Variance-bias tradeoff exercise

José Luis Barreiro Tomé

In this additional exercise to the course FYS-STK4155 I performed an analysis of the bias-variance tradeoff using three of the methods discussed in the course: Linear Regression (OLS, Ridge and Lasso), deep learning (feed forward neural network), Ensemble method (random forest) and support vector machines.

The analysis is performed on a classification problem, as a function of the complexity of the model. Here I have used the Franke function used in the Projects 1 and 2 of the course. In order to get the best possible estimates, the bootstrap resampling technique is used (75 bootstraps).

The bias-variance tradeoff:

The error for any supervised Machine Learning algorithm can be decomposed in Bias and Variance. There is a tradeoff between the model's ability to minimize these two components. Bias indicates the difference between the model's predictions and the true values. A model with high bias oversimplifies the prediction and underfit the data. The variance deals with the spread of the data. A model with high variance fit the training data closely but it's not able to fit new data which hasn't seen before. Such models are said to overfit the data. Finding the right balance between both components is called the bias-variance tradeoff.

In this exercise, I used the Franke function from Project 1 and 2 to assess the bias-variance tradeoff as a function of the model complexity. Linear regression, Feed-Forward-Neural-Network (FFNN) and Random Forest (RF) are the three regression methods assessed.

The bias-variance tradeoff for linear regression:

Ordinary-Least-Squares (OLS), Ridge and Lasso regression are assessed as a function of the model complexity. Figure 1 shows the errors as the polynomial degree increases, first up to degree 8 and in the right just to degree 5. The error keeps quite low until the polynomial degree 4. The variance increases dramatically thereafter, while the bias remains low. This is an indicative of overfitting.

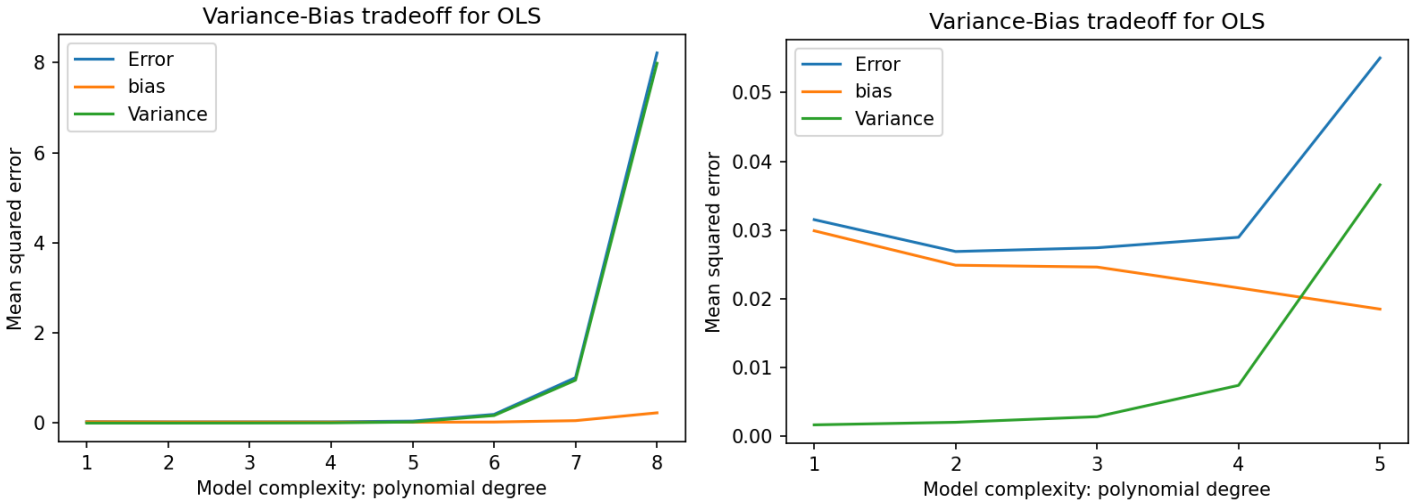

***Figure 1**. Bias-variance tradeoff for OLS*

For Ridge regression, the hyperparameter lambda was set at 1.e-6, and the error was tested as a function of the polynomial degree. Figure 2 shows how the bias and variance remain low at degree of complexity 5-12, and the variance increase greatly after that, while bias stay low. For small degree of complexity (1-5), it's the variance who stay lower than the bias, underfitting the data.

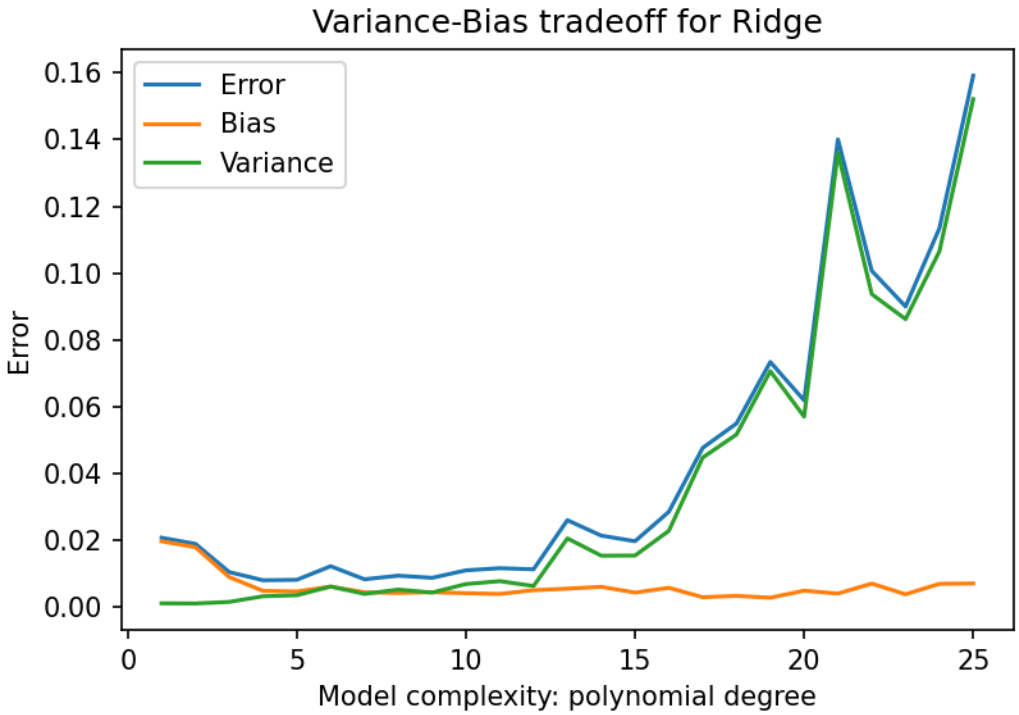

***Figure 2**. Bias-variance tradeoff for Ridge regression*

As in Ridge, the hyperparameter was set constant in the case of Lasso regression. Figure 3 shows how the bias stays higher than the variance for the whole range of assessed

polynomial degree. Nevertheless, the level of error is lower than in the other linear regression methods.

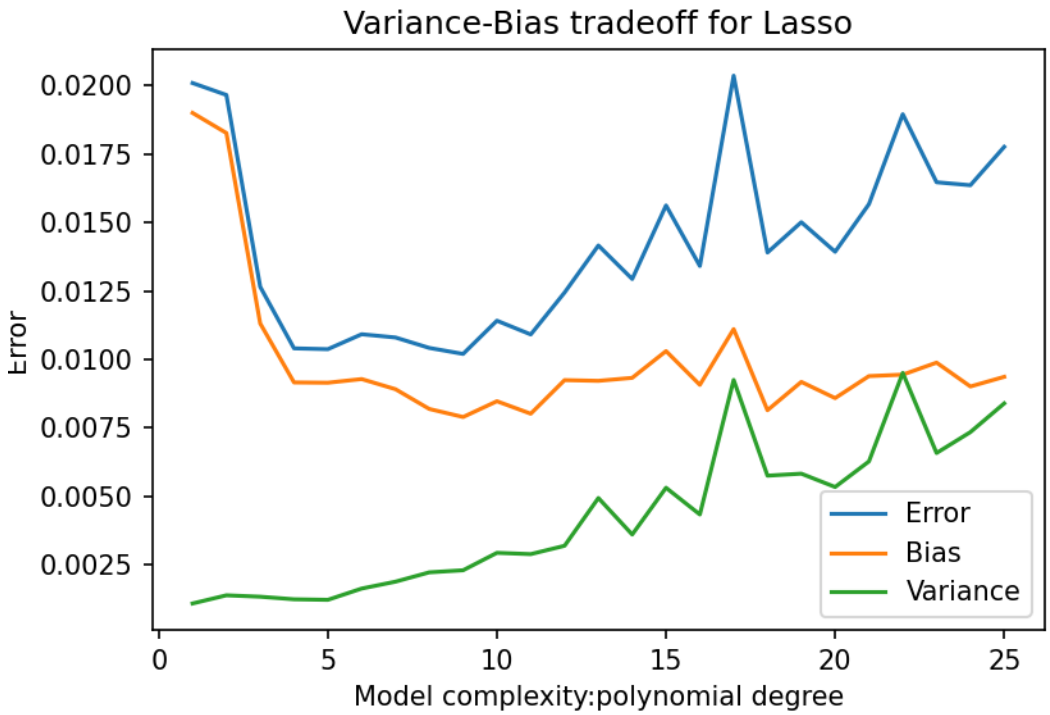

***Figure 3**. Bias-variance tradeoff for Lasso regression*

The bias-variance tradeoff for Neural Network:

The level of complexity in the FFNN is determined by the number of neurons per layer and the number of layers. Figure 4 shows the bias-variance tradeoff for both approaches. The number of neurons in one layer doesn't affect the level of error. The bias stays much higher than the variance for the whole range tested. The same happened for the number of layers. Therefore, the complexity of the model should be kept at a low level.

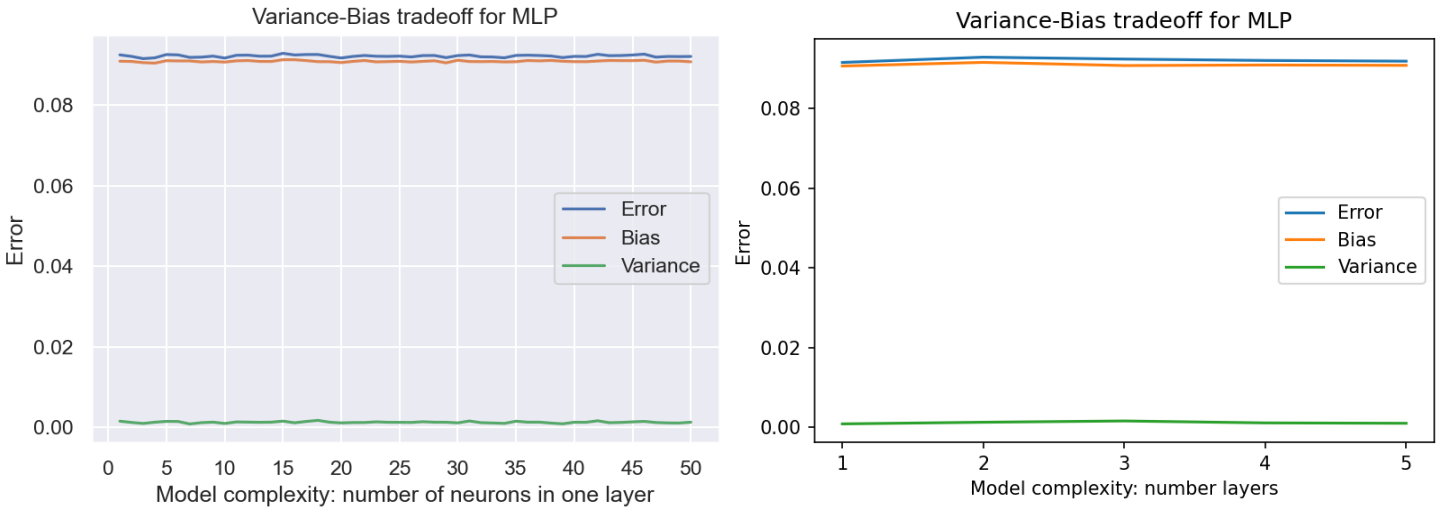

***Figure 4**. Bias-variance tradeoff for MLP*

The bias-variance tradeoff for Random Forest:

The level of complexity for RF is determined in this example by the maximum depth of the tree (*max_depth* in *Scikit learn*), and the minimum number of samples required to be at a leaf node (*min_samples_leaf*). They represent constrains by depth and by leaf, respectively. Figure 5 shows how the variance stays low during the whole test for both cases. Regarding the depth of the tree, bias and variance stabilized after depth 3. For the *min_samples_leaf*, the bias starts to increase dramatically after the value of 10, underfitting.

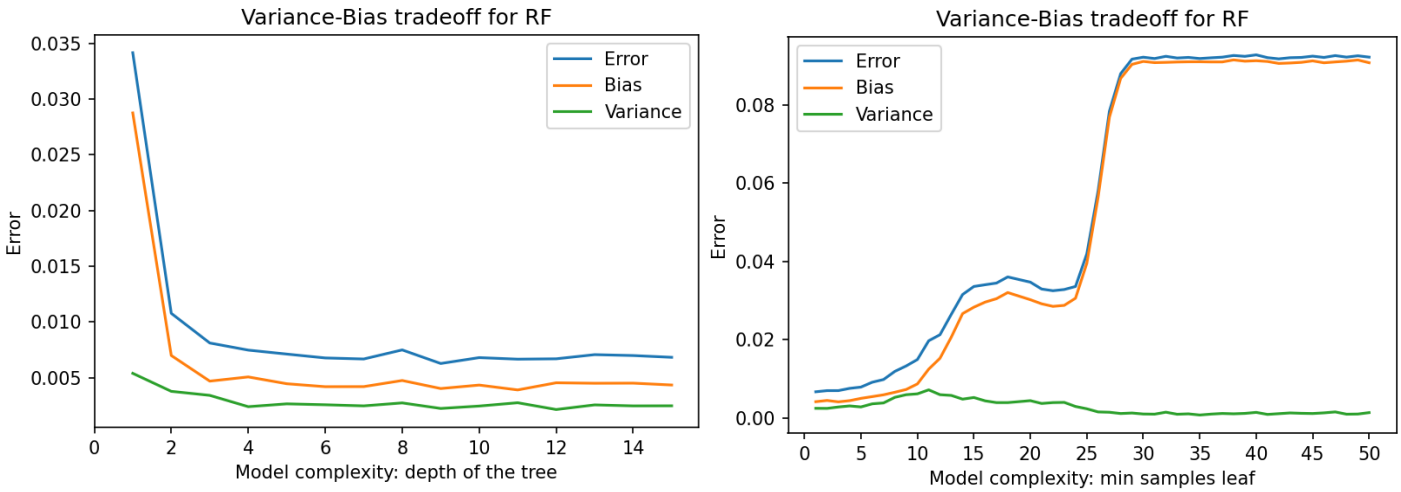

***Figure 5.** Bias-variance tradeoff for Random Forest*

Conclusions:

Random Forest and linear regression (OLS, Ridge and Lasso) gave the smallest level of errors. They all seem to have low bias and low model for certain degrees of complexity. MLP has the highest bias of all tested methods.