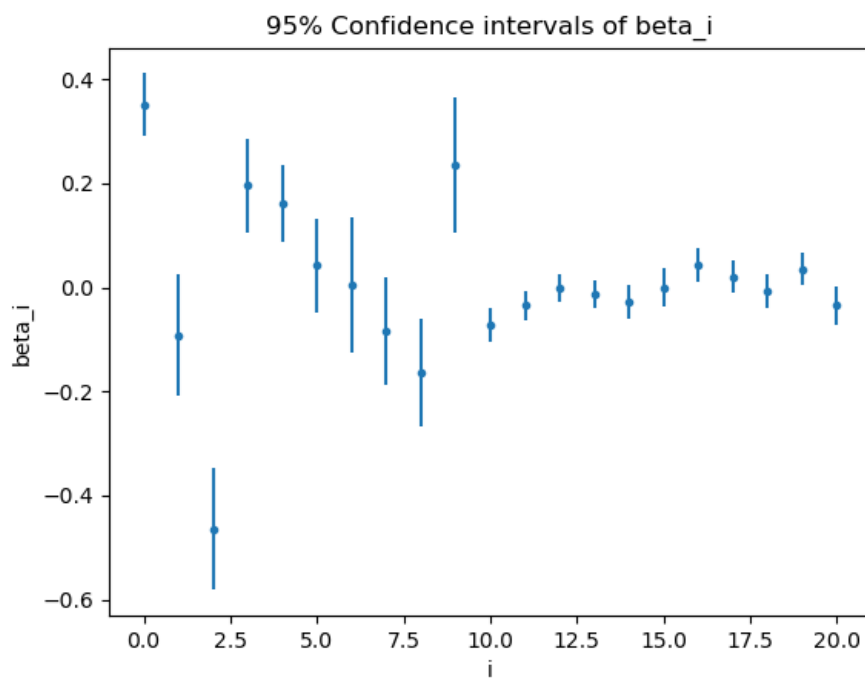


FYS-STK4155 Project 1

Espen Lønes

October 14, 2021

Exercise 1:



We see that the betas corresponding to lower degree parameters are larger in absolute value. Telling us these parameters are more important according to the model. But these also have the largest uncertainty.

Output from ex1.py:

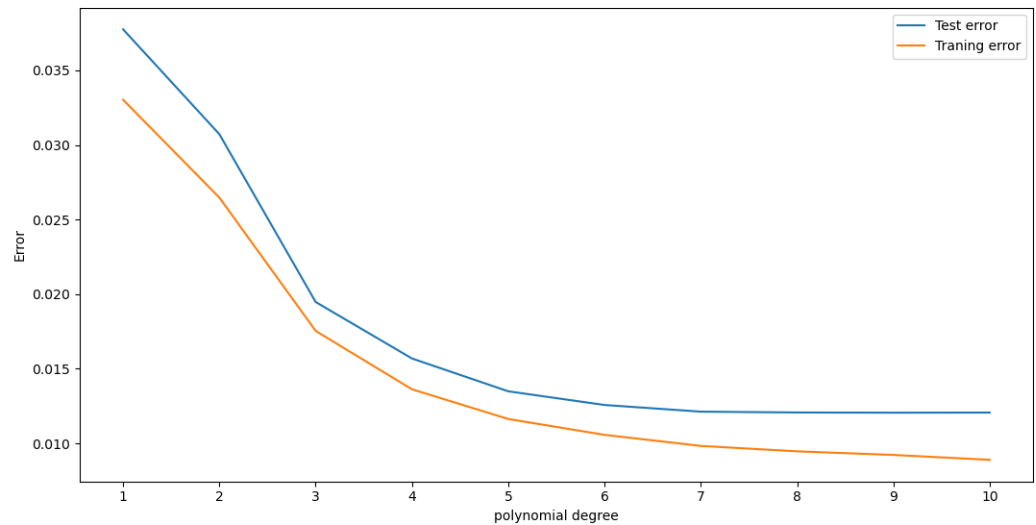
MSE train: 0.011979138954366192

MSE test: 0.012656233563893443

R2 train:0.8736114124595407

R2 test:0.8483730829731215

Exercise 2:



The test error indicates that for low polynomial degrees we probably have high bias compared to higher polynomial degrees.

As for the equation.

We have:

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

To Derive the wanted equation we use the fact that the variance of \mathbf{y} and ϵ are both σ^2 . The mean of ϵ is zero and \mathbf{f} is not stochastic for $\tilde{\mathbf{y}}$. And using the more compact notation of expected value, we get.

$$\mathbb{E}[(\mathbf{y} - \tilde{\mathbf{y}})^2] = \mathbb{E}[(\mathbf{f} + \epsilon - \tilde{\mathbf{y}})^2]$$

Then add and subtract $\mathbb{E}[\tilde{\mathbf{y}}]$

$$\mathbb{E}[(\mathbf{y} - \tilde{\mathbf{y}})^2] = \mathbb{E}[(\mathbf{f} + \epsilon - \tilde{\mathbf{y}} + \mathbb{E}[\tilde{\mathbf{y}}] - \mathbb{E}[\tilde{\mathbf{y}}])^2]$$

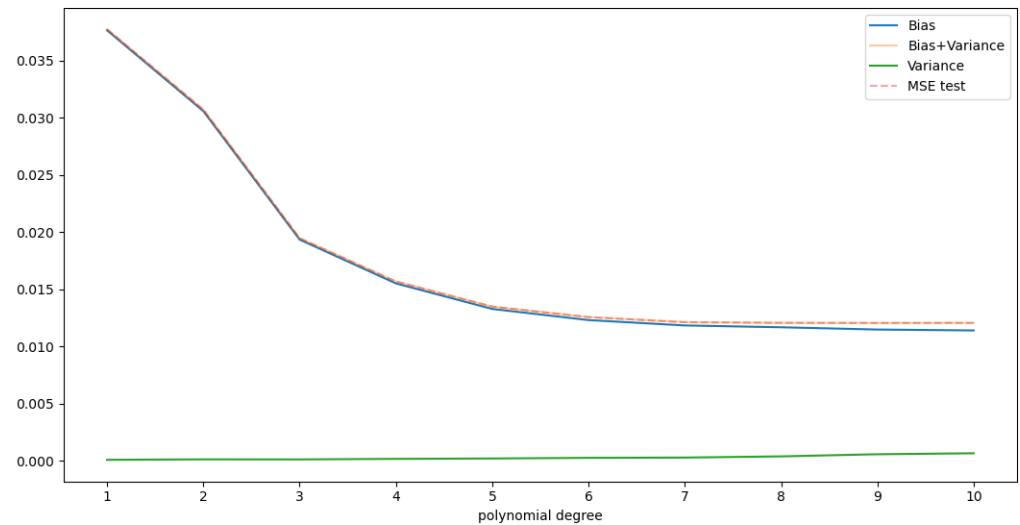
Then we use the expectation values mentioned above and get.

$$\mathbb{E}[(\mathbf{y} - \tilde{\mathbf{y}})^2] = \mathbb{E}[(\mathbf{y} - \mathbb{E}[\tilde{\mathbf{y}}])^2] + Var[\tilde{\mathbf{y}}] + \sigma^2$$

$$\mathbb{E}[(\mathbf{y} - \tilde{\mathbf{y}})^2] = \frac{1}{n} \sum_i (f_i - \mathbb{E}[\tilde{\mathbf{y}}])^2 + \frac{1}{n} \sum_i (\tilde{y}_i - \mathbb{E}[\tilde{\mathbf{y}}])^2 + \sigma^2$$

The three terms in this equation represent, in order. The bias of the model, the variance of the model and lastly the variance of the error.

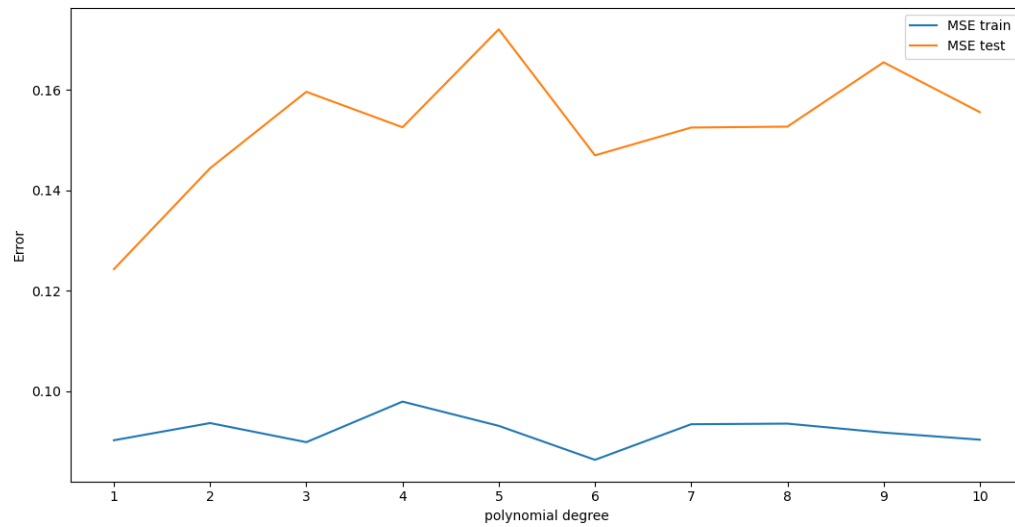
Now for the bias-variance trade-off



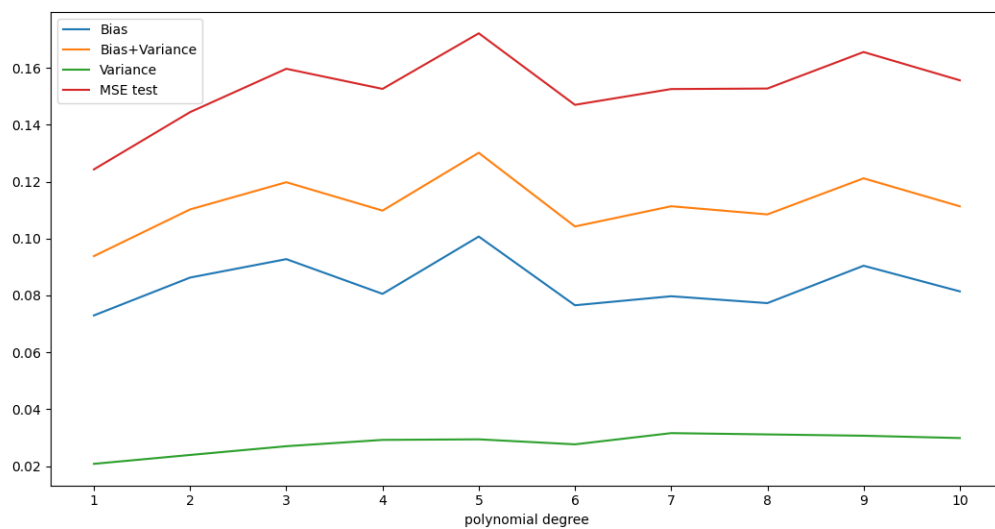
We see as expected that for increasing polynomial complexity we go from, high bias and low variance, to low bias and high variance. If we had used more points in the training data we would get more accurate values for the bias and variance (and error). The same if we used more bootstraps resamples. But for bootstrap resamples we are just getting closer to the true values for the samples we have. But if we generate more data from the franke function we get closer to the actual values for the franke function.

Exercise 3:

Cross-validation MSE:



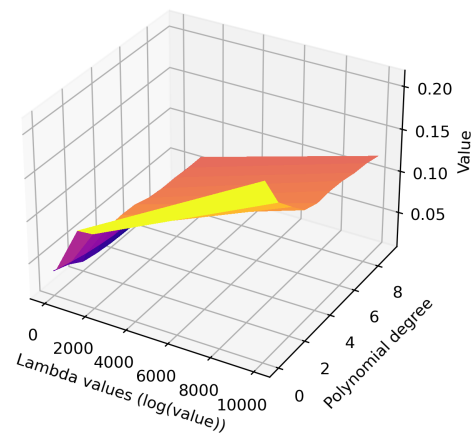
Cross-validation Bias-Variance:



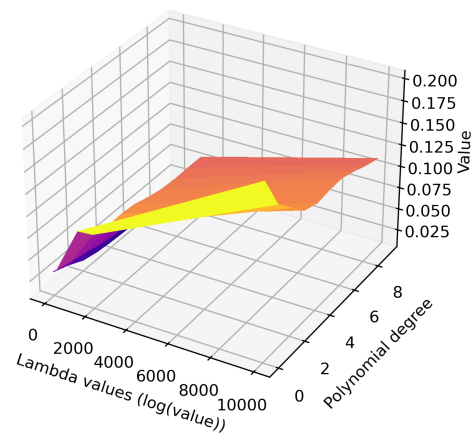
We see that the errors stay relatively constant and about 10 times higher than for bootstrap. This shows that cross-validation does not work well for this problem. We see similar effects with the bias.

Exercise 4:

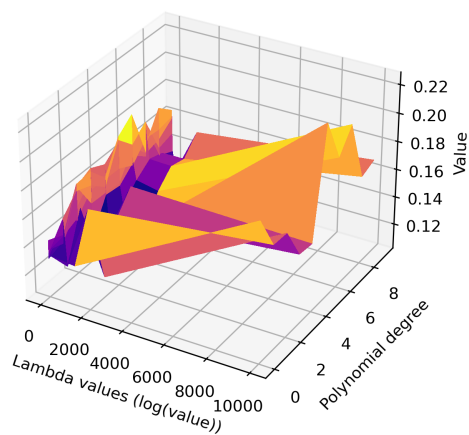
Test error bootstrap:



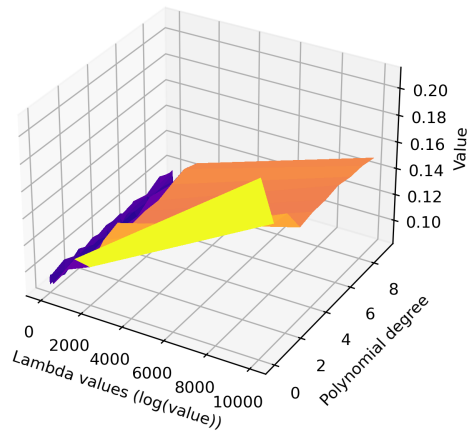
Train error bootstrap:



Test error cross-validation:



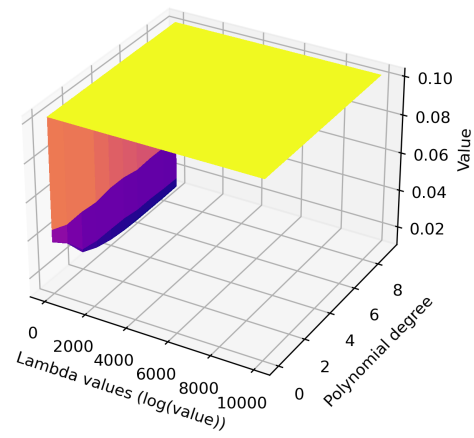
Train error cross-validation:



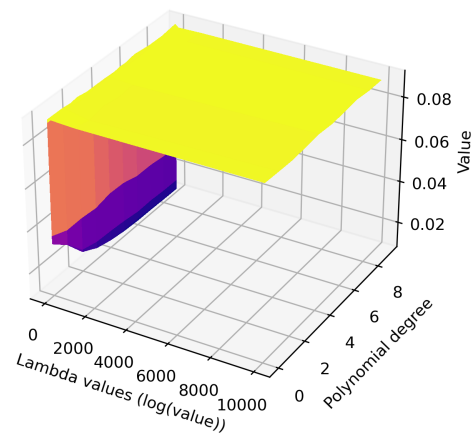
We see much the same as for OLS in the polynomial degree direction. As for the lambdas we see that it works best for lambda close to 0. Which is the same as doing normal OLS.

(I did also make plots for bias and variance, but did not include them in the report as the task does not ask for them. If interested look in the images folder.)

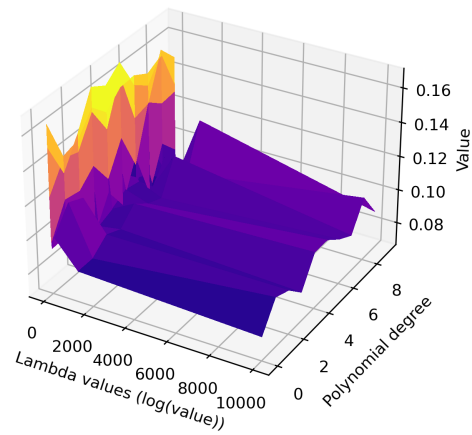
Exercise 5:
Test error bootstrap:



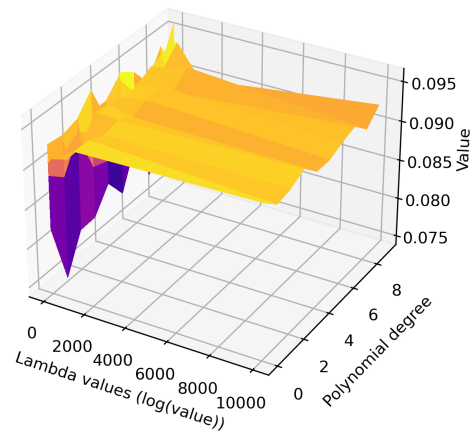
Train error bootstrap:



Test error cross-validation:



Train error cross-validation:



We have a similar result as from exercise 4. Where best result is from small lambdas, (i think something went wrong with test error cross-validation, should probably be flipped). But here the polynomial degree seems to have no effect on the error.

(I did also make plots for bias and variance, but did not include them in the report as the task does not ask for them. If interested look in the images folder.)

Exercise 6:

Did not have time to finish this task. But you may look at `ex6.py` to get an idea of what I wanted to do.