上海科技大学
ShanghaiTech University

CS286 AI for Science and Engineering

Lecture 2: Machine Learning Landscape

Jie Zheng (郑杰)

PhD, Associate Professor

School of Information Science and Technology (SIST), ShanghaiTech University

Fall Semester, 2020

立志成才 报国裕民

Outline

- What is machine learning?
- Why use machine learning?
- Types of machine learning systems
- Performance measures
- Main challenges

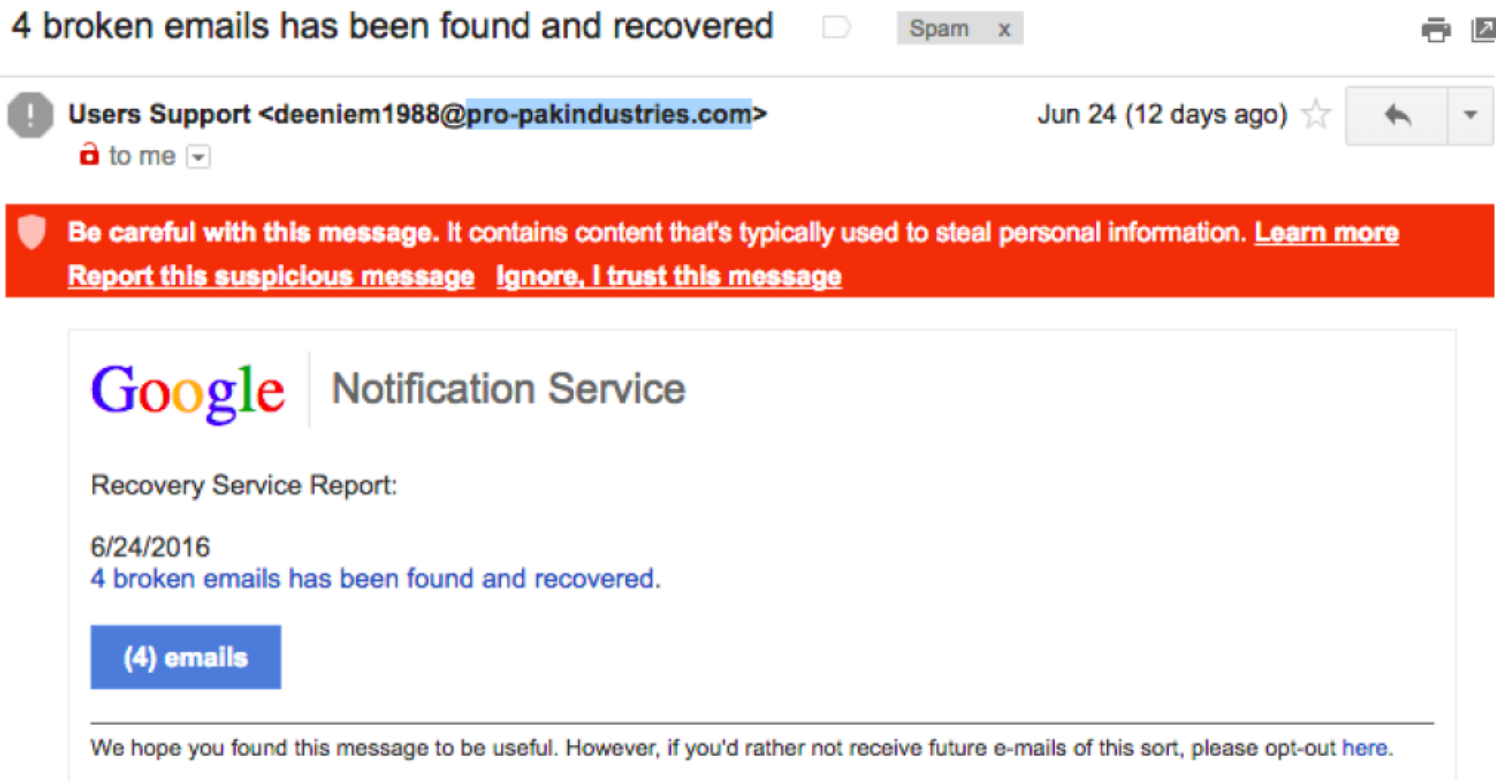
What is machine learning?

What is machine learning?

- **Machine learning** is the science (and art) of programming computers so they can *learn from data* :
  - General view: Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed (Arthur Samuel, 1959)
  - Engineering-oriented view: A computer program is said to **learn from experience E with respect to some task T and some performance measure P**, if its performance on T, as measured by P, improves with experience E (Tom Mitchell, 1997)
- A common theme is to solve a prediction problem:
  - Given an **input** x
  - Predict an "appropriate" **output** y
- Let us start with a canonical example

Example: Spam filtering

- **Input**: Incoming emails



- **Output**: "SPAM" or "NOT SPAM"
- A **binary classification** problem, i.e. having only 2 possible outputs

Example: Spam filtering (continued)

- **Task T**: To flag spam for new emails
- **Experience E**: The training data
  - Examples of spam emails (e.g. flagged by users)
  - Examples of regular (nonspam) emails
- **Performance measure P**: To be defined
  - e.g. **accuracy** (ratio of correctly classified emails)

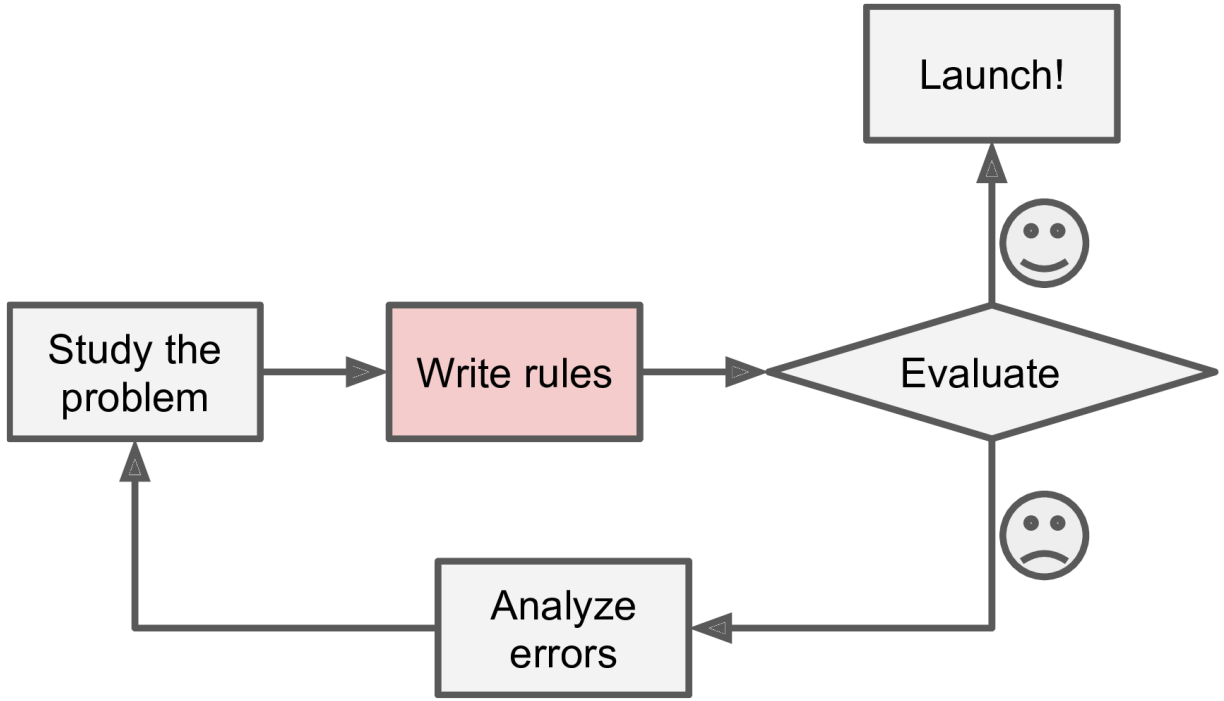
The prediction function

- A **prediction function** takes input x and produces an output y
- We look for prediction functions that solve particular problems
- Machine learning techniques aim to find the **best** prediction functions

Why use machine learning?

Traditional rule-based approaches

- A traditional method for spam filtering:
  - Study what spam typically looks like
  - A spam email usually contains some special key words in the subject (e.g. “4U”, “credit card”, …), encoded as rules
  - Write and test a spam detection algorithm based on the rules



Traditional rule-based approaches

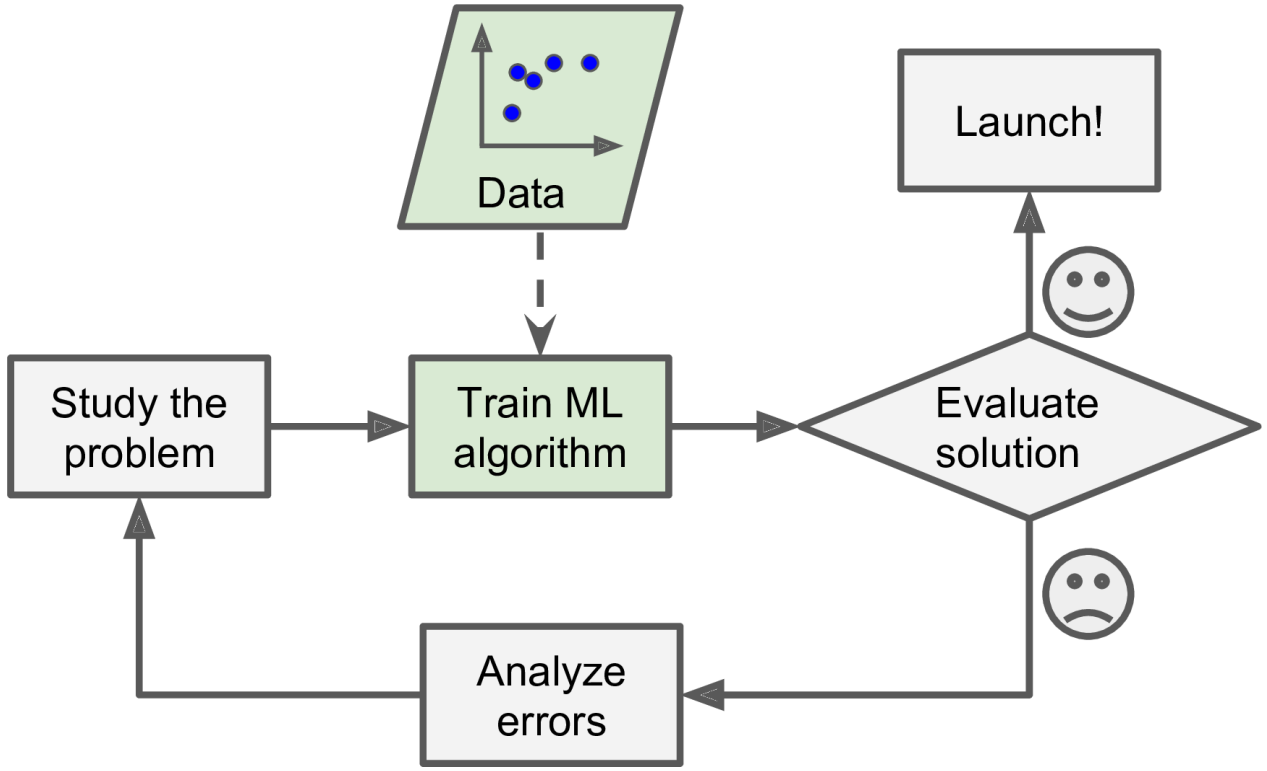
- Issues with rule-based approaches:
  - Very labor intensive to build a system
  - End up with a long list of rules, which is hard to maintain
  - Rules may work very well for specific areas they cover, but cannot naturally handle uncertainty

Machine learning approach

- A machine "learns" on its own without manually designed rules
- We provide "training data", i.e. many examples of (input x, output y), e.g.
  - A set of emails, and whether or not each is SPAM
  - A set of images, and whether or not each has a bird
- Learning from training data of this form is called **supervised learning**

Machine learning algorithm

- A machine learning algorithm:
  - **Input**: Training data
  - "Learn" from the training data
  - **Output**: A predicting function that produces output y given x



Strengths of machine learning

- Usually simplify code and perform better for complex problems
- Machine learning systems can
  - adapt to new data
  - help further get insights about complex problems and large amounts of data

Types of machine learning systems

Supervised/unsupervised learning

- **Supervised learning:** The training data fed into algorithms include desired solutions, i.e. labels, such as:
  - Spam classification: email and flag (SPAM/NOT SPAM) pairs
  - Stock price prediction: marketing items and historical stock price pairs
- **Unsupervised learning:** Training dataset is unlabeled
  - E.g. clustering: input data is grouped into several “clusters”
- **Semisupervised learning:** Training dataset is partially labeled
- **Reinforcement learning:** A system in which a machine learning model (agent) can interact with the environment, select and perform actions, and get rewards (or penalties). It learns by itself what is the best strategy (policy) to maximize the reward.

Batch and online learning

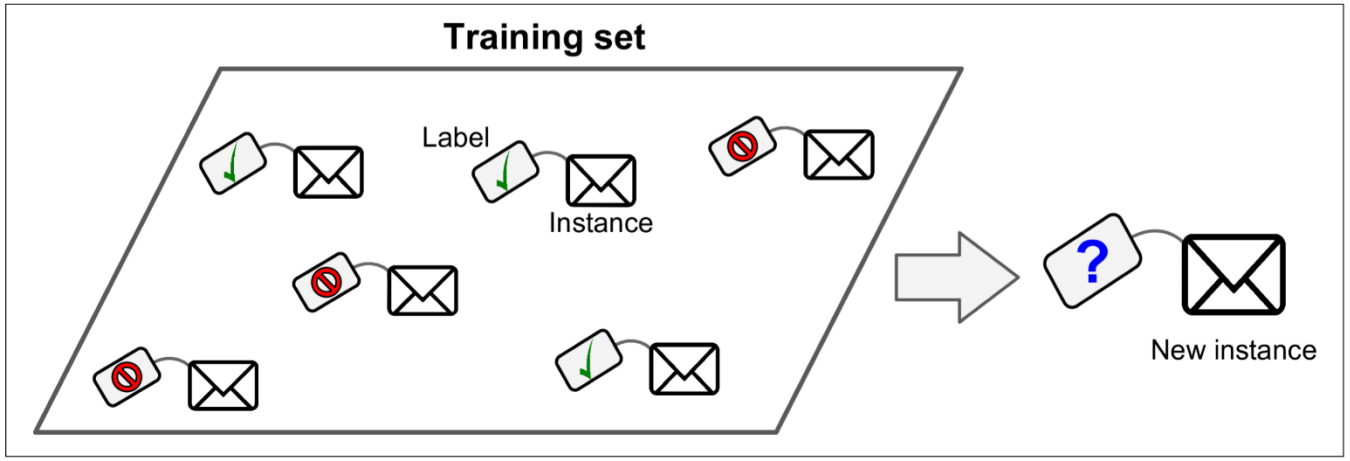
- **Batch learning:** The training process must use all the available data, and is separated from testing
  - High cost on time and computing resources
  - Also called **offline learning**

- **Online learning:** Train the system incrementally by feeding data instances sequentially, either individually or by mini-batches
  - Each learning step is fast and cheap
  - Not occupy additional space after new data are learned
  - Can train models on huge datasets that cannot fit in one main memory, thus also called **out-of-core learning**

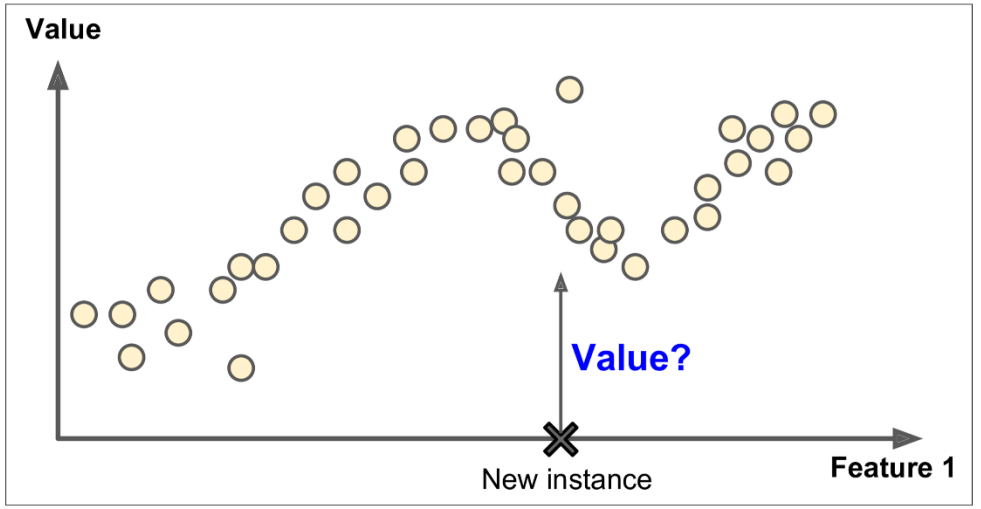
Performance measures

Classification vs. Regression

- **Classification** is a typical supervised learning task, e.g. the spam filter is trained with many example emails along with their class (spam or not spam) and it must learn how to classify new emails
- **Regression** is a task to predict a target numeric value, given a set of features called predictor, e.g. to predict the price of a car given its mileage, age, brand, etc.



Spam classification



Regression

How to measure the performance of a classification model (i.e. classifier)?

Training/test splitting

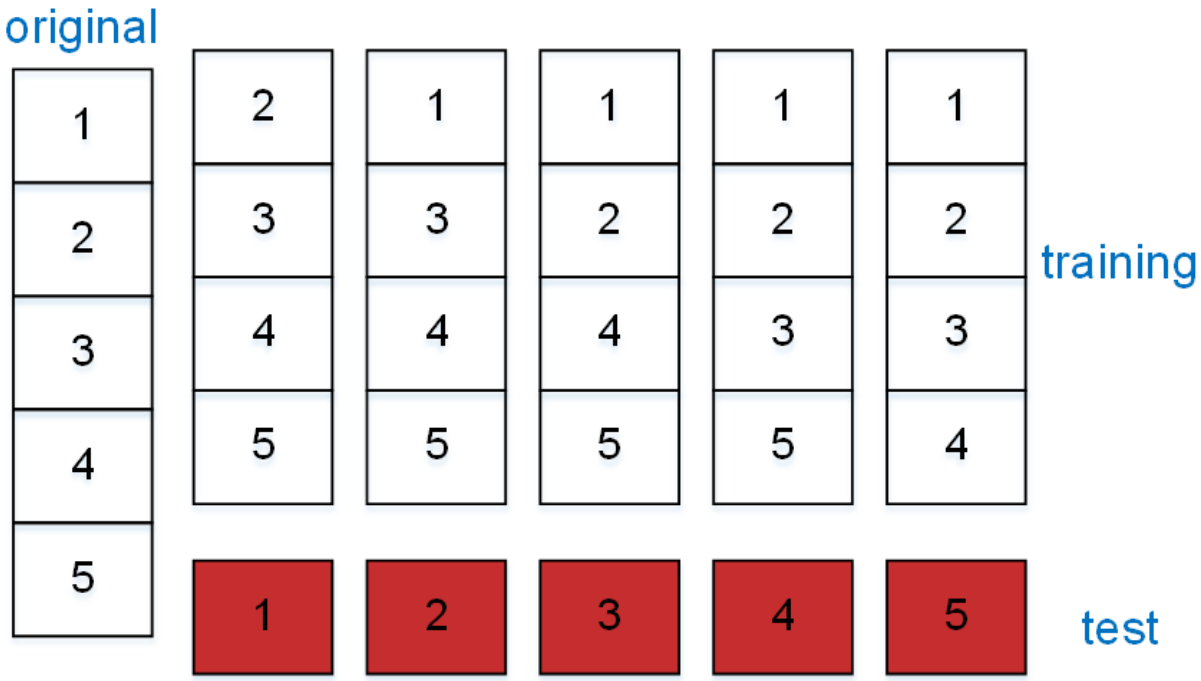
- Training set: only for training prediction functions
- Test set: only for evaluating model performance, and is independent of the training set
- Training/test splitting is usually done **randomly**
- But if the data contain time series, two sets should be created by splitting in time
  - Training set: data **before** time T
  - Test set: data **after** time T

Cross-validation

- An intuitive method to test the model when training:
  - Randomly make *training* / *validation* split on training data
  - Perform the machine learning algorithm using the *training* set
  - Estimate the future result with the *validation* set
- A very simple evaluation method ☺
- We might waste too much training data, and both too large or too small datasets will cause the model evaluation to be imprecise ☹

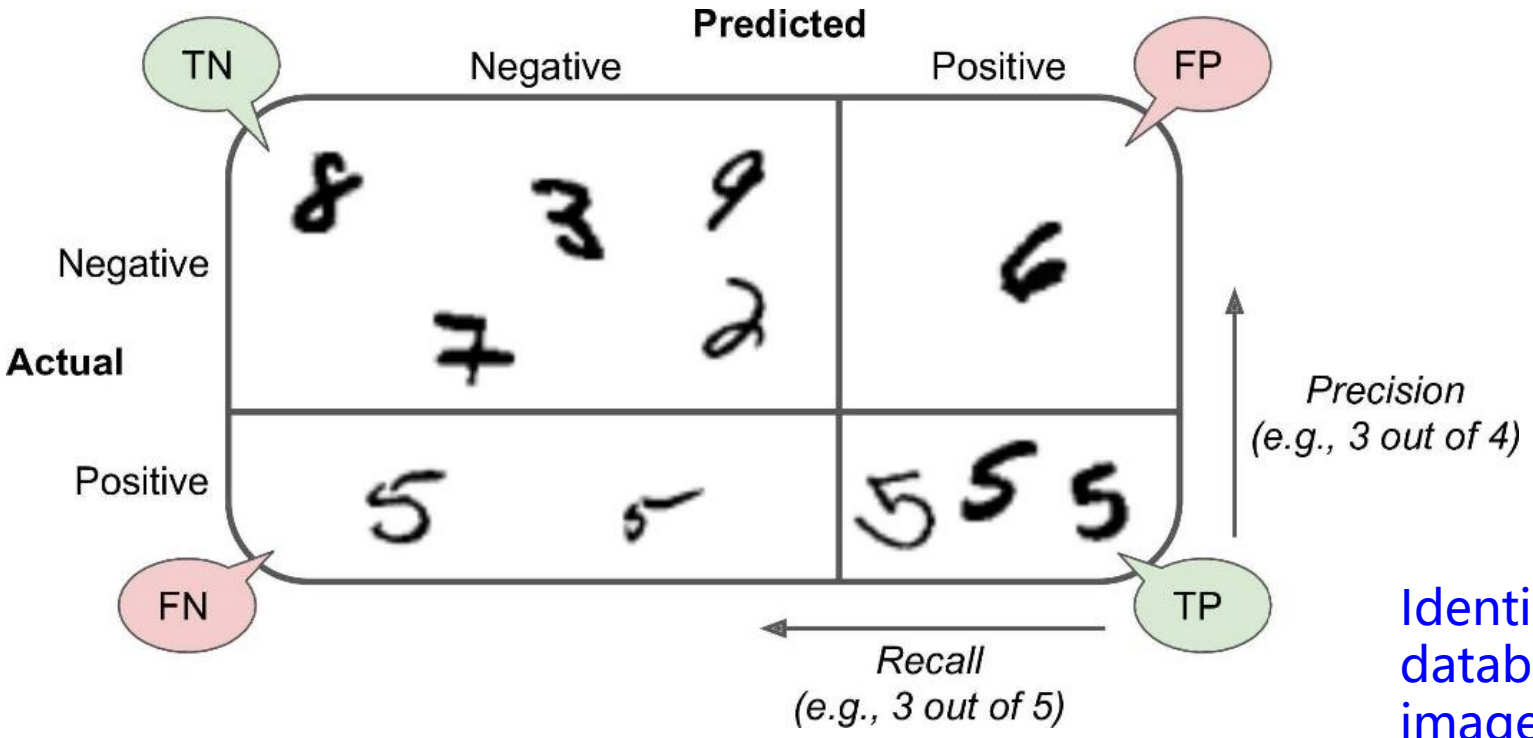
Cross-validation

- Make full use of a dataset with a k-**fold cross-validation**:
  1. uniformly divide data into k blocks
  2. for $j = 1$ to k
  3. train on blocks except k_{th} block
  4. test on k_{th} block
  5. evaluate the result
  6. average the results



Confusion matrix

- To evaluate a classifier, **accuracy** (i.e. ratio of correct predictions) may be biased for skewed datasets (e.g. when some classes are much more frequent than others)
- A better metric for classifier performance is to count the number of instances of class A classified as class B, e.g. for a **binary** classifier:

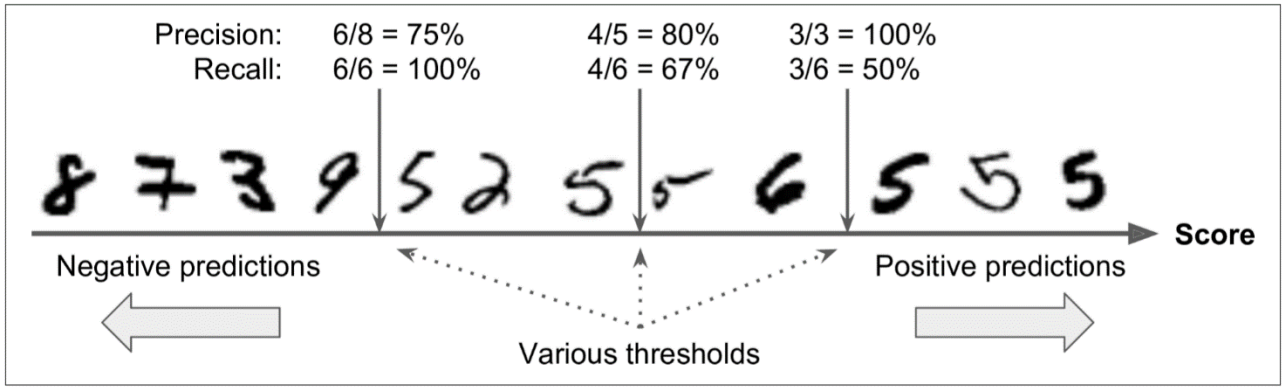


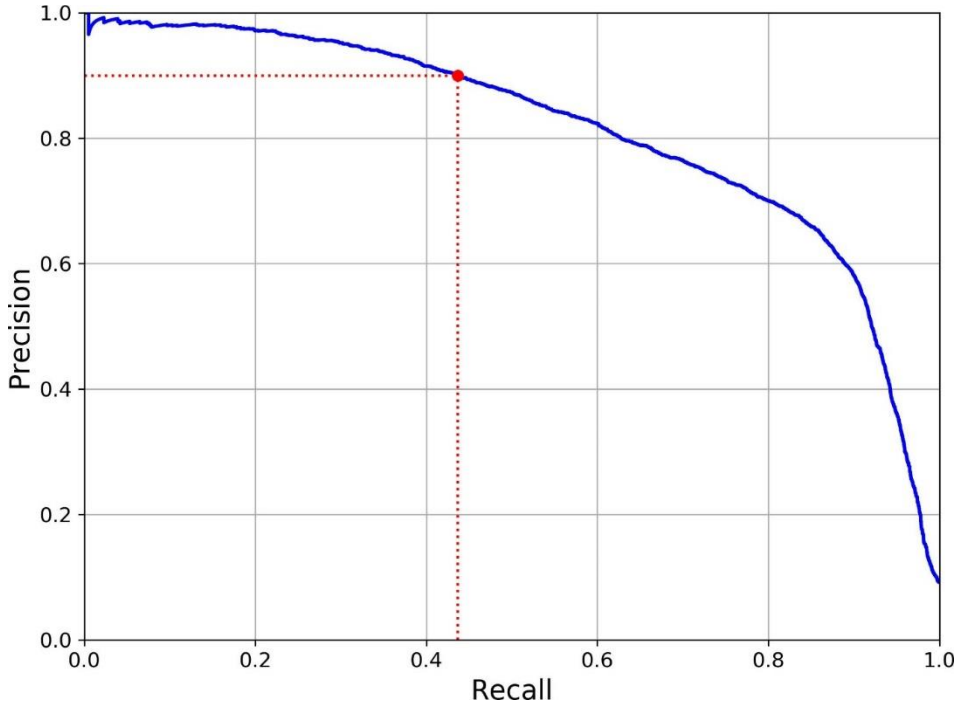
Identify digit 5 from the MNIST database of handwritten digit images (Fig. 3-2 of A. Géron book)

Precision/Recall tradeoff

$$\text{precision} = \frac{TP}{TP + FP} \qquad \text{recall} = \frac{TP}{TP + FN}$$

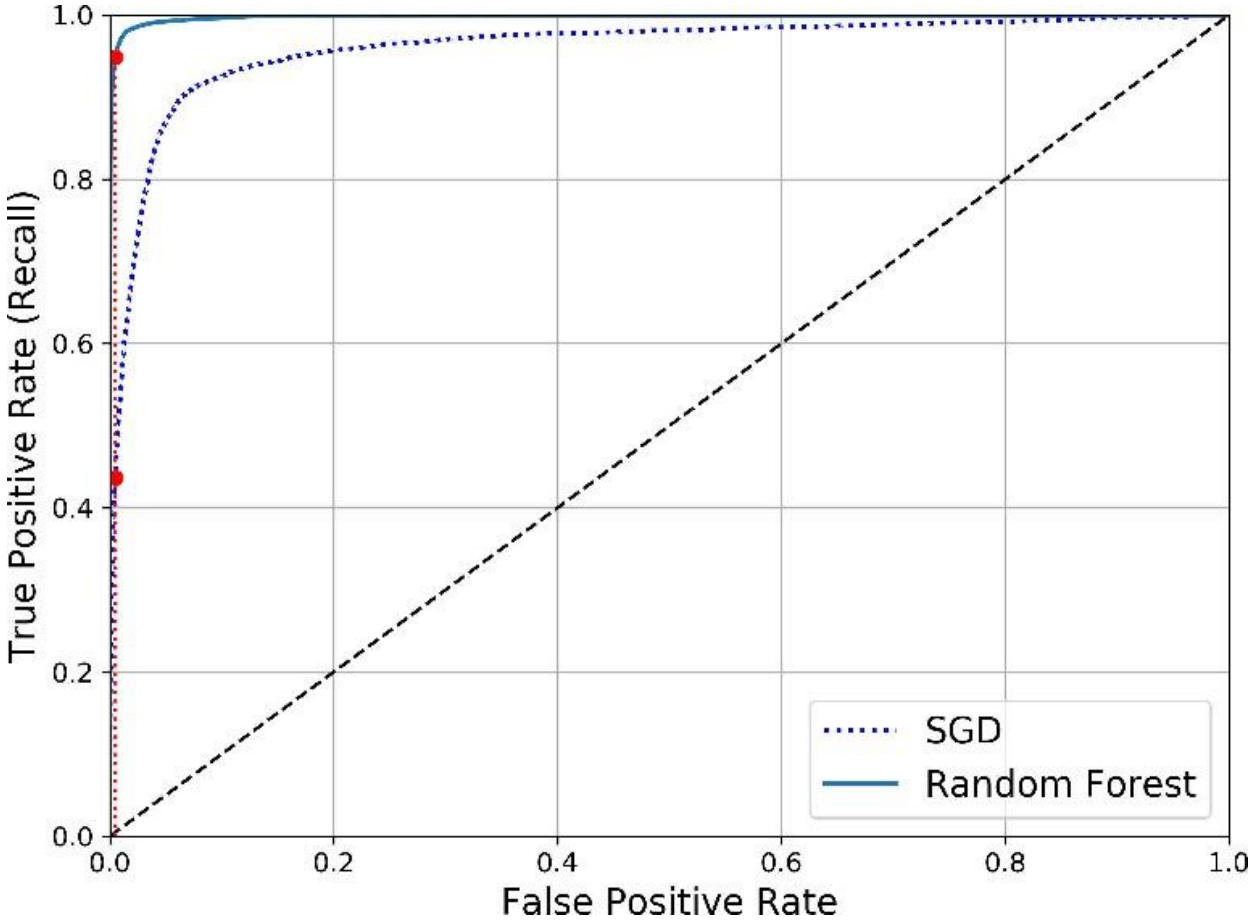
- Adjusting the **decision threshold** (i.e. if score is above such a threshold it is assigned positive, or else to negative) would change relative values on the **precision/recall (PR) curve**





ROC (receiver operating characteristic) curve

- Common used for binary classifiers: TP rate (recall) vs FP rate
- Ideal classifiers have **ROC AUC** (area under the curve) equal to 1



$$TPR = TP / (TP + FN)$$

$$FPR = FP / (FP + TN)$$

Hyperparameters (or “Tuning Parameters”)

- Almost every learning algorithm has
  - At least one “hyperparameter” or “tuning parameter”
- You must tune these values
- Hyperparameters control various things
  - Model complexity (e.g. polynomial order)
  - Type of model complexity control (e.g. L1 vs L2 regularization)
  - Optimization algorithm (e.g. learning rate)
  - Model type (e.g. loss function, kernel type, …)

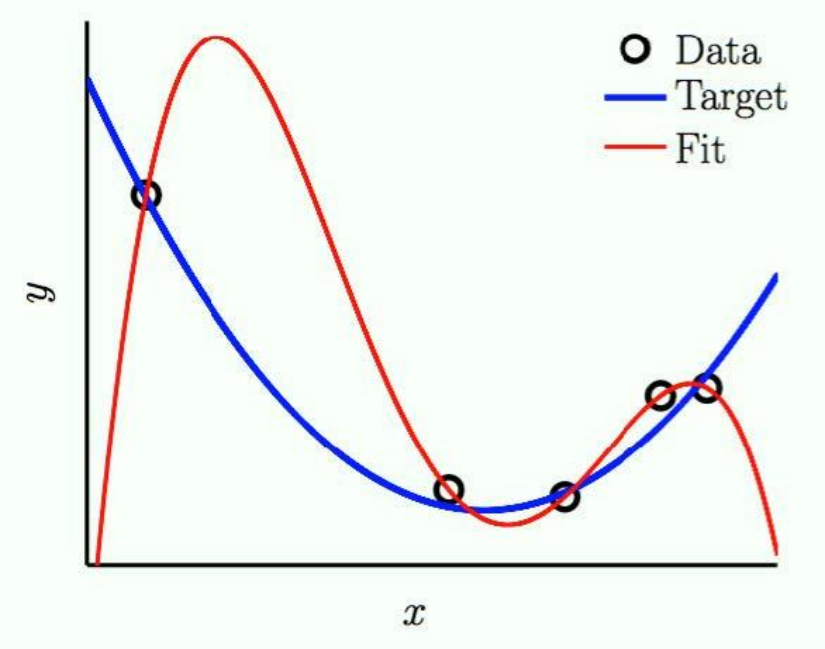
Main challenges

Main Challenges

- **Insufficient quantity of training data**: More data would provide more complete features learned by the model, and thus generate more accurate results
- **Nonrepresentative training data**: Representative training data could help a model generalize to new cases well; otherwise, the model would perform poorly on unseen instances
  - e.g. sampling bias due to “unfair” sampling methods
- **Poor-quality data**: Errors, outliers or noise in the data will interfere with the system to detect underlying patterns, i.e. “garbage in, garbage out”

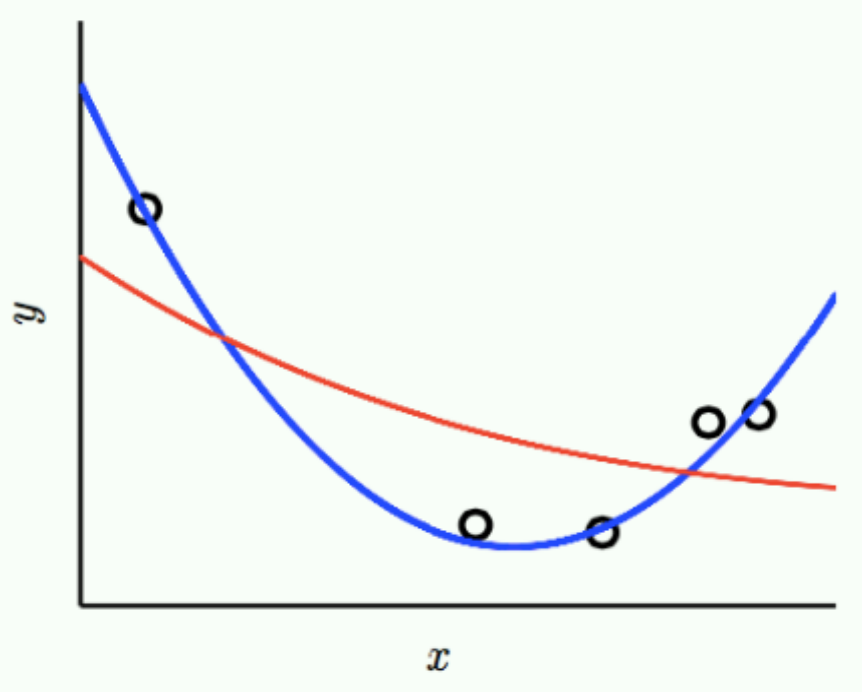
Overfitting

- **Overfitting the training data**:
  - Performance is good on the training data
  - But it does not generalize well
- Potential reasons:
  - **High dimensionality** (too many features)
  - **Model is too complex** (too many parameters)
  - Noise in the training data



Underfitting

- A model **underfits the training data** when it is too simple to learn the underlying structure of the data
- The model
  - cannot fit most of the data, and has poor training performance
  - should be more powerful to catch richer features



Summary

- In this lecture we have learned
  - What machine learning is
  - Why use machine learning: when traditional methods fail
  - Types of machine learning systems, e.g. supervised and unsupervised learning, etc.
  - Methods to measure the performance of classifiers, e.g. cross-validation, confusion matrix, ROC curve, etc.
  - Main challenges of machine learning

- For details, read Chapter 1 and Chapter 3 of Aurélien Géron's book "Hands-On Machine Learning ..." (2019)