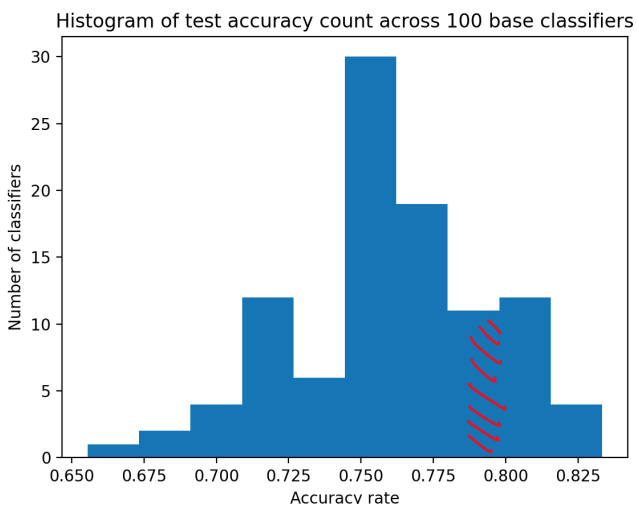
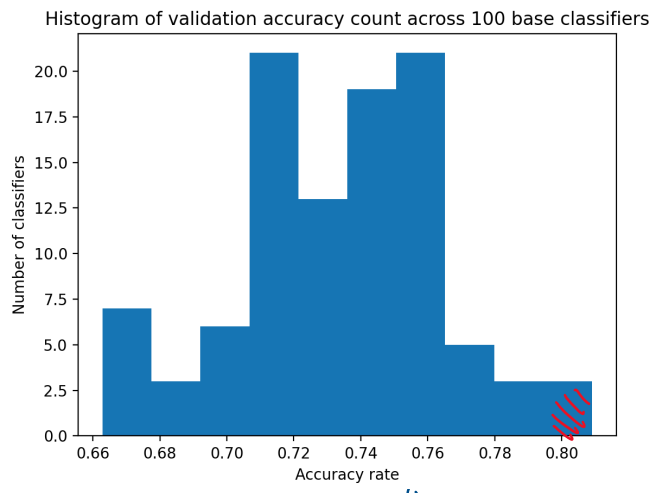
Assignment 1.5

Thursday, February 1, 2024 6:40 PM

5. I a) Histogram of 100 base + 1 ensemble classifiers, code in q_5.py





//// - rate of ensemble classifier

Ensemble classifier:
Validation accuracy rate: 78.65 % } better than average base classifier
Test accuracy rate: 80.00 % }

b) Changing split to 34/33/33:
Validation AR : 82.65 %
Test AR : 73.56 %

Changing max depth: 3, 5, 10
Validation AR : 77.5%. 80.9%. 80.9%.
Test AR: 83.3%. 81.1%. 82.22%.

- increasing the max depth seems to improve the accuracy of the ensemble classifiers
- however, the tradeoff is that it requires greater computational power, leads to overfitting of base classifier which will impact prediction of ensemble classifier.

Changing criterion to "Gini" (default) vs "Entropy"
Validation AR : 80.90 %
Test AR : 78.89 %
- improvement in accuracy rate (very little), may come with minor computation tradeoff as calculations differ

c) It would not change since our prediction is binary. For example, if 52 out of 100 classifier predicts 1, the majority vote would be 1 and the average would be 0.52, which rounds up to 1.

d) The benefit of bagging is that it reduces variance without influencing bias. It allows the reduction of sensitivity to individual datapoints by averaging all base classifiers to make a decision. Even with low accuracy rates in independant classifiers, the probability that the majority of them are wrong is low.
The drawbacks of the bagging method is that the ensemble model becomes more complex, less interpretble, and requires more computational power.
It is possible that the ensemble classifier performs worse than a single classifier, as it is only the average of all base classifiers. Example in part a), the some base classifiers has lower error rate than ensemble.

e) The Random Forest model combines traditional bagging with feature bagging. Each base tree is trained using a different random subsets of features in addition to a different subset of samples from training data. This allows to ensure lower correllation across decision trees, reducing the risk of overfitting.
The difference between the Random Forests and the simple bagging model in part a) is that simple decision trees consider all feature splits while Random Forest only selects a subset of features to split.

II a) Code in q_5.py * dataset is randomized with set seed before execution *
Accuracy rate:
Base: 76.40 %
Decision Tree: 75.28 %
Random Forest: 79.78 %

The acc rate of using Random Forest as the base estimator seems to be the highest. However, it took the longest to execute (finish predicting all the validation set). This makes sense as Random Forest introduces feature bagging, which should lead to higher accuracy but require more computational power

b) Gradient Boosting work by building trees sequentially and fitting new trees using the difference between predicted and actual value of the current model. It's goal is to capture patterns in the data that were not fully represented in the current model. Comparing it to Ada Boost, AdaBoost assigns weights to the errors of the base classifier to make them more influential, while Gradient Boosting fits a new classifier to the errors. They both adjust their models to the errors, but in different ways.

c) Code in q_5.py
Accuracy Rate:
AdaBoost: 76.40 %
GradBoost 79.78 %
XGBoost: 74.16 %

Overall, GradBoost provided the highest accuracy