Additional exercise

Maria Beatrice Cattini

Matteo D'Alessandro

Lisa Incollingo

Vanessa Ventura

December 13, 2022

1 Introduction

The bias–variance trade-off is a central problem in supervised learning. It concerns the relationship between the bias of a statistical model, which is the difference between the average prediction of our model and the correct value which we are trying to predict, and its variance, which represents the variability of the model's prediction.

In the context of this exercise we are going to consider a regression problem and the Mean Squared Error (MSE) as a metric of the quality of our predictions. If the real model is $y = f(x) + \varepsilon$ and our prediction is made as $\tilde{y} = \tilde{f}(x)$, the MSE can be decomposed in the following way:

$$MSE = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2 = \mathbb{E}[(y - \tilde{y})^2]$$

$$\mathbb{E}[(y - \tilde{y})^2] = \underbrace{\frac{1}{n}\sum(y_i - \mathbb{E}[\tilde{y}])^2}_{(\mathrm{Bias}[\tilde{y}])^2} + \mathrm{var}(\tilde{f}) + \sigma^2$$

Where the first term is the squared bias, the second is the variance of the model, and the last term σ^2 is the irreducible error, which is linked to the nature of the data and is not influenced by the chosen model.

The trade-off between bias and variance is regulated by the model's complexity: a more complex and flexible model is able to better adapt to the training data, reducing the bias, but will change more drastically with the introduction of new datapoints, increasing the variance of the prediction.

On the other hand, reducing the complexity of the model will reduce its variance but increase the bias. Figure 1 represents this relationship graphically. The overall

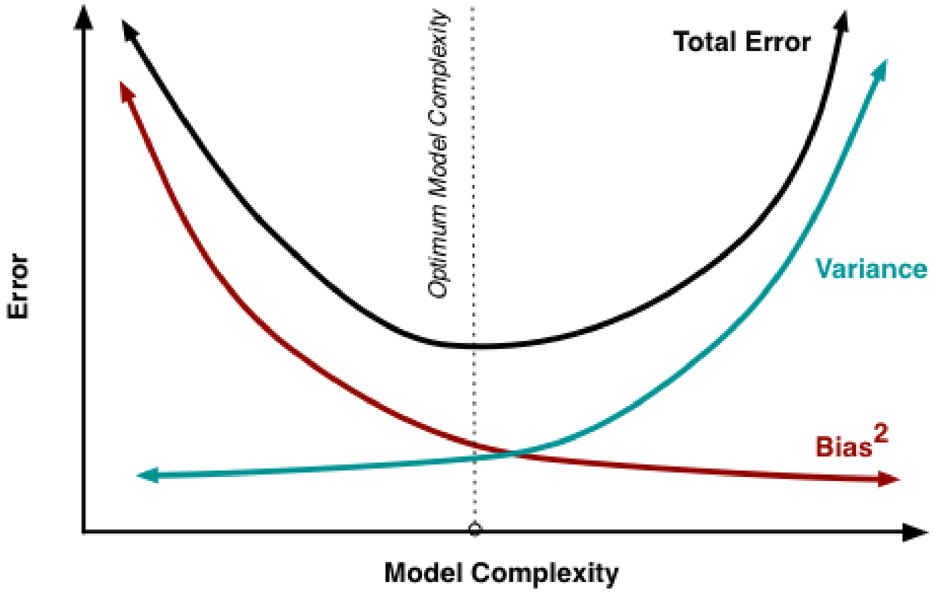

Figure 1: Visual representation of bias-variance decomposition with changing model complexity.

error, being given by the sum of bias and variance, usually assumes a typical "U-shape": finding the minimum of this curve by tuning the model complexity is a key step in obtaining the best results from the statistical methods we consider.

The aim of this exercise is to perform a bias-variance trade-off analysis for multiple classes of regression methods (linear regression, tree-based methods and neural networks) and understand similarities and differences between these methods' performances as the model complexity changes.
The data considered for this analysis consist of 100 points, generated from the function:

$$y = e^{-x^2} + 1.5e^{-(x-2)^2} + \varepsilon$$

With x coordinates distributed uniformally in the $[-3, 3]$ interval. ε is an error term distributed normally with mean 0 and variance 0.1.
Determining the bias and variance decomposition was done with boostrapping with 100 boostrap iterations. All the code related to this exercise can be found on the GitHub[1].

[1] https://github.com/matteodales/FYS-STK4155_Applied_Data_Analysis_and_Machine_Learning/tree/main/project3

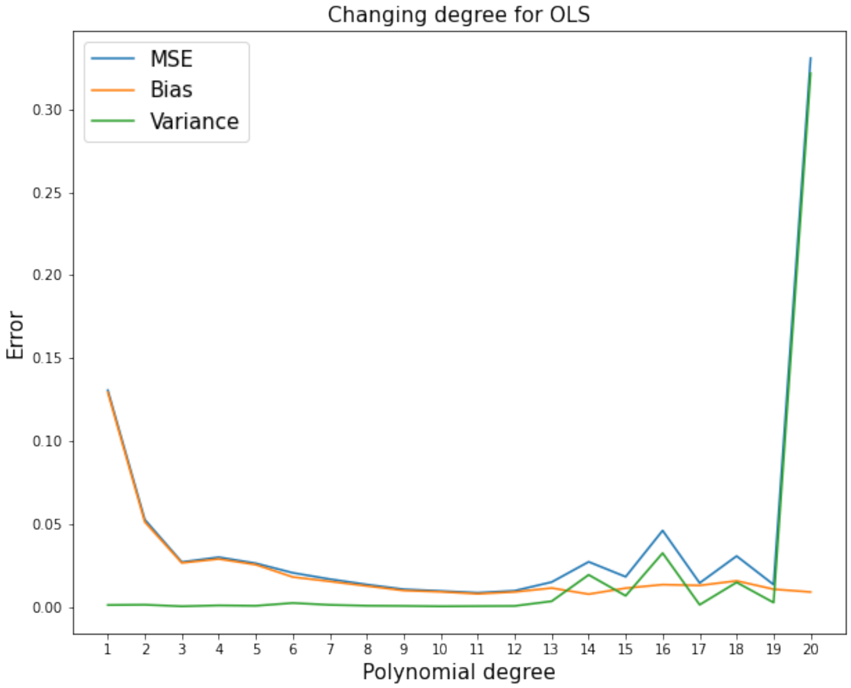

Figure 2: Bias-variance decomposition of the MSE for OLS for increasing polynomial degree.

2 Linear regression

2.1 OLS

The first regression method we consider is Ordinary Least Squares. In this case we are trying to fit the function f with a polynomial of different degree. The degree represents the model complexity in this case: a more complex polynomial can better fit the data, resulting in lower bias and higher variance.
Figure 2 shows that the results are similar to what we expected: the model starts off at degree 1 with very high bias, which is lowered as the degree increases; on the contrary, the variance is very low for small degrees, but shoots up to 0.3 when we reach degree 20.
In this case, the optimal choice for the degree was probably around 10.

2.2 Ridge and Lasso

In the case of shrinkage methods like Ridge and Lasso regression, the model complexity is represented by the penalty parameter λ: for high values of lambda the shrinkage is strong and the degrees of freedom of the model reduce, reducing the variance of the model but increasing its bias.

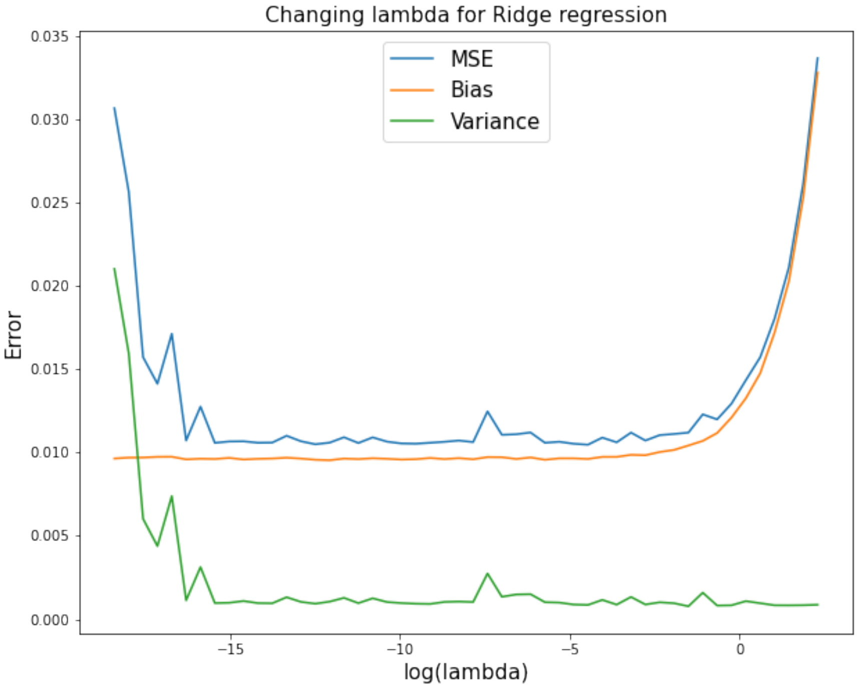

Figure 3: Bias-variance decomposition of the MSE for Ridge for changing λ. The polynomial degree is fixed to 12.

This is confirmed by Figures 3 and 4: the variance of the model is quickly shrinked for growing values of λ, but if the shrinkage is too strong, the bias of the model becomes prevalent. In the case of Lasso, we would expect the bias to reach a plateau over a certain value of λ, as all coefficients have been shrinked to 0 and the model is null.

3 Tree-based methods

3.1 Regression tree

When dealing with a decision tree, the complexity of the model is given by the depth of the tree. This kind of models usually deal with their high variance by performing a cost-complexity pruning which allows for a tuning of the bias-variance trade-off.
As we can see in Figure 5, the bias is strongly reduced when the depth of the tree increases. Somewhat surprisingly, except for a slight increase at the very left of the graph, the variance seems to not be strongly affected by the increasing depth.

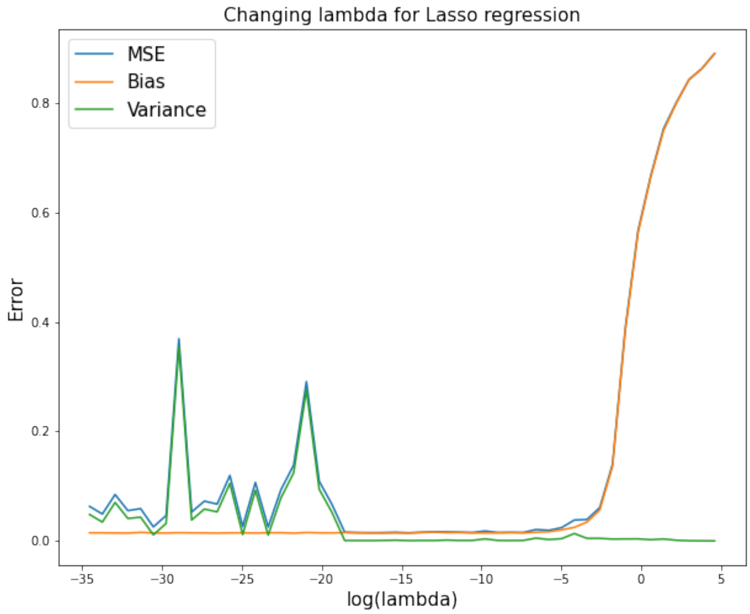

Figure 4: Bias-variance decomposition of the MSE for Lasso for changing λ. The polynomial degree is fixed to 12.

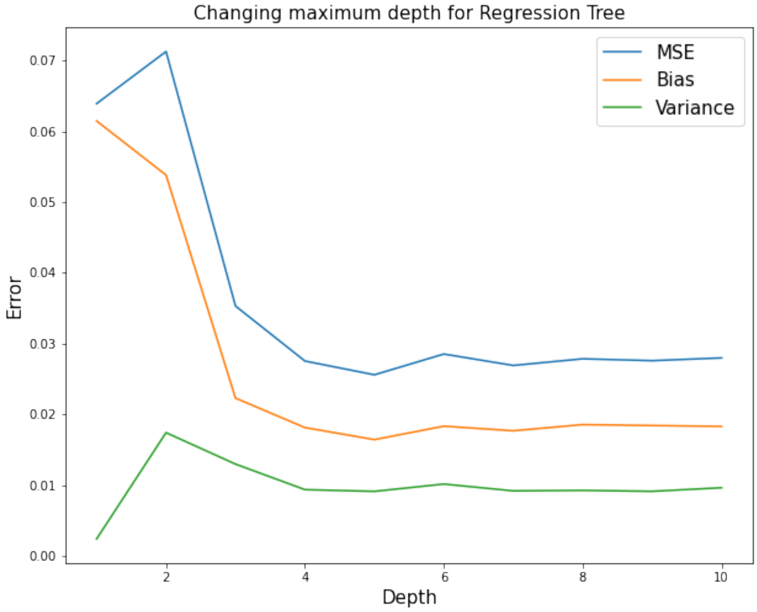

Figure 5: Bias-variance decomposition of the MSE for a regression tree with increasing maximum depth.

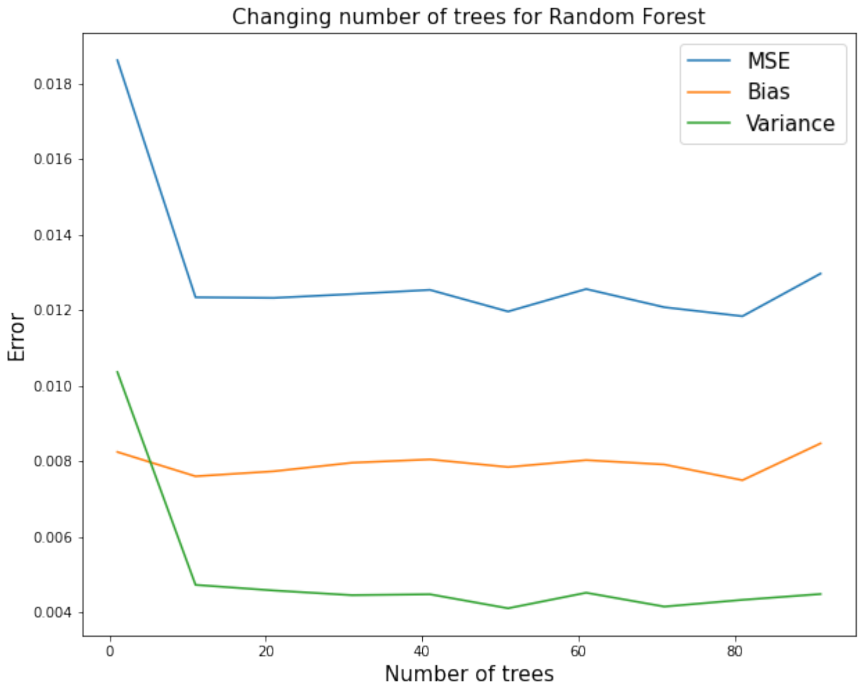

Figure 6: Bias-variance decomposition for a Random Forest with changing number of trees in the ensemble.

3.2 Random Forest

The definition of model complexity when dealing with ensemble methods is not as simple as in the previous cases. For example, in the case of Random Forest, a first variable seemingly in control of the complexity of the model is the number of trees in the ensemble: however, as proved by Breiman when firstly introducing the method [2], increasing the number of trees does not negatively affect the model's variance. This is verified in Figure 6 where, as the parameter changes, variance decreases and bias is not affected.

Another crucial parameter in the Random Forest algorithm is the number of features considered at each split for the ensemble trees, m. This parameter regulates how well the correlation between boostrap trees is handled and is usually either chosen as a function of the total number of variables ($m = \lfloor p/3 \rfloor$ in the case of regression) or tuned via cross-validation.
Since the dataset we have used so far in this analysis only contains a single variable, to study the bias-variance trade-off for changing m we consider a polynomial in x of degree 12. The number of predictors considered varies from 1 to all 13: Figure 7 shows how, surprinsigly, even the change in m does not really affect the

[2] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). `https://doi.org/10.1023/A:1010933404324`

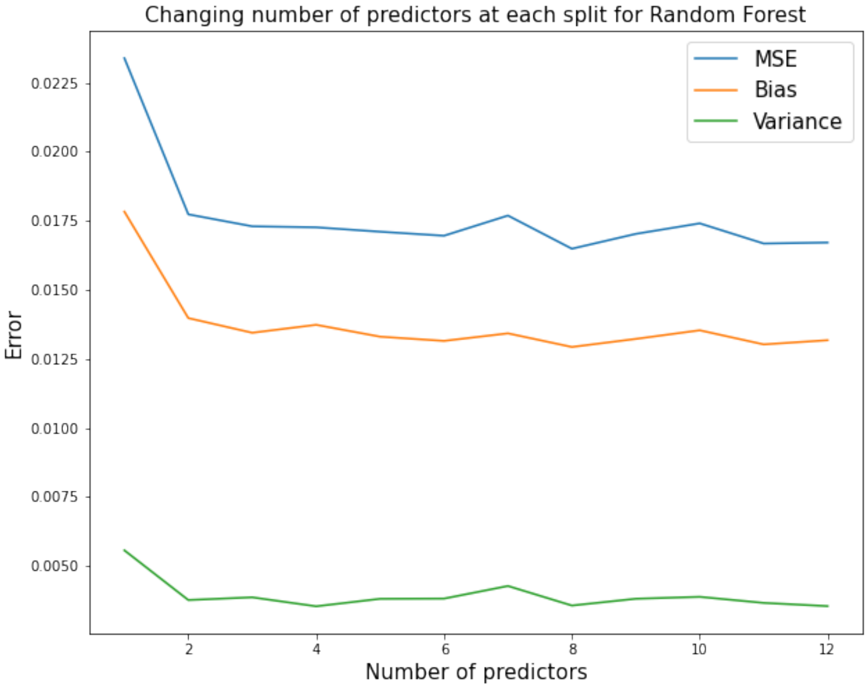

Figure 7: Bias-variance decomposition for a Random Forest with changing number of variables considered at each split.

variance of the model, proving that Random Forest handles the change in model complexity very well when considering our data.

3.3 AdaBoost

In the case of boosting, model complexity is usually regulated by the number of trees in the ensemble. However even here, as we have seen for Random Forest, increased complexity is not always linked to an increase in variance. In fact, depending on the data at hand, boosting algorithms have the ability to reduce bias as well as variance [3]. This is verified in Figure 8, where as the number of trees increases, both bias and variance are reduced, resulting in an optimal MSE. This makes boosting methods (and AdaBoost in particular) a great tool for prediction.

[3] Schapire, Robert E., et al. “Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods.” The Annals of Statistics, vol. 26, no. 5, 1998, pp. 1651–86. JSTOR, `http://www.jstor.org/stable/120016`. Accessed 12 Dec. 2022.

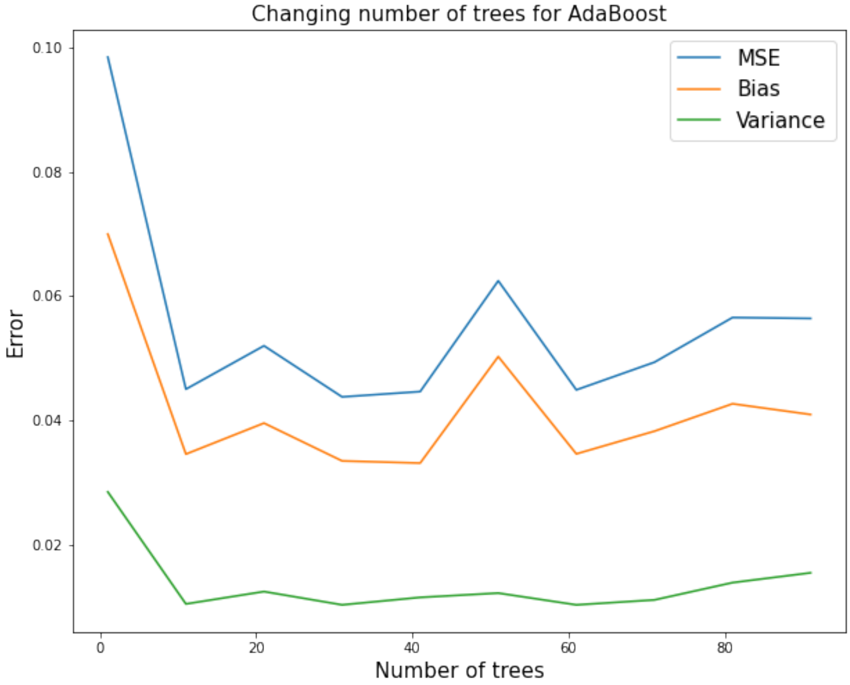

Figure 8: Bias-variance decomposition of the MSE for an AdaBoost ensemble with growing number of trees.

4 Feed-forward Neural Network

For the analysis of the FFNN we have many possible choices of variables that regulate the model complexity: in fact, the degrees of freedom of a neural network are linked to its number of coefficients, which is a result of the number of neurons and layers, and the possible penalties applied to them.

Firstly, we have chosen to study a single layer Neural Network with a sigmoid activation function and a changing number of neurons. The results are shown in Figure 9: both variance and bias are reduced when the number of neurons is increased. Results similar to this one have already been documented in previous researches, showing that there might not be a bias-variance tradeoff in neural networks with respect to network width.[4]

We also tried constructing a network with changing number of layers. For this simulation, each layer contains 20 neurons with sigmoid activation function. In this case, the results shown in Figure 10 show that when the number of layers increases excessively, the variance seems to be increased again as we would expect, giving the MSE curve the usual "U-shape".

It is important to note that this increased variance can only be detected in a deep network with over 10 layers: considering the reduced amount of data we are testing the methods on and the huge amount of coefficients that this type of network contains, we can still argue that Feed-Forward Neural Networks are resistant to overfitting.

[4] B. Neal, S. Mittal, A. Baratin, V. Tantia, M. Scicluna, S. Lacoste-Julien, I. Mitliagkas (2018). A Modern Take on the Bias-Variance Tradeoff in Neural Networks. arXiv. https://doi.org/10.48550/ARXIV.1810.08591

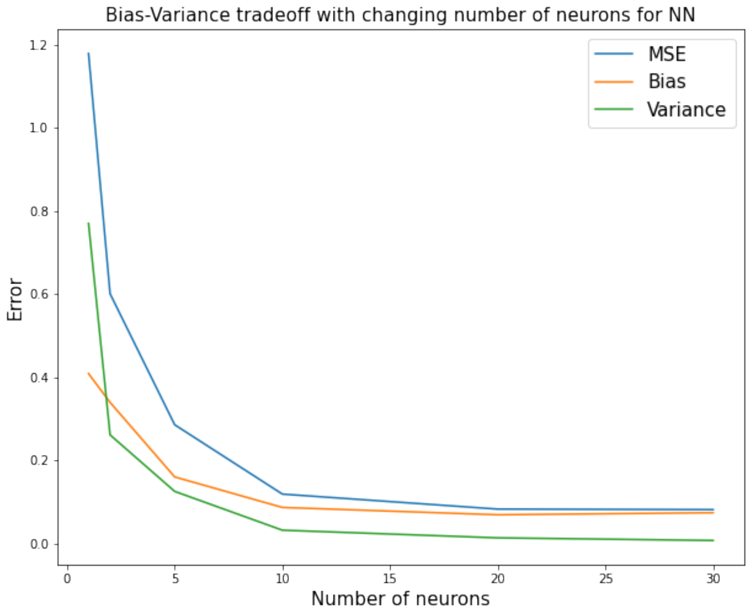

Figure 9: Bias-variance decomposition of the MSE for a single layer Neural Network with changing number of neurons.

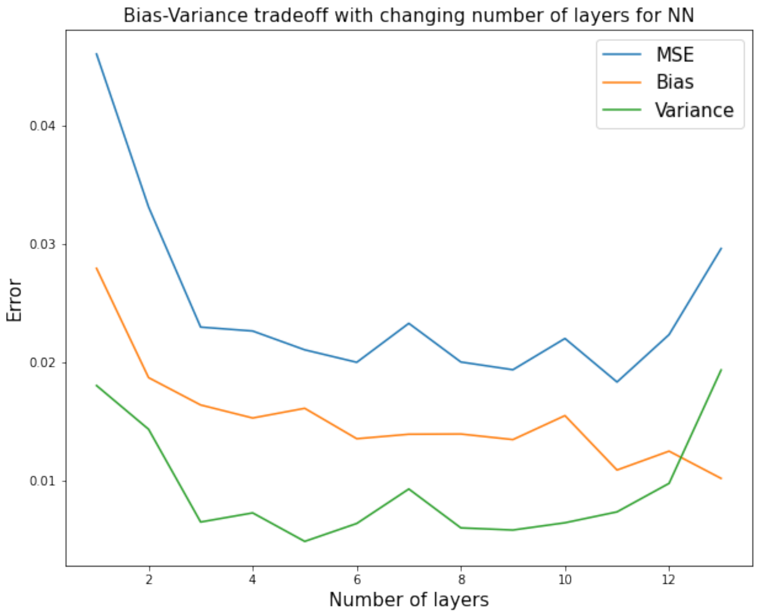

Figure 10: Bias-variance decomposition of the MSE for multiple layer Neural Network with changing number of layers. Each layer contains 20 neurons with sigmoid activation function.

5 Conclusion

The aim of this exercise was to perform a bias-variance trade-off analysis for multiple classes of regression methods by understanding which variables regulate the model complexity and how they affect the MSE and its decomposition in squared bias and model variance. We tested all methods on the same dataset generated as 100 points from a simple function with added noise.

We considered the OLS, which showed a typical reaction to the increase in model complexity, with the optimal model balancing underfitting and overfitting. Furthermore, the use of linear regression shrinkage methods (Ridge and Lasso) allows to regulate the model's variance but, if the shrinkage is too strong, cause an increase in its bias.

We also studied a regression tree with changing depth: differently from what we could have expected, increasing the tree's maximum depth doesn't affect the variance in a major way, reducing the risk of overfitting for this method. The use of ensemble methods is usually recommended to deal with decision trees' high variance. As we observed, Random Forest and AdaBoost both deal very well with an increase in the model's complexity: both bias and variance seem to be reduced as these methods progressed, making them very good regression tools.

Finally, we studied a Feed-Forward Neural Network: even in this case, to reach a point of overfitting and increased variance for the method, the number of neurons or layers considered has to greatly exceed the amount needed for the prediction on a small dataset like the one we considered. Both bias and variance are handled well by the network, resulting in a very low MSE.