Regression

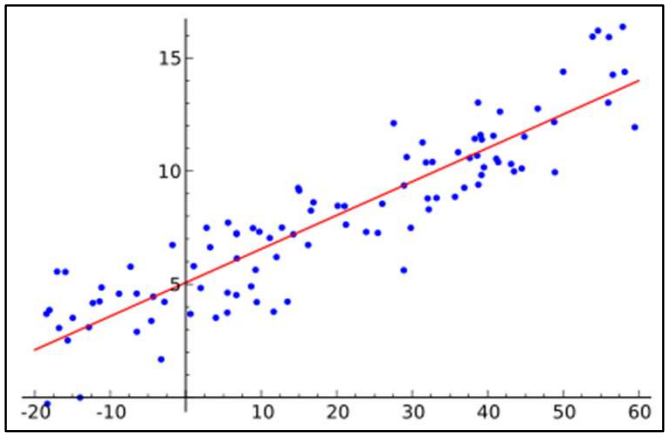
IE500 Data Mining

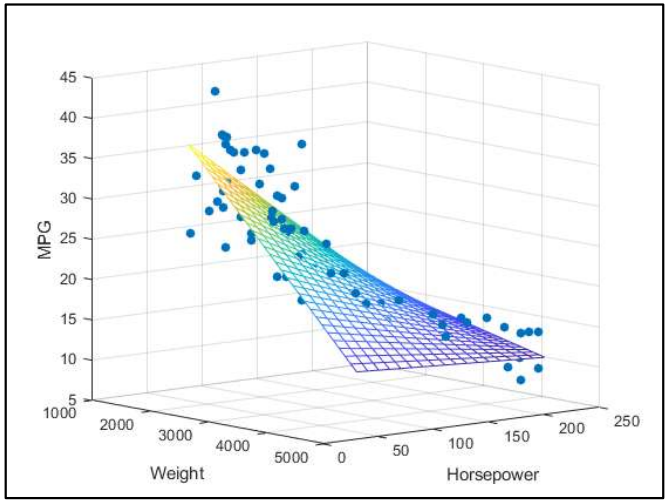
UNIVERSITY OF MANNHEIM
Data and Web Science Group

What is Regression?

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Goal: **predict a numerical value** from a possibly infinite set of possible values
- The predicted variable is called **dependent and is denoted $\hat{y}$**
- The other variables are called **explanatory variables or independent variables** denoted $X = x_1, x_2, \ldots, x_n$





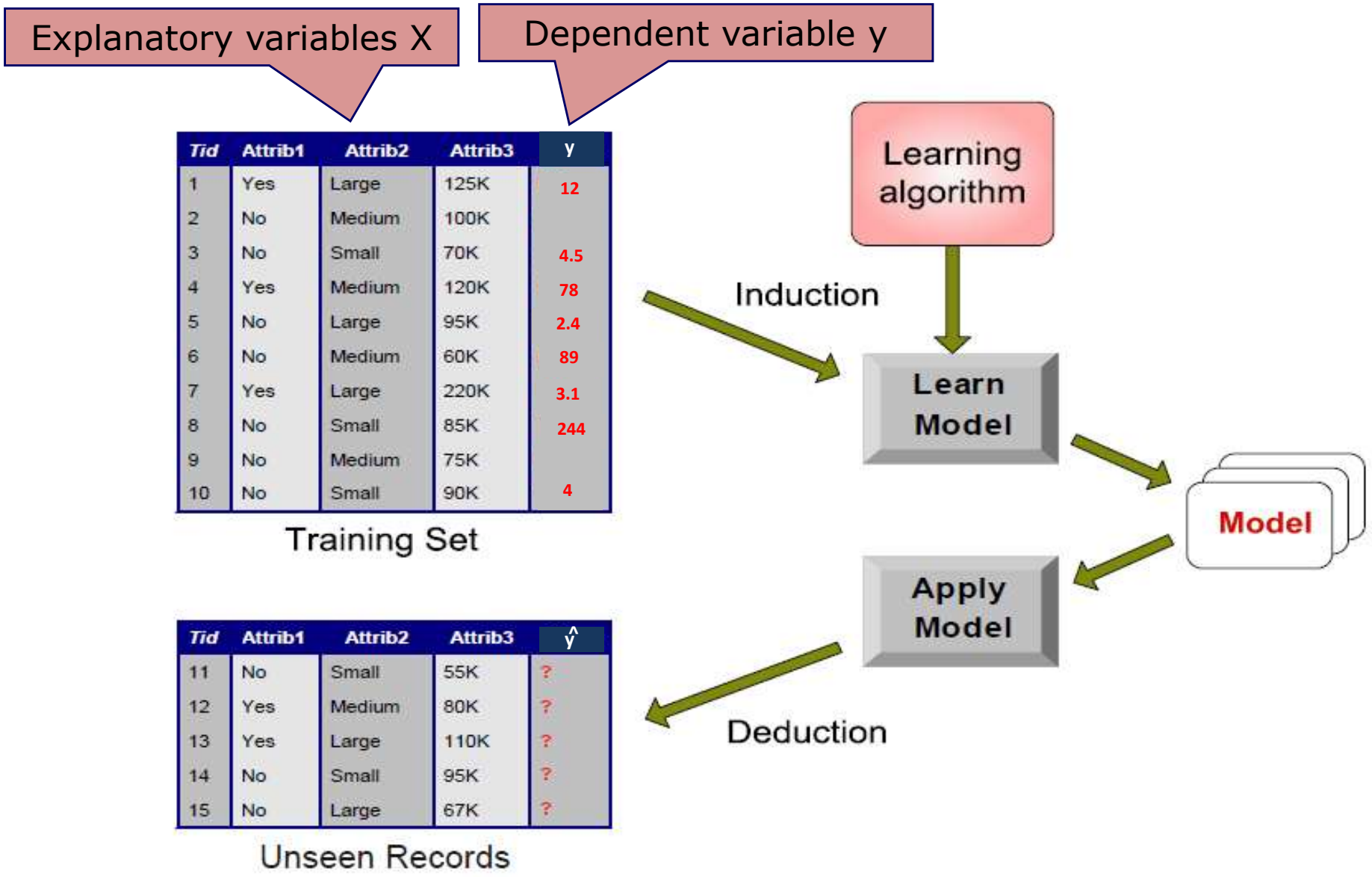
The Regression Problem

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Examples
  - Weather Forecasting
    - Dependent: wind speed
    - Explanatory variables: temperature, humidity, air pressure change
  - House Market
    - Dependent: price of a house
    - Explanatory variables: rooms, distance to public transport, size of garden
- Regression vs. Classification
  - Classification
    - Algorithm “knows” all possible labels, e.g. yes/no, low/medium/high
    - All labels appear in the training data
    - The prediction is always one of those labels
  - Regression
    - Algorithm “knows” some possible values, e.g., 18°C and 21°C
    - Prediction may also be a value not in the training data, e.g., 20°C

Regression Model Learning and Application

UNIVERSITY OF MANNHEIM
Data and Web Science Group



Outline

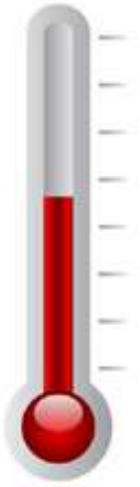
UNIVERSITY OF MANNHEIM
Data and Web Science Group

1. KNN for Regression
2. Model Evaluation
3. Regression Trees
4. Linear Regression
5. Polynomial Regression
6. Local Regression
7. ANNs for Regression
8. The Bias/Variance-Tradeoff

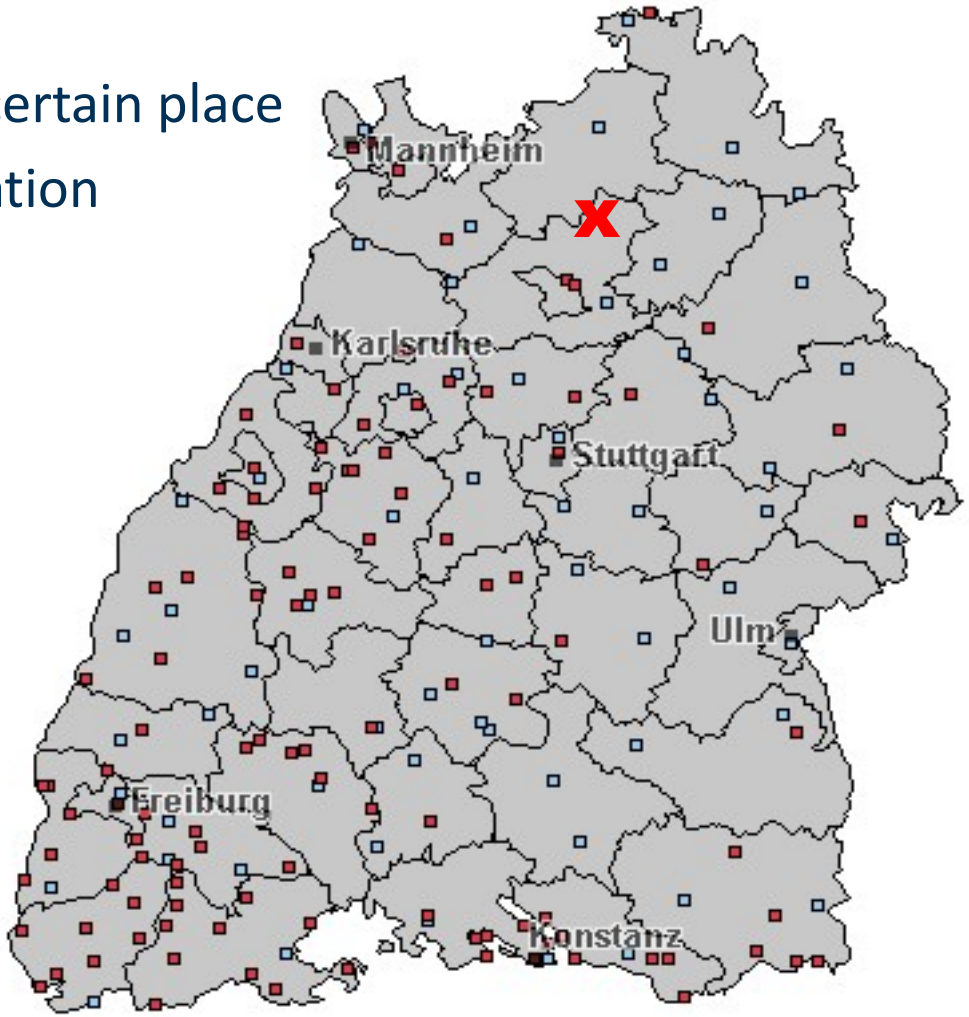
1. K-Nearest-Neighbors Regression

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Problem
  - Predict the **temperature** in a certain place
  - Where there is no weather station
  - How could you do that?

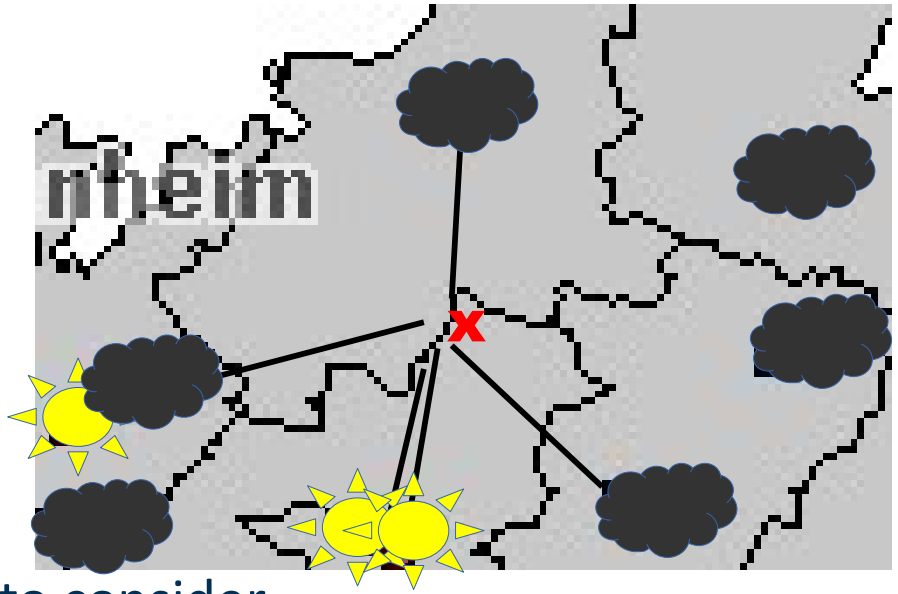


? °C



Recap: K-Nearest-Neighbors Classification

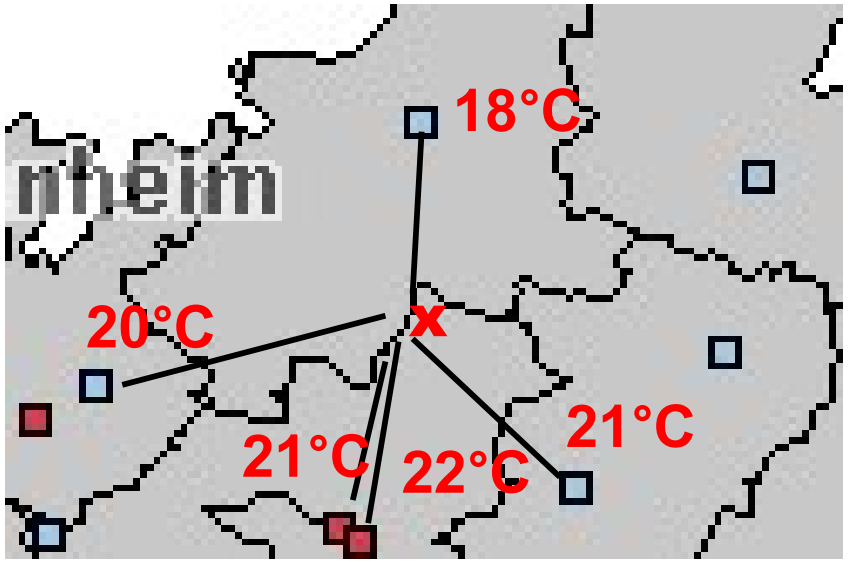
- Idea: **Vote of the nearest stations**
- Example:
  - 3x cloudy
  - 2x sunny
  - Result: cloudy
- Approach is called
  - “k nearest neighbors”
  - where k is the number of neighbors to consider
  - in the example: k=5
  - in the example: “near” denotes geographical proximity



K-Nearest-Neighbors Regression

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Idea: **Use the numeric average of the nearest stations**
- Example:
  - 18°C, 20°C, 21°C, 22°C, 21°C
- Compute the average
  - again: k=5
  - average = (18+20+21+22+21) / 5
  - prediction: $\hat{y}$ = 20.4°C



- Can also be weighted by the distance to the nearest neighbors

K-NN Regression in Python

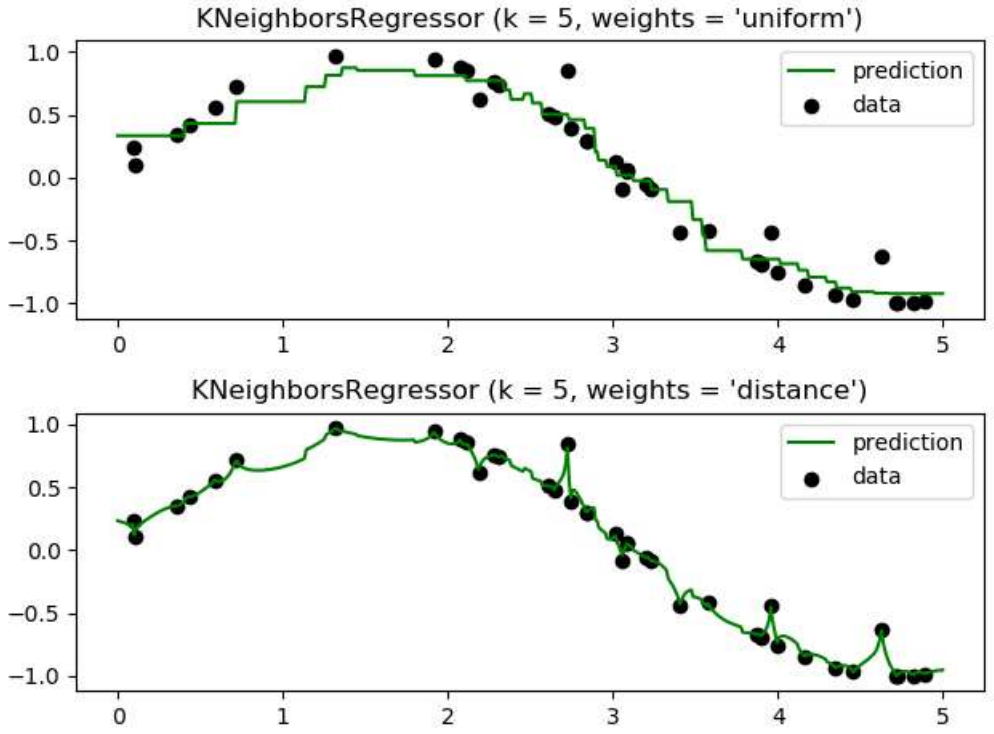
UNIVERSITY OF MANNHEIM
Data and Web Science Group

Python

```python
from sklearn.neighbors import KNeighborsRegressor

# Create and fit a KNN regressor
estimator = KNeighborsRegressor(n_neighbors=15)
estimator.fit(training_set_X, training_dependent_y)

# Make predictions for unseen examples
y_hat = estimator.predict(test_set_X)
```



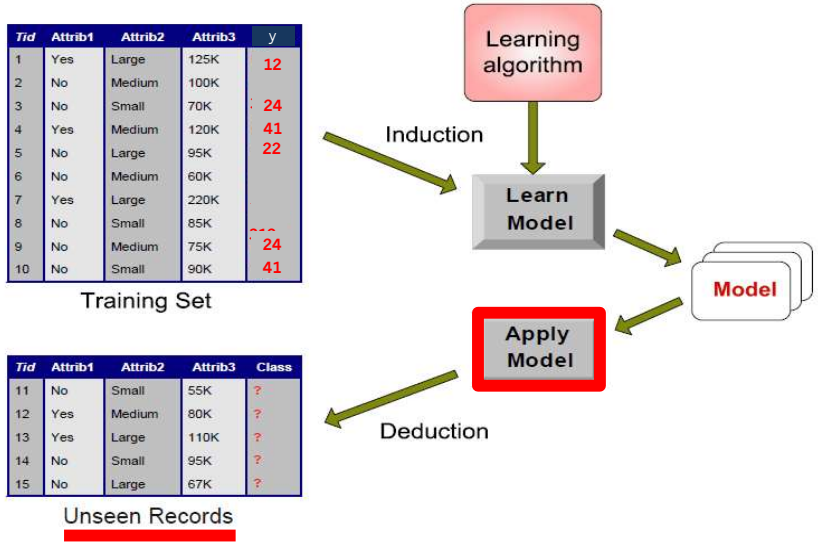
2. Model Evaluation

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Central Question:

How good is a model at predicting the dependent variable for unseen records?

(generalization performance)



3.1 Methods for Model Evaluation

- How to obtain reliable estimates?

3.2 Metrics for Model Evaluation

- How to measure the performance of a regression model?

2.1 Methods for Model Evaluation

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- The same considerations apply as for classification:
- Cross validation: 10-fold (90% for training, 10% for testing in each iteration)
- Nested cross-validation for hyperparameter selection
  - uses inner cross validation to select best hyperparameter values
  - uses outer cross validation to estimate generalization error of models learned using best hyperparameter values

Python

```python
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsRegressor

# Create KNN regressor
estimator_knn = KNeighborsRegressor()

# Specify the hyperparameter values for the search
grid = {"n_neighbors": range(1,20)}

# Create the grid search estimator for model selection
estimator_gs = GridSearchCV(estimator_knn, grid, cv=5, scoring='neg_mean_squared_error')

# Run nested cross-validation for model evaluation
mse_cv = cross_val_score(estimator_gs, X, y, cv=5, scoring='neg_mean_squared_error')
```

2.2 Evaluation Metrics

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- **Mean Absolute Error (MAE)** computes the average deviation between **predicted value $\hat{y}_i$** and the **actual value y_i**

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i|$$

Same scale as the domain
e.g. temperature

- **Mean Squared Error (MSE)** places more emphasis on larger deviations

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

- **Root Mean Squared Error (RMSE)** has similar scale as MAE and places more emphasis on larger deviations

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$$

Same scale as the domain
e.g. temperature

Evaluation Metrics

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- **R Squared** (also called **Coefficient of Determination**)
  - measures the proportion of variance (of y) that has been explained by the independent variables X in the model
  - best possible score is 1.0, lower values are worse
  - $R^2 = 1$: Perfect model as total variation of y can be completely explained from X

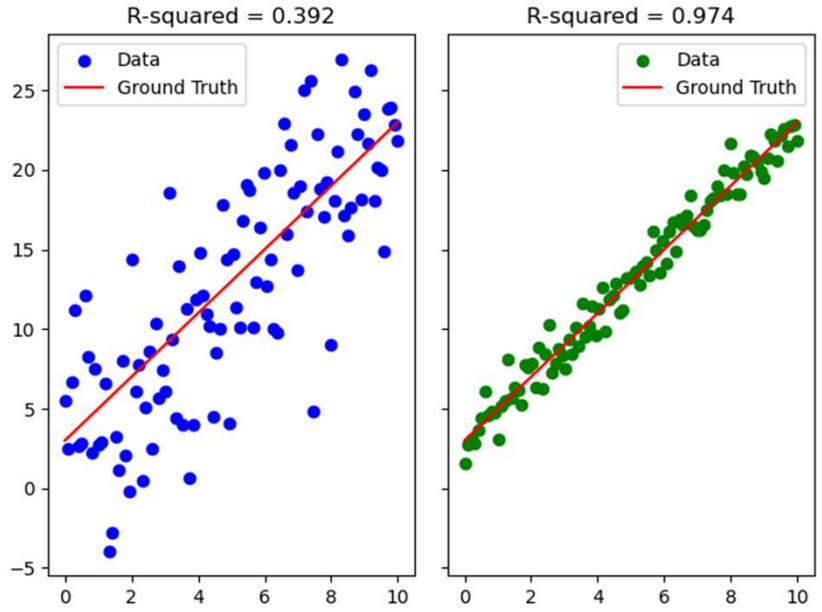
$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

sum of squares of residuals

total sum of squares

Model predicting the mean, will have $R^2 = 0$

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

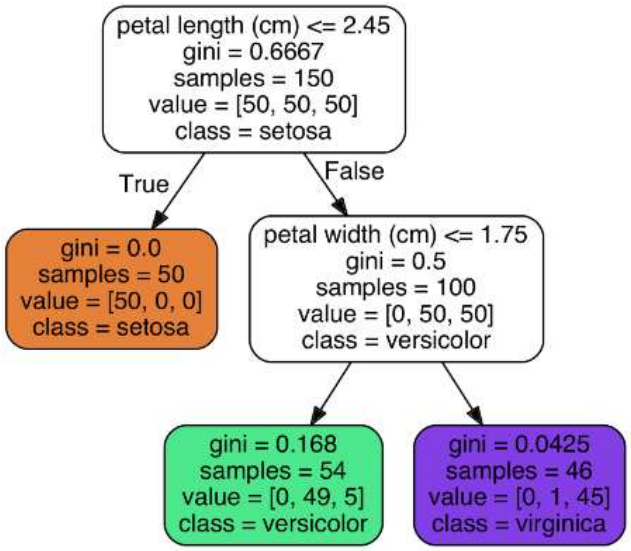


3. Regression Trees

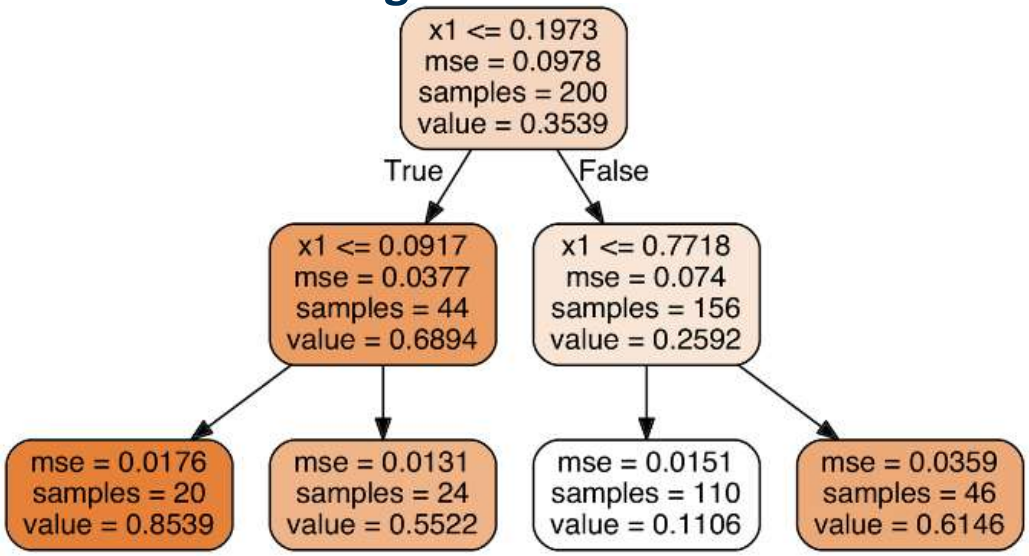
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- The basic idea of how to learn and apply decision trees can also be used for regression
- Differences:
  1. Splits are selected by maximizing the MSE reduction (not GINI)
  2. Prediction is average value of the training examples in a specific leaf

Decision Tree



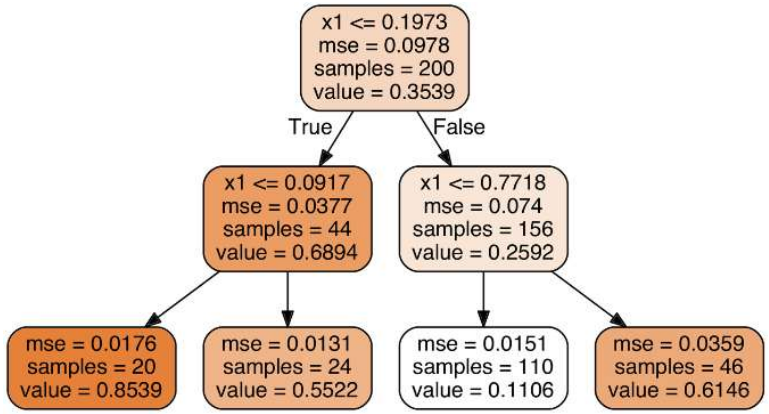
Regression Tree

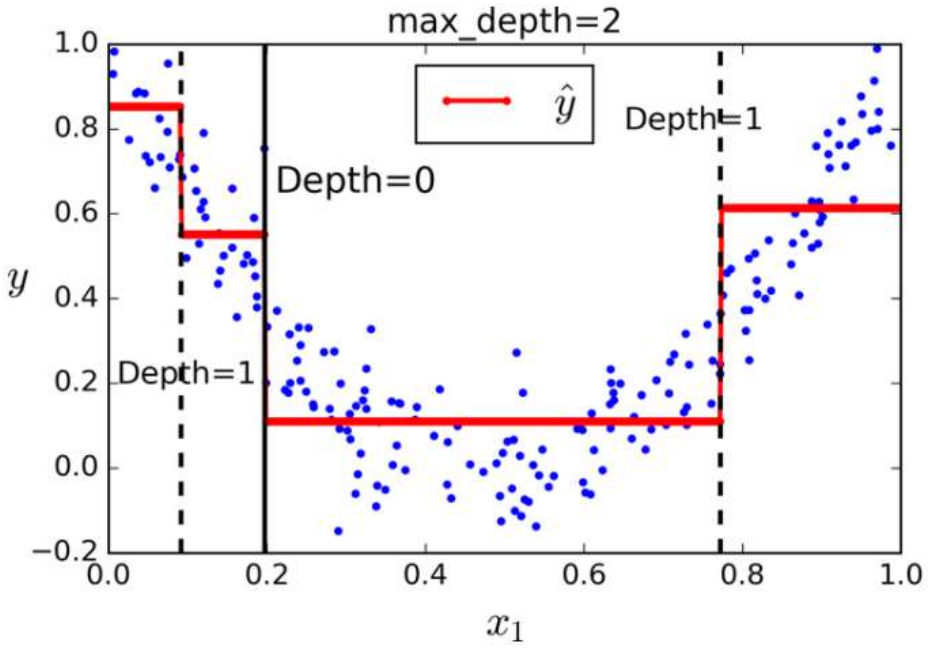


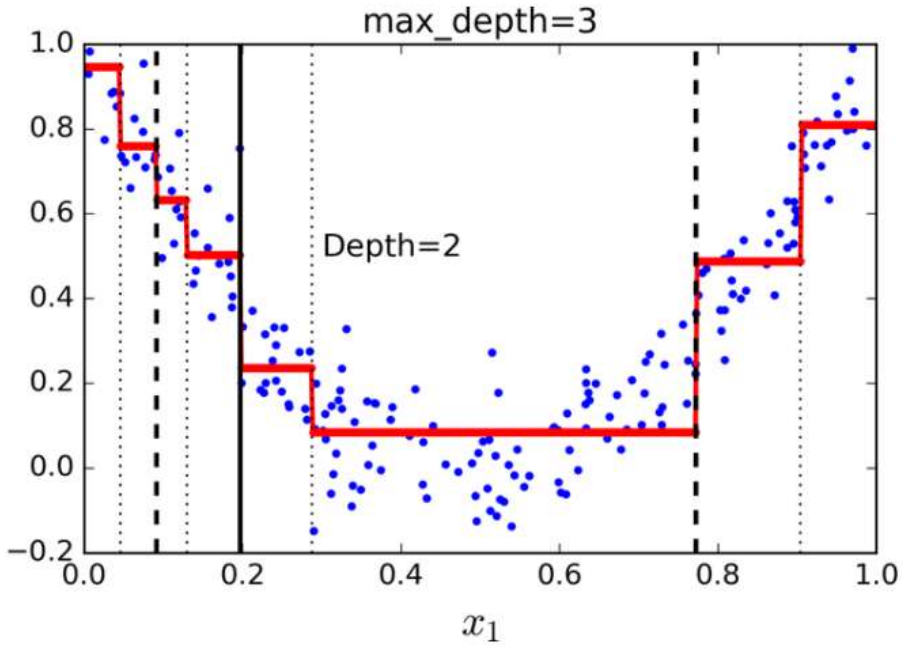
Regression Trees

UNIVERSITY OF MANNHEIM
Data and Web Science Group

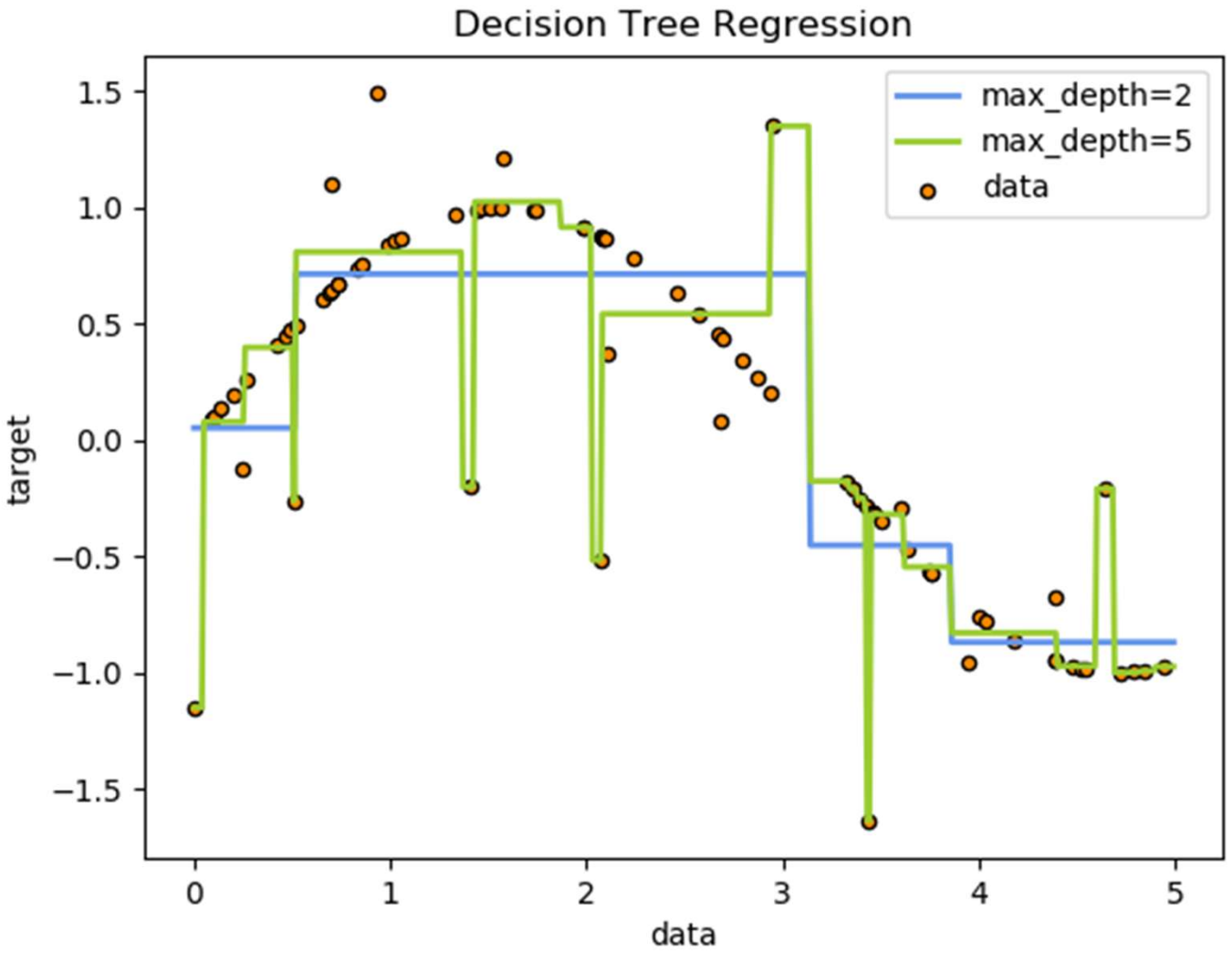
- Pre-pruning parameters determine how closely the tree fits the training data
  - E.g. max_depth parameter
- Resulting model: piecewise constant function







Overfitted Regression Tree

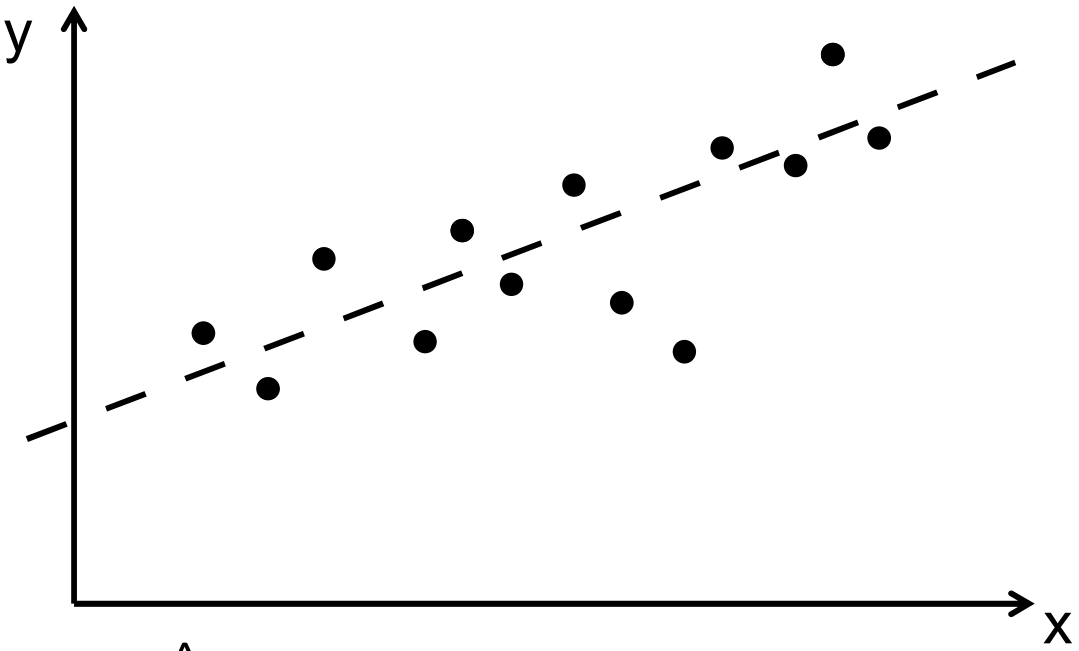


The learning algorithm uses the available tree depth to model the most extreme outliers

4. Linear Regression

UNIVERSITY OF MANNHEIM
Data and Web Science Group

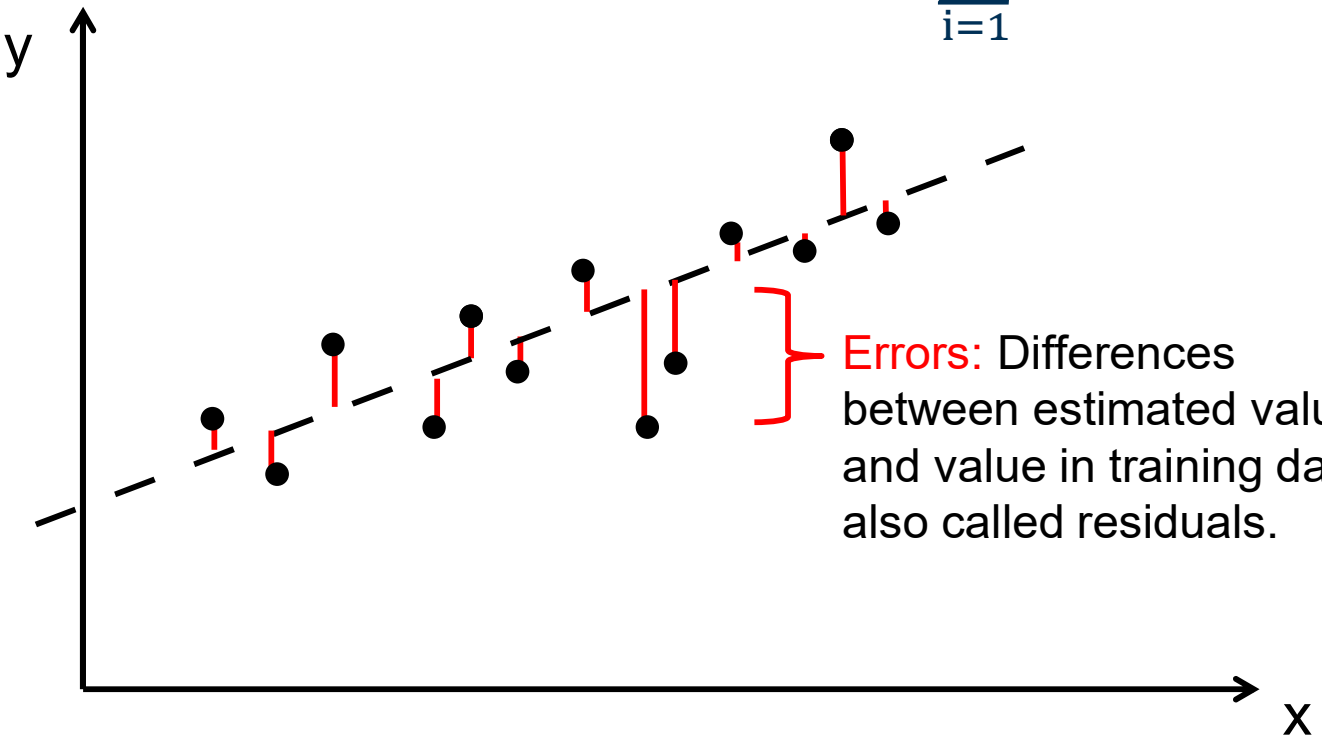
- Assumption: The target variable $\hat{y}$ is (approximately) linearly dependent on explanatory variables X
  - For visualization: we use one variable x (simple linear regression)
  - In reality: vector $X = x_1, x_2, \ldots, x_n$ (multiple linear regression)



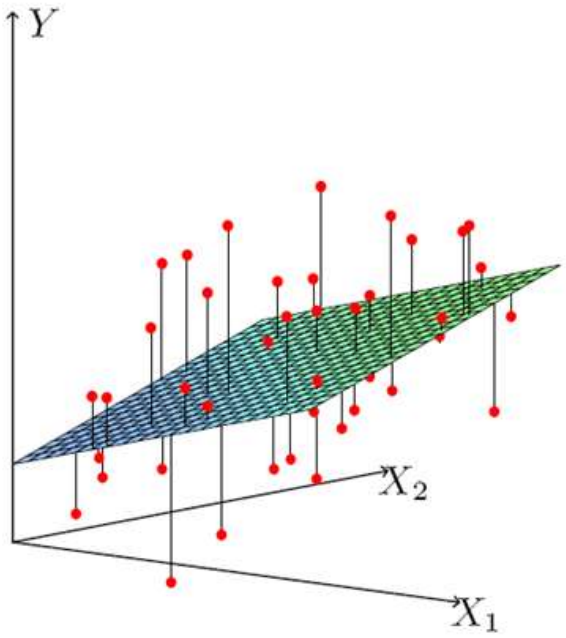
Fitting a Regression Function

Least-Squares Approach: Find the weight vector $W = (w_0, w_1, .., w_n)$ that minimizes the sum of squared error (SSE) over all training examples

$$SSE = \sum_{i=1}^{n} (y_i - f(x_i))^2$$



Errors: Differences between estimated value and value in training data, also called residuals.

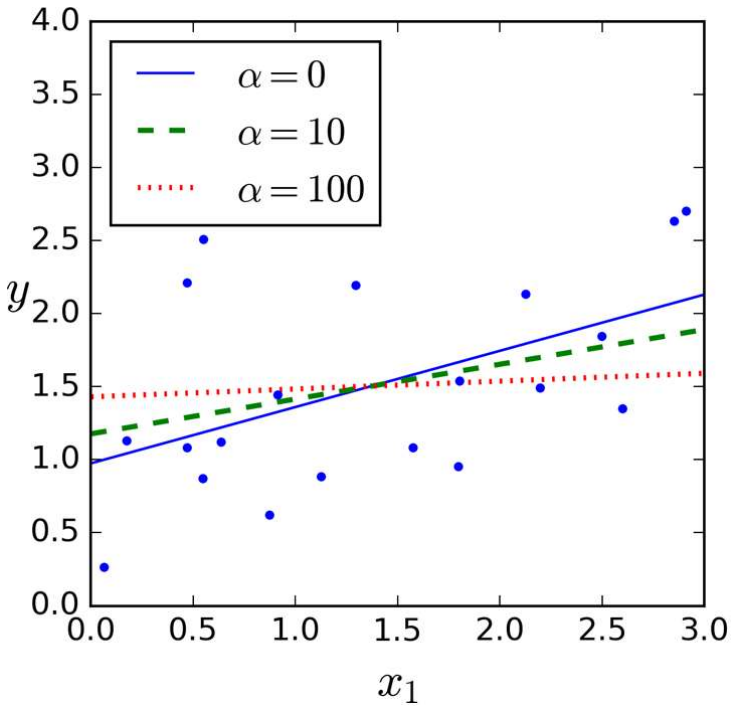


Ridge Regularization

- Variation of least squares approach which tries to avoid overfitting by keeping the weights W small
- Ridge regression cost function to minimize

$$C(W) = MSE(W) + \alpha \sum_{i=1}^{n} w_i^2$$

  - α = 0 : Normal least squares regression
  - α = 100 : Strongly regularized flat curve



Feature Selection

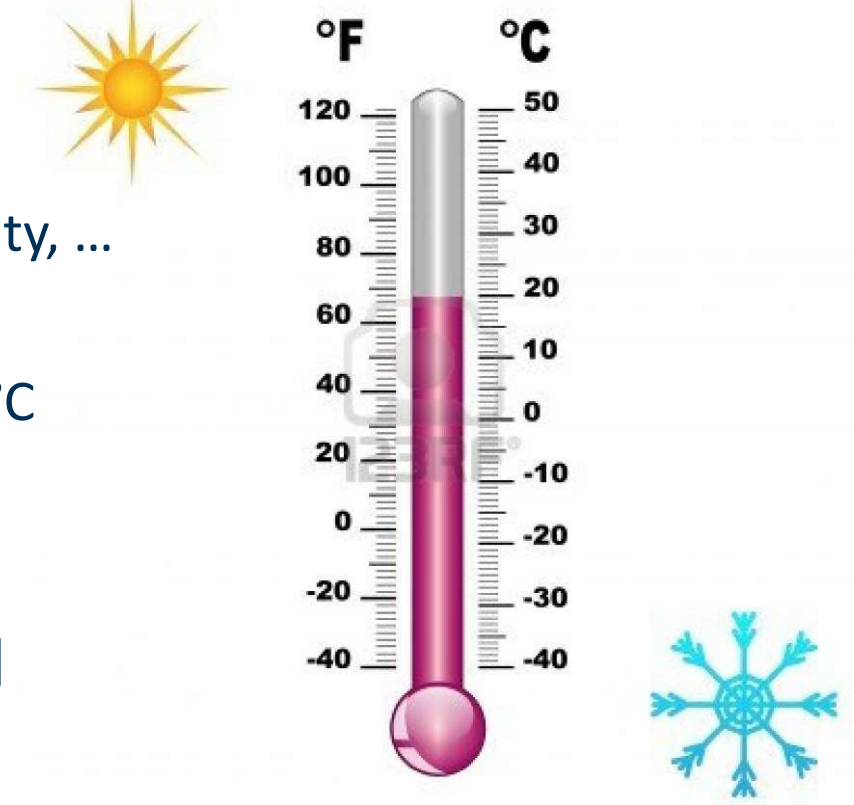
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Question: Do all explanatory variables X help to explain y or is only a subset of the variables useful?
- Problem 1: Highly correlated variables (e.g. height in cm and inch)
  - weights are meaningless and one variable should be removed for the better interpretability of the weights
- Problem 2: Insignificant variables (e.g. the weather for stock prices)
  - uncorrelated variables get w=0 or relatively small weights assigned
  - Question for variables having small weights: Is the variable still useful or did it get the weight by chance due to biased training data?
  - Answer: Statistical test with null-hypothesis "w=0 as variable is insignificant"
    - **t-stat:** number of standard deviations that w is away from 0
      - high t-stat ➔ Variable is significant as it is unlikely that weight is assigned by chance
    - **p-value:** Probability of wrongly rejecting the hull-hypothesis
      - p-value close to zero ➔ variable is significant
  - See: James, Witten, et al.: An Introduction to Statistical Learning. Chapter 3.1.2

Interpolation vs. Extrapolation

UNIVERSITY OF MANNHEIM
Data and Web Science Group

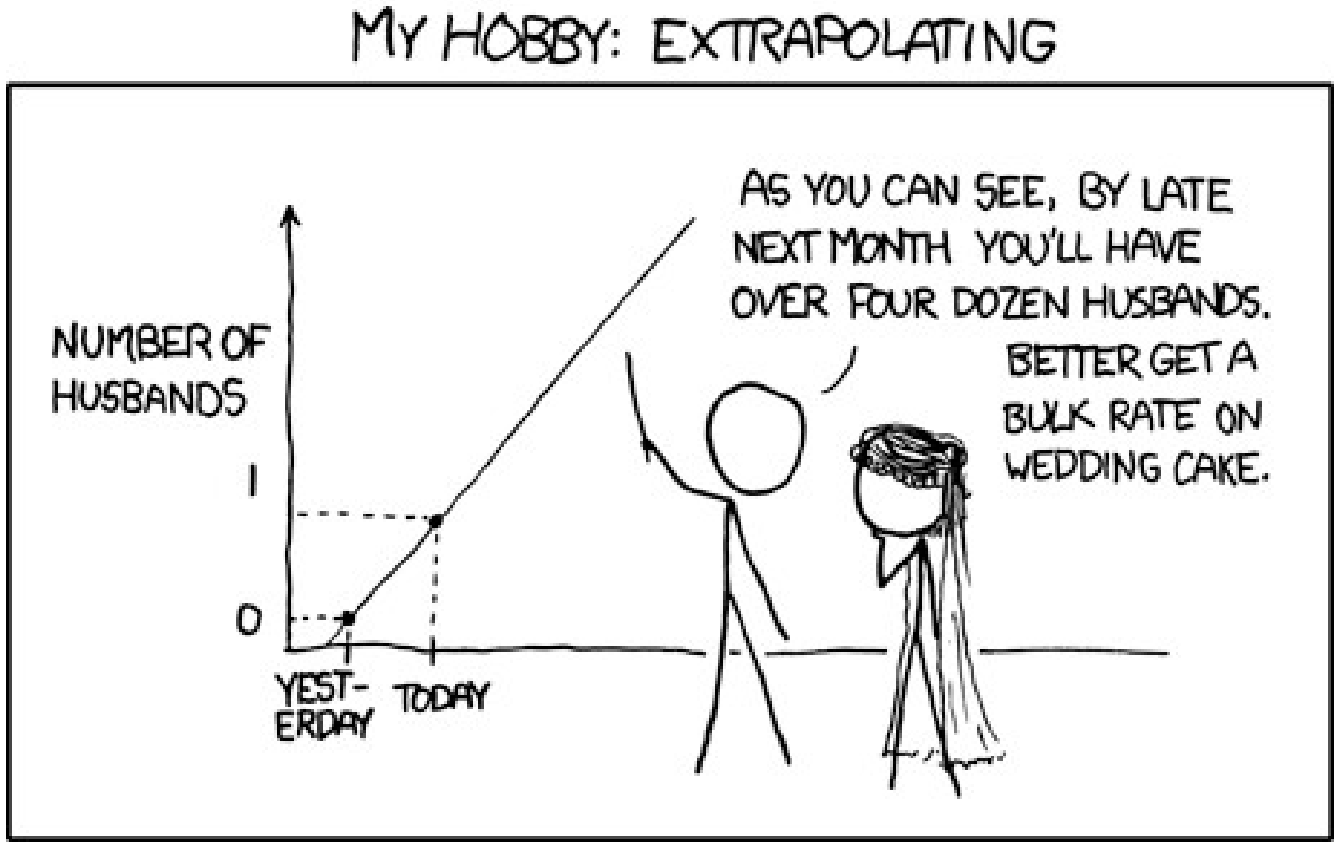
- Training data:
  - Weather observations for current day
  - E.g., temperature, wind speed, humidity, …
  - Target: temperature on the next day
  - Training values between -15°C and 32°C
- Interpolating regression
  - Only predicts values **from the training interval** [-15°C,32°C]
- Extrapolating regression
  - May also predict values ***outside*** **of this interval**



Interpolation vs. Extrapolation

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Interpolating regression is regarded as “safe”
  - i.e., only reasonable/realistic values are predicted



http://xkcd.com/605/

Interpolation vs. Extrapolation

UNIVERSITY OF MANNHEIM
Data and Web Science Group

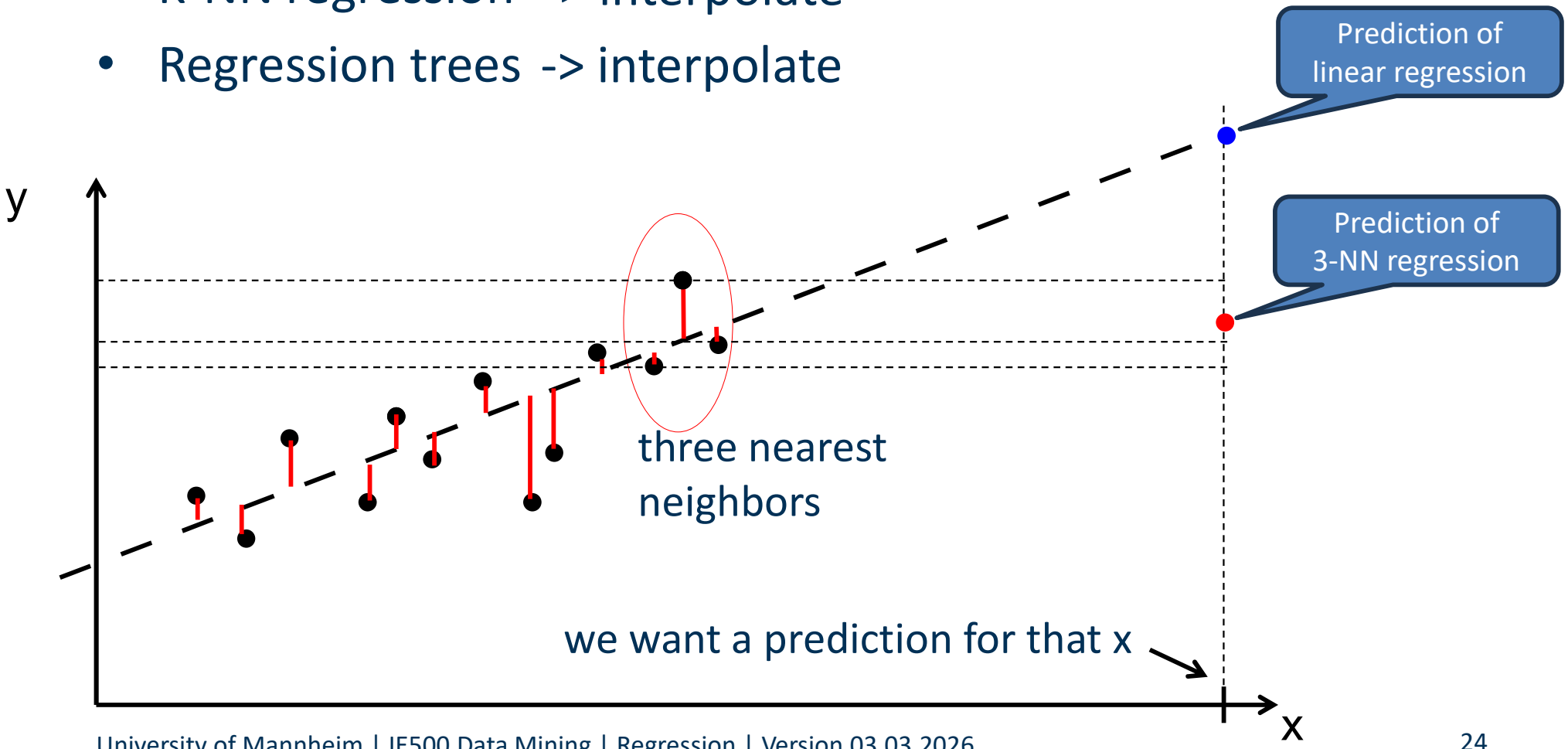
- Sometimes, however, extrapolation is interesting
  - how far will the sea level have risen by 2050?
  - how much will the temperature rise in my nuclear power plant?

http://i1.ytimg.com/vi/FVfiujbGLfM/hqdefault.jpg

Linear Regression vs. K-NN Regression

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Linear regression -> extrapolates
- K-NN regression -> interpolate
- Regression trees -> interpolate



Prediction Baseline

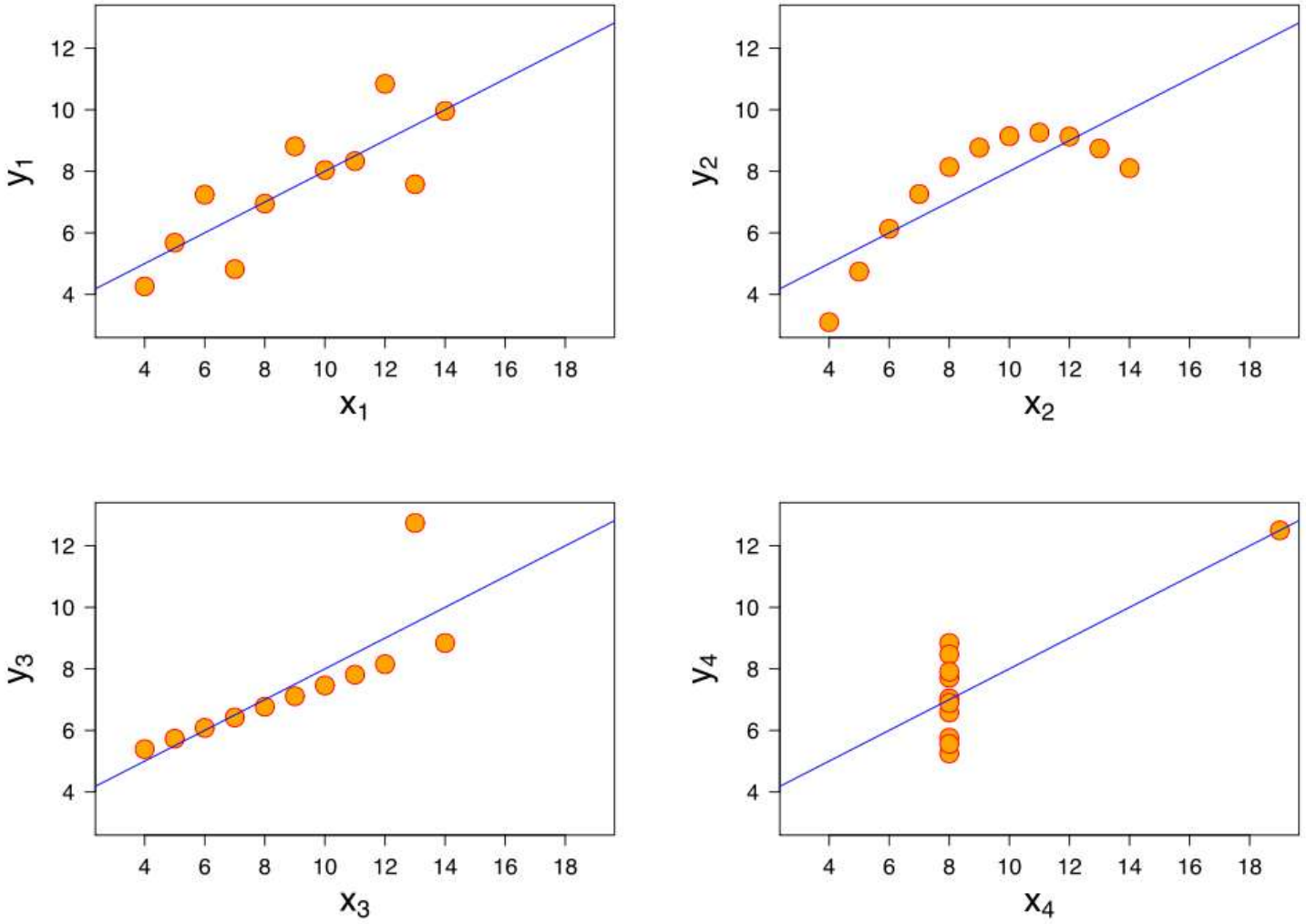
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- For classification:
  We predicted the most frequent label as baseline

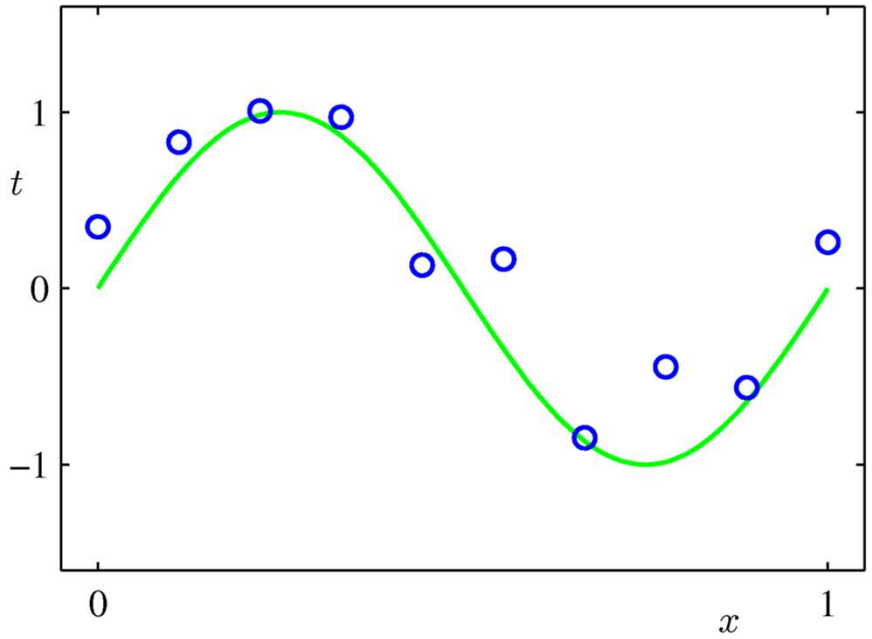
- For regression: We either use as baseline
  - mean value or
  - median or

Linear Regression Examples

UNIVERSITY OF MANNHEIM
Data and Web Science Group



…but what about non-linear Problems?



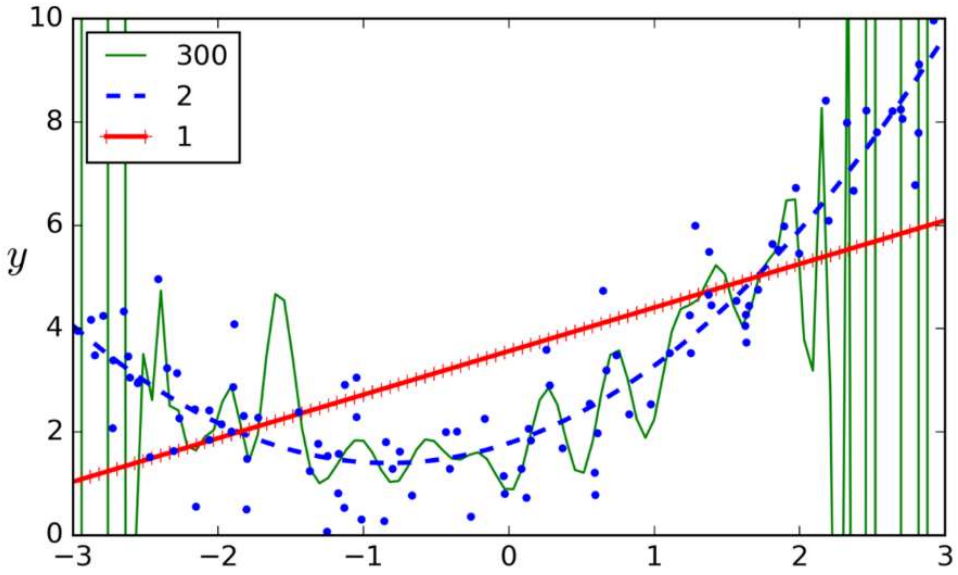
- One possibility is to apply transformations to the explanatory variables X within the regression function
  - e.g. log, exp, square root, square, etc.
  - polynomial transformation $y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$
- This allows linear regression techniques to model complex non-linear relationships in the data.

Polynomial Regression

UNIVERSITY OF MANNHEIM
Data and Web Science Group

$$\hat{y}(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d = \sum_{j=0}^{d} w_j x^j$$

- widely used extension of linear regression
- can also be fitted using the least squares method
- has tendency to over-fit training data for large degrees d
- Workarounds:
  - decrease d
  - increase amount of training data



Polynomial Regression in Python

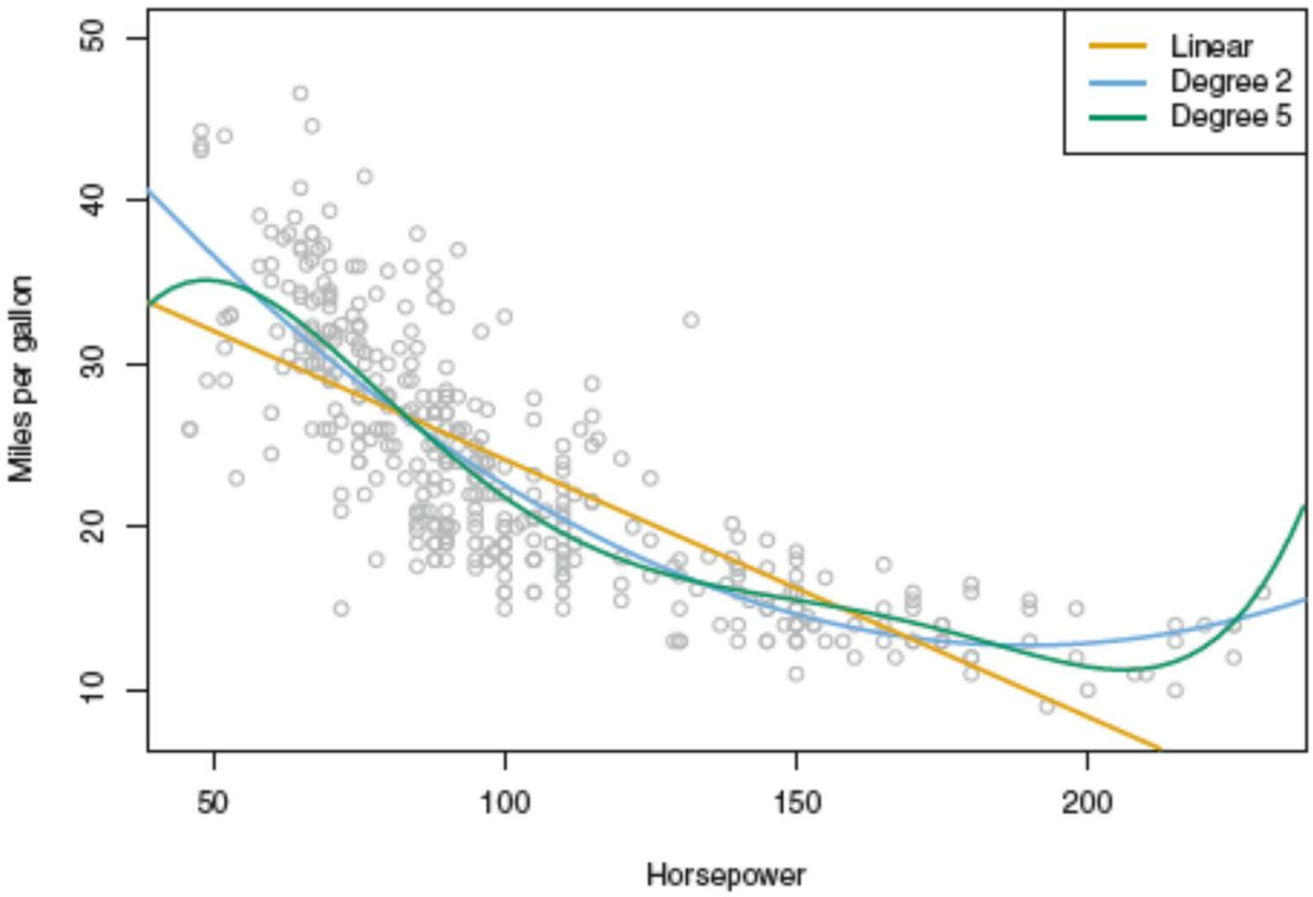
UNIVERSITY OF MANNHEIM
Data and Web Science Group

Python

```python
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression

poly_features = PolynomialFeatures(degree=2, include_bias=False).fit_transform(X, y)
estimator = LinearRegression()
estimator.fit(poly_features, y)
```

Polynomial Regression Overfitting Training Data



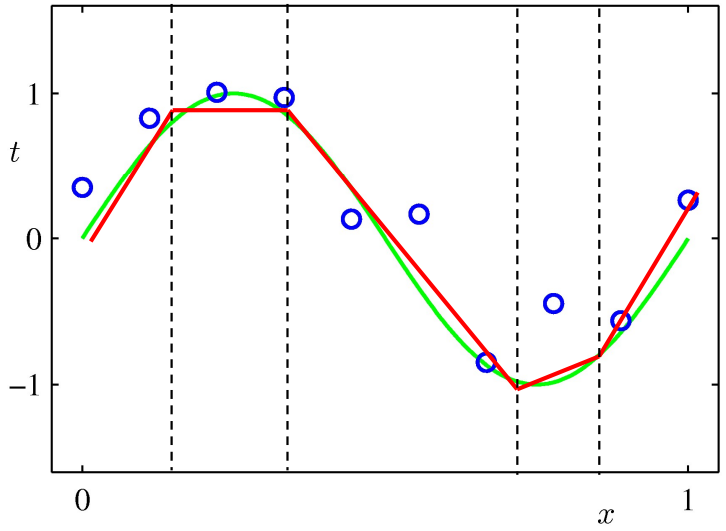
Overfitting often happens in **sparse regions**

- left and right side of green line
- workaround: Local regression

6. Local Regression

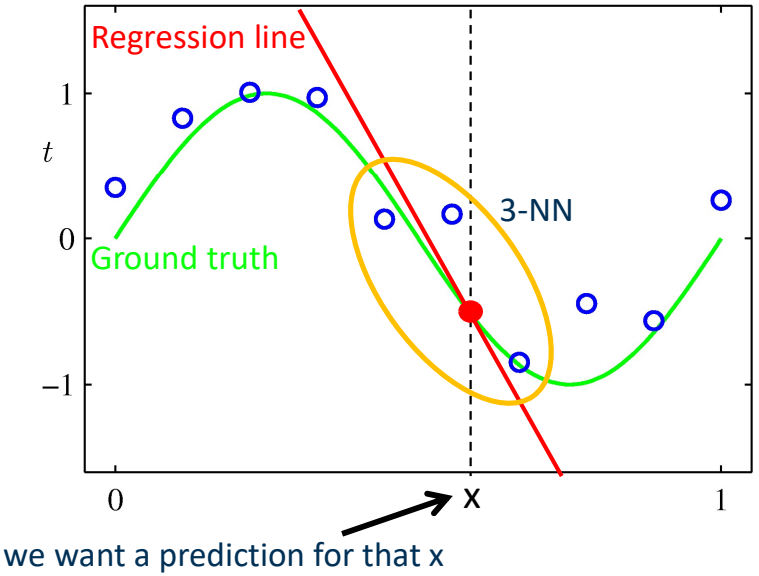
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Assumption: non-linear problems are approximately linear in local areas
- Idea:
  - use linear regression locally
  - only for the data point at hand (lazy learning)



Local Regression

- A combination of **k nearest neighbors** and **linear regression**
- Given a data point for prediction
  1. retrieve the k nearest neighbors
  2. learn a regression model using those neighbors
  3. use the learned model to predict y value



Discussion of Local Regression

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Advantage: fits non-linear models well
  - good local approximation
  - often better than pure k-NN

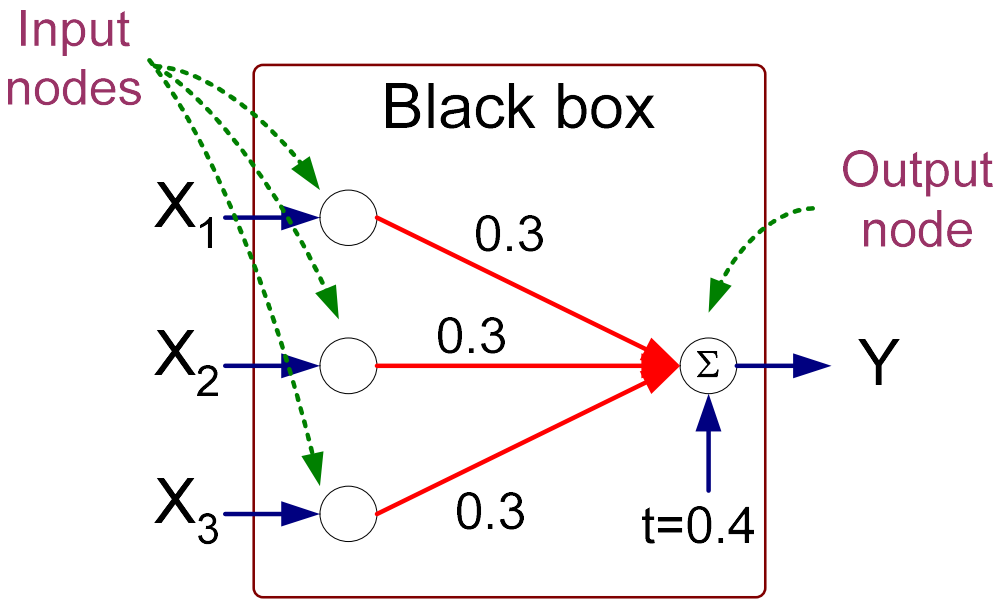
- Disadvantage
  - slow at runtime
  - as for each test example:
    1. find k nearest neighbors
    2. learn a local model
    3. use local model for prediction

7. Artificial Neural Networks (ANNs) for Regression

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- **Recap**: How did we use ANNs for classification?

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

$$\text{Where } I(z) = \begin{cases} 1 \text{ if } z \text{ is true} \\ 0 \text{ otherwise} \end{cases}$$

Artificial Neural Networks (ANNs) for Regression

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- The function $I(z)$ was used to separate the two classes:

$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0\ )$$

$$\text{Where } I(z) = \begin{cases} 1\ if\ z\ is\ true \\ 0\ otherwise \end{cases}$$

- However, we may simply use the inner formula to predict a numerical value (between 0 and 1):

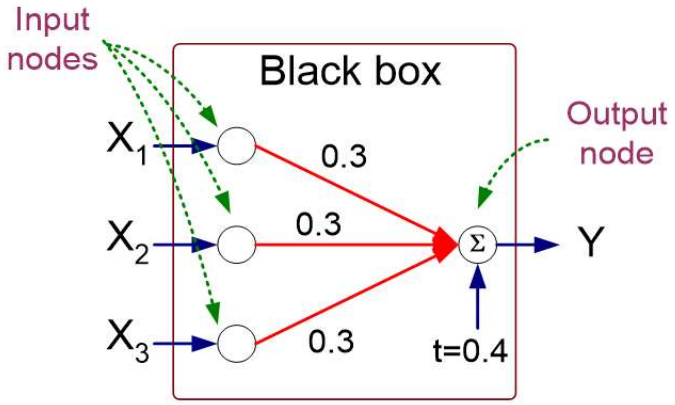
$$\hat{Y} = 0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4$$

ANNs for Regression

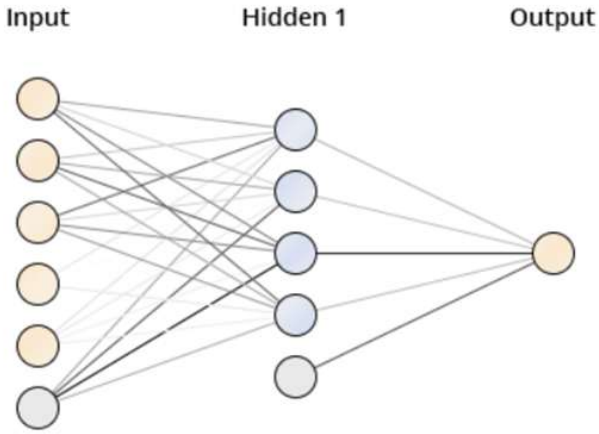
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Given that our formula is of the form

$$\hat{Y} = 0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4$$



- we can learn only linear models
  - i.e., the target variable is a linear combination the input variables
- More complex regression problems can be handled by using multiple hidden layers
  - this allows for approximating **arbitrary functions**
- Deep ANNs take this idea further by
  - employing millions of neurons
  - arranging them into specific network topologies
  - see Gemulla: Machine Learning + Deep Learning



- If you use ANNs be cautions about overfitting!

The Bias/Variance-Tradeoff

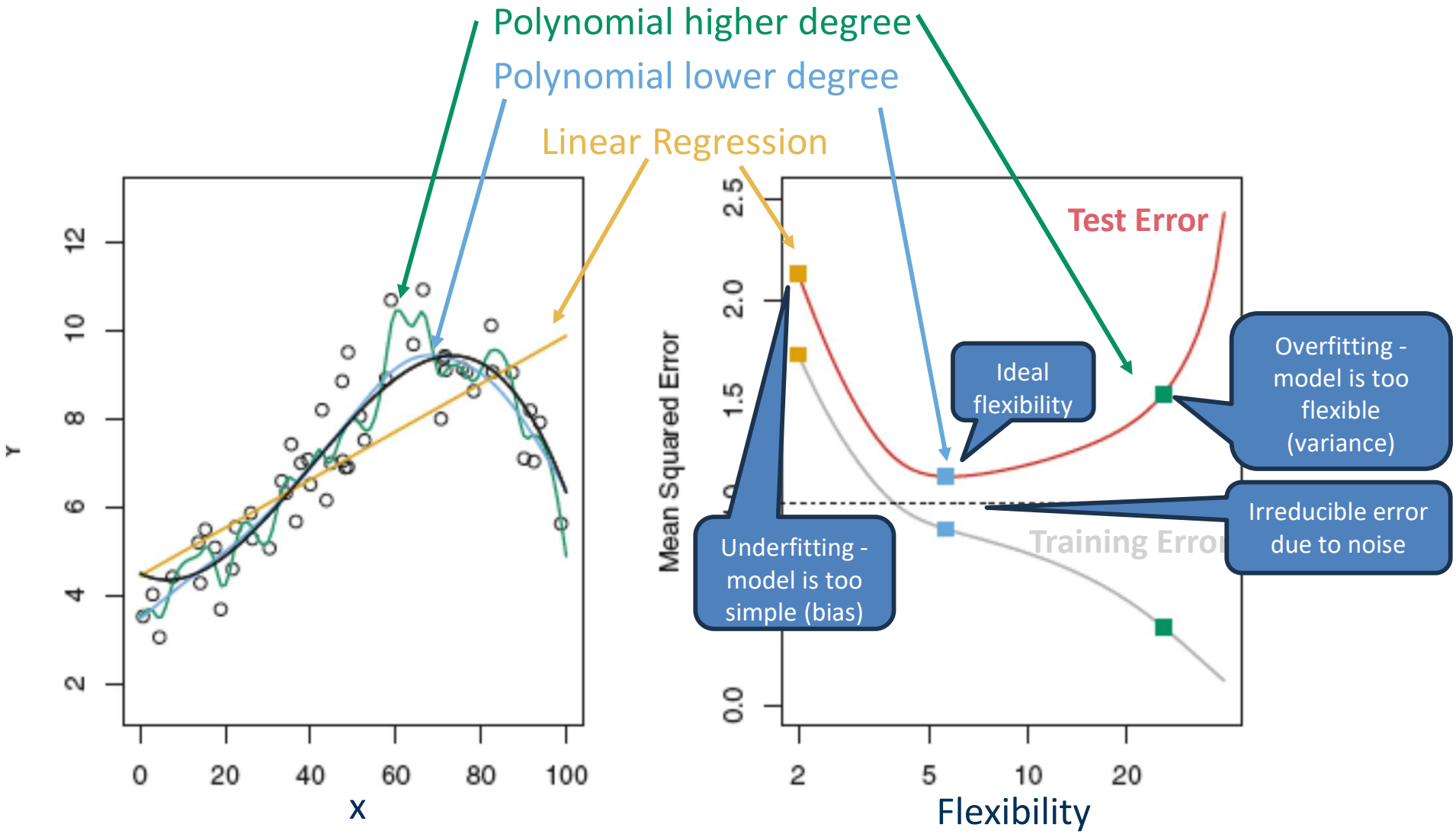
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- We want to learn regression as well as classification models that generalize well to **unseen data**
- The generalization error of any model can be understood as a sum of three errors:
  - **Bias**: Part of the generalization error due to wrong model complexity
    - Simple model (e.g. linear regression) used for complex real-world phenomena
    - Model thus **underfits** the training and test data
  - **Variance**: Part of the generalization error due to a model's excessive sensitivity to small variations in the training data
    - Models with high degree of freedom/flexibility (like polynomial regression models or deep trees) are likely to **overfit** the training data
  - **Irreducible Error**: Error due to noisiness of the data itself
    - To reduce this part of the error the training data needs to be cleansed (by removing outliers – only in training, fixing broken sensors)

8. The Bias/Variance-Tradeoff

UNIVERSITY OF MANNHEIM
Data and Web Science Group

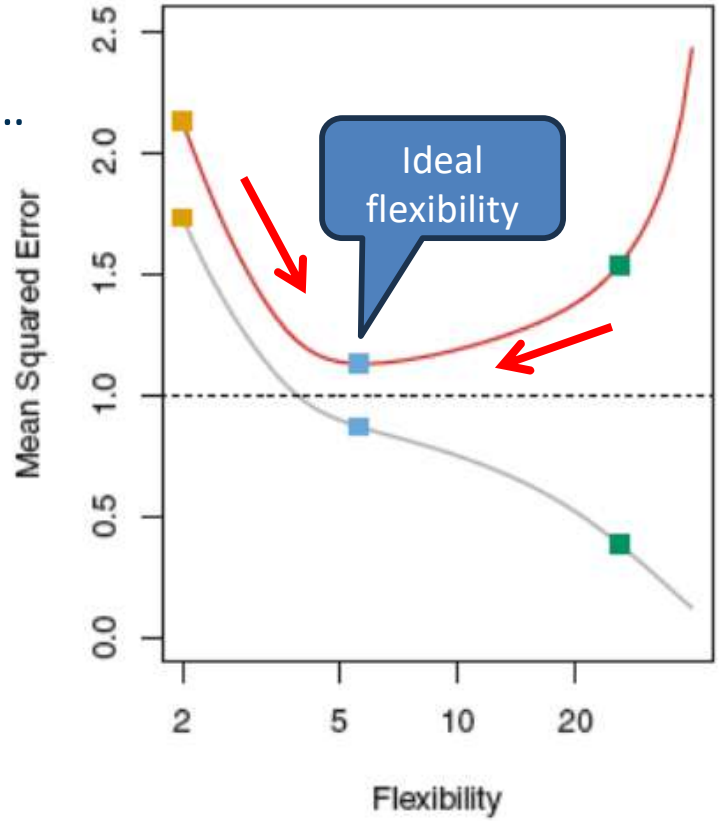
- Three models with different flexibility trying to fit a function



Learning Method and Hyperparameter Selection

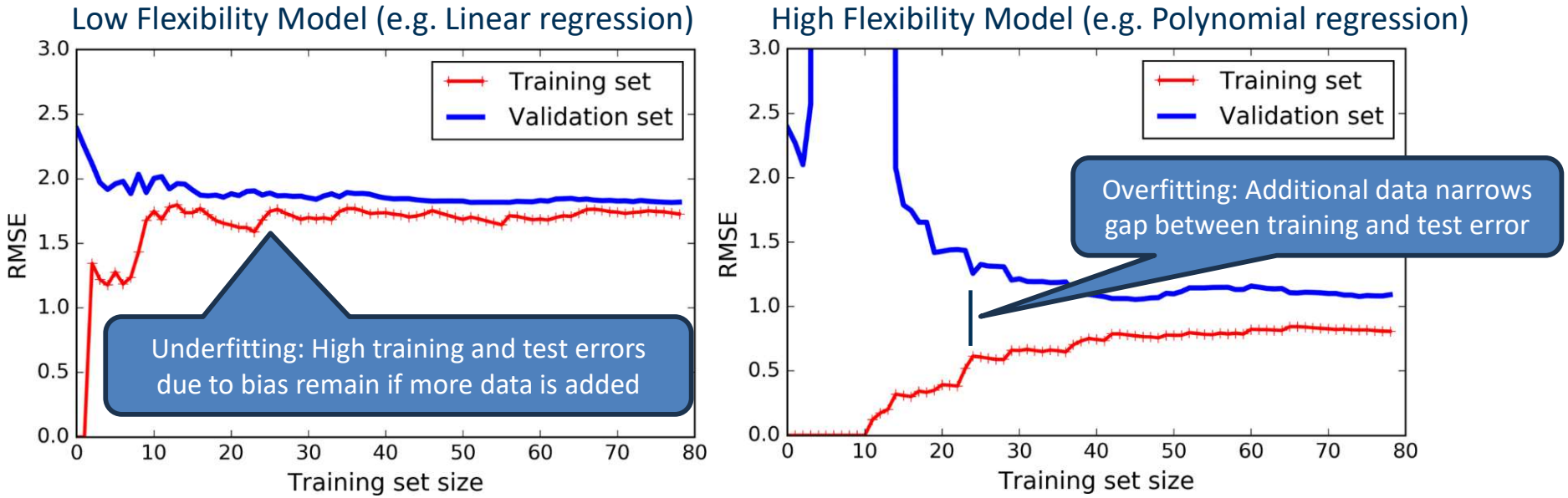
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- We try to find the **ideal flexibility** (bias/variance-tradeoff) by
  - testing different learning methods
    - linear regression, polynomial regression, …
    - decision Trees, ANNs, Naïve Bayes, …
  - testing different hyperparameters
    - degree of polynomial, ridge
    - max depth of tree, min examples branch
    - number of hidden layers of ANN
- But we have **three more options**:
  - increase the amount of training data
  - increase the interestingness of the data by including more corner cases
  - cleanse the training data



Learning Curves for Under- and Overfitting Models

- Visualize the training error and test error for **different training set sizes**



  - For overfitting models, the gap between training and test error can often be narrowed by adding more training data
  - Thus, having more training data also allows us to use models having a higher flexibility, e.g. Deep Learning

Summary

UNIVERSITY OF MANNHEIM
Data and Web Science Group

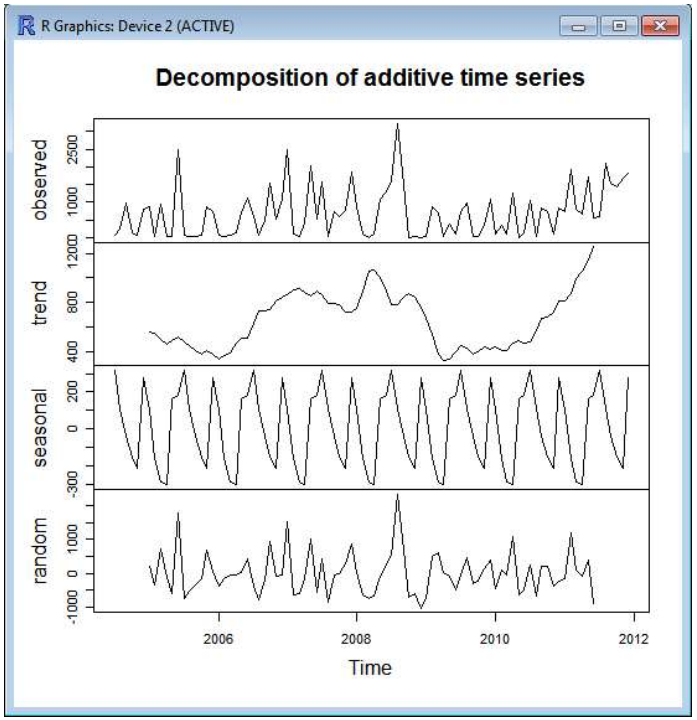
- Regression
  - predict numerical values instead of classes
- Model evaluation
  - metrics: (root) mean squared error, R squared
- Methods
  - K nearest neighbors, regression trees
  - linear regression, ridge regression
  - polynomial regression, local regression
  - artificial neural networks for regression
- For good performance on unseen data
  - choose learning method having the right flexibility (bias/variance-tradeoff)
  - use large quantities of interesting training data

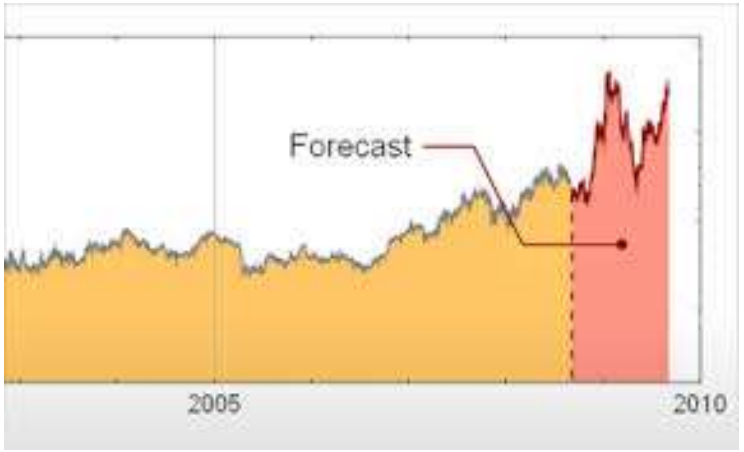
Online Lectures

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- This week additional material is about Time Series

Week	Wednesday (Offline Lecture, Room A5, B243)	Online Lecture (see Ilias Course)
11.02.2026	Introduction to Data Mining (PDF, 4 MB)	Nearest Centroids
18.02.2026	Classification 1 (PDF, 3 MB)	Ensembles
25.02.2026	Classification 2	Comparing Classifiers
04.03.2026	Regression	Time Series
11.03.2026	Preprocessing	Multi Modal Data
18.03.2026	Clustering and Anomalies	Hierarchical Clustering

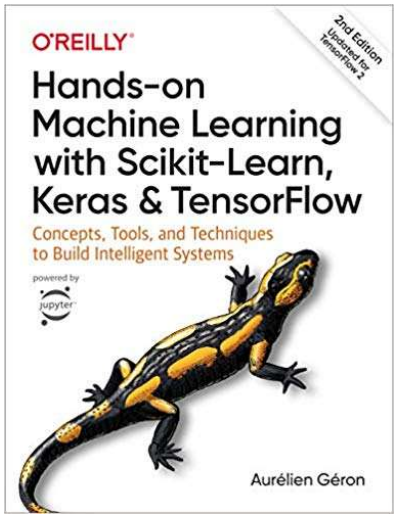




Literature for this Slideset

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Solving practical regression tasks using Python:
  - Geron: Hands-on Machine Learning - Chapter 4



- Sophisticated coverage of regression including theoretical background
  - James, Witten, et al.:
    An Introduction to Statistical Learning
    Chapters 3, 7, 8