1. Introduction to Artificial Intelligence
2. The Fundamentals of Machine Learning

Chapter 0

Introduction to Artificial Intelligence

- What is artificial intelligence
- AI types
- Difference between AI, ML, and DL
- Data Roles
- AI applications
- Companies

What is AI?

Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence.

3 Types of Artificial Intelligence

Artificial Narrow Intelligence (ANI)

Stage-1

Machine Learning

- Specialises in one area and solves one problem

Siri Alexa Cortana

Artificial General Intelligence (AGI)

Stage-2

Machine Intelligence

- Refers to a computer that is as smart as a human across the board

Artificial Super Intelligence (ASI)

Stage-3

Machine Consciousness

- An intellect that is much smarter than the best human brains in practically every field

ARTIFICIAL INTELLIGENCE

IS NOT NEW

ARTIFICIAL INTELLIGENCE

Any technique which enables computers to mimic human behavior

MACHINE LEARNING

AI techniques that give computers the ability to learn without being explicitly programmed to do so

DEEP LEARNING

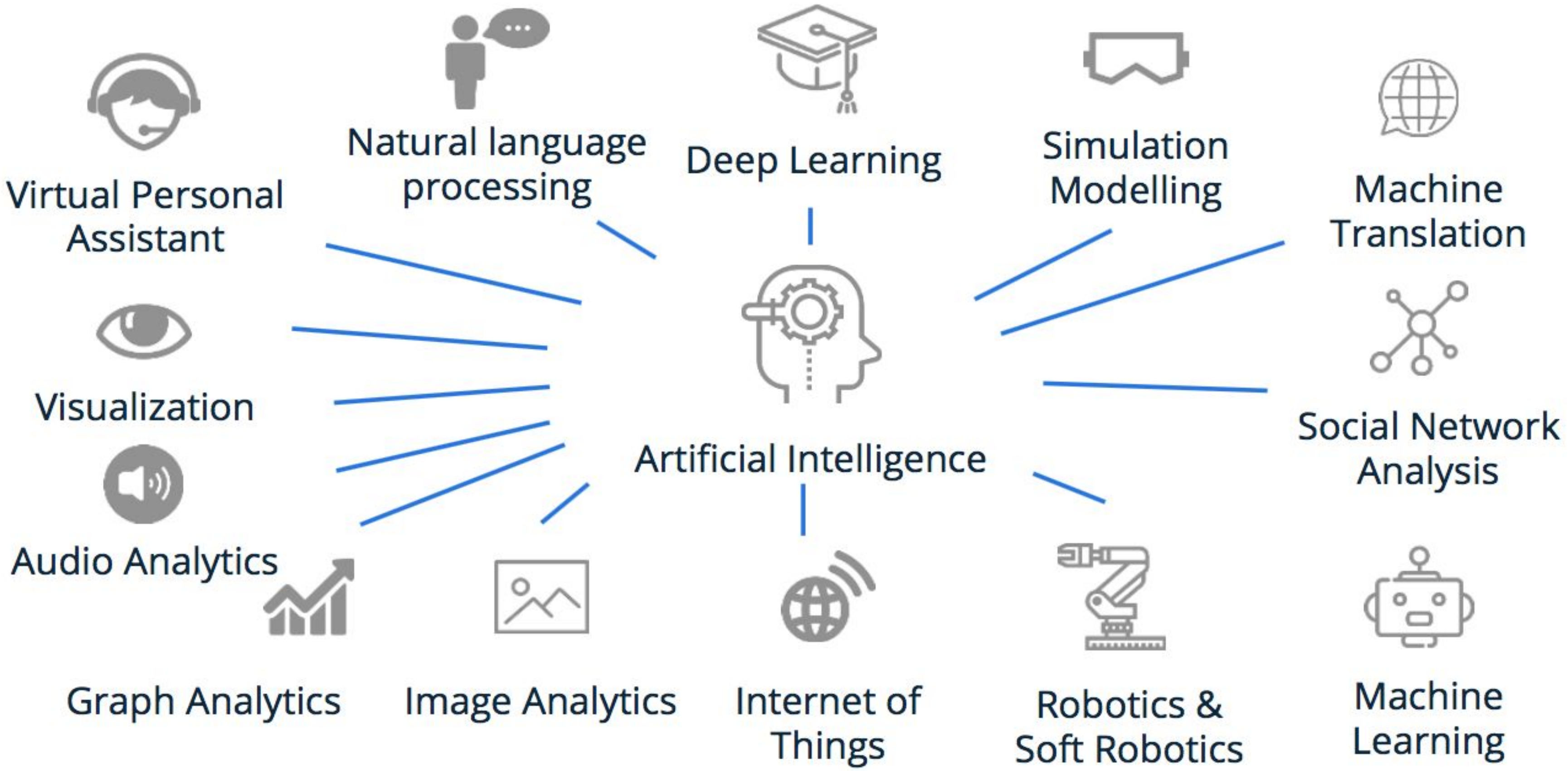
A subset of ML which make the computation of multi-layer neural networks feasible

1950's 1960's 1970's 1980's 1990's 2000's 2010s

Data Roles

- Business Analyst
- Data Engineers
- Data Analyst
- Data Scientist
- Machine Learning Engineer
- AI researcher

Possible applications for Artificial Intelligence



Companies in Jordan that use AI in their business

طقس العرب
ArabiaWeather

zain

orange™

Microsoft

قلم

arabot.
Intelligent Arabic Bot

Jawaker

البنك العربي
ARAB BANK

stech

pwc

موضوع
www.mawdoo3.com

Open>Insights

amazon

sts

بنك الاتحاد
Bank al Etihad

Atypon

SAMSUNG

O'REILLY®

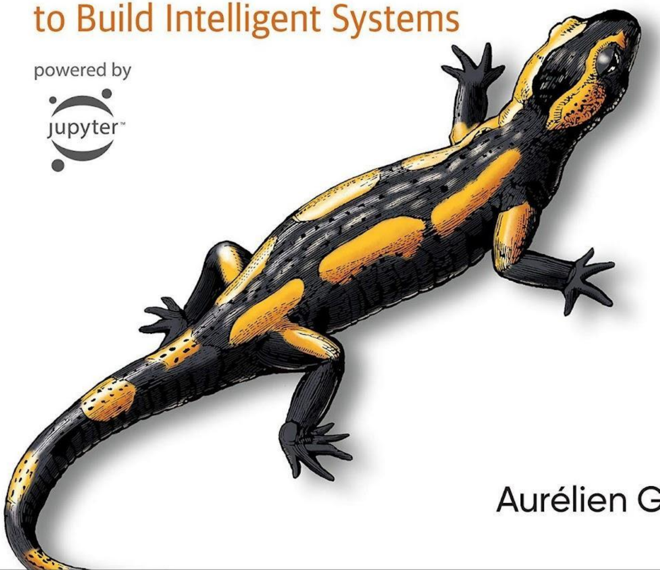
2nd Edition
Updated for TensorFlow 2

Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow

Concepts, Tools, and Techniques to Build Intelligent Systems

powered by

jupyter™



Aurélien Géron

SHAI
Share Artificial Intelligence

Things to do to get the most benefit from this course

- Never miss a lecture
- Use notebook
- Ask questions
- Answer questions
- Study at home
- Write code by yourself

1. Introduction to Artificial Intelligence
2. The Fundamentals of Machine Learning
3. End-to-End Machine Learning Project
4. Classification
5. Training Models
6. Decision tree

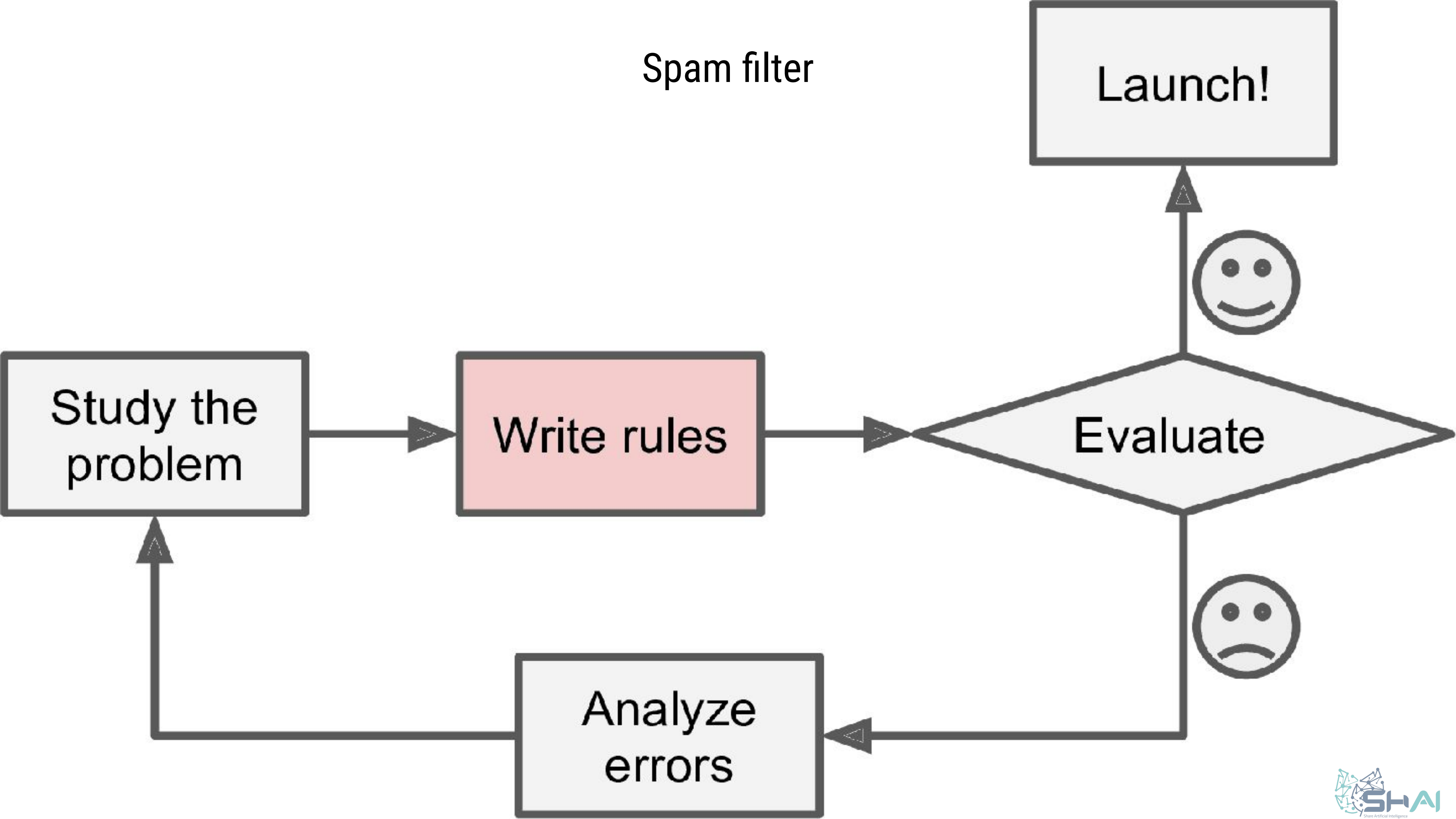
Chapter 1

The Fundamentals of Machine Learning

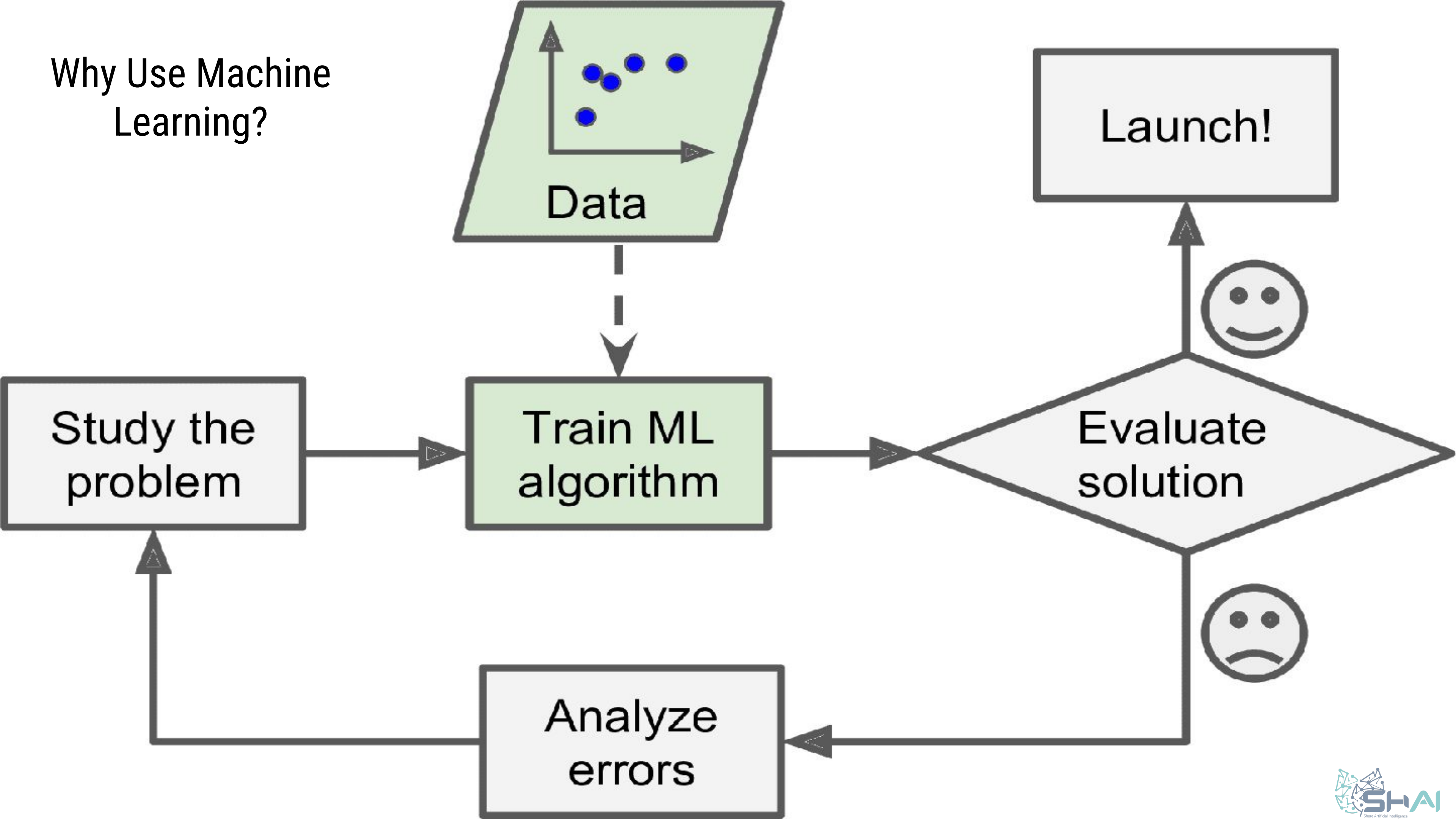
- What is ML
- Why use ML
- Examples of Applications
- Types of ML
    - Supervised
    - Unsupervised Learning
    - Semi Supervised Learning
    - Self Supervised Learning
    - Reinforcement Learning
- Batch and Online Learning
- Main Challenges of Machine Learning
    - Insufficient Quantity of Training Data
    - Nonrepresentative Training Data
    - Poor-Quality Data
    - Irrelevant Features
    - Overfitting the Training Data
    - Underfitting the Training Data
- Testing and Validation
- Hyperparameter Tuning and Model Selection
- Data Mismatch

What Is Machine Learning?

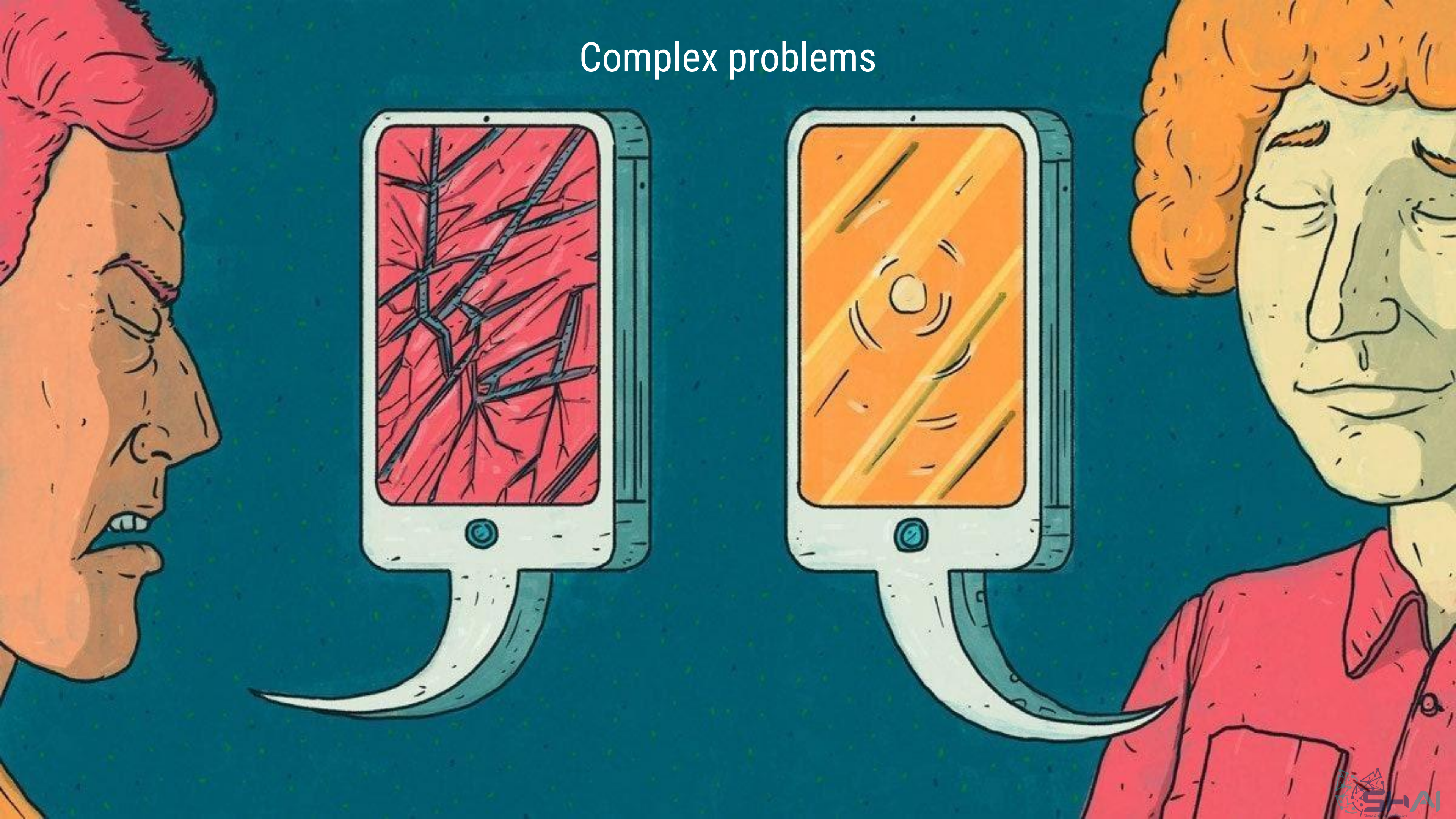
Spam filter


Launch!
Study the problem
Write rules
Evaluate
Analyze errors

Why Use Machine Learning?



Complex problems




DATA
MINING
PATTERN
RECOGNITION
INFORMATION
TECHNOLOGY
SECURITY
INTERNET
WEB
CONNECTION
AGGREGATION
INNOVATION
BUSINESS
SOFTWARE
CODE
SOLUTION
COLLECT
BUG
ERROR
STORAGE
CONNECTIVITY
ENCRYPTION
MANAGEMENT
DECISION
COMPUTER
CONCEPT
TRANSFORMATION
EXTRACTION
NETWORK
ANALYTICS
ELABORATION
BINARY
ANALYSIS
PROCESS
SYSTEM
CALCULATION
MACHINE
LEARNING
CUSTOMER
TROUBLESHOOT

To summarize, Machine Learning is great for:

- Problems for which existing solutions require **a lot of fine-tuning** or **long lists** of rules: one Machine Learning algorithm can often simplify code and perform better than the traditional approach.
- **Complex problems** for which using a traditional approach yields no good solution: the best Machine Learning techniques can perhaps find a solution.
- **Fluctuating environments**: a Machine Learning system can adapt to new data.
- **Getting insights** about complex problems and large amounts of data.

Examples of AI Applications

01
Detecting tumors
Semantic segmentation

02
Flagging offensive comments
Text classification, using NLP tools

03
Chatbots and personal assistant
Many NLP components, including NLU

04
Segmenting Clients
Clustering

05
Recommending a product
Recommender system

06
Intelligent bot
Reinforcement Learning

Types of Machine Learning

- Supervised Learning
- Unsupervised Learning
- Semi Supervised Learning
- Self Supervised Learning
- Reinforcement Learning

01

Supervised Learning

In supervised learning, the training set you feed to the algorithm includes the **desired solutions, called labels**

Supervised

- Regression
- classification

Some of the most important supervised learning algorithms

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

02
Unsupervised learning

Some of the most important unsupervised learning algorithms

- Clustering
    - K-Means
    - DBSCAN
    - Hierarchical Cluster Analysis (HCA)

- Anomaly detection and novelty detection
    - One-class SVM
    - Isolation Forest

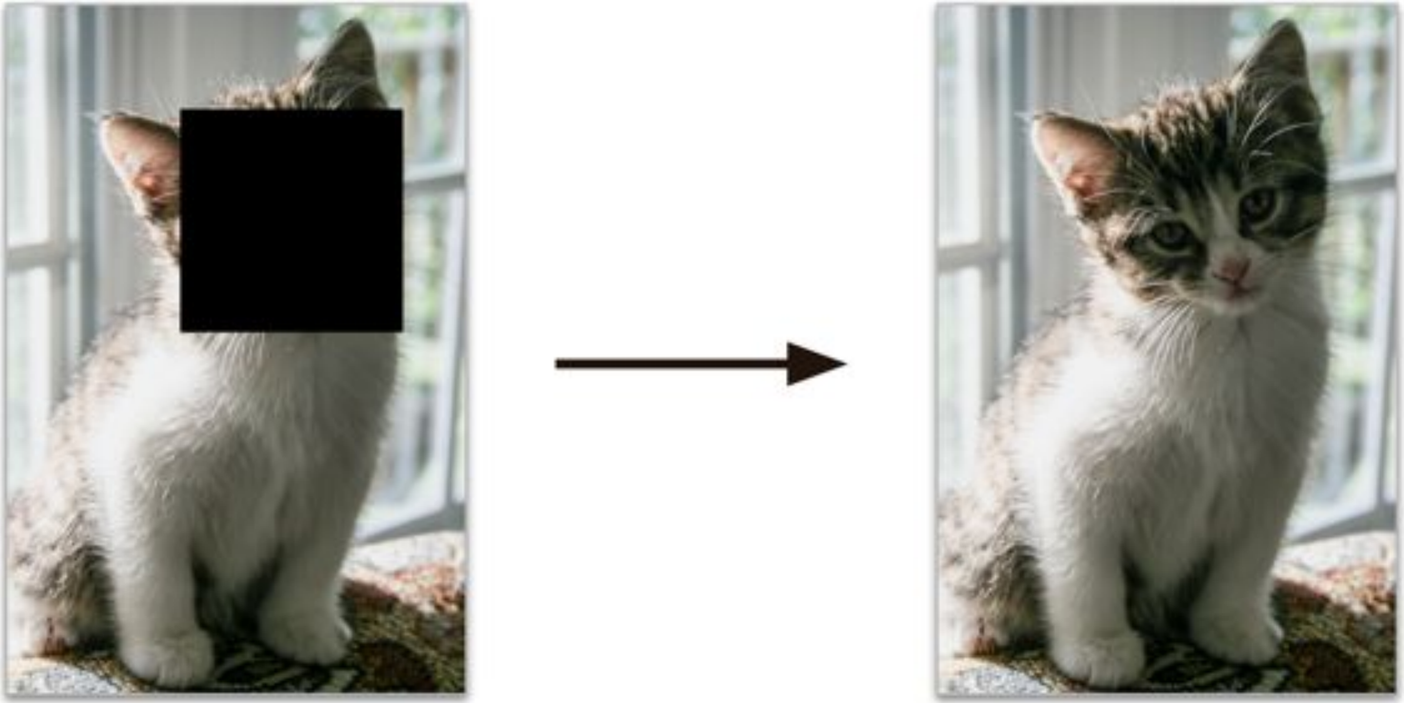
- Visualization and dimensionality reduction
    - Principal Component Analysis (PCA)
    - Kernel PCA
    - Locally Linear Embedding (LLE)
    - t-Distributed Stochastic Neighbor Embedding (t-SNE)

03

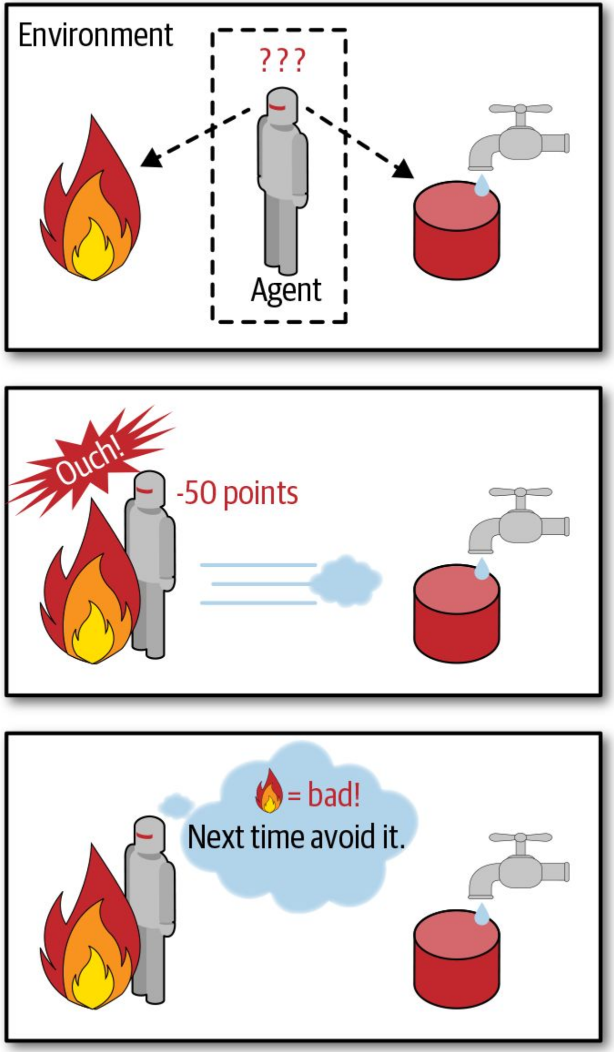
Semi Supervised learning

04

Self Supervised learning



05
Reinforcement learning



1. Observe
2. Select action using policy
3. Action!
4. Get reward or penalty
5. Update policy (learning step)
6. Iterate until an optimal policy is found

Batch and Online Learning

Another criterion used to classify Machine Learning systems is whether or not the system can learn incrementally from a stream of incoming data.

Batch learning

In batch learning, the system is incapable of learning incrementally: it must be **trained using all the available data.** This will generally **take a lot of time** and **computing resources,** so it is typically done offline. First the system is trained, and then it is launched into production and runs without learning anymore; it just applies what it has learned. This is called **offline learning.**

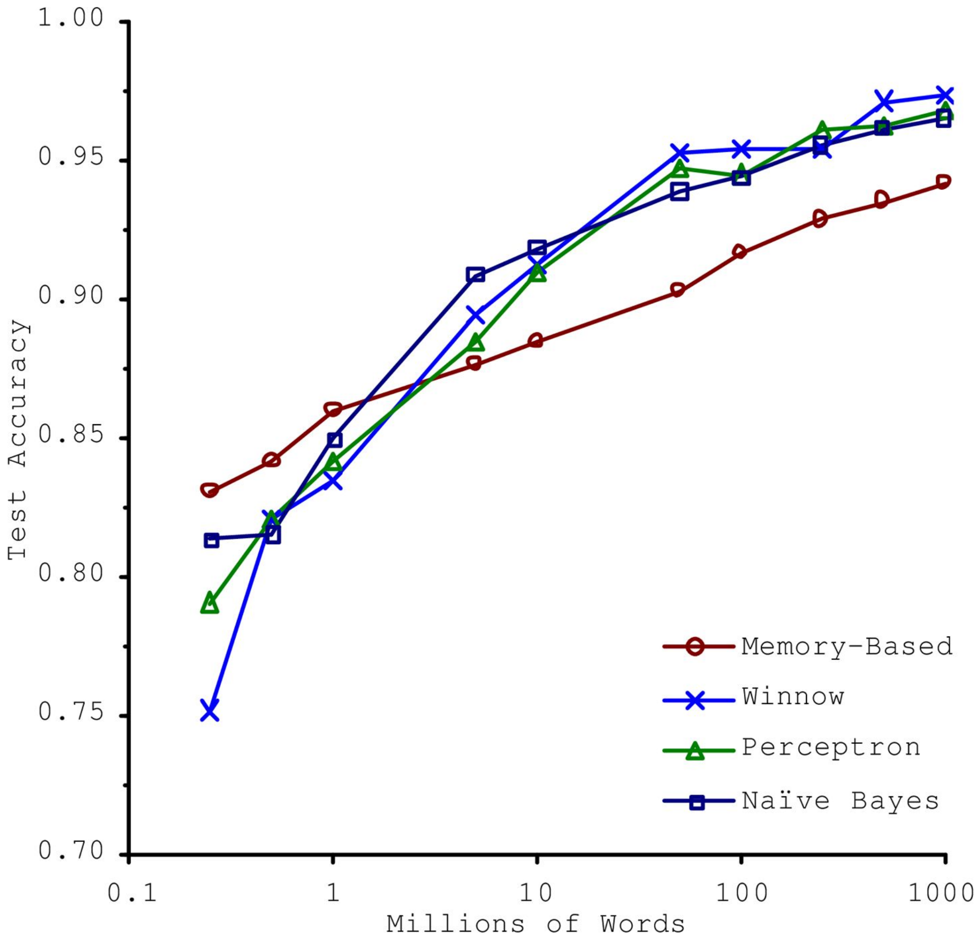
Online learning

In online learning, you train the system incrementally by feeding it data instances sequentially, either **individually** or in small groups called **mini-batches**. Each learning step is **fast and cheap**, so the system can learn about new data on the fly, as it arrives

Main Challenges of Machine Learning

- Insufficient Quantity of Training Data
- Nonrepresentative Training Data
- Poor-Quality Data
- Irrelevant Features
- Overfitting the Training Data
- Underfitting the Training Data

Insufficient Quantity of Training Data



Learning style

Possibly the most trivial form of learning is simply to learn by heart. If you were to create a spam filter this way, it would just flag all emails that are identical to emails that have already been flagged by users—not the worst solution, but certainly not the best

Instance-Based

Another way to generalize from a set of examples is to build a model of these exam- ples and then use that model to make predictions

Model-Based Learning

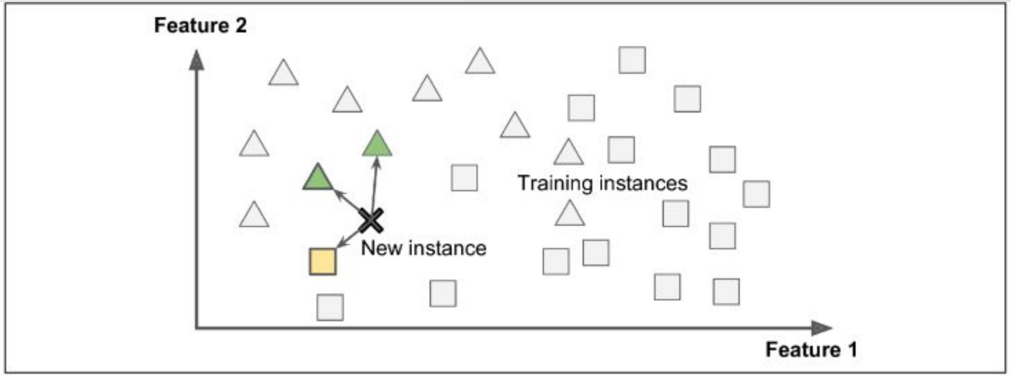
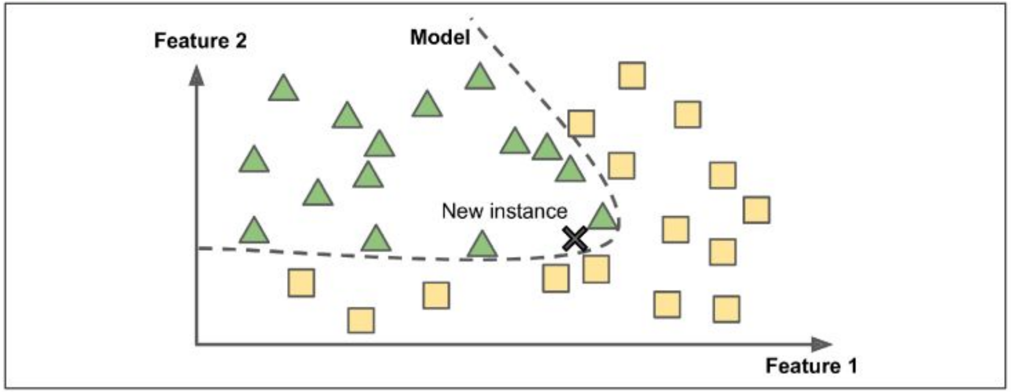


Figure 1-15. Instance-based learning

Model-based learning

Another way to generalize from a set of examples is to build a model of these examples and then use that model to make *predictions*. This is called *model-based learning* (Figure 1-16).



Linear model

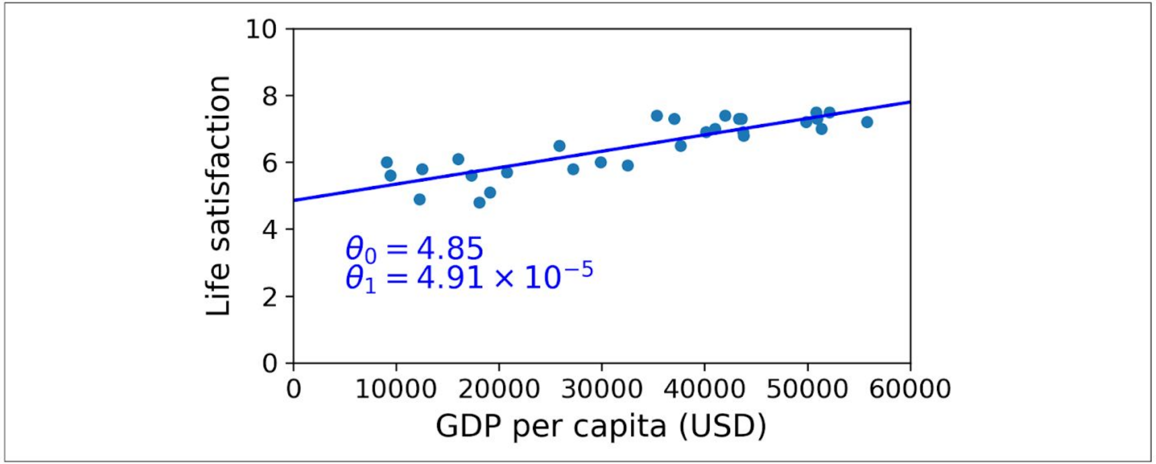
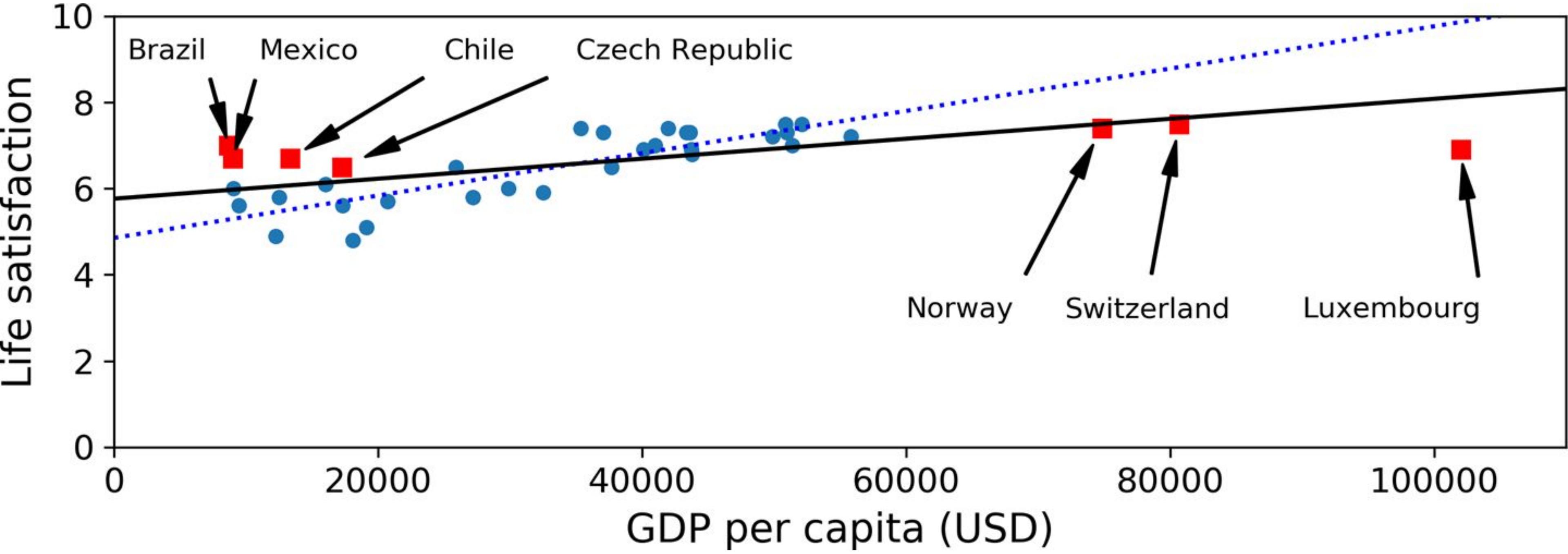


Figure 1-19. The linear model that fits the training data best

Nonrepresentative Training Data



Poor-Quality Data

- If some instances are clearly outliers, it may help to simply discard them or try to fix the errors manually.

- If some instances are missing a few features (e.g., 5% of your customers did not specify their age), you must decide whether you want to 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

- 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

There’s just one last important topic to cover: once you have trained a model, you don’t want to just “hope” it generalizes to new cases. You want to evaluate it and fine-tune it if necessary. Let’s see how to do that

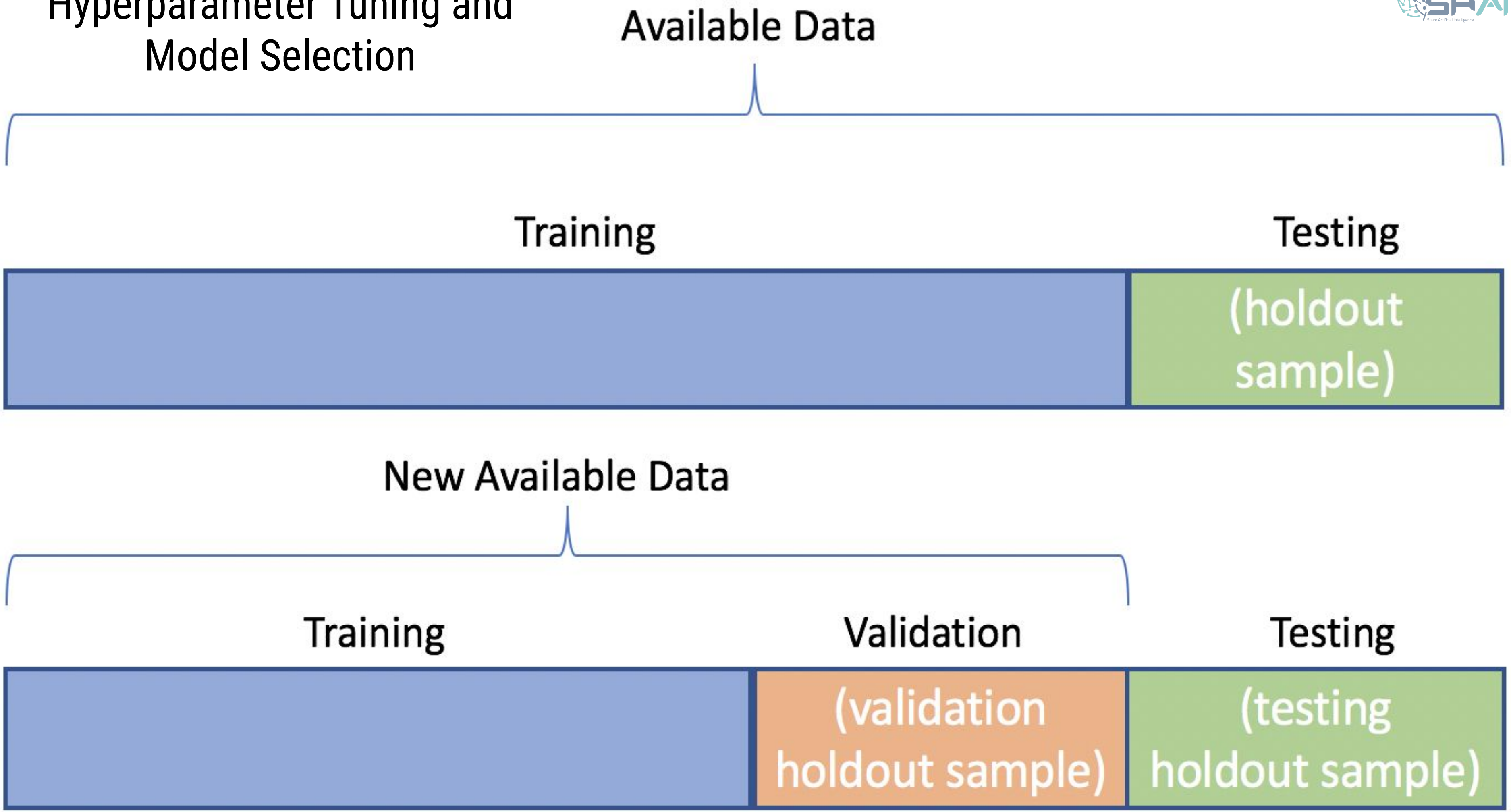
Testing and Validating

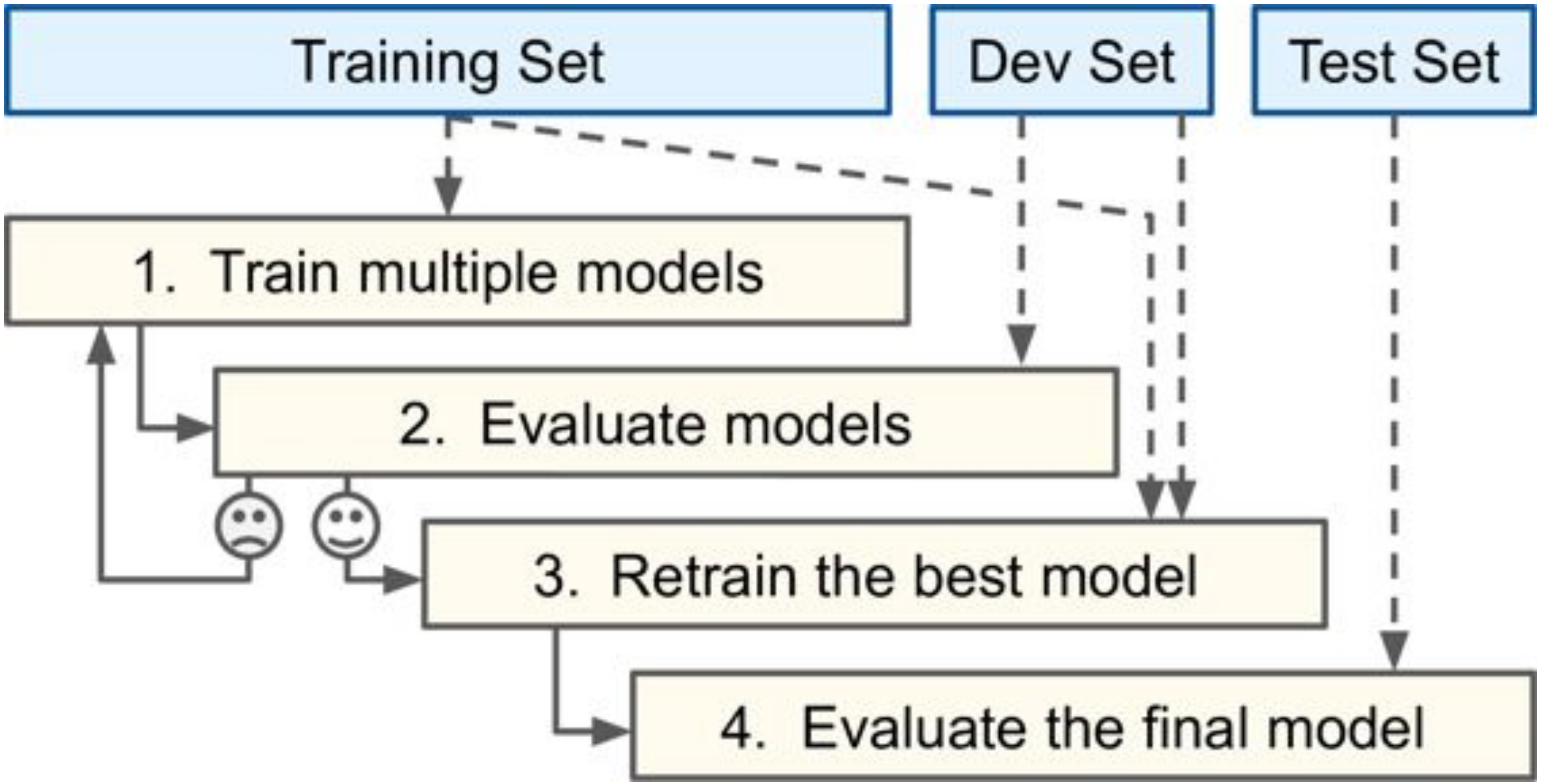
The only way to know how well a model will generalize to new cases is to actually try it out on new cases. One way to do that is to put your model in production and monitor how well it performs. This works well, but if your model is horribly bad, your users will complain—not the best idea.

A better option is to split your data into two sets: the **training set and the test set**. You train your model using the training set, and you test it using the test set. The error rate on new cases is called the **generalization error (or out-of-sample error)**, and by evaluating your model on the test set, you get an estimate of this 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

Hyperparameter Tuning and Model Selection




Training Set
Dev Set
Test Set
1. Train multiple models
2. Evaluate models
3. Retrain the best model
4. Evaluate the final model

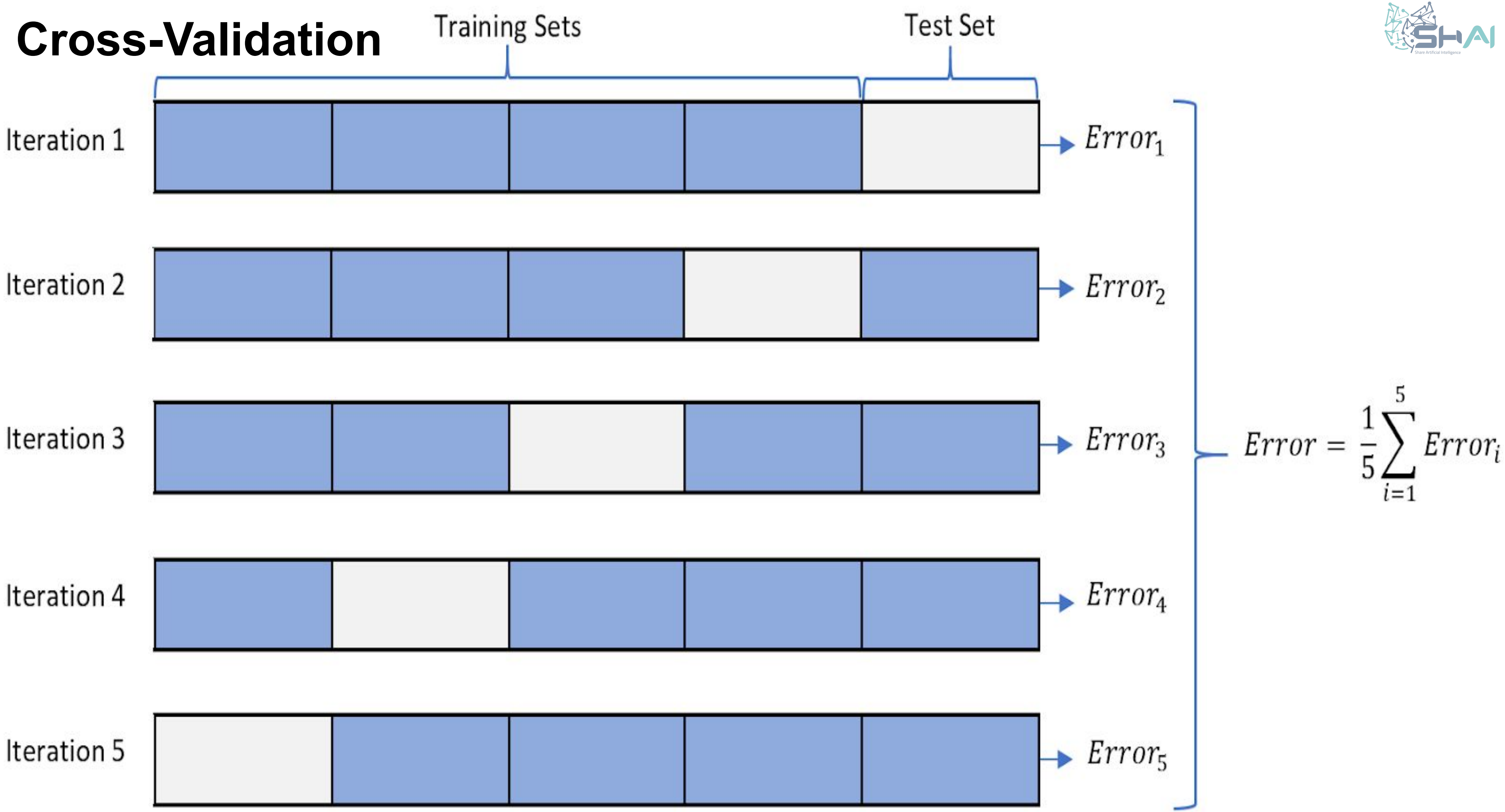
Problems with Validation set

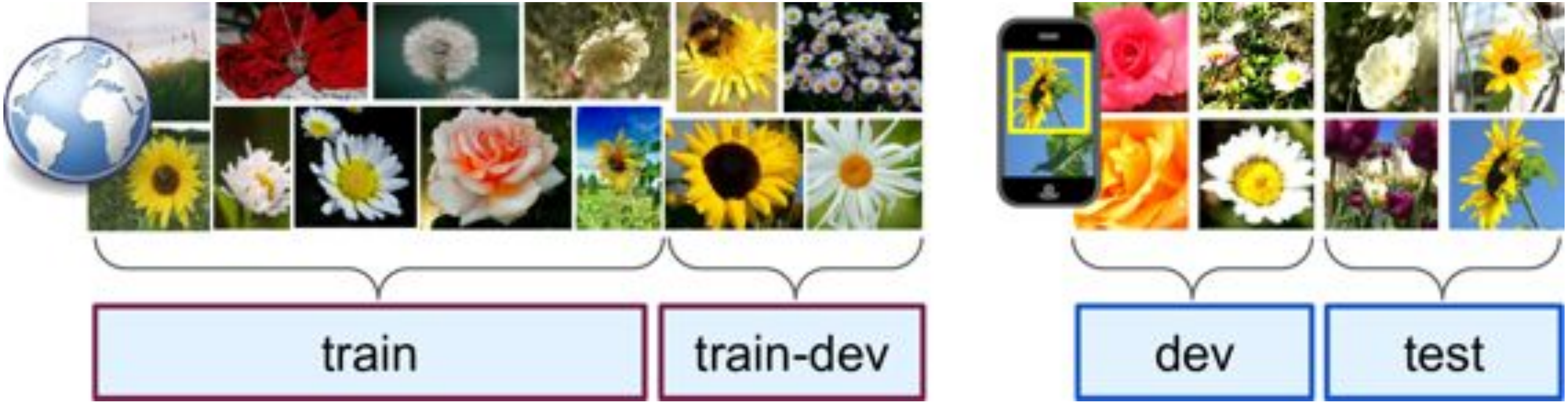
Too Big

Too Small

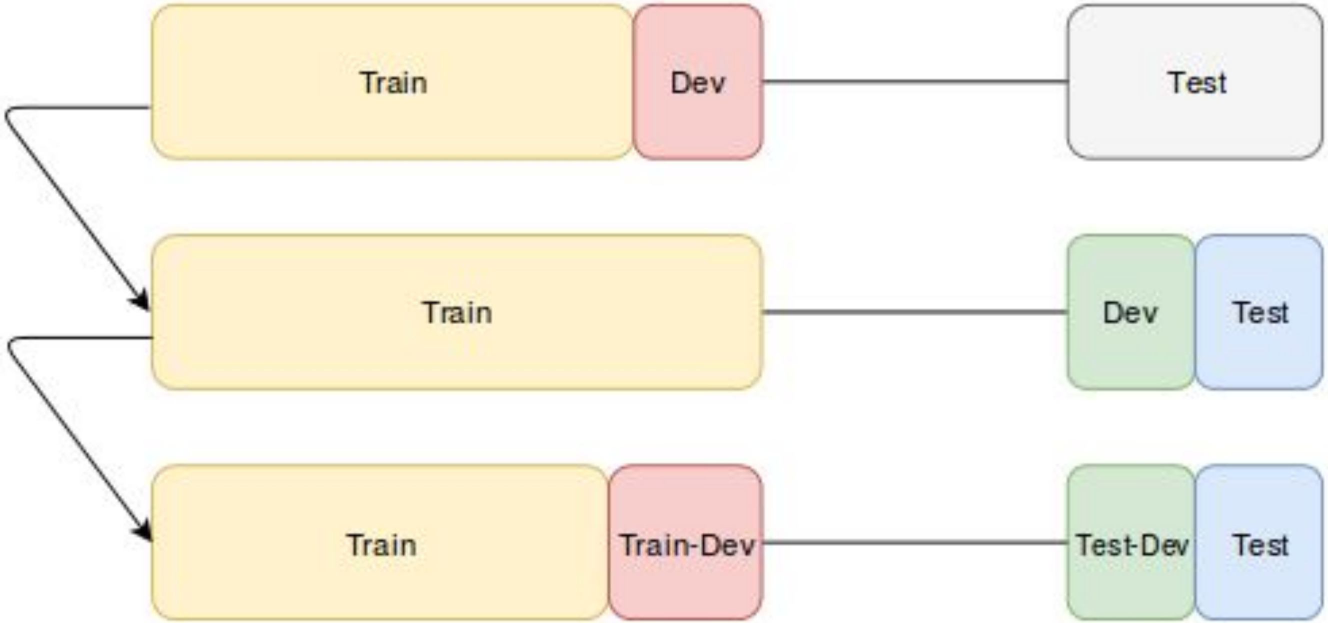
Cross Validation

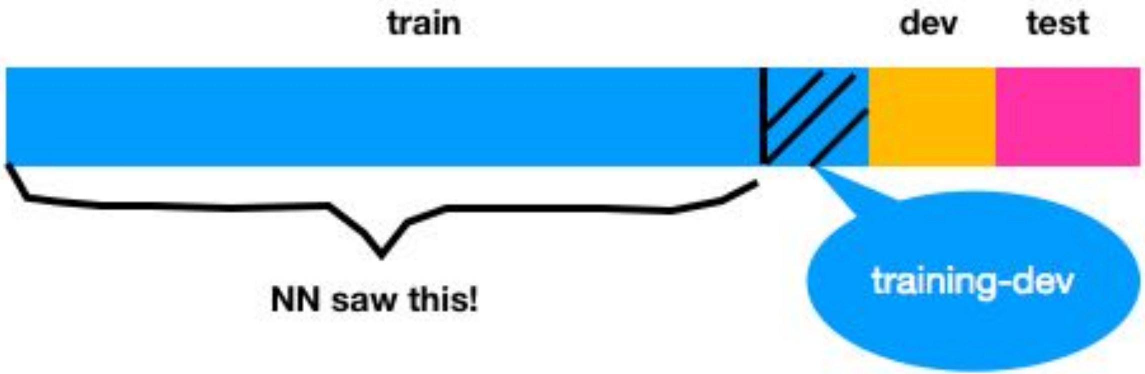
Cross-Validation

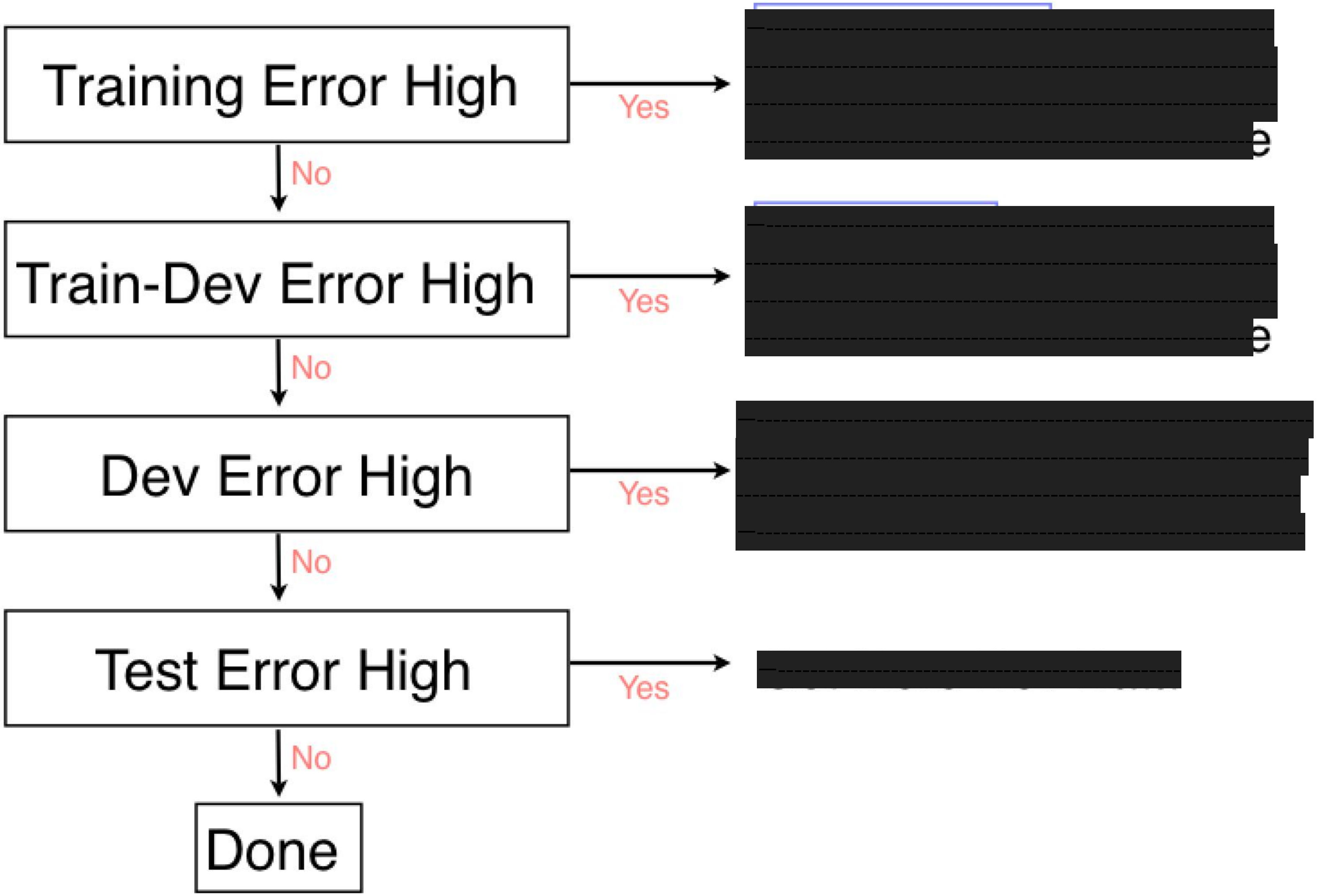



train
train-dev
dev
test

Data Mismatch






Training Error High
Yes
[redacted]
No
Train-Dev Error High
Yes
[redacted]
No
Dev Error High
Yes
[redacted]
No
Test Error High
Yes
[redacted]
No
Done

Is it possible to have a higher train error than a test error in machine learning? And why is that?

Questions?