | 1000 | {'n\_estimators': 50, 'min\_samples\_split': 2, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'max\_depth': 90, 'criterion': 'gini', 'bootstrap': False | 1 | 1 |
| 1800 | 5000 | 'n\_estimators': 75, 'min\_samples\_split': 2, 'min\_samples\_leaf': 1, 'max\_features': 'sqrt', 'max\_depth': 40, 'criterion': 'gini', 'bootstrap': False | 0.9998999899989999 | 0.9998999699909973 |

Gradient Boosting Classifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C | D | Best Parameters | Accuracy | F1 Score |
| 300 | 100 | 'n\_estimators': 75, 'max\_depth': 30, 'loss': 'deviance', 'learning\_rate': 0.5, 'criterion': 'squared\_error' | 0.6080402010050251 | 0.6213592233009709 |
| 300 | 1000 | 'n\_estimators': 75, 'max\_depth': 50, 'loss': 'deviance', 'learning\_rate': 0.1, 'criterion': 'squared\_error' | 0.6803401700850426 | 0.6838198911429985 |
| 300 | 5000 | 'n\_estimators': 75, 'max\_depth': None, 'loss': 'exponential', 'learning\_rate': 0.5, 'criterion': 'squared\_error' | 0.7705770577057706 | 0.7680016181229773 |
| 500 | 100 | 'n\_estimators': 75, 'max\_depth': 50, 'loss': 'deviance', 'learning\_rate': 0.1, 'criterion': 'squared\_error' | 0.6030150753768844 | 0.6220095693779905 |
| 500 | 1000 | 'n\_estimators': 75, 'max\_depth': None, 'loss': 'exponential', 'learning\_rate': 0.5, 'criterion': 'squared\_error' | 0.6983491745872936 | 0.6946835443037975 |
| 500 | 5000 | 'n\_estimators': 75, 'max\_depth': 80, 'loss': 'exponential', 'learning\_rate': 0.5, 'criterion': 'friedman\_mse' | 0.7946794679467947 | 0.7974745980073001 |
| 1000 | 100 | 'n\_estimators': 75, 'max\_depth': 80, 'loss': 'exponential', 'learning\_rate': 0.5, 'criterion': 'friedman\_mse' | 0.7035175879396985 | 0.7203791469194313 |
| 1000 | 1000 | 'n\_estimators': 75, 'max\_depth': 80, 'loss': 'deviance', 'learning\_rate': 0.05, 'criterion': 'friedman\_mse' | 0.8179089544772387 | 0.8263358778625953 |
| 1000 | 5000 | 'n\_estimators': 75, 'max\_depth': None, 'loss': 'exponential', 'learning\_rate': 0.5, 'criterion': 'squared\_error' | 0.8661866186618662 | 0.8677604269618503 |
| 1500 | 100 | 'n\_estimators': 100, 'max\_depth': None, 'loss': 'deviance', 'learning\_rate': 0.01, 'criterion': 'squared\_error' | 0.9045226130653267 | 0.9073170731707318 |
| 1500 | 1000 | 'n\_estimators': 75, 'max\_depth': None, 'loss': 'exponential', 'learning\_rate': 0.5, 'criterion': 'squared\_error' | 0.9379689844922461 | 0.9386745796241345 |
| 1500 | 5000 | 'n\_estimators': 75, 'max\_depth': 30, 'loss': 'deviance', 'learning\_rate': 0.5, 'criterion': 'squared\_error' | 0.9583958395839584 | 0.958771060455897 |
| 1800 | 100 | 'n\_estimators': 75, 'max\_depth': 30, 'loss': 'deviance', 'learning\_rate': 0.5, 'criterion': 'squared\_error' | 0.9597989949748744 | 0.9595959595959596 |
| 1800 | 1000 | 'n\_estimators': 75, 'max\_depth': None, 'loss': 'exponential', 'learning\_rate': 0.5, 'criterion': 'squared\_error' | 0.9769884942471235 | 0.977250247279921 |
| 1800 | 5000 | 'n\_estimators': 75, 'max\_depth': 80, 'loss': 'exponential', 'learning\_rate': 0.5, 'criterion': 'friedman\_mse' | 0.9890989098909891 | 0.9891141516029163 |

5) Questions:

All 4 classifiers were trained multiple times over different random states. The following answers consider the results observed from the other trainings. Results submitted will be reproducible.

1. The best classifier by far is the random forest classifier. They sample a random set of clauses for the weak learners. I think that the rules given by the clauses will be simple because of fixed clause length and when the number of examples and/or features is high, many of the possibilities will be in the dataset. Hence, random forests outperform every other classifier for these datasets.
2. A) For decision trees, random forest and gradient boosting, the increase in training examples leads to increase in accuracy and F1 scores.  
   B) For bagging, the accuracy increases from 100 to 1000 but takes a slight hit from 1000 to 5000.
3. Increase in the number of clauses leads to increase in accuracy and f1 scores for all classifiers. But because of the high accuracies given by random forests, the difference between higher order clauses becomes inconsequential.

Extra Credit:

1. Decision Tree: 0.8788
2. Bagging: 0.9434
3. Random Forest: 0.9701
4. Gradient Boosting: 0.9458

Random forests give the best generalization for MNIST. Random forests essentially decorrelate the tree by introducing a random subset of feature for every tree. And they work well with high dimensional data. Each decision is high variance low bias but since we average the variance, they have moderate variance low bias. Hence, they are better for the MNIST dataset.

Part 2: K means (File Name: KMeans.ipynb)

All images will be displayed when the python notebook is executed. Every K is run 5 times and their means and variances are calculated based on given outputs.

1. It is observed that more or less for higher number of iterations, the centroids will be similar for random initializations.
2. Koala: (Log of variances)

|  |  |  |
| --- | --- | --- |
| K | Means | Variances |
| 2 | 5.8981366209868344 | -9.871291130882316 |
| 5 | 4.422130630112669 | -7.747965941455444 |
| 10 | 4.758954405850295 | -7.130259437832572 |
| 15 | 4.896190903667025 | -5.597818134165285 |
| 20 | 4.968966905025227 | -7.617158373238529 |

1. Penguin:

|  |  |  |
| --- | --- | --- |
| K | Means | Variances |
| 2 | 9.162747306906992 | 0 ( Cannot log) |
| 5 | 7.404852273191958 | -5.895844720720679 |
| 10 | 6.6408273781813225 | -7.314462895449868 |
| 15 | 6.551860025102984 | -4.159760538479097 |
| 20 | 6.653909913061483 | -5.360430613082757 |

1. There is a tradeoff between image quality and compression. The choice of K depends on image quality and compression ratio. A good choice for K will be high image quality and high compression ratio. Foe both images, K =20 seems to be the best choice. If your concern is only the size of compressed image, then K = 2 is the best bet.