CMPT 733
Practical Machine Learning

SLIDES BY:

JIANNAN WANG

https://www.cs.sfu.ca/~jnwang/

Machine Learning (ML)

People are not only interested in understanding the past but also in predicting the future

Understanding the past (SQL Analytics)	Predicating the future (Machine Learning)
How many Apple Watches did we sell this year?	How many Apple Watches will we sell next year?
Which movies did "Bob" watch before?	Which movies will "Bob" like to watch in the future?
. . .	. . .

ML in Class vs. ML in Practice

ML in Class

- Data is clean, and comes from a single source
- Developing a new ML model is quite important
- A model should have good properties in theory

ML in Practice

- Data is messy, and often comes from multiple sources
- Feature selection and parameter tuning are quite important
- A model should have good performance in production

Outline

Feature Selection

Crowdsourcing and Active Learning

Practical ML in Spark

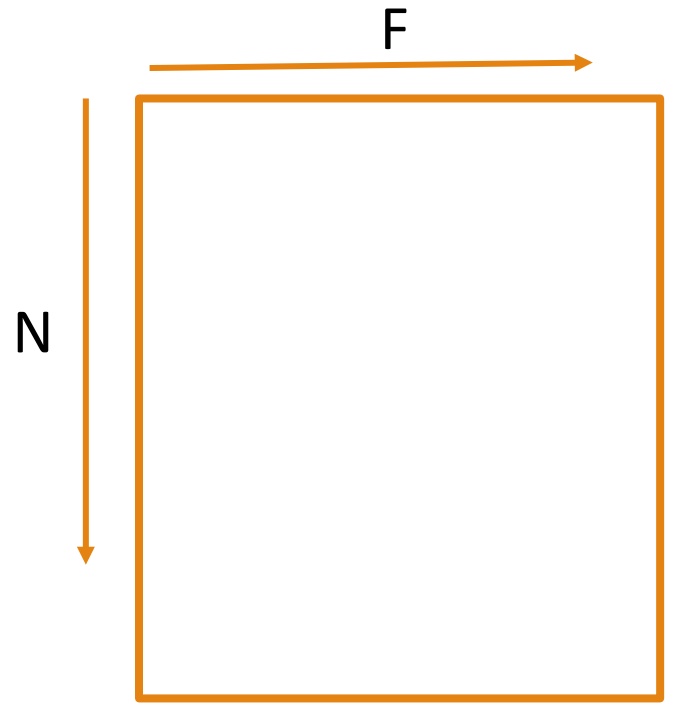
Case Study: Network Intrusion

Feature Selection

What? and Why?

Data are often in the form of a table

- N: # of training examples (e.g., tweets, images)
- F: # of features (e.g., bag of words, color histogram)



Feature Selection

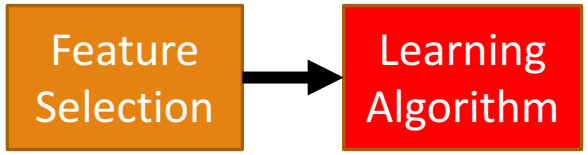
- Selecting a subset of features for use in model construction.

What's bad about "Big F"?

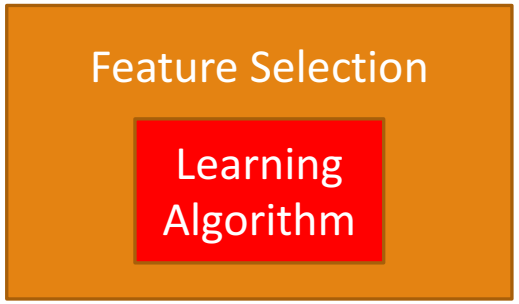
- Slow (training time)
- Inaccurate (due to overfitting and the curse of dimensionality)
- Hard to interpret models

How?

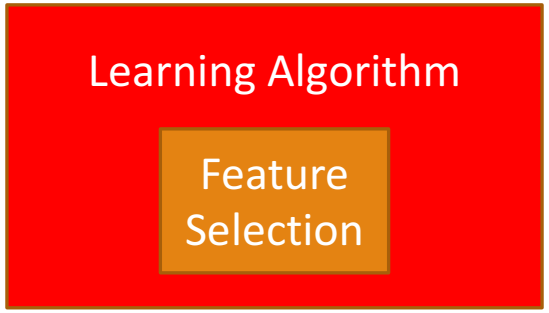
Filter Method



Wrapper Method



Embedded Method



Filter Method

Basic Idea

- Assign a score to each feature
- Filter out useless features based on the scores

Many popular scores [see Yang and Pederson '97]

- Classification: Chi-squared, information gain, document frequency
- Regression: correlation, mutual information

Wrapper Method

Basic Idea

- Evaluate subsets of features
- Select the best subset

How to evaluate a subset of features?

- Test Error (estimated by cross validation)

How to find the best subset?

- Greedy Algorithms (e.g., forward selection, backward elimination)

Embedded Method

Basic Idea

- Modify a learning algorithm such that it can automatically penalize useless features

Lasso Regression

$$\underset{\beta}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

Penalize useless features

Comparisons

Filter Method

- 😀 Efficient
- 😀 Robust to overfitting
- ☹️ Fails to capture relationships between features

Wrapper Method

- 😀 Capture relationships between features
- ☹️ Inefficient
- ☹️ May suffer from overfitting

Embedded Method

- 😀 Combine the advantages of the above methods
- ☹️ Specific to a learning algorithm

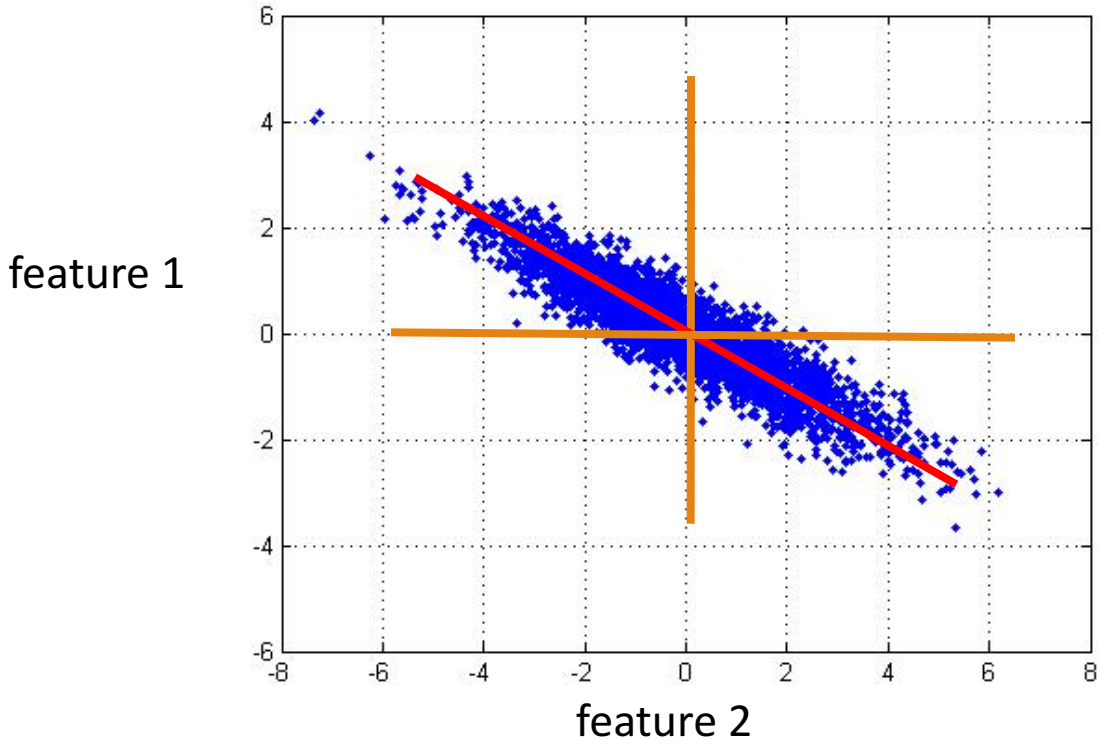
Dimensionality Reduction

Feature Selection

- New features have to be a subset of old features

Feature Transformation (e.g., PCA)

- New features may NOT be a subset of old features



Feature Selection Summary

Why feature selection?

Feature-selection methods

- Filter method
- Wrapper method
- Embedded method

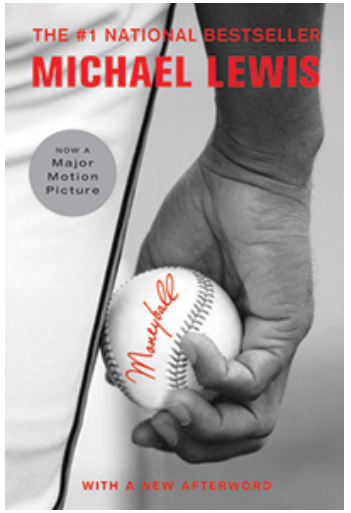
Feature Selection vs. PCA

Crowdsourcing

CMPT 884: Human-in-the-loop Data Management (SFU, Fall 2016)
https://sfu-db.github.io/cmpt884-fall16/

Data Science Job

Extract insights from data



Key Resources

Algorithms

- Machine Learning, Statistical Methods
- Prediction, Business Intelligence

Machines

- Clusters and Clouds
- Warehouse Scale Computing

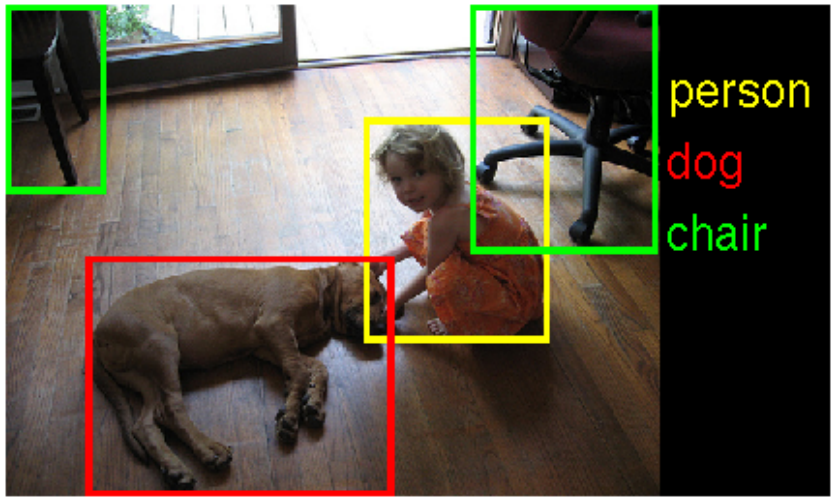
People

- Crowdsourcing, Human Computation
- Data Scientists, Analysts

From Prof. Michael Franklin's talk

An Example of Using Three Resources

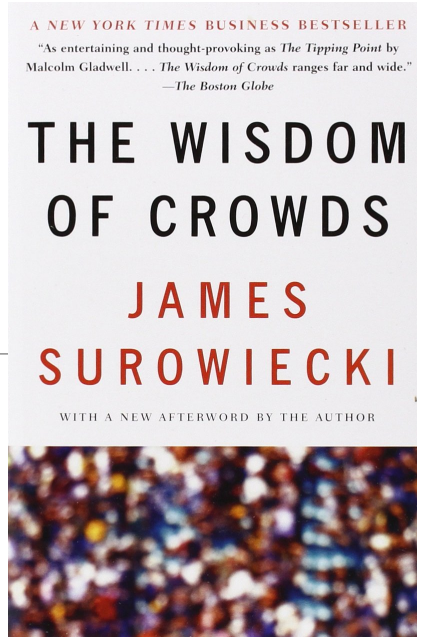
What are in the image?



How to solve the problem?

Deep Learning (Algorithms)
GPU Cluster (Machines)
ImageNet (People)

The Wisdom of Crowds

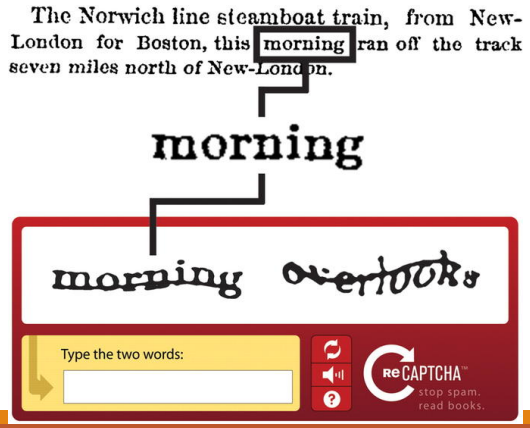


What does it mean?

- Two heads are better than one

Some famous examples

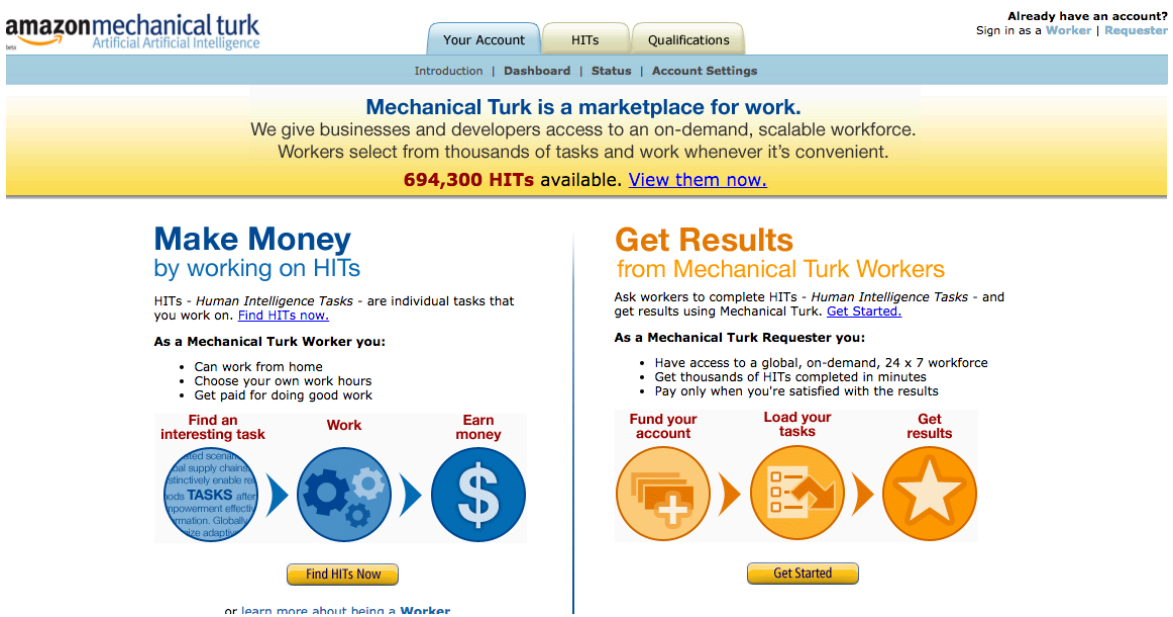
WIKIPEDIA

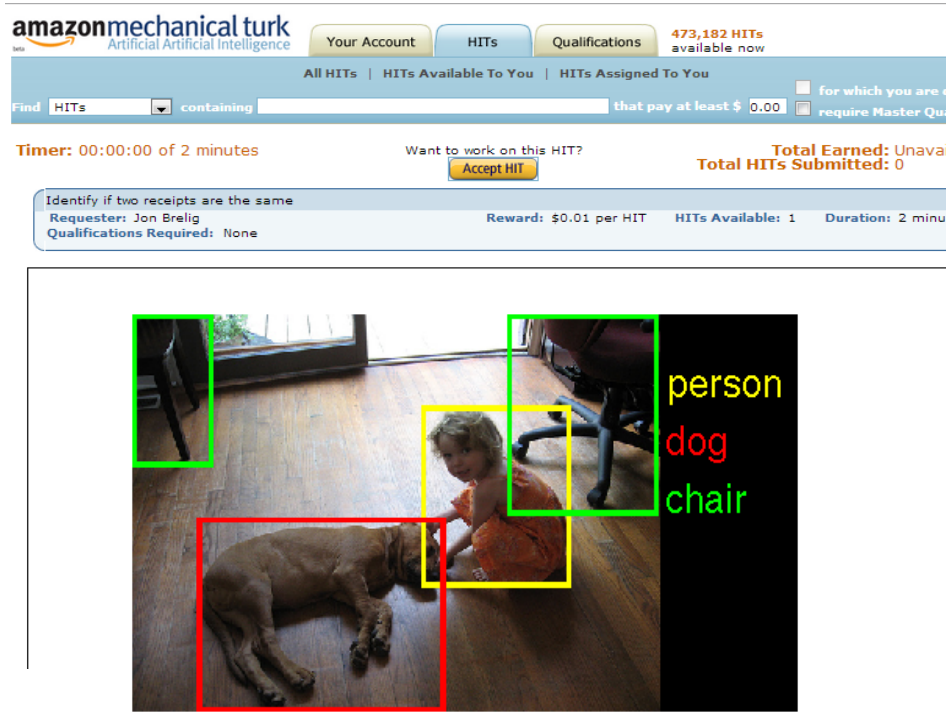


Crowdsourcing Platforms

Amazon Mechanical Turk

500K+ workers*





* https://requester.mturk.com/tour

Industrial Survey

Company	Team	Persona
Amazon	Product classification	Largely single-case user
Captricity	Focus of large part of company	Largely single-case user
Dropbox	Single person consulting several teams	Multi-case user / Internal provider
Facebook	Entities team	Multi-case user
Flipora	Startup CTO	Multi-case user
GoDaddy	Small business data extraction	Multi-case user
Groupon	Merchant data team	Multi-case user
Google	Internal crowdsourcing team	Internal provider
Google	Web knowledge discovery team	Multi-case user
LinkedIn	Single person consulting several teams	Multi-case user / Internal provider
Microsoft	Internal crowdsouricng team	Internal provider
Microsoft	Search relevance team	Multi-case user
Youtube	Crowdsourcing team	Largely single-case user

Use Cases

Use Cases	# Participants
Classification	12
Entity resolution	6
Data cleaning	5
Ranking	5
Spam detection	5
Data extraction	5
Text generation	5

Research Challenges

Trade-off

- **Cost.** How much will it cost?
- **Latency.** How long will it take?
- **Quality.** How accurate will it be?

Cost Control

- Task Selection, Answer Deduction, Pruning

Latency Control

- Task Pricing, Straggler Mitigation, Pool Maintenance

Quality Control

- Worker Elimination, Answer Aggregation, Task Assignment

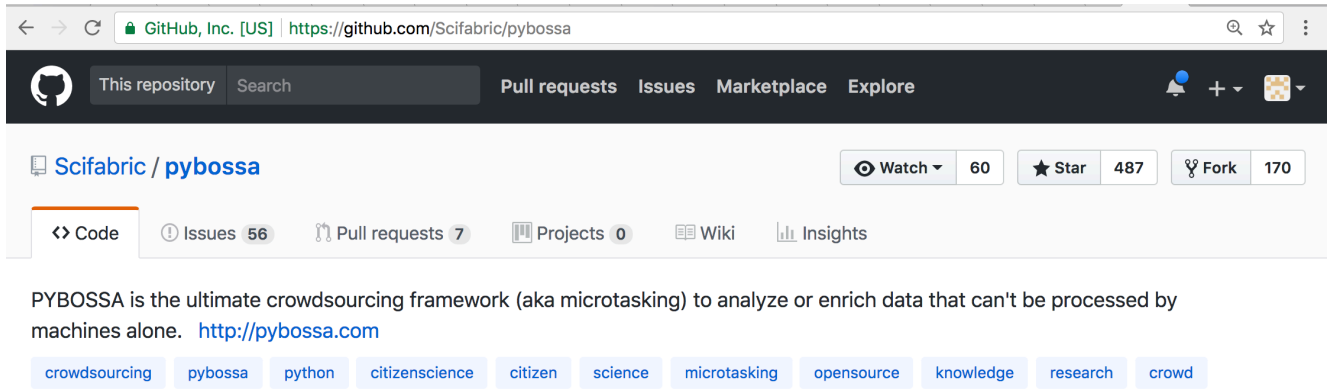
Guoliang Li, Jiannan Wang, Yudian Zheng, Michael Franklin. "Crowdsourced data management: A survey." TKDE 2016

Crowdsourcing may not work ☹

What if your data is confidential?

- E.g., Medical Data, Customer Data

Internal Crowdsourcing Platform



Crowdsourcing may not work ☹

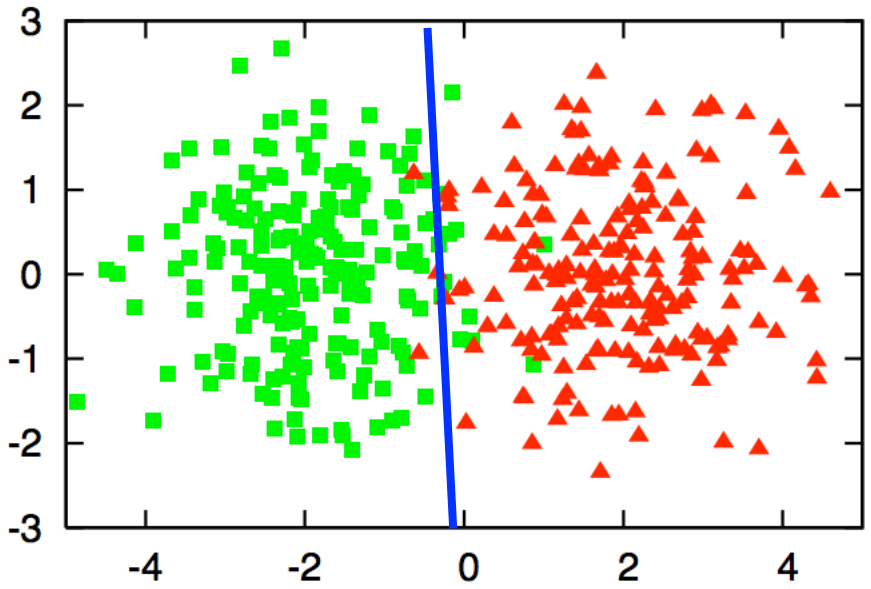
What if your data is so big?

- E.g., Label 10 million images

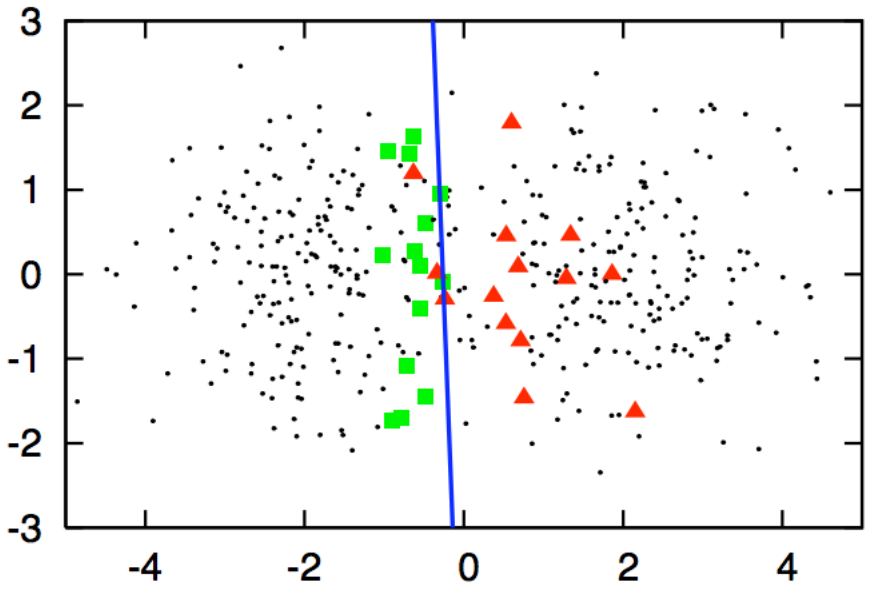
Active Learning

Active Learning

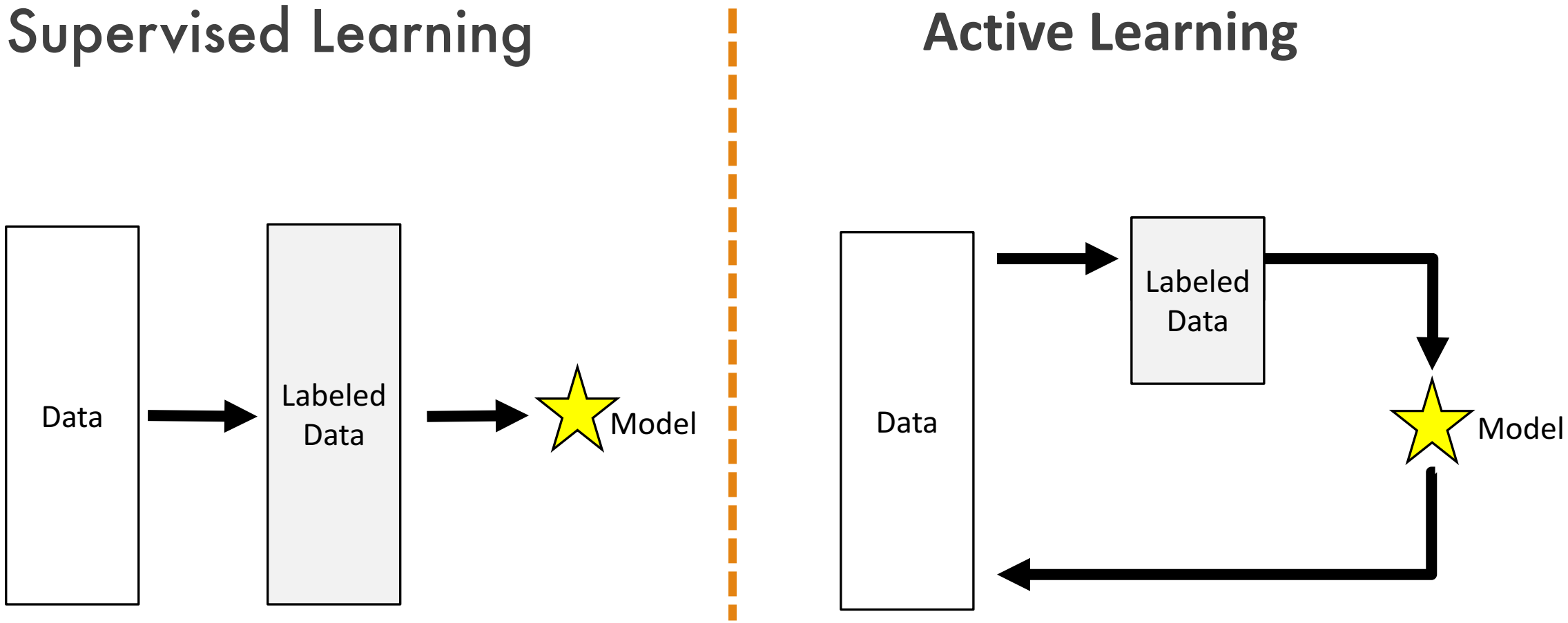
Supervised Learning



Active Learning



Workflow


Supervised Learning
Data
Labeled Data
Model
Active Learning
Data
Labeled Data
Model

Query Strategy

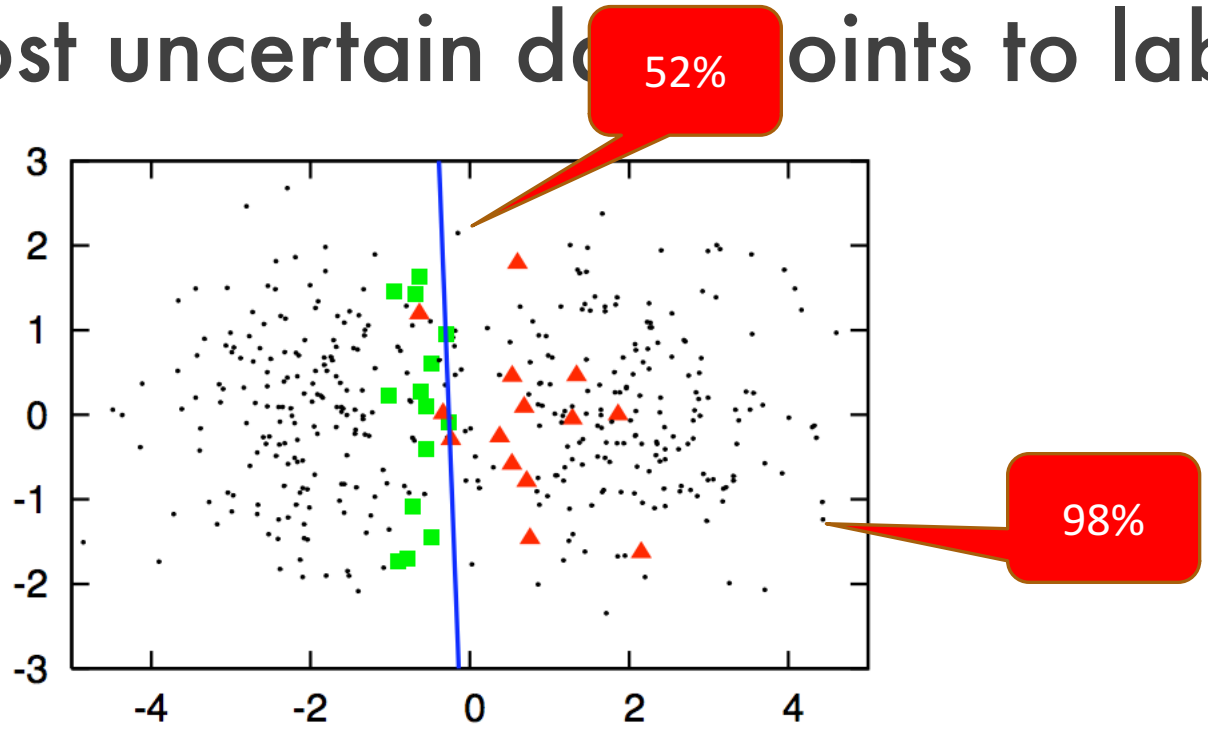
Which data points should be labeled?

- Uncertain Sampling
- Query-By-Committee
- Expected Error Reduction
- Expected Model Change
- Variance Reduction
- Density-Weighted Methods

Settles, Burr. "Active learning literature survey." University of Wisconsin, Madison 52.55-66 (2010): 11.

Uncertain Sampling

Pick up most uncertain data points to label



Logistic Regression

- predict_proba(X)

Summary

Crowdsourcing

- Why crowdsourcing?
- How does it work?

Active Learning

- Why active learning?
- How does it work?

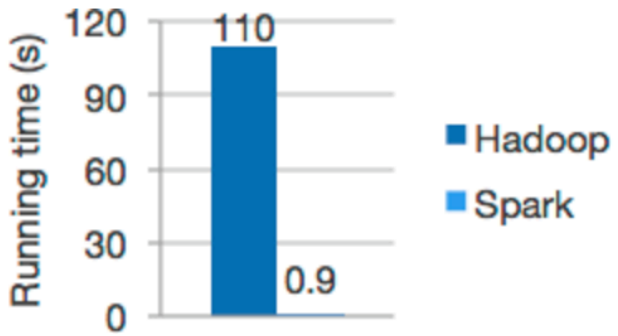
Practical ML in Spark

Recap of Spark

Spark is a fast and general engine for large-scale data processing

Improving MapReduce from two aspects

- Efficiency: in-memory computing
- Usability : high-level operators



Logistic regression in Hadoop and Spark

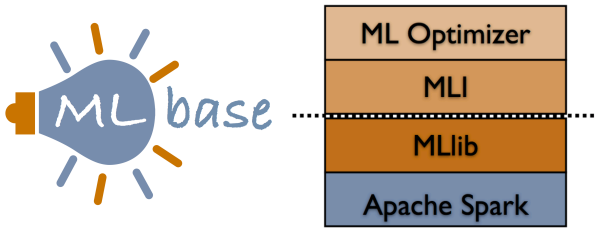
```
text_file = spark.textFile("hdfs://...")

text_file.flatMap(lambda line: line.split())
    .map(lambda word: (word, 1))
    .reduceByKey(lambda a, b: a+b)
```

Word count in Spark's Python API

A Brief History of MLlib

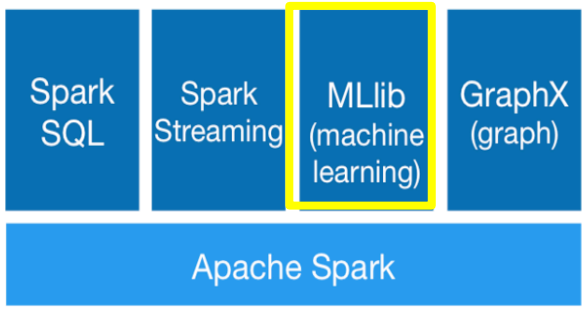
Mlbase (2012)

- Started in the AMPLab
- Goal: making distributed machine learning easy



MLlib enters Spark v0.8 (2013)

- One of the four big libraries built on top of Spark
- A good coverage of distributed machine learning algorithms



A New High-Level API for MLlib (2015)

- spark.ml provides higher-level API built on top of DataFrames for constructing ML pipelines

MLlib's Mission

"Making **practical machine learning** **scalable** and **easy**"

How did MLlib achieve the goal of scalability?

- Implementing distributed ML algorithms using Spark

Some Basic Concepts of Distributed ML

What is distributed ML?

Distributed ML vs. Non-distributed ML

Performance metrics

What is Distributed ML?

A Hot topic! Many Aliases:

- Scalable Machine Learning
- Large-scale Machine Learning
- Machine Learning for Big Data

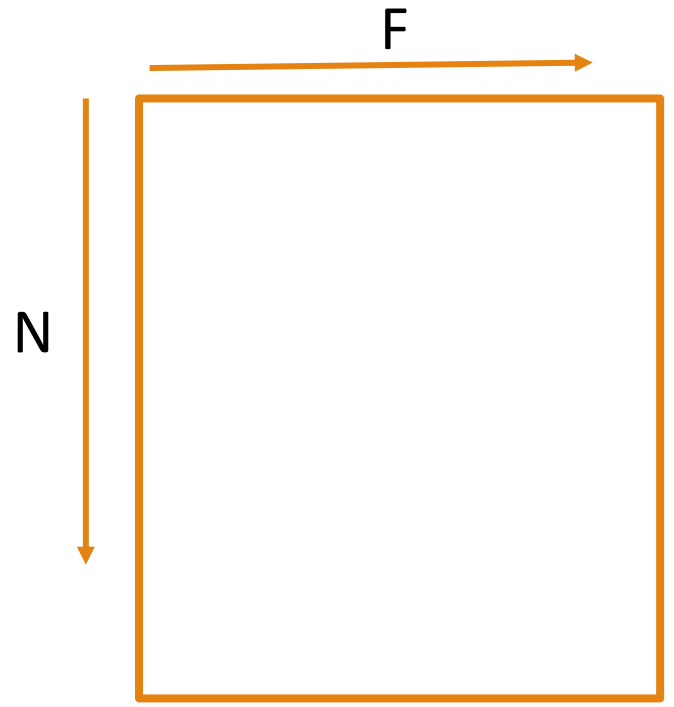
Take a look at these courses

- SML: Scalable Machine Learning (UC Berkeley, 2012)
- Large-scale Machine Learning (NYU, 2013)
- Scalable Machine Learning (edX, 2015)

An intuitive explanation

Data are often in the form of a table

- N: # of training examples (e.g., tweets, images)
- F: # of features (e.g., bag of words, color histogram)



An ML algorithm can be thought of as a process that turns the table into a model

Distributed ML studies how to make the process work for the following cases:

- Big N, Small F
- Small N, Big F
- Big N, Big F

Distributed ML vs. Non-distributed ML

Require distributed data storage and access

- Thanks to HDFS and Spark!

Network communication is often the bottleneck

- Non-distributed ML focuses on reducing CPU time and I/O cost, but distributed ML often seeks to reduce network communication

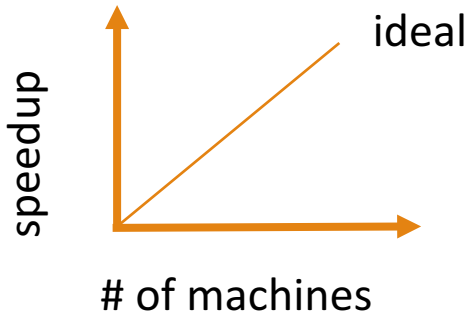
More design choices

- Broadcast (Recall Assignment3B in CMPT 732)
- Caching (which intermediate results should be cached in Memory?)
- Parallelization (which part in an ML algorithm should be parallelized?

Performance metrics of Distributed ML

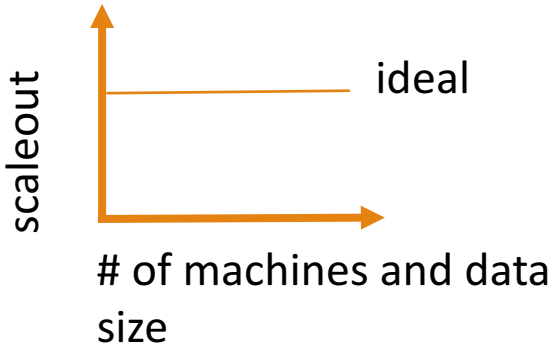
Speedup

- More machines → Higher speed



Scaleout

- More machines → Can process more data



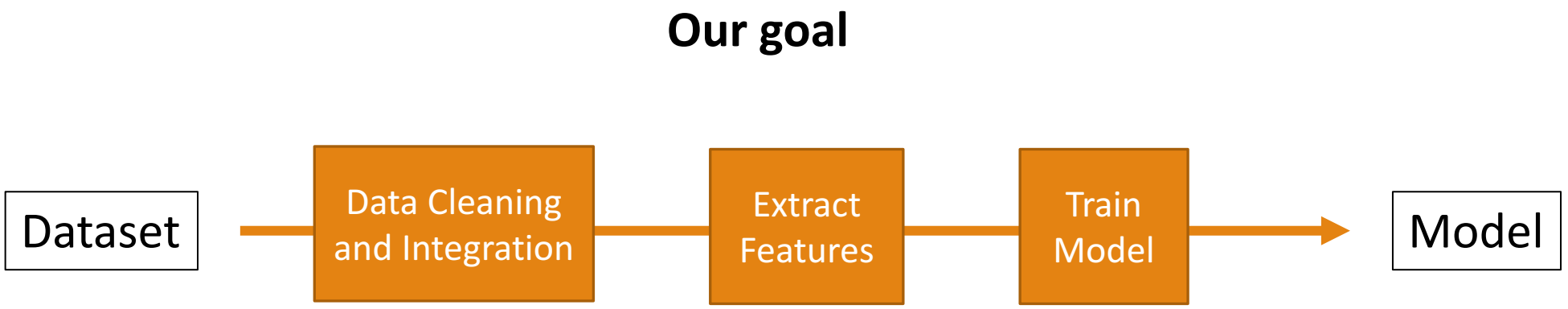
MLlib's Mission

"Making **practical machine learning** **scalable** and **easy**"

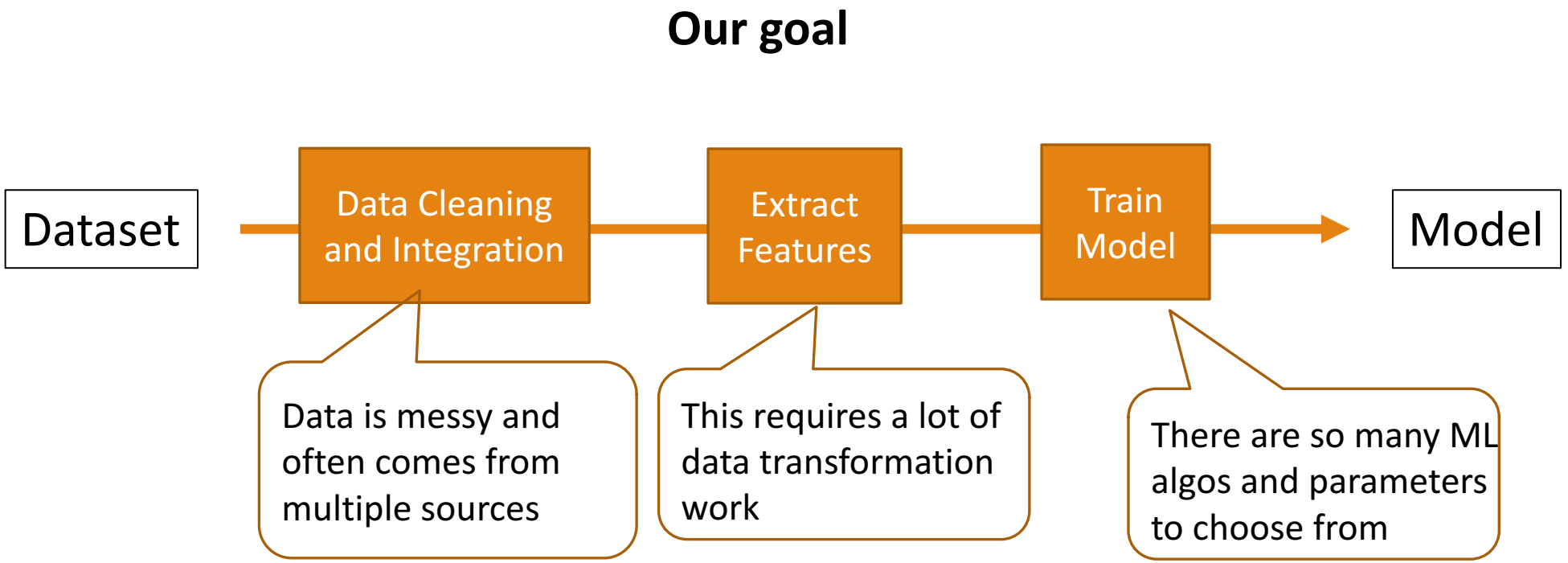
How did MLlib achieve the goal of ease of use?

- The new ML Pipeline API

ML Workflow


Our goal
Dataset
Data Cleaning and Integration
Extract Features
Train Model
Model

ML Workflows are complex



Pain points

Manipulating and managing RDDs are painful

Writing the workflow as a script is hard for workflow reuse

Tuning parameters are painful

The new ML pipeline API

Pain Point 1

- Manipulating and managing RDDs are painful

Basic Idea: Using DataFrame (instead of RDD)

- DataFrame adds schema and relational operations on RDD

easy to manage

```
rdd.filter(lambda x: x[1] > 30)
```

vs.

```
df.filter("age > 30")
```

easy to manipulate

```
rdd.map(lambda (x, y): (x, (y, 1))) \
    .reduceByKey(lambda x, y: (x[0]+y[0], x[1]+y[1])) \
    .map(lambda (x, y): (x, y[0]*1.0/y[1])).collect()
```

vs.

```
df.groupby("name").avg("score").collect()
```

The new ML pipeline API (cont'd)

Pain Point 2: Writing as a script is hard for workflow reuse.

Basic Idea: Abstracting ML stages into two components

- Transformer: DataFrame → DataFrame
- Estimator: DataFrame → Model

A pipeline consists of a sequence of Transformers and Estimators

- Create a new pipeline: `pipeline = Pipeline(stages=[Transformer1, Transformer2, Estimator])`
- Apply the pipeline to a dataset: `model = pipeline.fit(dataset)`
- Use a different estimator: `pipeline' = Pipeline(stages=[Transformer1', Transformer2, Estimator])`
- Apply to a new dataset: `model' = pipeline'.fit(new-dataset)`

The new ML pipeline API (cont'd)

Pain Point 3: Parameter tuning is painful.

Basic Idea: grid search and cross validation

- Grid search enumerates every possible combination of parameters

numFeatures = 10, 100, 1000
regParam = 0.1, 0.01

Grid Search →

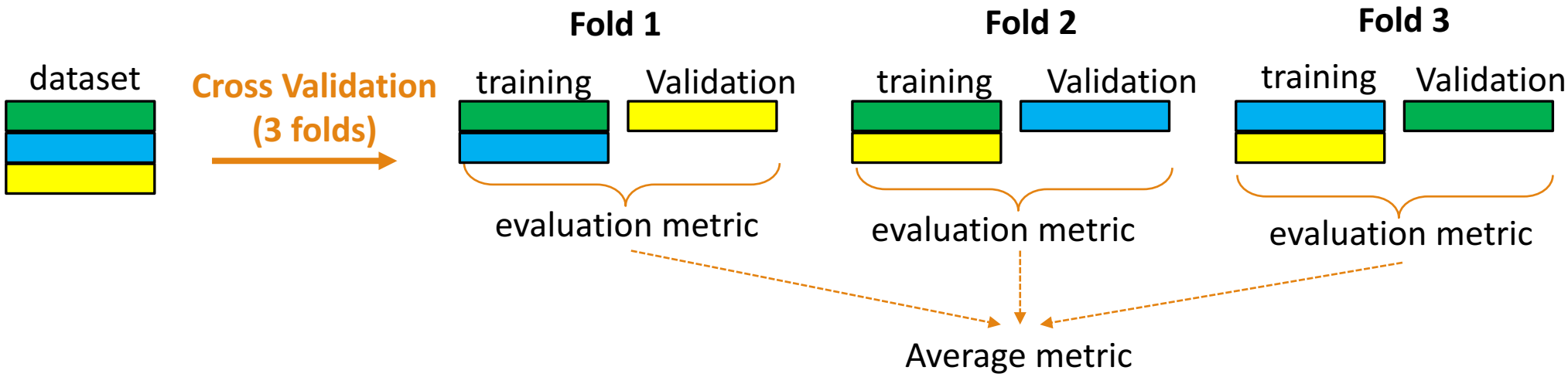
(10, 0.1), (10, 0.01), (100, 0.1), (100, 0.01), (1000, 0.1), (1000, 0.01)

The new ML pipeline API (cont'd)

Pain Point 3: Parameter tuning is painful.

Basic Idea: grid search and cross validation

- Grid search enumerates every possible combination of parameters
- Cross validation evaluates which combination performs the best



spark.mllib vs. spark.ml

Advantage of spark.ml

- spark.mllib is the **old** ML API in Spark. It only focuses on making ML scalable.
- spark.ml is the **new** ML API in Spark. It focuses on making ML both scalable and **ease of use**.

Disadvantage of spark.ml

- spark.ml contains fewer ML algorithms than spark.mllib.

Summary

What is distributed ML?

Distributed ML vs. Non-distributed ML

Performance metrics

Spark ML Pipeline API