Challenges of Machine Learning

Challenges

- Main task is to select a learning algorithm and train it on some data
  - Bad Data
  - Bad Algorithm

Bad Data

- **Insufficient Quantity of Training Data**
  - The child is able to recognize apples in all sorts of colors and shapes
  - Machine Learning takes a lot of data for most Machine Learning algorithms to work properly
  - Very simple problems you typically need thousands of examples, and for complex problems such as image or speech recognition you may need millions of examples
  - (unless you can reuse parts of an existing model).

UNREASONABLE EFFECTIVENESS OF DATA

- Different Machine Learning algorithms, including fairly simple ones, performed almost identically well on a complex problem of natural language disambiguation once they were given enough data

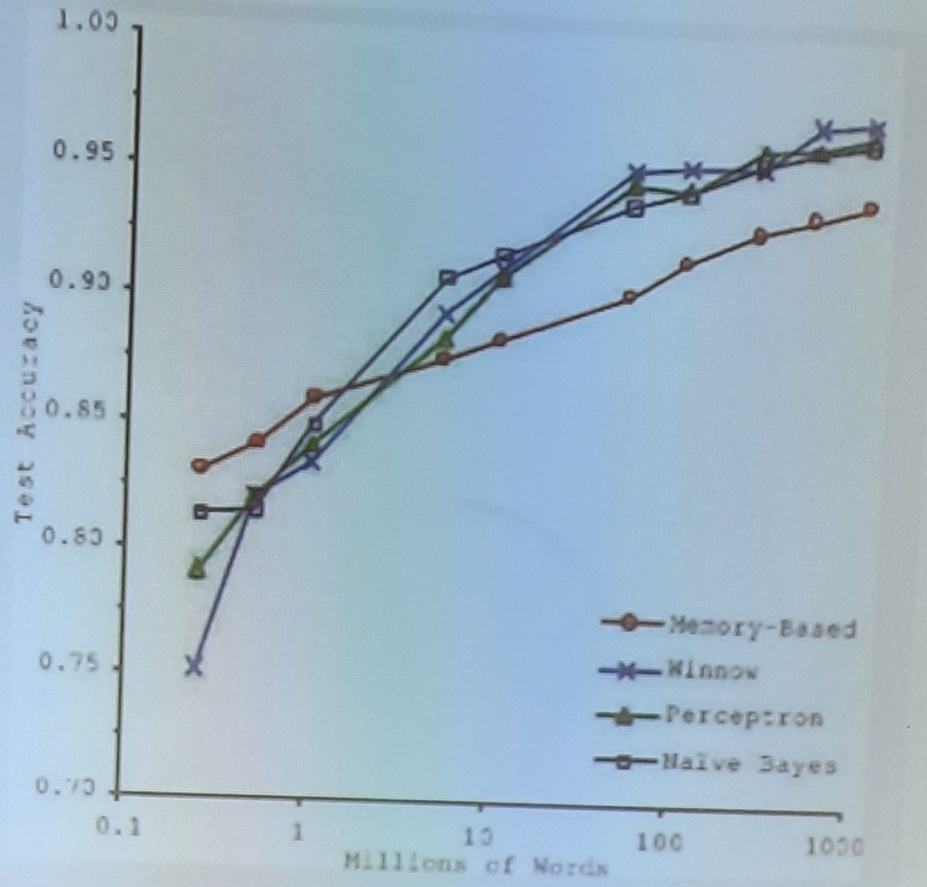


Figure 1-20. The importance of data versus algorithms[9]

Nonrepresentative Training Data

- In order to generalize well, it is crucial that your training data be representative of the new cases you want to generalize to. This is true whether you use instance-based learning or model-based learning

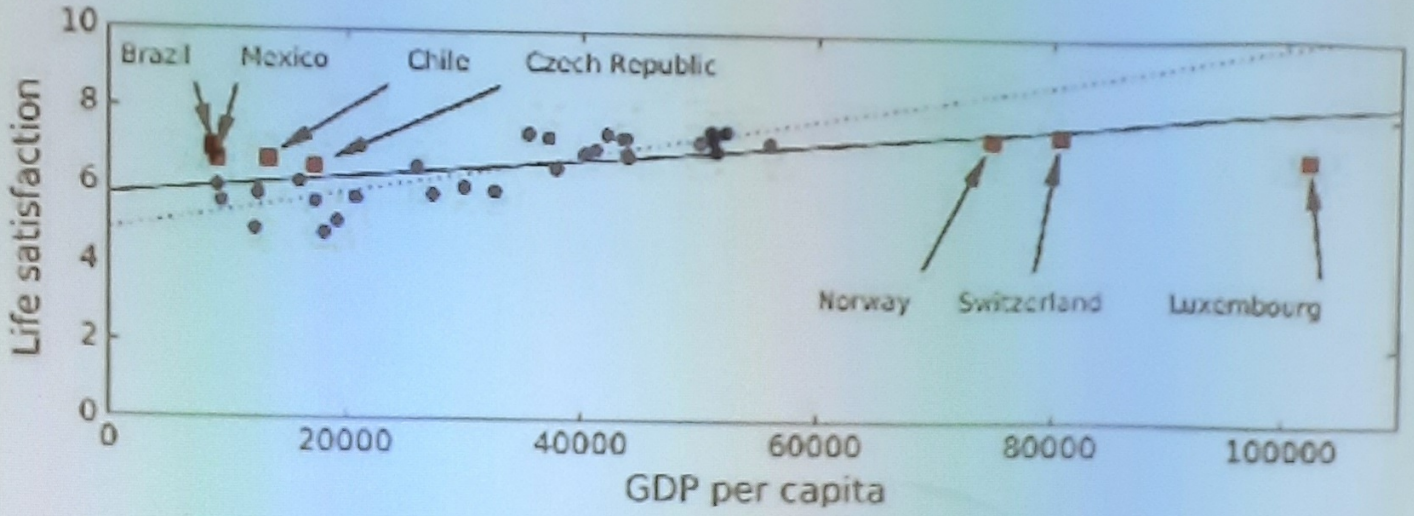


Figure 1-21. A more representative training sample

If you train a linear model on this data, you get the solid line, while the old model is represented by the dotted line. As you can see, not only does adding a few missing countries significantly alter the model

Poor-Quality Data

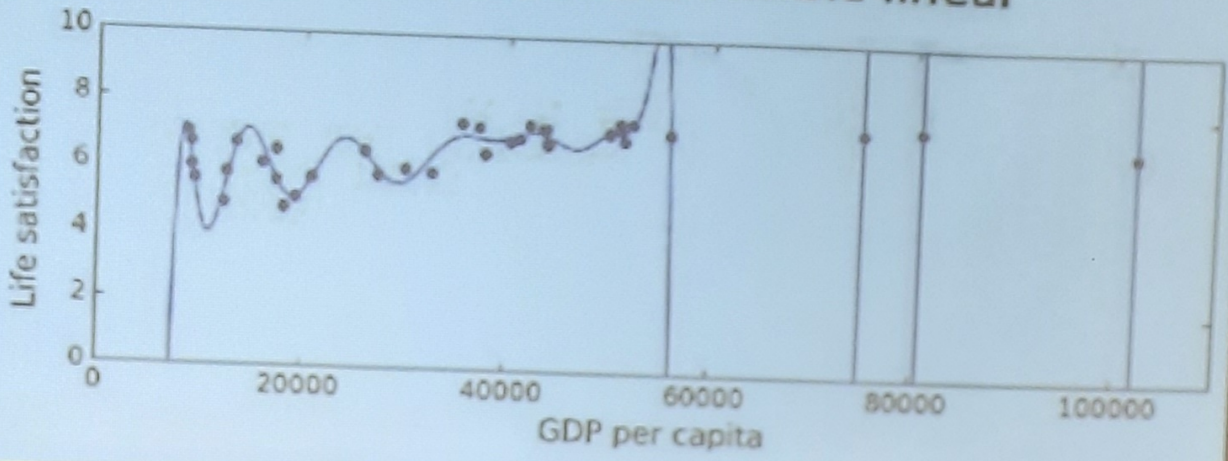
- Training data is full of errors, outliers, and noise (e.g., due to poor-quality measurements)
- Harder for the system to detect the underlying patterns,
- Most data scientists spend a significant part of their time doing just that. For example:
  - If some instances are clearly outliers,
    - **simply discard them**
    - **fix the errors manually**.
  - If some instances are missing a few features (e.g., 5% of your customers did not specify their age),
    - **ignore this attribute altogether**,
    - **ignore these instances**,
    - **fill in the missing values** (e.g., with the median age), or
    - train one model with the feature and one model without it.

Irrelevant Features

- Enough relevant features and not too many irrelevant
- Success of a Machine Learning project is coming up with a good set of features to train on
- *Feature Engineering;*
  - *Feature selection*: selecting the most useful features to train on among existing features.
    Feature extraction: combining existing features to produce a more useful one (as we saw earlier,
  - dimensionality reduction algorithms can help).
  - Creating new features by gathering new data.

Overfitting the Training Data

- *Overfitting*: it means that the model performs well on the training data, but it does not generalize well
- An example of a high-degree polynomial life satisfaction model that strongly overfits the training data.
- it performs much better on **the training data** than the simple linear model



Underfitting the Training Data

- *Underfitting* is the opposite of overfitting: it occurs when your model is too simple to learn the underlying structure of the data.
- Reality is just more complex than the model, so its predictions are bound to be inaccurate, even on the training examples.
- The main options to fix this problem are:
  - Selecting a more powerful model, with more parameters
  - Feeding better features to the learning algorithm (feature engineering)
  - Reducing the constraints on the model (e.g., reducing the regularization hyperparameter)

Underfitting the Training Data

- *Underfitting* is the opposite of overfitting: it occurs when your model is too simple to learn the underlying structure of the data.
- Reality is just more complex than the model, so its predictions are bound to be inaccurate, even on the training examples.
- The main options to fix this problem are:
  - Selecting a more powerful model, with more parameters
  - Feeding better features to the learning algorithm (feature engineering)
  - Reducing the constraints on the model (e.g., reducing the regularization hyperparameter)