UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Machine Learning
2024/2025

AMCO
ARTIFICIAL INTELLIGENCE, MACHINE
LEARNING AND CONTROL RESEARCH GROUP

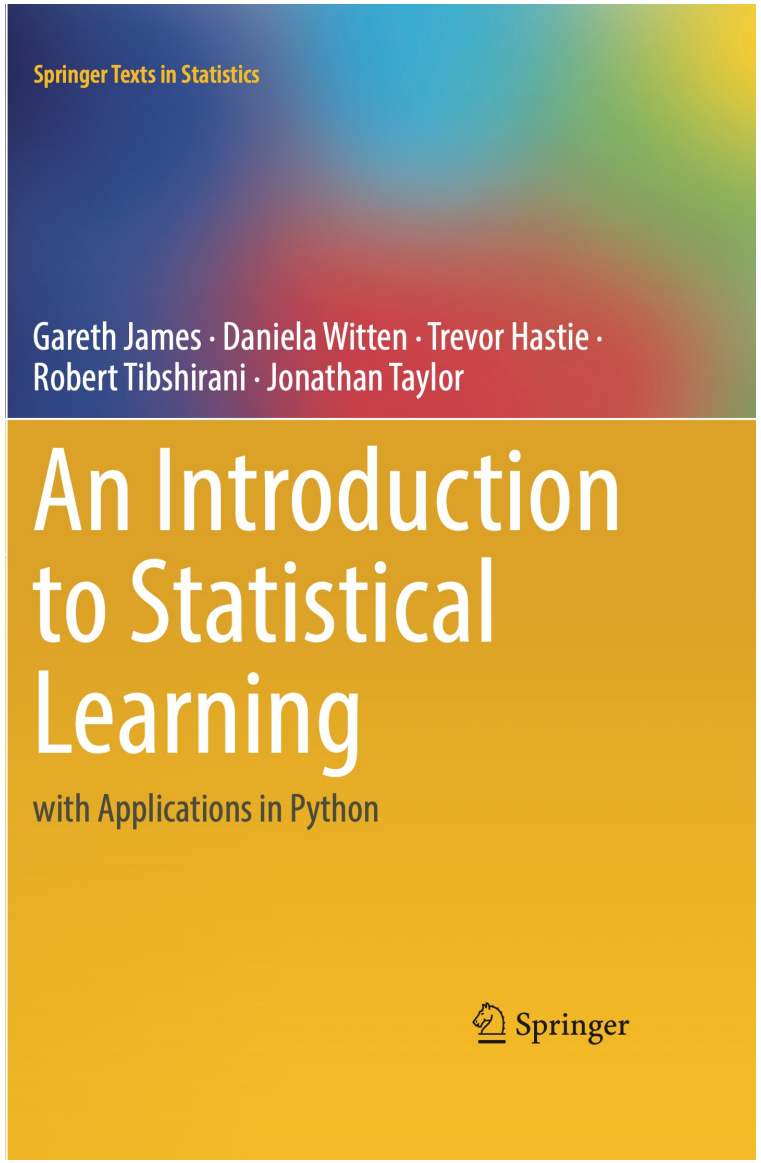
Lecture #07 Introduction to Supervised Learning and Linear Regression

Gian Antonio Susto

Before starting: Additional Material for Practical Training

Lab 2.3: intro to python (basic, numpy, graphics, data handling)
Lab 12.5: unsupervised learning (only 12.5.1 related to PCA)

Pay attention, the book uses some datasets that are available at
https://www.statlearning.com/resources-python



Recap – The Machine Learning pipeline

- Definition
- Expected Impact
- Evaluation metric

- Conversion
- Parsing
- Aggregation
- Alignment

- Quality
- Reconciliation
- Missing data handling
- Denoising
- Outlier detection

- Feature Extraction
- Building
- Evaluation/ Comparison

Modeling

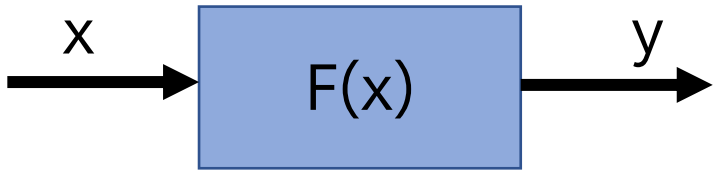
- On-line implementation
- Business outcome
- Improvement

Supervised Tasks

Setup: available historical data

Data: (x – ‘input’, y – ‘output’)

Objective: earn a map/function that, when fed with new 'x', provides output estimates of 'y'



Supervised Tasks

Depending on the nature of the output, we distinguish two subclasses of problems:

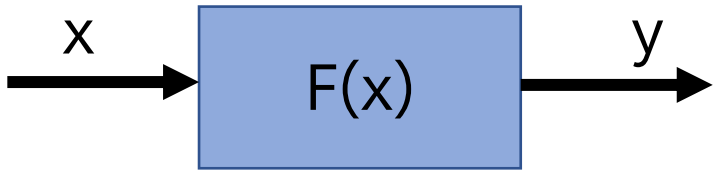
- If y is a continuous variable -> Regression Problem

- If y is a categorical variable -> Classification Problem

Setup: available historical data

Data: (x – 'input', y – 'output')

Objective: earn a map/function that, when fed with new 'x', provides output estimates of 'y'



An example of a Regression Task

- Goal: estimating the selling prize of an house [1]
- Thanks to an historical data of *n* transactions (for example the California housing dataset) with information such as
  - Median price house (output - *Y*)
  - # Rooms (input - *X*)
  - Squared meters (input - *X*)
  - Built year (input - *X*)
  - Address (input - *X*)

[1] *Machine Learning and the Spatial Structure of House Prices and Housing Returns* – A. Caplin et al.

www.immobiliare.it

An example of a Classification Task

- Goal: estimating the Iris type
- Thanks to an historical data of *n* data sample with information such as
  - Class (‘setosa’, ‘virginica’, ’versicolur’) (output - *Y*)
  - Sepal length (input - *X*)
  - Sepal width (input - *X*)
  - Petal length (input - *X*)
  - Petal width (input - *X*)

An example of a Classification Task

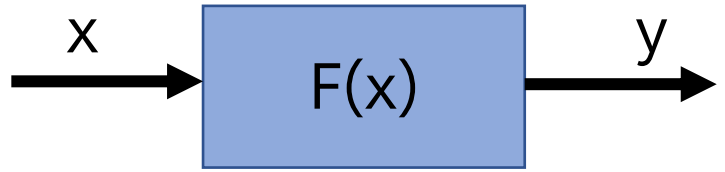
- Goal: recognizing a song from a small (3-4 sec.) data sample
- Currently handling a 100 million class problem
- Historically, first results on Shazam talked about a ‘digital footprint (X)’: mainly a feature engineering approach made the solution feasible!

Supervised Tasks

Setup: available historical data

Data: (x – 'input', y – 'output')

Objective: earn a map/function that, when fed with new 'x', provides output estimates of 'y'



- If y is a continuous variable -> Regression
- If y is a categorical variable -> Classification

Note #01: We will initially concentrate on Regression

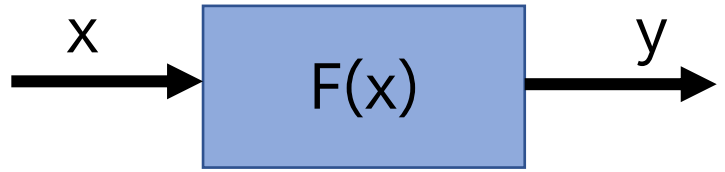
Note #02: With small changes, regression approaches can be adapted to Classification (and vice versa)

Supervised Tasks

Setup: available historical data

Data: (x – 'input', y – 'output')

Objective: earn a map/function that, when fed with new 'x', provides output estimates of 'y'



- If y is a continuous variable -> Regression
- If y is a categorical variable -> Classification

Note #01: We will initially concentrate on Regression

Note #02: With small changes, regression approaches can be adapted to Classification (and vice versa)

We'll talk a lot of supervised learning

WEEK 01
2025-02-24 Monday - Lecture 01: Course introduction & Motivation & Taxonomy of ML
2025-02-27 Thursday - Lecture 02: Introduction to Statistics for Machine Learning
2025-02-28 Friday - Lecture 03 (Lab 01): Introduction to Python

WEEK 02
2025-03-01 Monday - Lecture 04: Data Visualization
2025-03-04 Thursday - Lecture 05: Principal Component Analysis & Multivariate modeling
2025-03-05 Friday - Lecture 06 (Lab 02): Elaborate and Visualize data

WEEK 03
2025-03-10 Monday - Lecture 07: Supervised Learning, linear regression, training vs testing
2025-03-13 Thursday - Lecture 08: Overfitting and Ridge Regression, crossvalidation
2025-03-14 Friday - Lecture 09 (Lab 03): Linear Regression and Ridge Regression

WEEK 04
2025-03-17 Monday - Lecture 10: Ridge Regression vs LASSO, gradient descent
2025-03-20 Thursday - Lecture 11: Classification, Logistic Regression
2025-03-21 Friday - Lecture 12 (Lab 04): Regularization & Classification

We'll talk a lot of supervised learning

WEEK 05
2025-03-24 Monday - Lecture 13: Multiclass Classification and Softmax Regression, Introduction to performance metrics: accuracy, precision, recall, F1-score / Handling unbalanced data
2025-03-27 Thursday - Lecture 14: Decision trees, overfitting and pruning
2025-03-28 Friday - Lecture 15 (Lab 05): Decision Trees

WEEK 06
2025-03-31 Monday - Lecture 16: Ensemble Methods: Bagging, Random Forests, bootstrap aggregating
2025-04-03 Thursday - Lecture 17: AdaBoost, XGBoost, Catboost
2025-04-04 Friday - Lecture 18 (Lab 06): Ensemble approaches

WEEK 07
2025-03-17 Monday - Lecture 19: Support Vector Machines (SVM), Linear and kernel-based approaches, Concept of the margin and kernel trick
2025-04-10 Thursday - Lecture 20: Unsupervised Learning: K-Means Clustering. Evaluating clustering performance.
2025-04-11 Friday - Lecture 21 (Lab 07): SVM and Clustering

We'll talk a lot of supervised learning

WEEK 08
2025-04-14 Monday - Lecture 22: Anomaly Detection
2025-04-17 Thursday - Lecture 23: Introduction to Neural Networks, Activation functions (ReLU, sigmoid, softmax), Perceptrons

WEEK 09
2025-04-24 Thursday - Lecture 24: Training of NN #01

WEEK 10
2025-04-28 Monday - Lecture 25: Training of NN #02

WEEK 11
2025-05-05 Monday - Lecture 26: CNN
2025-05-08 Thursday - Lecture 27: Autoencoders
2025-05-09 Friday - Lecture 28 (Lab 08): NN #01

WEEK 12
2025-05-13 Monday - Lecture 29: RNN
2025-05-16 Friday - Lecture 30 (Lab 09): NN #02

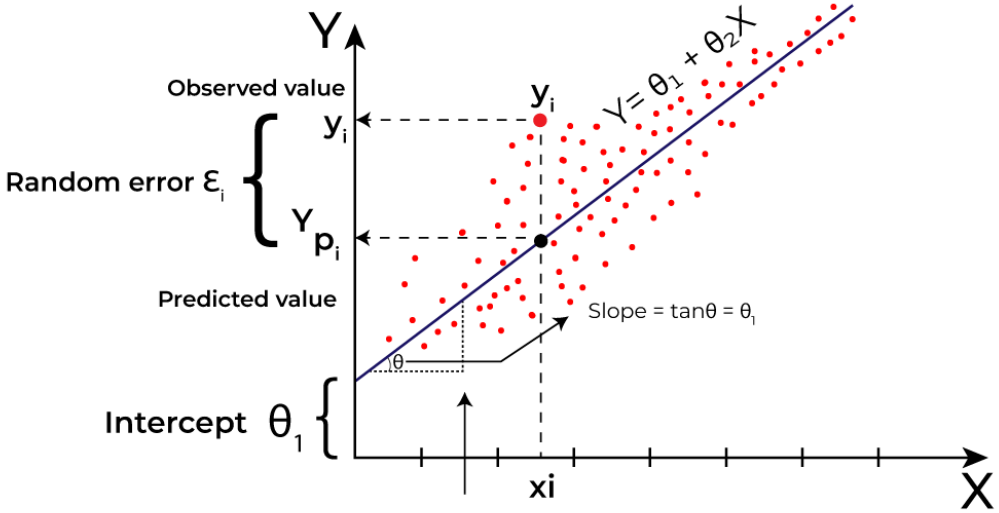
We’ll talk a lot of supervised learning

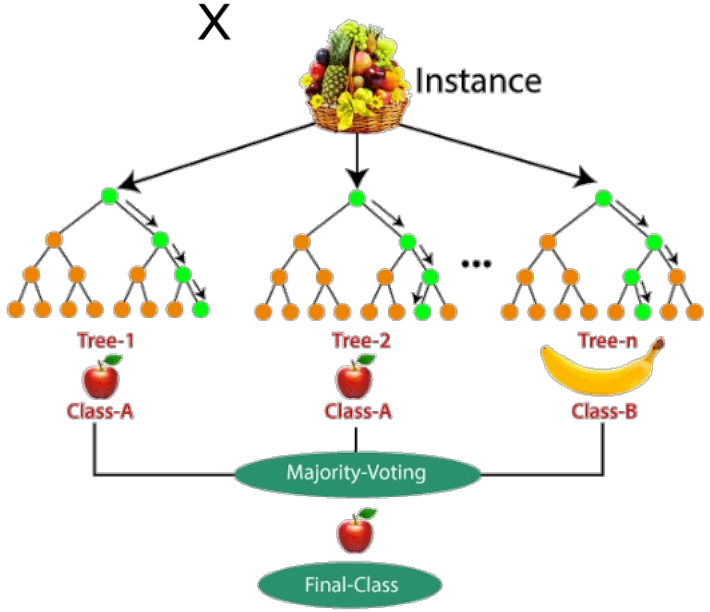
WEEK 14
2025-05-26 Monday - Lecture 31: XAI #01
2025-05-29 Thursday - Lecture 32: XAI #02
2025-05-30 Friday - Lecture 33 (Lab 10): XAI

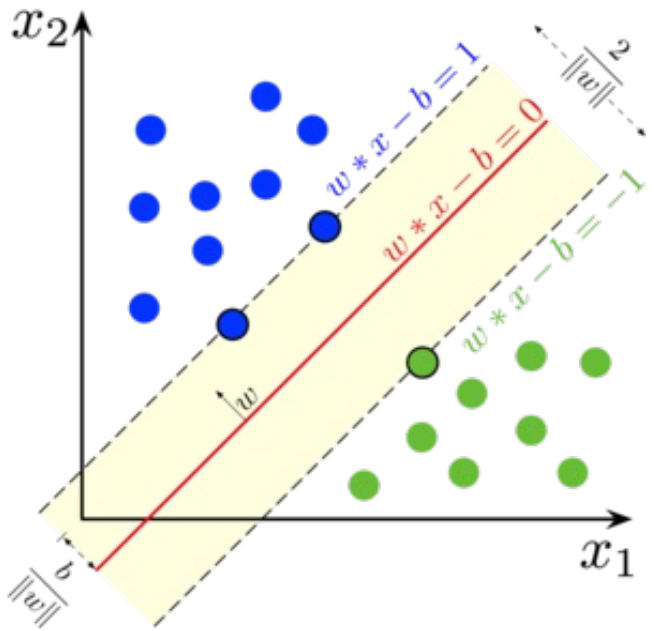
WEEK 15
2025-06-05 Thursday - Lecture 34: Fairness in ML
2025-06-06 Friday - Lecture 35: Real-world Applications and MLOps

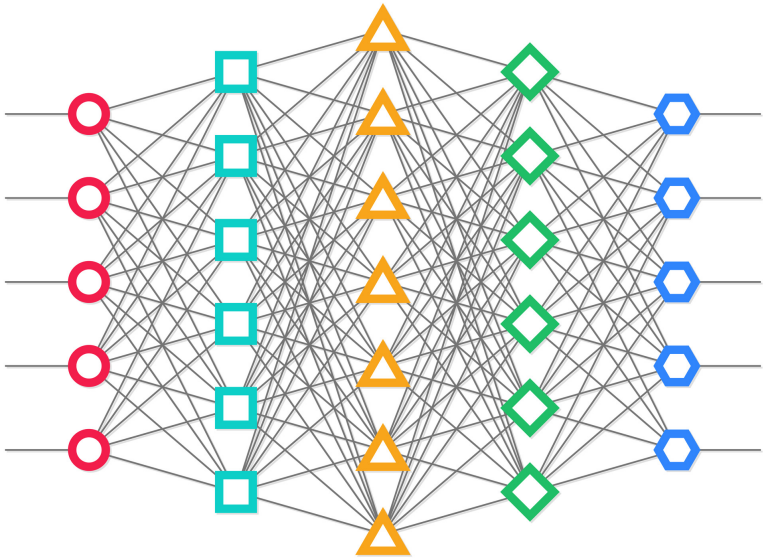
WEEK 16
2025-06-09 Monday - Lecture 36: What’s next

Regression Approaches

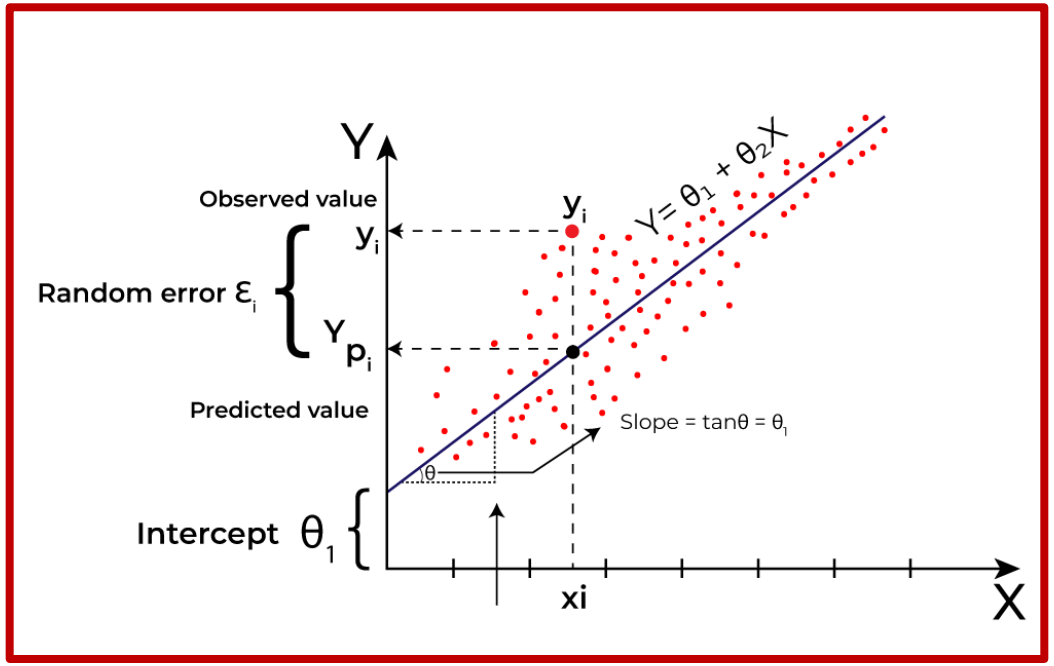


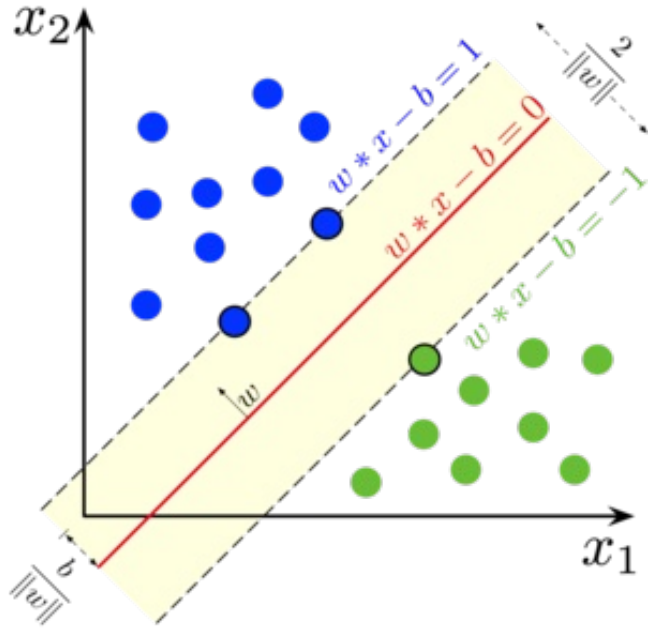


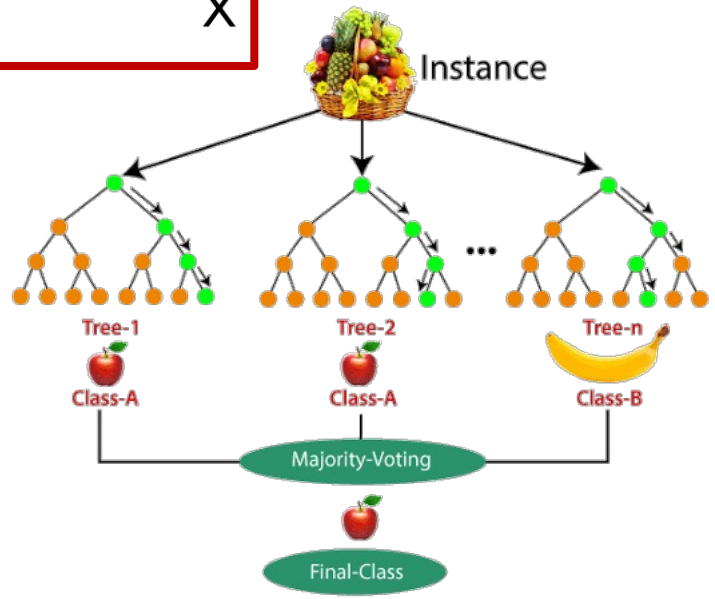


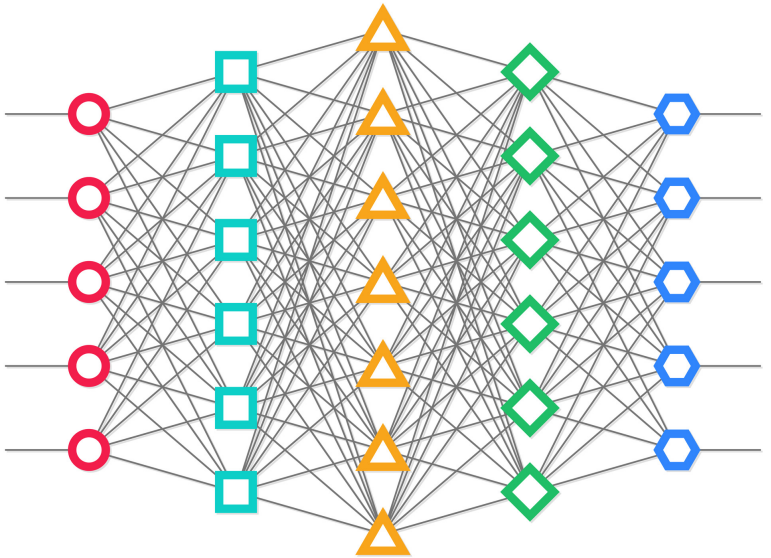


Regression Approaches

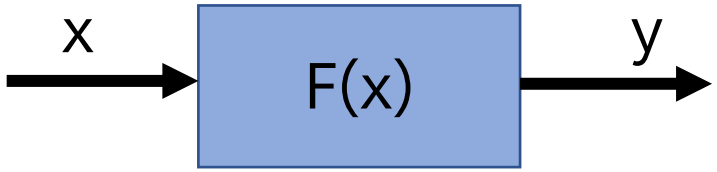








Linear Regression



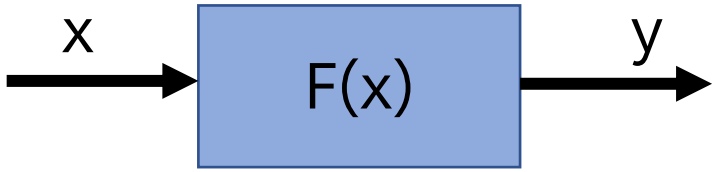
We are looking for a model *F(x)* that allows us to make predictions (estimates) of a target variable *y* based on the features $x = [x_1\ x_2\ \ldots\ x_p]$

We initially consider linear models, ie. models with the form:

$$F(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Ie. House Price = 150k$ + 10k$*[# bathrooms] + … - 1k$*[house age]

Linear Regression



We are looking for a model *F(x)* that allows us to make predictions (estimates) of a target variable *y* based on the features $x = [x_1\ x_2\ \dots\ x_p]$

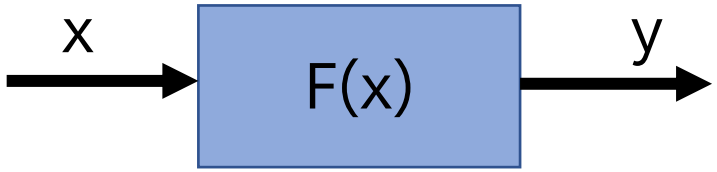
We initially consider linear models, ie. models with the form:

$F(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$

- These are called parameters/coefficients

Ie. House Price = 150k$ + 10k$*[# bathrooms] + … - 1k$*[house age]

Linear Regression



We are looking for a model *F(x)* that allows us to make predictions (estimates) of a target variable *y* based on the features $x = [x_1\ x_2 \dots x_p]$

We initially consider linear models, ie. models with the form:

$F(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$

- These are called parameters/coefficients
- One the parameter is the so-called intercept, the 'constant' coefficient

Ie. House Price = 150k$ + 10k$*[# bathrooms] + … - 1k$*[house age]

Linear Regression: Parameters Search and Objective Function

How to choose the ‘best’ parameters

$F_\beta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$?

Data will tell us!

Linear Regression: Parameters Search and Objective Function

How to choose the ‘best’ parameters

$F_\beta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$?

Data will tell us!

We need a criterium (an objective) to choose such parameters: we will choose the ‘best’ parameters to optimized our objective. Ideas?

The objective will be called ‘Cost function’

Linear Regression: Parameters Search and Objective Function

How to choose the 'best' parameters

$F_\beta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$?

Data will tell us!

We need a criterium (an objective) to choose such parameters: we will choose the 'best' parameters to optimized our objective. Ideas?

A common objective is to minimize the sum of (squared) prediction errors, ie. I will choose β s.t. $\frac{\sum_{i=1}^{n}[y^{(i)} - F_\beta(x^{(i)})]^2}{n}$ is the possible lowest value!

Linear Regression: Parameters Search and Objective Function

How to choose the 'best' parameters

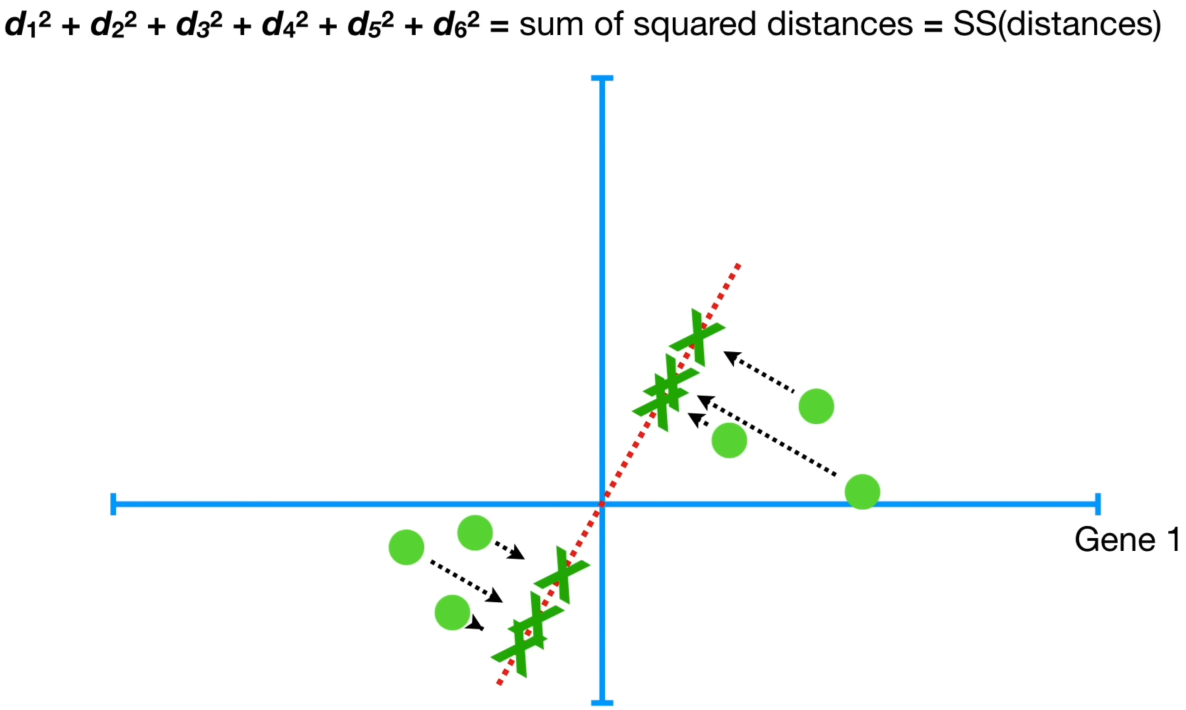
$F_\beta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$?

Data will tell us!

We need a criterium (an objective) to choose such parameters: we will choose the 'best' parameters to optimized our objective. Ideas?

A common objective is to minimize the sum of (squared) prediction errors, ie. I will choose β s.t. $\frac{\sum_{i=1}^{n}[y^{(i)} - F_\beta(x^{(i)})]^2}{n}$ is the possible lowest value!

We have already seen something similar in PCA



Linear Regression: Parameters Search and Objective Function

How to choose the 'best' parameters

$F_\beta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$?

Data will tell us!

We need a criterium (an objective) to choose such parameters: we will choose the 'best' parameters to optimized our objective. Ideas?

A common objective is to minimize the sum of (squared) prediction errors, ie. I will choose β s.t. $\frac{\sum_{i=1}^{n}[y^{(i)} - F_\beta(x^{(i)})]^2}{n}$ is the possible lowest value!

This metric is called Mean Squared Error (MSE), and it is widely used in regression.

The Root Mean Squared Error $\mathrm{RMSE} = \sqrt{MSE}$ is also widely used

We will shortly see why we use the 'squared' version of the error

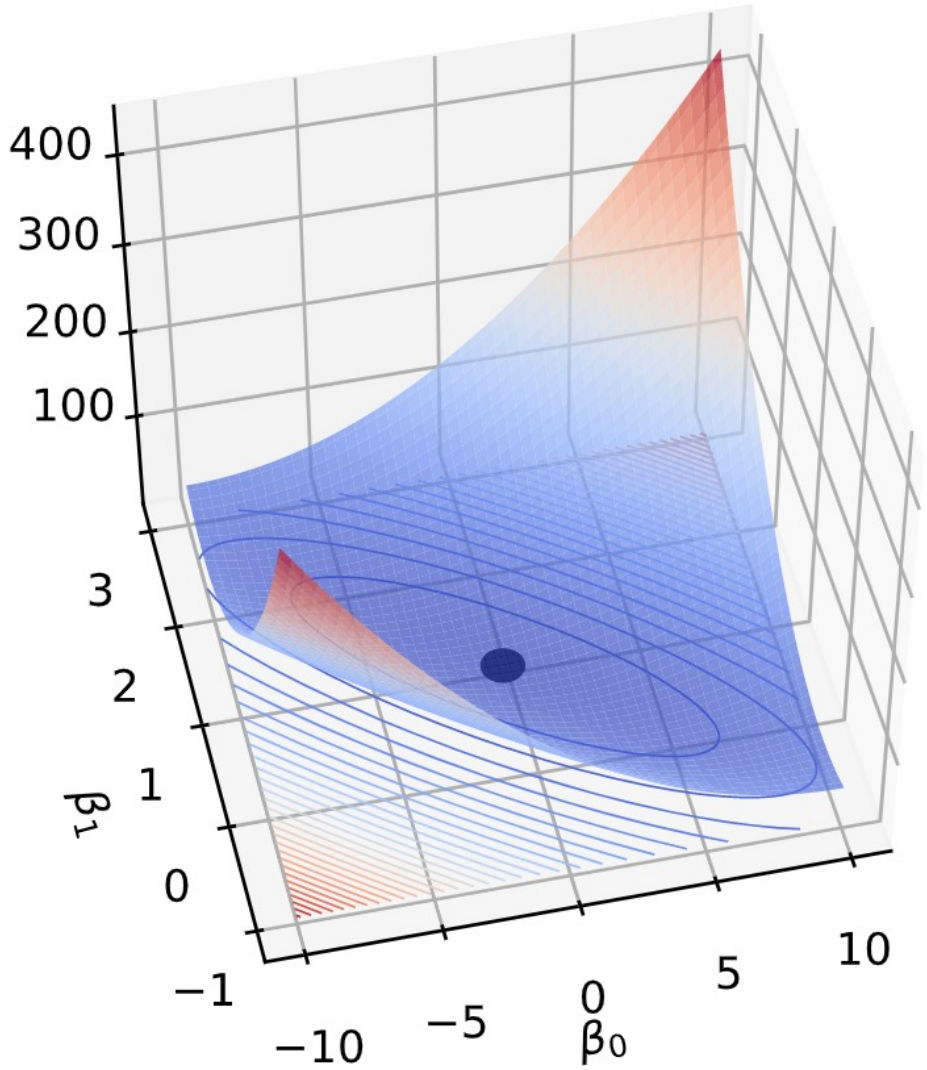
Linear Regression: Parameters Search and Objective Function

At the change of parameters combinations, different MSE are achieved (our predictions will become better or worse)

A nice property of regression approaches is that the cost function is **convex** in the space of the parameters (a line segment connecting any two points on the function's graph never lies below the function itself): this will become extremely useful later!

the sum of (squared) prediction errors,

ie. I will choose β s.t. $\frac{\sum_{i=1}^{n}[y^{(i)} - F_{\beta}(x^{(i)})]^2}{n}$ is the possible lowest value!



Linear Regression: Parameters Search and Objective Function

How to ch

$F_\beta(x) = \beta_0 +$

Data will te

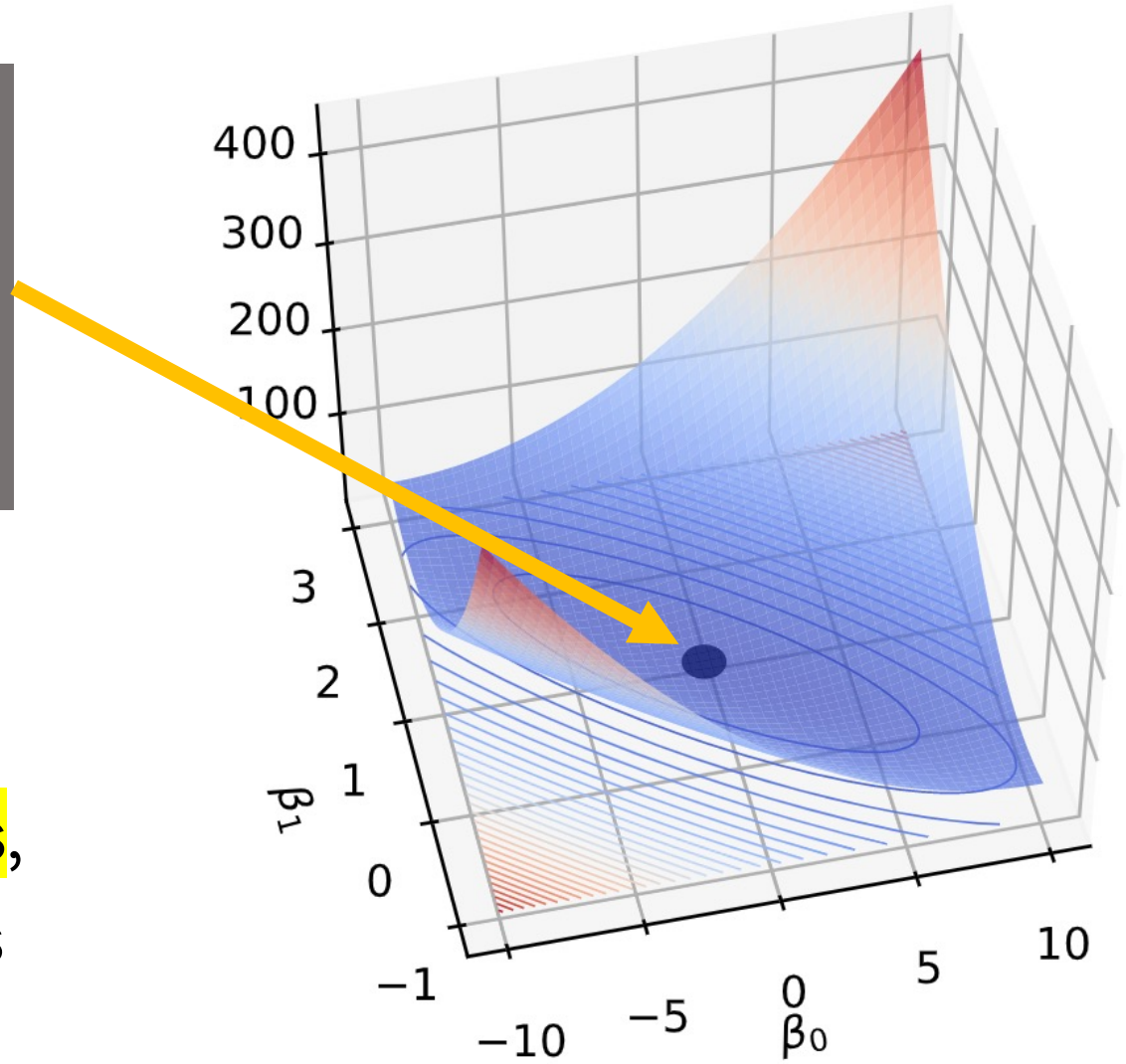
We need a
choose such parameters: we will
choose the 'best' parameters to
optimized our objective. Ideas?

We have a unique set of 'optimal' parameters that minimizes the cost function

We indicate with β* the set of 'optimal' parameters

A common objective is to minimize the sum of (squared) prediction errors,

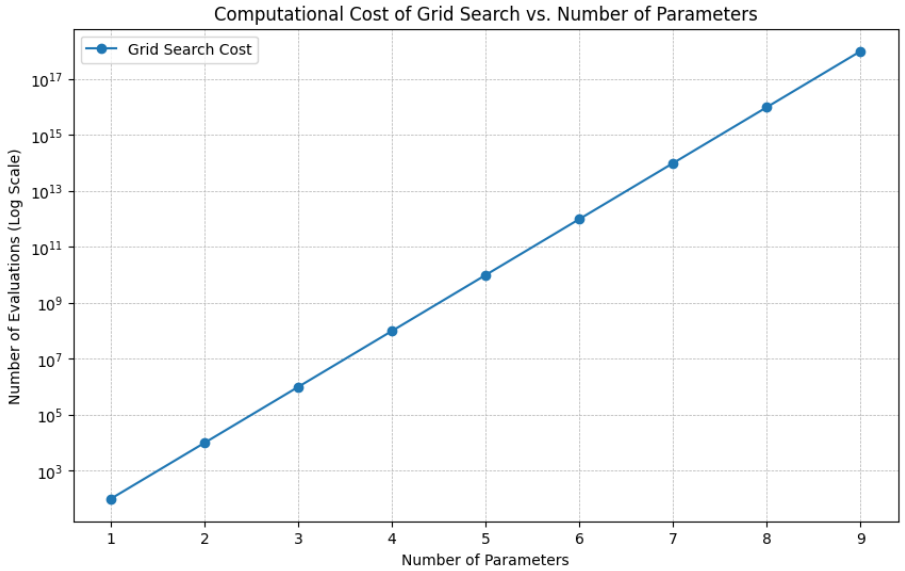
ie. I will choose β s.t. $\frac{\sum_{i=1}^{n}[y^{(i)} - F_\beta(x^{(i)})]^2}{n}$ is the possible lowest value!



Linear Regression: Parameters Search and Objective Function – Ideas?

Linear Regression: Parameters Search and Objective Function

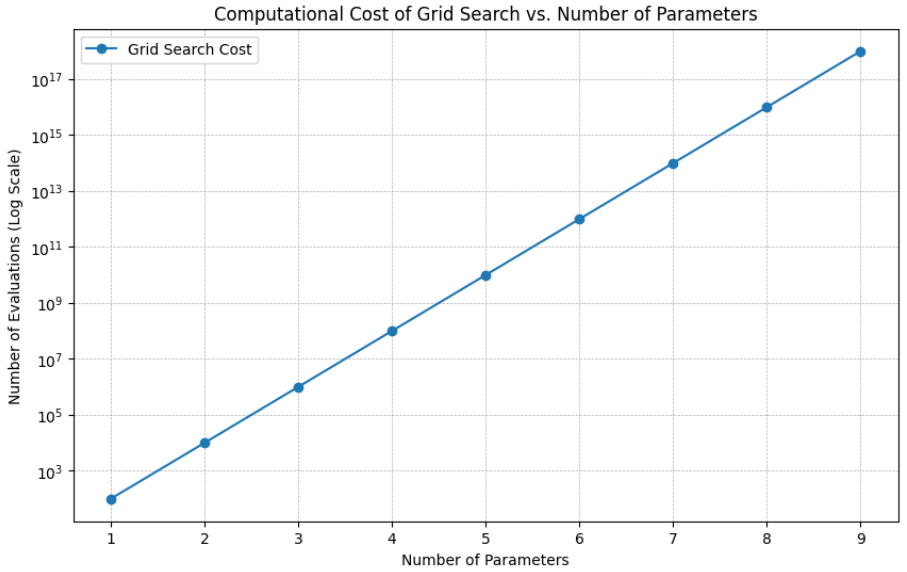
An approach called ’grid search’ consists in trying all possible combination of parameters β and then choose β* that minimizes the cost function

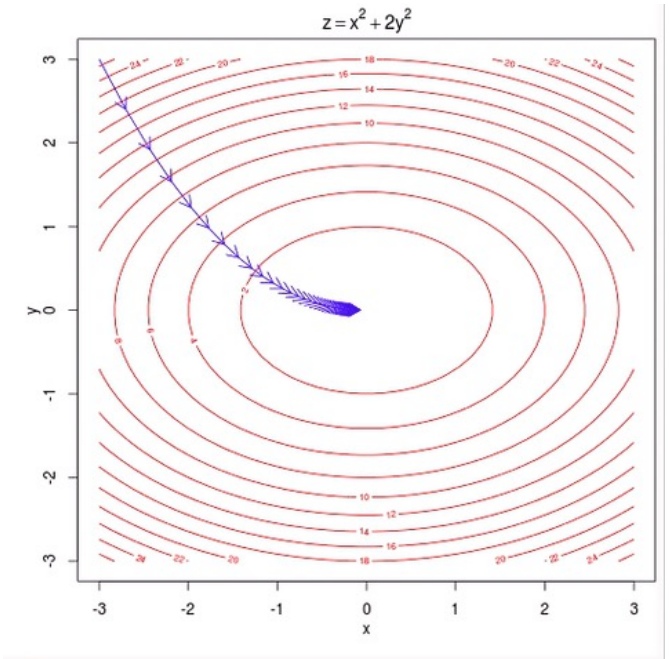


Linear Regression: Parameters Search and Objective Function

An approach called 'grid search' consists in trying all possible combination of parameters β and then choose β* that minimizes the cost function

This is a costly approach; instead in ML we use 'optimization' approaches that will find such parameters in a 'fast' way





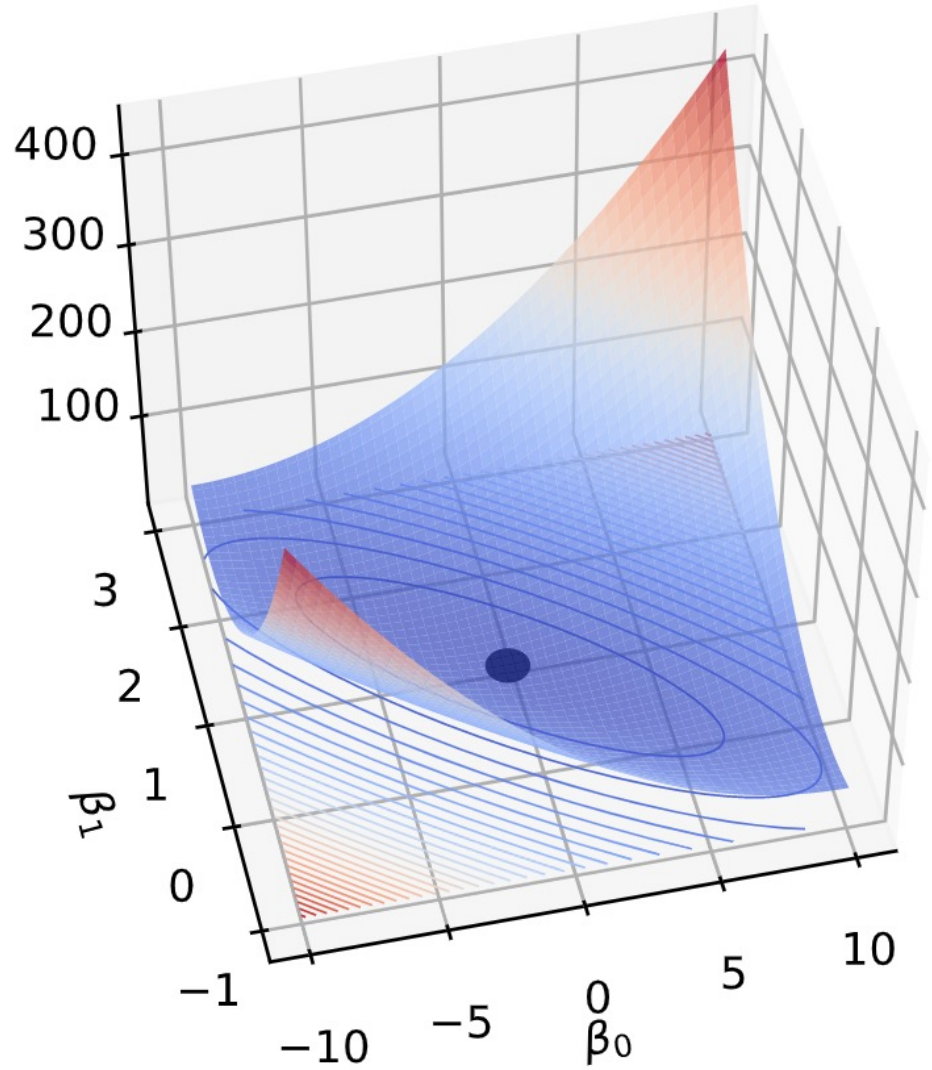
Linear regression: a closed-form solution (1/3)

Linear regression has a really simple way to find the optimal parameters: a closed-form solution!

We need to minimize this cost function, by smart choices of the parameters

$$J(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$
$$= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

How to do that?



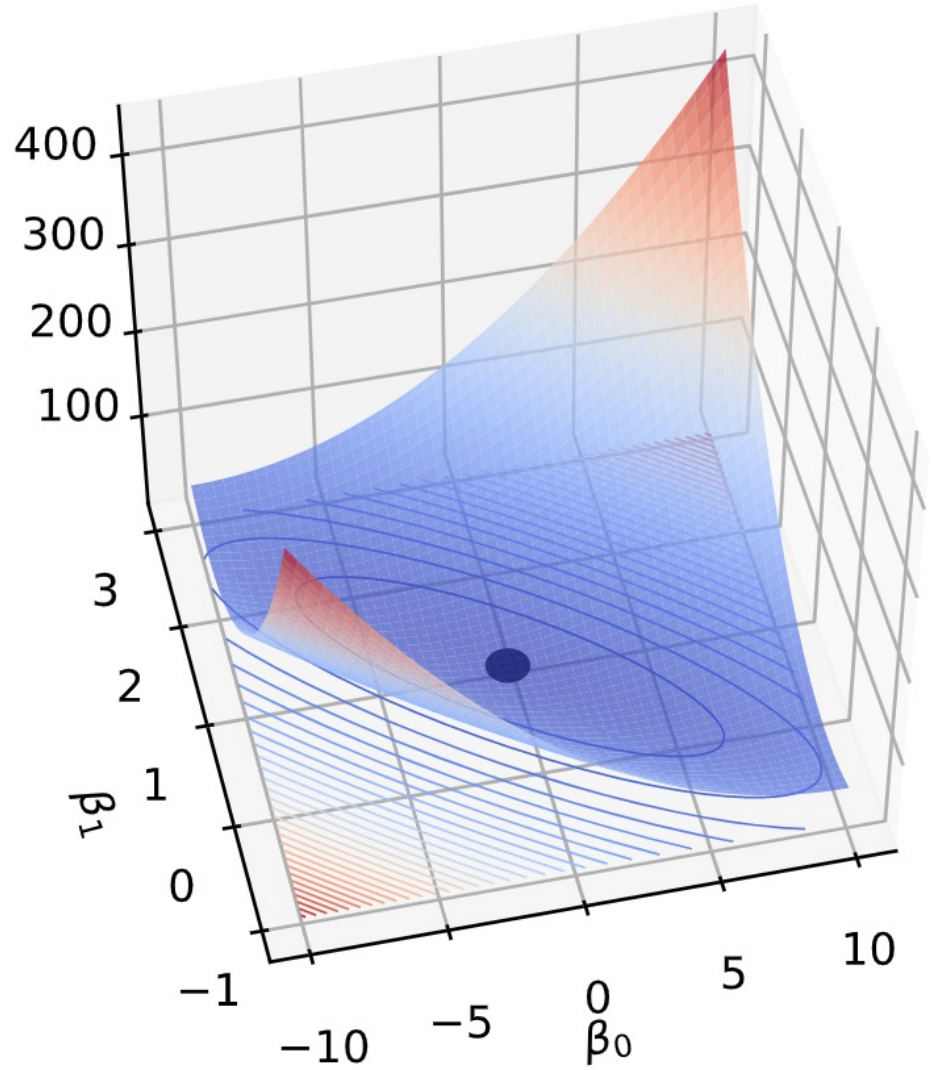
Linear regression: a closed-form solution (1/3)

Linear regression has a really simple way to find the optimal parameters: a closed-form solution!

We need to minimize this cost function, by smart choices of the parameters

$$J(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$
$$= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

How to do that? Let's compute the derivative of J w.r.t. the coefficients!



Linear regression: a closed-form solution (1/3)

Linear regression has a way to find the optimal closed-form solution!

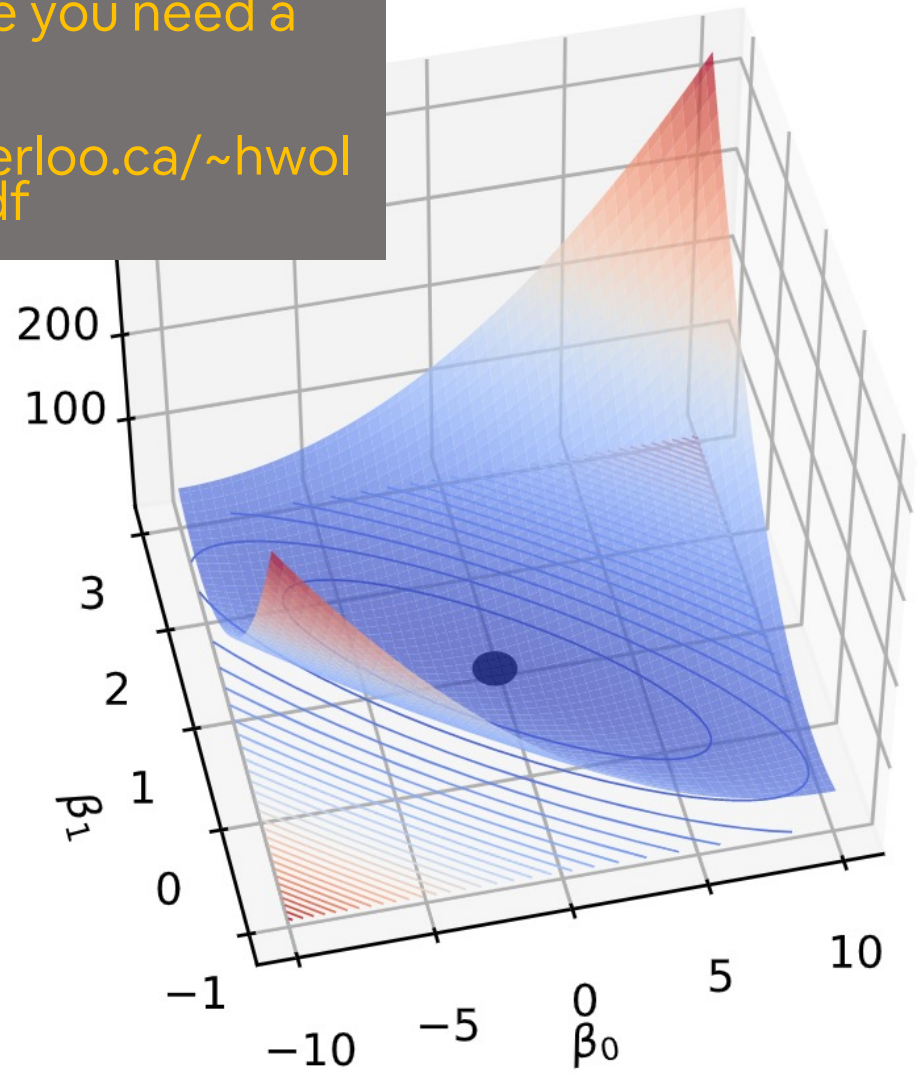
In the following we will use matrix manipulations: in the case you need a refresh, refer to:

https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

We need to minimize this cost function, by smart choices of the parameters

$$J(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$
$$= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

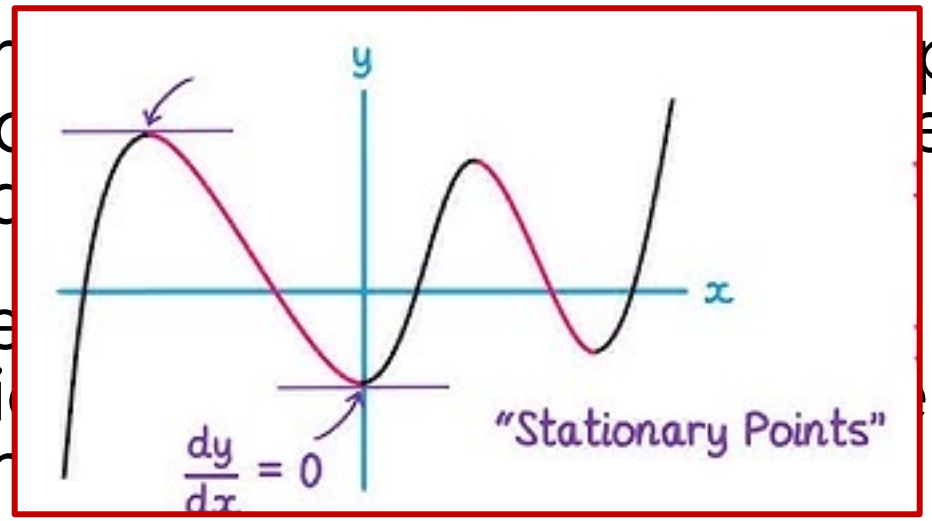
How to do that? Let's compute the derivative of J w.r.t. the coefficients!



Linear regression: a closed-form solution (1/3)

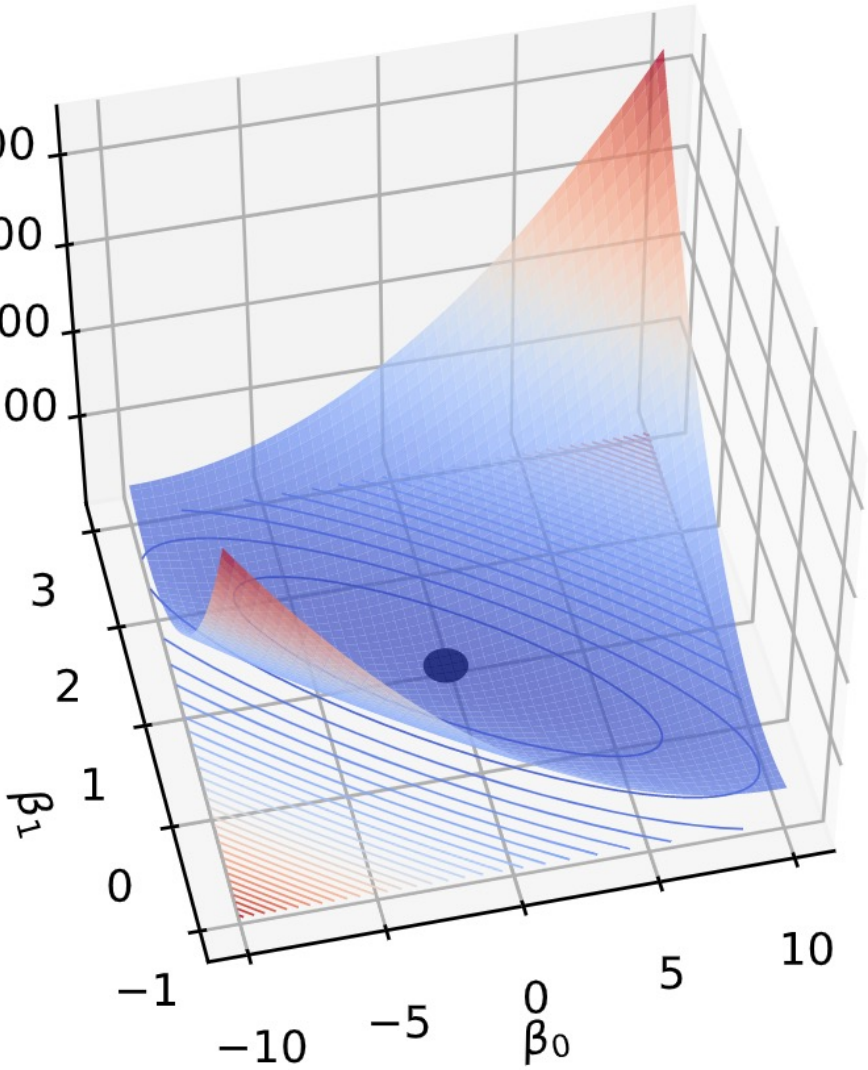
Linear ple
way to ers: a
closed

We ne
functi
param



$$J(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$
$$= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

How to do that? Let's compute the derivative of J w.r.t. the coefficients!



Linear regression: a closed-form solution (2/3)

$$J(\boldsymbol{\beta}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}$$

Linear regression: a closed-form solution (2/3)

$$J(\beta) = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X}\beta$$

$$\frac{\partial}{\partial\beta}\mathbf{y}^T\mathbf{y} = 0$$

Linear regression: a closed-form solution (2/3)

$$J(\boldsymbol{\beta}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}$$

$$\frac{\partial}{\partial \boldsymbol{\beta}}(-2\mathbf{y}^T\mathbf{X}\boldsymbol{\beta})$$

$$\frac{\partial}{\partial \boldsymbol{\beta}}\mathbf{y}^T\mathbf{y} = 0$$

$$\frac{\partial}{\partial \boldsymbol{\beta}}(\mathbf{a}^T\mathbf{b}) = \mathbf{a}$$

$$-2\mathbf{X}^T\mathbf{y}$$

Linear regression: a closed-form solution (2/3)

$$J(\beta) = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X}\beta$$

$$\frac{\partial}{\partial\beta}(-2\mathbf{y}^T\mathbf{X}\beta)$$

$$\frac{\partial}{\partial\beta}\mathbf{y}^T\mathbf{y} = 0$$

$$\frac{\partial}{\partial\beta}(\beta^T\mathbf{X}^T\mathbf{X}\beta)$$

$$\frac{\partial}{\partial\beta}(\mathbf{a}^T\mathbf{b}) = \mathbf{a}$$

$$\frac{\partial}{\partial\beta}(\beta^T\mathbf{A}\beta) = 2\mathbf{A}\beta, \quad \text{(if } \mathbf{A} \text{ is symmetric)}$$

$$-2\mathbf{X}^T\mathbf{y}$$

$$2\mathbf{X}^T\mathbf{X}\beta$$

Linear regression: a closed-form solution (3/3)

$$2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\mathbf{X}^T\mathbf{y} = 0$$

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}$$

Linear regression: a closed-form solution (3/3)

$$2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\mathbf{X}^T\mathbf{y} = 0$$

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}$$

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Coefficients are derived from data! We see that both input and output data are necessary to derive the coefficients.

The equation reported above is the closed-form solution for Ordinary Least Squares (OLS) regression.

Linear regression: a closed-form solution (3/3)

$$2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\mathbf{X}^T\mathbf{y} = 0$$

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}$$

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

We are assuming X'X is invertible for the moment

Coefficients are derived from data! We see that both input and output data are necessary to derive the coefficients.

The equation reported above is the closed-form solution for Ordinary Least Squares (OLS) regression.

Linear regression: a closed-form solution (3/3)

$$2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\mathbf{X}^T\mathbf{y} = 0$$

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}$$

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

There is also a model called 'Partial Least Squares' (PLS): we are probably not going to see this in the course

We are assuming X'X is invertible for the moment

Coefficients are derived from data! We see that both input and output data are necessary to derive the coefficients.

The equation reported above is the closed-form solution for Ordinary Least Squares (OLS) regression.

Example on California Housing dataset

Target (Y):

- MedHouseVal (Median House Value in block group) Represents the median house price in the block group. Measured in hundreds of thousands of dollars (capped at $500,000 in the dataset).

Inputs:*

- MedInc (Median Income in block group). Represents the median income of households in the block group. Measured in tens of thousands of dollars.
- HouseAge (Median House Age in block group). Represents the median age of houses in the area. Measured in years.
- AveRooms (Average Number of Rooms per Dwelling). Computed as the total number of rooms in the block group divided by the number of households. Helps indicate the general size of homes in an area.
- AveOccup (Average Number of Occupants per Household). Computed as the total population in the block group divided by the number of households.

* A subset of the available inputs

Example on California Housing dataset

Target (Y):

- MedHouseVal (Median House Value in block group) Represents the median house price in the block group. Measured in hundreds of thousands of dollars (capped at $500,000 in the dataset).

Inputs:

- MedInc (Median Income in block group). Represents the median income of households in the block group. Measured in tens of thousands of dollars.
- HouseAge (Median House Age in block group). Represents the median age of houses in the area. Measured in years.
- AveRooms (Average Number of Rooms per Dwelling). Computed as the total number of rooms in the block group divided by the number of households. Helps indicate the general size of homes in an area.
- AveOccup (Average Number of Occupants per Household). Computed as the total population in the block group divided by the number of households.

```
                    coef
--------------------------
const             0.0314
MedInc            0.4433
HouseAge          0.0169
AveRooms         -0.0273
AveOccup         -0.0045
```

```
RMSE on training data:
```

```
0.8046838844313648
```

Example on California Housing dataset

Target (Y):

- MedHouseVal (Median House Value in block group) Represents the median house price in the block group. Measured in hundreds of thousands of dollars (capped at $500,000 in the dataset).

Inputs:

- MedInc (Median Income in block group). Represents the median income of households in the block group. Measured in tens of thousands of dollars.
- HouseAge (Median House Age in block group). Represents the median age of houses in the area. Measured in years.
- AveRooms (Average Number of Rooms per Dwelling). Computed as the total number of rooms in the block group divided by the number of households. Helps indicate the general size of homes in an area.
- AveOccup (Average Number of Occupants per Household). Computed as the total population in the block group divided by the number of households.

	coef
const	0.0314
MedInc	0.4433
HouseAge	0.0169
AveRooms	-0.0273
AveOccup	-0.0045

```
RMSE on training data:
```

```
0.8046838844313648
```

Is it 'good'?

R-Squared

R-squared (R^2), or the Coefficient of Determination, is a statistical measure that indicates how well a regression model explains the variance in the target variable. Where:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

- Residual Sum of Squares, measures the total squared difference between the actual and predicted values

$$SS_{res} = \sum (y_i - \hat{y}_i)^2$$

- Total Sum of Squares, measures the total variance in the target variable

$$SS_{tot} = \sum (y_i - \bar{y})^2$$

Interpretation:

- R-squared = 1, perfect fit!
- R-squared = 0, the model does no better than simply predicting the mean
- R-squared < 0, the model performs worse than a simple average prediction

R-Squared

R-squared (R^2), or the Coefficient of Determination, is a statistical measure that indicates how well a regression model explains the variance in the target variable. Where:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

- Residual Sum of Squares, measures the total squared difference between the actual and predicted values

$$SS_{res} = \sum (y_i - \hat{y}_i)^2$$

- Total Sum of Squares, measures the total variance in the target variable

$$SS_{tot} = \sum (y_i - \bar{y})^2$$

Interpretation:

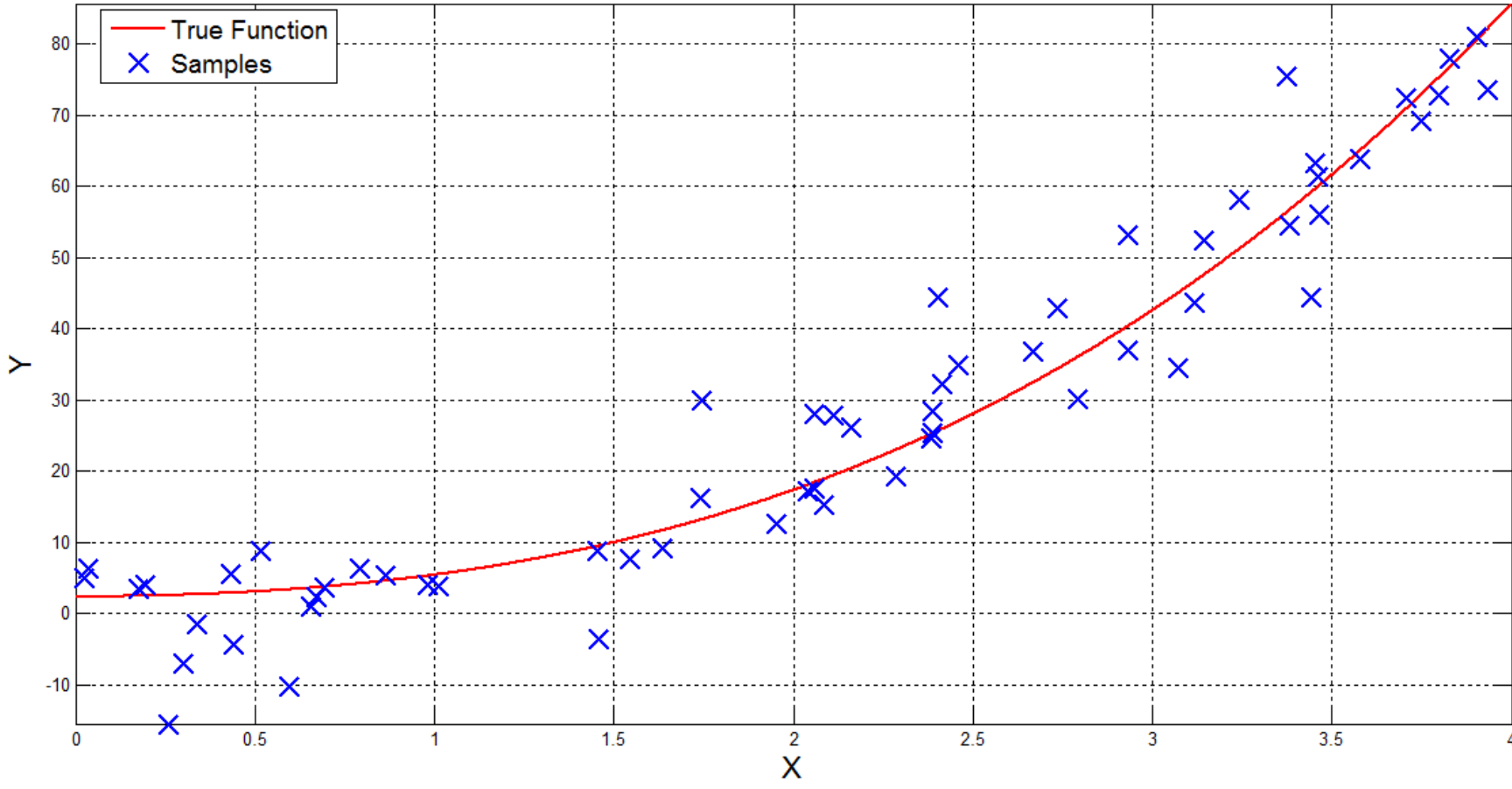
- R-squared = 1, perfect fit!
- R-squared = 0, the model does no better than simply predicting the mean
- R-squared < 0, the model performs worse than a simple average prediction

```
R-squared:
```

```
0.514
```

Minimization of MSE on training data

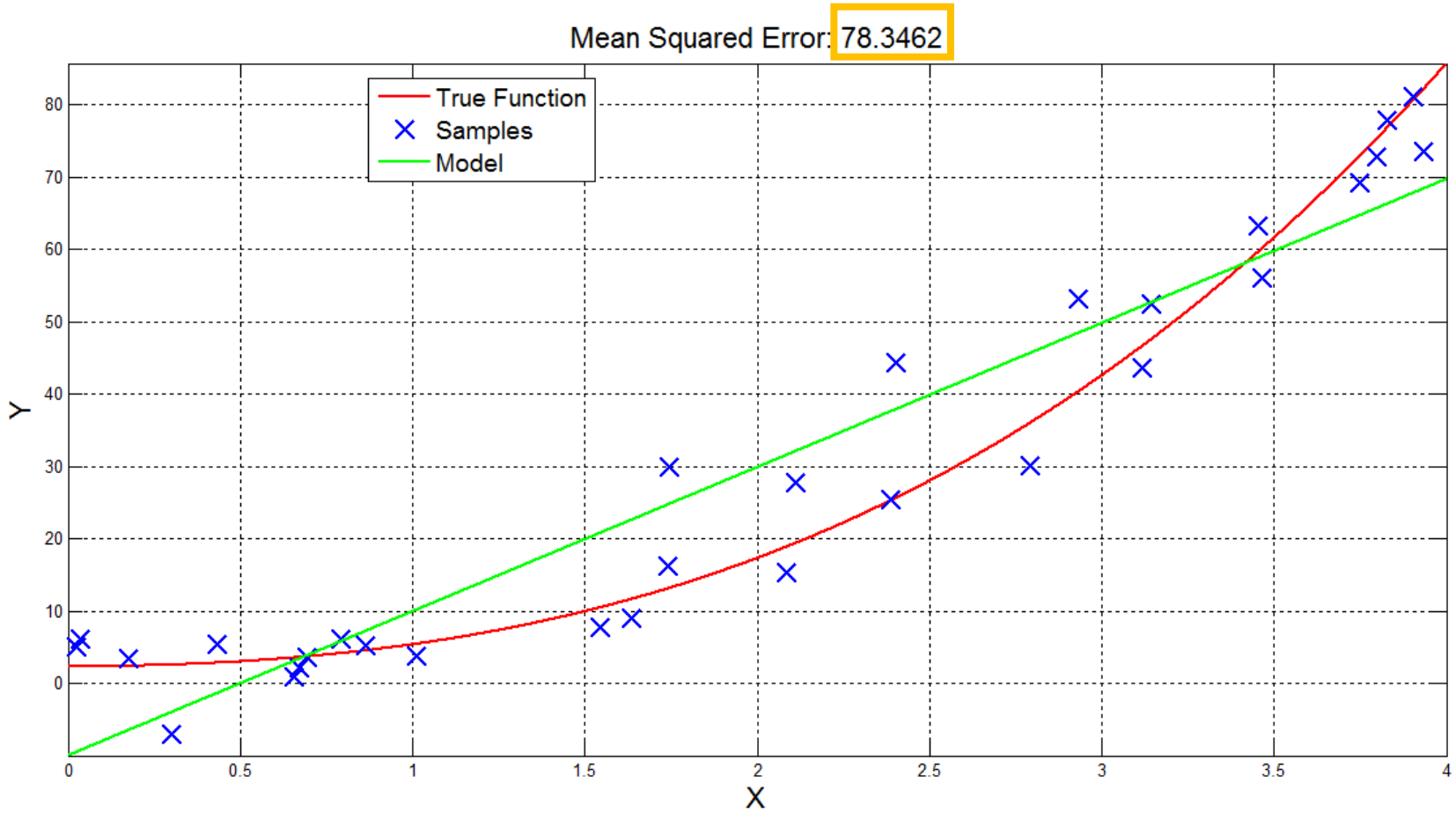
Let’s consider a univariate example:



Minimization of MSE on training data

OLS: linear coefficients to variable X and to the intercept (constant):

$Y = a+bX$

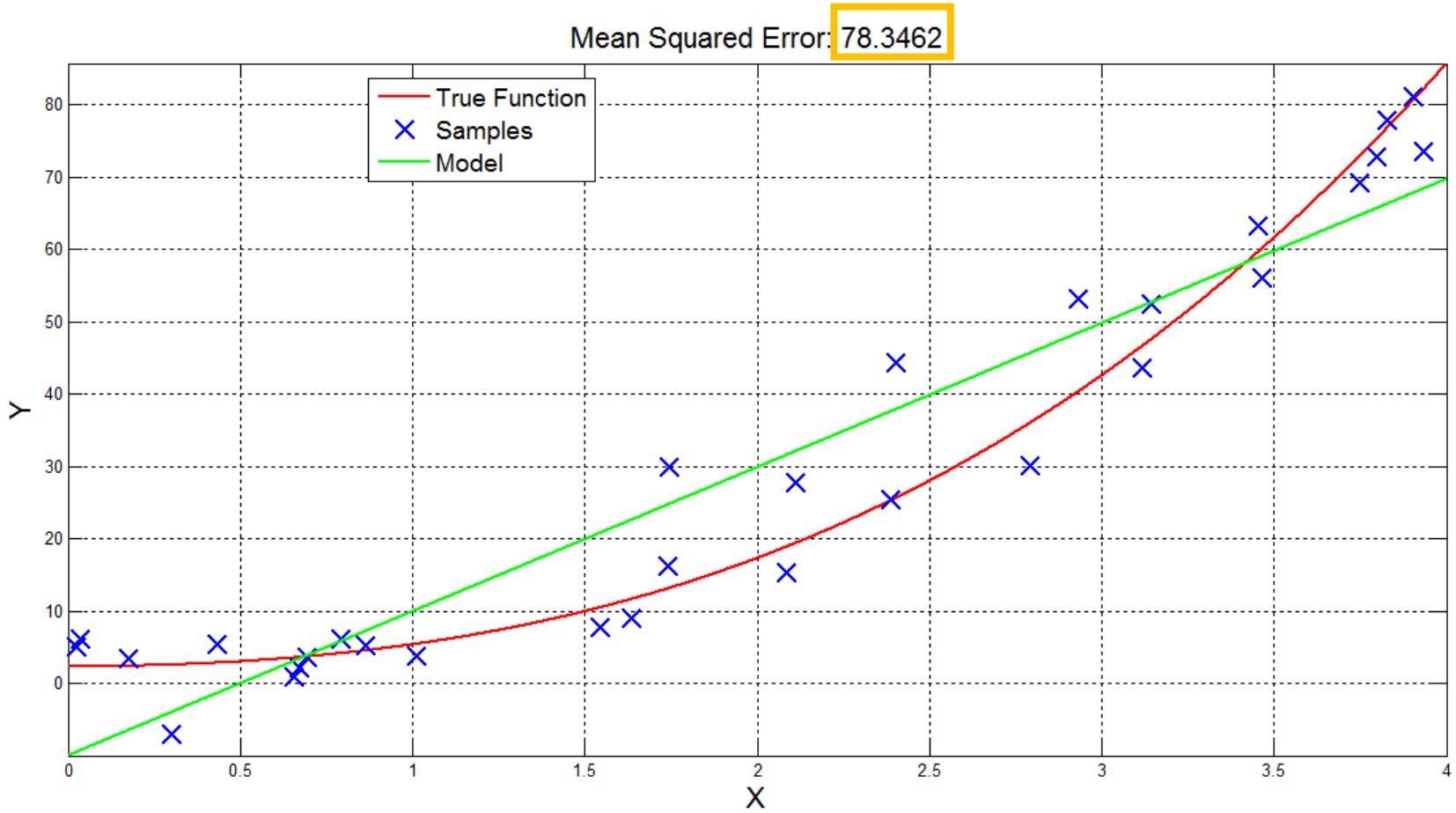


Minimization of MSE on training data

OLS: linear coefficients to variable X and to the intercept (constant):

$Y = a+bX$

Can we do better?

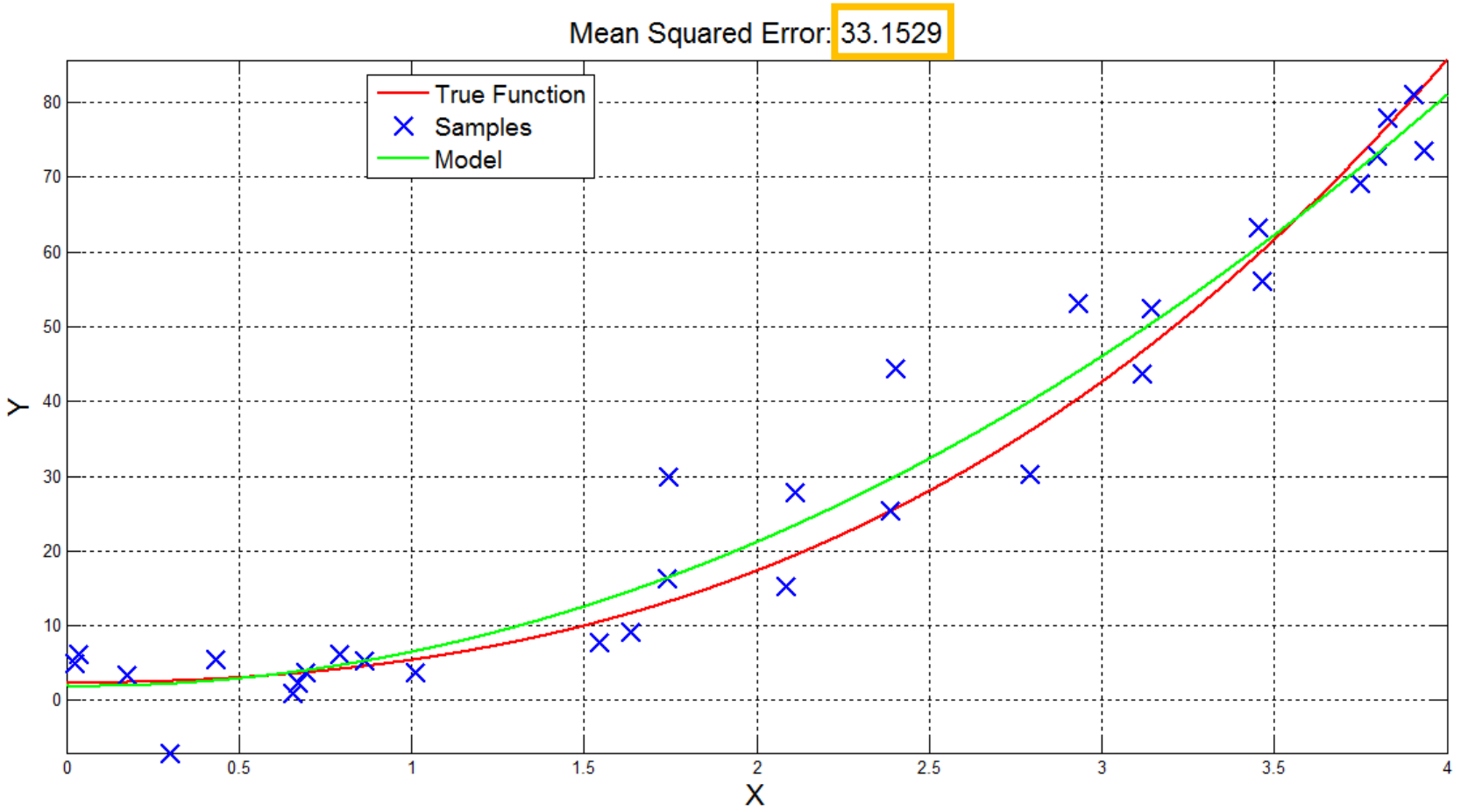


Minimization of MSE on training data

Let's also consider a transformation of the input data!

$Y = a+bX+cX^2$

This approach is called 'basis expansion' and it is a popular feature engineering step!

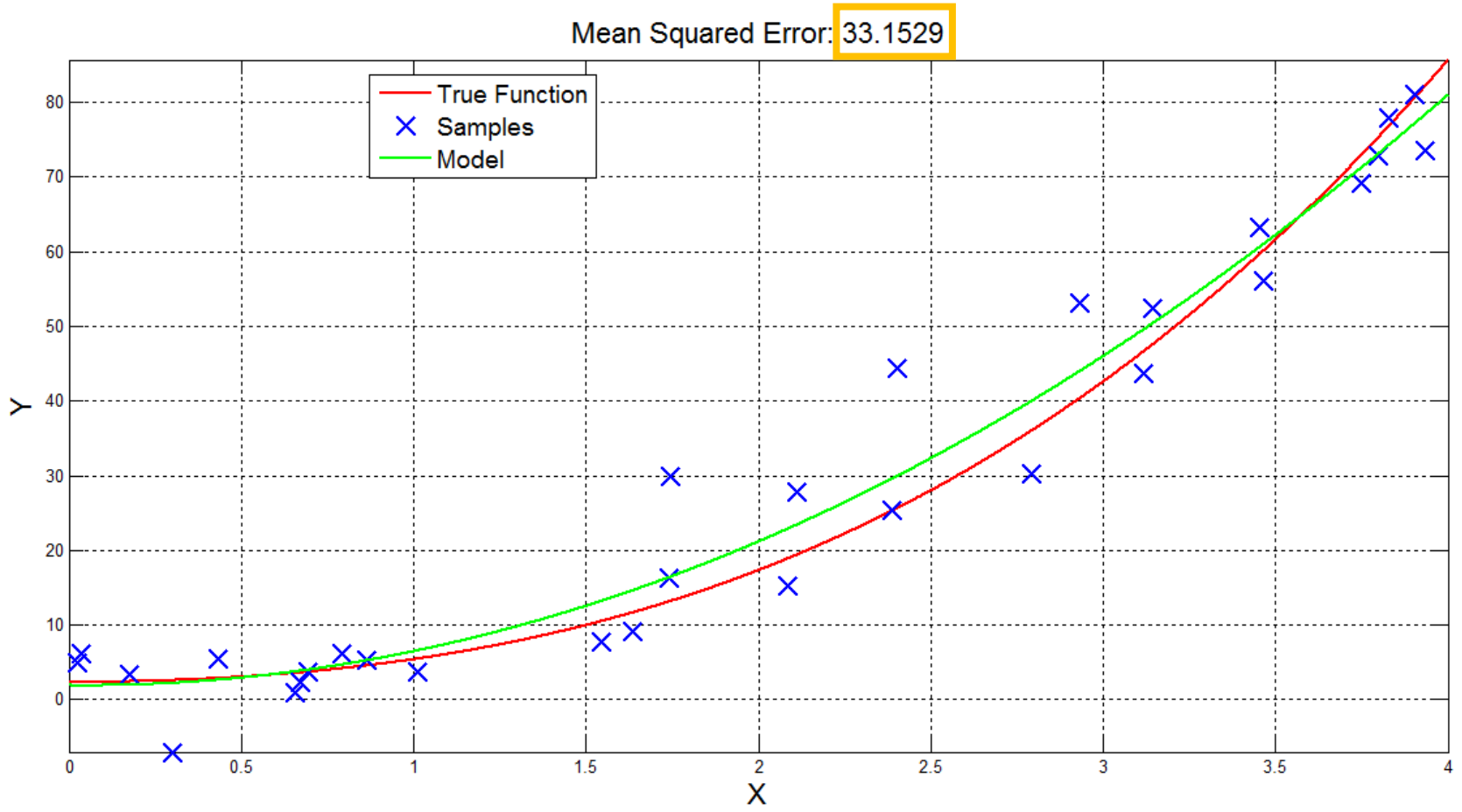


Minimization of MSE on training data

Let's also consider a transformation of the input data!

$Y = a+bX+cX^2$

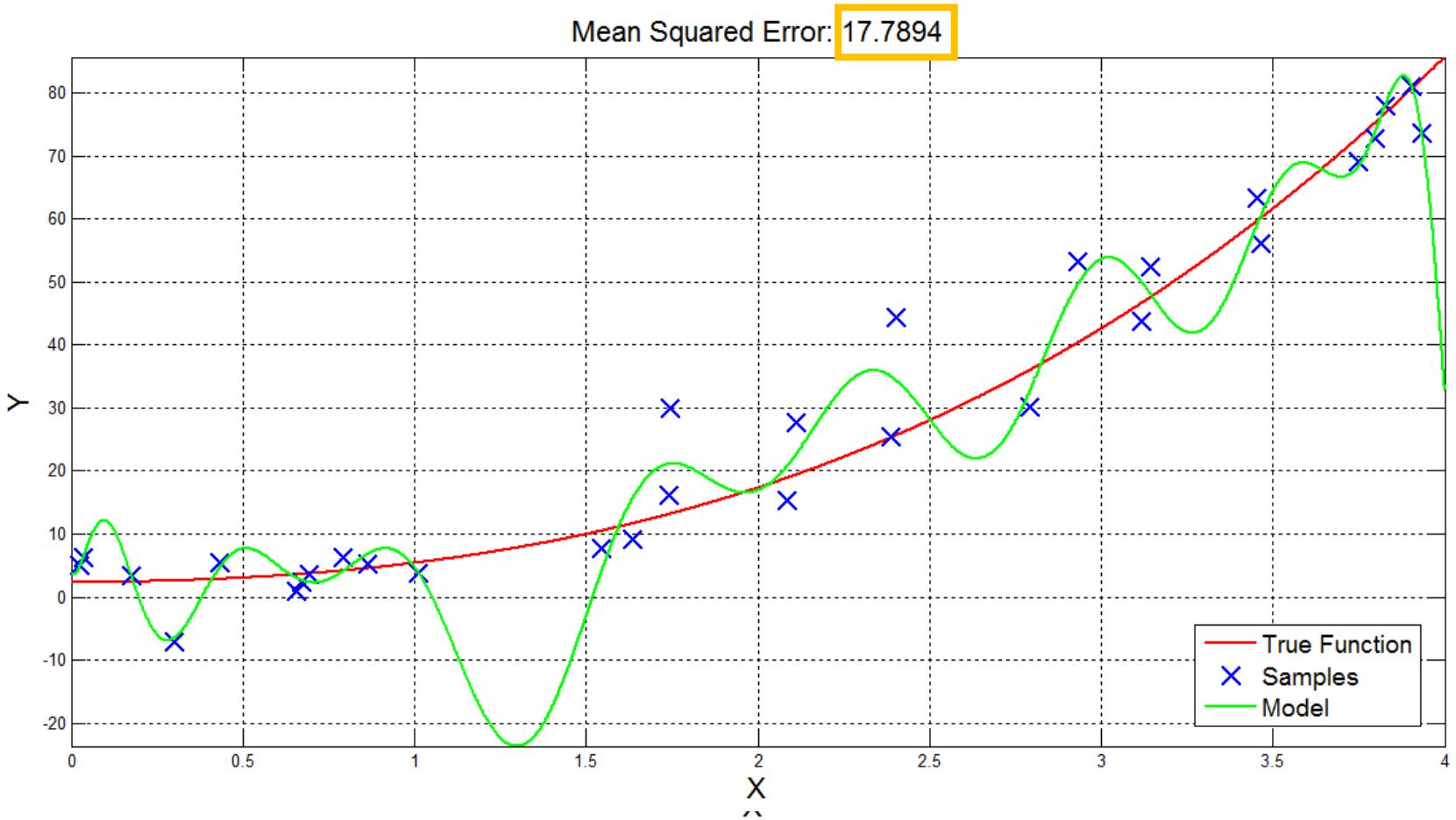
Can we do better again?



Minimization of MSE on training data

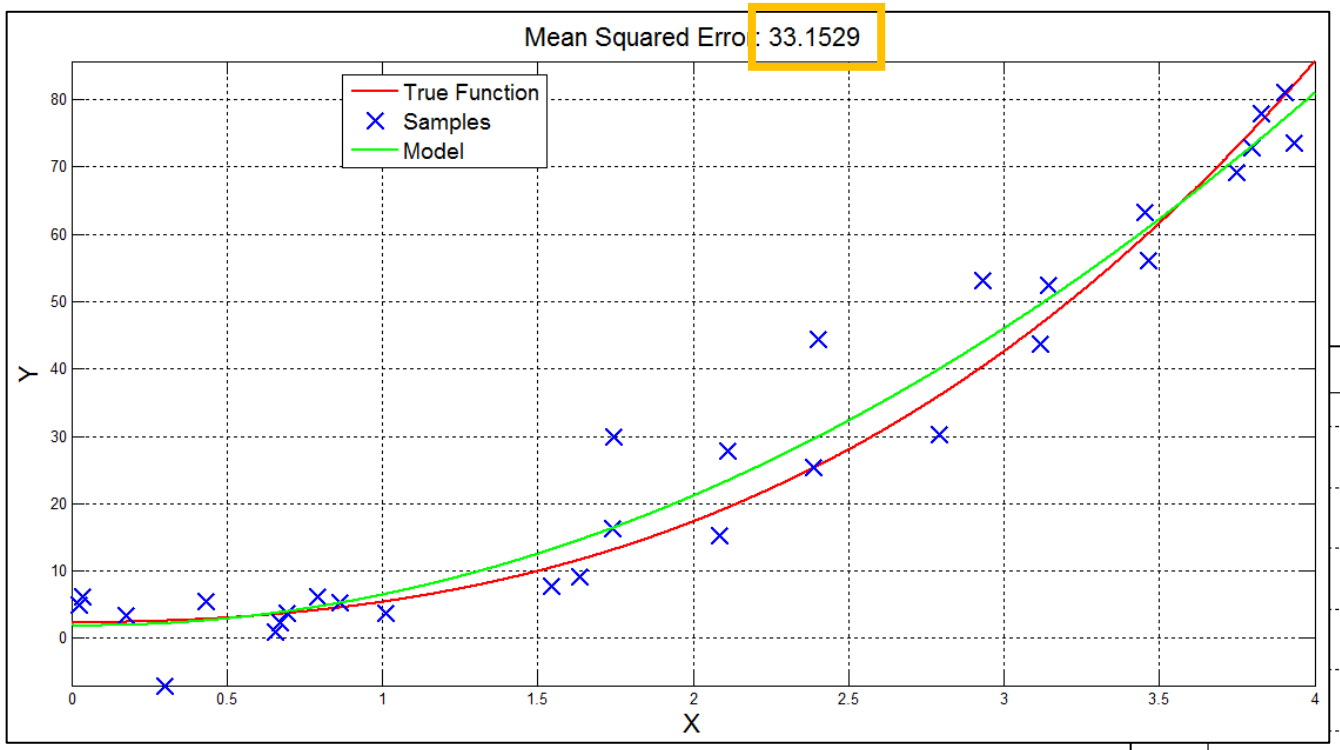
Extension to the 20-th order

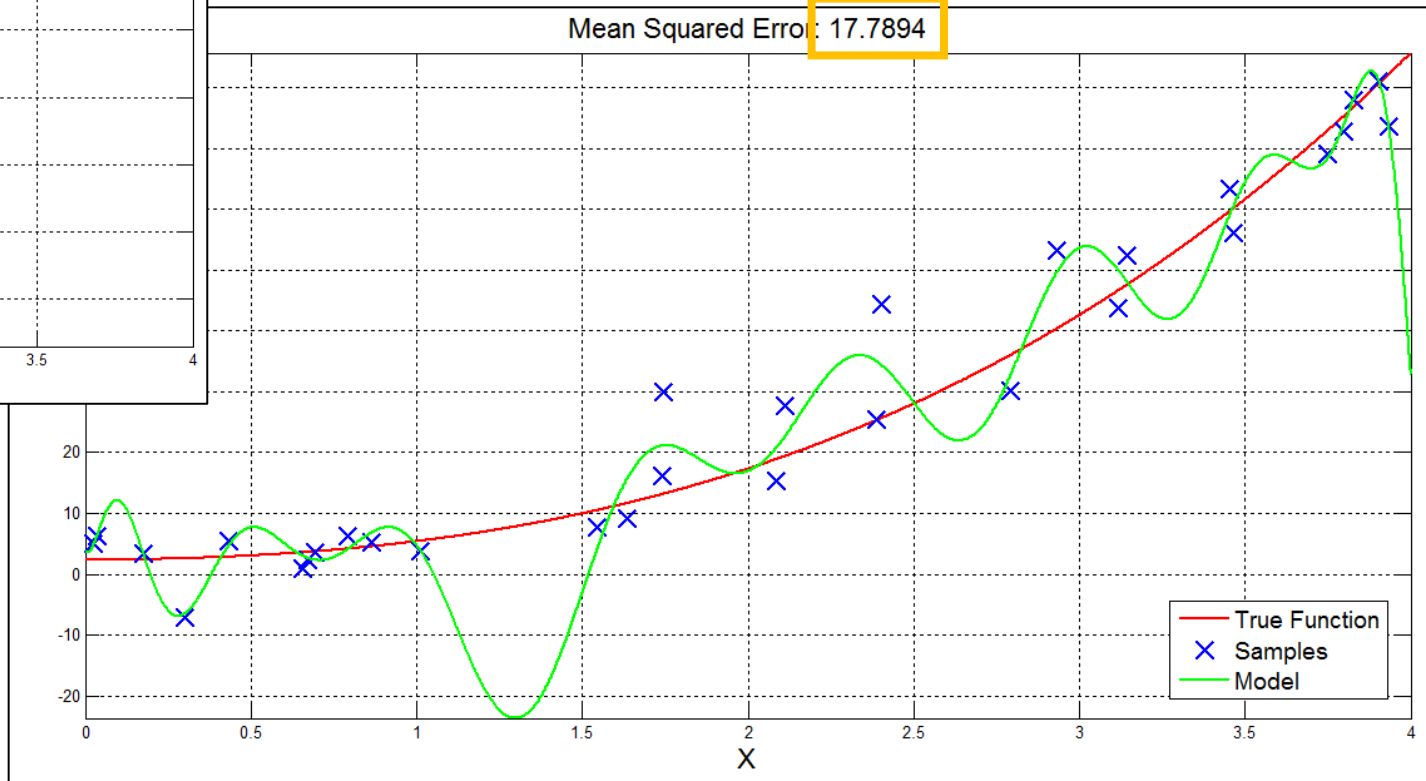
$Y = a+bX+cX^2+ \ldots +vX^{20}$



Minimization of MSE on training data

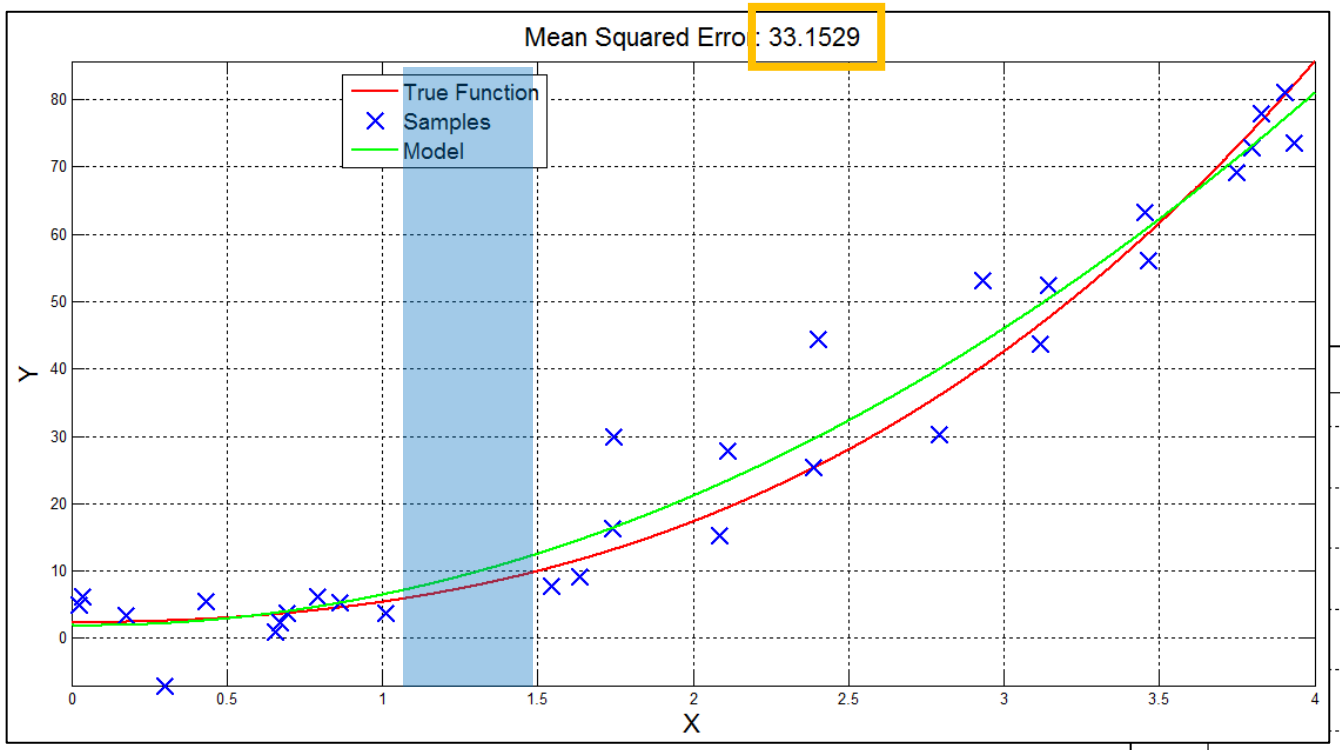
Which one is more ‘reasonable’?

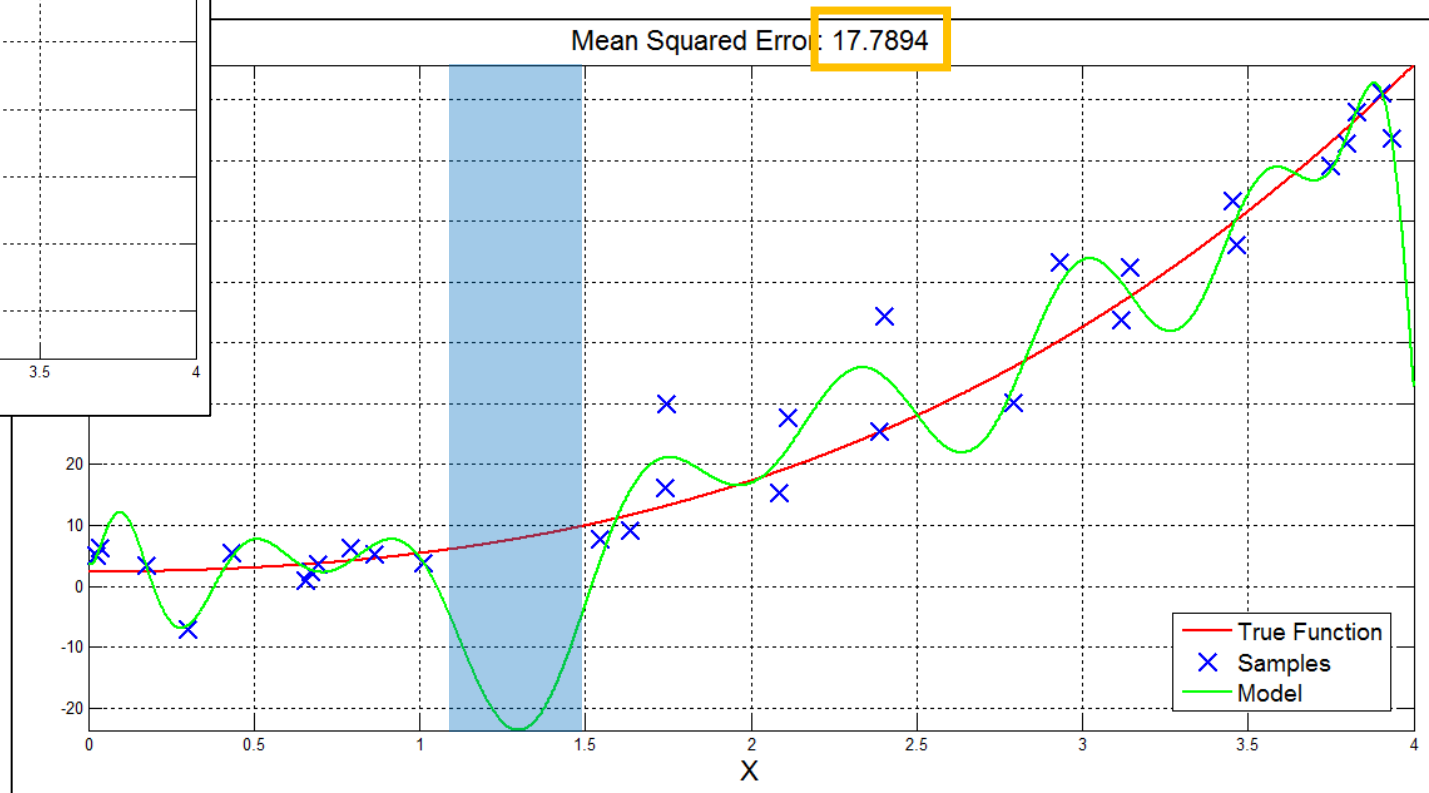




Minimization of MSE on training data

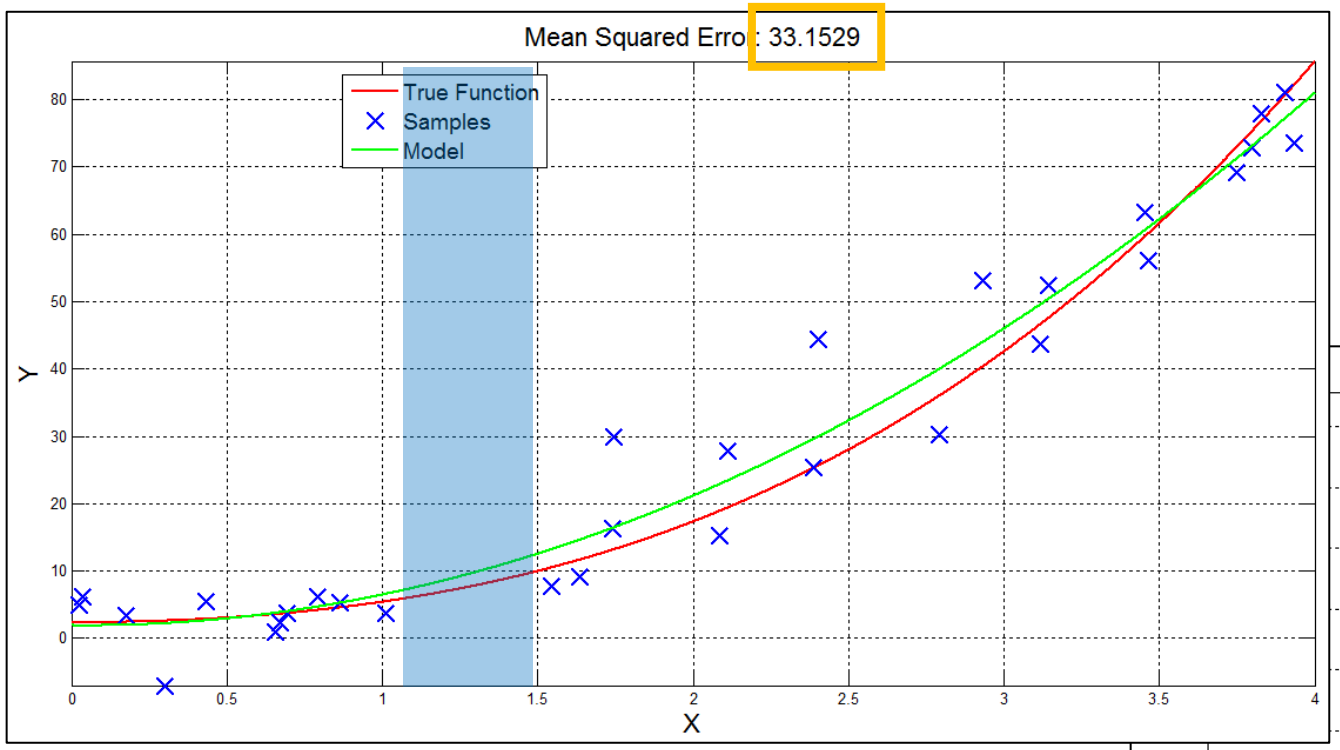
Which one is more ‘reasonable’?



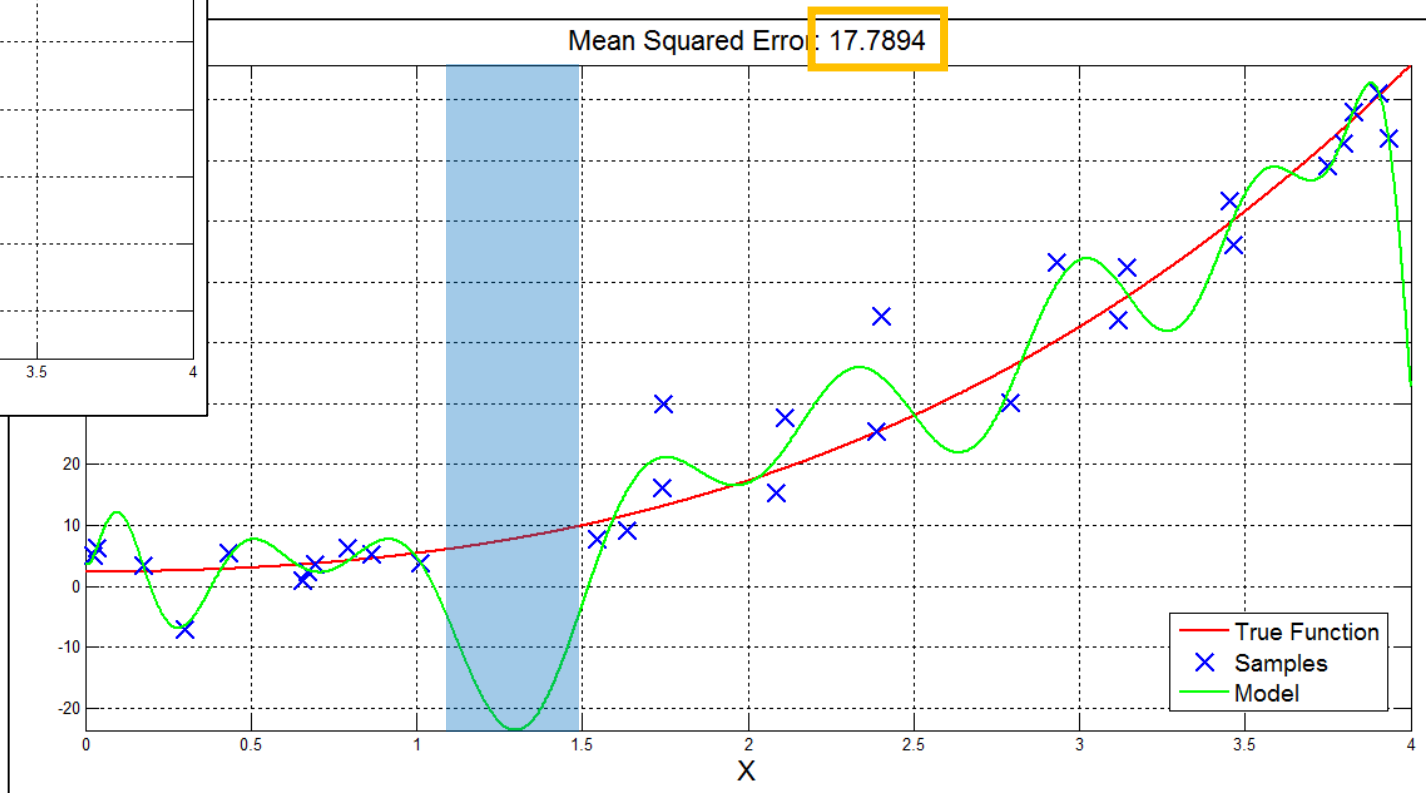


Minimization of MSE on training data

Which one is more ‘reasonable’?

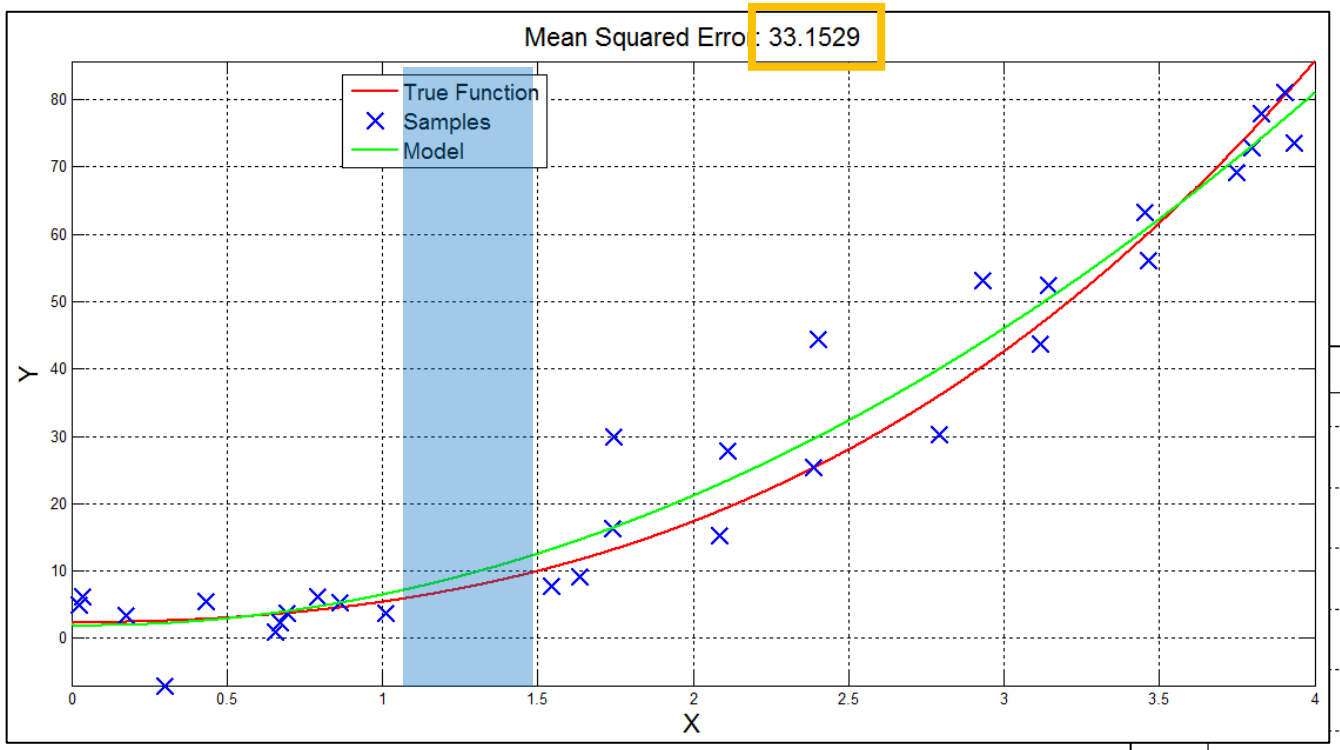


Complicating a model does not always improve its accuracy on new data! (lack of generalization)

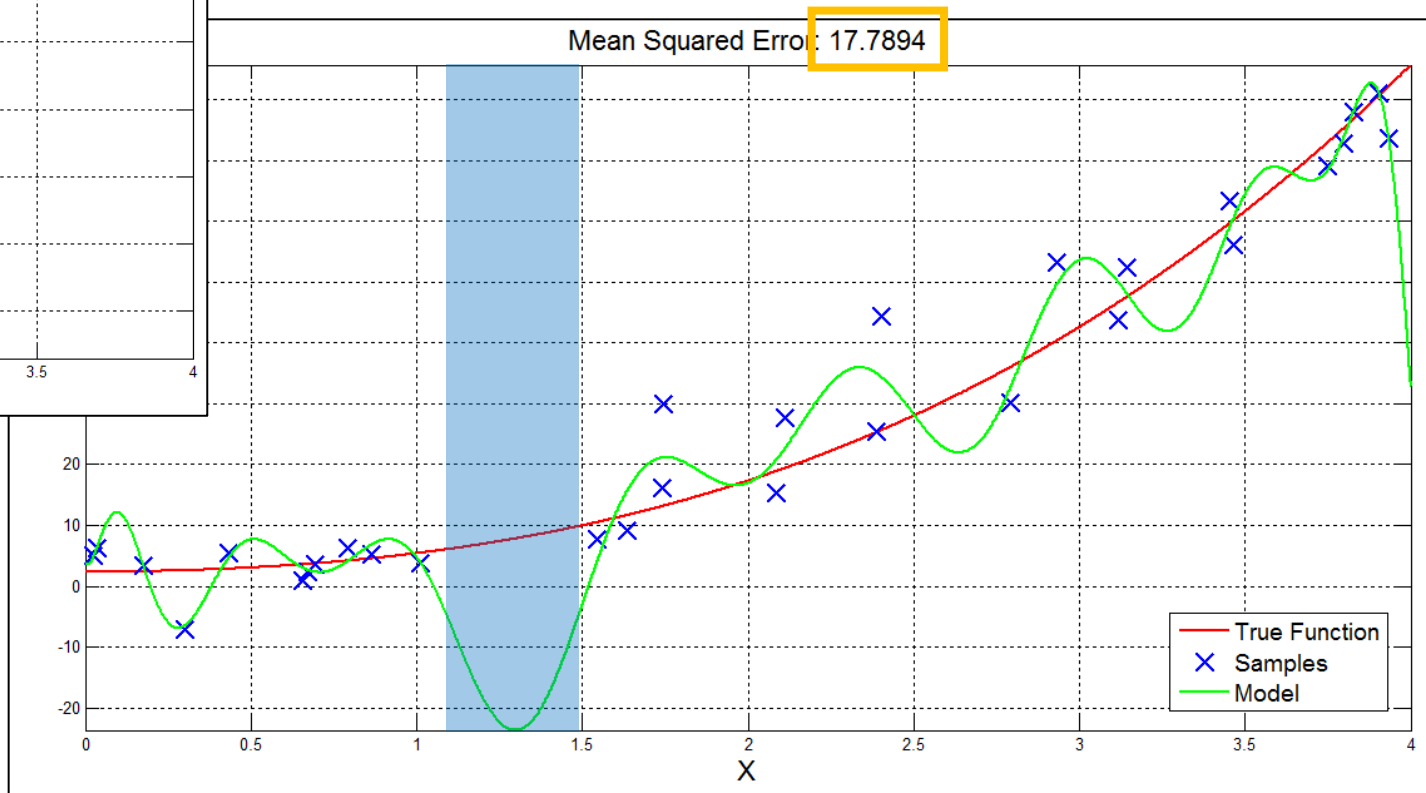


Minimization of MSE on training data

Which one is more ‘reasonable’?



Complicating a model does not always improve its accuracy on new data! (lack of generalization)
-> We need an ‘independent’ dataset, known as a validation set, where we see performances on new cases!



Training vs. Test

- To understand how good a model is in generalizing, we need a test set: a set of data that was not used to find the model parameters.
- Such data should not share data sample with the training set: a set of data used to builing a model, finding the ‘best’ parameters

Example on California Housing dataset

We put randomly 20% of the dataset in testing, while keeping the rest in training

```
Mean Squared Error (MSE): 0.657451727882265
R-squared (R2): 0.498285085954743 74

Model Coefficients:
MedInc: 0.445465596586923 64
HouseAge: 0.016904055548308032
AveRooms: -0.028380689805166 48
AveOccup: -0.004143822818663251

Intercept: 0.026697367635455382
```

Example on California Housing dataset

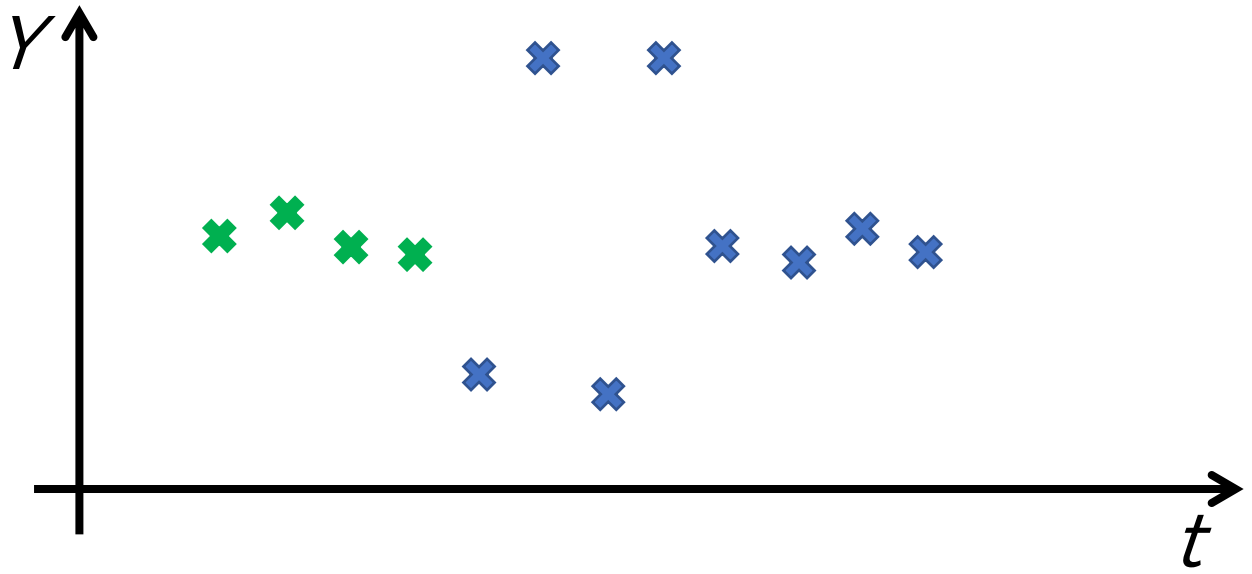
We put randomly 20% of the dataset in testing, while keeping the rest in training

```
Mean Squared Error (MSE): 0.657451727882265
R-squared (R2): 0.49828508595474374

Model Coefficients:
MedInc: 0.445465596586923 64
HouseAge: 0.016904055548308032
AveRooms: -0.02838068980516648
AveOccup: -0.004143822818663251

Intercept: 0.026697367635455382
```

Arbitrary choice: random choices are always safe?

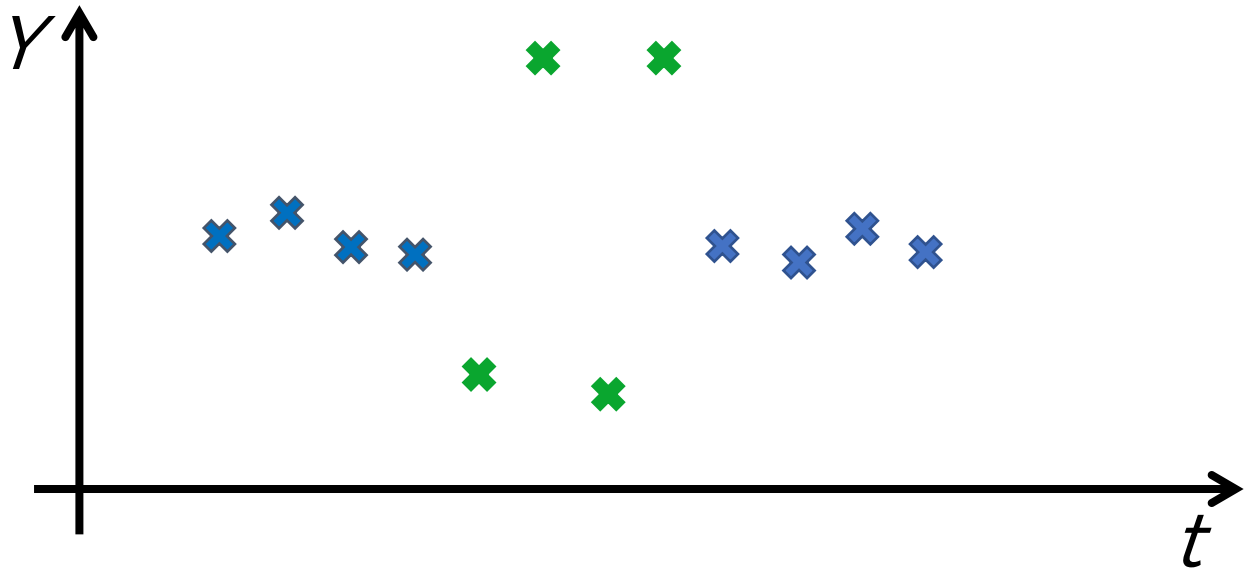
Cross-Validation

- As said, in modeling we divide data into:
  - Training ~~& Validation~~ for model building
  - Test for performance estimation
- Based on the random choice, performance can dramatically change! Especially with ‘small’ datasets



Cross-Validation

- As said, in modeling we divide data into:
  - Training ~~& Validation~~ for model building
  - Test for performance estimation
- Based on the random choice, performance can dramatically change! Especially with ‘small’ datasets



Cross-Validation

- To avoid biases in performance evaluation we use cross-validation
- Approaches
  - K-fold

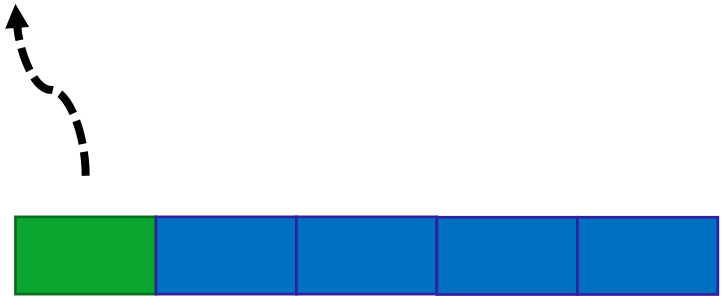
Test Data

Training ~~& Validation~~ Data

Cross-Validation

- To avoid biases in performance evaluation we use cross-validation
- Approaches
  - K-fold

1MSE (Mean Squared Error)... or other performance metric!

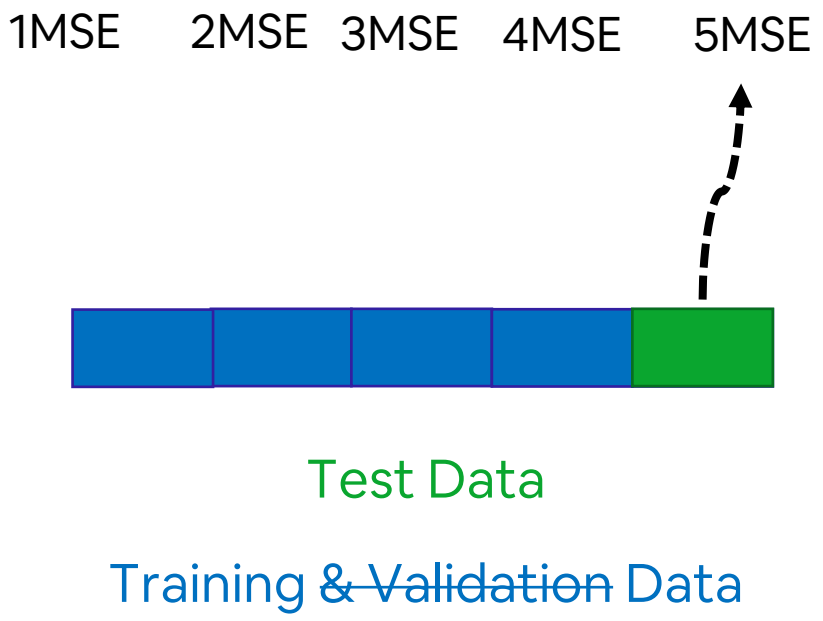


Test Data

Training ~~& Validation~~ Data

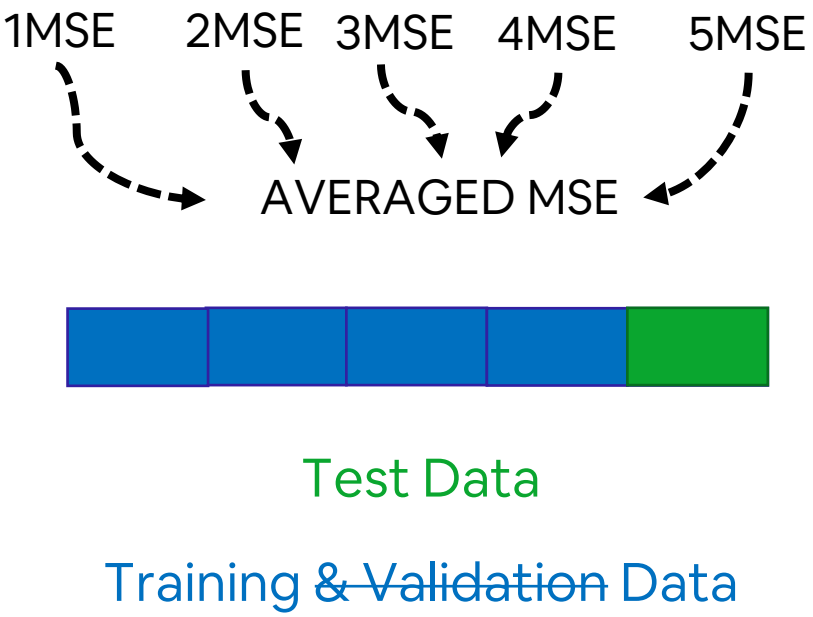
Cross-Validation

- To avoid biases in performance evaluation we use cross-validation
- Approaches
  - K-fold



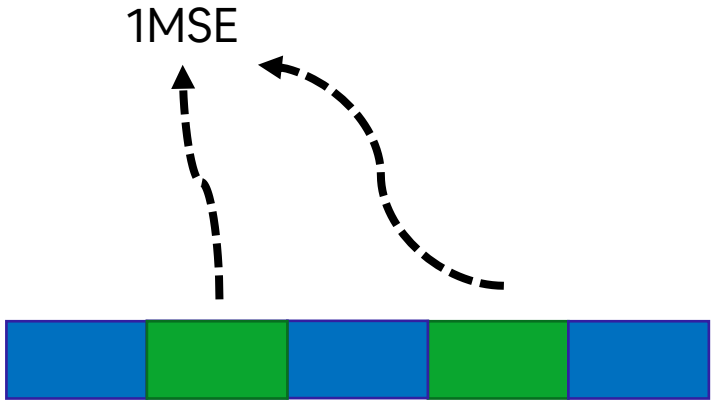
Cross-Validation

- To avoid biases in performance evaluation we use cross-validation
- Approaches
  - K-fold



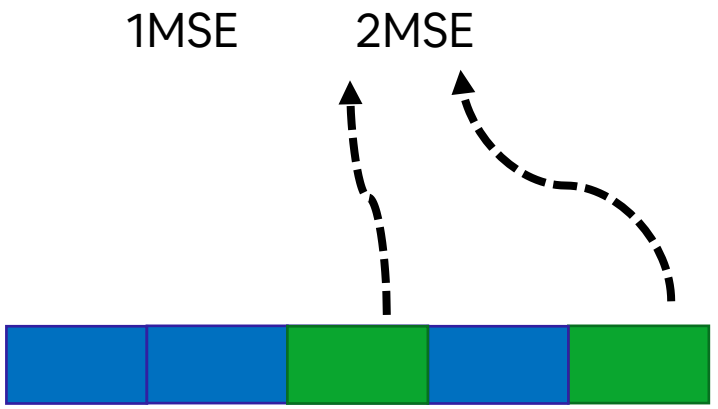
Cross-Validation

- To avoid biases in performance evaluation we use cross-validation
- Approaches
  - K-fold
  - MonteCarlo



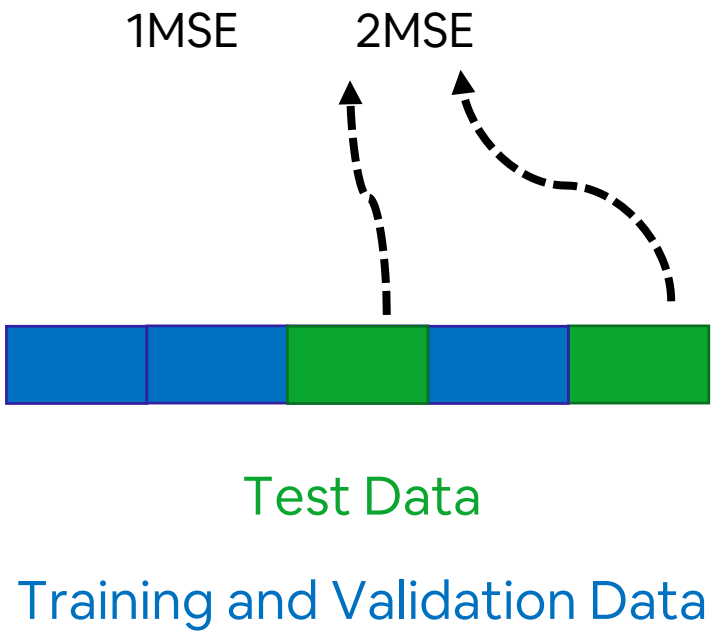
Cross-Validation

- To avoid biases in performance evaluation we use cross-validation
- Approaches
  - K-fold
  - MonteCarlo



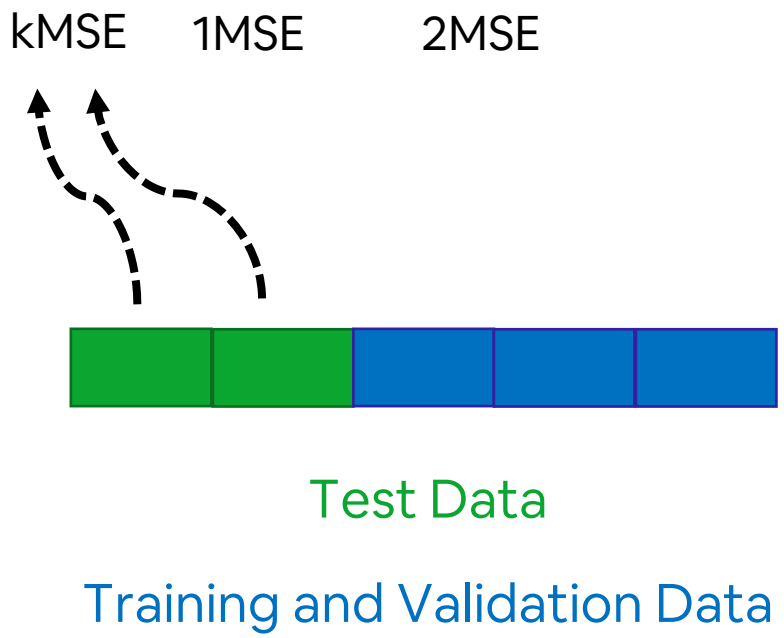
Cross-Validation

- To avoid biases in performance evaluation we use cross-validation
- Approaches
  - K-fold
  - MonteCarlo



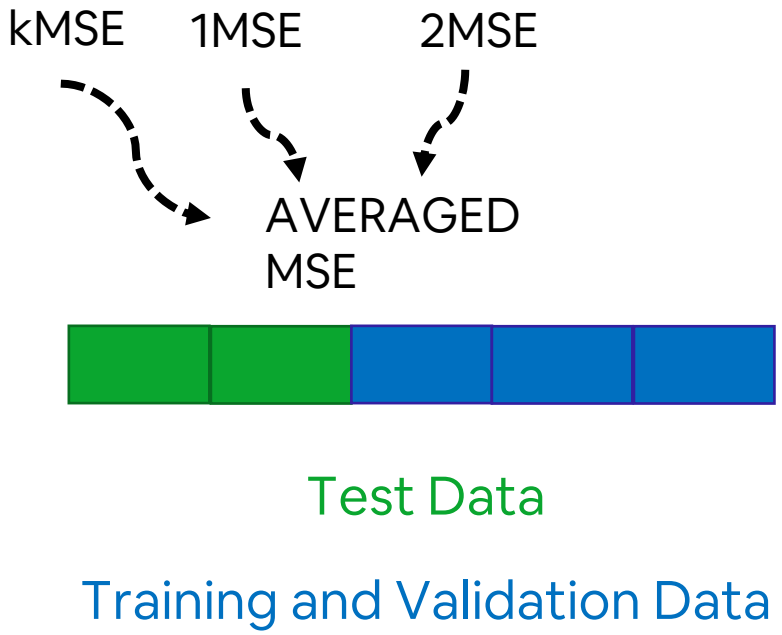
Cross-Validation

- To avoid biases in performance evaluation we use cross-validation
- Approaches
  - K-fold
  - MonteCarlo



Cross-Validation

- To avoid biases in performance evaluation we use cross-validation
- Approaches
  - K-fold
  - MonteCarlo



Thank you!

Gian Antonio Susto