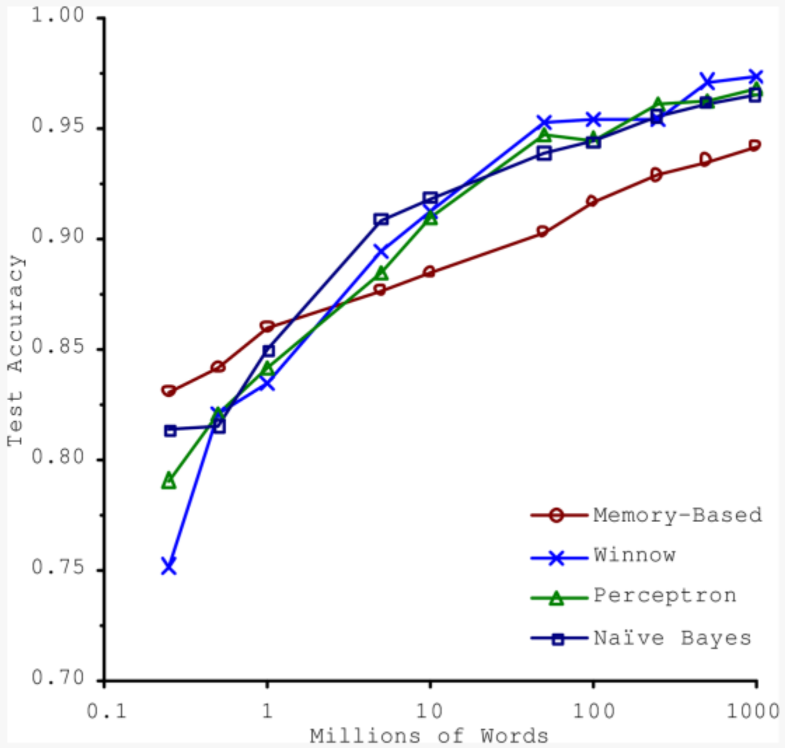
3.
Challenges of Machine Learning

1. Insufficient Quantity of Training Data

- ML systems cannot learn as fast as toddlers yet.
  - Even for very simple problems you typically need thousands of examples.
  - For complex problems such as image or speech recognition you may need millions of examples.
- Some researchers believe that data matters more than algorithms for complex problems.
  - Halevy, A., Norvig, P. and Pereira, F., 2009. The unreasonable effectiveness of data. *IEEE intelligent systems*, *24*(2), pp.8-12.

The unreasonable effectiveness of data

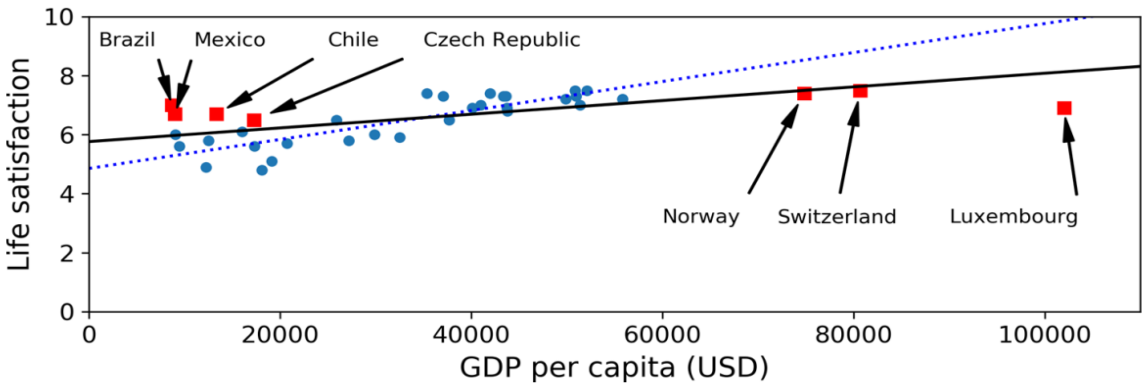
- Microsoft researchers showed ML algorithms, including fairly simple ones, performed almost identically well on a complex problem of natural language disambiguation once they were given enough data.



The importance of data versus algorithms

2. Non-representative Training Data

- The training set must be representative of the cases you want to generalize to.
- *Sampling noise:* very small sample results in non-representative data.
- *Sampling bias:* very large samples can be non-representative if the sampling method is flawed.

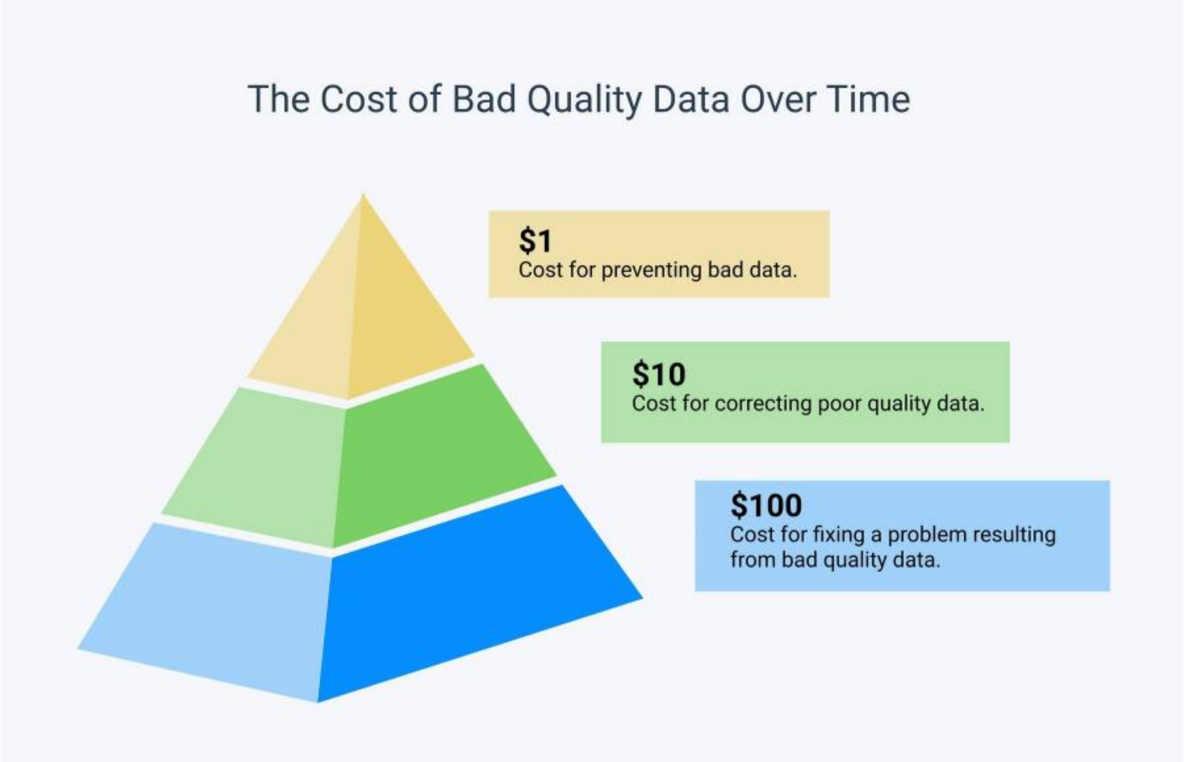


3. Poor-Quality Data

- If training data is full of errors, outliers, and noise (e.g., due to poor-quality measurements), it will make it harder for the system to detect the underlying patterns.
  - garbage in, garbage out

- It is often well worth the effort to spend time cleaning up your training data.
  - It is estimated that data scientists spend about **80%** of their time cleaning data.

Cost of Poor-Quality Data

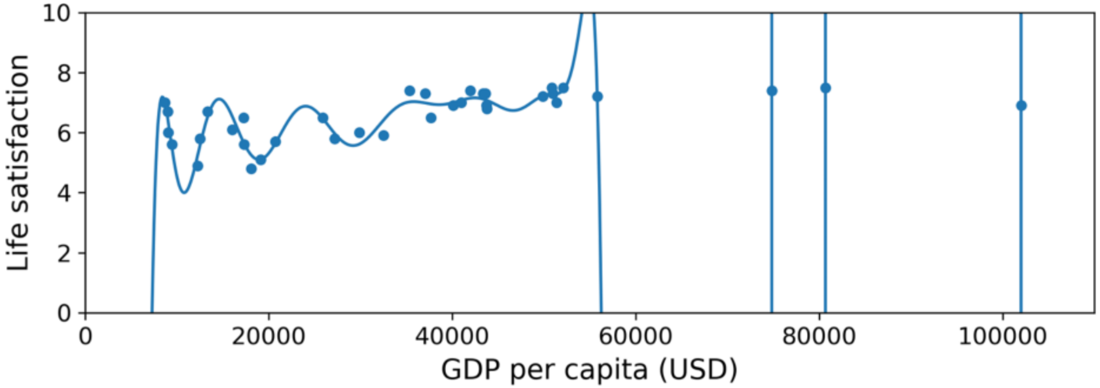


4. Irrelevant Features

- ML systems can learn if the training data contains enough relevant features and not too many irrelevant ones.
- Feature engineering: coming up with a good set of features to train on
  - *Feature selection:* selecting the most useful features to train on among existing features.
  - *Feature extraction:* combining existing features to produce a more useful one.
  - Creating new features by gathering new data.

5. Overfitting the Training Data

- *Overfitting*: the ML model performs well on the training data, but it does not generalize well.
- A high-degree polynomial life satisfaction model strongly overfits the training data.
- It performs much better on the training data than the simple linear model, but its predictions are not trustworthy.

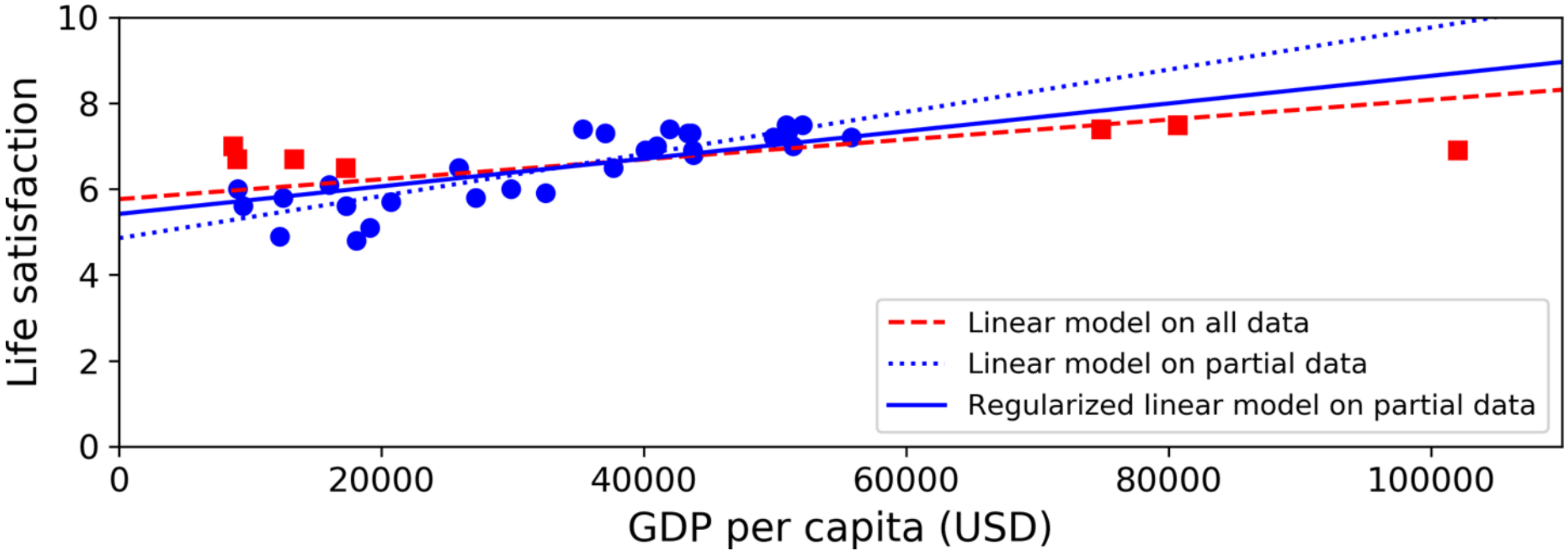


Solutions for Overfitting

- Simplify the model:
  - by selecting a model with fewer parameters (e.g., a linear model rather than a high-degree polynomial model)
  - by reducing the number of features in the training data
  - by constraining the model (*regularization*)
- Gather more training data.
- Reduce the noise in the training data (e.g., fix data errors and remove outliers).

Regularization

- The linear model has two parameters, θ_0 and θ_1 (two degrees of freedom).
- If we allow the algorithm to modify θ_1 but we force it to keep it small, then the it will effectively have between one and two degrees of freedom.



Hyperparameters

- The amount of regularization to apply during learning can be controlled by a *hyperparameter*.
- A hyperparameter is a parameter of a learning algorithm (not of the model).
    - It must be set prior to training and remains constant during training.
- In a linear model, if you set the regularization hyperparameter to a very large value, you will get an almost flat model (a slope close to zero).

6. Underfitting the Training Data

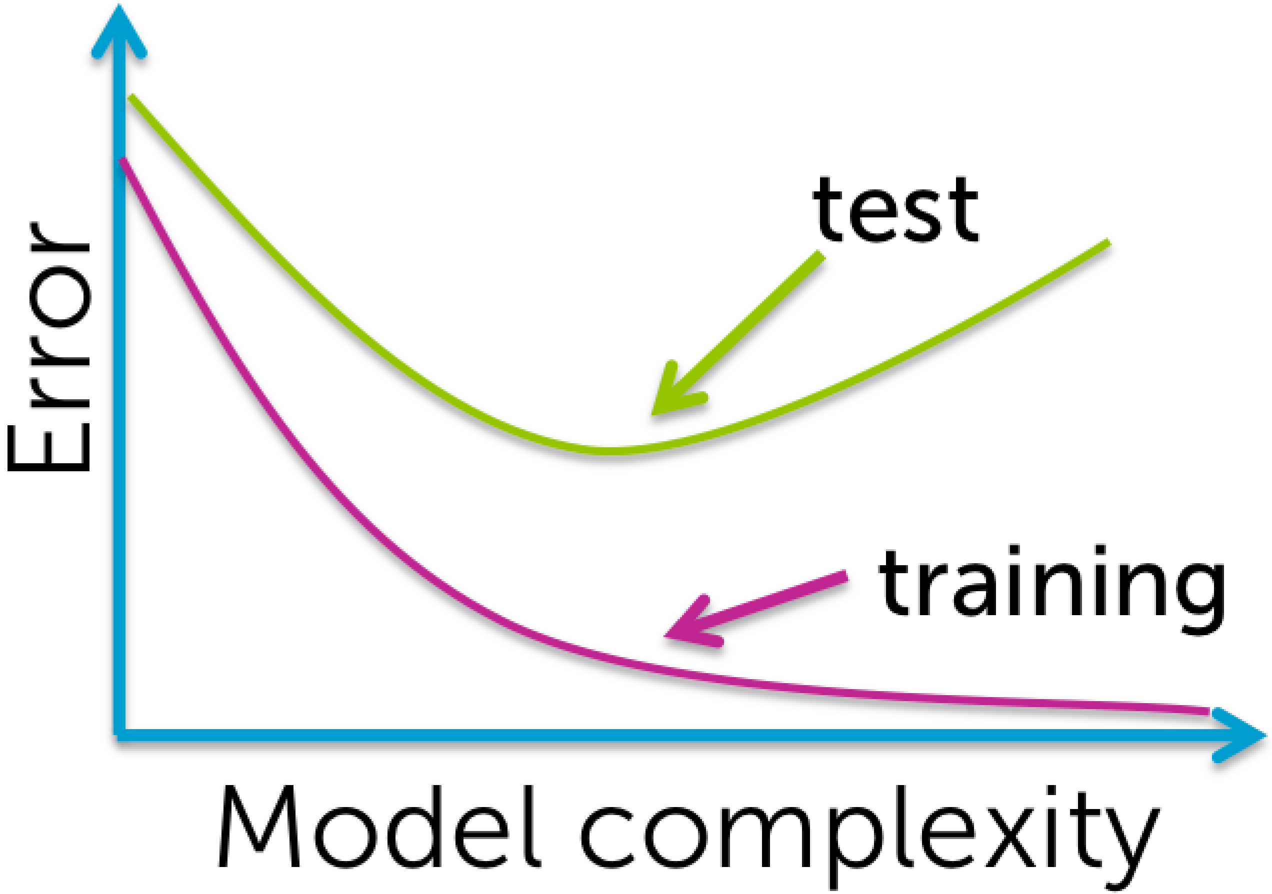
- *Underfitting* occurs when your model is too simple to learn the underlying structure of the data.

- Solutions for undefitting:
  - Select a more powerful model, with more parameters.
  - Feed better features to the learning algorithm (feature engineering).
  - Reduce the constraints on the model (e.g., reduce the regularization hyperparameter).

4.
Testing and Validating

Generalization Error

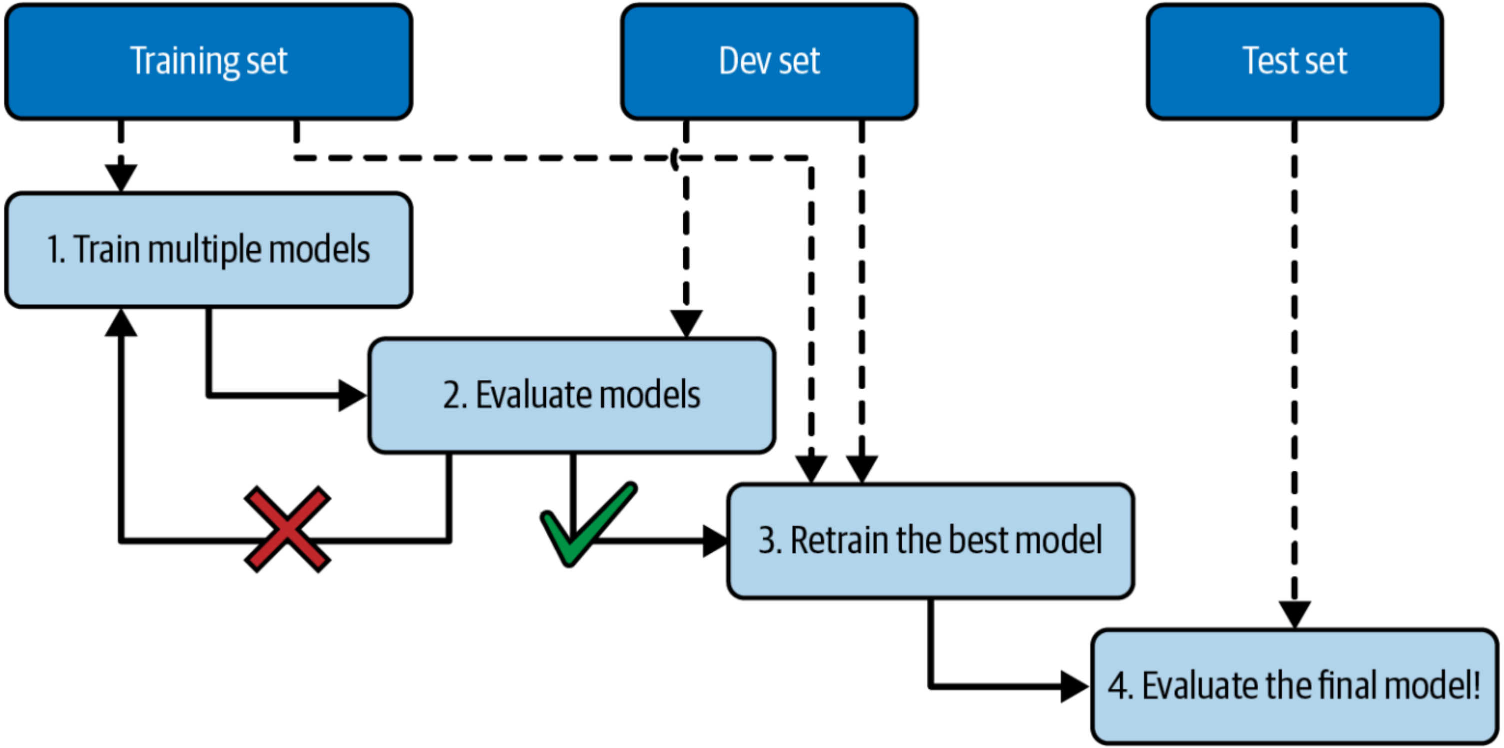
- Split your data into two sets: the *training set* and the *test set*.
- The error rate on new cases is called the *generalization error* (or *out-of-sample error*).
    - This value tells you how well your model will perform on instances it has never seen before.
- If the training error is low (i.e., your model makes few mistakes on the training set) but the generalization error is high, it means that your model is overfitting the training data.


Error
test
training
Model complexity

Model Selection

- We train different models (e.g. two types of model and 100 different values for the hyperparameter) and compare how well they generalize using the test set.
- Problem: we measured the generalization error multiple times on the same test set, and adapted the model and hyperparameters to produce the best model *for that particular set.*
- *Holdout validation*: you hold out part of the *training set* to evaluate several candidate models and select the best one.
  - The new held-out set is called the *validation set* (or the *development set*, or *dev set*).

Holdout Validation



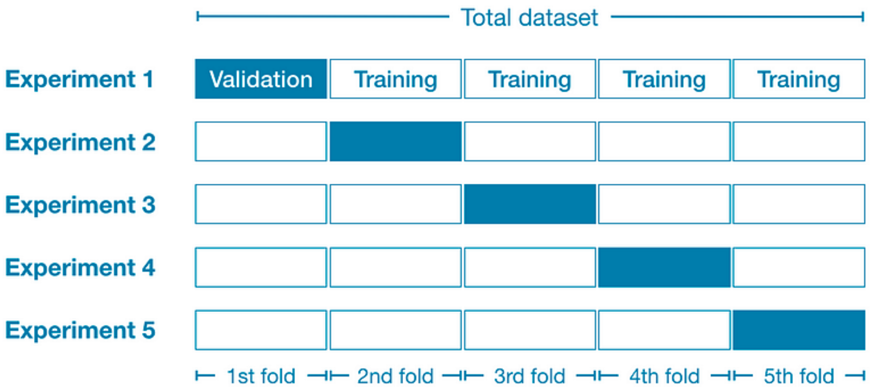
Cross Validation

- Small validation set: model evaluations will be imprecise and you may end up selecting a suboptimal model by mistake.

- Large validation set: the remaining training set will be much smaller than the full training set.
    - The final model is trained on the full training set, it is not ideal to compare candidate models trained on a smaller training set.

Cross Validation

- Solution: perform repeated *cross-validation*, using many small validation sets.
  - Each model is evaluated once per validation set after it is trained on the rest of the data.
  - By averaging out all the evaluations of a model, you get a more accurate measure of its performance.



No Free Lunch Theorem

- A model is a simplified version of the observations.
- The simplifications are meant to discard the details that are unlikely to generalize to new instances.
    - To decide what data to discard and what data to keep, you must make *assumptions*.
- If you make absolutely no assumption about the data, then there is no reason to prefer one model over any other.
- There is no model that is *a priori* guaranteed to work better.