Deep learning for agronomy

3rd year engineer

Ammouri Bilel

: ESAM
in: ammouri-bilel
: bilelammouri
: Ammouri-Bilel
: 0000-0002-5491-5172

Plan

Regression analysis

Definition

- Regression analysis is a predictive modeling technique that examines the relationship between the target or dependent variable and independent variables in a dataset.
- Different types of regression analysis techniques are employed when the target and independent variables demonstrate a linear or non-linear correlation, and the target variable consists of continuous values .
- This technique is primarily utilized to assess predictor strength, forecast trends, analyze time series data, and establish cause and effect relationships.

Regression analysis

Types of Regression Analysis Techniques

There are numerous types of regression analysis techniques, each chosen based on various factors. These factors include the nature of the target variable, the form of the regression line, and the number of independent variables.

1. Linear Regression
2. Logistic Regression
3. Ridge Regression
4. Lasso Regression
5. Polynomial Regression
6. Gausian Regression
7. Bayesian Linear Regression
8. Neural Network Regression
9. Decision Tree Regression
10. Support Vector Regression (SVR)

Regression analysis

Linear Regression

The linear regression model consists of a predictor variable and a dependent variable related linearly to each other.
There are various types of linear regression, with the two primary forms being simple linear regression and multiple linear regression.

$$Y_i = \alpha_0 + \alpha_1 \times X_i + \epsilon_i$$

$$Y_i = \alpha_0 + \alpha_1 \times X_i^1 + \cdots + \alpha_p \times X_i^p + \epsilon_i$$

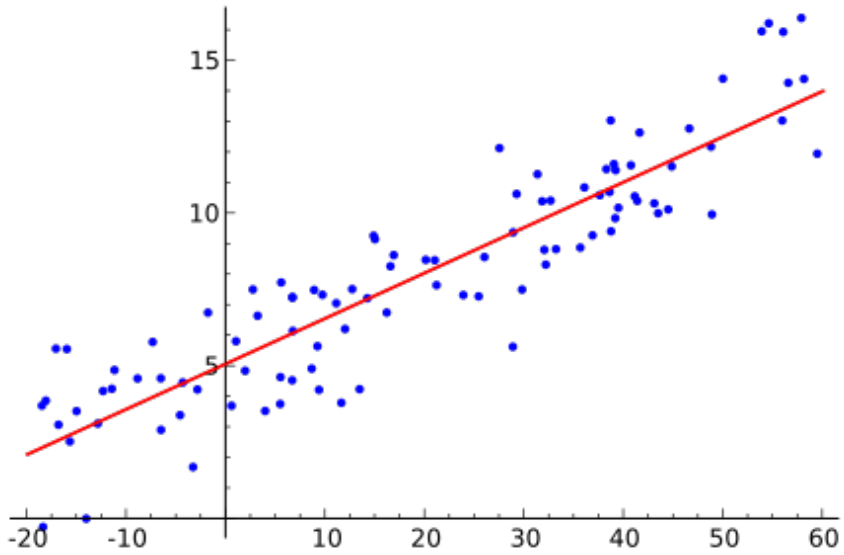


Figura 1: Linear Regression

Regression analysis

Logistic Regression

Logistic regression is a type of regression analysis technique used when the dependent variable is discrete, for example, 0 or 1, true or false, etc.
This implies that the target variable can only assume two values, with the relationship between the target variable and the independent variable being denoted by a sigmoid curve.

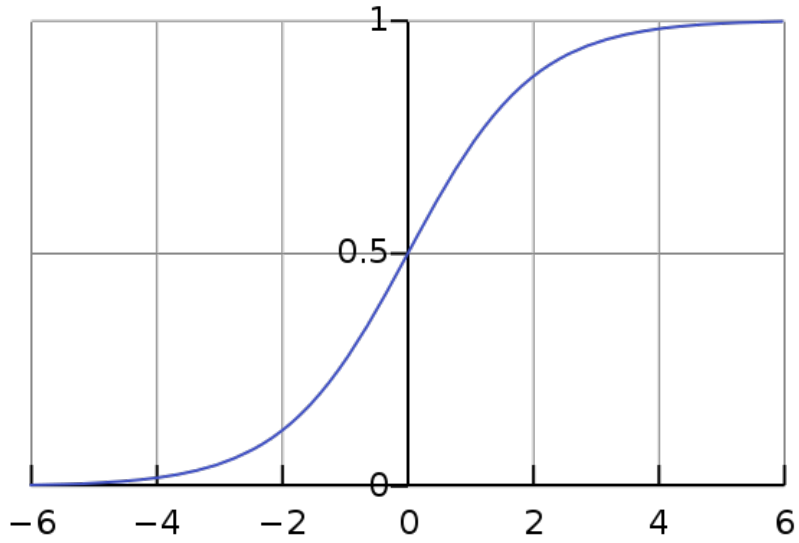


Figura 2: Logistic Regression

$$Logit(p) = \ln \frac{p}{(1-p)} = \alpha_0 + \alpha_1 \times X_i^1 + \cdots + \alpha_p \times X_i^p$$

Regression analysis

Ridge Regression

Ridge regression is typically used when there is a high correlation between the independent variables. This is because, in the case of multicollinear data, the least square estimates provide unbiased values. However, when collinearity is very high, some bias may occur.
To address this issue, a bias matrix is introduced in the equation of Ridge Regression.
This powerful regression method helps make the model less susceptible to overfitting.

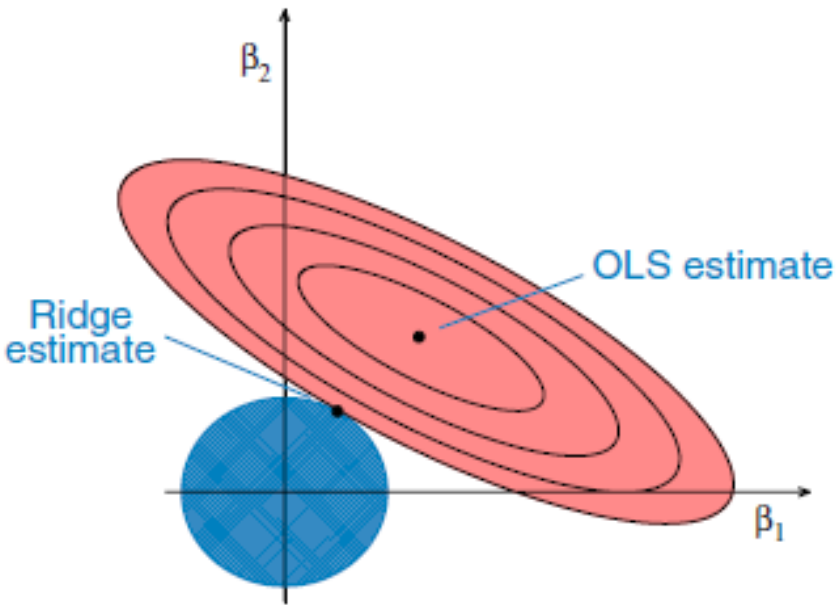


Figura 3: Ridge Regression

$$\beta = (X^{tr}X + \lambda \times I)^{-1}X^{tr}Y$$

Regression analysis

Lasso Regression

Lasso Regression is a regression technique that combines regularization with feature selection, restricting the absolute size of the regression coefficient.
Consequently, the coefficient value approaches zero, distinguishing it from Ridge Regression.
This feature selection characteristic in Lasso Regression allows for the selection of a specific set of features from the dataset to build the model.
With Lasso Regression, only the necessary features are utilized, while the others are forced to zero.
This helps prevent overfitting in the model.
In instances where the independent variables are highly collinear, Lasso regression selects only one variable and reduces the others to zero.

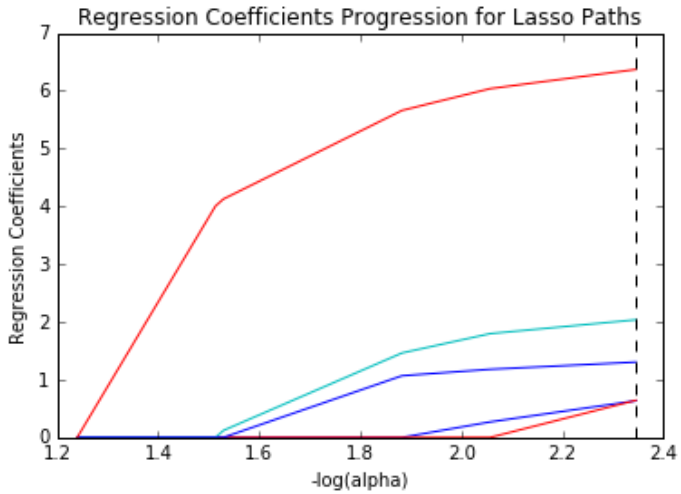


Figura 4: Lasso Regression

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i, y_I, \alpha, \beta)$$

Regression analysis

Polynomial Regression

Polynomial Regression represents the relationship between the independent and dependent variables, denoted by the nth degree of X and Y.
It serves as a linear model estimator, utilizing the Least Mean Squared Method in its calculations.
Unlike simple linear regression, the best fit line in Polynomial Regression is not a straight line, but rather a curved line that is determined by the power of X or the value of n.

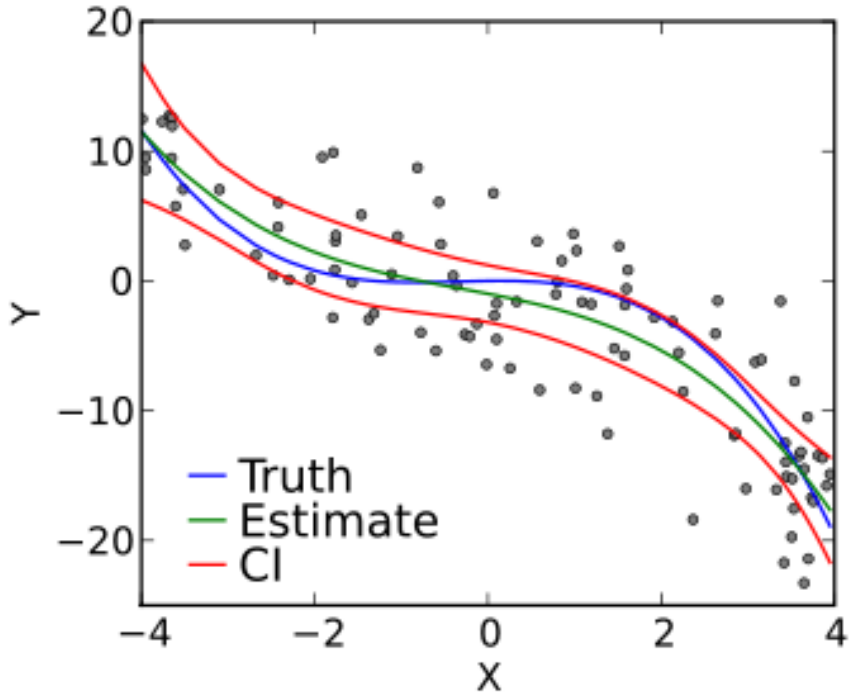


Figura 5: Polynomial Regression

$$Y_i = \alpha_0 + \alpha_1 \times X_i^1 + \cdots + \alpha_d \times X_i^d + \epsilon_i$$

Regression analysis

Gaussian Regression

Gaussian regression algorithms are known for their representation flexibility and inherent uncertainty measures in predictions. A Gaussian process is built upon fundamental concepts, including the multivariate normal distribution, non-parametric models, kernels, and joint and conditional probabilities.
A Gaussian process regression (GPR) model can make predictions using prior knowledge (kernels) and provide uncertainty measures for these predictions. It is a supervised learning method developed by the computer science and statistics communities.

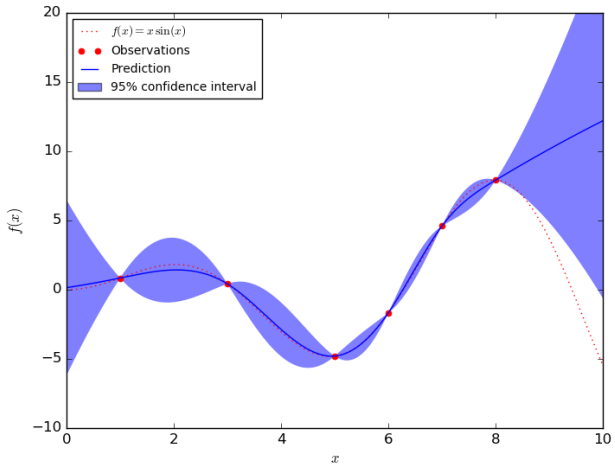


Figura 6: Gaussian Regression

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

$$PDF : P_X(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Due to the nonparametric nature of Gaussian process regression, it is not restricted by any specific functional form. Instead of calculating the probability distribution of a specific function's parameters, GPR calculates the probability distribution of all permissible functions that fit the data.

Regression analysis

Bayesian Linear Regression

Bayesian Regression is a statistical approach that utilizes Bayes' theorem to determine the values of regression coefficients.
This method focuses on determining the posterior distribution of features instead of solely finding the least squares.
Bayesian Linear Regression combines elements from both Linear Regression and Ridge Regression but is notably more robust than simple Linear Regression.

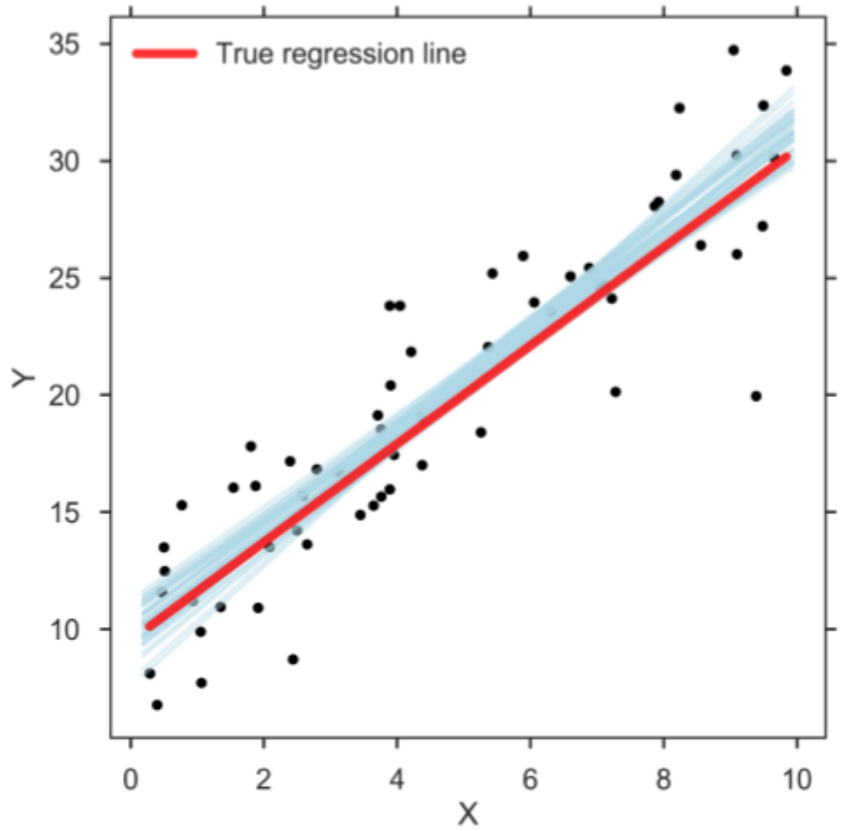


Figura 7: Bayesian Linear Regression

Regression analysis

Decision Tree Regression

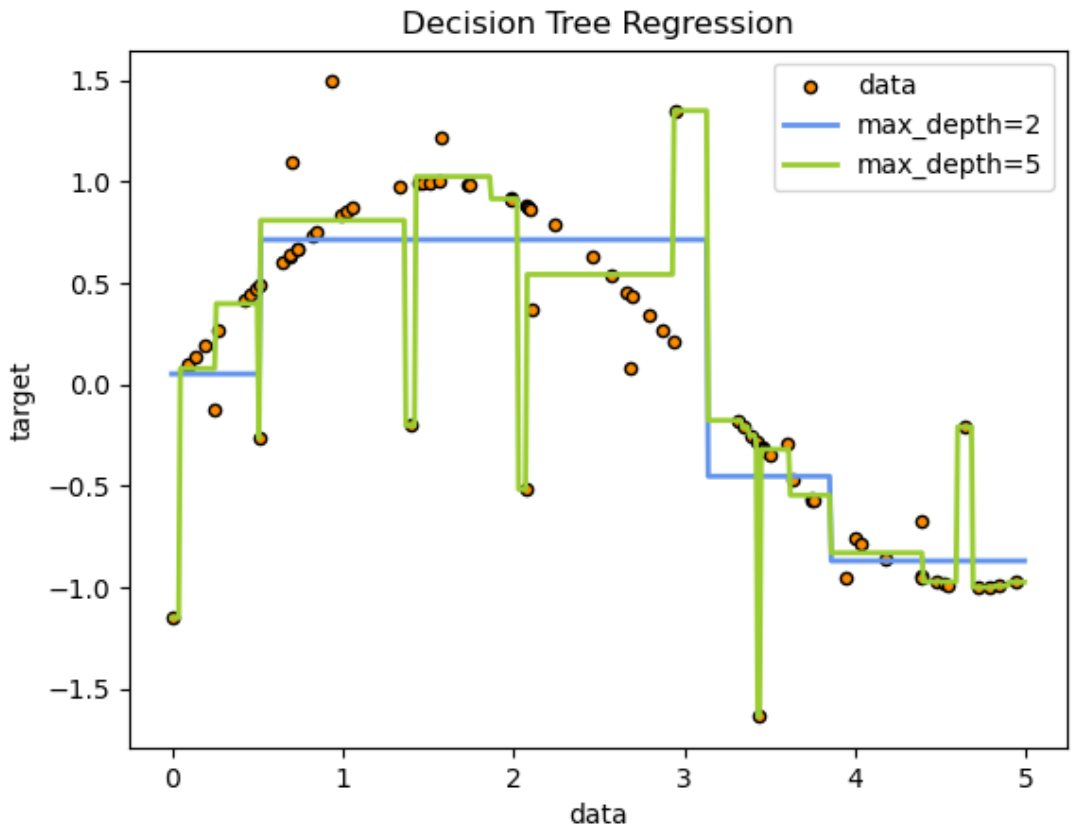


Figura 8: Decision Tree Regression

Regression analysis

Support Vector Regression

Model

$$y = wx + b$$

Minimize

$$\frac{1}{2}\|w\|^2 c + \sum_{i=1}^{N} (\xi_i + \xi_i^*)$$

Constraints

$$y_i - wx_i - b \leq \epsilon + \xi_i$$

$$wx_i + b - y_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

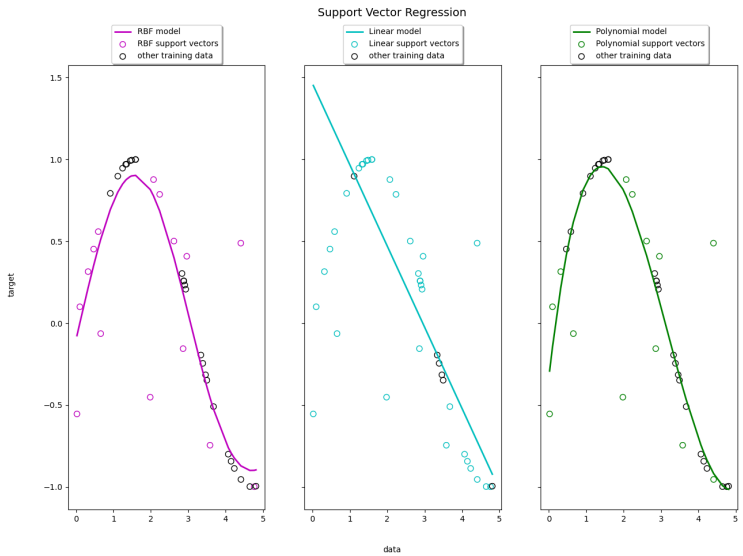


Figura 9: Support Vector Regression

Performance Metrics

Evaluating the performance of a machine learning regression model is crucial to understand how well the model is performing and whether it can generalize to new, unseen data. There are several commonly used metrics to evaluate the performance of a regression model.

Performance Metrics

Train/Test Split

- It is crucial to strike a balance when choosing the appropriate size for the test set in estimating future performance.
- Commonly, a 80:20 or 90:10 ratio is allocated for the training set to the test set.
- The size of the test set directly impacts the reliability of the estimated model performance.
- Allocating more data to the test set reduces the amount available for training, potentially limiting the model's learning capacity, particularly when obtaining additional data is not feasible.
- Adhering to the principle of never training the model on data from the test set is imperative to ensure a realistic assessment of its generalization capabilities.
- The determination of the test set size should consider the specific problem requirements, data availability, and the trade-offs between training and testing data sizes.
- This strategic balance contributes to the development of a robust and reliable machine learning model for future predictions.

Performance Metrics

Mean Squared Error (MSE)

MSE measures the average of the squares of the errors or deviations — the difference between the actual and predicted values. It penalizes larger errors more than smaller ones.

MSE

$$MSE = \frac{1}{N}\sum_{N}(Y_i - \hat{Y}_i)^2$$

Performance Metrics

Root Mean Squared Error (RMSE)

RMSE is the square root of the MSE, providing a measure of the standard deviation of the residuals. It is in the same unit as the target variable, making it easily interpretable.

RMSE

$$RMSE = \sqrt{\frac{1}{N}\sum_{N}(Y_i - \hat{Y}_i)^2}$$

Performance Metrics

Mean Absolute Error (MAE)

MAE calculates the average of the absolute differences between the predicted and actual values. It is less sensitive to outliers compared to MSE.

MAE

$$MAE = \frac{1}{N} \sum_{N} \left| Y_i - \hat{Y}_i \right|$$

Performance Metrics

R-squared (R^2)

R-squared represents the proportion of the variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, with 1 indicating that the model explains all the variability of the response data around its mean.

R^2

$$R^2 = \frac{SCE}{SCT} = 1 - \frac{SCR}{SCT} = 1 - \frac{\sum_N (Y - \hat{Y}_i)^2}{\sum_N (Y_i - \bar{Y})^2}$$

Performance Metrics

Mean Absolute Percentage Error (MAPE)

MAPE measures the prediction accuracy of a forecasting method in percentage terms. It provides a percentage error for each data point and averages these values.

MAPE

$$MAPE = \frac{1}{N} \sum_{N} \left| \frac{(Y_i - \hat{Y}_i}{Y_i} \right| \times 100\%$$

Performance Metrics

Train/True Error

True Error

"True Error"refers to the overall error of the model when it is applied to all possible instances, including those not present in the training dataset. It represents the model's performance in making predictions on unseen data. The True Error is typically not directly measurable due to the unavailability of the complete population data.

Train Error

"Train Error"or "Training Error"refers to the error rate that the model produces on the data points it was trained on. It is the error between the predicted values and the actual values of the data points in the training set. The Training Error is an estimation of how well the model fits the training data.

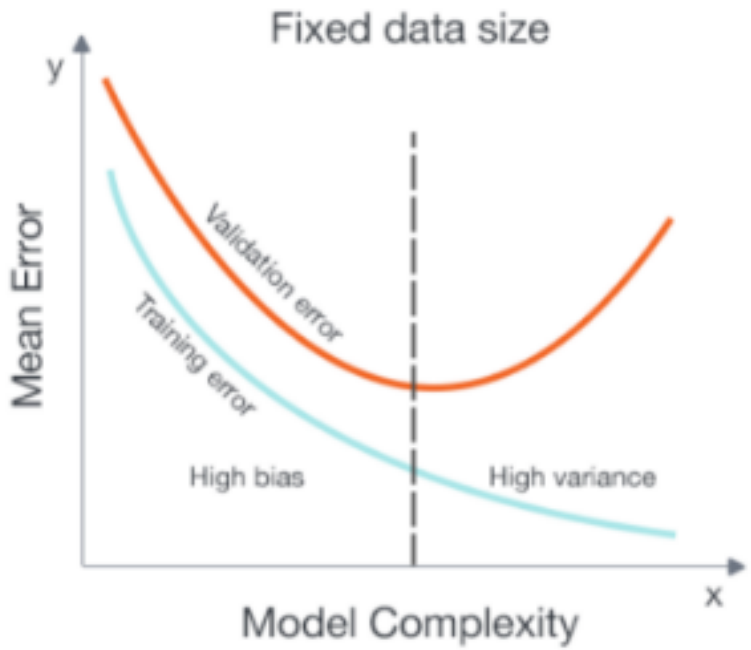


Figura 10: Train/True Error

Performance Metrics

Under/Over fitting

Underfitting

happens when a model cannot capture the complex patterns between a training set's features and its output values.
Underfitting is usually easy to fix because we can get a low training error by:

- Removing uninformative features
- Using more complex models (by adding more features or introducing more non-linearities to capture non-linear patterns)

Overfitting

happens when we too closely match the training data and fail to generalize.
Overfitting occurs when you train a predictor $\hat{w}$ but there exists another predictor $w^{'}$ from the same model class such that:

- $error_{true}(w^{'}) < error_{true}(\hat{w})$
- $error_{train}(w^{'}) > error_{train}(\hat{w})$

Performance Metrics

Under/Over fitting

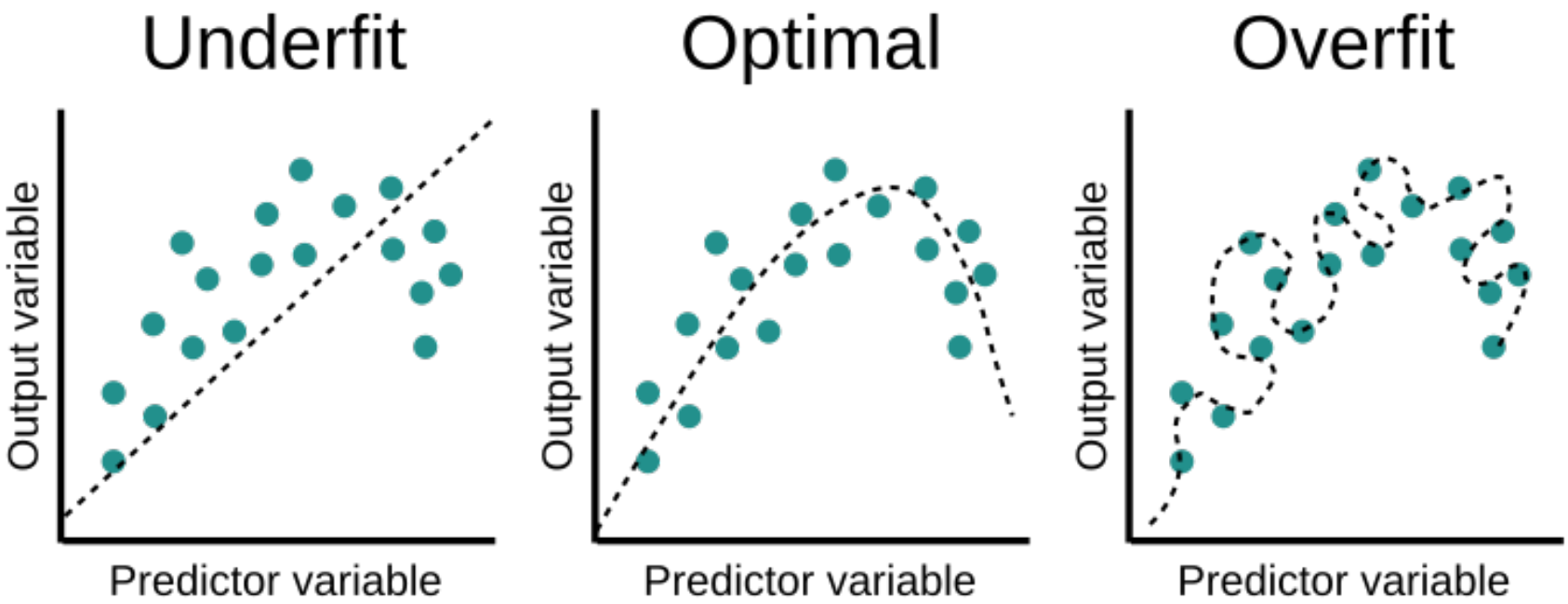


Figura 11: Under/Over fitting

Source of errors

Bias

A model that is too simple fails to capture the complex patterns in the dataset, which signifies a fundamental limitation of the model. We call this a bias error.
Bias is the difference between the average prediction of our model and the expected value which we are trying to predict.

Variance

A model that is too complicated for the task overly fits to small fluctuations. The flexibility of the complicated model makes it capable of memorizing answers rather than learning general patterns. This contributes to the error as variance.
Variance is the variability in the model prediction, meaning how much the predictions will change if a different training dataset is used.

Irreducible Errors

Irreducible error is the one that we can't avoid or possibly eliminate. They are caused by elements outside of our control, such as noise from observations.

Tradeoff

Tradeoff between bias and variance:

- Simple models: High bias + Low variance
- Complex models: Low bias + High variance

Source of errors for a particular model $\hat{f}$ using *MSE* loss function:

$$\mathbb{E}\left[\left(y-\hat{f}(x)\right)^2\right] = bias\left[\hat{f}(x)\right]^2 + var\left(\hat{f}(x)\right) + \sigma_\epsilon^2$$

Error = Biased squared + Variance + Irreducible Error

Tradeoff

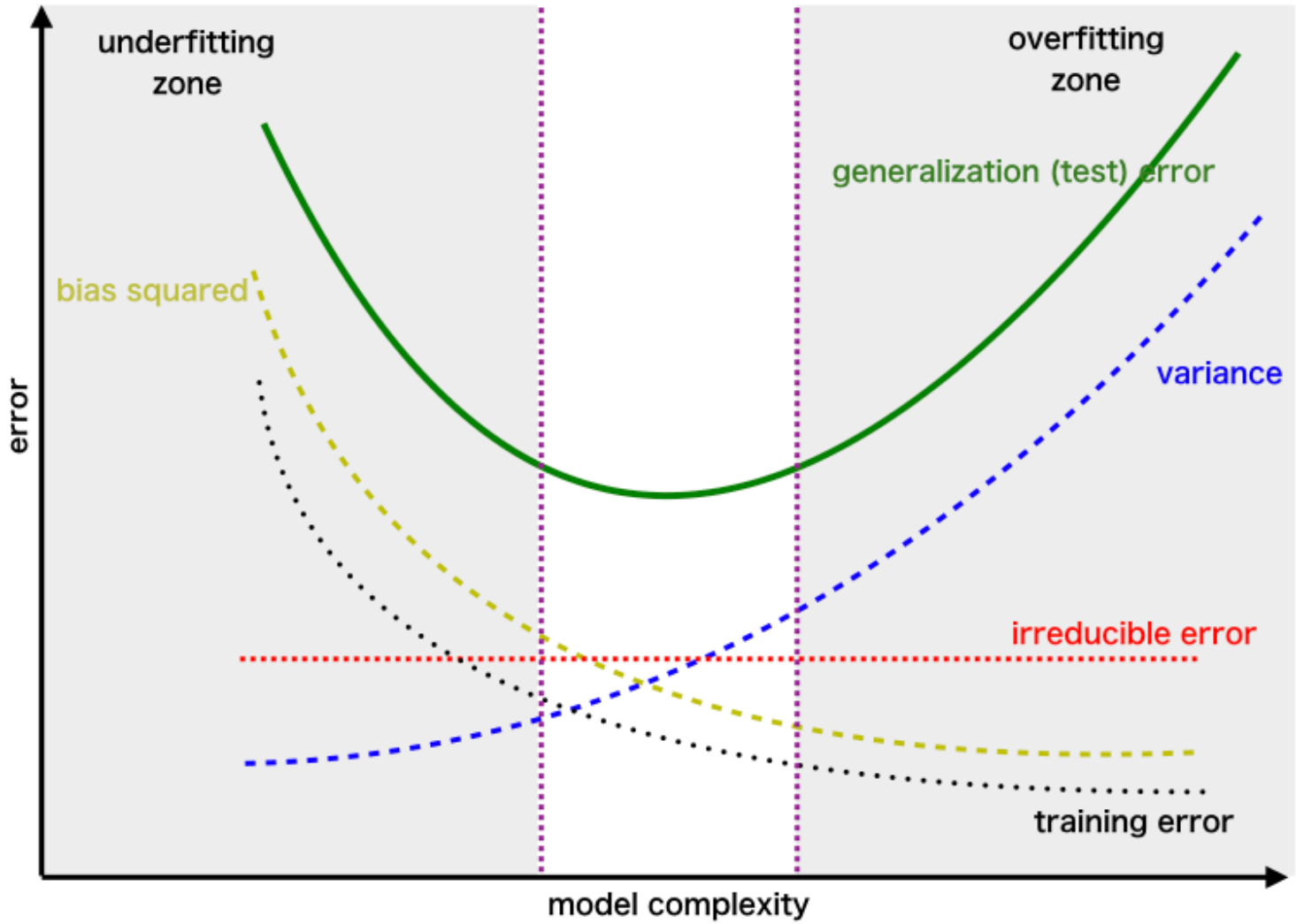


Figura 12: Tradeoff

Dataset Size error

Model complexity doesn't depend on the size of the training set
The larger the training set, the lower the variance of the model, thus less overfitting

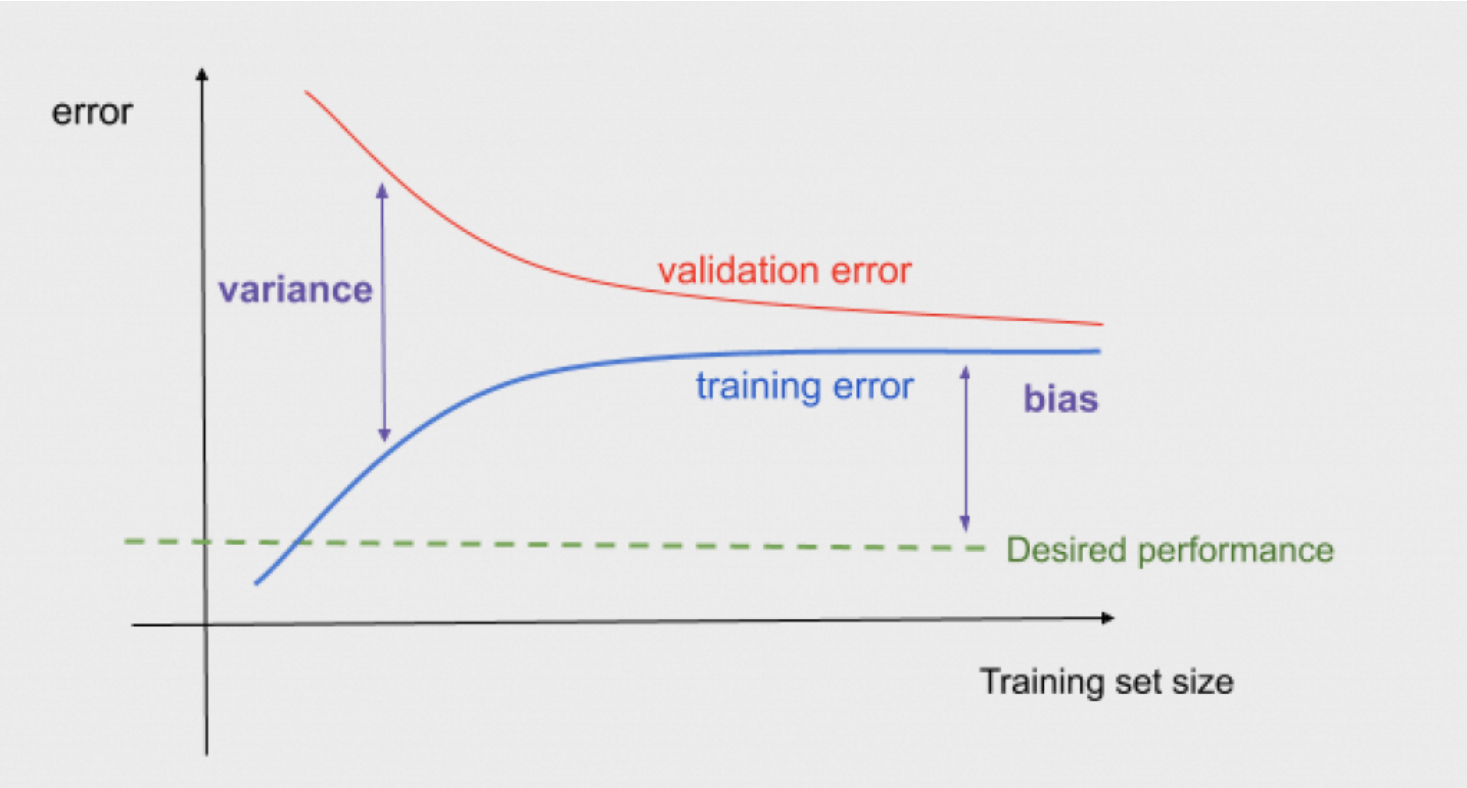


Figura 13: Tradeoff

Lineaire regression

Goal

Predict how much my agricultural land is worth.
Have data from my neighborhood :

$$(x_1, y_1) = (2.3, 300$$
$$(x_2, y_2) = (1.9, 250)$$
$$\vdots$$
$$(x_n, y_n) = (2, 320)$$

Assumption

There is a relationship between $y \in \mathbb{R}$ and $x \in \mathbb{R}^d$

$$y \approx f(x)$$

x is the input data. Can potentially have many inputs
y is the outcome/response/target/label/dependent variable

Lineaire regression

Model

A model is how we assume the world works : **Regression model**

$$\begin{aligned} y &= f(x) \\ &= \sum w_i h_i(x) \\ &= w^{tr} h(x) \end{aligned}$$

Quality Metric / Loss function

$$MSE = \frac{1}{N} \sum_{N} (Y_i - \hat{Y}_i)^2$$

Predictor

$$\hat{w} = \arg\min_{w} MSE(w)$$

Optimization Algorithm

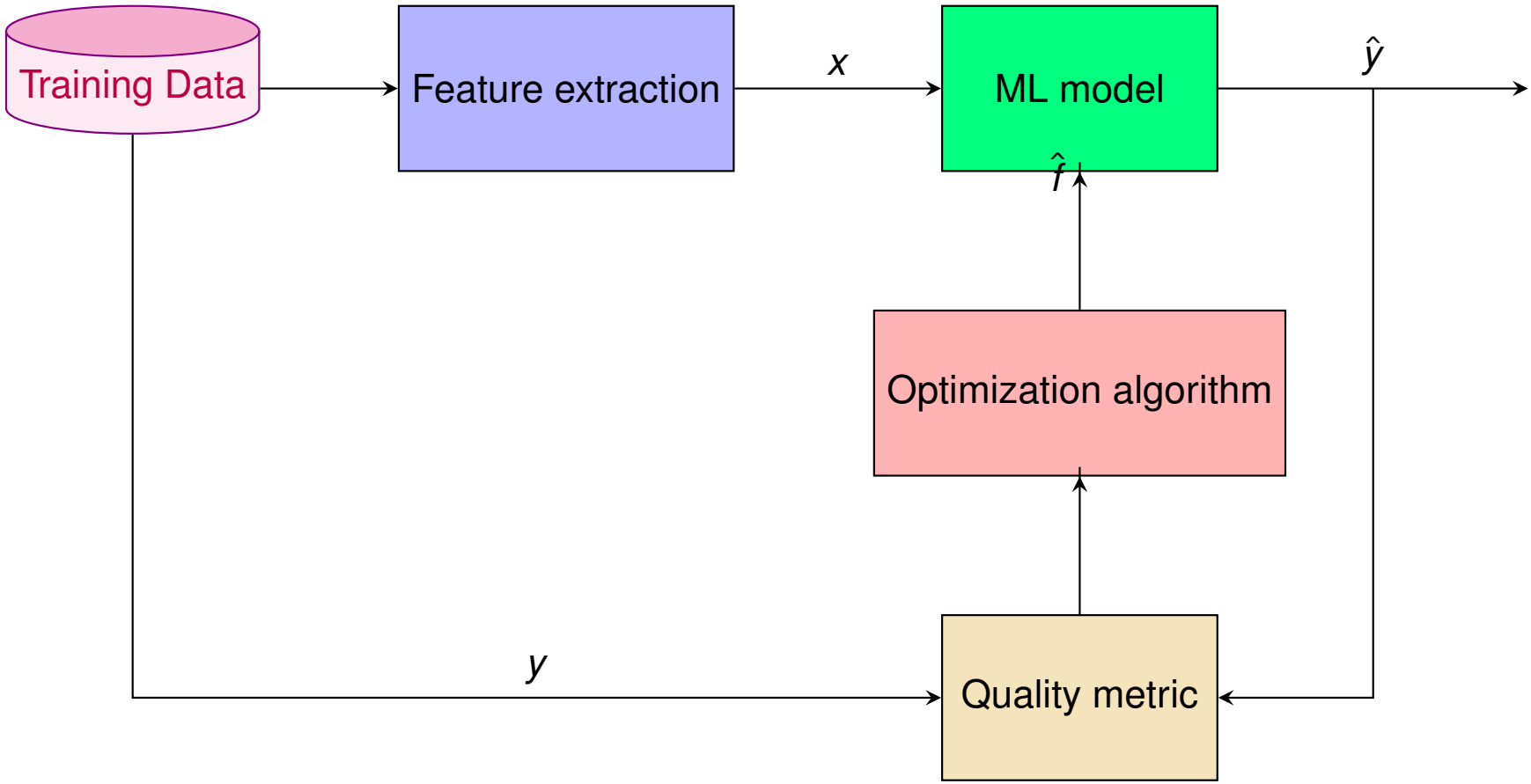
Optimized using Gradient Descent

Prediction

We don't know f! We need to learn it from the data! Use machine learning to learn a predictor $\hat{f}$ from the data
For a given input x, predict:
$\hat{y} = \hat{f}(x)$

Lineaire regression

Pipeline



MSE

Sort the following lines by their MSE (mean-squared errors) on the data, from smallest to largest. (estimate, don't actually compute).

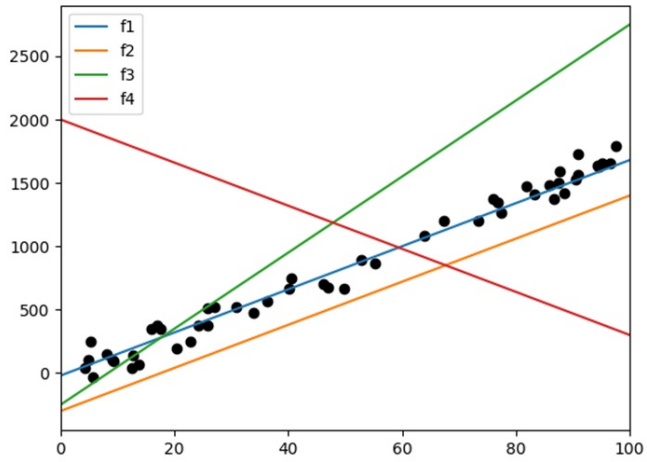


Figura 14: Caption

Overfiting

What describes overfitting?

1. Low train error, low test error
2. High train error, high test error
3. Low train error, high test error
4. High train error, low test error

The right degree polynomial

Suppose I wanted to figure out the right degree polynomial for my data-set (we'll try p from 1 to 20). What procedure should I use to do this? Pick the best option

For each possible degree polynomial p:

1. Train a model with degree p on the training set, pick p that has the lowest test error
2. Train a model with degree p on the training set, pick p that has the highest test error
3. Train a model with degree p on the test set, pick p that has the lowest test error
4. Train a model with degree p on the test set, pick p that has the highest test error
5. None of the above