Final Project: Predicting Boston Housing Prices

Andre A. S. T. Ribeiro*

E-mail: aar2163@gmail.com

1 Model Analysis

1.1 Model Complexity Graphs

The squared error between the output of a given model and the expected responses of the input vector of the training set on which the model was based can be decomposed into "bias" and "variance" terms: [1]

$$E_D[f(\mathbf{x},D) - E[y|\mathbf{x}]] = (E_D[f(\mathbf{x},D)] - E[y|\mathbf{x}])^2 + E_D[(f(\mathbf{x},D) - E_D[f(\mathbf{x},D)])^2] \quad (1)$$

As a consequence of Equation 1 there is a tradeoff between minimizing the bias and variance of a given model, in the sense that having a model that completely fits the training data will also fit the corresponding noise and lead to a loss of generalization power, *i.e.*, overfitting.

In the following subsections I have calculated the model complexity graphs for four different machine learning techniques, namely, Decision Tree, K-Nearest Neighbors, Neural Networks and Boosting. The Boston Housing Prices are divided in a training set and a test set. I have also made different divisions of the overall data into training/test sets, allowing a more thorough assesment of the optimal model complexity in each case.

*To whom correspondence should be addressed

1.1.1 Decision Tree

The results obtained for the decision tree algorithm indicate that the bias is reduced to almost zero for maximum depths greater than 10, however, the variance term and, consequently, the generalization performance is not reduced further, staying approximatelly constant after a maximum depth of 4 or 5 is reached, depending on the size of the training set. The training size significantly influences the test error, as will be discussed in Section 1.2. The optimal model complexity seems to be around a maximum depth of 15.

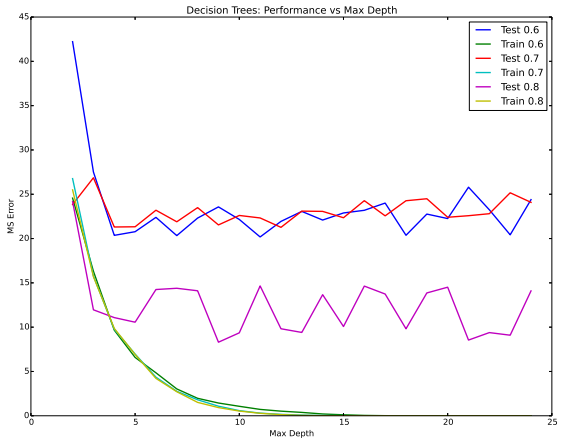

Figure 1: Errors on the Training and Test Sets as a function of Maximum Depth for the Decision Tree algorithm. Different sizes of the training set were used in the calculation, with the values for 60%, 70% and 80% of the total data included in the training set shown in blue/green, red/cyan and purple/yellow, respectively.

1.1.2 k-Nearest Neighbors

The results of the k-Nearest Neighbors method clearly indicate that both types of error increase with increasing model complexity, with the error on the test set reaching a minimum at $k = 4$, which is thus the optimal complexity for this method. It is evident that increased model complexity does not lead to overfitting, as the bias increases with k.

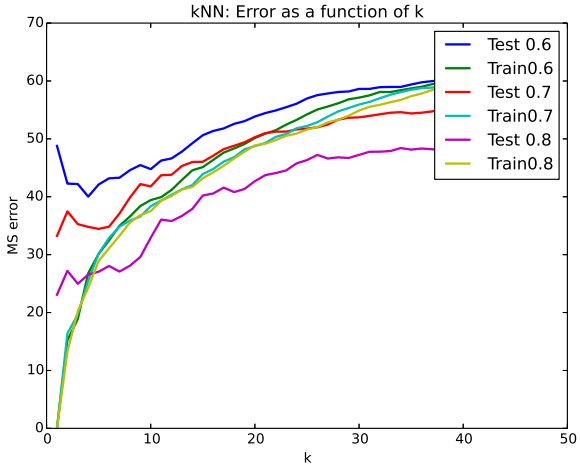

Figure 2: Errors on the Training and Test Sets as a function of k for the k-Nearest Neighbors algorithm. Legends are the same as in Figure 1.

1.1.3 Neural Networks

The results obtained for different number of hidden layers of Neural Networks show that this class of estimators has a relatively low bias. The comparison between different training sizes also indicates low variance, as will be more clearly shown in Section 1.2. It is evident that overfitting does not occur for the model complexities considered here and that the optimal complexity is most likely 3, 4 or 5 hidden layers.

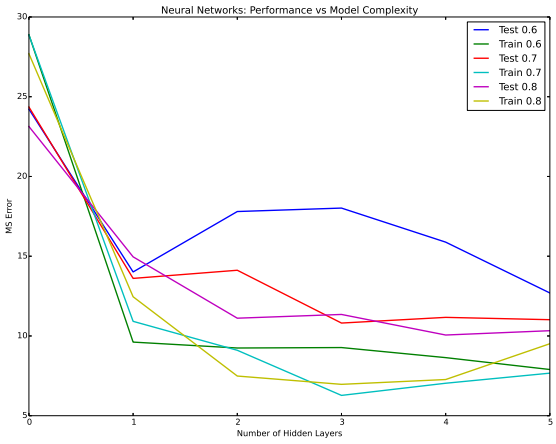

Figure 3: Errors on the Training and Test Sets as a function of the number of hidden layers for the Neural Networks algorithm. Legends are the same as in Figure 1.

1.1.4 Boosting

The complexity of the Boosting method depends on both the number and complexity of the base learners. The following Figure addresses these issues. The left panel shows the results for the Boosting algorithm with Decision Trees of maximum depth equal to 5 as base learners, while the right panel shows the corresponding results for maximum depth equal to 15. The results indicate that both types of error reach an approximatelly constant value when the number of base learners is higher than 50. It is evident that increased model complexities do not lead to overfitting. Furthermore, usage of more complex base learners reduces the bias to essentially zero, a fact that was also noted for the Decision Trees of higher complexity.

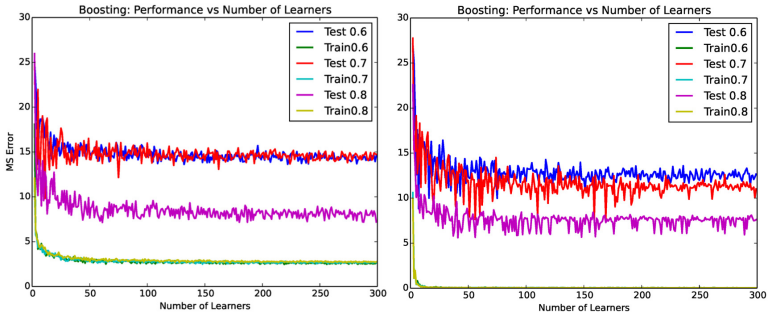

Figure 4: Errors on the Training and Test Sets as a function of the number of learners for the Boosting algorithm. Legends are the same as in Figure 1. Decision Trees with maximum depth of 5 (Left Panel) and 15 (Right Panel) were used.

1.2 Learning Curves

The construction of model complexity graphs showed that the results are somewhat dependent on the size of the training set and the random state of the data shuffling algorithm. In order to identify the best learning algorithm, a more quantitative approach is necessary. In the following subsections, I have constructed the learning curves for the different algorithms using multiple runs. The reported errors are the average values for the different runs. Each run used a different random state of the data shuffling algorithm. The number of runs was 30 for the Neural Networks and Boosting algorithms and 90 for the Decision Tree and kNN algorithms. The maximum size of the

training set was 70% of the total data.

1.2.1 Decision Tree

The following Figure shows the learning curve for the Decision Tree with maximum depth equal to 15. The mean squared test error for the maximum training size was 21.5 ± 0.7 (thousands of dollars)^2. It is evident that the present model is a high-variance estimator and that increased training sizes could lead to better generalization performance.

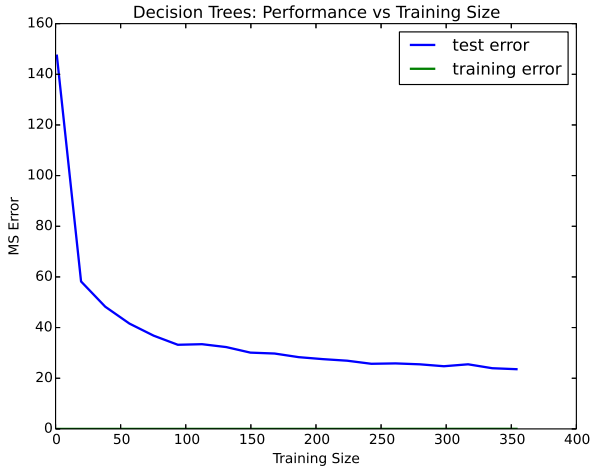

Figure 5: Errors on the Training and Test Sets as a function of Training Set size for the Decision Tree algorithm with maximum depth equal to 15.

1.2.2 k-Nearest Neighbors

The following Figure shows the learning curve for the kNN algorithm with k equal to 4. The mean squared test error for the maximum training size was 44.4 ± 0.9 (thousands of dollars)^2. As previously discussed, the bias for this class of algorithm applied to the present dataset is high, leading to poor generalization. Increasing the training size will not lead to significant generalization improvements.

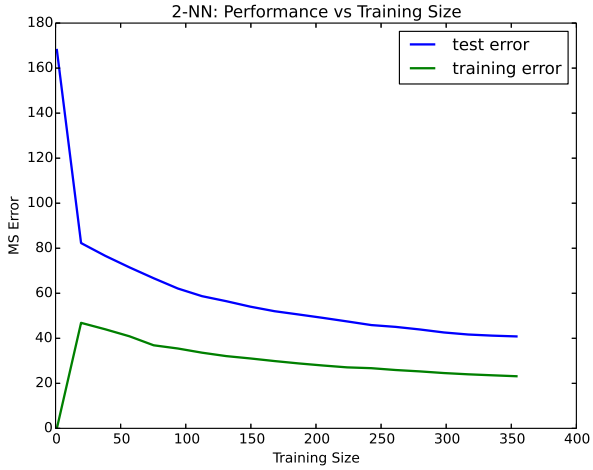

Figure 6: Errors on the Training and Test Sets as a function of Training Set size for the k-Nearest Neighbors algorithm with $k = 4$.

1.2.3 Neural Networks

The following Figure shows the learning curve for a Neural Network with 3 hidden layers. The mean squared test error for the maximum training size was 13.9 ± 0.7 (thousands of dollars)^2. The corresponding analysis for 4 and 5 hidden layers yielded the error values of 15.8 ± 0.9 and 15.2 ± 0.8 (thousands of dollars)^2, respectively. The figure shows that this estimator has a relatively low bias and that the generalization does not improve significantly for training sizes greater than 100, indicating a relatively low variance as well.

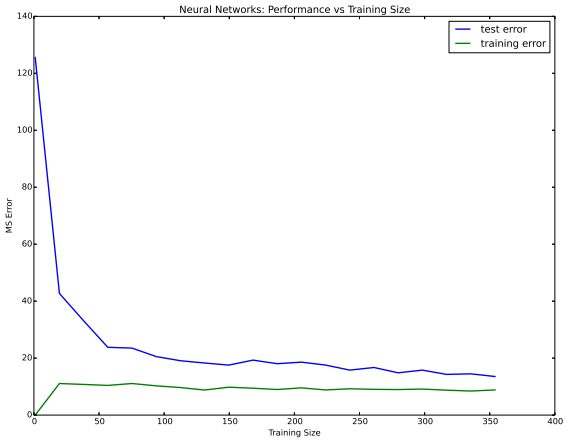

Figure 7: Errors on the Training and Test Sets as a function of Training Set size for the Neural Networks algorithm with 3 hidden layers.

1.2.4 Boosting

The following Figure shows the learning curve for the Boosting Algorithm with 50 base learners, with each learner being a Decision Tree of maximum depth equal to 15. The mean squared test error for the maximum training size was 12.7 ± 0.7 (thousands of dollars)^2. The corresponding analysis for 100 base learners yielded the error value of 11.4 ± 0.6 (thousands of dollars)^2. Comparison of this result to the Decision Tree case reveals a lower variance and equivalent (almost zero) bias. Increasing the training size could lead to increased generalization performance.

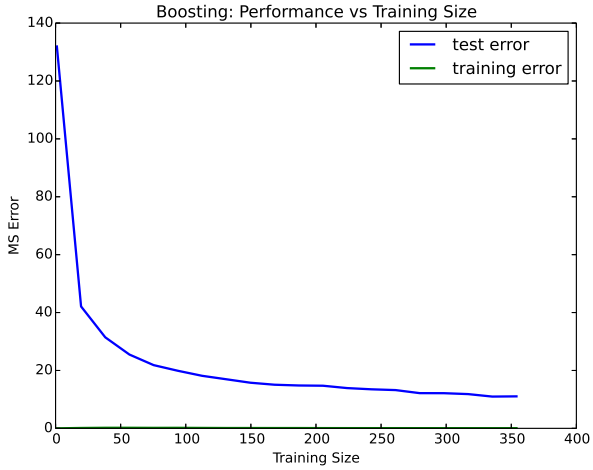

Figure 8: Errors on the Training and Test Sets as a function of Training Set size for the Boosting algorithm with 50 base learners.

2 Prediction of Price for the Selected Feature Set

The results shown in Section 1.2 indicate that the methods that exhibit the best generalization performance of the Boston Housing Prices are the Neural Networks algorithm with 3 hidden layers and the Boosting algorithm with 100 base learners (Decision Trees of maximum depth equal to 15). To predict the price of the selected feature set and the associated error for both models a 10-fold cross validation was performed. The following Table lists the mean of the predicted prices, the mean of the root mean squared errors on test and training sets, the 95% confidence interval and the minimum and maximum values of the 10-fold cross validation fitted models for both methods.

Table 1: Mean, Mean of the Root Mean Squared Error on Test and Training Sets, 95% Confidence Interval and Minimum and Maximum values of 10-fold cross validation using the algorithms Boosting (100 base learners) and Neural Network (3 hidden layers).

Method	Mean	Mean RMSE (Test)	Mean RMSE (Train)	95% CI	Min.	Max.
Boosting	25.87	2.93	0.19	[20.11,31.62]	20.80	27.90
Neural Network	25.73	3.45	2.77	[18.95,32.51]	22.69	28.13

The confidence interval is calculated based on the assumption of normally distributed independent random errors. The following Figure shows the histogram of the test residuals of the 10-fold

cross validation of the Boosting method. A Gaussian distribution was fitted to the values, yielding the parameters shown in the Figure. The Shapiro-Wilk test indicates that the residual distribution is actually not Gaussian ($p \approx 10^{-20}$), and the Figure clearly shows that the Confidence Interval based on a Gaussian distribution is a conservative estimate.

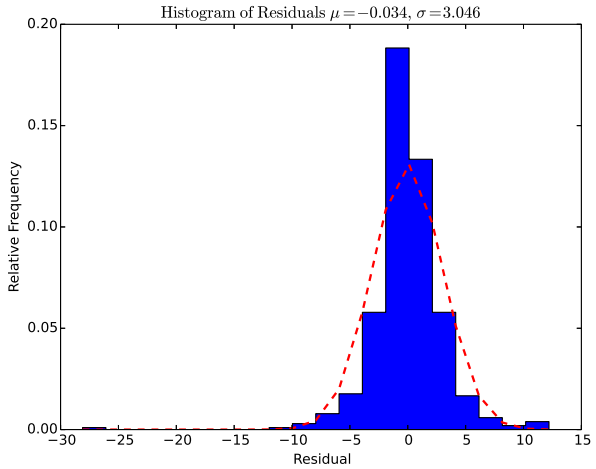

Figure 9: Histogram of residuals between observed and predicted test set values for a 10-fold cross validation of the Boosting algorithm with 100 base learners.

It can be seen that the mean values predicted by the two methods are similar. The range of predicted values is broader for the Boosting method, as can be seen by comparing the minimum and maximum values. The mean RMSE on the test sets is somewhat lower for the Boosting method. Furthermore, as discussed previously, the bias is essentially zero for the Boosting method, indicating that a larger dataset could improve its performance. This is not observed for the Neural Network. The results thus indicate that Boosting based on 100 Decision Trees of maximum depth equal to 15 is the preferred method for predicting Boston Housing Prices.

3 Explanation to a Layman

Computational methods can be used to analyze/model data and make predictions about unknown outcomes. Analysis of the Boston Housing Dataset allowed the estimation of the price of the selected house. A value of 25.8 ± 5.7 thousands of dollars was obtained, indicating that there is

a 95% probability that the price of the selected house does not lie outside of the interval between 20.1 and 31.6 thousands of dollars.

References

(1) Geman, S.; Bienenstock, E.; Doursat, R. *Neural computation* **1992**, *4*, 1–58.