Modeling

Main challenges

Prof. Dr. Jan Kirenz
HdM Stuttgart

Poor-quality data

Data preparation

Generalization

Sampling noise

Sampling bias

Outliers

Noisy data

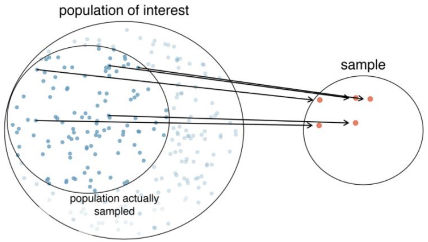
Missing data

We want our model to generalize well.

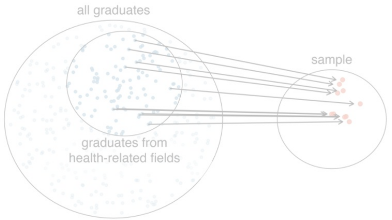
That means training data needs to be representative.

Possible issues:

1. dataset to small: sampling noise



2. sampling method flawed: sampling bias

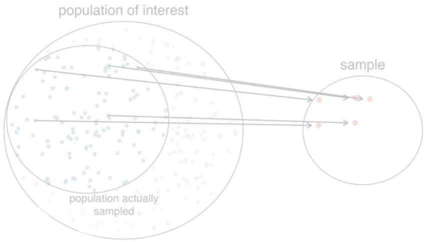


We want our model to generalize well.

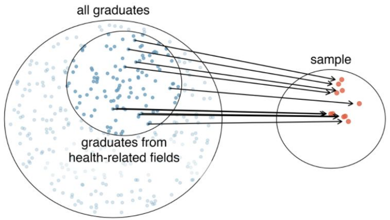
That means training data needs to be representative.

Possible issues:

1. dataset to small: sampling noise



2. sampling method flawed: sampling bias



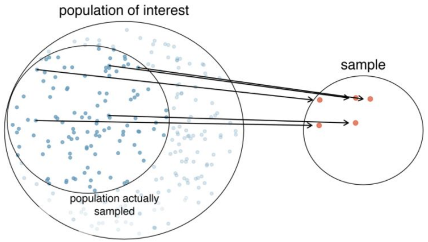
Source: Géron (2019)

We want our model to generalize well.

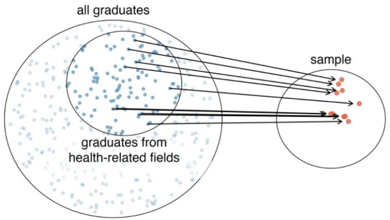
That means training data needs to be representative.

Possible issues:

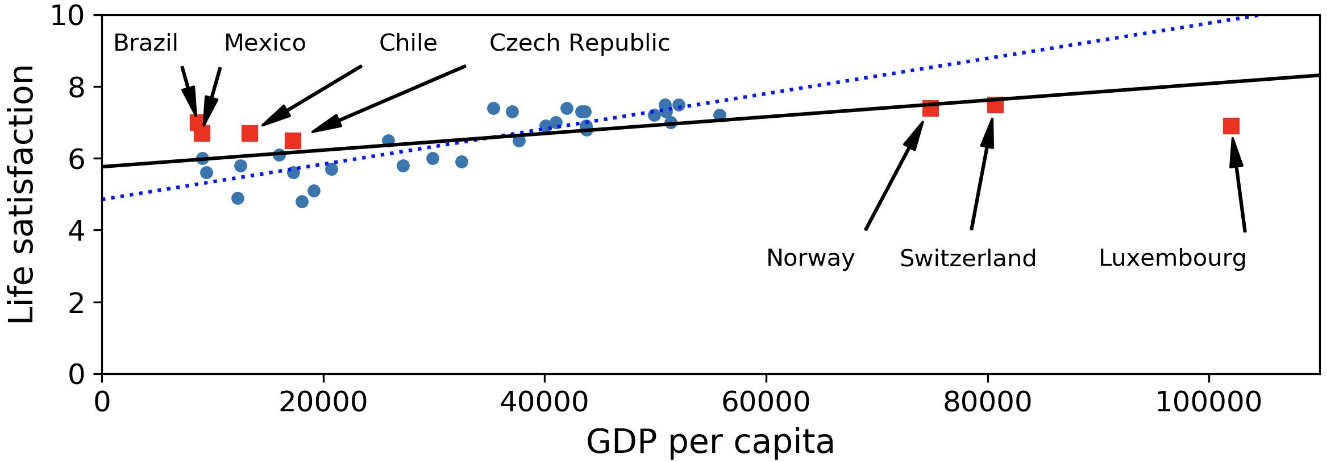
1. dataset to small: sampling noise



2. sampling method flawed: sampling bias

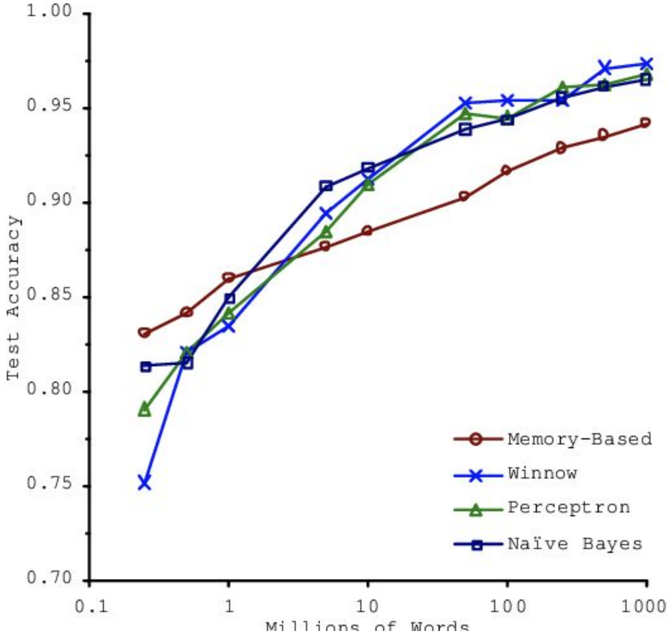


Source: Géron (2019)



Linear regression with a more representative data sample

Banko, M., & Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics* (pp. 26-33).



The Importance of data versus algorithms

Poor-quality data

Data preparation:

1. Get rid of outliers
2. Check for noise (e.g., poor quality measurement)
3. Handle missing data

Poor-quality data

Data preparation:

1. Get rid of outliers

2. Check for noise (e.g., poor quality measurement)

3. Handle missing data

Poor-quality data

Data preparation:

1. Get rid of outliers

2. Check for noise (e.g., poor quality measurement)

3. Handle missing data

Irrelevant features

Feature engineering

Feature selection

Feature extraction

Feature creation

Irrelevant Features

Feature engineering:

1. Feature selection (select the most useful features)

2. Feature extraction (combine existing features)

3. Feature creation (make new features)

Source: Géron (2019)

Irrelevant Features

Feature engineering:

1. Feature selection (select the most useful features)

2. Feature extraction (combine existing features)

3. Feature creation (make new features)

Irrelevant Features

Feature engineering:

1. Feature selection (select the most useful features)

2. Feature extraction (combine existing features)

3. Feature creation (make new features)

Source: Géron (2019)

Irrelevant Features

Feature engineering:

1. Feature selection (select the most useful features)

2. Feature extraction (combine existing features)

3. Feature creation (make new features)

Source: Géron (2019)

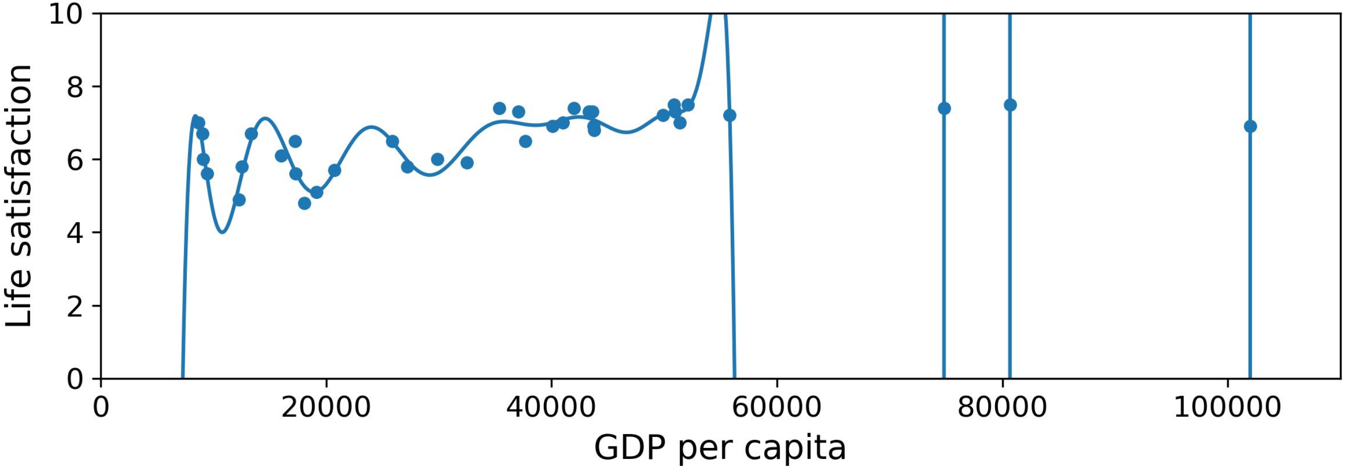
Overfitting

Model with high variance

Regularization

Hyperparameters

Noise reduction



(Too) Complex model: Polynomial Linear Regression

Source: Géron (2019)

Overfitting the training data

The model performs well on the training data, but it does not generalize well

- Happens if the model is **too complex**
- Model detects patterns in the noise
- This means the variance is high

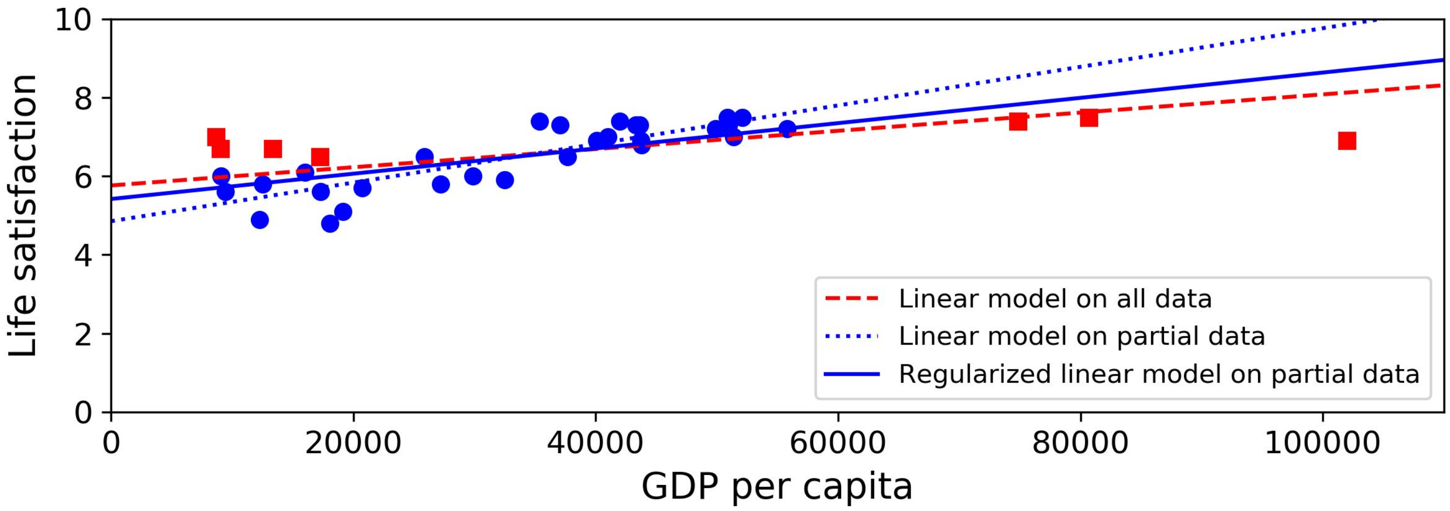
Source: Géron (2019)

Solution 1: simplify the model

A. Reduce number of features

B. Use fewer parameters

C. Constrain the model (regularization)



Regularization reduces the risk of overfitting

Source: Géron (2019)

Regularization

The amount of regularization can be controlled by a hyperparameter

- A hyperparameter is a parameter of the algorithm (not of the model)
- Must be set prior to training
- Remains constant during training

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 = \mathrm{RSS}$$

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

Source: Géron (2019)

Regularization

The amount of regularization can be controlled by a hyperparameter

- A hyperparameter is a parameter of the algorithm (not of the model)
- Must be set prior to training
- Remains constant during training

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 = \mathrm{RSS}.$$

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

Source: Géron (2019)

Regularization

The amount of regularization can be controlled by a hyperparameter

- A hyperparameter is a parameter of the algorithm (not of the model)
- Must be set prior to training
- Remains constant during training

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 = \mathrm{RSS}.$$

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \boxed{\lambda\sum_{j=1}^{p}|\beta_j|} = \mathrm{RSS} + \boxed{\lambda\sum_{j=1}^{p}|\beta_j|.}$$

Source: Géron (2019)

Solution 2: reduce noise in the data

A. Fix data errors

B. Remove outliers

Solution 3: more data

A. Get more training data

Underfitting

Model with high bias

Bias

More parameters

Better features

Reduce constraints

Underfitting the data: Bias

Model is too simple to learn the underlying structure of the data

- Predictions will be inaccurate
- This is called bias

1) More parameters

Select a more powerful model, with more parameters

2) Better Features

Use better features in your model (feature engineering)

3) Reduce constraints

Reduce the constraints on the model (e.g., reduce the regularization hyperparameter)