

Project 1

FYS-STK4155

Håvard Skåli, Erik Røset & Oskar Idland
University of Oslo, Department of Physics
(Dated: September 30, 2024)

We have used various regression techniques and resampling methods within the context of machine learning, and tested these on the Franke function, a synthetic benchmark used in numerical analysis, as well as on output data from a cosmological N-body simulation performed with the public GASOLINE2 code. The main objective was to analyze and compare the performance of Ordinary Least Squares (OLS), Ridge, and Lasso regression in fitting synthetic and real-world data, focusing on the bias-variance tradeoff and model generalizability. To assess model performance we employed resampling methods such as bootstrap and k -fold cross-validation, examining how they help to evaluate model accuracy under different training and test data conditions. Our study provided insights into how model complexity and regularization parameters affect the bias-variance tradeoff and prediction error, enabling us to critically evaluate each regression technique's performance using statistical and resampling approaches.

rewrite when results are gathered and discussion/conclusion is finished

<https://github.com/Oskar-Idland/FYS-STK4155-Projects>

I. INTRODUCTION

In this work, we explore regression analysis and resampling methods in the context of machine learning, focusing on both theoretical and practical aspects. The primary goal is to develop a solid understanding of various regression techniques, including Ordinary Least Squares (OLS), Ridge, and Lasso regression, and their application to both synthetic and real-world data. We will also investigate how resampling methods such as bootstrap and cross-validation can help assess model performance.

We begin by applying the abovementioned methods to the Franke function, a widely used test function in numerical analysis, and extend the analysis to cosmological N-body simulation data made with the public version of the GASOLINE2 Smoothed Particle Hydrodynamics (SPH) code [1]. Our approach involves fitting polynomial regression models of varying complexity to the Franke function as well as the simulation data, and studying the bias-variance tradeoff, an essential concept in machine learning, in both cases. This allows us to evaluate the impact of model complexity, noise, and the size of training data on the accuracy and generalizability of the models. The overarching aim is to gain insights into how different regression methods handle overfitting, model complexity, and data variability, and to develop a framework for critically evaluating model performance using statistical and resampling techniques. rewrite?

In section II we present relevant background theory, including central concepts such as model bias, model variance and the bias-variance tradeoff, as well as the Franke function. The most important expressions introduced here are derived in appendix A. Our methodology is explained in section III, specifically regression analysis, where we focus on OLS, Ridge and Lasso regression. We also present the advantages and disadvan-

tages of two crucial resampling methods; bootstrapping and cross-validation, both of which will be used in this work. In this section we also present our dataset, specify how we implement the methods and give an overview of our code structure. The results of our analyses are presented in section IV, and in section V we discuss our findings in light of what we would expect from the regression variants and resampling methods implemented edit after discussion is written. Lastly, in section VI we summarize and conclude the main findings of our work mention reflection?.

II. THEORY

A. Regression Analysis

Regression analysis is a fundamental statistical technique used to model the relationship between one or more independent variables (also known as predictors or features) and a dependent variable (or target). The goal of regression analysis is to find the mathematical relationship that best explains the variation in the dependent variable based on the values of the independent variables.

At its core, regression analysis seeks to find a mathematical function that relates the independent variables to the dependent variable. In its most basic form, the relationship between a dependent variable \mathbf{y} and an independent variable \mathbf{x} is modeled as

$$\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\epsilon}, \quad (1)$$

where $f(\mathbf{x})$ is the function we are trying to estimate, which represents the relationship between \mathbf{x} and \mathbf{y} , and $\boldsymbol{\epsilon}$ is the error term, representing the part of the variation in \mathbf{y} that is not explained by the model (due to noise or

other unobserved factors). move expression to theory?

explain cost function, MSE, score function

include analytic part of week 37

1. Ordinary Least Squares

The Ordinary Least Squares (OLS) method is one of the most fundamental and widely used techniques in regression analysis. Its objective is to find the best-fitting line or curve for a given set of data by minimizing the sum of the squared differences between the observed values and the predicted values. These squared differences are called residuals, and minimizing them ensures the best possible fit of the model to the data. continue, introduce expressions (week 37), cite?

mention the cost function, and refer to MSE?

2. Ridge

While OLS provides a foundational method for regression, it can lead to problems when the data has high multicollinearity or when there are more features than data points, leading to overfitting. OLS attempts to minimize the sum of squared errors, but it does not impose any restrictions on the model complexity. This often results in high variance when the model learns to fit noise in the training data. To address this, Ridge regression introduces a regularization term to the OLS cost function, penalizing large coefficients and preventing the model from becoming overly complex. By introducing the hyperparameter λ , Ridge regression balances the trade-off between bias and variance, allowing us to control model complexity and fit the data more effectively. double check, introduce expressions (week 37), cite?

3. Lasso

Like Ridge regression, Lasso regression (Least Absolute Shrinkage and Selection Operator) is a regularization technique designed to improve the generalizability of the model by introducing a penalty term to the cost function. While Ridge regression uses the L2 norm (squared magnitude of coefficients) for regularization, Lasso regression employs the L1 norm, which leads to a different form of regularization. This difference has important implications, particularly for feature selection and sparse models. double check, introduce expressions, cite?

B. Properties of Predictive Models

maybe change title of section

explain what y , \tilde{y} etc. represent. see week 38

1. Predicted Values

In the context of predictive modeling, \tilde{y} represents the model's predicted values of the target variable y . These predictions are made based on the features and parameters learned from the training data. For new or unseen data, the true target values y are often unknown, but the model generates an estimate \tilde{y} that approximates these values.

The performance of the model's predictions can be analyzed by decomposing the error into several components. The error of \tilde{y} stems from three main sources: noise variance σ^2 , model bias and model variance. The latter two are expressed as

$$\text{Bias}[\tilde{y}] = \mathbb{E}[(\mathbf{y} - \mathbb{E}[\tilde{y}])^2], \quad (2)$$

$$\text{Var}[\tilde{y}] = \mathbb{E}[(\tilde{y} - \mathbb{E}[\tilde{y}])^2]. \quad (3)$$

Each of these represents a different aspect of the total error that affects the model's performance.

2. Model Bias

From the expression (2) we see that the model bias measures how far the average model prediction $\mathbb{E}[\tilde{y}]$ is from the true target value y . More specifically, it refers to the systematic error introduced by the model's inability to capture the true underlying relationship between the independent and dependent variables. This error arises when the model makes incorrect assumptions about the data or oversimplifies the relationship. For example, a linear model trying to fit highly non-linear data will result in high bias because the model cannot flexibly represent the complexity of the data.

3. Model Variance

We see from the expression (3) for the model variance that it represents the variability in the model's predictions across different training sets. A high model variance indicates that the model is overly sensitive to the specific training data, leading to overfitting, where the model performs well on the training set but poorly on new, unseen data such as the test set. This happens because the model has "memorized" the noise in the training data rather than capturing the true underlying pattern. High variance typically arises in overly complex models that are capable of capturing minute details, which may not generalize well to new data.

4. Bias-Variance Tradeoff

Measures of a predictive model's total error and how well it is likely to predict future samples are commonly expressed in terms of the mean-squared error (MSE) and the score function, respectively. These are given by

$$\text{MSE}(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \tilde{y}_i)^2, \quad (4)$$

$$R^2(\mathbf{y}, \tilde{\mathbf{y}}) = 1 - \frac{\sum_{i=0}^{n-1} (y_i - \tilde{y}_i)^2}{\sum_{i=0}^{n-1} (y_i - \bar{y})^2}, \quad (5)$$

where \bar{y} is the mean value of \mathbf{y} and n is the number of data points. In appendix A [maybe move up here?] we show that the MSE alternatively can be expressed in terms of the model bias, model variance and noise variance as

$$\mathbb{E}[(\mathbf{y} - \tilde{\mathbf{y}})^2] = \text{Bias}[\tilde{y}] + \text{Var}[\tilde{y}] + \sigma^2. \quad (6)$$

Since the noise variance adds an irreducible contribution to the total model error, an ideal model strikes a balance between bias and variance, where it is flexible enough to capture the underlying patterns in the data but not so flexible that it fits the noise. This balance is referred to as the bias-variance tradeoff. It is a fundamental concept in machine learning, specifically when building predictive models, and will be studied in great detail throughout this work.

C. Resampling Methods

Resampling methods are statistical techniques used to generate additional data samples from the available data. These methods are particularly useful in machine learning and data analysis when the dataset is limited, and we want to better assess the performance of a model. The primary goal of resampling is to estimate the accuracy of a model by splitting the data into different subsets or generating new samples of the data, repeatedly fitting the model, and evaluating its performance on different sets of data. In this work we implement two commonly used resampling techniques: bootstrap and cross-validation.

1. Bootstrap

The Bootstrap method is a powerful resampling technique used to estimate the uncertainty and variability of a model by repeatedly drawing random samples, with replacement, from the original dataset. This resampling creates multiple "bootstrapped" datasets, each the same size as the original, but with some samples appearing multiple times and others potentially omitted.

The process of bootstrap resampling approximates the underlying distribution of a statistic—whether it's model

performance, a parameter estimate, or prediction error—without requiring strong parametric assumptions about the data. This makes it particularly useful when the theoretical distribution of a statistic is unknown or difficult to calculate. Bootstrap methods are frequently used to estimate confidence intervals, standard errors, and model variance in both regression and classification tasks. [week 38, lecture notes, cite?]

2. Cross-Validation

Cross-validation is another resampling technique that involves partitioning the dataset into several distinct subsets (or "folds"), and then systematically training the model on one subset while testing it on another. A common form is k -fold cross-validation, where the dataset is divided into k equally-sized folds. The model is trained on $k - 1$ folds and tested on the remaining fold. This process is repeated k times, with each fold serving as the test set exactly once.

The steps used in k -fold cross-validation are

1. Split the dataset into k equal-sized folds (typically $k = 5$ or $k = 10$).
2. Train the model on $k - 1$ folds and evaluate it on the remaining fold.
3. Repeat the process k times, so that each fold is used as a test set exactly once.
4. Calculate the average performance across all folds.

[continue, proof read, cite lecture notes?]

D. The Franke Function

The Franke function is a widely used two-dimensional synthetic function in numerical analysis and computational mathematics, particularly in the fields of interpolation, regression analysis, and surface fitting. It is defined over the unit square $[0, 1] \times [0, 1]$ and expressed as

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

which leads to the surface shown in fig. 1. We will use the function multiple times throughout this work, as it produces a surface with both smooth and non-smooth

regions, making it an excellent benchmark for testing regression and resampling algorithms. It simulates real-world data that may contain both complex variations and noise, mimicking scenarios that we will encounter as we move on to implementing the algorithms on actual simulation data.

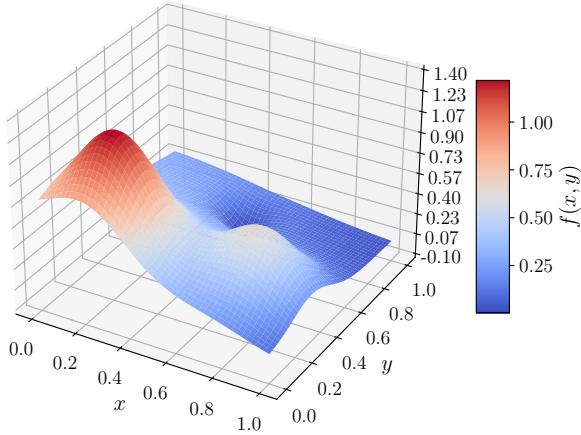


FIG. 1: Visualization of the two-dimensional Franke function expressed in (7).

E. Cosmological Simulation Data

After implementing regression analysis and resampling methods using the Franke function we repeat the process using data from an N-body simulation of dark matter structure formation made with the public GASOLINE2 SPH code [1], as mentioned in section I. The initial conditions for the simulation were generated with MUSIC (MULTi-Scale Initial Conditions) [2], and standard Planck cosmology parameters $\Omega_m = 0.3077$, $\Omega_\Lambda = 0.6923$ and $H_0 = 67.81 \text{ km/s/Mpc}$ were used. The simulation box has length, width and height $L = 20 \text{ Mpc}$, corresponding to a $500 \times 500 \times 500$ grid, and contains 64^3 dark matter particles.

The output data includes 128 “snapshots” of the box through time, where the time parameter in the simulation is the scale factor, defined as

$$a = \frac{1}{1+z}. \quad (8)$$

Here z is the cosmological redshift, which is 0 today and thus in the 128th snapshot. We focus on the dark matter density ρ , and study a snapshot at redshift $z \approx 12.88$ in addition to the snapshot at $z = 0$, because the distribution of ρ is smoother at larger z . To get a function of two variables we average over one of the three axes so that we get a 2D grid, and because the density is so large and

vastly different at different points in the box we take $\ln \rho$ to be our dependent variable instead of ρ . The resulting data is shown as intensity plots in figure 2.

Because the Universe expands as time goes on the box does as well, and we therefore scale the coordinates of the box by multiplying with a . Similarly, space between the dark matter particles also increases with time, which is why the density is scaled by multiplying with a^{-3} . GASOLINE2 uses the rest of the scaling factors by default.
move to method? explain better?

III. METHODS & IMPLEMENTATION

write in past tense from here on!

A. Regression Analysis

explain implementation on both Franke and data

1. Scaling the Data

2. Model Complexity

task a, OLS

3. Hyperparameter

task b and c, Ridge and Lasso

B. Bias-Variance Tradeoff

explain how we recreate 2.11

1. Bootstrap With OLS

model complexity, number of data points

2. Cross-Validation

go into detail how we implement

compare with bootstrap include Ridge and Lasso

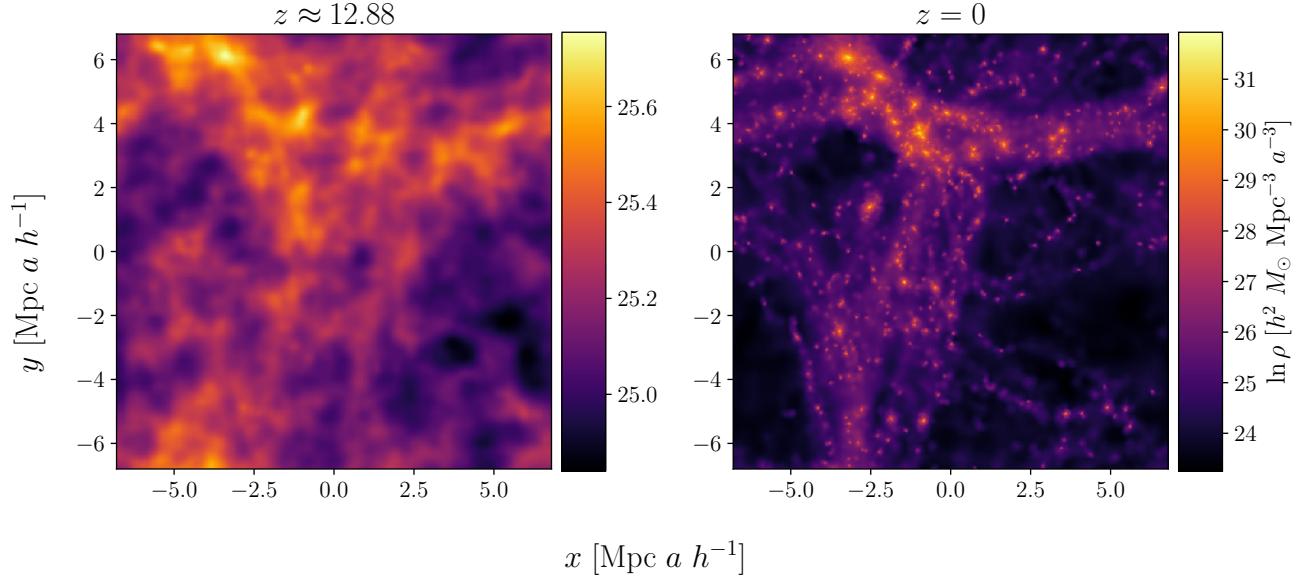


FIG. 2: Dark matter density in the simulation box averaged over one of the three axes at redshifts $z \approx 12.88$ (left) and $z = 0$ (right). better caption?

C. The Program

1. Code Structure

2. Tools

raw and smoothed surfaces, explain reason/method

IV. RESULTS

V. DISCUSSION

VI. CONCLUSION

VII. REFLECTION

ACKNOWLEDGEMENTS

REFERENCES

- [1] T. R. Q. James W. Wadsley, Benjamin W. Keller, “Gasoline2: A Modern SPH Code,” <https://arxiv.org/abs/1707.03824> (Accessed: September 2024).
- [2] O. Hahn, “MUSIC,” <https://www-n.oca.eu/ohahn/MUSIC/> (Accessed: September 2024).

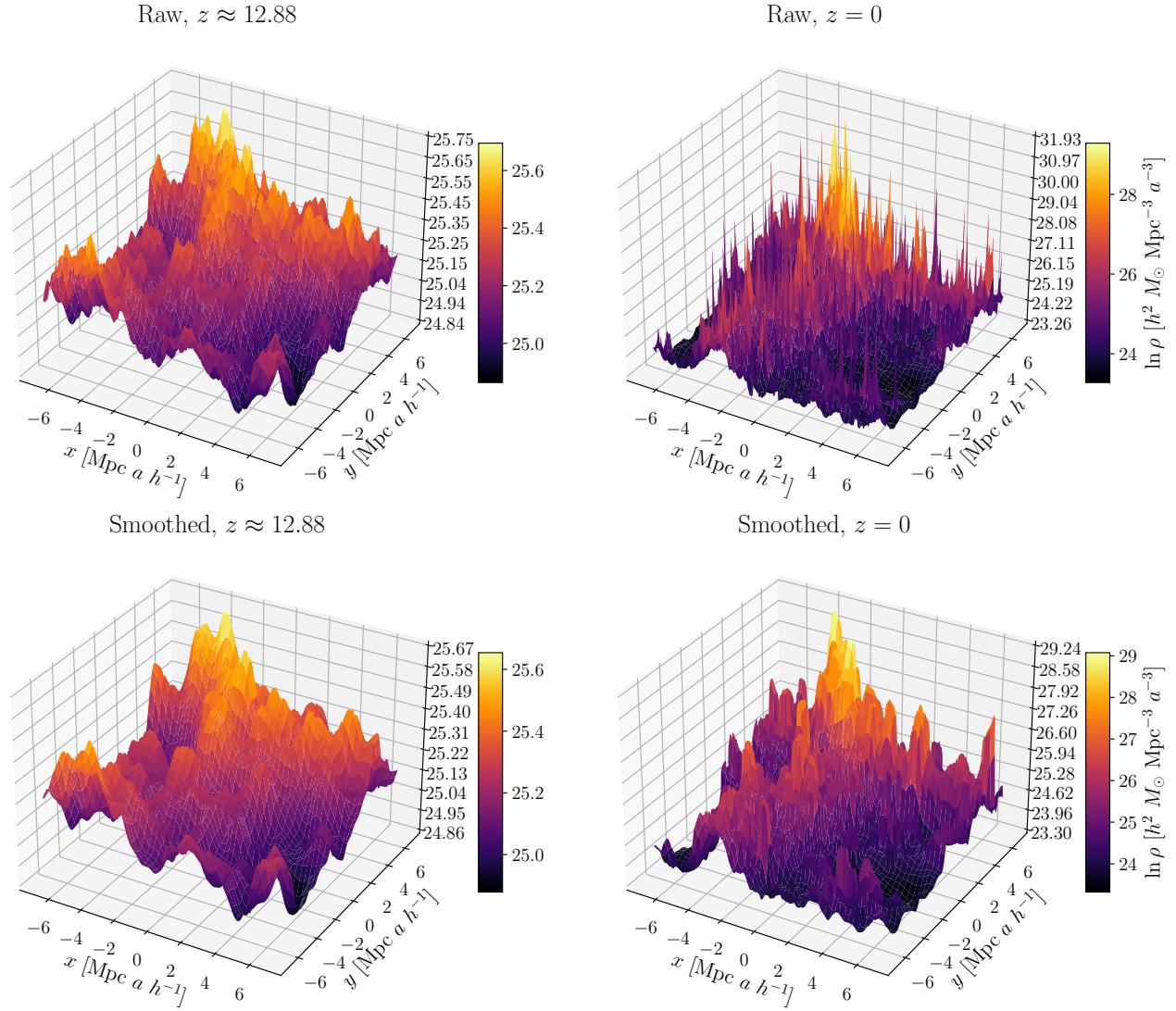


FIG. 3: caption

Appendix A: Derivations**1. Expectation Value and Variance of β_{OLS}**

rewrite to fit report

Since the error in \mathbf{y} is normal distributed as $\boldsymbol{\varepsilon} \sim N(0, \sigma^2)$ we know that the expectation value and variance of the i 'th element of $\boldsymbol{\varepsilon}$ is $\mathbb{E}(\varepsilon_i) = 0$ and $\text{Var}(\varepsilon_i) = \sigma^2$. Thus, since we approximate $f(\mathbf{x})$ with $\tilde{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}$ the expectation value of \mathbf{y} becomes

$$\mathbb{E}(\mathbf{y}) = \mathbb{E}(\mathbf{X}\boldsymbol{\beta}) + \mathbb{E}(\boldsymbol{\varepsilon}) = \mathbb{E}(\mathbf{X}\boldsymbol{\beta}), \quad (\text{A1})$$

which gives us the expectation value of \mathbf{y} for a given element i :

$$\mathbb{E}(y_i) = \mathbb{E} \left(\sum_j X_{ij} \beta_j \right) = \sum_j X_{ij} \beta_j = \mathbf{X}_{i,*} \boldsymbol{\beta}. \quad (\text{A2})$$

Here we have used that the sum $\sum_j X_{ij} \beta_j$ is known to be the value of \tilde{y}_i , hence its expectation value is itself. Moreover, we can similarly find the variance of \mathbf{y} for a given element i by using that the aforementioned sum is known to be \tilde{y}_i

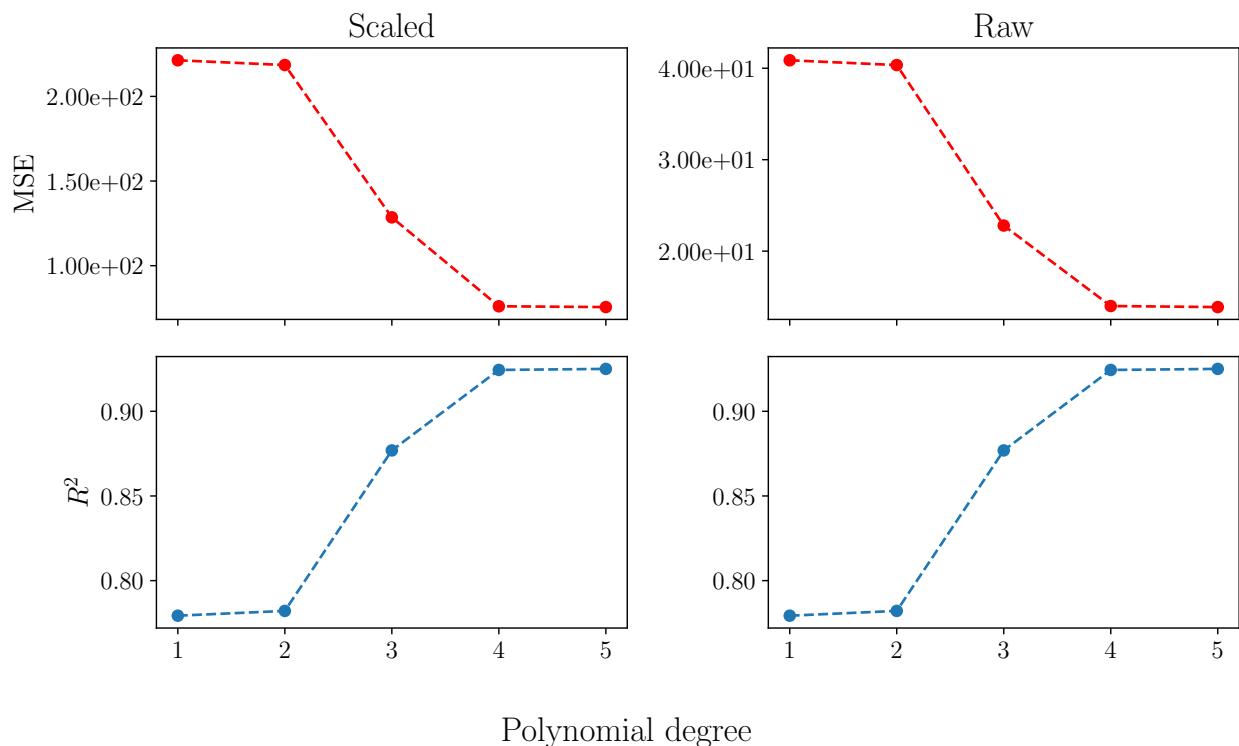


FIG. 4: [caption] remove figure title?

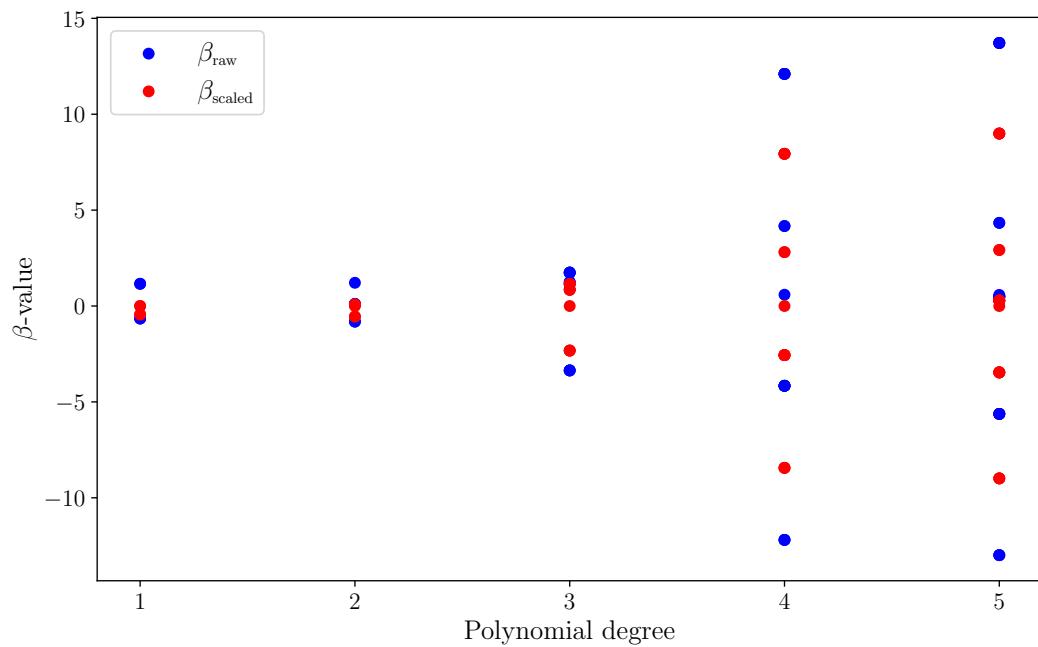


FIG. 5: [caption] remove figure title?

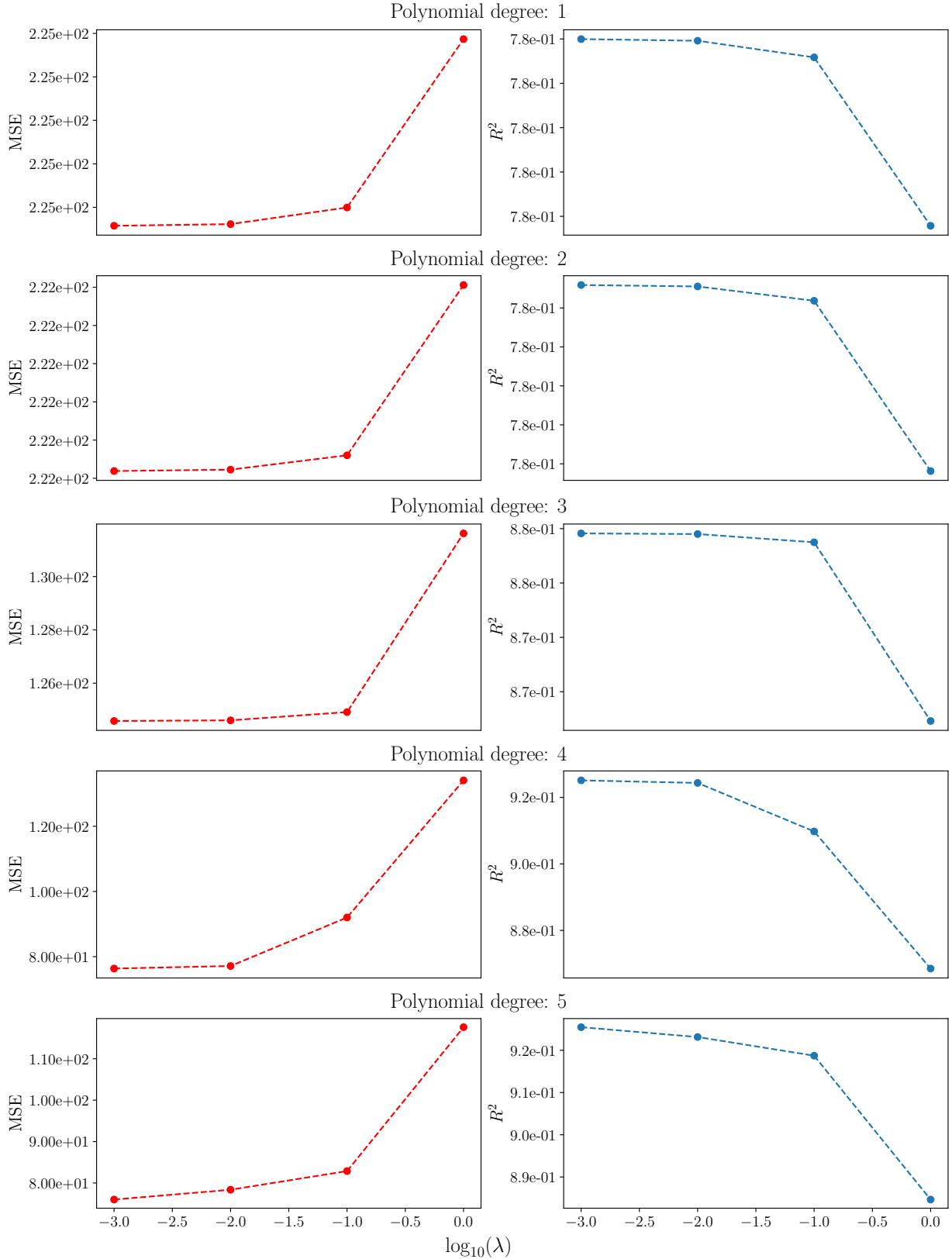


FIG. 6: [caption]

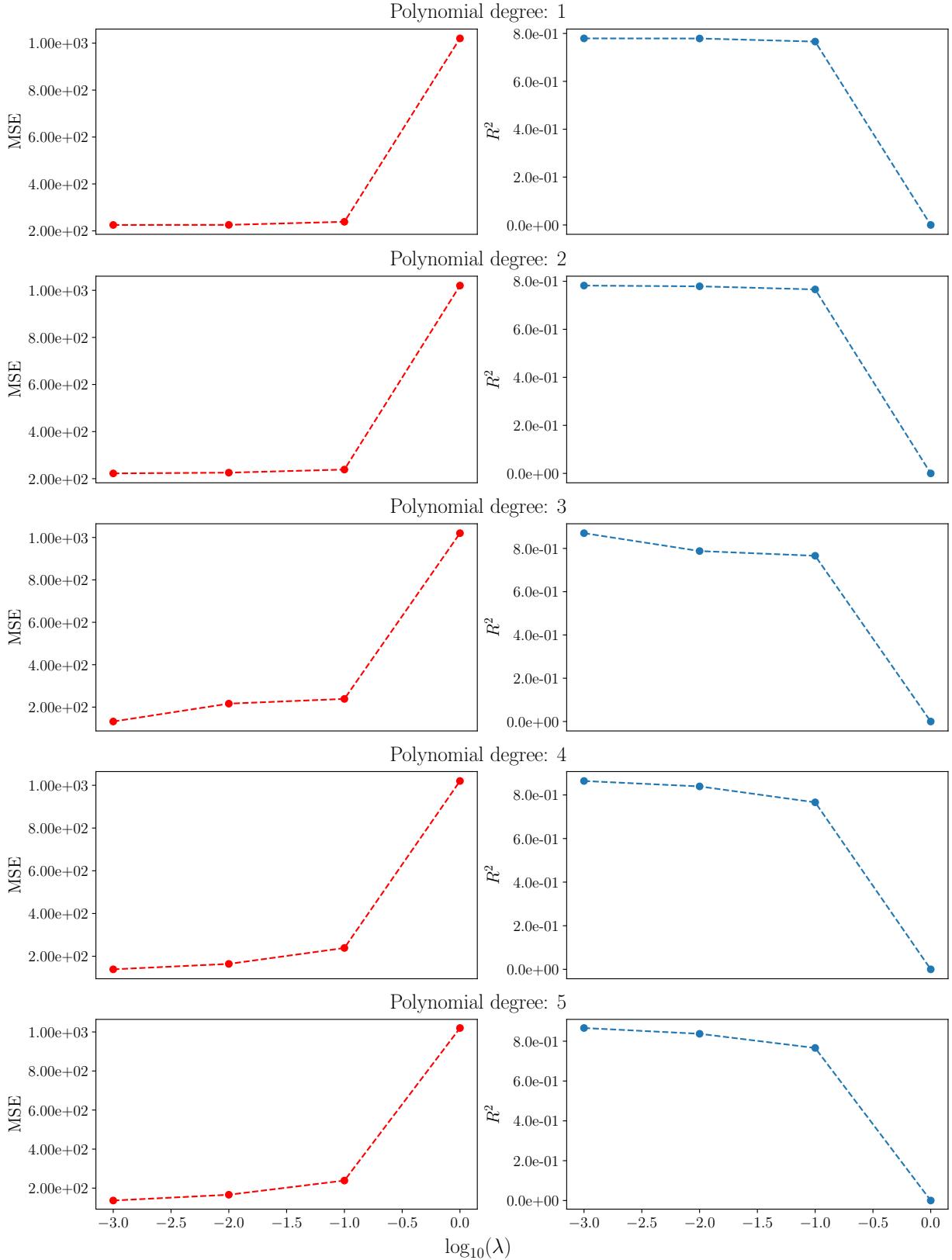


FIG. 7: [caption]

for all i , i.e. $\text{Var}(\mathbf{X}_{i,*}\boldsymbol{\beta}) = 0$. Thus, we have

$$\text{Var}(y_i) = \text{Var}(\mathbf{X}_{i,*}\boldsymbol{\beta}) + \text{Var}(\varepsilon_i) = \sigma^2, \quad (\text{A3})$$

and consequently

$$y_i \sim N(\mathbf{X}_{i,*}\boldsymbol{\beta}, \sigma^2). \quad (\text{A4})$$

Now, using that the optimal parameters in OLS are given by $\boldsymbol{\beta}_{\text{OLS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$, their expectation values become

$$\begin{aligned} \mathbb{E}(\boldsymbol{\beta}_{\text{OLS}}) &= \mathbb{E} \left\{ (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \right\} \\ &= \mathbb{E} \left\{ (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right\} \mathbb{E}(\mathbf{y}) \\ &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} \boldsymbol{\beta} \\ &= \boldsymbol{\beta}. \end{aligned} \quad (\text{A5})$$

Furthermore, if x and y are two independent variables, the variance of their product is given by

$$\begin{aligned} \text{Var}(xy) &= \mathbb{E}(x^2 y^2) - (\mathbb{E}(xy))^2, \\ &= \mathbb{E}(x^2) \mathbb{E}(y^2) - (\mathbb{E}(x))^2 (\mathbb{E}(y))^2, \\ &= [\mathbb{E}(x^2) - (\mathbb{E}(x))^2 + (\mathbb{E}(x))^2] [\mathbb{E}(y^2) - (\mathbb{E}(y))^2 + (\mathbb{E}(y))^2] - \mathbb{E}(x^2) \mathbb{E}(y^2), \\ &= [\text{Var}(x) + (\mathbb{E}(x))^2] [\text{Var}(y) + (\mathbb{E}(y))^2] - \mathbb{E}(x^2) \mathbb{E}(y^2), \\ &= \text{Var}(x) \text{Var}(y) + \text{Var}(x)(\mathbb{E}(y))^2 + \text{Var}(y)(\mathbb{E}(x))^2 + \mathbb{E}(x^2) \mathbb{E}(y^2) - \mathbb{E}(x^2) \mathbb{E}(y^2), \\ &= \text{Var}(x) \text{Var}(y) + \text{Var}(x)(\mathbb{E}(y))^2 + \text{Var}(y)(\mathbb{E}(x))^2, \end{aligned}$$

so if we now set $x = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ and $y = \mathbf{y}$ we find that the variance of $\boldsymbol{\beta}_{\text{OLS}}$ is

$$\begin{aligned} \text{Var}(\boldsymbol{\beta}_{\text{OLS}}) &= \underbrace{\text{Var} \left[(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right]}_0 \text{Var}(\mathbf{y}) + \underbrace{\text{Var} \left[(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right]}_0 (\mathbb{E}(\mathbf{y}))^2 + \text{Var}(\mathbf{y}) \left(\mathbb{E} \left[(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right] \right)^2, \\ &= \sigma^2 \left[(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right]^2, \\ &= \sigma^2 \left[(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right] \left[(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right]^T, \\ &= \sigma^2 \left[(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right] \left[\mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \right], \\ &= \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}. \end{aligned} \quad (\text{A6})$$

Here we have used that the transpose of $(\mathbf{X}^T \mathbf{X})^{-1}$ is itself since it is square and symmetric.

2. Alternative Expression for MSE

rewrite to fit report

Substituting \mathbf{y} with $f(\mathbf{x}) + \epsilon$, and adding and subtracting $\mathbb{E}[\tilde{\mathbf{y}}]$, we find that

$$\begin{aligned}
\mathbb{E}[(\mathbf{y} - \tilde{\mathbf{y}})^2] &= \mathbb{E}\left[\underbrace{(f(\mathbf{x}) + \epsilon - \tilde{\mathbf{y}})}_{\mathbf{f}}^2\right], \\
&= \mathbb{E}[(\mathbf{f} + \epsilon - \tilde{\mathbf{y}} + \mathbb{E}[\tilde{\mathbf{y}}] - \mathbb{E}[\tilde{\mathbf{y}}])^2], \\
&= \mathbb{E}\left[\mathbf{f}^2 + \mathbf{f}\epsilon - \mathbf{f}\tilde{\mathbf{y}} + \mathbf{f}\mathbb{E}[\tilde{\mathbf{y}}] - \mathbf{f}\mathbb{E}[\tilde{\mathbf{y}}]\right. \\
&\quad + \epsilon\mathbf{f} + \epsilon^2 - \epsilon\tilde{\mathbf{y}} + \epsilon\mathbb{E}[\tilde{\mathbf{y}}] - \epsilon\mathbb{E}[\tilde{\mathbf{y}}] \\
&\quad - \tilde{\mathbf{y}}\mathbf{f} - \tilde{\mathbf{y}}\epsilon + \tilde{\mathbf{y}}^2 - \tilde{\mathbf{y}}\mathbb{E}[\tilde{\mathbf{y}}] + \tilde{\mathbf{y}}\mathbb{E}[\tilde{\mathbf{y}}] \\
&\quad + \mathbb{E}[\tilde{\mathbf{y}}]\mathbf{f} + \mathbb{E}[\tilde{\mathbf{y}}]\epsilon - \mathbb{E}[\tilde{\mathbf{y}}]\tilde{\mathbf{y}} + (\mathbb{E}[\tilde{\mathbf{y}}])^2 - (\mathbb{E}[\tilde{\mathbf{y}}])^2 \\
&\quad \left. - \mathbb{E}[\tilde{\mathbf{y}}]\mathbf{f} - \mathbb{E}[\tilde{\mathbf{y}}]\epsilon + \mathbb{E}[\tilde{\mathbf{y}}]\tilde{\mathbf{y}} - (\mathbb{E}[\tilde{\mathbf{y}}])^2 + (\mathbb{E}[\tilde{\mathbf{y}}])^2\right], \\
&= \mathbb{E}\left[\mathbf{f}^2 + \mathbf{f}\epsilon + \epsilon\mathbf{f} + \epsilon^2 - \mathbf{f}\mathbb{E}[\tilde{\mathbf{y}}] - \epsilon\mathbb{E}[\tilde{\mathbf{y}}] - \mathbb{E}[\tilde{\mathbf{y}}]\mathbf{f} - \mathbb{E}[\tilde{\mathbf{y}}]\epsilon + (\mathbb{E}[\tilde{\mathbf{y}}])^2\right] \\
&\quad + \mathbb{E}\left[\tilde{\mathbf{y}}^2 - \tilde{\mathbf{y}}\mathbb{E}[\tilde{\mathbf{y}}] - \mathbb{E}[\tilde{\mathbf{y}}]\tilde{\mathbf{y}} + (\mathbb{E}[\tilde{\mathbf{y}}])^2\right] \\
&\quad + \mathbb{E}\left[-\mathbf{f}\tilde{\mathbf{y}} - \epsilon\tilde{\mathbf{y}} + \mathbf{f}\mathbb{E}[\tilde{\mathbf{y}}] + \epsilon\mathbb{E}[\tilde{\mathbf{y}}] - \tilde{\mathbf{y}}\mathbf{f} - \tilde{\mathbf{y}}\epsilon + \tilde{\mathbf{y}}\mathbb{E}[\tilde{\mathbf{y}}] + \mathbb{E}[\tilde{\mathbf{y}}]\mathbf{f} + \mathbb{E}[\tilde{\mathbf{y}}]\epsilon + \mathbb{E}[\tilde{\mathbf{y}}]\tilde{\mathbf{y}} - 2(\mathbb{E}[\tilde{\mathbf{y}}])^2\right].
\end{aligned}$$

Before we move further we may note that the exact function $f(\mathbf{x})$ generally is not known, and we may therefore assume that our data is a good representation and replace \mathbf{f} with \mathbf{y} in the expression above. In practise this \mathbf{y} is then the part of the data set that we have chosen as test set, while the model is made with the remaining data set (the training set). **correct?** Thus, using that $\mathbb{E}[\mathbb{E}[\mathbf{x}]] = \mathbb{E}[\mathbf{x}]$, $\mathbb{E}[(\mathbb{E}[\mathbf{x}])^2] = (\mathbb{E}[\mathbf{x}])^2$ and $\mathbb{E}[\mathbf{x}, \mathbf{y}] = \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}]$ for any statistically independent \mathbf{x} and \mathbf{y} , and that $\mathbb{E}[\epsilon] = 0$ so that we can remove all first order terms in ϵ , we get

$$\begin{aligned}
\mathbb{E}[(\mathbf{y} - \tilde{\mathbf{y}})^2] &= \mathbb{E}\left[\mathbf{y}^2 + \epsilon^2 - \mathbf{y}\mathbb{E}[\tilde{\mathbf{y}}] - \mathbb{E}[\tilde{\mathbf{y}}]\mathbf{y} + (\mathbb{E}[\tilde{\mathbf{y}}])^2\right] \\
&\quad + \mathbb{E}\left[\tilde{\mathbf{y}}^2 - \tilde{\mathbf{y}}\mathbb{E}[\tilde{\mathbf{y}}] - \mathbb{E}[\tilde{\mathbf{y}}]\tilde{\mathbf{y}} + (\mathbb{E}[\tilde{\mathbf{y}}])^2\right] \\
&\quad + \mathbb{E}\left[-\mathbf{y}\tilde{\mathbf{y}} + \mathbf{y}\mathbb{E}[\tilde{\mathbf{y}}] - \tilde{\mathbf{y}}\mathbf{y} + \tilde{\mathbf{y}}\mathbb{E}[\tilde{\mathbf{y}}] + \mathbb{E}[\tilde{\mathbf{y}}]\mathbf{y} + \mathbb{E}[\tilde{\mathbf{y}}]\tilde{\mathbf{y}} - 2(\mathbb{E}[\tilde{\mathbf{y}}])^2\right], \\
&= \mathbb{E}\left[(\mathbf{y} - \mathbb{E}[\tilde{\mathbf{y}}])^2\right] + \mathbb{E}[\epsilon^2] + \mathbb{E}\left[(\tilde{\mathbf{y}} - \mathbb{E}[\tilde{\mathbf{y}}])^2\right] \\
&\quad - \mathbb{E}[\mathbf{y}]\mathbb{E}[\tilde{\mathbf{y}}] + \mathbb{E}[\mathbf{y}]\mathbb{E}[\tilde{\mathbf{y}}] - \mathbb{E}[\tilde{\mathbf{y}}]\mathbb{E}[\mathbf{y}] + \mathbb{E}[\tilde{\mathbf{y}}]\mathbb{E}[\tilde{\mathbf{y}}] + \mathbb{E}[\tilde{\mathbf{y}}]\mathbb{E}[\mathbf{y}] + \mathbb{E}[\mathbf{y}]\mathbb{E}[\tilde{\mathbf{y}}] - 2(\mathbb{E}[\tilde{\mathbf{y}}])^2, \\
&= \text{Bias}[\tilde{\mathbf{y}}] + \text{Var}[\tilde{\mathbf{y}}] + \sigma^2. \tag{A7}
\end{aligned}$$

Appendix B: Additional Figures