Classification of Handwritten Digits Using Ensemble Methods

Weiqian Hou & Mansi Modi

Math 285 : Classification techniques for MNIST Handwritten Digits Dataset

Introduction

In statistical terms, classification is a technique to identify the set of categories to which the new observation belongs to, on the basis of a sample training data set in which the category of each observation is known. Some of the most commonly used classification techniques are: Classification Tree and its ensemble methods: Random Forest, Bagging and Boosting.

Classification Tree

- Classification tree uses a decision tree as a predictive model which uses several decision criteria and maps the observation to the correct category, or as it is more generally said "class."

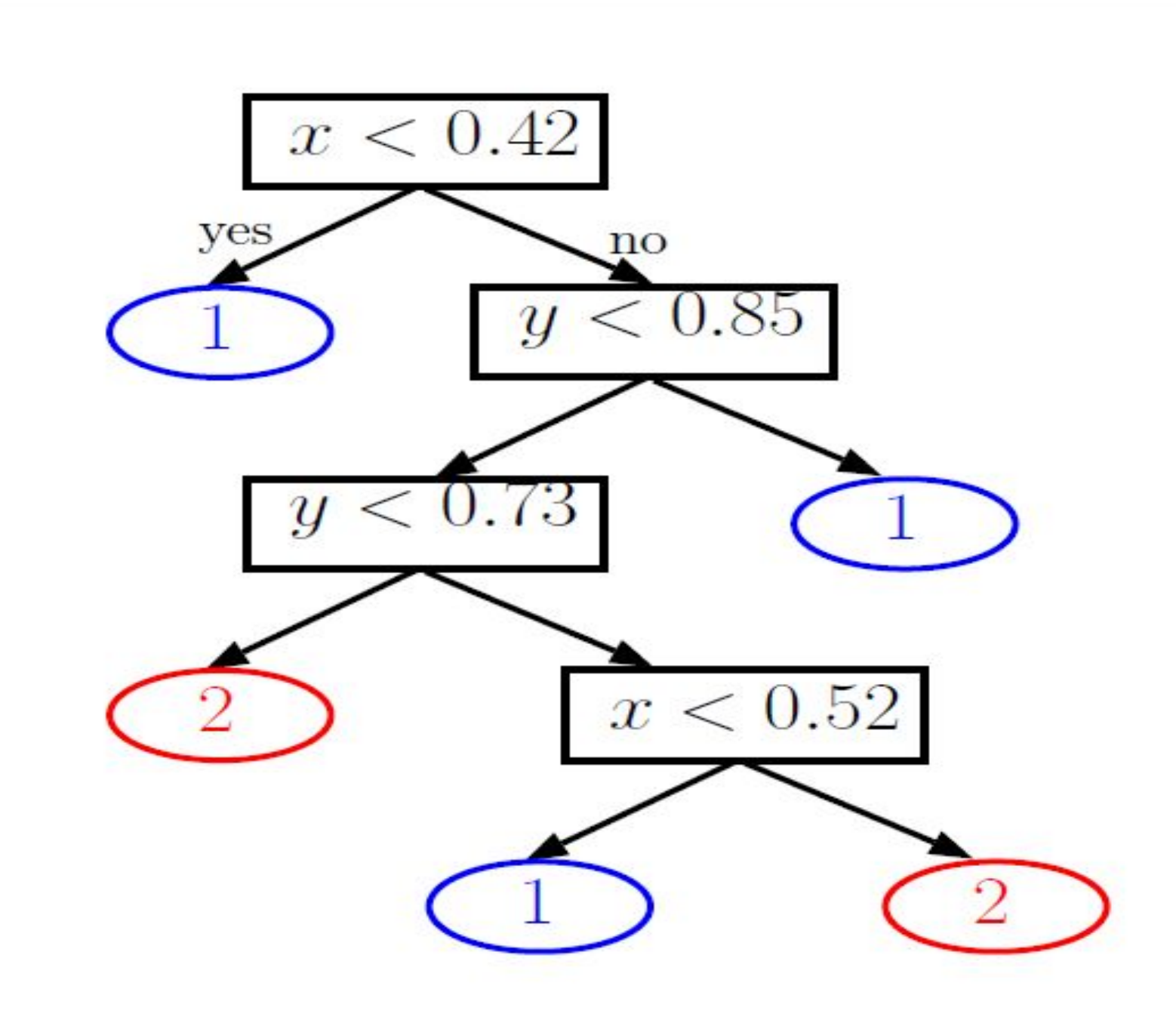

Figure 1: Classification tree for toy data set

- Figure 1 shows a classification tree for a toy data set with binary classes 1 and 2.
- Splitting probability at each node is different.
- Splitting stops when a terminal node is achieved.
- Each observation is classified in either of the 2 classes based on the tree.
- Overfitting of the data will give a long tree- not useful for test data.
- Several pruning rules are implied for large data sets and higher dimensions.

- Figure 2 shows the decision boundary for toy data set.
- Boundary is piece- wise linear
- Splitting criterion depends on Gini Index, Entropy and Misclassification Error.
- Boundary changes with the minor changes in data set.
- Decision tree is a weak learner, poor performance due to higher variance.

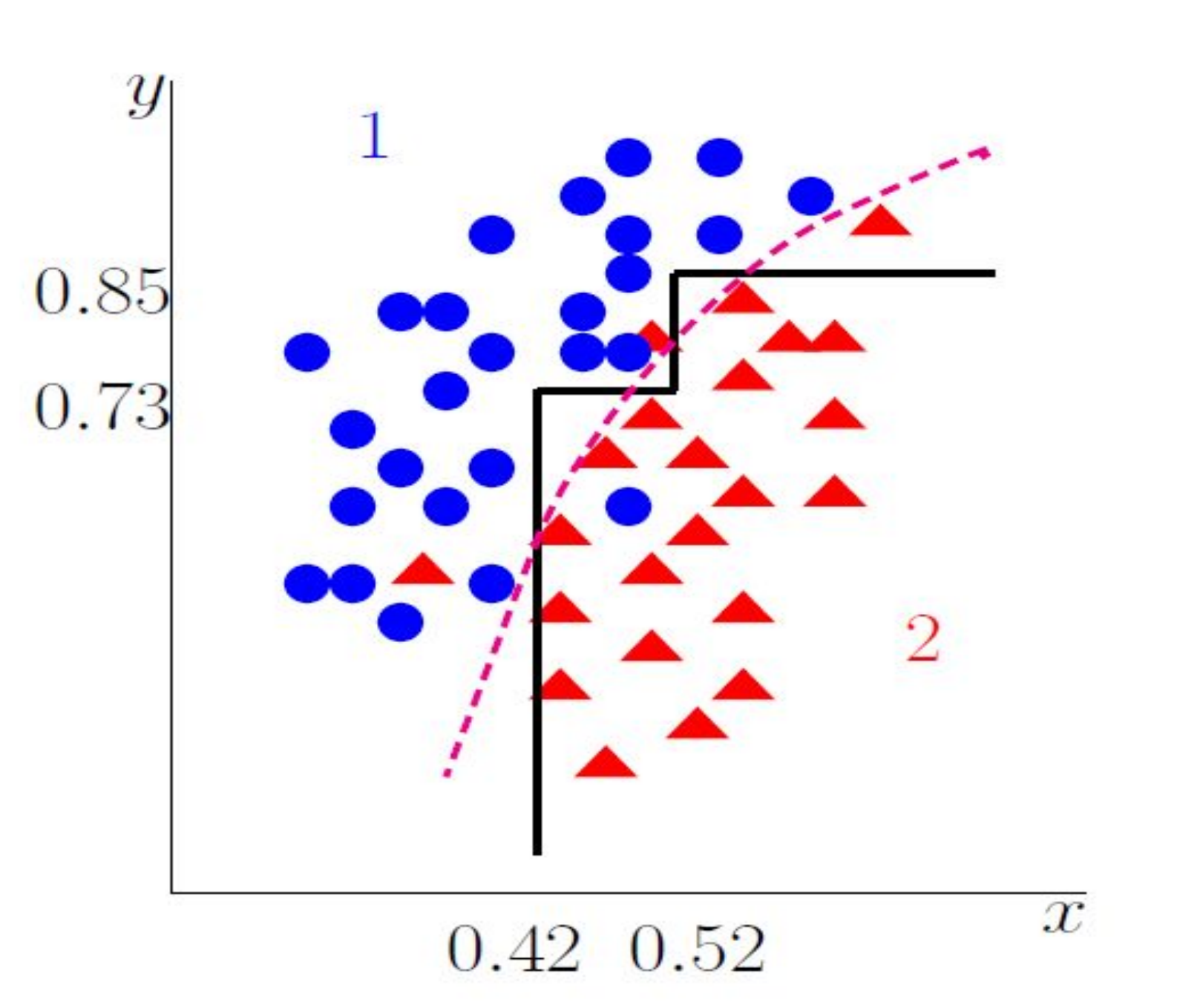

Figure 2: Decision boundary of a classification tree for toy data set

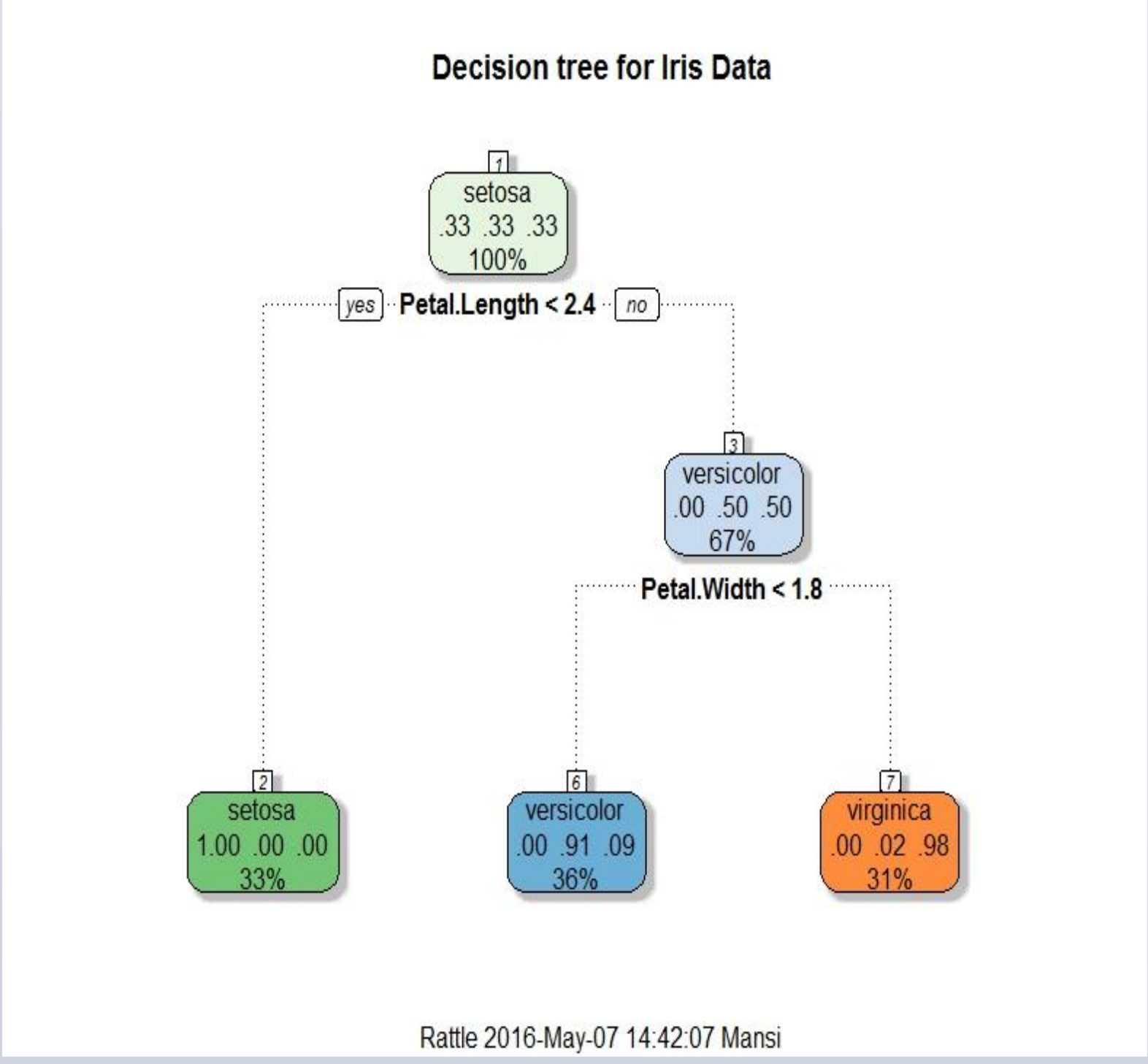

Figure 3: Example of a classification tree for Iris data set

- Figure 3 shows the decision tree for Fisher Iris Data set.
- Dataset: 150 instances, 4 attributes and 3 classes.
- First split based on petal length and terminal nodes give the classes with respective probabilities.
- **The test error for MNIST data using a single decision tree is about 12.23% with 240 intermediate branches.**
- The run time for a single decision tree is about 1 min but the results are not visually attractive.

Bagging and Random Forest

- Many independent trees from different bootstrap samples of the training data, (Random samples with replacement) are build and a vote of the predictions is considered to get a final prediction.
- Random forest is a variation of bagging as it allows to use different subsets of the variables at the nodes of any tree in the ensemble.

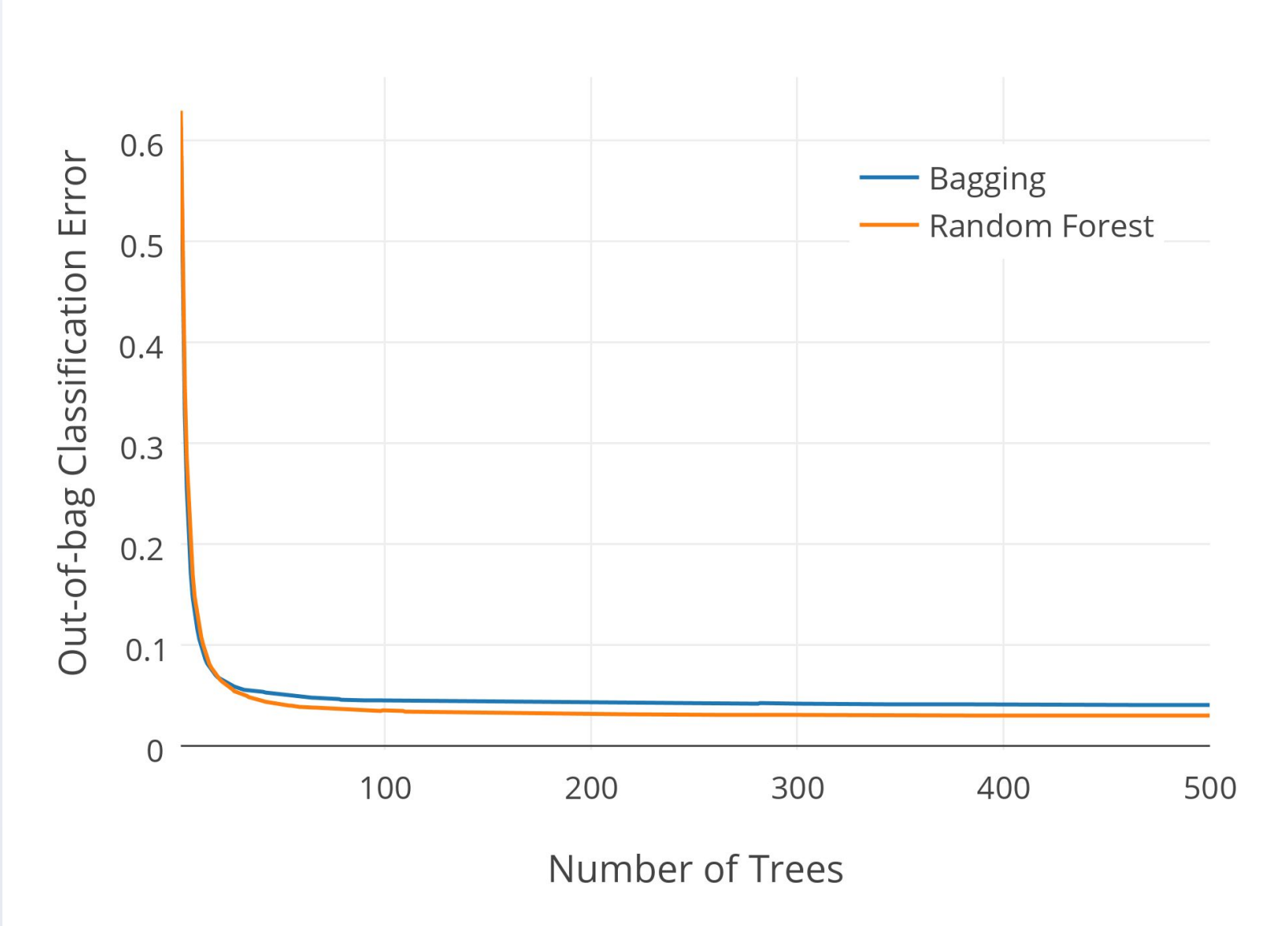

Figure 4: OobError curve for bagging vs random forest for 500 trees.

- The error rates of the two methods converge around 50 trees.
- Random forest has better performance than bagging in terms of speed and accuracy.
- Bagging consumes 353 minutes while it takes random forest 16 minutes.

- The error rate is computed at 25, 50 and 500 trees with the two methods.
- Random forest always performs better than bagging.
- The difference in their error rates does not change a lot.

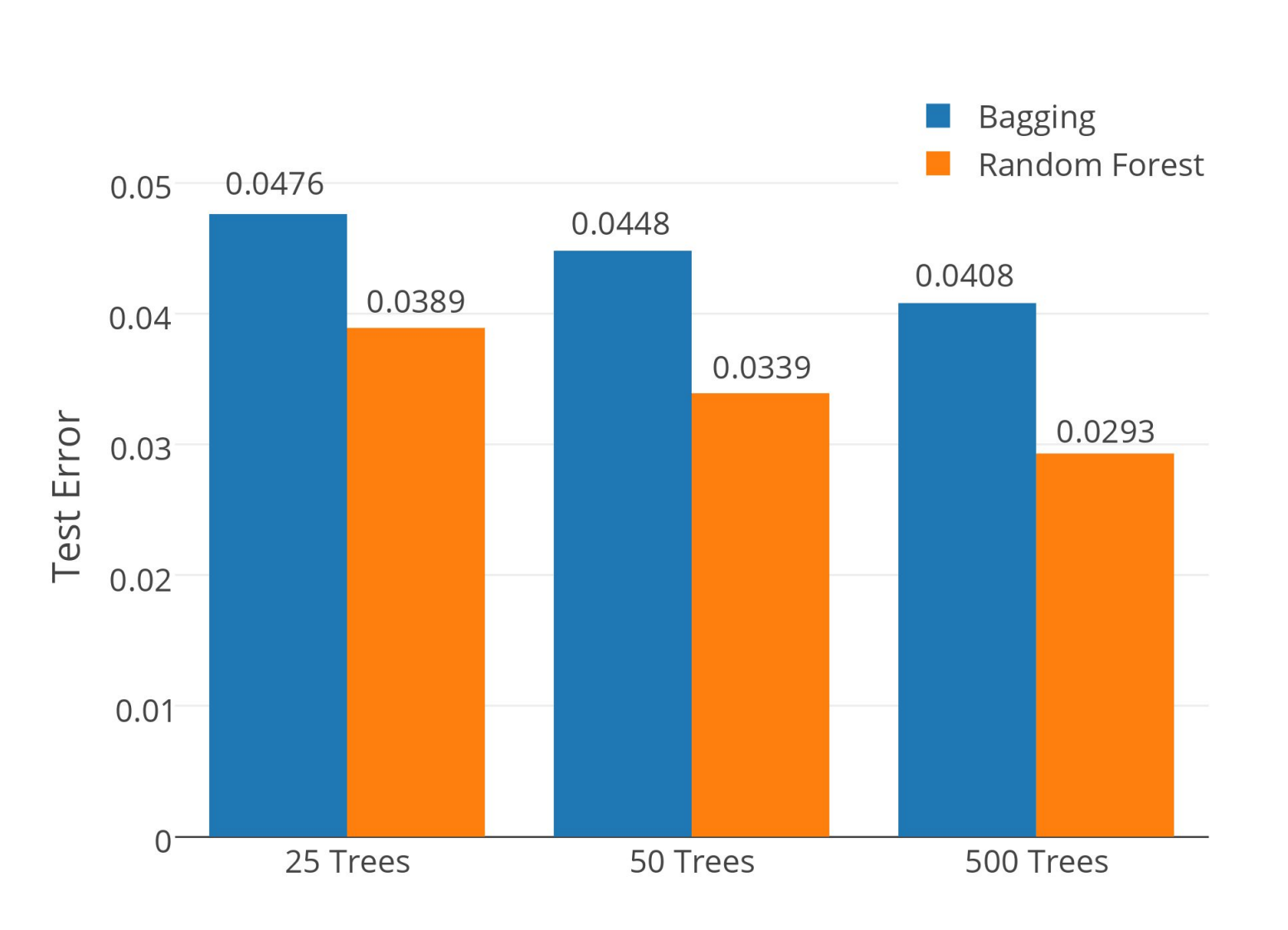

Figure 5: The test errors against number of trees.

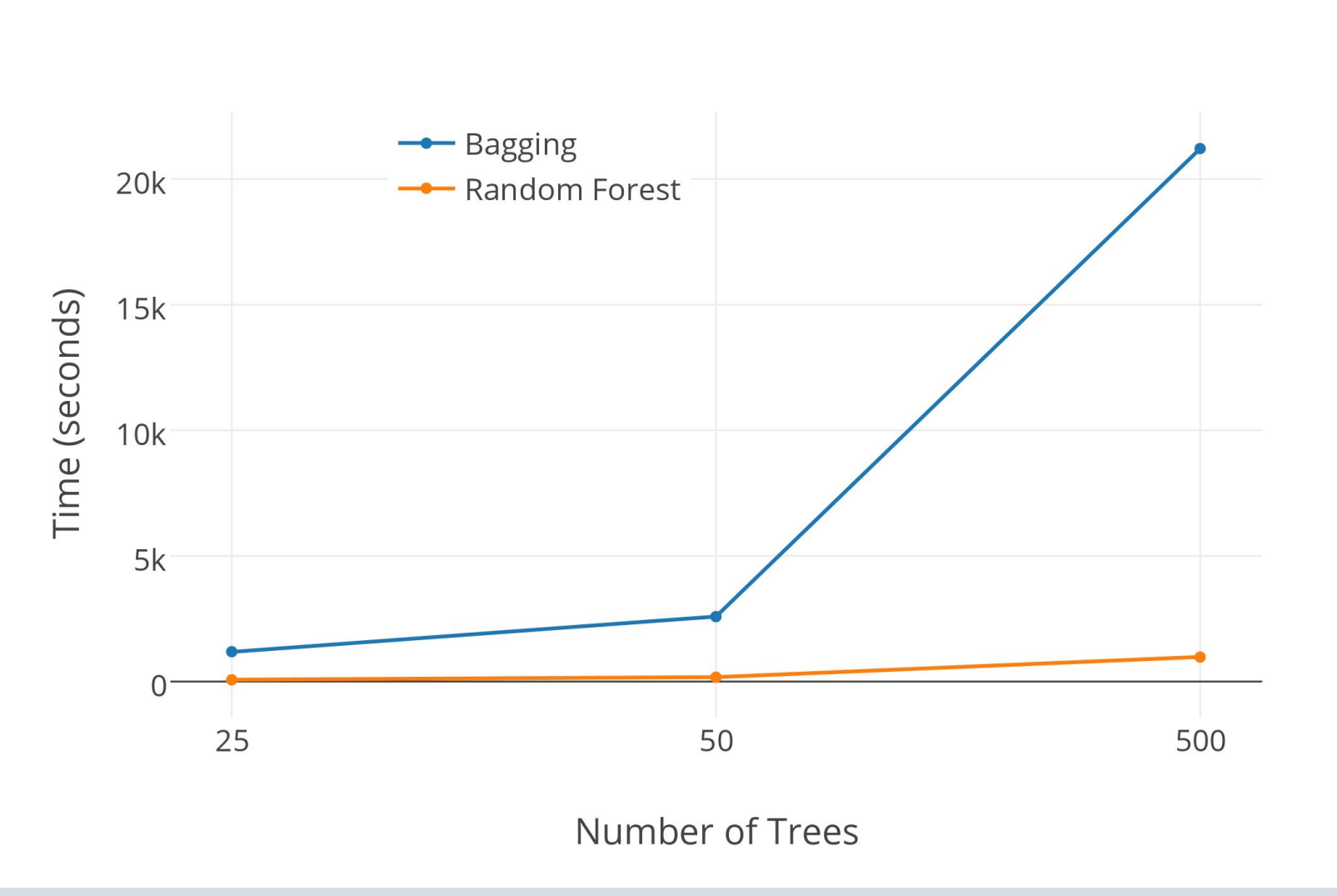

Figure 6: Speed comparison for bagging and random forest.

- The time of running n = 25, 50 and 500 trees with bagging and random forest methods is measured.
- As the number of trees increases, bagging takes much longer than random forest.
- **Conclusion: Random forest is more accurate and faster as compared to bagging.**

Boosting

- Boosting is building many trees adaptively and then adding their predictions. The four boosting techniques implemented here are: LogitBoost, AdaBoost, GentleBoost and Gradient Boost.
- LogitBoost can be seen as a convex optimization problem. The LogitBoost algorithm minimizes the logistic loss:

$$\sum_i \log(1 + e^{-y_i f(x_i)})$$

- AdaBoost builds tree classifiers sequentially by "focusing more attention on training errors made by the preceding trees and then add their predictions.
- GentleBoost puts less weight on outlier data points.
- Gradient Boosting use gradient descent algorithm to optimize a loss function.
- The objective of this part is to compare the performance of these four boosting methods at different number of trees.

LogitBoost, AdaBoost, Gentle Boost and Gradient Boost

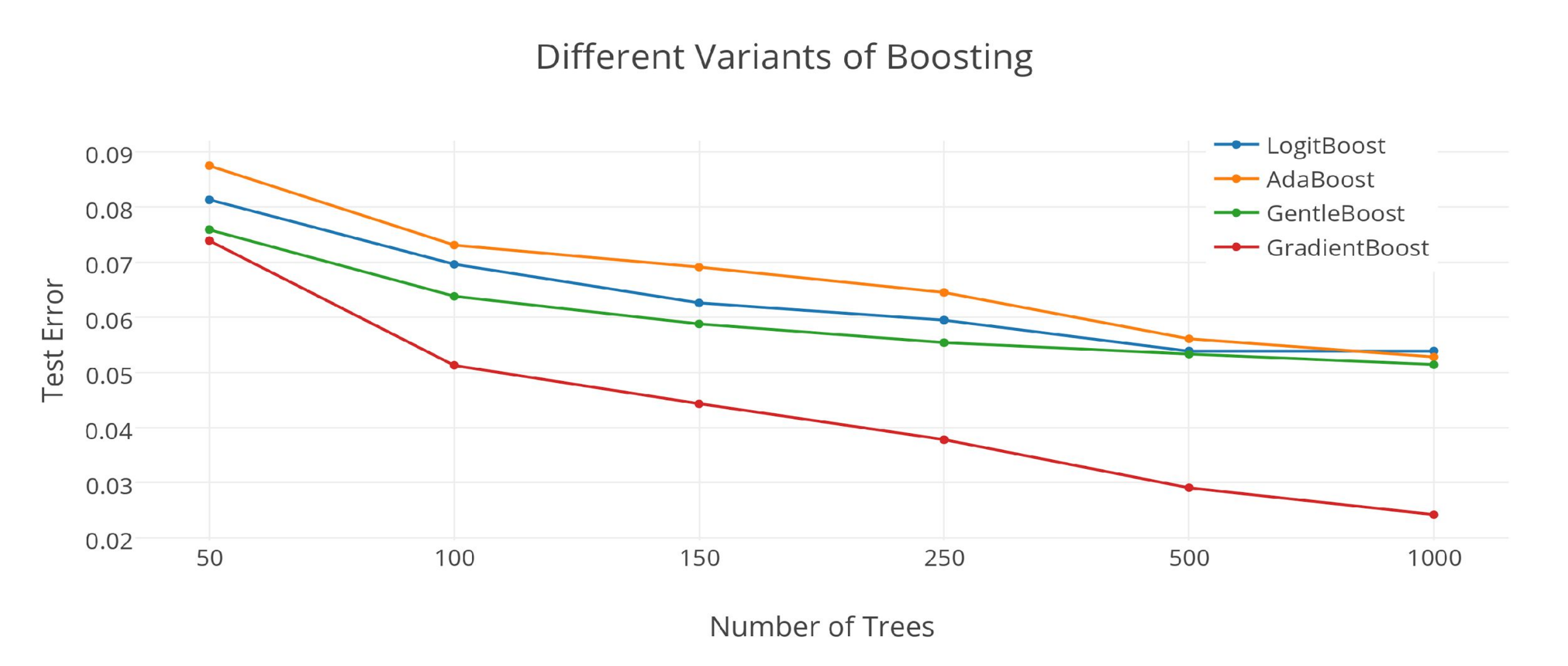

Figure 7: Test errors of LogitBoost, AdaBoost, GentleBoost and GradientBoost against number of trees.

- Apply one-versus-one extension to two-class methods LogitBoost, AdaBoost and GentleBoost. Their error rate seems get closer and closer as the number of trees increases.
- Gradient Boosting does the best job for the MNIST handwritten digits data.

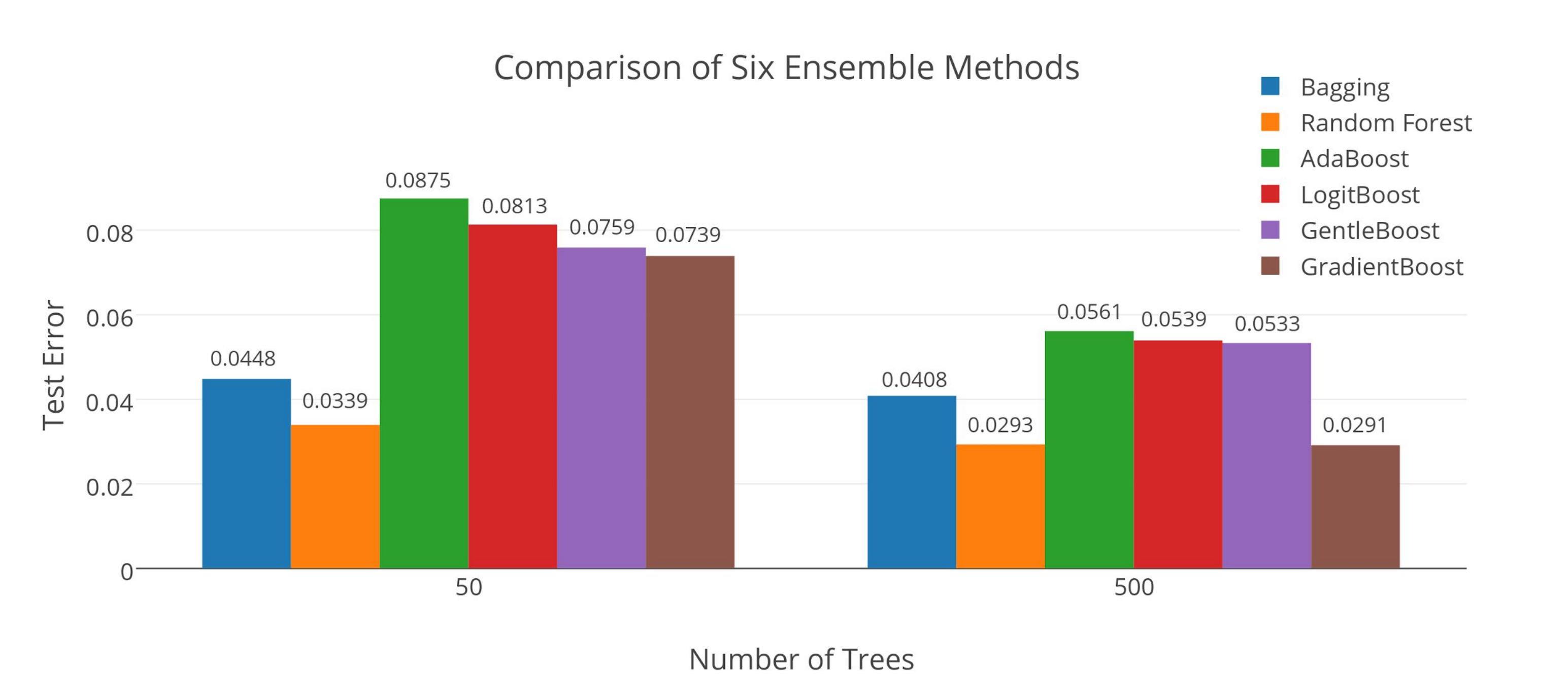

Figure 8: Comparison of test error for all the six ensemble methods working on the MNIST handwritten digits against both small and large number of trees..

Conclusion:

- The random forest is the best method when the number of trees is small, while gradient boosting does the best job among all the models when the number of trees is increased to 500.
- The bagging team (bagging and random forest) converge early, while the boosting team (LogitBoost, AdaBoost, GentleBoost and GradientBoost) a long time to obtain the good results.
- GradientBoost >random forest >bagging > GentleBoost >LogitBoost >AdaBoost >single tree.

References

[1] LEC 8: Classification trees and ensemble learning by Dr. Guangliang Chen, SJSU http://www.math.sjsu.edu/~gchen/Math285S16/lec8ensemble.pdf

[2] Chapter 4: Classification: Basic Concepts, Decision trees, and Model Evaluation, Introduction to data mining by Pang-Ning Tan, Michael Steinbach, Vipin Kumar. http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

[3] LogitBoost, Wikipedia, the Free Encyclopedia, December 4, 2014. https://en.wikipedia.org/w/index.php?title=LogitBoost&oldid=636575418.

[4] Gradient Boosting, Wikipedia, the Free Encyclopedia, May 8, 2016. https://en.wikipedia.org/w/index.php?title=Gradient_boosting&oldid=719234230.