Data Analysis and Machine Learning
Project1.
Regression analysis and re-sampling methods

Andrei Kukharenka, Anton Fofanov and Anna Gribkovskaya
FYS-STK 4155

October 10, 2018

Abstract

In this project we have performed the Ordinary Least Squares (OSL), Ridge and Lasso regression. The solvers for OSL and Ridge have been developed from scratch and the Lasso regression have been implemented using a Sci-Kit learn library. The estimators were used on the test Franke function ([1]) and on the real data - terrain data for Norway.The k-fold cross validation for all the methods have been also implemented and the bias-variance was studied for all the models. The OLS method with polynomial degree 6 was used for fitting this specific terrain data, the regression parameter $R^2 = 0.98$.

Introduction

Regression analysis is a widely used tool in data science. It incorporates many different techniques and models for estimating relations between variables. In this project we aim to study most widely used estimators - OSL, Ridge and Lasso. OSL regression is one of the most popular models here. It has been implemented in many software packages is various programming languages and is used by many data scientist everyday for performing analysis. In this project we aim to implement it from scratch in order to have a better understanding of its machinery. Rigde regression is also implemented from scratch here.
A regression analysis aims to find the relations between various variables not only to explain or extract a functional dependency, but also to be able to predict outcomes for some unknown values of this variables. Here we need to be very careful with our model, because even if we were able to implement it with the smallest possible error and even hit all the variables provided for estimation of the relations, we can't be sure it will provide same good results when applied to the new set of data. The problem here is a possibility of over-fitting.In order to prevent this we use so-called re-sampling and cross-validation. In this project a k-fold cross validation have beed applied for all three methods.
Structure of the report:
In section 1 we provide a brief theoretical review of the implemented models and re-sampling techniques.
In section 2 we provide results for the test Franke function and for the real terrain data. In section conclusion 3 conclusion and some possibilities for further research are discussed.

1 General description of the linear regression methods

Regression analysis is a powerful tool to analyze the data. Let's start with some definitions needed to understand how it works.

Definition 1. *Variables* $\hat{x} = [x_0, x_1, x_2, \ldots, x_{n-1}]^T$, *are called independent or explanatory variable. Here* n *is number of samples measured.*

Definition 2. *Variables* $\hat{y} = [y_0, y_1, y_2, \ldots, y_{n-1}]^T$, *are called dependent or response variable. Here* n *is number of samples measured.*

The aim of the regression is to estimate the relationship between $\hat{x}$ and $\hat{y}$ variables in order to make predictions and find functional dependences.
Let's assume we measure a set of parameters for each sample. Number of parameters is denoted by p. In this case instead for vector we would get a matrix $\mathbf{X}$ and $\mathbf{Y}$. Matrix $\mathbf{X}$ is called **design matrix** and matrix $\mathbf{Y}$ is **response matrix**. Goal of regression analysis is to determine the functional dependence between $\mathbf{X}$ and $\mathbf{Y}$ or one may say explain one in terms of the other. Here we have no knowledge on the function is available in advance. However one may assume the function

to be linear with respect to some unknown parameters $\beta = (\beta_1, \ldots, \beta_p)^\top$.Such assumption leads to so called **linear regression** and set of β are **regression parameters**. There are many types of linear regressions and in the sections below we are going to describe some of them.
Our goal if to find the functional dependence between variables in form $y = f(x)$, which is not actually possible in the closed form. Instead we are going to fin an approximation for y in a form:

$$y = y(x) \to y(x_i) = \tilde{y}_i + \epsilon_i = \sum_{j=0}^{n-1} \beta_i x_i^j + \epsilon_i. \tag{1}$$

In matrix form it can be written as:

$$\hat{y} = \hat{X}\hat{\beta} + \hat{\epsilon}, \tag{2}$$

here $\hat{\epsilon}$ is error vector. In order to obtain $\hat{\beta}$ values we have to minimize the error. In order to do so we set up the function that provide us the difference between the exact y and approximated values $\tilde{y}$. This function is denoted as $Q(\hat{\beta})$ and has a specific form for each of the regression methods discussed below.

1.1 Ordinary Least Squared and Ridge regression

We present both methods here because OLS can be presented as Ridge regression for specific value of regerssion parameter.

$$Q(\hat{\beta}) = \left\| \left(\hat{y} - \hat{X}\hat{\beta} \right) \right\|_2^2 + \lambda \left\| \hat{\beta} \right\|_2^2. \tag{3}$$

If $\lambda = 0$ we obtain the OLS regression. Namely:

$$Q(\hat{\beta}) = \left\| \left(\hat{y} - \hat{X}\hat{\beta} \right) \right\|_2^2 = \left(\hat{y} - \hat{X}\hat{\beta} \right)^T \left(\hat{y} - \hat{X}\hat{\beta} \right). \tag{4}$$

Minimizing the $Q(\hat{\beta})$ with respect to $\hat{\beta}$ we obtain:

$$\frac{\partial Q(\hat{\beta})}{\partial \hat{\beta}} = 0 = \hat{X}^T \left(\hat{y} - \hat{X}\hat{\beta} \right). \tag{5}$$

And then, if our design matrix $\hat{X}$ in invertible we get

$$\hat{\beta}_{\text{OLS}} = \left(\hat{X}^T \hat{X} \right)^{-1} \hat{X}^T \hat{y}. \tag{6}$$

If $\lambda \neq 0$ we obtain the Ridge regression. Namely:

$$\frac{\partial Q(\hat{\beta})}{\partial \hat{\beta}} = 0 = \hat{X}^T \left(\hat{y} - \hat{X}\hat{\beta} \right) - \lambda \hat{\beta}. \tag{7}$$

And then we obtain:

$$\left(\hat{X}^T \hat{X} + \lambda I \right)^{-1} \hat{\beta} = \hat{X}^T \hat{y}. \tag{8}$$

Then we finally get equation for $\hat{\beta}_{\text{Ridge}}$

$$\hat{\beta}_{\text{Ridge}} = \left(\hat{X}^T \hat{X} + \lambda I \right)^{-1} \hat{X}^T \hat{y}. \tag{9}$$

1.2 Lasso

For Lasso regression $Q(\hat{\beta})$ function is written as:

$$Q(\hat{\beta}) = \left\| \left(\hat{y} - \hat{X}\hat{\beta} \right) \right\|_2^2 + \lambda \left\| \hat{\beta} \right\|_1 . \tag{10}$$

It is very similar to the one we have for Ridge regression, only we now take L_1-norm for $\hat{\beta}_{\text{Ridge}}$, not L_2-norm.

1.3 Resampling methods

As we have already mention the main goal of regression analysis is to find the relations between the dependent and independent variables by estimating set of parameters. Once we have sampled the date and run the regression analysis we have obtained this parameters. However we only can do it once for a single data sample and can't predict how the model we have made will explain new data. The process of collecting data might also be expensive both in computational time and in real life currency, so one might need a recipe to avoid collecting more data and re-use the existing. Another problem here is overfitting. If our model provides us an estimate that corresponds too close to the provided set of data (usually called training data) it might fail to predict the fit for any additional data (usually refereed to as testing data). Generally speaking resampling is a method that aims to improve the prediction accuracy of the model. This is achieved by implementing a very simple idea - splitting the data set into training and testing data. There many such methods for example k-fold cross-validation and bootstrapping.
K-fold cross-validation require dividing the data set into a set of k-subsets (folds). After doing so one of the subsets is considered as a test one and the remaining ones are train data $(k-1)$. This is done for all subsets, so that each of them has to be used as testing data, which means that the process should be repeated k times. Bootstrapping is a bit different - test data set is selected once and used for every selection of the training data. Training data is selected randomly, by taking some values from data set with replacement. Random selection with replacement means that each value may appear multiple times in the one sample.

1.4 Bias-variance tradeoff

As it was mentioned in 1.3 overfitting is a problem we want to avoid. This many be done by reducing the complexity of the model. However, it may lead to another problem - underfitting, which is means that we are ignoring some important features of the train data. The problem of balancing these two is called bias-variance tradeoff or bias-variance dilemma [2]. This is a well known problem in data analysis. Here high bias correspond to underfitting and high variance to overfitting. High bias usually means that the model is not complex enough and correspondingly high variance means that model is too complex.

The higher complexity of the model means that we have better approximation and that lead to lower bias. However any real model has noise and higher approximation is not always good, as we want to avoid the influence of noise. Reducing complexity we obtain lower approximation and higher variance. The best case scenario here is to balance bias and variance. Below is the equation for estimator the error:

$$\begin{aligned} E[(y-\hat{f}(x))^2] = \\ \Big(E[f(x)-\hat{f}(x)]^2\Big) + \Big(E[\hat{f}(x)^2] - E[\hat{f}(x)]^2\Big) + \sigma^2 = \\ \text{Bias}(\hat{f}(x))^2 + \text{Var}(\hat{f}(x)) + \sigma^2, \end{aligned} \tag{11}$$

here σ^2 represents noise and $baise^2$ represent deviation form exact data.

1.5 Regression parameter R^2

Parameter R^2 is used to measure how close the data are to fitted regression line. It is given by:

$$R^2(y,\tilde{y}) = 1 - \frac{\sum_{i=0}^{n-1}(y_i-\tilde{y}_i)^2}{\sum_{i=0}^{n-1}(y_i-\bar{y})^2} \tag{12}$$

with $\bar{y}$ defined as

$$\bar{y} = \frac{1}{n}\sum_{i=0}^{n-1} y_i$$

Another parameter we are using to compare models is Mean Square Error (MSE) and is given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=0}^{n-1}(y_i-\tilde{y}_i)^2, \tag{13}$$

and is simply the averaged squared errors.

1.6 The Franke function

The Franke function is given by following equation:

$$\begin{aligned} f(x,y) = \frac{3}{4}\exp\left(-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right) + \frac{3}{4}\exp\left(-\frac{(9x+1)^2}{49} - \frac{(9y+1)}{10}\right) \\ + \frac{1}{2}\exp\left(-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right) - \frac{1}{5}\exp\left(-(9x-4)^2 - (9y-7)^2\right). \end{aligned}$$

It is presented on Fig. 1.

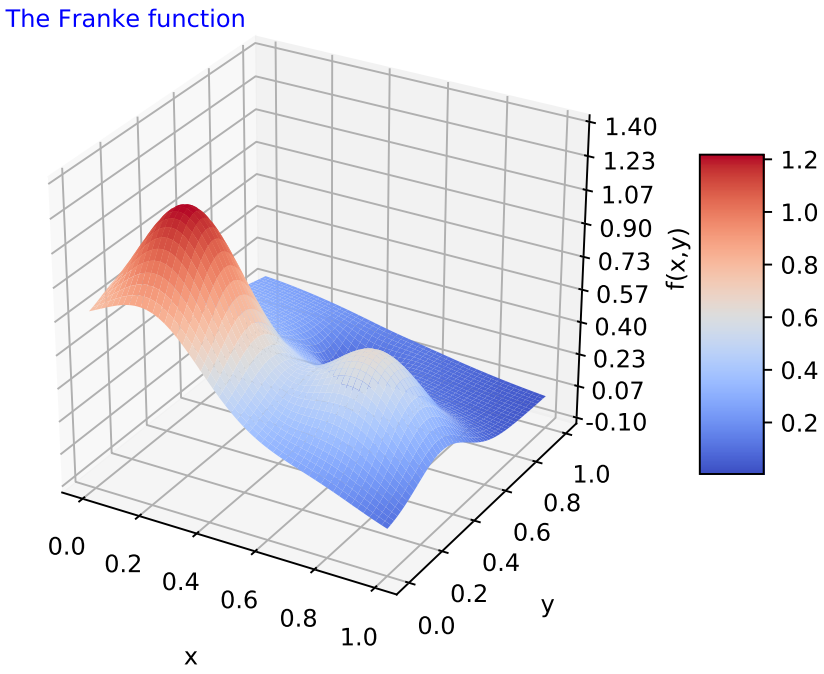

Figure 1: The exact Franke function used in the project to test the fitting.

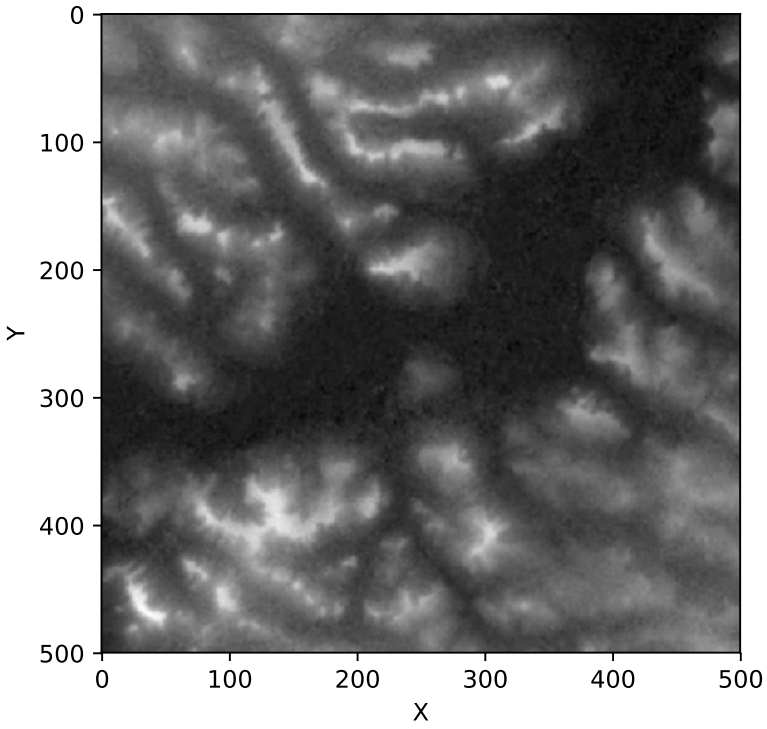

Figure 2: Terrain data in 2D

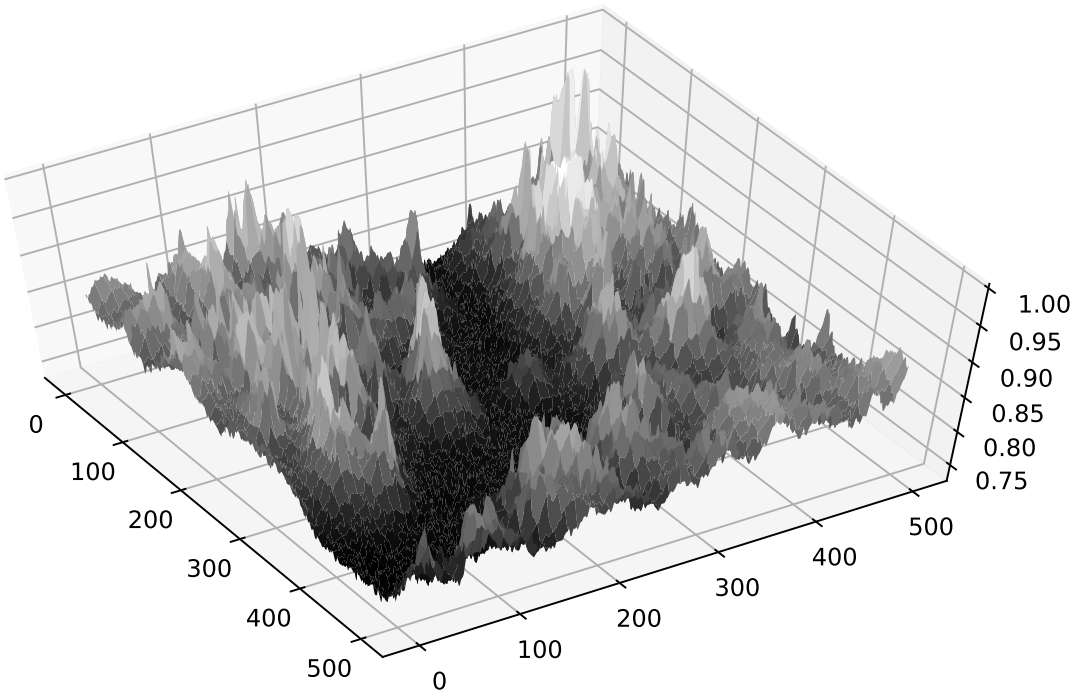

Figure 3: Terrain data raw

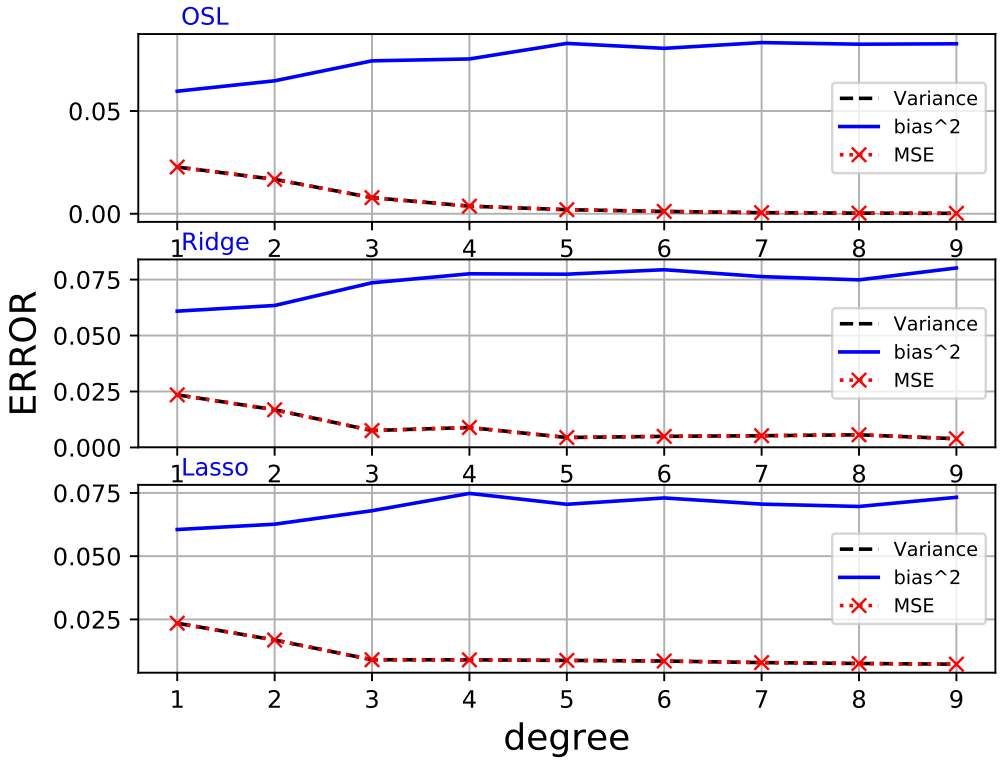

Figure 4: $Biase^2$, *variance* and MSE for Franke function as function the polynomial degree for OLS, Lasso and Ridge. No resampling.

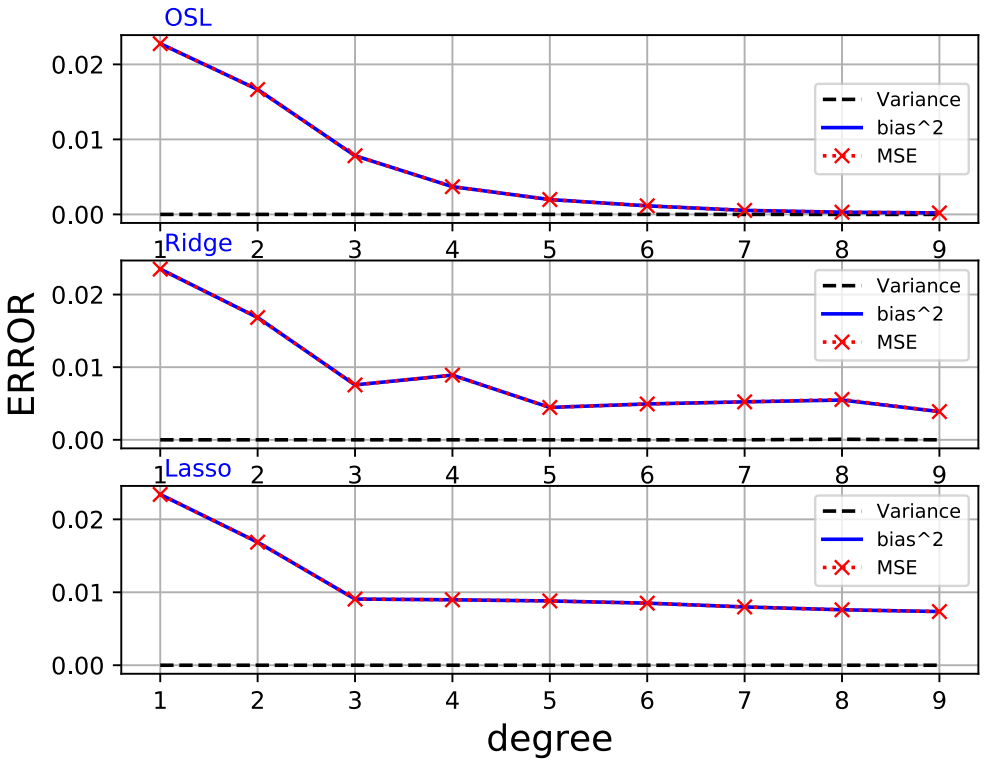

Figure 5: $Biase^2$, *variance* and MSE for Franke function as function the polynomial degree for OLS, Lasso and Ridge. Resampling with five-hold cross-validation.

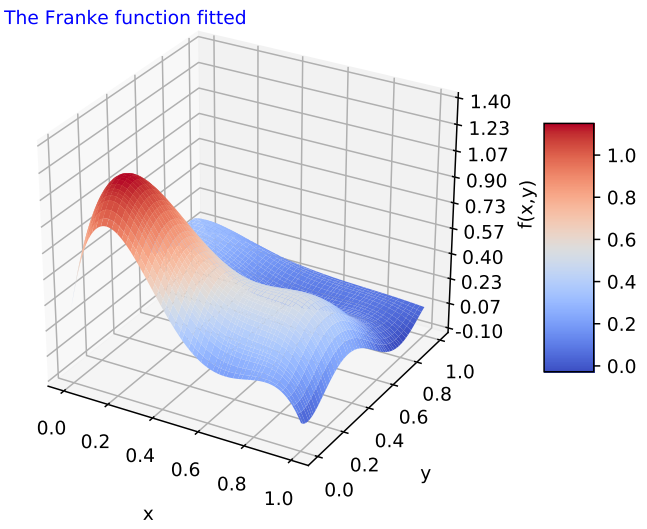

Figure 6: The Franke function fitted by OLS, polynomial fit with degree 5.

1.7 The Terrain Data

Terrain data is taken from the border aria between Guinea and Mali. On Fig. 3 and Fig. 2 are presented plots for this are in 3D and in 2D.

2 Results and discussion

We begin with the fitting of Franke function. On Fig. 6 and Fig. 7 one may see the Franke function fitted using polynomial degree 5 and two different methods, OLS and Ridge regressions.
On Fig. 4 and Fig. 5 we presented data for the error estimation for all three methods with and without resampling. The resampling turns to improve our model a lot. Date for the optimal fit with degree 5 are presented in Table 1. However if we increase the polynomial degree the model become worse because R^2 turns to become much smaller.

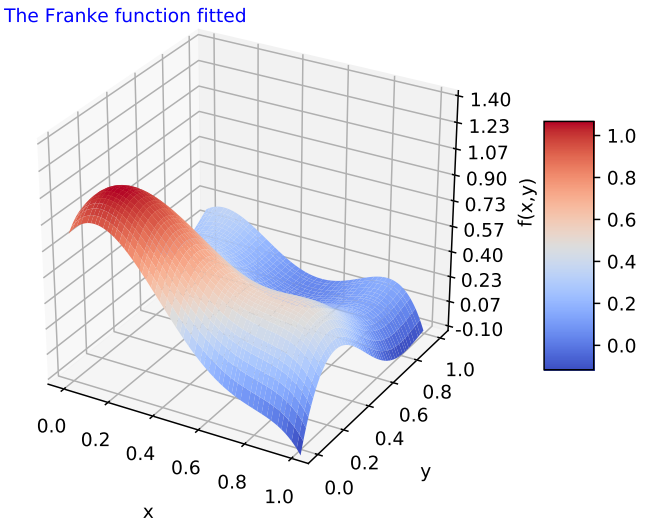

Figure 7: The Franke function fitted by Ridge, polynomial fit with degree 5.

The effect of noise can be seen on Fig. 8 and Fig.9 compared to Fig. 10 and Fig. 11. The presence of noise increase an variance in regression coefficients and effect models in a negative way. However Ridge and Lasso turns to be more sensitive to noise then OLS.
We have also studies the terrain data estimation as a function of the degree of polynomial fit. On Fig. 13 one can see how the R^2 drops for the OLS as we increase the degree of the polynomial fit. The best possible model considering the bais-variance tradeoff turns to be the one with OLS, fitted by 6^{th} order polynomial.

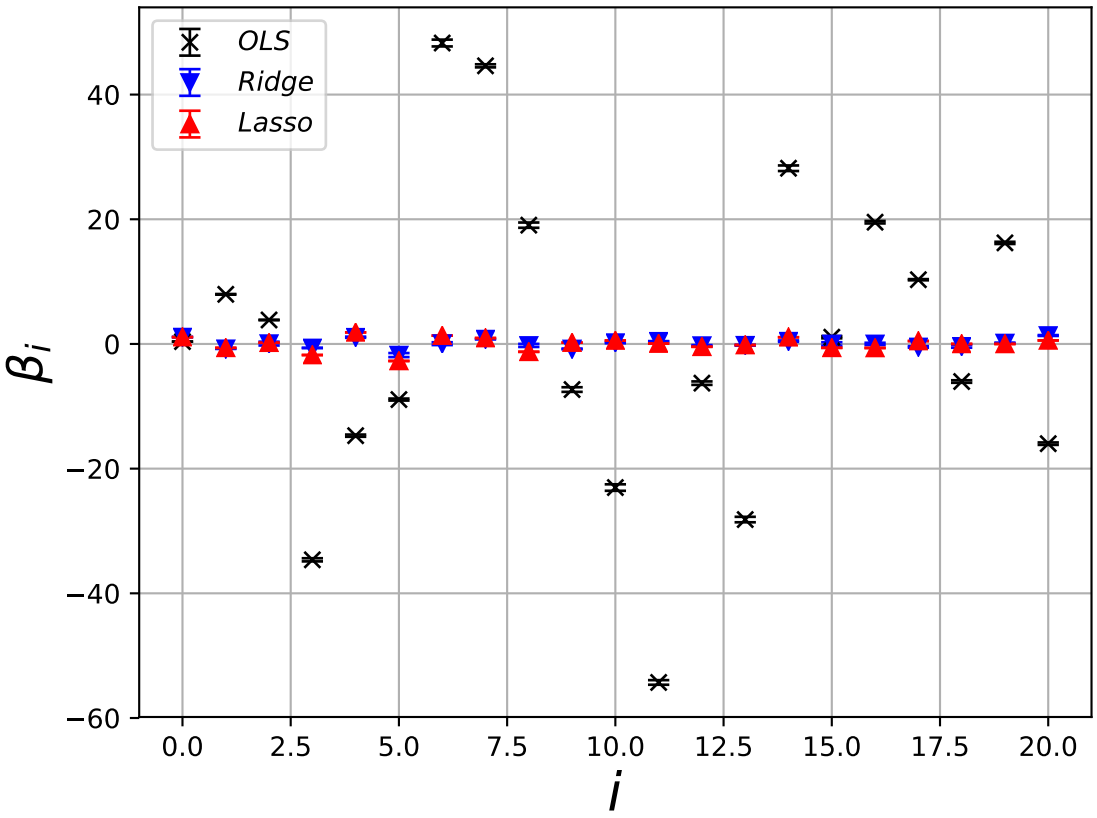

Figure 8: β coefficients for Franke function with polynomial fit of degree 5 for OLS, Lasso, Ridge. Noise added, with five-fold cross-validation. Error bars are in 95% confidence interval.

Method	MSE	R^2	$Bias^2$	Variance
OLS	0.00197089	0.9766113	0.08229618	0.00197094
OLS resampled	0.02266839	0.99999329	0.022668371	4.10271144×10^{-8}
Ridge	0.07820809	0.94582540	0.07820809	0.00448738
Ridge resampled	0.02294272	0.99999307	0.02294268	3.63551460×10^{-8}
Lasso	0.03390619	0.57353504	0.02018597	0.03390694
Lasso resampled	0.03392132	0.99997866	0.033921304	2.0627578×10^{-8}

Table 1: MSE, R^2, $Bias^2$ and variance for OLS, Ridge and Lasso regression. Regression parameter $\lambda = 0.2$ for Ridge and $\lambda = 0.015$ for Lasso. Five-fold cross-validation was used.

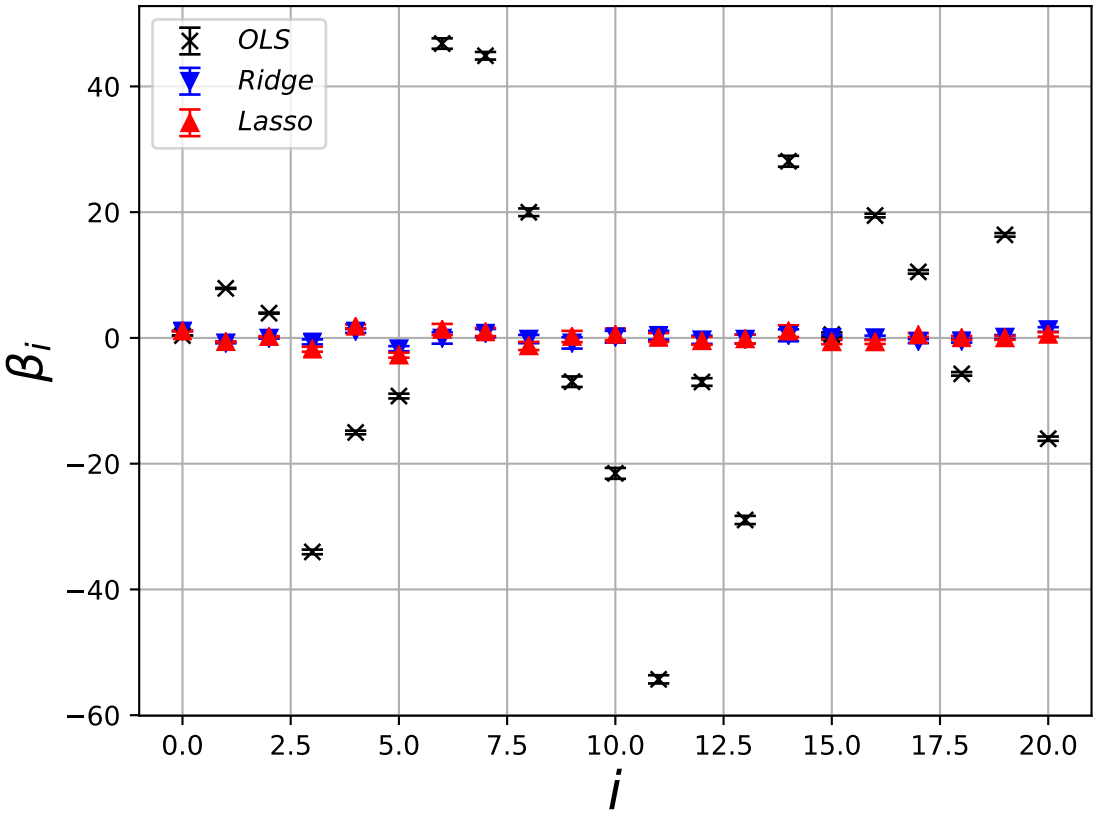

Figure 9: β coefficients for Franke function with polynomial fit of degree 5 for OLS, Lasso, Ridge. Noise added. Without resampling. Error bars are in 95% confidence interval.

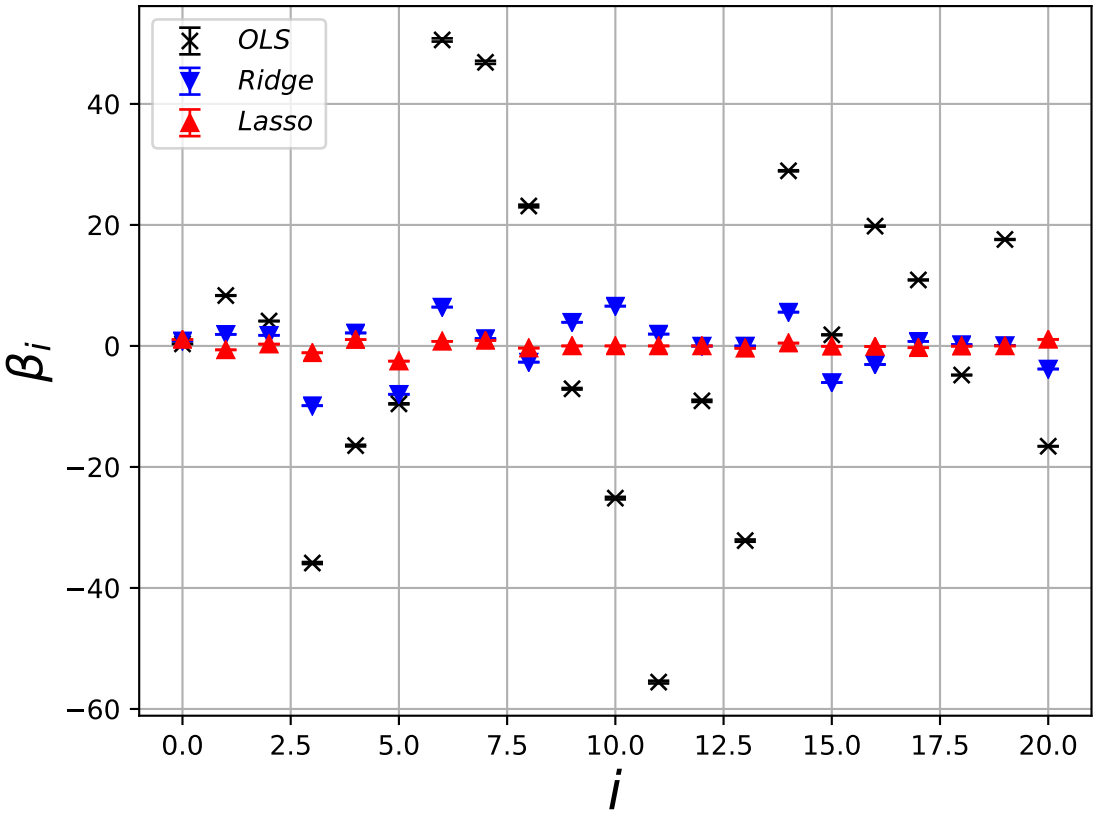

Figure 10: β coefficients for Franke function with polynomial fit of degree 5 for OLS, Lasso, Ridge. No noise, with five-fold cross-validation. Error bars are in 95% confidence interval.

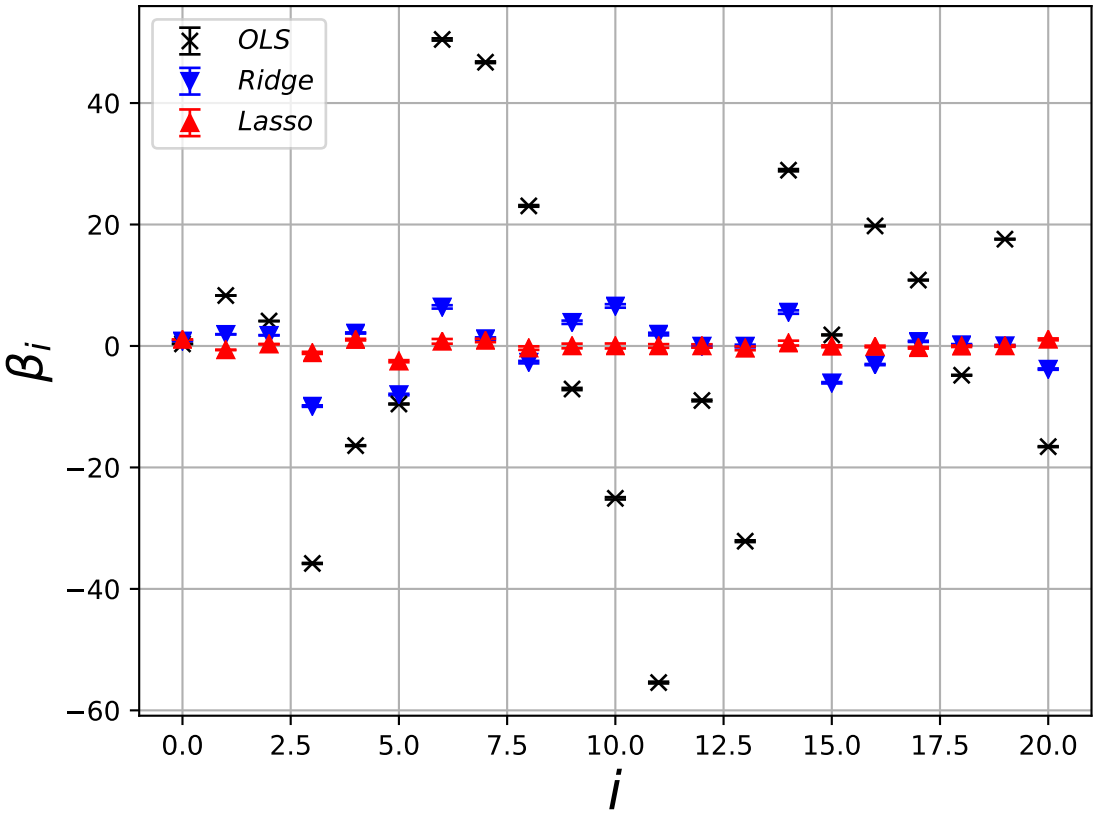

Figure 11: β coefficients for Franke function with polynomial fit of degree 5 for OLS, Lasso, Ridge. No noise. Without resampling. Error bars are in 95% confidence interval.

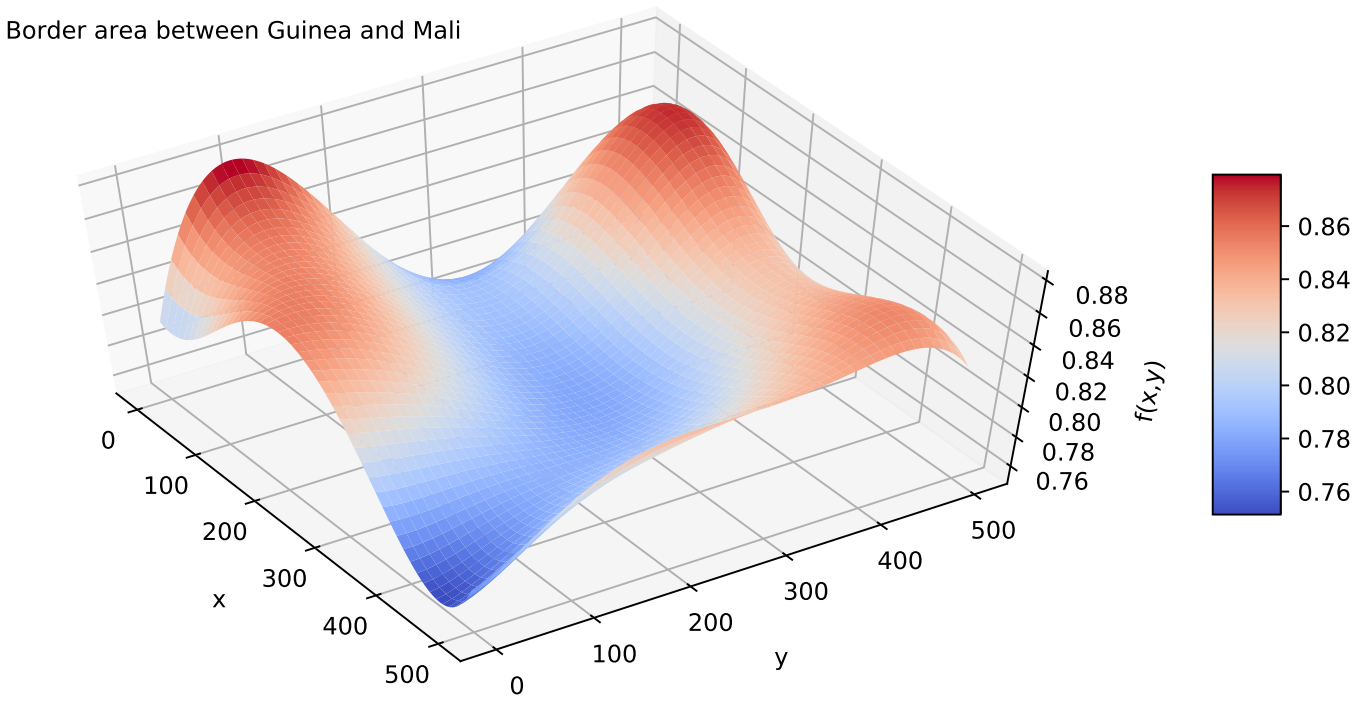

Figure 12: Terrain data fit

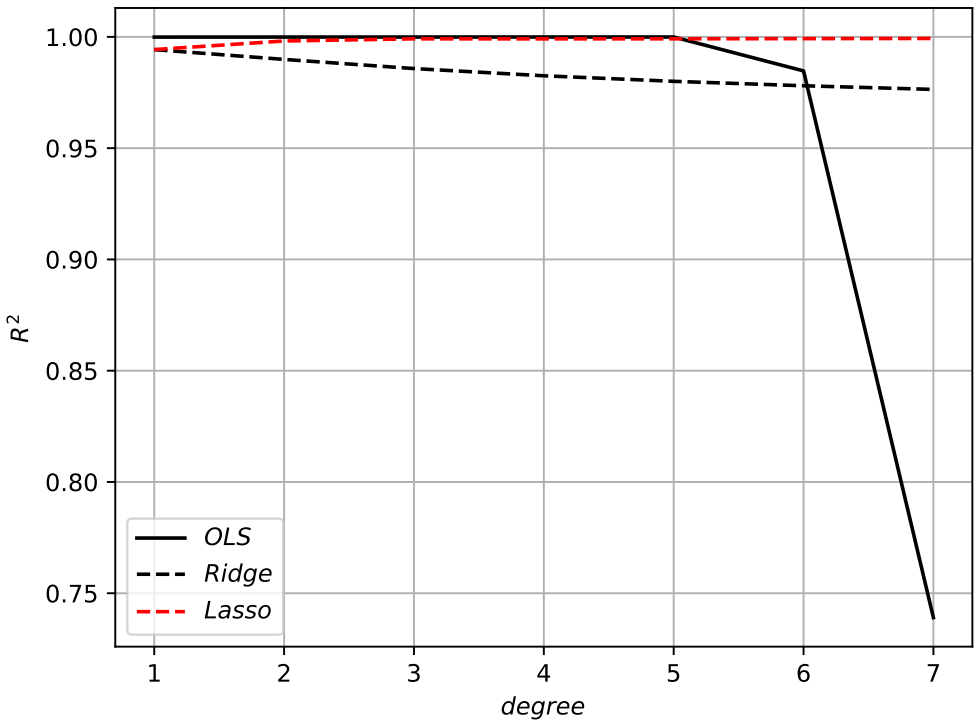

Figure 13: Terrain data R^2

3 Conclusion, discussion and possibilities for improvement

In this project we have studied tree different regression methods, namely OLS, Ridge and Lasso. The method were tested on Franke function and the real terrain data. K-fold cross-validation have been done for all cases and turns to be a very good technique for the improvement of the fit.
We do not spend much time on Ridge and Lasso methods as they depend on parameter λ. However, when running the program we have to be very careful with it, because it influenced performance a lot. One of the possible further studies though might be to implement the Lasso method from scratch and study it more carefully.
As for the terrain data we have studied the result presented on Fig. 12. Also the result for $R^2 = 0.98$ is quite good for this model.
As for the future work, we would like to spend a bit time on organizing the code and getting a bit more result from it. Also it would be interesting to investigate how different finds of terrains are fitted, not only the one we have chosen for this project.

References

[1] Richard Franke. *A critical comparison of some methods for interpolation of scattered data.* Technical report, Monterey, California: Naval Postgraduate School., 1979.

[2] Pankaj Mehta et al. *A high-bias, low-variance introduction to machine learning for physicists.* In: arXiv preprint arXiv:1803.08823 (2018). Addison-Wesley, Reading, Massachusetts, 1993.