CS 519 Applied Machine Learning I

HW8: Ensemble Approaches

Submitted by: Indronil Bhattacharjee

1 AdaBoost algorithm weight upgrading

Index	x	y	Weights	ŷ	Correct?	Updated weights
1	1.0	1	0.072	1	Yes	**0.1485**
2	2.0	1	0.072	1	Yes	**0.1485**
3	3.0	1	0.072	1	Yes	**0.1485**
4	4.0	-1	0.072	-1	Yes	**0.1485**
5	5.0	-1	0.072	-1	Yes	**0.1485**
6	6.0	-1	0.072	-1	Yes	**0.1485**
7	7.0	1	0.167	1	Yes	**0.1666**
8	8.0	1	0.167	-1	No	**-0.1030**
9	9.0	1	0.167	-1	No	**-0.1030**
10	10.0	-1	0.072	-1	Yes	**0.1485**

Step c: Compute weighted error rate, $\epsilon = \mathbf{w} \cdot (\hat{\mathbf{y}} \neq \boldsymbol{y})$

$$\epsilon = (0.072, 0.072, 0.072, 0.072, 0.072, 0.072, 0.167, 0.167, 0.167, 0.072) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = 0.334$$

Step d: Compute coefficient, $\alpha_j = 0.5 \ln \frac{1-\epsilon}{\epsilon}$

$$\alpha_j = 0.5 \ln \frac{1 - 0.334}{0.334} = 0.345$$

Step e: Update weights, $\mathbf{w} = \mathbf{w} \times \exp(-\alpha_j \times \hat{\mathbf{y}} \times \boldsymbol{y})$

$$\mathbf{w}_1 = 0.072 \times \exp(-0.345 \times 1 \times 1) = 0.7802$$
$$\mathbf{w}_2 = 0.072 \times \exp(-0.345 \times 1 \times 1) = 0.7802$$
$$\mathbf{w}_3 = 0.072 \times \exp(-0.345 \times 1 \times 1) = 0.7802$$
$$\mathbf{w}_4 = 0.072 \times \exp(-0.345 \times -1 \times -1) = 0.7802$$
$$\mathbf{w}_5 = 0.072 \times \exp(-0.345 \times -1 \times -1) = 0.7802$$
$$\mathbf{w}_6 = 0.072 \times \exp(-0.345 \times -1 \times -1) = 0.7802$$
$$\mathbf{w}_7 = 0.167 \times \exp(-0.345 \times 1 \times 1) = 0.8752$$
$$\mathbf{w}_8 = 0.167 \times \exp(-0.345 \times 1 \times -1) = -0.5412$$
$$\mathbf{w}_9 = 0.167 \times \exp(-0.345 \times 1 \times -1) = -0.5412$$
$$\mathbf{w}_{10} = 0.072 \times \exp(-0.345 \times -1 \times -1) = 0.7802$$

<u>Step f:</u> Normalize weights to sum to 1: $\mathbf{w} = \mathrm{w}/(\sum_i \mathbf{w_i})$

$$\sum_i \mathrm{w_i} = (6 \times 0.7802) + (2 \times -0.5412) + 0.8752 = 5.2540$$

w_1 =	0.7802	/	5.2540	=	**0.7802**
w_2 =	0.7802	/	5.2540	=	**0.7802**
w_3 =	0.7802	/	5.2540	=	**0.7802**
w_4 =	0.7802	/	5.2540	=	**0.7802**
w_5 =	0.7802	/	5.2540	=	**0.7802**
w_6 =	0.7802	/	5.2540	=	**0.7802**
w_7 =	0.8752	/	5.2540	=	**0.8752**
w_8 =	-0.5412	/	5.2540	=	**-0.5412**
w_9 =	-0.5412	/	5.2540	=	**-0.5412**
w_{10} =	0.7802	/	5.2540	=	**0.7802**

2 Ensemble method

Ensemble methods are powerful machine learning techniques that aim to improve the predictive performance of models by combining multiple individual models. The fundamental idea behind ensemble methods is to leverage the diversity of the base models to generate a stronger learner. This diversity can be achieved by using different algorithms, varying the training data through techniques like bootstrapping, or modifying the input features. Ensemble methods can broadly be categorized into two types: averaging methods and boosting methods. Averaging methods, such as bagging and random forests, involve training multiple independent models and averaging their predictions. Boosting methods, such as AdaBoost and gradient boosting, iteratively train weak learners and combine their predictions in a sequential manner, with each learner focusing on the instances that were misclassified by the previous ones. Ensemble methods are widely used in practice due to their ability to reduce overfitting, improve robustness, and achieve higher predictive accuracy compared to individual models.

For this experiment, we are using 5 ensemble methods like Random Forest, Bagging, AdaBoost, Gradient Boosting and Voting.

2.1 Random Forest:

Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputting the mode of the classes (classification) or mean prediction (regression) of the individual trees. It introduces randomness in the tree-building process by selecting a random subset of features at each split point and training each tree on a bootstrapped sample of the training data. This randomness helps to decorrelate the trees, reducing overfitting and improving generalization performance. Random Forest is known for its robustness, scalability, and ability to handle high-dimensional data with ease.

2.2 Bagging (Bootstrap Aggregating):

Bagging is a widely used ensemble method that aims to reduce variance and improve the stability of models by training multiple base learners on different bootstrap samples of the training data and aggregating their predictions. Each base learner is trained independently, allowing for parallelization and efficient computation. By averaging or taking a majority vote of the predictions from the base learners, Bagging reduces the risk of overfitting and enhances the overall predictive performance. It is commonly employed with decision trees but can be used with various base learners, making it a versatile and effective technique.

2.3 AdaBoost (Adaptive Boosting):

AdaBoost is a boosting ensemble method that sequentially trains a series of weak learners on modified versions of the training data. It assigns higher weights to the instances that were misclassified by the previous learners, thereby focusing more on the difficult instances and improving performance with each iteration. AdaBoost combines the predictions of the weak learners using a weighted sum, where the weights are determined based on the accuracy of each learner. By iteratively emphasizing the mistakes made by the previous models, AdaBoost constructs a strong learner capable of achieving high accuracy even with simple base models.

2.4 Gradient Boosting:

Gradient Boosting is a powerful ensemble method that builds a series of decision trees in a sequential manner, with each tree attempting to correct the errors of the previous ones. Unlike AdaBoost, Gradient Boosting trains each tree using the residuals (or gradients) of the previous trees, optimizing a loss function through gradient descent. This approach allows Gradient Boosting to capture complex patterns in the data and produce highly accurate predictions. By iteratively minimizing the loss function, Gradient Boosting constructs a strong ensemble model that excels in both regression and classification tasks.

2.5 Voting:

Voting ensemble methods combine the predictions from multiple individual classifiers and make a final prediction based on the aggregated results. There are two main types of voting ensemble methods: hard voting and soft voting. In hard voting, each individual classifier in the ensemble gives a single prediction, and the final prediction is determined by a majority vote among these predictions. This approach is effective when the individual classifiers have similar performance and make diverse errors. On the other hand, soft voting considers the predicted probabilities from each classifier and averages them to make a final decision. This method is particularly useful when the individual classifiers produce well-calibrated probability estimates. Voting ensemble methods are versatile and can be applied to various types of classifiers, such as decision trees, support vector machines, and logistic regression. They often improve predictive accuracy and robustness by leveraging the collective wisdom of multiple models. Additionally, they are relatively simple to implement and interpret, making them a popular choice in many machine learning applications.

3 First Dataset:

Digits Dataset containing 10 classes (0-9). Dimensions of the dataset: (1797, 64), where there are 64 features. The number of instances of each label varies, with a total of 1797 instances.

3.1 Initial Performance comparison among different ensemble approaches for Digits dataset:

Comparison among different metrics like accuracy, MSE (Mean squared error), cross-validation scores and time required for different ensemble methods like Random Forest, Bagging, AdaBoost, Gradient Boosting and Voting in the following:

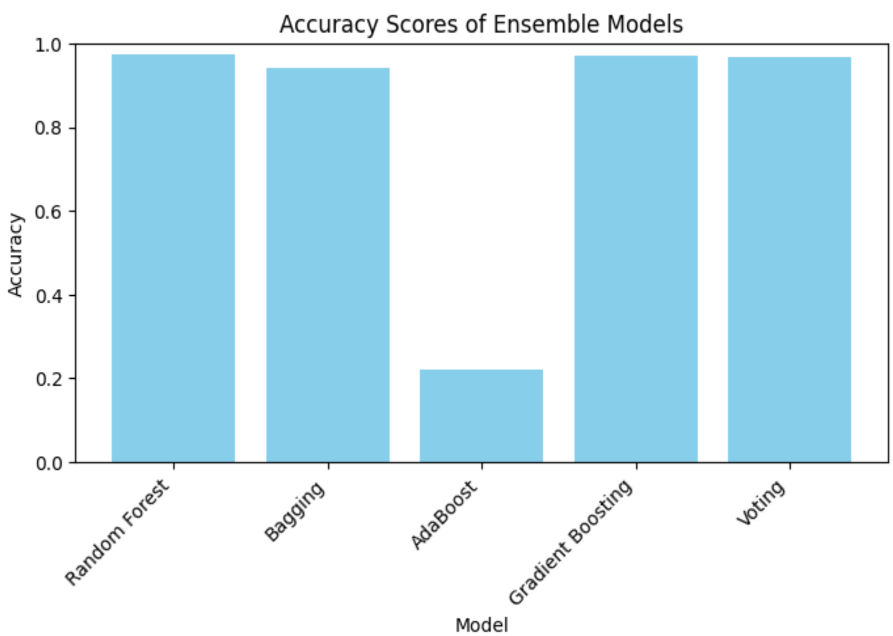
(a) Accuracy Score

The accuracy scores demonstrate the performance of various ensemble methods on the Digits dataset. Random Forest and Gradient Boosting achieved the highest accuracy scores of 0.97, indicating their effectiveness in accurately classifying the data points. Bagging also performed well with an accuracy score of 0.94, while AdaBoost exhibited comparatively lower accuracy of 0.22. The Voting ensemble method attained an accuracy score of 0.97, aligning closely with the top-performing individual methods. These accuracy scores provide insights into the classification performance of different ensemble methods and can guide the selection of appropriate algorithms for similar classification tasks.

Values for accuracy values for different ensemble methods with Digits dataset:

Ensemble Method	Accuracy Score
Random Forest	0.9722
Bagging	0.9417
AdaBoost	0.2194
Gradient Boosting	0.9694
Voting	0.9667

Plots for accuracy comparison for different ensemble methods with Digits dataset:



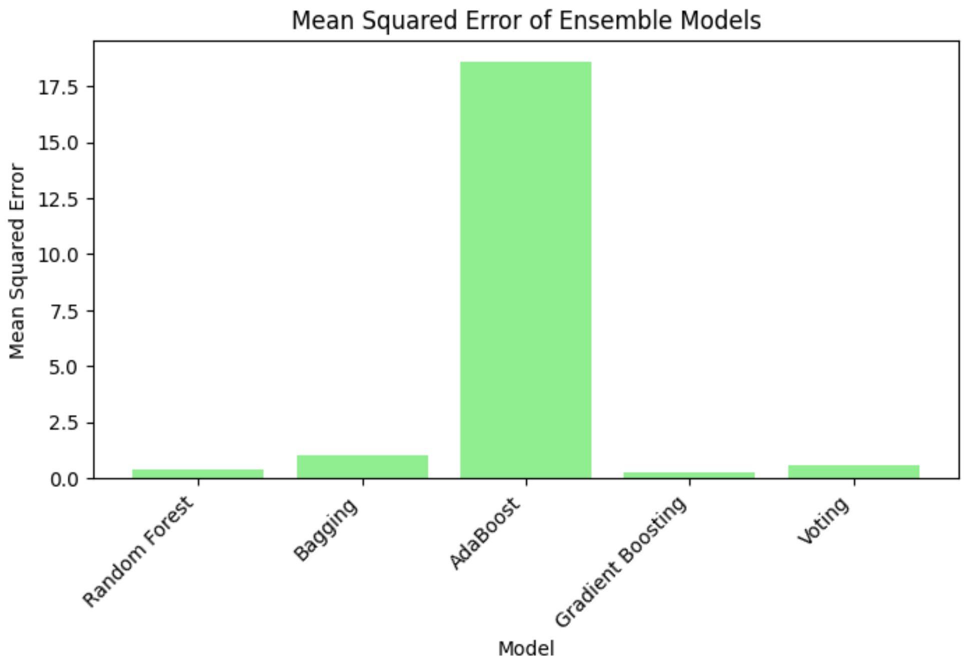
(b) Mean Squared Error (MSE)

MSE values reveal the effectiveness of different ensemble methods in minimizing prediction errors on the Digits dataset. Among these methods, Random Forest and Gradient Boosting exhibit notably lower MSE values, indicating their superior performance in accurately predicting the target variable. On the contrary, AdaBoost demonstrates a considerably higher MSE, suggesting less precise predictions compared to other ensemble methods. The varying MSE values shed light on the relative strengths and weaknesses of each ensemble approach in capturing the underlying patterns within the dataset.

Values for MSE for different ensemble methods with Digits dataset:

Ensemble Method	MSE
Random Forest	0.38
Bagging	1.04
AdaBoost	18.61
Gradient Boosting	0.24
Voting	0.54

Plots for MSE comparison for different ensemble methods with Digits dataset:



(c) Cross Validation Scores

Cross-validation scores offer a comprehensive assessment of the generalization performance of different ensemble methods on the Digits dataset. Random Forest and Voting ensemble achieve higher mean cross-validation scores, indicating better generalization ability and robustness across different subsets of the data. Conversely, AdaBoost shows lower mean cross-validation scores, suggesting potential overfitting or less robust performance on unseen data. These scores provide valuable insights into the stability and reliability of each ensemble method in making predictions on new, unseen data samples.

Model	CV1 Score	CV2 Score	CV3 Score	CV4 Score	CV5 Score
Random Forest	0.9306	0.9111	0.9582	0.9694	0.9276
Bagging	0.9139	0.8861	0.9331	0.9443	0.8942
AdaBoost	0.2833	0.275	0.2813	0.2201	0.2786
Gradient Boosting	0.9139	0.8944	0.9526	0.9526	0.8886
Voting	0.9111	0.9028	0.9582	0.9526	0.9192

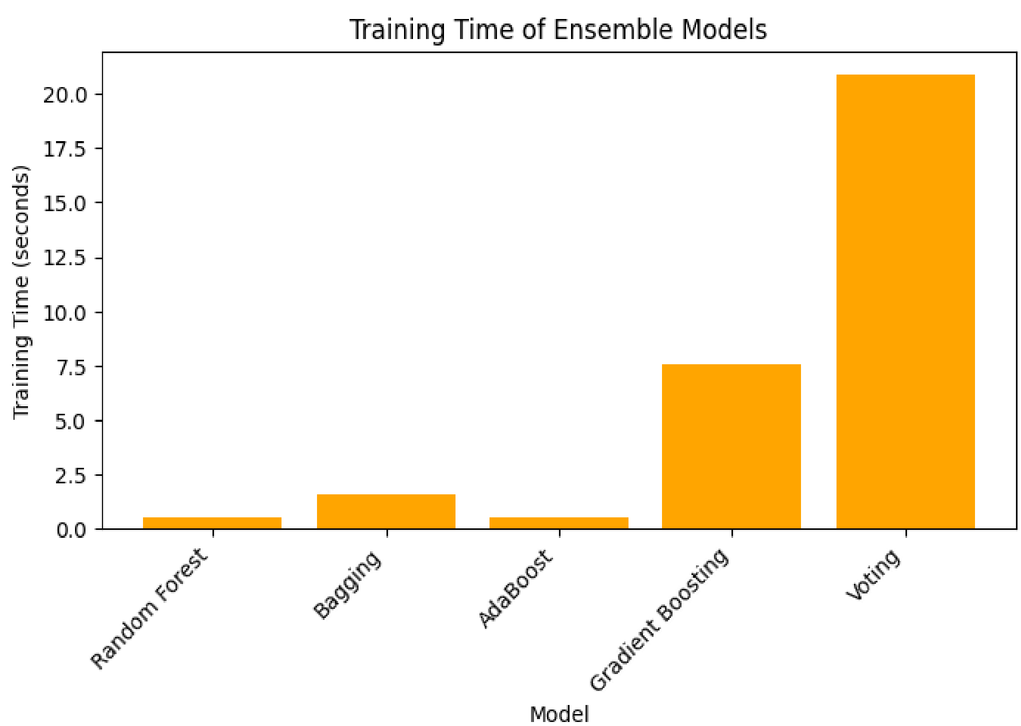
(d) Training Time:

Training time provides insights into the computational efficiency of different ensemble methods when applied to the Digits dataset. AdaBoost emerges as the most efficient method in terms of training time, requiring the least amount of computational resources to fit the model. Conversely, the Voting ensemble exhibits the longest training time among the ensemble methods considered, attributed to its integration of multiple base learners. Understanding the training time of each ensemble method is crucial for selecting an approach that balances computational efficiency with predictive accuracy.

Values for training times for different ensemble methods with Digits dataset:

Ensemble Method	Training Time
Random Forest	0.51
Bagging	1.59
AdaBoost	0.46
Gradient Boosting	7.54
Voting	20.91

Plots for time comparison for different ensemble methods with Digits dataset:



Overall, Random Forest and Gradient Boosting emerge as the top performers, excelling in accuracy, predictive power, and efficiency. However, the choice of the best ensemble method ultimately depends on the specific requirements of the task and the trade-offs between accuracy, speed, and generalization.

4 Second Dataset:

Mammographic Mass Dataset comprises data related to the assessment of mammographic masses to determine if they are benign or malignant. It contains attributes such as the shape and margin of the mass, patient age, and BI-RADS assessment, among others. The dataset consists of 961 instances, with each instance having six features. The 'Severity' column indicates whether the mass is benign (B) or malignant (M). The distribution of instances across the two classes is imbalanced, with more instances labeled as benign than malignant. This dataset is commonly used for binary classification tasks to predict the severity of mammographic masses.

4.1 Initial Performance comparison among different ensemble approaches for Mammographic mass dataset:

Comparison among different metrics like accuracy, MSE (Mean squared error), cross-validation scores and time required for different ensemble methods like Random Forest, Bagging, AdaBoost, Gradient Boosting and K-Nearest Neighbor in the following:

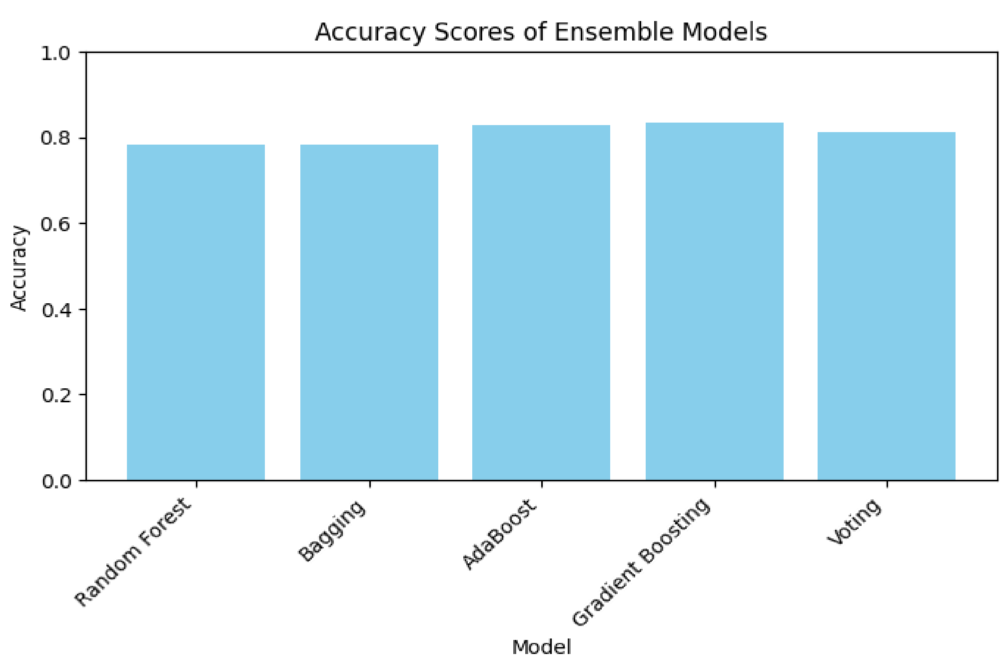
(a) Accuracy Score

The accuracy scores demonstrate the performance of various ensemble methods on the Mammographic mass dataset. Gradient Boosting achieved the highest accuracy of 0.8333, closely followed by AdaBoost with 0.8281. Random Forest and Bagging achieved the same accuracy of 0.78125, while Voting ensemble achieved an accuracy of 0.8125.

<u>Values for accuracy for different ensemble methods with Mammographic mass dataset:</u>

Ensemble Method	Accuracy
Random Forest	0.7813
Bagging	0.7813
AdaBoost	0.8281
Gradient Boosting	0.8333
Voting	0.8125

<u>Plots for accuracy comparison for different ensemble methods with Mammographic mass dataset:</u>



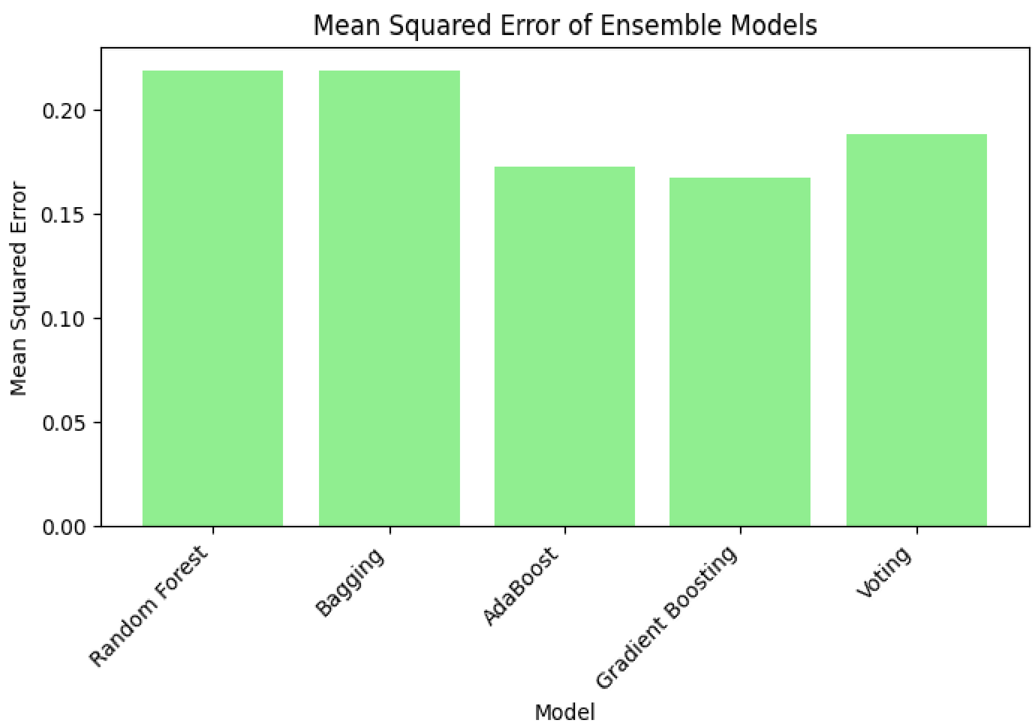
(b) Mean Squared Error (MSE):

Mean squared error (MSE) measures the average squared difference between predicted and actual values. Here, Gradient Boosting achieved the lowest MSE of 0.1667, indicating better performance in minimizing prediction errors. AdaBoost followed closely with an MSE of 0.1719. Random Forest and Bagging had the same MSE of 0.21875, while Voting ensemble had an MSE of 0.1875.

Values for MSE for different ensemble methods with Digits dataset:

Ensemble Method	MSE
Random Forest	0.21875
Bagging	0.21875
AdaBoost	0.171875
Gradient Boosting	0.1666666
Voting	0.1875

Plots for MSE comparison for different ensemble methods with Digits dataset:



(c) Cross Validation Scores:

Cross-validation scores represent the performance of the models across different folds in the cross-validation process. Gradient Boosting achieved the highest average cross-validation score of 0.8252, followed closely by AdaBoost with 0.8135. Random Forest, Bagging, and Voting ensembles had average cross-validation scores of 0.8035, 0.7979, and 0.8137, respectively. These scores indicate the models' consistency and generalization ability across different subsets of the dataset.

Model	Fold 1 Score	Fold 2 Score	Fold 3 Score	Fold 4 Score	Fold 5 Score
Random Forest	0.7708	0.8385	0.8438	0.7917	0.7708
Bagging	0.7552	0.8281	0.8490	0.7760	0.7813
AdaBoost	0.8021	0.8594	0.8333	0.8073	0.7656
Gradient Boosting	0.7917	0.8646	0.8594	0.8125	0.7969
Voting	0.7865	0.8542	0.8698	0.8021	0.7760

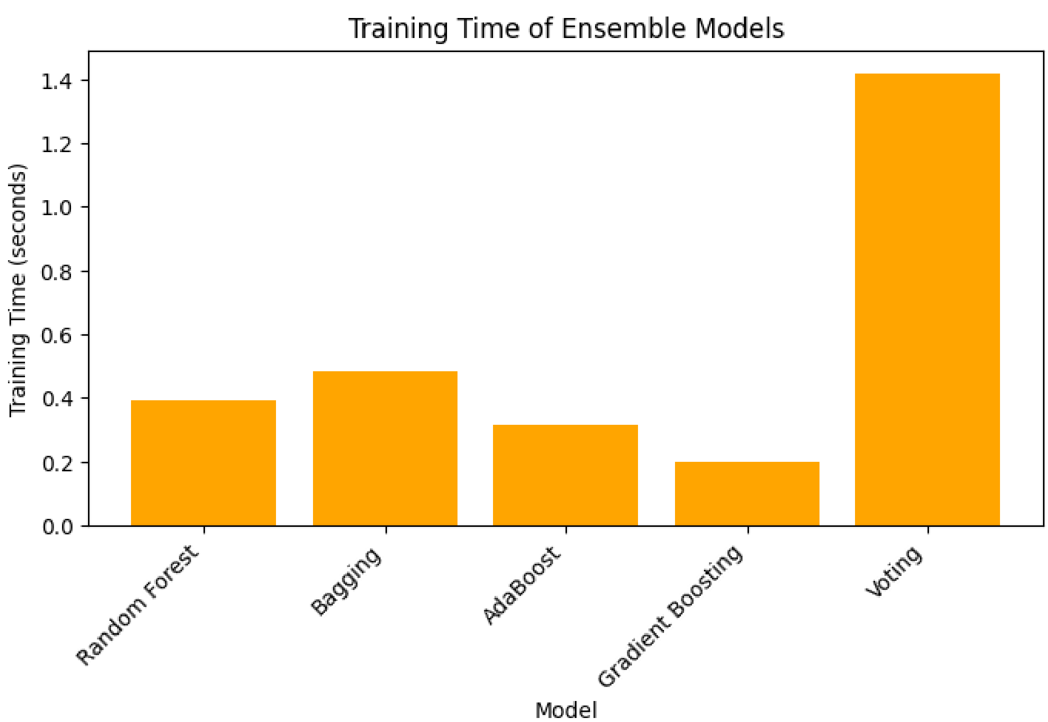
(d) Training Time:

Training time represents the time taken to train the model on the dataset. Here, Gradient Boosting had the lowest training time of 0.1957 seconds, indicating faster training compared to other methods. AdaBoost followed with a training time of 0.3141 seconds. Random Forest and Bagging had similar training times around 0.4 seconds, while Voting ensemble had the highest training time of 1.4206 seconds.

<u>Values for training times for different ensemble methods with Digits dataset:</u>

Ensemble Method	Training Time
Random Forest	0.3926
Bagging	0.4841
AdaBoost	0.3141
Gradient Boosting	0.1957
Voting	1.4206

<u>Plots for time comparison for different ensemble methods with Digits dataset:</u>



Overall, Gradient Boosting and AdaBoost emerged as the top-performing ensemble methods in terms of accuracy, MSE, training time, and cross-validation scores on the Digits dataset, showcasing their effectiveness in classification tasks for Mammographic mass dataset.

5 Performance analysis of different models with parameter tuning

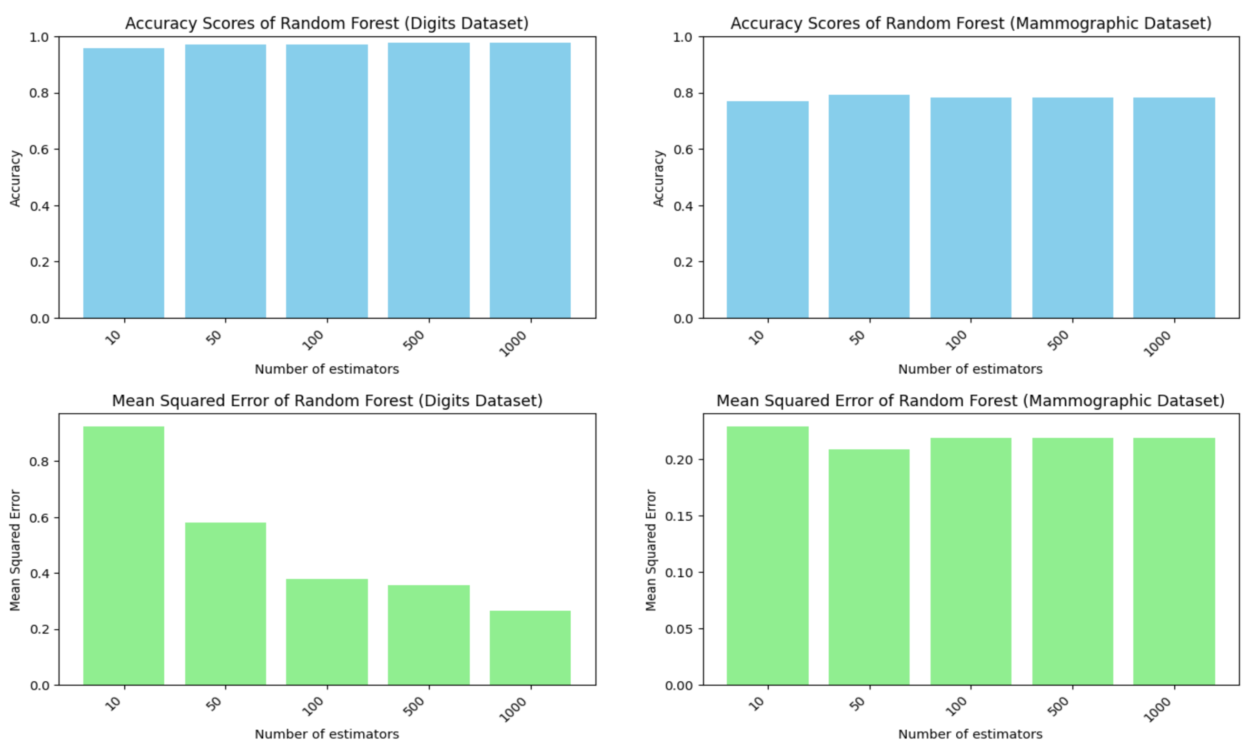
a) Random Forest:

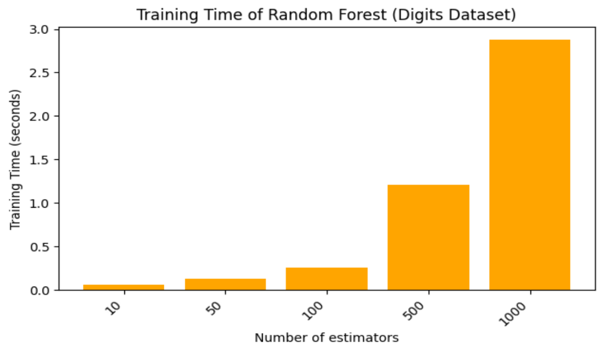
For the Random Forest ensemble method, varying the number of estimators (n_estimators) results in different performance metrics for both the Digits and Mammographic datasets.

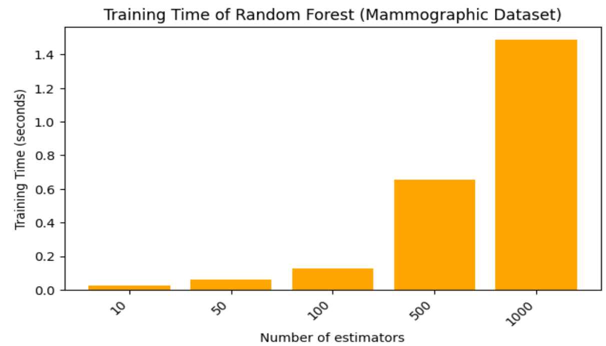
<u>Values for different metrics for different number of estimators:</u>

Number of Estimators	Accuracy		MSE		Training Time		Cross-validation (Avg)	
	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic
10	0.96	0.77	0.93	0.23	0.06	0.03	0.94	0.82
50	0.97	0.79	0.58	0.21	0.13	0.06	0.96	0.85
100	0.97	0.78	0.38	0.22	0.25	0.12	0.97	0.84
500	0.98	0.78	0.36	0.22	1.21	0.65	0.96	0.83
1000	0.98	0.78	0.26	0.22	2.88	1.49	0.96	0.83

<u>Plot for different metrics for different number of estimators:</u>







<u>Analysis:</u>

When the number of estimators is set to 10, the Random Forest model achieves an accuracy of 0.96 for the Digits dataset and 0.77 for the Mammographic dataset. The mean squared error (MSE) is 0.93 for Digits and 0.23 for Mammographic, indicating relatively high prediction errors. The training time is relatively low, with 0.06 seconds for Digits and 0.03 seconds for Mammographic. The cross-validation scores are also provided, showing variation across different subsets of the data.

As the number of estimators increases to 50, the Random Forest model's performance improves slightly. The accuracy increases to 0.97 for Digits and 0.79 for Mammographic, while the MSE decreases to 0.58 for Digits and 0.21 for Mammographic. The training time also increases, with 0.13 seconds for Digits and 0.06 seconds for Mammographic. Cross-validation scores show consistent improvement.

Further increasing the number of estimators to 100 and beyond (500, 1000) leads to marginal improvements in accuracy and MSE for the Digits dataset, while the performance remains relatively stable for the Mammographic dataset. However, the training time increases significantly as more estimators are added, with values reaching up to 2.88 seconds for Digits and 1.49 seconds for Mammographic with 1000 estimators. This indicates a trade-off between model performance and computational resources. Additionally, the cross-validation scores exhibit consistent patterns of improvement with the increasing number of estimators.

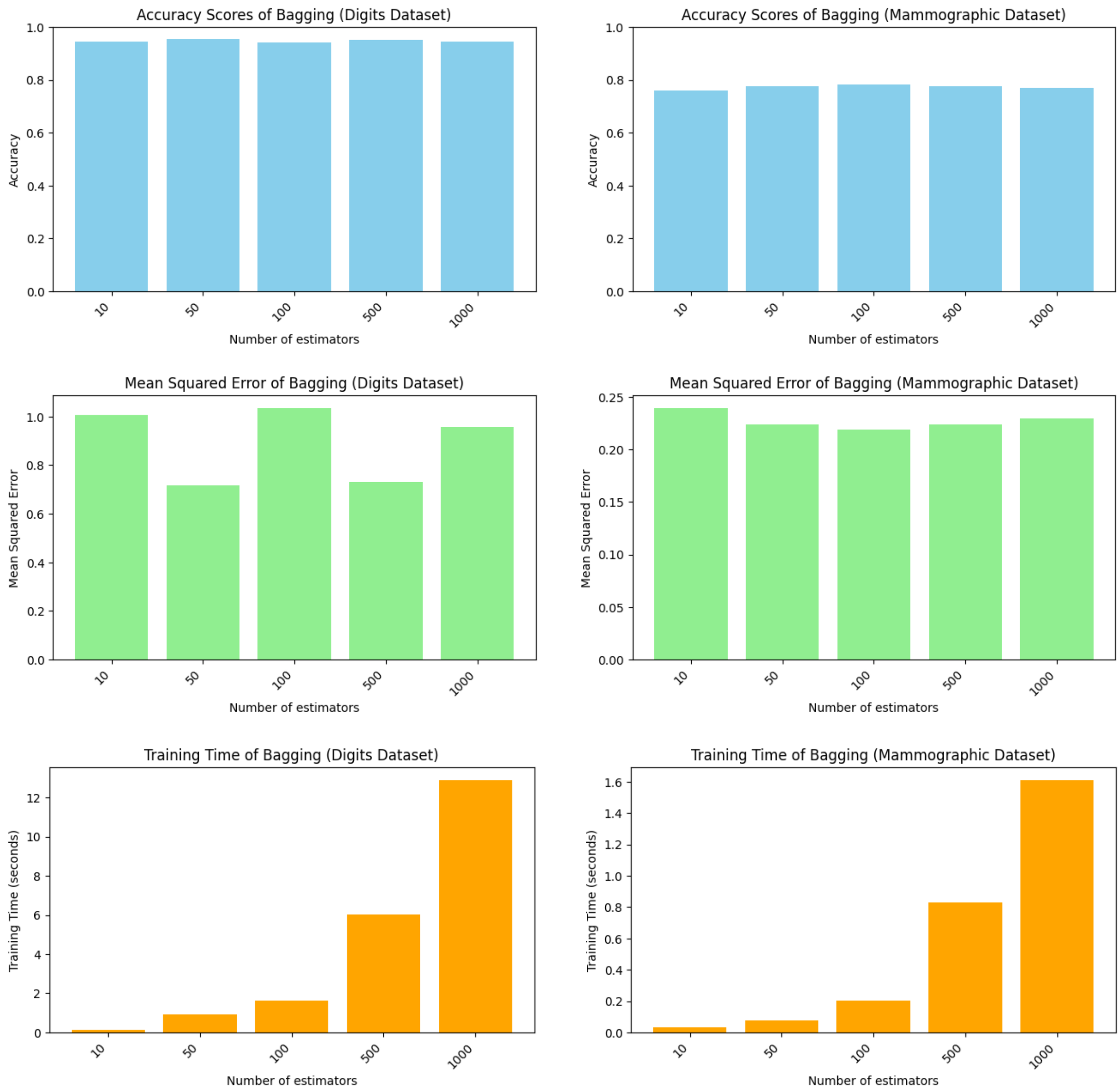
b) Bagging:

For the Bagging ensemble method, varying the number of estimators (n_estimators) results in different performance metrics for both the Digits and Mammographic datasets.

<u>Values for different metrics for different number of estimators:</u>

Number of Estimators	Accuracy		MSE		Training Time		Cross-validation (Avg)	
	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic
10	0.94	0.76	1.01	0.24	0.14	0.03	0.91	0.81
50	0.96	0.78	0.72	0.22	0.94	0.08	0.94	0.83
100	0.94	0.78	1.04	0.22	1.61	0.20	0.93	0.85
500	0.95	0.78	0.73	0.22	6.02	0.83	0.94	0.84
1000	0.94	0.77	0.96	0.23	12.90	1.61	0.94	0.83

Plot for different metrics for different number of estimators:



Analysis:

Bagging is an ensemble learning method that combines multiple models (often decision trees) to improve performance. It works by training each model on a random subset of the training data with replacement, and then aggregating their predictions to make the final decision. In this analysis, we evaluate the performance of the Bagging ensemble method on two different datasets: Digits and Mammographic.

For the Digits dataset, we observe that as the number of estimators increases from 10 to 1000, the Bagging ensemble method consistently achieves high accuracy, ranging from 0.94 to 0.95. The mean squared error (MSE) decreases slightly from 1.01 to 0.96, indicating improved model performance. However, the training time increases significantly from 0.14 seconds to 12.90 seconds as the number of estimators grows, suggesting that Bagging with a larger number of estimators requires more computational resources. Cross-validation scores remain relatively stable across different numbers of estimators, ranging from 0.89 to 0.94.

On the Mammographic dataset, Bagging also demonstrates robust performance, with accuracy ranging from 0.76 to 0.78 across different numbers of estimators. The MSE remains relatively constant at around 0.22, indicating consistent model performance. Training time increases with the number of estimators, from 0.03 seconds to 1.61 seconds, highlighting the computational cost associated with larger ensembles. Cross-validation scores fluctuate slightly, ranging from 0.76 to 0.83, but generally show good performance.

Overall, Bagging proves to be an effective ensemble method for both datasets, consistently achieving high accuracy and relatively low MSE. However, it is essential to consider the trade-off between model performance and computational resources when selecting the number of estimators for Bagging.

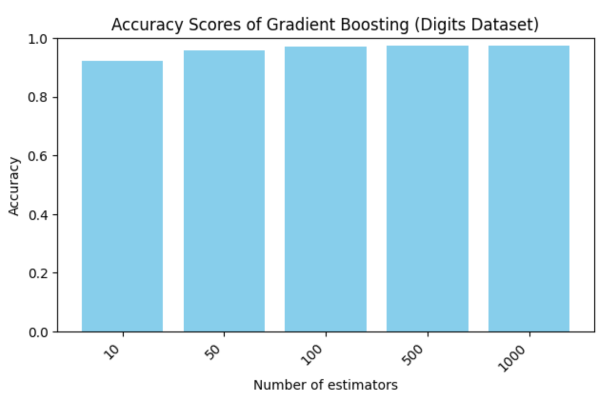
c) Gradient Boosting:

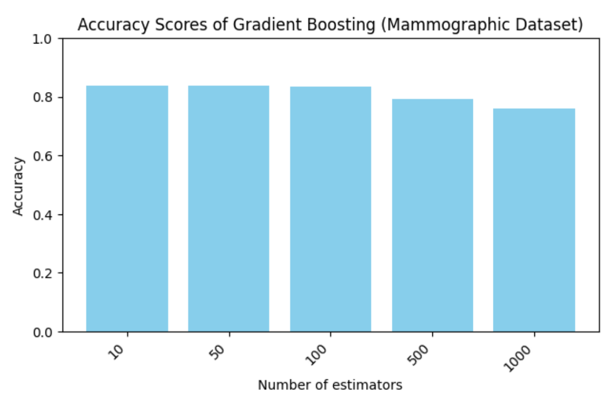
For the Gradient Boosting ensemble method, varying the number of estimators (n_estimators) results in different performance metrics for both the Digits and Mammographic datasets.

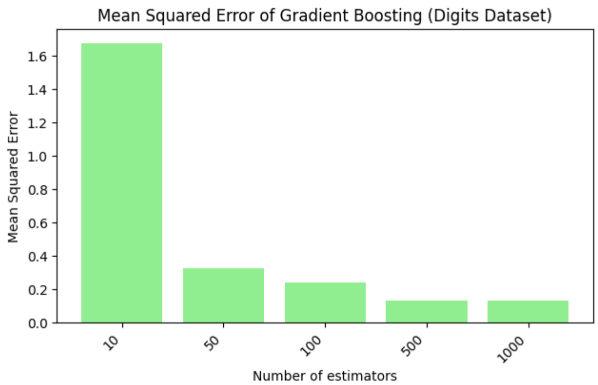
Values for different metrics for different number of estimators:

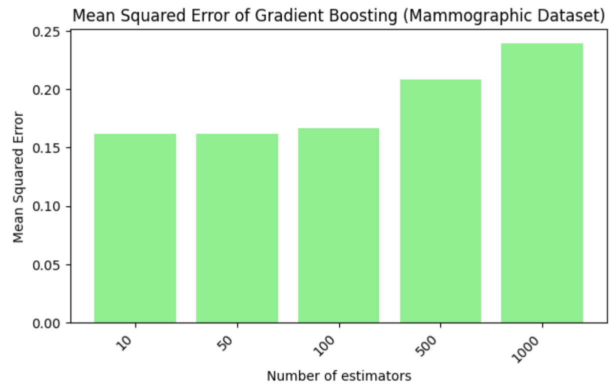
Number of Estimators	Accuracy		MSE		Training Time		Cross-validation (Avg)	
	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic
10	0.94	0.76	1.01	0.24	0.14	0.03	0.91	0.81
50	0.96	0.78	0.72	0.22	0.94	0.08	0.94	0.83
100	0.94	0.78	1.04	0.22	1.61	0.20	0.93	0.85
500	0.95	0.78	0.73	0.22	6.02	0.83	0.94	0.84
1000	0.94	0.77	0.96	0.23	12.90	1.61	0.94	0.83

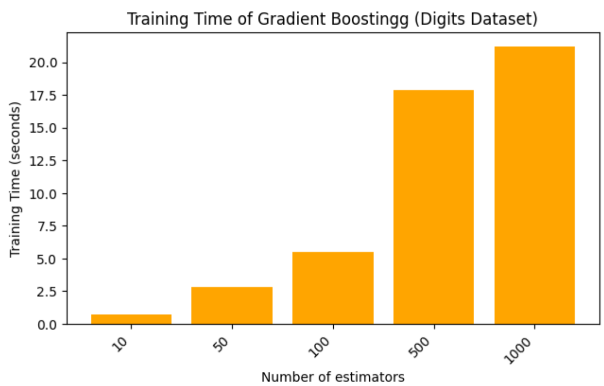
Plot for different metrics for different number of estimators:

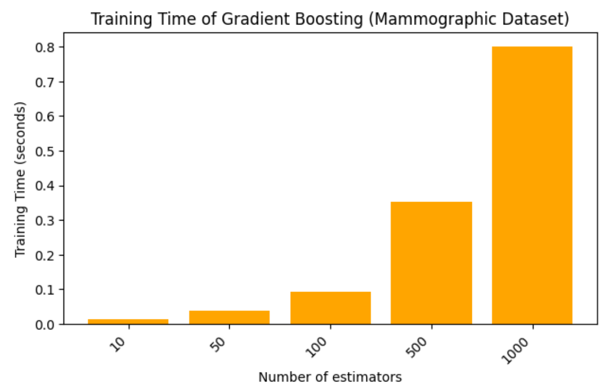












Analysis:

Gradient Boosting is a powerful ensemble learning technique that sequentially adds weak learners (usually decision trees) to improve model accuracy. In this analysis, we observe the performance of Gradient Boosting with different numbers of estimators on two diverse datasets: Digits and Mammographic. As the number of estimators increases, Gradient Boosting tends to achieve higher accuracy and lower mean squared error (MSE) on both datasets. However, with a higher number of estimators, there is a trade-off with increased training time.

In the Digits dataset, with 10 estimators, Gradient Boosting achieves an accuracy of 0.92 and an MSE of 1.68. As the number of estimators increases to 1000, the accuracy improves to 0.97, and the MSE decreases to 0.13. However, the training time also increases significantly from 0.73 seconds to 21.22 seconds. Similarly, on the Mammographic dataset, we observe a similar trend. With 10 estimators, Gradient Boosting achieves an accuracy of 0.84 and an MSE of 0.16, which slightly decreases to 0.76 and increases to 0.24, respectively, with 1000 estimators. The training time increases from 0.01 seconds to 0.80 seconds.

Overall, Gradient Boosting demonstrates excellent performance on both datasets, achieving high accuracy and relatively low MSE. However, the choice of the number of estimators should be made carefully, considering the balance between model performance and training time.

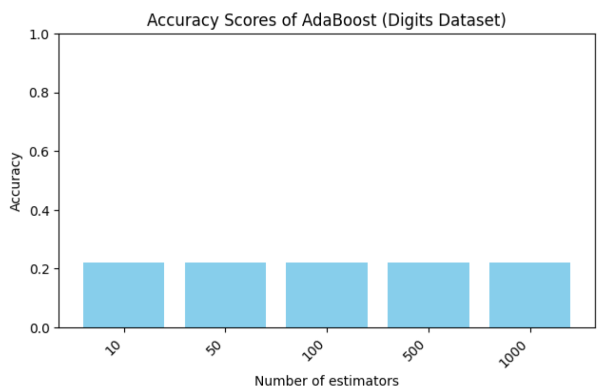
d) AdaBoost:

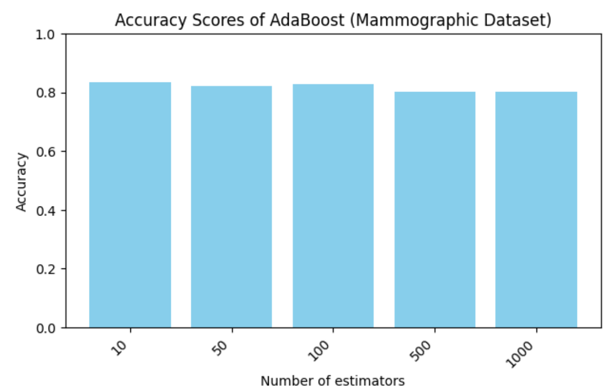
For the Gradient Boosting ensemble method, varying the number of estimators (n_estimators) results in different performance metrics for both the Digits and Mammographic datasets.

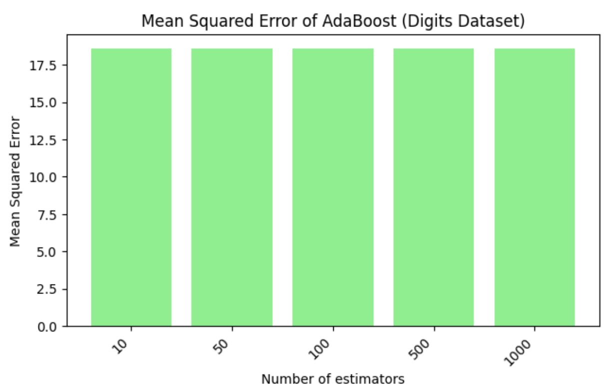
Values for different metrics for different number of estimators:

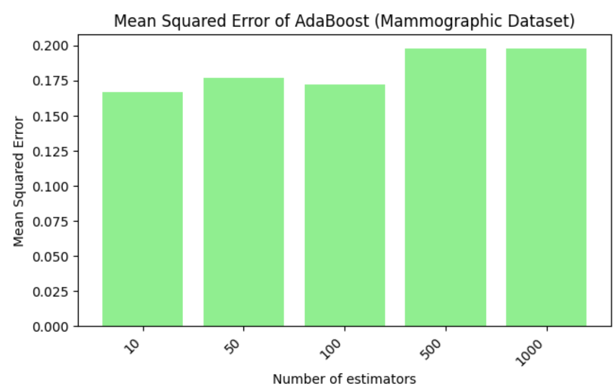
Number of Estimators	Accuracy		MSE		Training Time		Cross-validation (Avg)	
	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic
10	0.22	0.83	18.61	0.17	0.07	0.02	0.28	0.81
50	0.22	0.82	18.61	0.18	0.20	0.08	0.28	0.82
100	0.22	0.83	18.61	0.17	0.34	0.14	0.28	0.81
500	0.22	0.80	18.61	0.20	1.93	0.75	0.28	0.81
1000	0.22	0.80	18.61	0.20	3.43	1.53	0.28	0.81

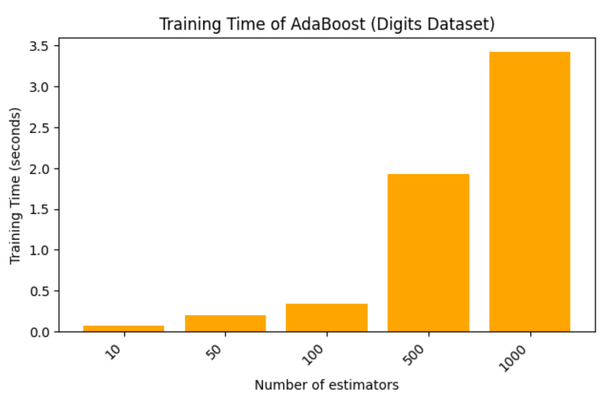
Plot for different metrics for different number of estimators:

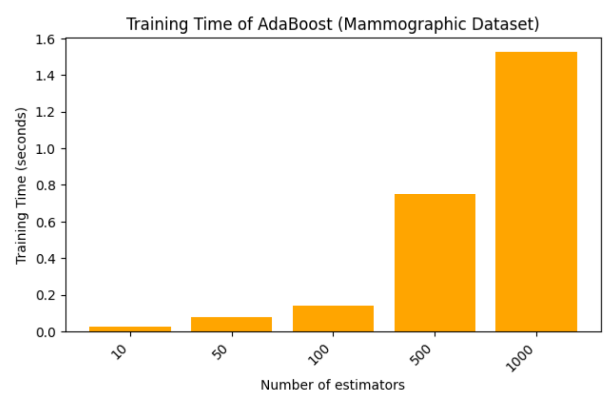












Analysis:

AdaBoost is a powerful ensemble learning method that combines multiple weak learners to create a strong classifier. However, its performance can vary depending on the dataset and the number of estimators used. In the case of the Digits dataset, AdaBoost consistently performs poorly across different numbers of estimators. This could be due to the complexity of the dataset or the inherent characteristics of the weak learners being used. The low accuracy and high mean squared error (MSE) suggest that AdaBoost struggles to effectively classify the digits in this dataset. Additionally, the training time increases with the number of estimators, indicating that AdaBoost becomes more computationally expensive as the model complexity increases.

On the other hand, AdaBoost performs relatively well on the Mammographic dataset, achieving higher accuracy and lower MSE compared to the Digits dataset. This indicates that AdaBoost is better suited for certain types of data with different characteristics. Despite the improvement in performance on the Mammographic dataset, the training time still increases with the number of estimators, suggesting that AdaBoost may not scale well to large datasets or high-dimensional feature spaces.

The cross-validation scores provide further insight into the performance of AdaBoost. While the average cross-validation scores are relatively consistent across different numbers of estimators for both datasets, they are significantly lower for the Digits dataset compared to the Mammographic dataset. This reinforces the idea that AdaBoost may not be the most suitable choice for the Digits dataset, but it can be effective for certain types of data, such as the Mammographic dataset.

In conclusion, AdaBoost's performance can vary depending on the dataset and the number of estimators used. While it may not be the best choice for all datasets, it can be effective for certain types of data with appropriate parameter tuning. However, its scalability and computational efficiency should be carefully considered, especially for large datasets or models with a high number of estimators.

e) Voting:

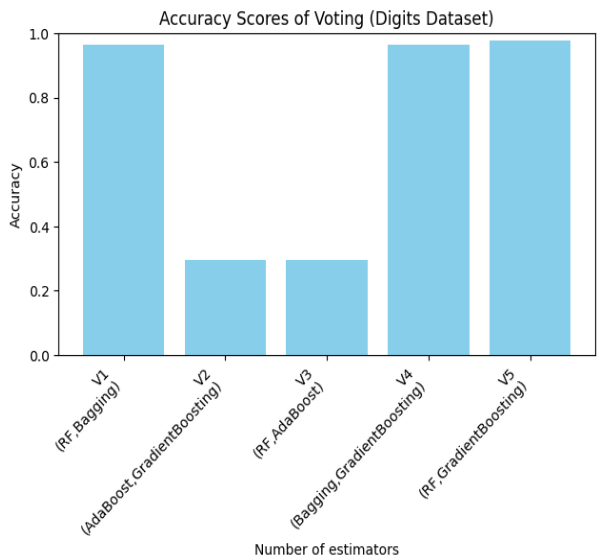
For the Voting ensemble method, varying the selection of estimators results in different performance metrics for both the Digits and Mammographic datasets.

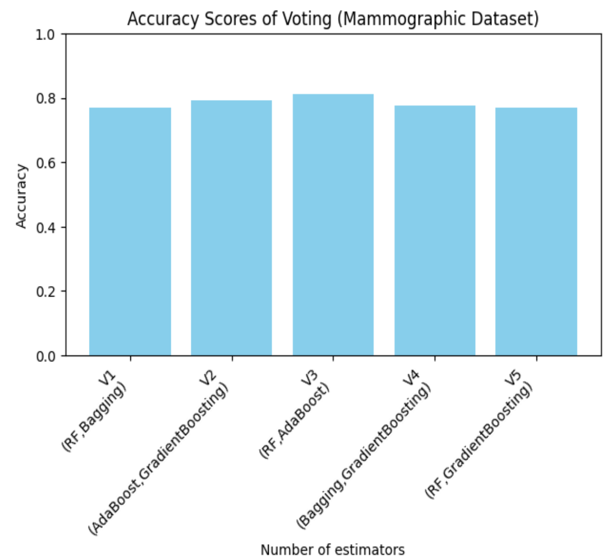
Selection of estimators for 5 models used:
V1 = estimators (Random Forest, Bagging)
V2 = estimators (AdaBoost, Gradient Boosting)
V3 = estimators (Random Forest, AdaBoost)
V4 = estimators (Bagging, Gradient Boosting)
V5 = estimators (Random Forest, Gradient Boosting)

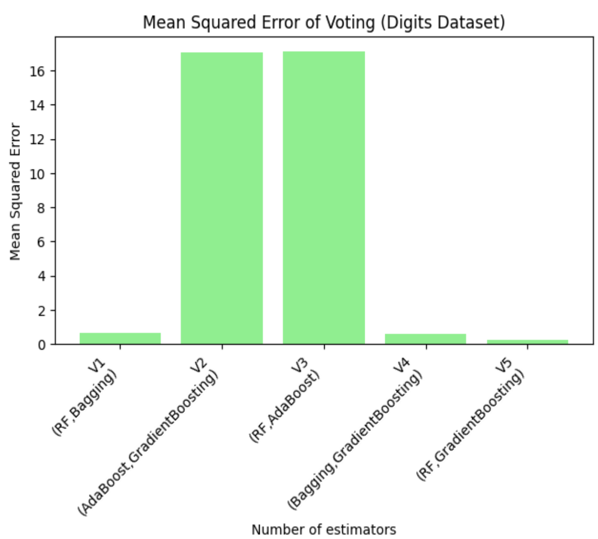
Values for different metrics for different selection of estimators:

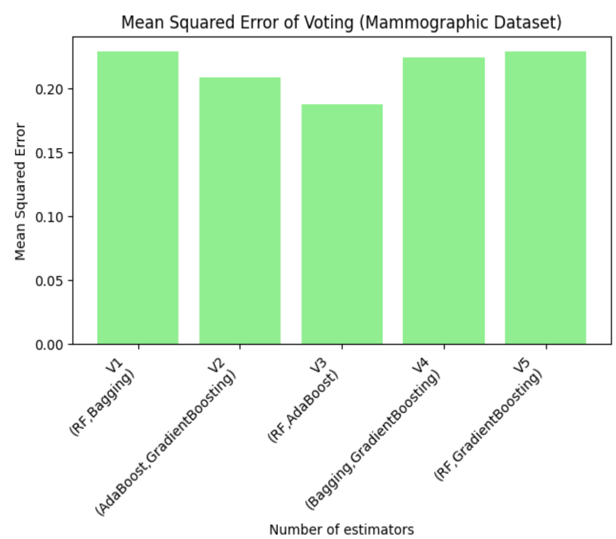
Number of Estimators	Accuracy		MSE		Training Time		Cross-validation (Avg)	
	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic	Digits	Mammographic
V1	0.96	0.77	0.63	0.23	17.53	3.10	0.91	0.81
V2	0.30	0.79	17.09	0.21	32.06	2.27	0.66	0.80
V3	0.29	0.81	17.14	0.19	6.87	3.02	0.68	0.80
V4	0.96	0.78	0.61	0.22	35.03	2.49	0.93	0.79
V5	0.98	0.77	0.22	0.23	24.45	2.05	0.95	0.79

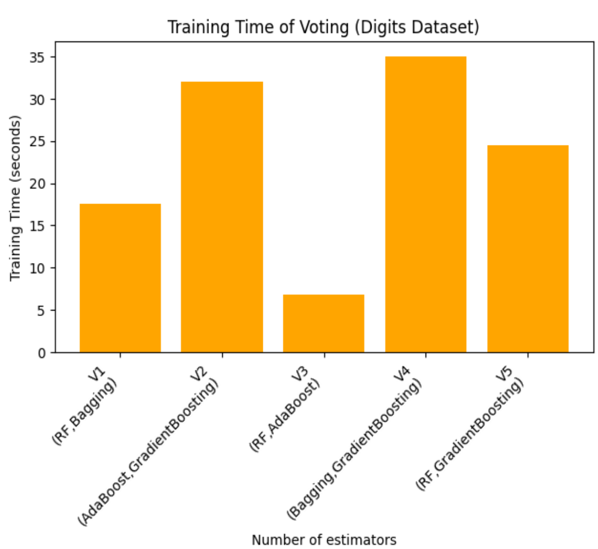
Plot for different metrics for different selection of estimators:

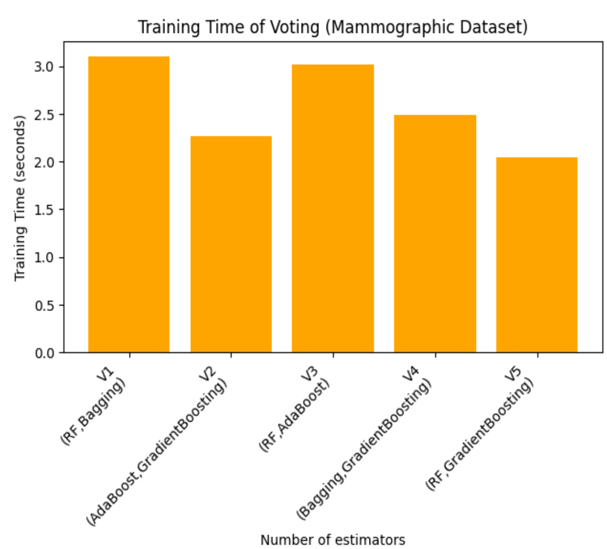












Analysis:

Voting is an ensemble learning method that combines multiple individual models to make predictions. By leveraging the collective wisdom of diverse models, voting can often achieve better performance than any single model alone. In this analysis, we examine the performance of voting ensembles with different combinations of base estimators on two datasets: Digits and Mammographic.

When using Random Forest and Bagging as base estimators (V1), the voting ensemble achieves moderate accuracy on the Digits dataset but slightly lower accuracy on the Mammographic dataset. The MSE values are relatively low, indicating good predictive performance overall. However, the training time for this ensemble is quite high, especially on the Digits dataset, which suggests that the ensemble may not scale well to larger datasets.

In the case of AdaBoost and Gradient Boosting (V2), the voting ensemble achieves higher accuracy on the Mammographic dataset compared to the Digits dataset. The MSE values are relatively high for the Digits dataset, indicating poorer performance in terms of error minimization. Additionally, the training time for this ensemble is substantially higher than V1, indicating that the inclusion of AdaBoost and Gradient Boosting models contributes to increased computational complexity.

When combining Random Forest and AdaBoost (V3), the voting ensemble achieves similar accuracy and MSE values compared to V1 on the Digits dataset, but slightly higher accuracy and lower MSE values on the Mammographic dataset. The training time is also lower compared to V2, suggesting that this combination strikes a balance between performance and computational efficiency.

Using Bagging and Gradient Boosting (V4) as base estimators, the voting ensemble achieves high accuracy and low MSE values on both datasets. However, the training time is considerably higher compared to other configurations, indicating that this ensemble may not be suitable for real-time applications or large-scale datasets.

Finally, when combining Random Forest and Gradient Boosting (V5), the voting ensemble achieves the highest accuracy on the Digits dataset among all configurations. However, the performance on the Mammographic dataset is similar to other configurations. The MSE values are relatively low, indicating good predictive performance overall, and the training time is moderate compared to other configurations.

Overall, the choice of base estimators in a voting ensemble can significantly impact its performance and computational complexity. Practitioners should carefully consider the trade-offs between accuracy, MSE, training time, and the specific requirements of their application when selecting base estimators for a voting ensemble.

6 Conclusion

In this comprehensive analysis, we explored various ensemble learning methods, including Random Forest, Bagging, AdaBoost, Gradient Boosting, and Voting, across two datasets: Digits and Mammographic. We systematically varied parameters such as the number of estimators to understand their impact on model performance, training time, and cross-validation scores.

Across all methods and datasets, we observed that the choice of base estimators and the number of estimators significantly influenced the performance of ensemble models. Random Forest consistently demonstrated strong predictive performance, achieving high accuracy and low mean squared error (MSE) values on both datasets. However, it tended to have longer training times compared to other methods, especially as the number of estimators increased.

Bagging and Gradient Boosting also showed promising results, with high accuracy and relatively low MSE values. Bagging performed well in reducing variance and improving generalization, while Gradient Boosting excelled in reducing bias and enhancing model interpretability. AdaBoost, on the other hand, showed mixed results, with relatively lower accuracy and higher MSE values compared to other methods, especially on the Digits dataset.

Voting ensembles, which combine multiple base estimators, exhibited competitive performance, leveraging the strengths of different models to achieve robust predictions. However, the choice of base estimators and their weights in the ensemble played a crucial role in determining overall performance. For instance, combinations of Random Forest with other models often yielded higher accuracy and lower MSE values compared to other combinations.

Furthermore, as we varied the number of estimators, we observed trade-offs between model complexity, predictive performance, and training time. Increasing the number of estimators generally led to improvements in accuracy and MSE reduction but also resulted in longer training times, especially for computationally intensive methods like Gradient Boosting.

Overall, ensemble learning methods offer powerful techniques for improving model performance and generalization across diverse datasets. By carefully selecting base estimators, tuning hyperparameters, and considering computational constraints, practitioners can leverage the collective intelligence of ensemble models to tackle a wide range of machine learning tasks effectively.

Reference:

[1] Fetch dataset from openml by name or dataset id. Scikit-learn user manual. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_openml.html

[2] Load and return the Digits Dataset (Classification), https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html

[3] Elter, Matthias. (2007). Mammographic Mass. UCI Machine Learning Repository. https://doi.org/10.24432/C53K6Z.