FYS-STK3155 Project 1

Lasse Pladsen, Parham Qanbari, & Sander V. Vattøy

October 6, 2023

Abstract

We have studied three different regression methods, ordinary least squares, ridge, and lasso, and what affects their learned prediction accuracy. We found ... . We have done a bias-variance analysis on OLS using a bootstrap resampling technique and found that ... . We then study the cross-validation resampling method and analyze all three regression methods ... . Finally we applied our models to real world topographic data and we saw that ... .

I. INTRODUCTION

In today's modern world machine learning has emerged as a revolutionary technology. Even if just superficially, machine learning has become well known around the world. By allowing computers to learn from supplied data we create powerful tools with incredible real world applications for analyzation and prediction.

In this project we will explore and study low-level machine learning methods. Our focus will be on understanding and optimizing the potential of data regression methods such as ordinary least squares (OLS), ridge regression, and lasso regression. We will look into the factors that influence their accuracy and predictive capabilities, employing a variety of strategies to minimize errors and improve performance. Finally, we will apply our finely-tuned algorithms to analyze real-world topographic data.

II. THEORY

A. Design matrix for linear regression

We create a so-called design matrix $\mathbf{X}$ for the regression methods from two input variables $\mathbf{x}$ and $\mathbf{y}$. This matrix is used to create our linear regression models. Each row in $\mathbf{X}$ represents polynomial variables from one data sample. We choose a max polynomial degree p with n data samples such that $\mathbf{X} \in \mathbb{R}^{n \times l}$ where

$$l = floor[\frac{1}{2}(p+1)(p+2)],$$

where the $floor$-function rounds down, and the design matrix is then given by (Hjorth-Jensen, 2023)

$$\mathbf{x} = \begin{bmatrix} 1 & x_0 & y_0 & x_0^2 & x_0 y_0 & y_0^2 & \cdots & x_0^{p-1} & \cdots & y_0^{p-1} \\ 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 & \cdots & x_1^{p-1} & \cdots & y_1^{p-1} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & y_{n-1} & \cdots & \cdots & \cdots & \cdots & x_{n-1}^{p-1} & \cdots & y_{n-1}^{p-1} \end{bmatrix} \tag{1}$$

We then make our linear regression predictions by creating the polynomials from $\mathbf{X}$ and the polynomial coefficients $\boldsymbol{\beta} \in \mathbb{R}^l$ as follows (Hjorth-Jensen, 2023)

$$\tilde{\mathbf{z}} = \boldsymbol{X\beta} \tag{2}$$

B. Regression methods

The different methods have different ways of calculating the optimal $\boldsymbol{\beta}$-coefficients which we call $\hat{\boldsymbol{\beta}}$. The following expressions are derived in the course lecture notes (Hjorth-Jensen, 2023) unless otherwise noted.

B1. Ordinary least squares

In the method ordinary least squares we find our optimal coefficients $\hat{\boldsymbol{\beta}}$ by minimizing the squared difference between $\mathbf{y}$ and $\tilde{\mathbf{y}}$:

$$\frac{1}{n}(y_i - \tilde{\mathbf{y}}_i)^2 = \frac{1}{n}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2$$

where we use the norm-2 definition for a vector

$$||\mathbf{x}||_\mathbf{2} = \sqrt{\sum_\mathbf{i} \mathbf{x_i^2}}$$

. We then attain the following expression

$$\hat{\boldsymbol{\beta}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{z} \tag{3}$$

B2. Ridge

For the ridge regression method we add a regularization parameter λ such that we minimize the following expression

$$\frac{1}{n}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||x_i||_1$$

This leads to the following expression

$$\hat{\boldsymbol{\beta}}_{Ridge} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{z} \tag{4}$$

where $\mathbf{I}$ is the identity matrix. In addition the parameter must be non-negative $\lambda \geq 0$.

B3. Lasso

Lasso regression stands for 'least absolute shrinkage and selection operator' where we instead want to minimize

$$\frac{1}{n}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||x_i||_1 \tag{5}$$

where we here also use the the norm-1 definition for a vector

$$||\mathbf{x}||_2 = \sum_{\mathbf{i}} |\mathbf{x_i}|$$

. We do not have an analytical expression for $\boldsymbol{\beta}_{Lasso}$ so this needs to be numerically calculated.

C. Measuring error and coefficient of determination

To calculate our regression models' prediction error from the actual data we will be using the mean square error (MSE) defined by

$$MSE(\mathbf{z}, \tilde{\mathbf{z}}_i) = \frac{1}{n} \sum_{i=0}^{n-1} (z_i - \tilde{z}_i)^2, \tag{6}$$

The coefficient of determination, denoted R^2, is a measure that tells us how well our models can predict new data and is defined as

$$R^2(\mathbf{z}, \tilde{\mathbf{z}}_i) = 1 - \frac{\sum_{i=0}^{n-1} (z_i - \tilde{z}_i)^2}{\sum_{i=0}^{n-1} (z_i - \bar{z})^2} \tag{7}$$

where $\bar{z}$ is the mean value of $\mathbf{z}$, and a value of $R^2 = 1$ will represent the best possible model determination from our regression method.

D. Resampling

Resampling is to repeatedly gather samples from the same data set to increase a model's performance or to possibly obtain extra information which otherwise would not be possible with only one sample/data split. When it comes to the linear regression methods of this project we can resample our data's training split then do a regression fit on each sample. By using the average we can hopefully the model's accuracy. The two resampling techniques we are going to use in this project is bootstrap and k-fold cross-validation.

E. Bootstrap

The bootstrap resampling technique uses random sample selection from the data set n number of iterations with replacement. Because we replace each data sample before every iteration some data may be sampled more than once and some may never be sampled.

F. k-fold cross-validation

The k-fold cross-validation resampling technique is where the data set gets split into k number of equally sized subsets (folds) containing the data. A single fold is used for testing the model, and the rest $k - 1$ number of folds is then used to train the model. This is repeated k times where every iteration a different fold is used for the testing and the rest for training (Hjorth-Jensen, 2023).

G. Bias and variance

Expressing (6) as the expectation value

$$MSE = \mathbb{E}[(\mathbf{z} - \tilde{\mathbf{z}})^2]$$

we can rewrite this as

$$MSE = \text{Bias}[\tilde{\mathbf{z}}] + \text{var}[\tilde{\mathbf{z}}] + \sigma^2 \tag{8}$$

where

$$\text{Bias}\left[\tilde{\mathbf{z}}\right] = \mathbb{E}\left[(\mathbf{z} - \mathbb{E}\left[\tilde{\mathbf{z}}\right])^2\right] \tag{9}$$

and

$$\text{var}\left[\tilde{\mathbf{z}}\right] = \mathbb{E}\left[(\tilde{\mathbf{z}} - \mathbb{E}\left[\tilde{\mathbf{z}}\right])^2\right] = \frac{1}{n} \sum_i (\tilde{z}_i - \mathbb{E}\left[\tilde{\mathbf{z}}\right])^2 \tag{10}$$

Here σ is the standard deviation of our stochastic noise $\boldsymbol{\epsilon}$. The derivations of (8) can be found in the appendix section C.

Here we see that the MSE is just the sum of the bias and the variance of the model, in addition to the σ^2 term (the noise variance). Figure 1 shows how the idealized regions of low vs high bias and variance im comparision to the MSE. The figure also shows an example of overfitting. This is where we fit the model too close to the training data which results in less prediction performance for unseen data like the testing data set.

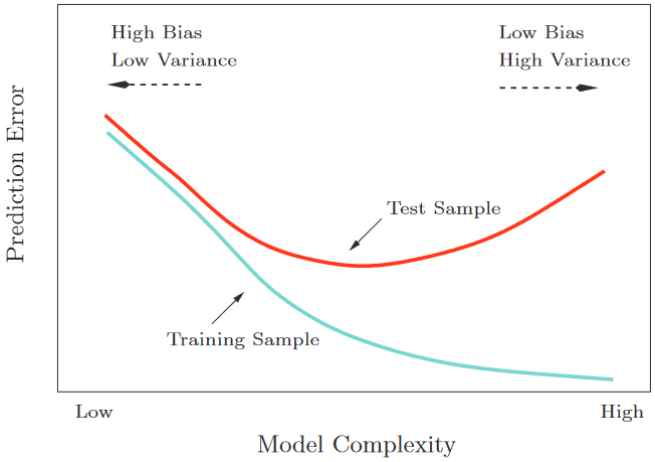


Figure 1: Figure 2.11 from Hastie et al. (2009). This idealized graph shows the bias-variance trade-off for our models, and shows the concept of overfitting model to the training data. This happens when we use too high complexity for the model leading to higher errors and worse prediction capabilities.

H. The Franke function

To study our regression methods we will be using the Franke function, which is a two dimensional weighted

sum of four exponentials given as

$$f(x,y) = \frac{3}{4}\exp\left[-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right] + \frac{3}{4}\exp\left[-\frac{(9x+1)^2}{49} - \frac{(9y+1)^2}{10}\right] + \frac{1}{2}\exp\left[-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right] - \frac{1}{5}\exp\left[-(9x-4)^2 - (9y-7)^2\right] \quad (11)$$

III. METHODS

In this project we will write our code in Python, and we will many functionalities of the well used machine learning python module `SciKit-Learn`. Our first step is to study how the MSE and R^2 changes as the model complexity changes using our three regression methods. For this we will vary the complexity by changing the maximum polynomial degree p in our design matrix $\mathbf{X}$, where we will go up to $p = 6$.

We will be using the Franke function (11) with $x_i, y_i \in [0, 1]$ while simulating a stochastic noise from a normal distribution $\epsilon \sim \mathcal{N}(0, 1)$. We will split our data into a training set and a testing set using `Scikit-Learn`'s `model_selection.train_test_split` function.

Firstly we will do the OLS regression, writing our own code from (3), then we will do a ridge regression from (4) with varying λ-parameter, to find the error in each case. Thirdly we will use `Scikit-Learn`'s implementation of lasso regression (`sklearn.linear_model.Lasso`). For each of these regression methods we will study the prediction accuracy with respect to varying maximum polynomial degree p.

Next up we will be introducing bootstrap resampling where we use `Scikit-Learn`'s implementation `sklearn.utils.resample` with 1000 resamples. Thereafter we do a bias-variance trade-off analysis of the OLS regression, and recreate the theoretical low vs high bias and variance. From this we will find how the mean square error changes with respect to polynomial degree.

Now we create a code for the cross-validation resampling method, where we use `Scikit-Learn`'s implementation `sklearn.model_selection.KFold`. Here we will study the MSE of all three regression methods using cross-validation with different fold parameters $k \in [5, 10]$, and thereafter compare the error with the result found using the bootstrap method.

The final thing we will preform is using the three regression methods discussed, on a real data set over Oslo, Norway. This way we will try to make predictions.

IV. RESULTS

Firstly we find the following results from the OLS-regression shown in figure 2.

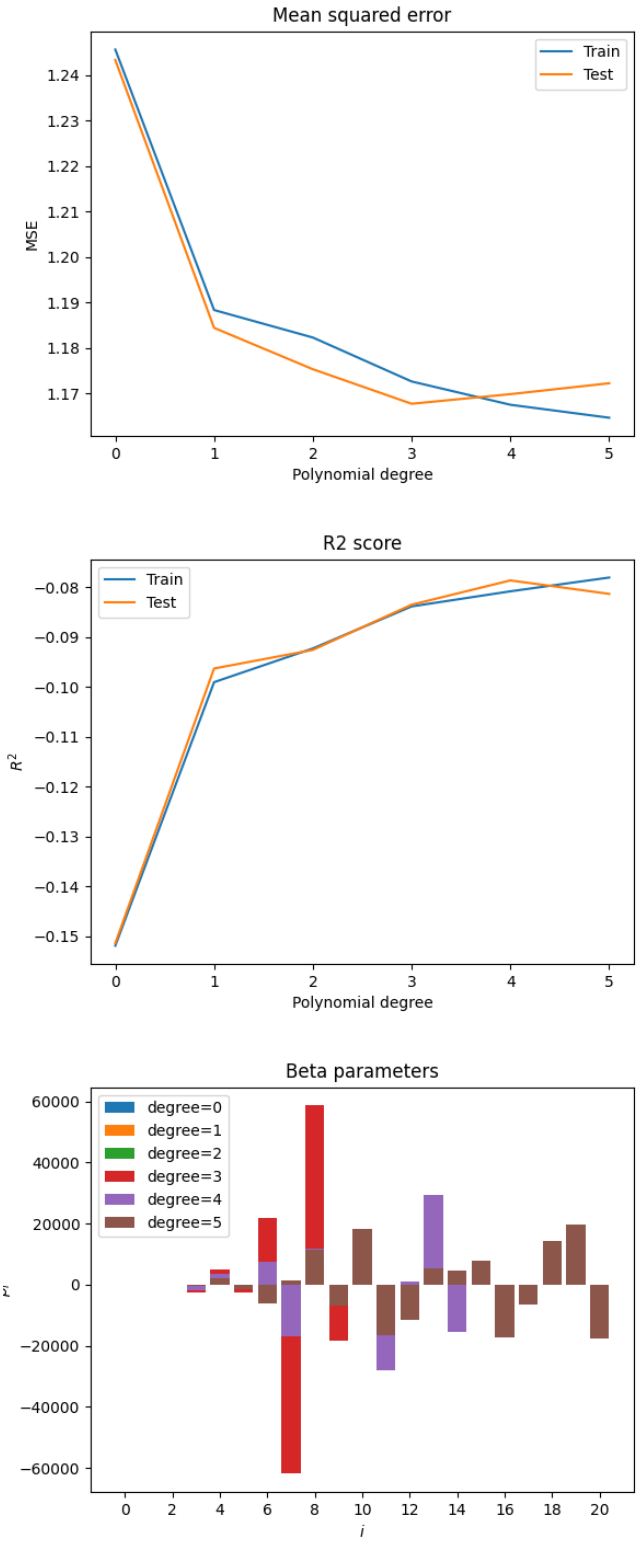
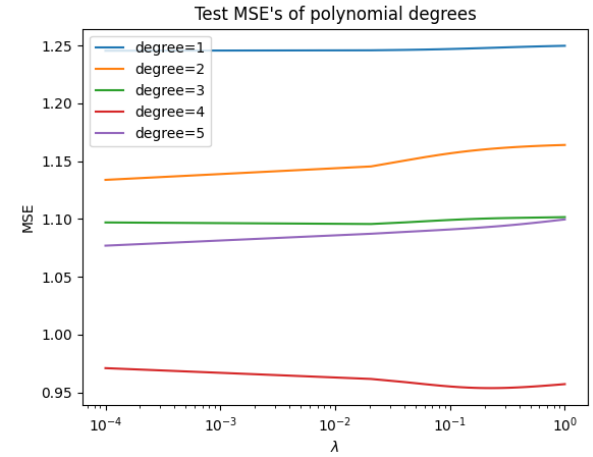


Figure 2: Caption.

Then we find the related results from the Ridge-regression shown in figure 3.

For the Lasso regression we find the following shown



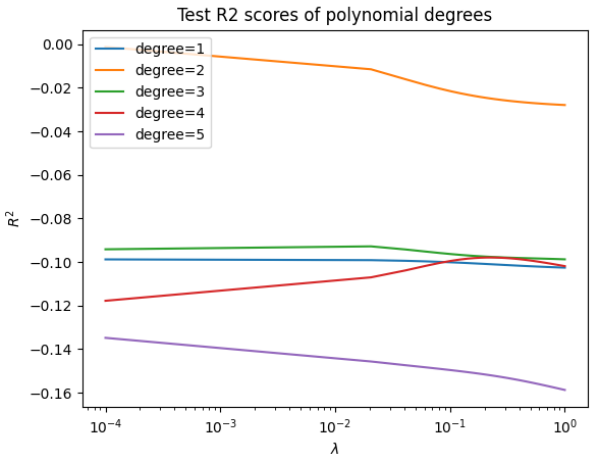
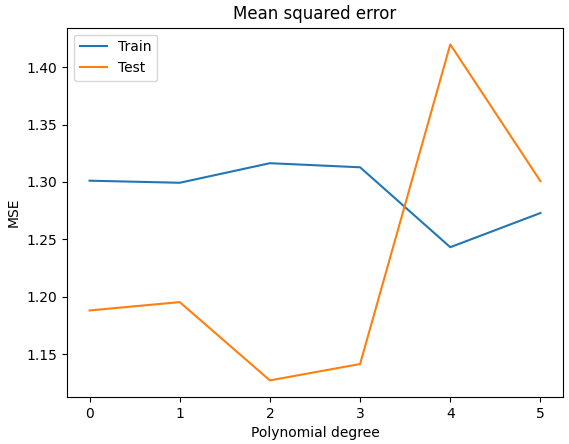


Figure 3: Caption.



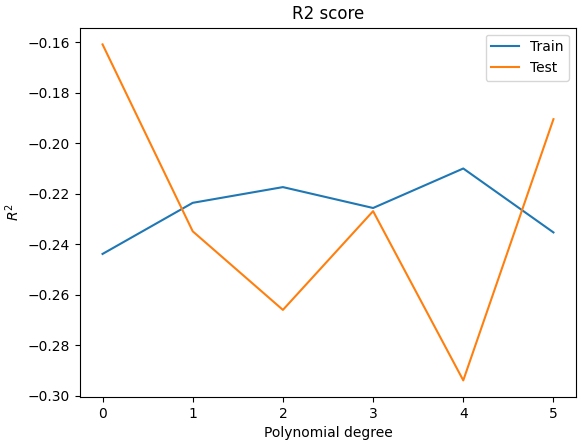


Figure 4: Caption.

in figure 4.

We then do the bias-variance-tradeoff, and get the results in figure ...

Thereafter we do the cross-validation, which end up producing the error in figure ... Here we have done the resampling over $k \in [5, 10]$, but only included $k = 5, 10$ because we see a steady progression of the graphs.

We now find the data for the real-data problem...

V. DISCUSSION

VI. CONCLUSION

Appendix A. Github repository

https://github.com/LassePladsen/FYS-STK3155-projects/tree/main/project1

Appendix B. List of source code

Here is a list of the code we have developed in this project which can be found in the above Github repository:

- ...
- ...
- ...

Appendix C. Analytical derivations

Subsection 1 is for part e of the project, and all the other subsections are for part d.

C1. Bias-variance: equation (8)

This is the derivation of equation (8) for part e of the project. First we write the MSE (6) as the following expectation value

$$MSE = \mathbb{E}[(z - \tilde{z})^2]$$

the we multiply the paranthesis and split the expression to three expectation values. For the rest of these derivations in this project we will write vectors and matrices without boldface for ease of writing $\mathbf{z} = z, \mathbf{X} = X$ etc.

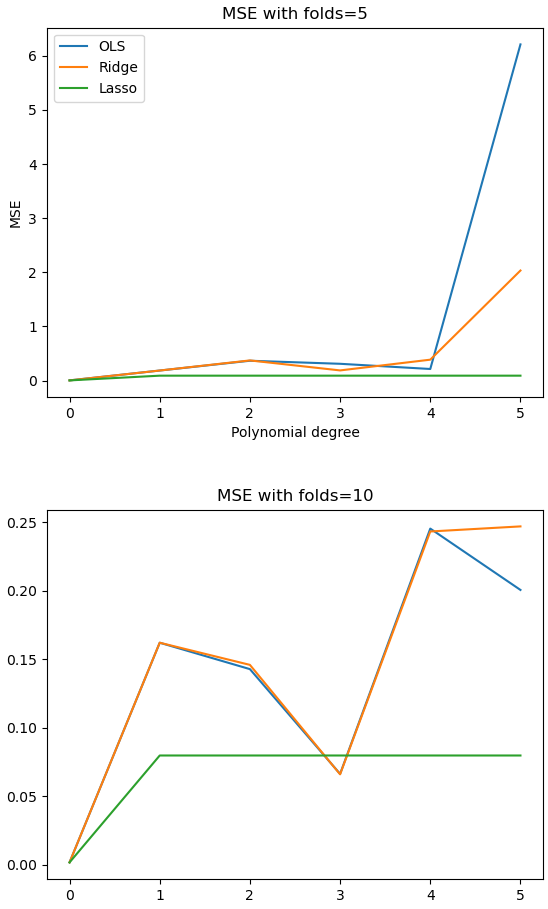


Figure 5: Caption.

We write

$$\begin{aligned} MSE &= \mathbb{E}\left[z^2 - 2z\tilde{z} + \tilde{z}\right] \\ &= \mathbb{E}[z^2] - 2\mathbb{E}[z\tilde{z}] + \mathbb{E}[\tilde{z}^2] \\ &= \mathbb{E}[z^2] - 2\mathbb{E}[z\tilde{z}] + \mathbb{E}[\tilde{z}^2] \end{aligned}$$

We will split this up and take on these three terms one by one, firstly we write $z = f + \epsilon$ which represents our data with noise. For the first term:

$$\begin{aligned} \mathbb{E}[z^2] &= \mathbb{E}[(f+\epsilon)^2] \\ &= \mathbb{E}[f^2 + 2f\epsilon + \epsilon^2] \\ &= \mathbb{E}[f^2] + \mathbb{E}[2f\epsilon] + \mathbb{E}[\epsilon^2] \end{aligned}$$

here we use that $\mathbb{E}[\epsilon] = 0, \mathbb{E}[\epsilon^2] = \sigma^2$ since $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

$$\begin{aligned} &= \mathbb{E}[f^2] + 2f\mathbb{E}[\epsilon] + \sigma^2 \\ &= \underline{\mathbb{E}[f^2] + \sigma^2} \end{aligned}$$

Now for the second term:

$$\begin{aligned} \mathbb{E}[z\tilde{z}] &= \mathbb{E}[(f+\epsilon)\tilde{z}] \\ &= \mathbb{E}[f\tilde{z} + \epsilon\tilde{z}] \\ &= \mathbb{E}[f\tilde{z}] + \mathbb{E}[\epsilon\tilde{z}] \\ &= \mathbb{E}[f\tilde{z}] + \mathbb{E}[\epsilon]\mathbb{E}[\tilde{z}] \\ &= \underline{\mathbb{E}[f\tilde{z}]} \end{aligned}$$

Now for the third and final term:

$$\mathbb{E}[\tilde{z}^2] = \underline{var[\tilde{z}] + (\mathbb{E}[\tilde{z}])^2}$$

where we have used the definition of variance $var(x) = \mathbb{E}[x^2] - (\mathbb{E}[x])^2$.

We now put these three terms into our expression for MSE:

$$\begin{aligned} MSE &= \mathbb{E}\left[(z-\tilde{z})^2\right] \\ &= \mathbb{E}[z^2] - 2\mathbb{E}[z\tilde{z}] + \mathbb{E}[\tilde{z}^2] \\ &= \mathbb{E}[f^2] + \sigma^2 - 2\mathbb{E}[f\tilde{z}] + var[\tilde{z}] + (\mathbb{E}[\tilde{z}])^2 \\ &= \mathbb{E}[f^2] - 2f\mathbb{E}[\tilde{z}] + (\mathbb{E}[\tilde{z}])^2 + \sigma^2 + var[\tilde{z}] \\ &= \mathbb{E}\left[(f - \mathbb{E}[\tilde{z}])^2\right] + var[\tilde{z}] + \sigma^2 \\ &\simeq \mathbb{E}\left[(z - \mathbb{E}[\tilde{z}])^2\right] + var[\tilde{z}] + \sigma^2 \\ &= \underline{Bias[\tilde{z}] + var[\tilde{z}] + \sigma^2} \end{aligned}$$

here we approximated $f \simeq z$ and used the expressions for Bias (9) and variance (10).

C2. Expectation value of y

We will show that $\mathbb{E}(y_i) = X_{i,*}\beta$ by using $y = f + \epsilon \simeq X\beta + \epsilon$ and separating the expectation value of a sum. Here we approximated f with $X\beta$ using OLS. Then taking a value of y with index i we get $y_i = \sum_j X_{ij}\beta_j + \epsilon_i$:

$$\begin{aligned} \mathbb{E}(y_i) &= \mathbb{E}(\Sigma_j X_{ij}\beta_j + \epsilon_i) \\ &= \mathbb{E}(\Sigma_j X_{ij}\beta_j) + \mathbb{E}(\epsilon_i) \\ &= \mathbb{E}(\Sigma_j X_{ij}\beta_j) \\ &= \Sigma_j X_{ij}\beta_j \\ &= \underline{X_{i,*}\beta} \end{aligned}$$

C3. Variance of y

Using the same method as above we will now show $Var(y_i) = \sigma^2$ where σ^2 is the variance of our data's stochastic noise ϵ. Here we use the definition of variance being $Var(x) = \mathbb{E}(x^2) - \mathbb{E}(x)^2$:

$$
\begin{aligned}
Var(y_i) &= \mathbb{E}(y_i^2) - \mathbb{E}(y_i)^2 \\
&= \mathbb{E}[(X_{i,*}\beta + \epsilon_i)^2] - (X_{i,*}\beta)^2 \\
&= \mathbb{E}[(X_{i,*}\beta^2 + \epsilon_i^2 + 2X_{i,*}\beta\epsilon_i] - (X_{i,*}\beta)^2 \\
&= \mathbb{E}[(X_{i,*}\beta)^2] + \mathbb{E}[\epsilon_i^2] + \mathbb{E}[2X_{i,*}\beta\epsilon_i] - (X_{i,*}\beta)^2 \\
&= (X_{i,*}\beta)^2 + \mathbb{E}[\epsilon_i^2] + 2X_{i,*}\beta\mathbb{E}[\epsilon_i] - (X_{i,*}\beta)^2 \\
&= \mathbb{E}[\epsilon_i^2] \\
&= Var(\epsilon_i) + \mathbb{E}(\epsilon)^2 \\
&= Var(\epsilon_i) \\
&= \underline{\sigma^2}
\end{aligned}
$$

C4. Expectation value of β_{OLS}

Here we will show that the expectation value for the optimal β for OLS, $\hat{\beta}_{OLS}$, equals β_{OLS}:

$$
\begin{aligned}
\mathbb{E}(\hat{\beta}_{OLS}) &= \mathbb{E}[(X^TX)^{-1}X^Ty] \\
&= (X^TX)^{-1}X^T\mathbb{E}[y] \\
&= (X^TX)^{-1}X^TX\beta_{OLS} \\
&= \underline{\beta_{OLS}}
\end{aligned}
$$

Here we used that the expectation value of the non-stochastic matrix is just the matrix itself ($\mathbb{E}(X) = X$) since it has zero variance (non-stochastic).

C5. Variance of β_{OLS}

Here we will show $Var(\hat{\beta}_{OLS}) = \sigma^2(X^TX)^{-1}$:

$$
\begin{aligned}
Var(\hat{\beta}_{OLS}) &= \mathbb{E}[(\hat{\beta}_{OLS})^2] - \mathbb{E}[\hat{\beta}_{OLS}]^2 \\
&= \mathbb{E}[(X^TX)^{-1}X^Ty)^2] - \beta^2 \\
&= \mathbb{E}[(X^TX)^{-1}X^Ty)(X^TX)^{-1}X^Ty)^T] - \beta^T\beta \\
&= \mathbb{E}[(X^TX)^{-1}X^Tyy^TX(X^TX)^{-1}] - \beta^T\beta \\
&= (X^TX)^{-1}X^T\mathbb{E}[yy^T]X(X^TX)^{-1} - \beta^T\beta
\end{aligned}
$$

Here we will calculate $\mathbb{E}[yy^T]$ separately for ease:

$$
\begin{aligned}
\mathbb{E}[yy^T] &= \mathbb{E}[(X\beta + \epsilon)(X\beta + \epsilon)^T] \\
&= \mathbb{E}[(X\beta\beta^TX^T + \epsilon\epsilon^T + X\beta\epsilon^T + \epsilon\beta^TX^T] \\
&= X\beta\beta^TX^T + \mathbb{E}[\epsilon\epsilon^T] + X\beta\mathbb{E}[\epsilon^T] + \mathbb{E}[\epsilon]\beta^TX^T \\
&= X\beta\beta^TX^T + \mathbb{E}[\epsilon\epsilon^T] \\
&= X\beta\beta^TX^T + \sigma^2I
\end{aligned}
$$

Now we put this back into the Variance expression:

$$
\begin{aligned}
Var(\hat{\beta}_{OLS}) &= (X^TX)^{-1}X^T(X\beta\beta^TX^T + \sigma^2I)X(X^TX)^{-1} - \beta^T\beta \\
&= [\beta\beta^TX^T + (X^TX)^{-1}X^T\sigma^2]X(X^TX)^{-1} - \beta^T\beta \\
&= [\beta\beta^T + \sigma^2(X^TX)^{-1}] - \beta^T\beta \\
&= \underline{\sigma^2(X^TX)^{-1}}
\end{aligned}
$$

References

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction.* Springer Verlag, Berlin. Retrieved from https://github.com/CompPhysics/MLErasmus/blob/master/doc/Textbooks/elementsstat.pdf

Hjorth-Jensen, M. (2023). *Applied data analysis and machine learning.* https://compphysics.github.io/MachineLearning/doc/LectureNotes/_build/html/intro.html. ([Online; accessed 30-September-2023])