Machine Learning in Practice: a Crash Course

Lecture 2: Reframe & Generalization & Metrics

胡津铭
DolphinDB

Recap

- Machine learning is the study of computer systems that improve their performance through experience (mostly, data)

- To use ML, there should be some patterns in the data
  - Sometimes we know these patterns/features, but not know how to use it, for example the parameters. ML learn how to use them
  - Sometimes ML can discover the patterns themselves

- In machine learning, we study two types of problems:
  - Supervised learning
  - Unsupervised learning

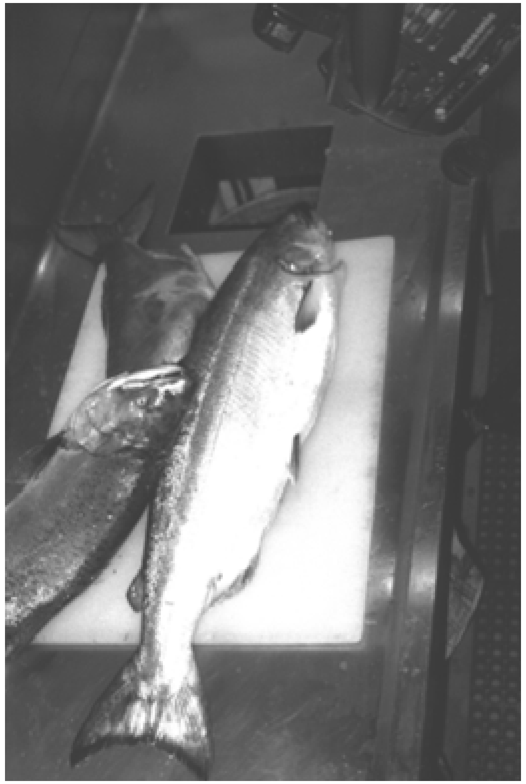
Recap

Define a ML problem → Construct dataset → Transform data & get features → Design & train a model → Use the model to predict

- It is often an iterative process

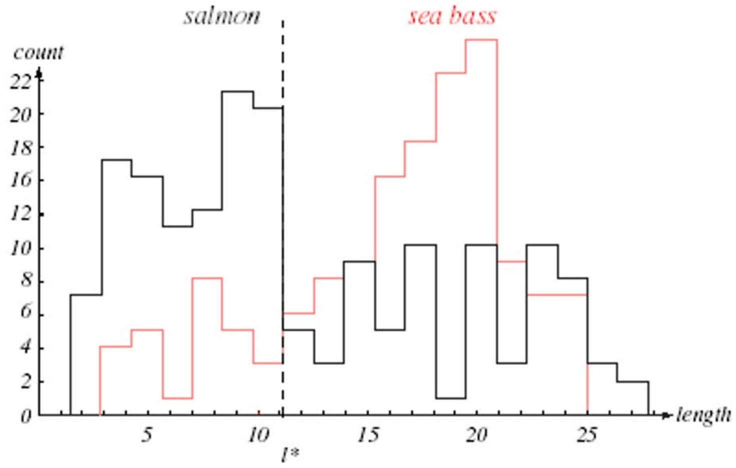
A toy example

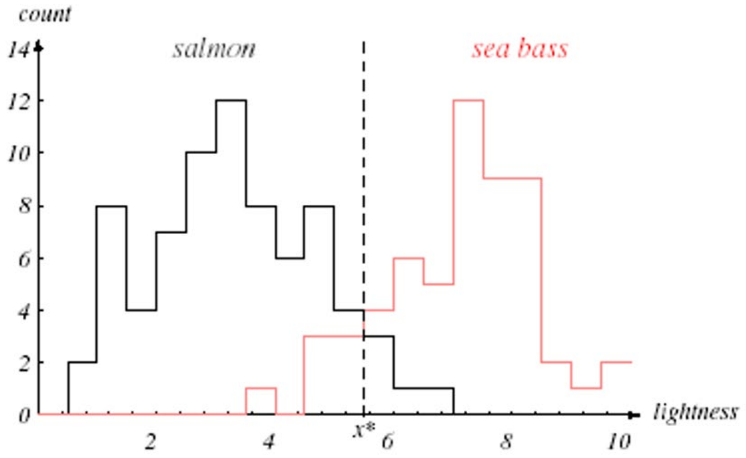
- Fish Classification: Salmon v. Sea Bass
- Feature: length. Model: histogram
- Now the model is not good enough
- We ask some experts in this area, they suggest we’d better use other features. Length of images are not very robust.



Fish Classification: Salmon v. Sea Bass

- 3. Transform data & Get features
  - Previously we use the length of a fish
  - Maybe lightness is a better feature.
  - How to show this?



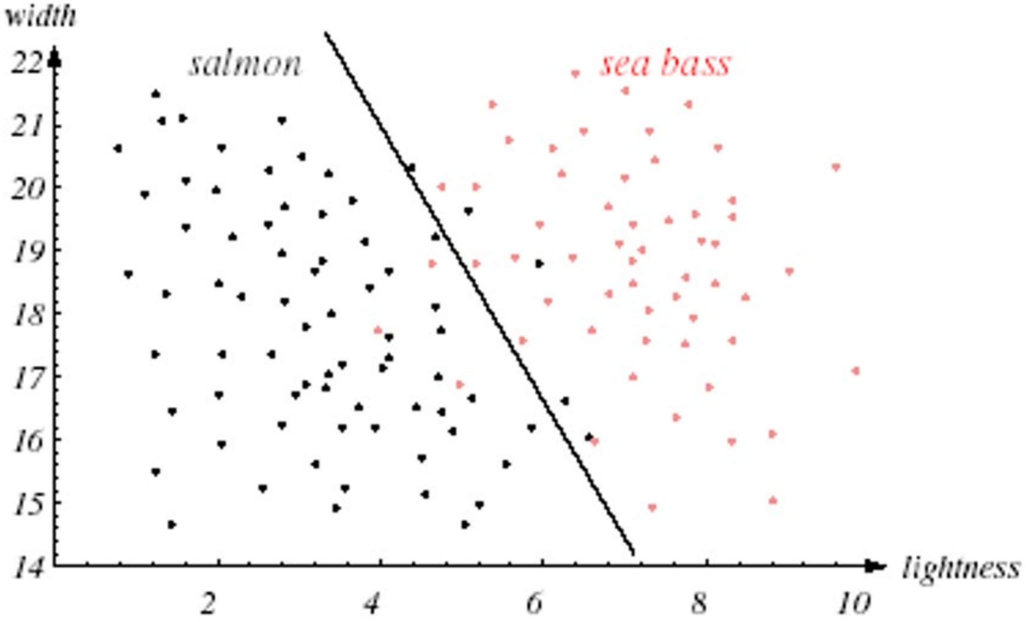


Fish Classification: Salmon v. Sea Bass

- 3. Transform data & Get features
  - Maybe using more features may help?
  - The fishers tell us that width is another useful feature
  - So we add this feature
  - (How to make sure that this is useful?)

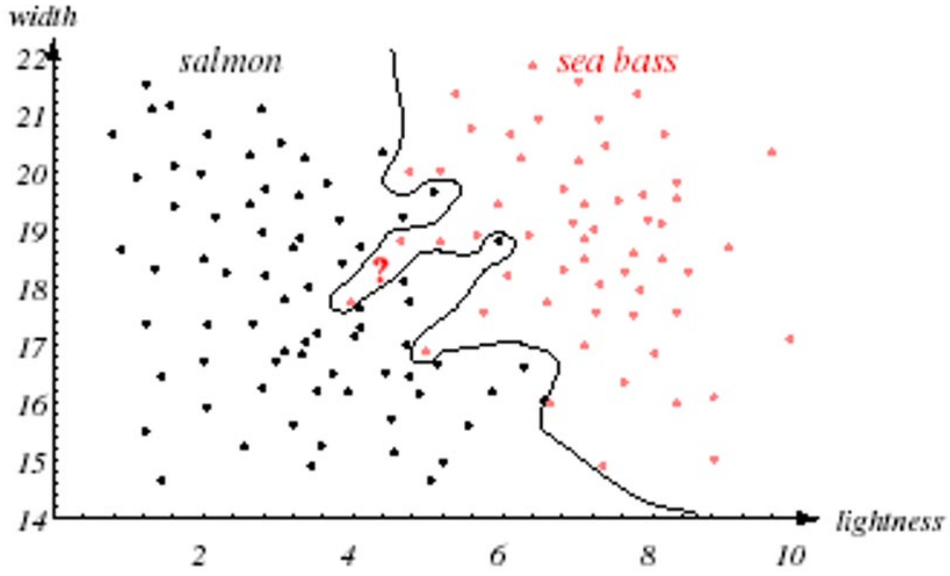
Fish Classification: Salmon v. Sea Bass

- 4. Design & train a model (Training)
  - Now there are more features, we should use other models.
  - Maybe a linear (simple) boundary?



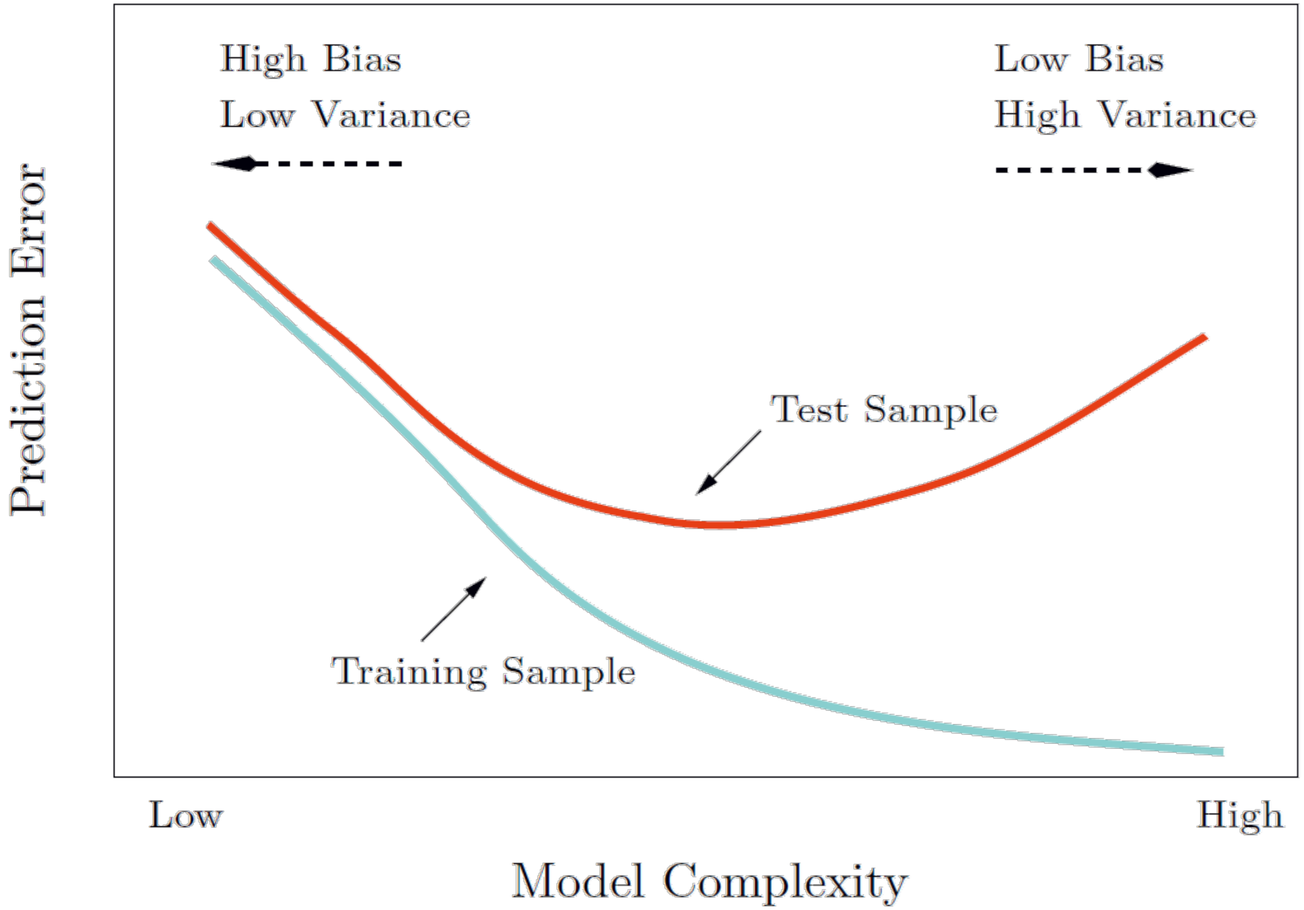
Fish Classification: Salmon v. Sea Bass

- 4. Design & train a model (Training)
  - How about a more complicated model like this?
  - This model seems to give lower (actually 0) training error, so is it a better model? Why?
  - Why do we need the test set?



lightness	width	label
10	20	sea bass
5	15	salmon
15	20	?

Generalization: why we need test set?



Generalization: why we need test set?

- A generalization of a concept is an extension of the concept to less-specific criteria.
- Generalization of the classifier (model)
  - The performance of the classifier on test data.

- Training error:
  - Simple model → large training error
  - Complex model → less training error

- Test error:
  - Simple model → ?
  - Complex model → ?

Generalization: why we need test set?

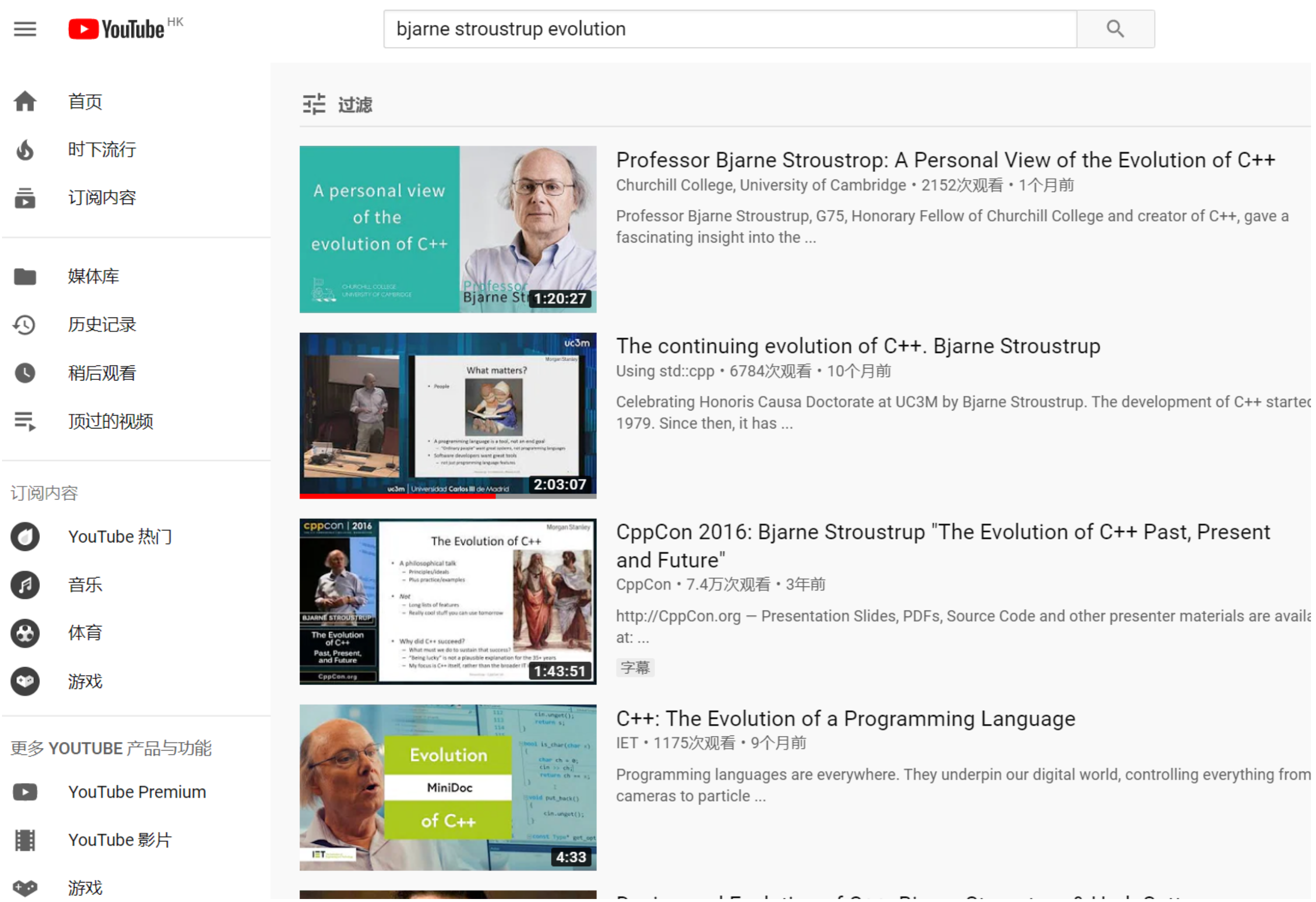
- 5. Use the model to predict
  - Testing
  - Deploying & serving
- What if the training data is significant different from the test data?
- Will the metrics on testing data good or not?
- What if the testing data is significant different from the real-world (serving) data?
- Will the ML system give good results?
- We should try to align the training data and testing data.
- Also, try to align the testing data with the real-world data.

ML Pipeline: a real-world case study

- 1. Define a ML problem
    - Articulate your problem
    - What are the labels and where are they from?
        - Are these labels appropriate?
    - What is the metric?

1. Define a ML problem

- Let's try a real-world problem!
- Assume we are engineers in Youtube. The main income of our company is from ads.
- Youtube will display ads at the beginning, and, say, every 5mins. The more/longer ads the user watches, the more we earn.
- But we also do not want to annoy users.



1. Define a ML problem

- How to convert it to a ML problem? What is the target?
  - Maybe we can use ML to rank the list. How to rank?
  - A straightforward is to predict the click-through rate (CTR). Then rank the list according to CTR. So CTR is target.
    - What is the problem with this target?
  - Maybe the expected watching time? Why is this target better?
    - What is the problem with this target?
  - Maybe a trade-off between CTR and expected watching time.
  - Say we use expected watching time as target.
    - Supervised or Unsupervised?
    - Classification or Regression

1. Define a ML problem

- Are there patterns in the data? Ask yourself!
  - We have special preferences for different videos.
  - People may have different preferences, but similar people may share similar preferences. Also, if we prefer some kinds of videos, maybe we will prefer similar items. So there are patterns.
- Can we get data easily? What is the label? Where are they from?
  - From the user watching history data.
  - The label is directly from the user watch history.
- What are the metrics?
  - Mean Squared Error. Will talk more about metrics later.
  - $\mathrm{MSE}(f, \boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i, \boldsymbol{\theta}))^2$

2. Construct dataset

- Collect it from user watching history.
- Split it into train/test dataset. Will talk more later.

3. Transform data & feature engineering

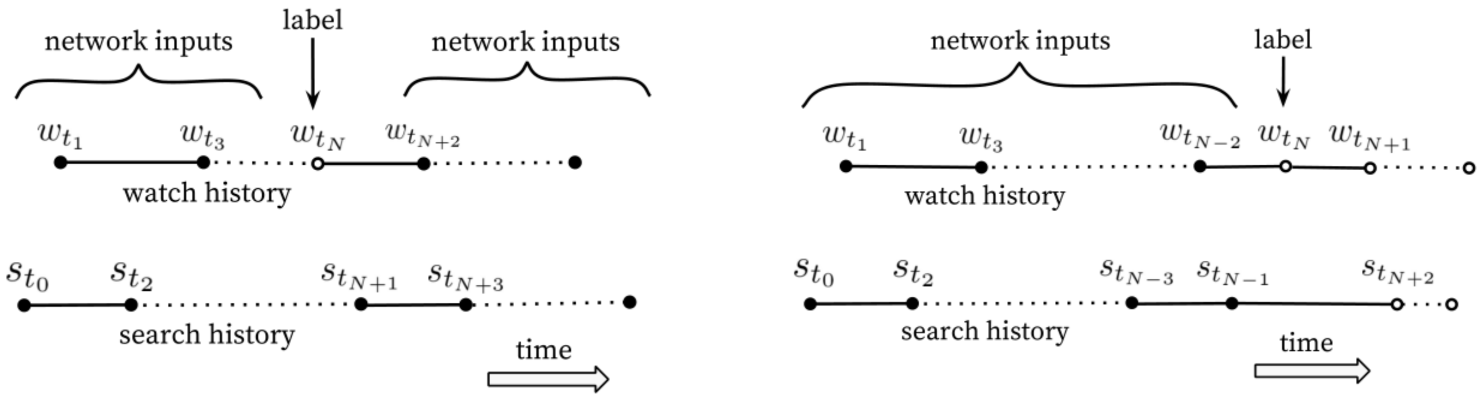
- What are the possible useful features? Ask yourself!
- User features:
  - Personal features: age, gender, job, income ……
  - Geographic: where are the users from?
  - Watching history
- Video features:
  - Uploaders, when is it uploaded,…
  - contents: topics, …, visual contents, length, …
  - who watched this?
- Context:
  - When the users search? What is hot at this time?
  - The search tokens.

4. Design & train models

- We will talk a lot more later in this course. So we skip it currently.

5. Use the model to predict

- Testing
  - How to evaluate the model? What is the testing dataset?
  - Remember, try to make testing align with the real-world serving scenario!
- Deploying & Serving
  - Wont talk about it here.



(a) Predicting held-out watch

(b) Predicting future watch

Question?

Metrics for supervised learning

- Classification
  - $Accuracy = \frac{\sum_i (y_i != f(x_i))}{n}$
  - What are the problems with this metric?
    - Consider cancer detection. Just classify people as not getting cancer can get an accuracy over 99.9%
  - Are all the errors equally important?
    - Consider the bomb detection in railway station. Also, the cancer detection.
    - Consider the spam/ham email detection.
  - $Precision = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN}$
  - Some other metrics for different scenarios…

	Spam(label)	Ham(label)
Spam(predict)	TP	FP
Ham(predict)	FN	TN

Metrics for supervised learning

- Regression
  - $\text{MSE}(f, \boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i, \boldsymbol{\theta}))^2$
  - Why cant we use accuracy?
  - What are the problems with this metric?
    - Outliers will contribute a lot to it
  - Mean Absolute Error $\text{MAE}(f, \boldsymbol{\theta}) = \frac{1}{n}|y_i - f(x_i, \boldsymbol{\theta})|$

Metrics for supervised learning

- Cost
- What is the speed? For example, how many FPS?
    - In some scenarios, speed is extremely important.
- What is the memory consumption?
- What is the platform required for running?
    - CPU vs GPU
    - Server, workstation, laptop, mobile/embedded system
    - The requirement can be very different for different platforms
- What is scale of data required for this model?

Define a ML problem → Construct dataset → Transform data & get features → Design & train a model → Use the model to predict

- It is often an iterative process

Question?

Thanks and welcome to give us suggestions and feedbacks afterwards.