Politecnico di Bari

Politecnico di Bari

Dipartimento di Ingegneria Elettrica e dell'Informazione

Corso di Laurea Triennale in Ingegneria dei Sistemi Medicali

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE

Bioinformatics and Big Data Analytics

Model Selection

Eng. Nicola **Altini**, *Ph.D. Student*
Eng. Giacomo Donato **Cascarano**, *Ph.D. Student*
Prof. *Eng*. Vitoantonio **Bevilacqua**, *Ph.D.*

INDUSTRIAL INFORMATICS LAB
LABORATORIO DI INFORMATICA INDUSTRIALE

Anno Accademico 2019/2020

apulian bioengineering company

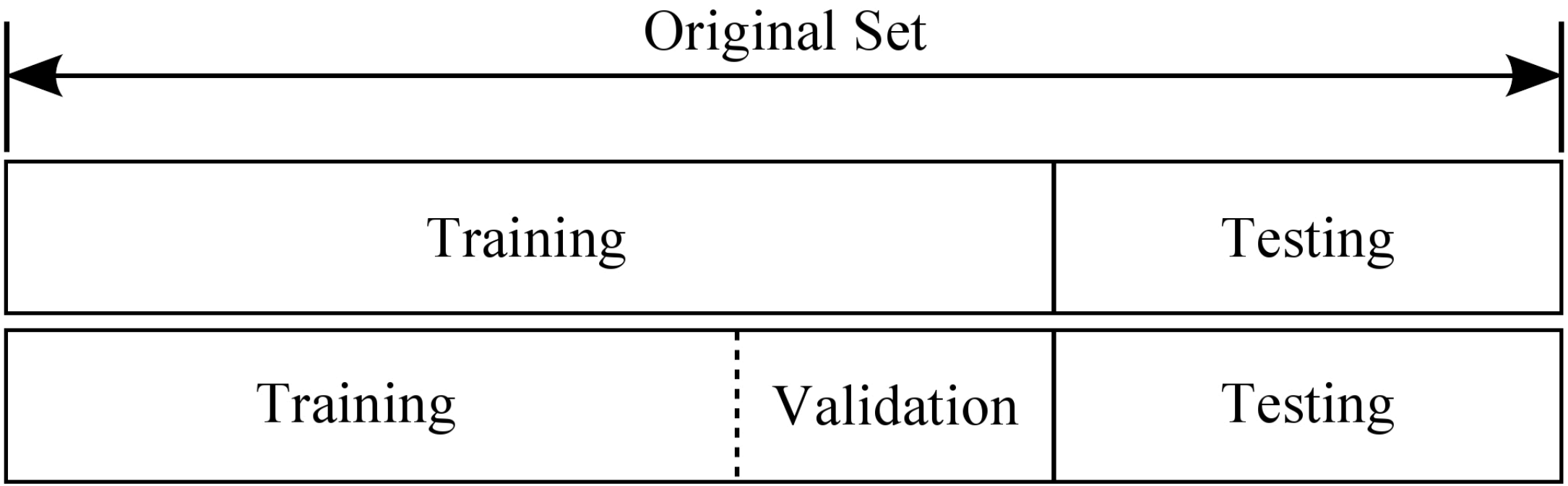
Model Selection

- Model selection is the task of selecting a statistical model from a set of candidate models, given data. In the simplest cases, a pre-existing set of data is considered.
- However, the task can also involve the design of experiments such that the data collected is well-suited to the problem of model selection.
- Given candidate models of similar predictive or explanatory power, the simplest model is most likely to be the best choice (**Occam's razor**).

Training, validation, and test sets

- The model is initially fit on a training dataset, that is a set of examples used to fit the parameters (e.g. weights of connections between neurons in artificial neural networks) of the model. The model (e.g. a neural net) is trained on the training dataset using a supervised learning method (e.g. gradient descent).
- Successively, the fitted model is used to predict the responses for the observations in a second dataset called the validation dataset. The validation dataset provides an unbiased evaluation of a model fit on the training dataset while tuning the model's hyperparameters.
- Finally, the test dataset is a dataset used to provide an unbiased evaluation of a final model fit.

Training, validation, and test sets



Training, validation, and test sets

```
function [x_train, y_train, x_test, y_test] = train_test_split(x,y,split_percentage)
    n_features = size(x,1);
    temp = [x;y];
    rng(0)
    p = randperm(size(temp,2)); % Genero un vettore di permutazioni casuali

    train_size = floor(size(temp,2)*split_percentage); % Percentuale di dati per il primo subset
    p_train = p(1:train_size); % Seleziono la percentuale di indici per il primo subset
    p_test = p(train_size+1:end); % Seleziono la percentuale di indici per il secondo subset

    % Divisione del dataset
    x_train = temp(1:n_features,p_train);
    y_train = temp(n_features+1,p_train);

    x_test = temp(1:n_features,p_test);
    y_test = temp(n_features,p_test);
end
```

Training, validation, and test sets

```
split_percentage = 0.8;
[x_trainval, y_trainval, x_test, y_test] = train_test_split(x,y,split_percentage);
[x_train, y_train, x_val, y_val] = train_test_split(x_trainval, y_trainval, split_percentage);
```

Example:
Size dataset = 100

Size TrainVal set = 80
Size Train set = 64
Size Val set = 16

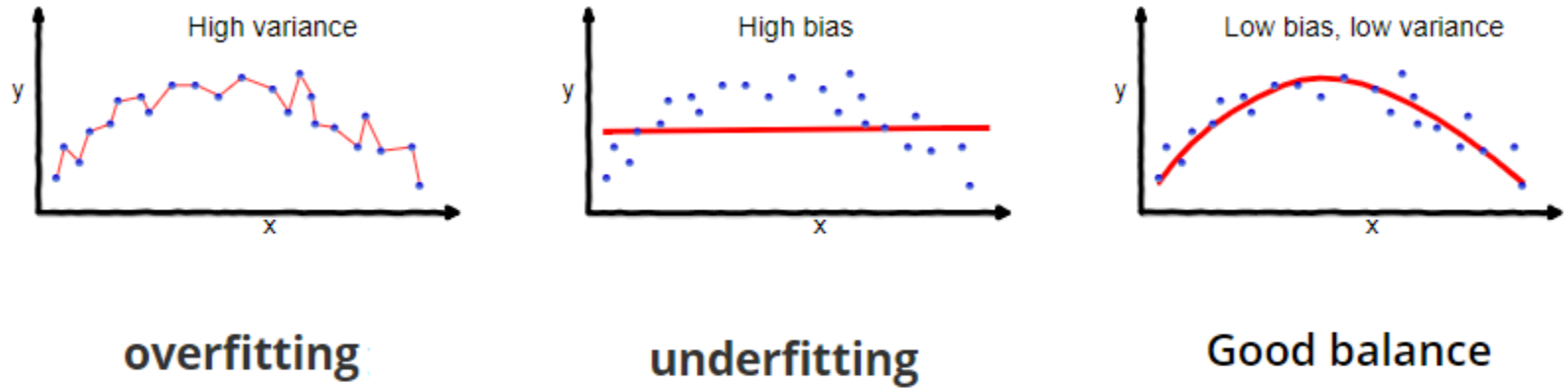
Size Test set = 20

Bias and Variance

- The bias–variance tradeoff is the property of a set of predictive models whereby models with a lower bias in parameter estimation have a higher variance of the parameter estimates across samples, and vice versa.
- The bias–variance decomposition is a way of analyzing a learning algorithm's expected generalization error with respect to a particular problem as a sum of three terms, the bias, variance, and a quantity called the irreducible error, resulting from noise in the problem itself.

Bias and Variance

- The **bias** error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (**underfitting**).
- The **variance** is an error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (**overfitting**).



Bias and Variance

- Supervised learning problem: find $f^*(x)$ which approximates true underlying function $y = f(x)$, where x is the independent variable and y is the dependent variable (target variable).
- Whichever function $f^*(X)$ we choose, we can decompose its expected error on an unseen examples as follows:

$$E\left[\left(y - f^*(x)\right)^2\right] = E\left[\left(f(x) + \epsilon - f^*(x)\right)^2\right] = (Bias[f^*(x)])^2 + Var[f^*(x)] + \sigma^2$$

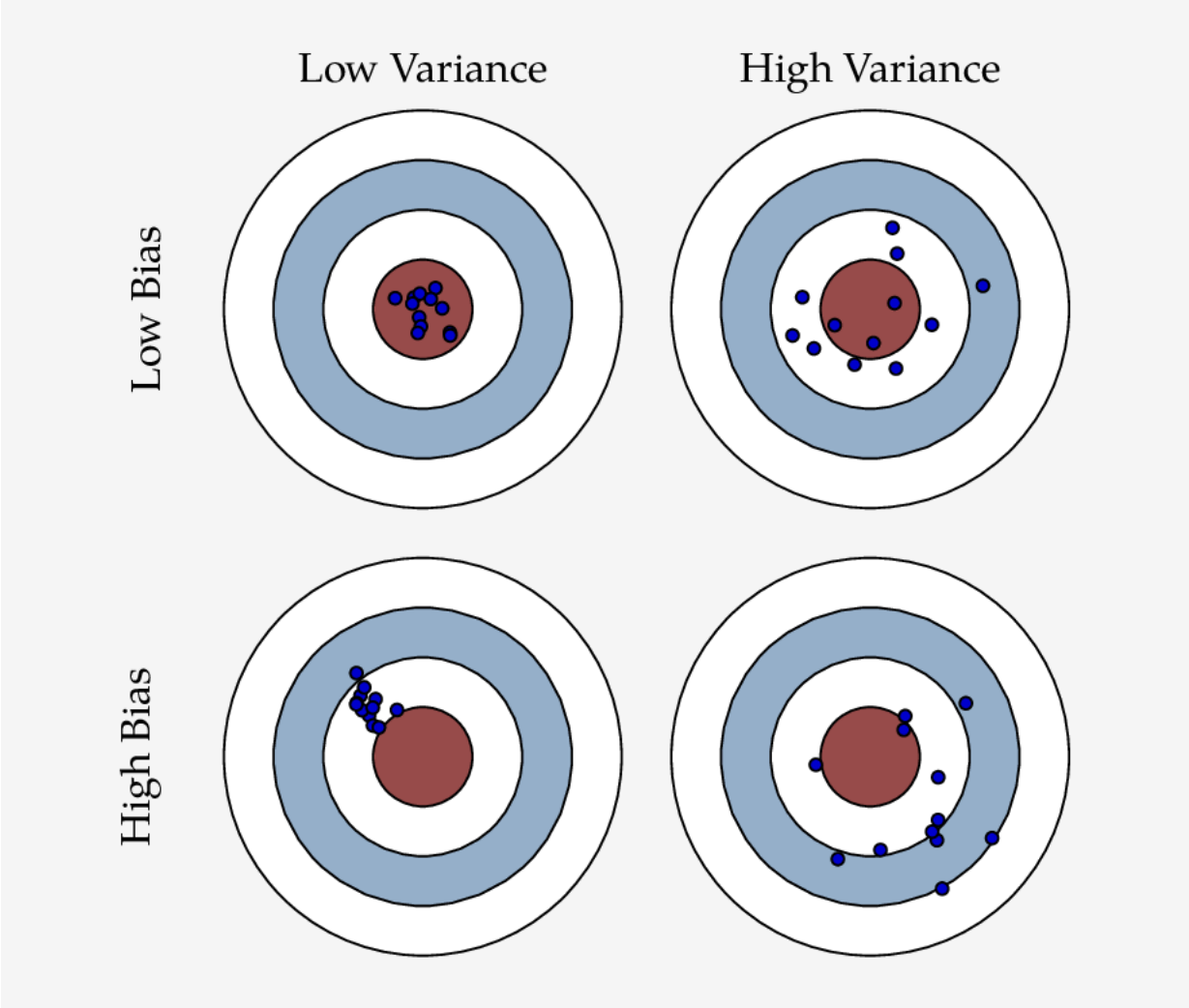
- Where

$$Bias[f^*(x)] = E[f^*(x)] - E[f(x)]$$

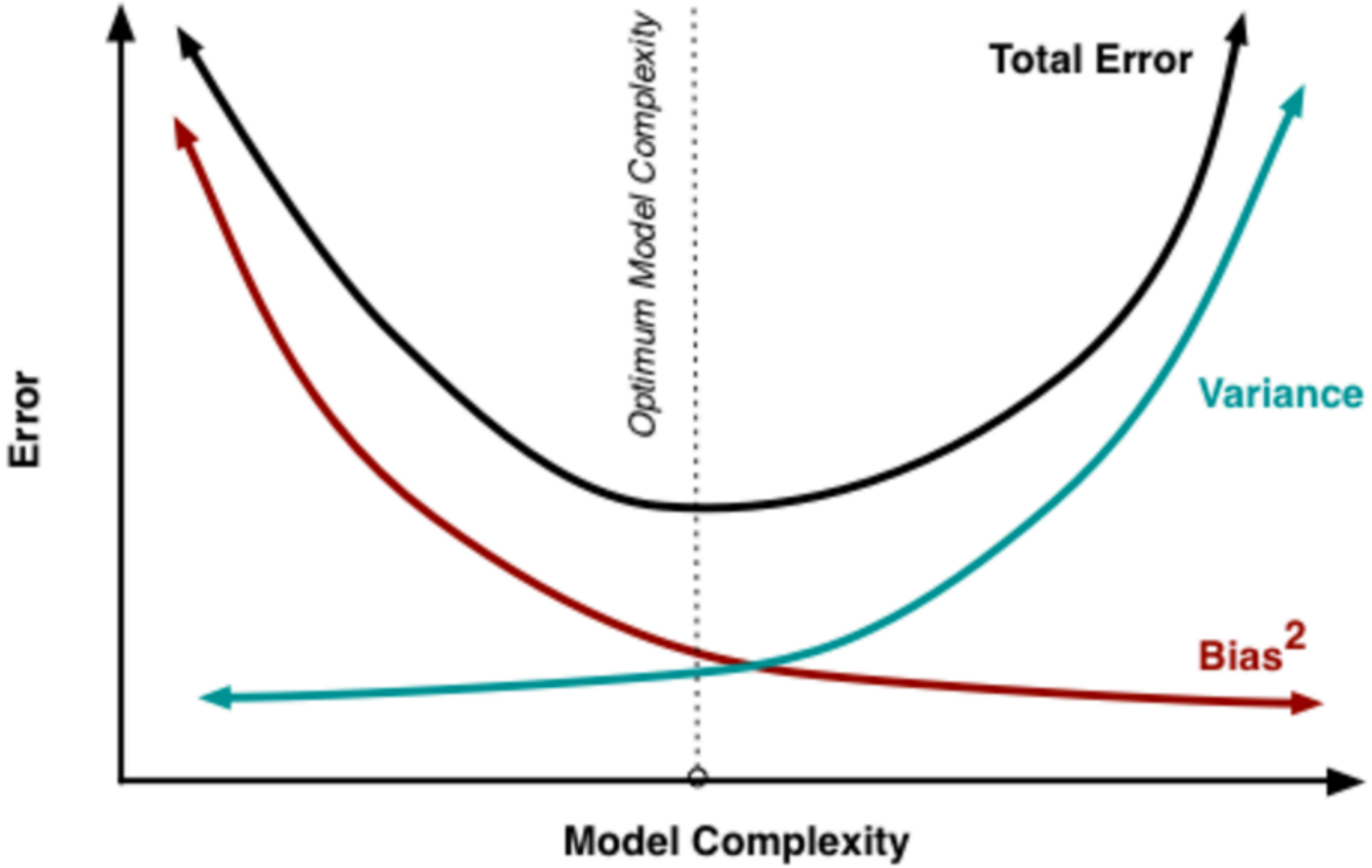
$$Var[f^*(x)] = E[f^*(x)^2] - E[f^*(x)]^2$$

σ^2 is the irreducible error

Bias and Variance



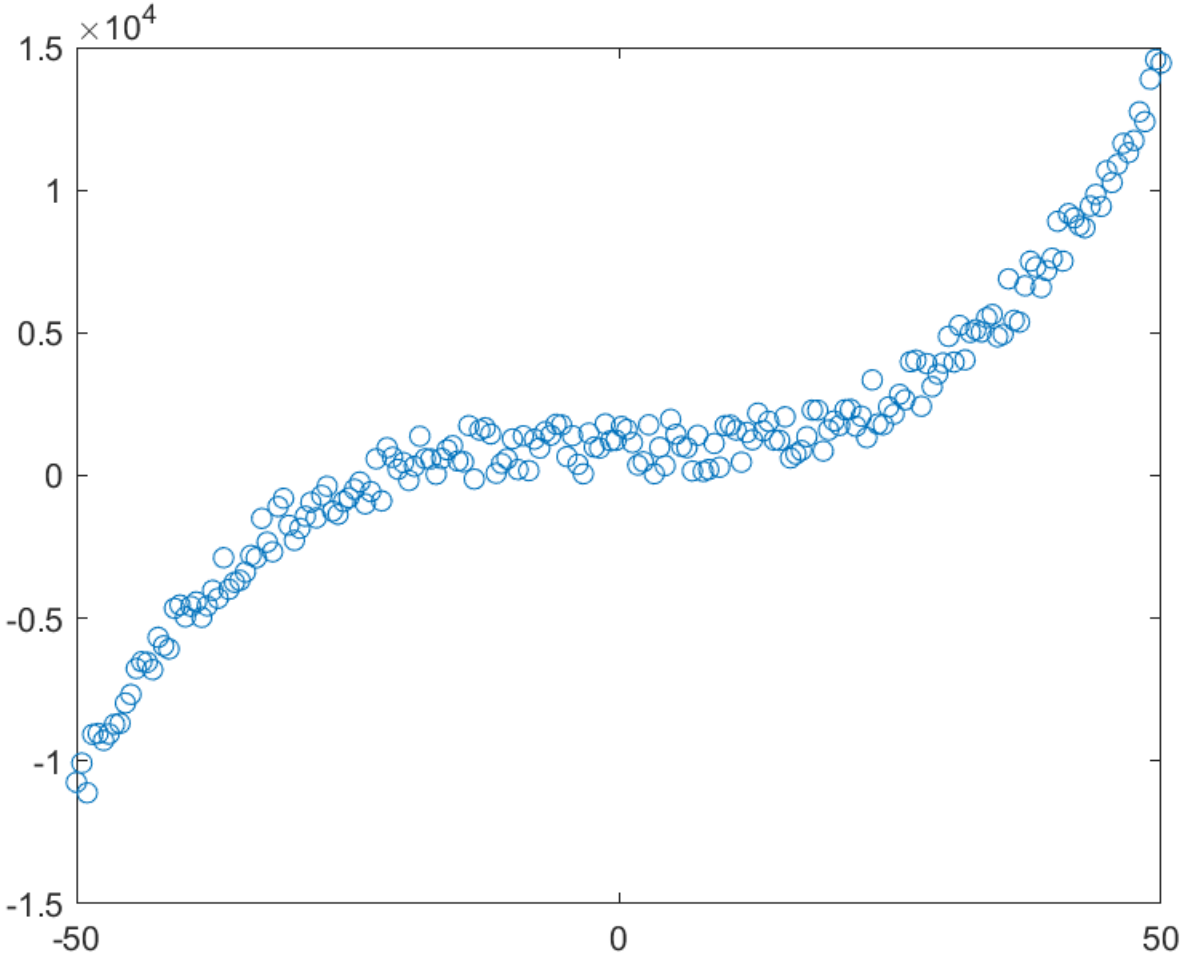
Bias and Variance



Polynomial Regression

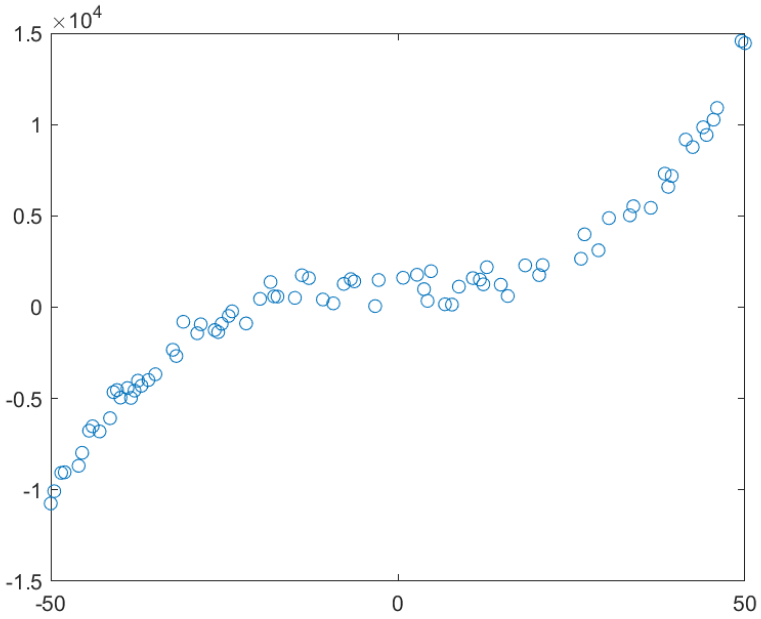
- Model capacity is represented by the grade of the polynomial.
- The higher is the grade, the higher is the model capacity. This means that the error on the train set generally will be reduced when we increase the grade. The error on the vail set may increase, since we are also increasing the variance of the model.
- How can we determine which is the optimal grade for a polynomial regression?
  - We use a train set to fit the model parameters (polyniamial coefficients).
  - We use a validation set to fit the model hyperparameters (grade of the polynomial).
  - We obtain a final unbiased estimate of the generalization capability of our model on a hold-out test set.

Polynomial Regression - Dataset

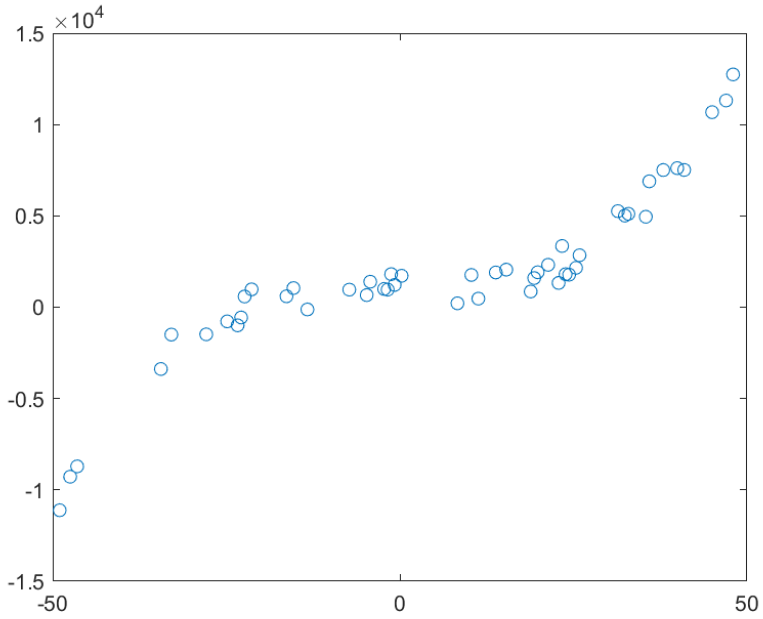


Train, Validation, Test sets

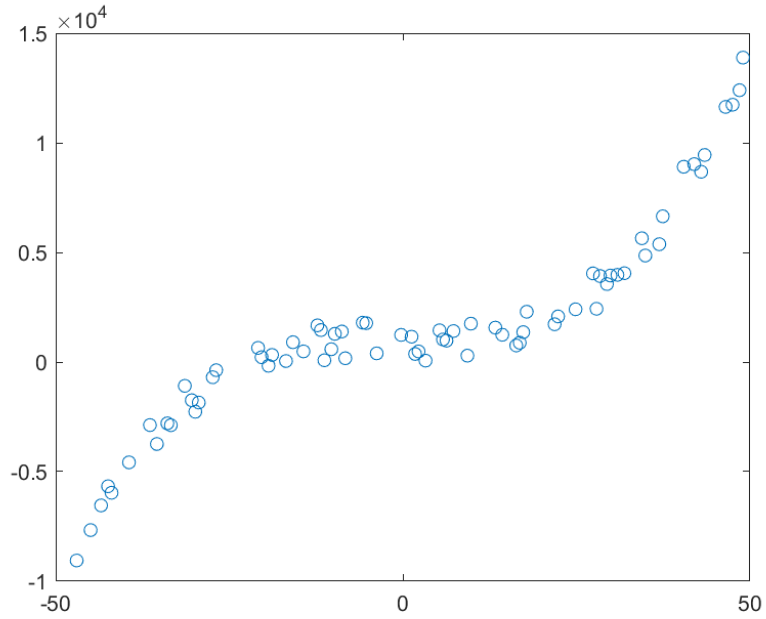
Train



Validation



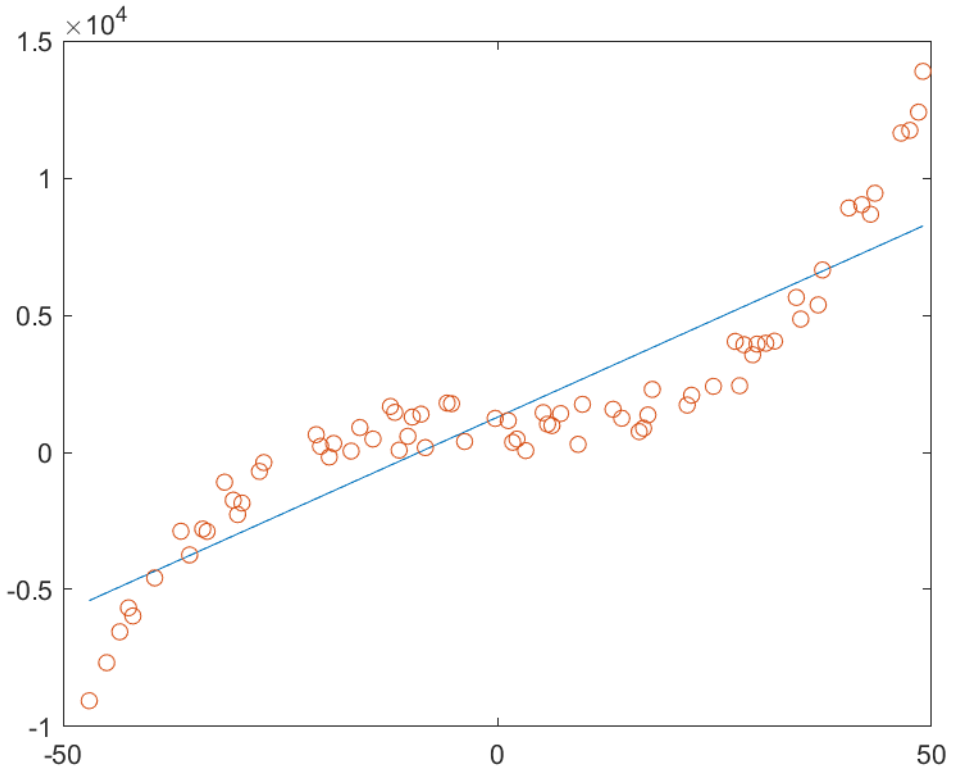
Test

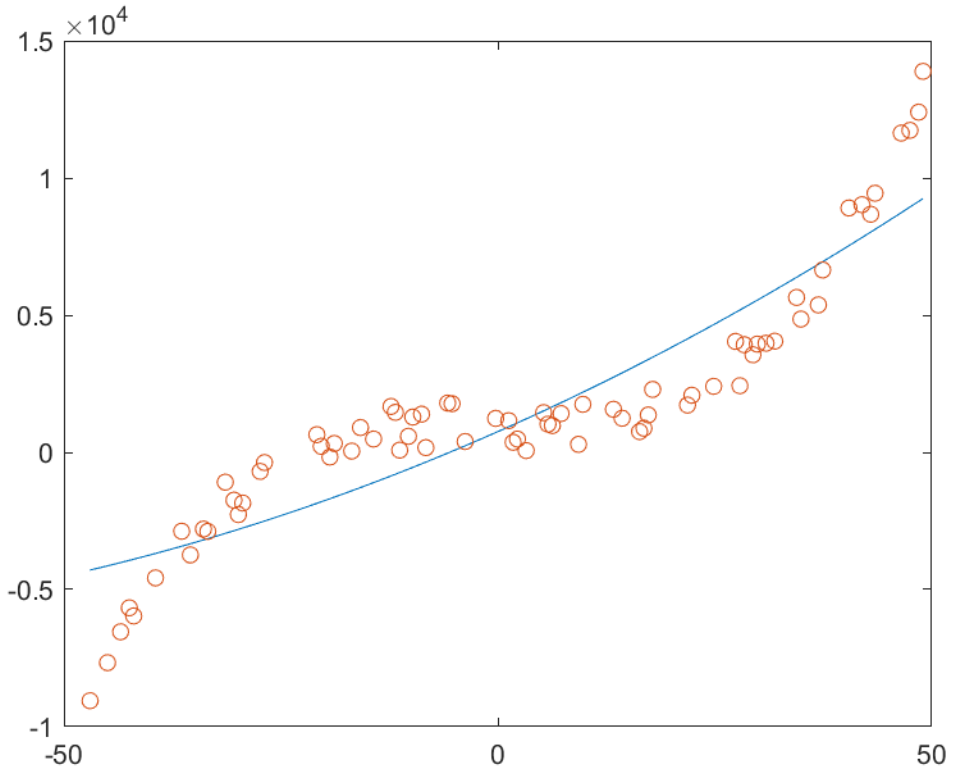


Hyperparameters Tuning

- Possible algorithm:
- Initialize a list of polynomials and a list of error values
- Iterate on grade between 1 and desired max value:
  - Calculate fitting of polynomial at current grade
  - Calculate output of fitted polynomial
  - Measure validation error
- Pick the polynomial with the lowest validation error
- Get a final and unbiased estimate of the model by measuring the error on a hold-out test set

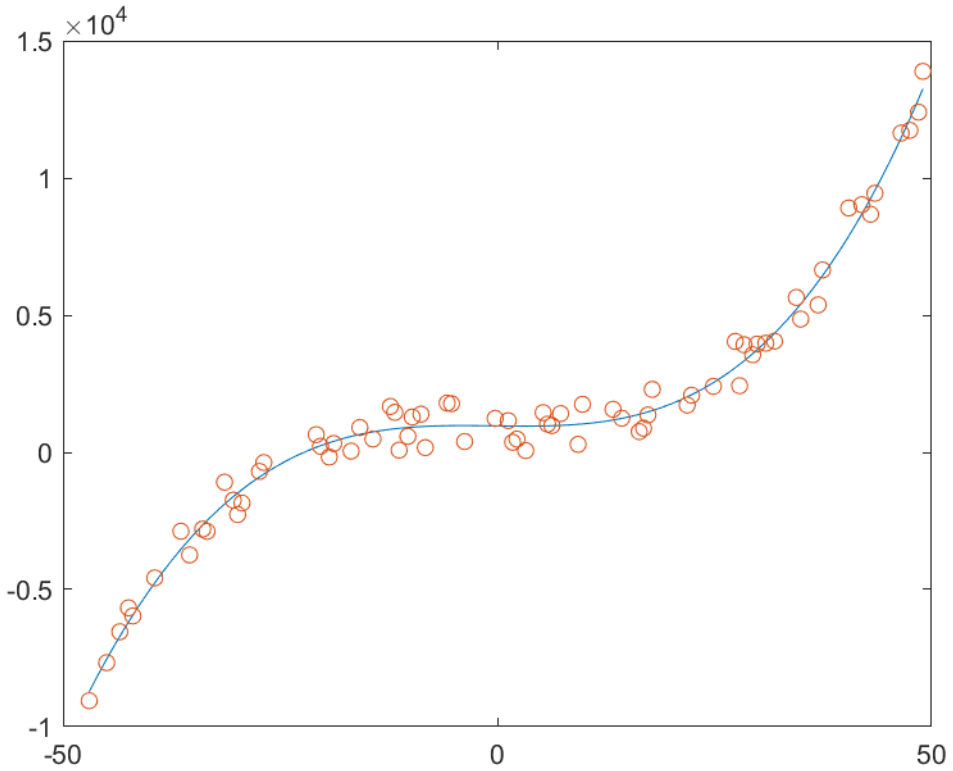
Hyperparameters Tuning

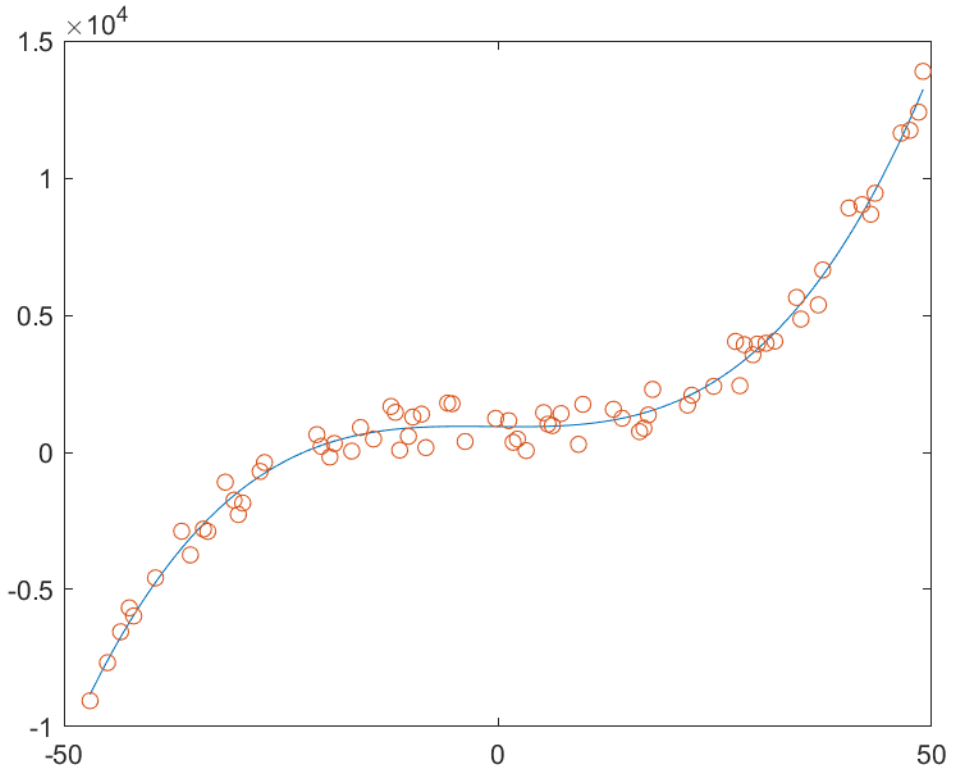




Polynomial of grade 1 or 2 have no sufficient capacity to handle the current dataset.

Hyperparameters Tuning





Polynomial of grade 3 has the right capacity to fit the data.
Polynomial of higher grades lead in overfitting and increased variance.
We prefer polynomial of grade 3 because it is the simplest model which can get proper fitting (**Occam's razor**).