Introduction to Machine Learning

MACHINE LEARNING FOR DATA SCIENCE: INDUSTRY APPLICATIONS

Dr. Roselyn Isimeto
Lecturer
University of Lagos

David Akanji
Data Scientists
University of Lagos

Agenda

- **What is machine learning?**
- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Generative models

Machine Learning:

“The science of getting computers to act without being explicitly programmed.”

- Andrew Ng

When should we use machine learning?

- Machine learning is not appropriate for every task!
  - If you can solve it analytically, that's better! (More explainable)
- When you have access to **data which "matches" the task**
- When **errors are allowable**
- When it's **cost effective**
  - Machine learning costs include (i) **dataset creation and processing** and (ii) **model development, deployment, and maintenance**

Machine learning strengths and limitations

Strengths

- Performing tasks at scale
- Modeling complex systems
- Generating derived data
- Integrating with other methods, e.g. domain and physical models

Limitations

- “Garbage in, garbage out”
- Inherits biases in data + human design/use
- Assumes patterns are persistent
- Finds correlation, not causation

Types of learning

- **Supervised learning**
  Learning to predict or classify labels based on labeled input data
  Performance feedback

- **Unsupervised learning**
  Finding patterns in unlabeled data
  No performance feedback

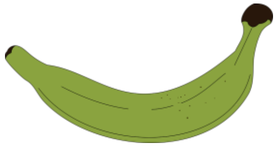
- **Reinforcement learning**
  Learning well-performing behavior from state observations and rewards
  Performance feedback

Supervised vs. Unsupervised learning

Supervised

Apple

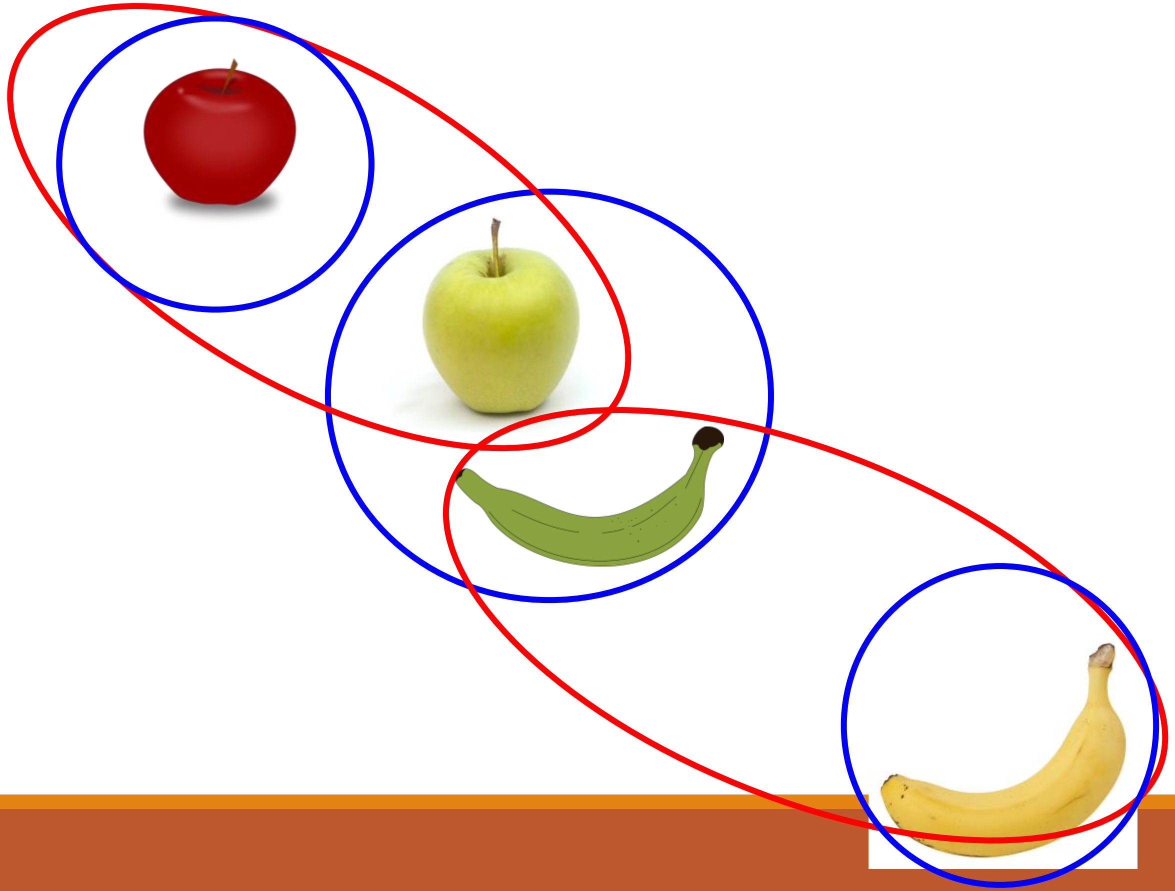
Apple



Banana

Banana

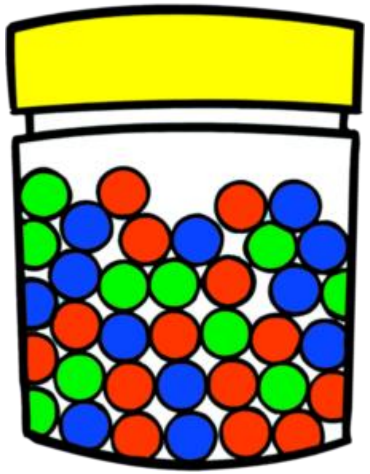
Unsupervised



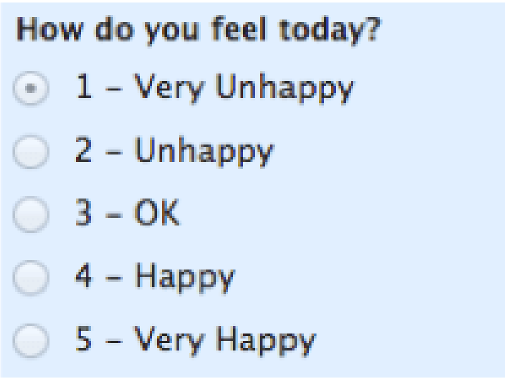
Data Types

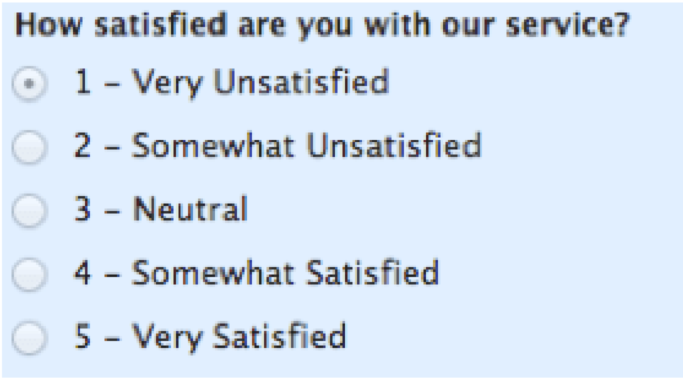
- **Continuous**

- **Discrete**
- **Categorical**



- **Binary**
  - Special case of categorical
- **Ordinal**

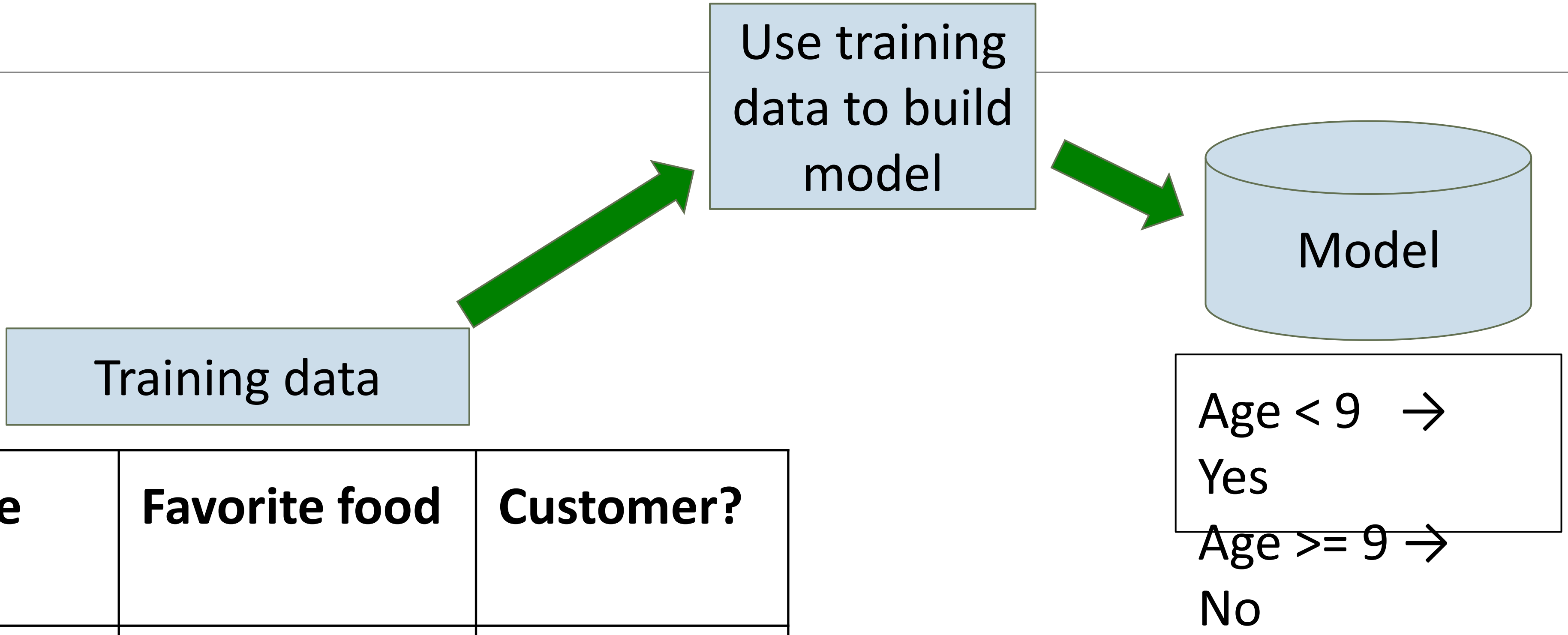




Agenda

- What is machine learning?
- **Supervised learning**
- Unsupervised learning
- Reinforcement learning
- Generative models

Supervised learning: training



Person	**Age**	**Favorite food**	**Customer?**
Cat	8	Sushi	Yes
Elephant	13	Salad	No
Dog	7	Sandwich	Yes

Supervised learning: test / prediction

Model

Age < 9 → Yes
Age >= 9 → No

Evaluate performance using test data

Person	Age	Favorite food	Customer?
Penguin	6	Sushi	Yes

Use on unseen real-world data

Person	Age	Favorite food
Chicken	8	Salad

Customer?
Yes

Supervised learning: categorical versus continuous labels

- Classification: **categorical labels**
  - Examples: pregnant or not, from which country, which type of road sign
- Regression: **continuous labels**
  - Examples: future stock price, life expectancy, distance to obstacle

Example: predicting bicycle counts

https://www.climatechange.ai/papers/iclr2023/15

Given: historical data of the number of bicycles in certain locations per hour

Want to predict: number of bicycles in future times at those locations

Which type are the labels? Categorical or continuous?

Supervised models

Model	When to use it?
KNN	➢ Little / no training time, large prediction time ➢ Given small / medium dataset size
Linear / polynomial regression	➢ Linear / polynomial relationship between input and output ➢ Small training time ➢ Given small / medium dataset size
Logistic regression, SVM, decision tree	➢ Categorical output ➢ Given small / medium training time
Neural network	➢ Large training time, large computer ➢ Given large dataset size

k-Nearest Neighbors Algorithm

Training set: n instances, each with a feature vector and an output category

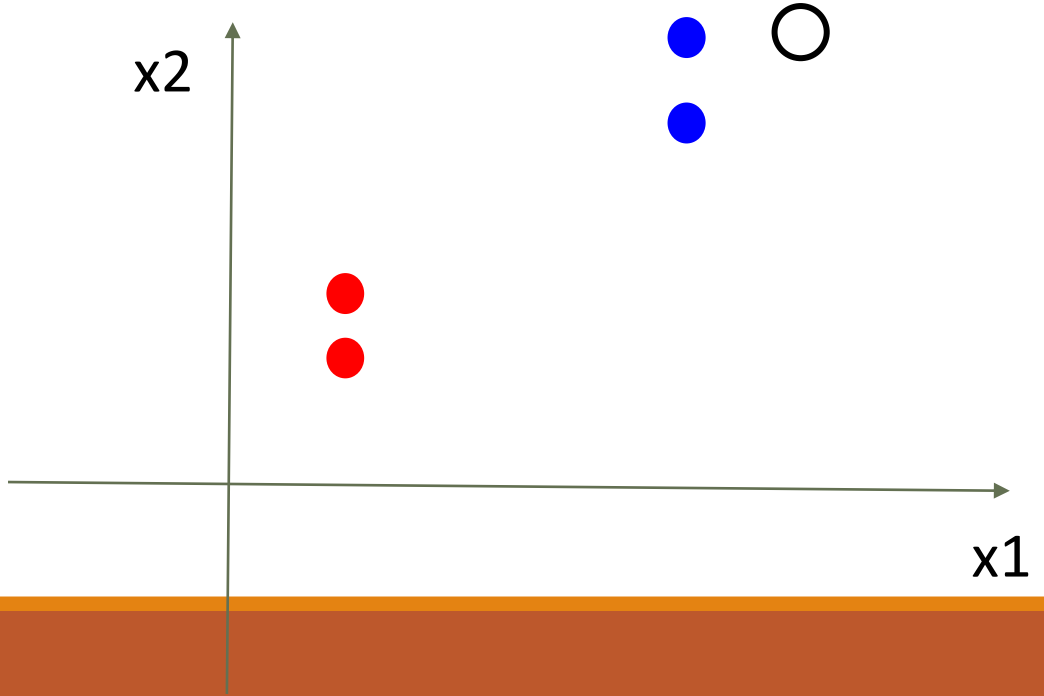
Now, given another (unseen) instance, we want to determine its category

Check the k instances in the training data that are closest to your new instance

- ➢ Categorical: choose the majority of those values
- ➢ Continuous: choose the mean/median of those values

Training set:
(1,2) → red
(1,3) → red
(5,5) → blue
(5,6) → blue

New instance:
(6,6) → blue



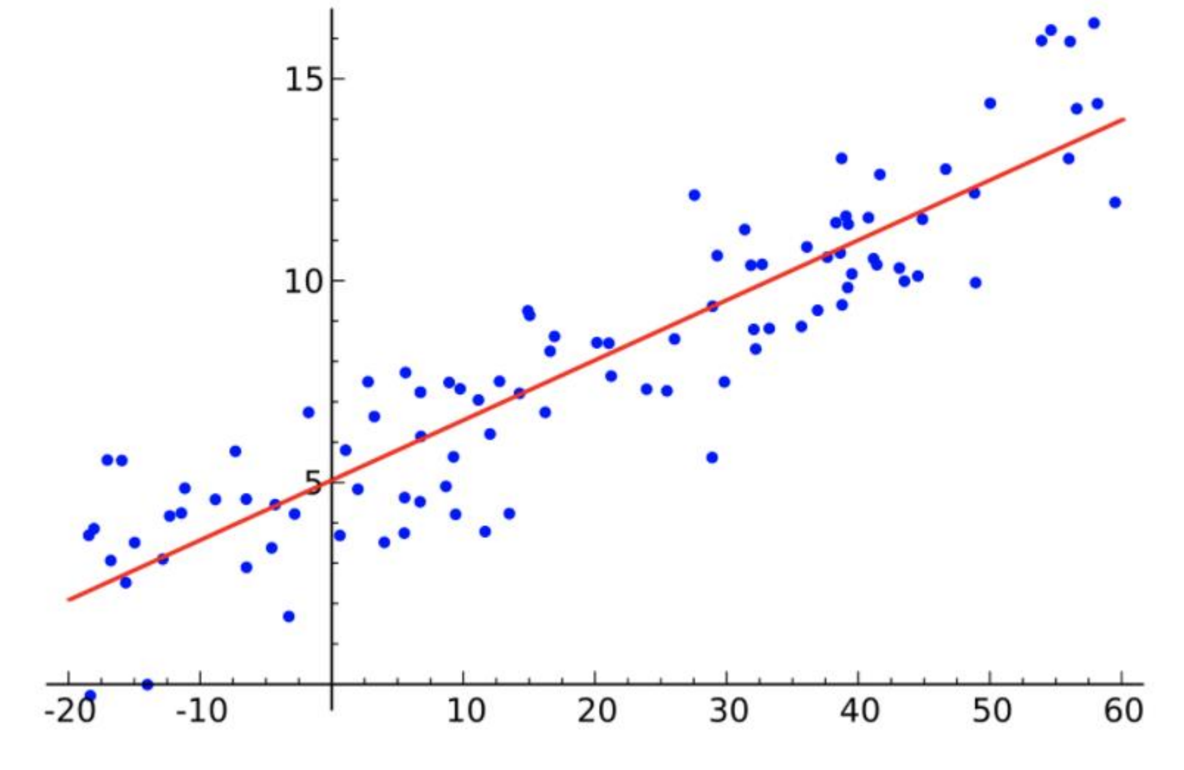
Linear / polynomial regression

Given x ∈ ℝ and y ∈ ℝ

Find a function f: x → y

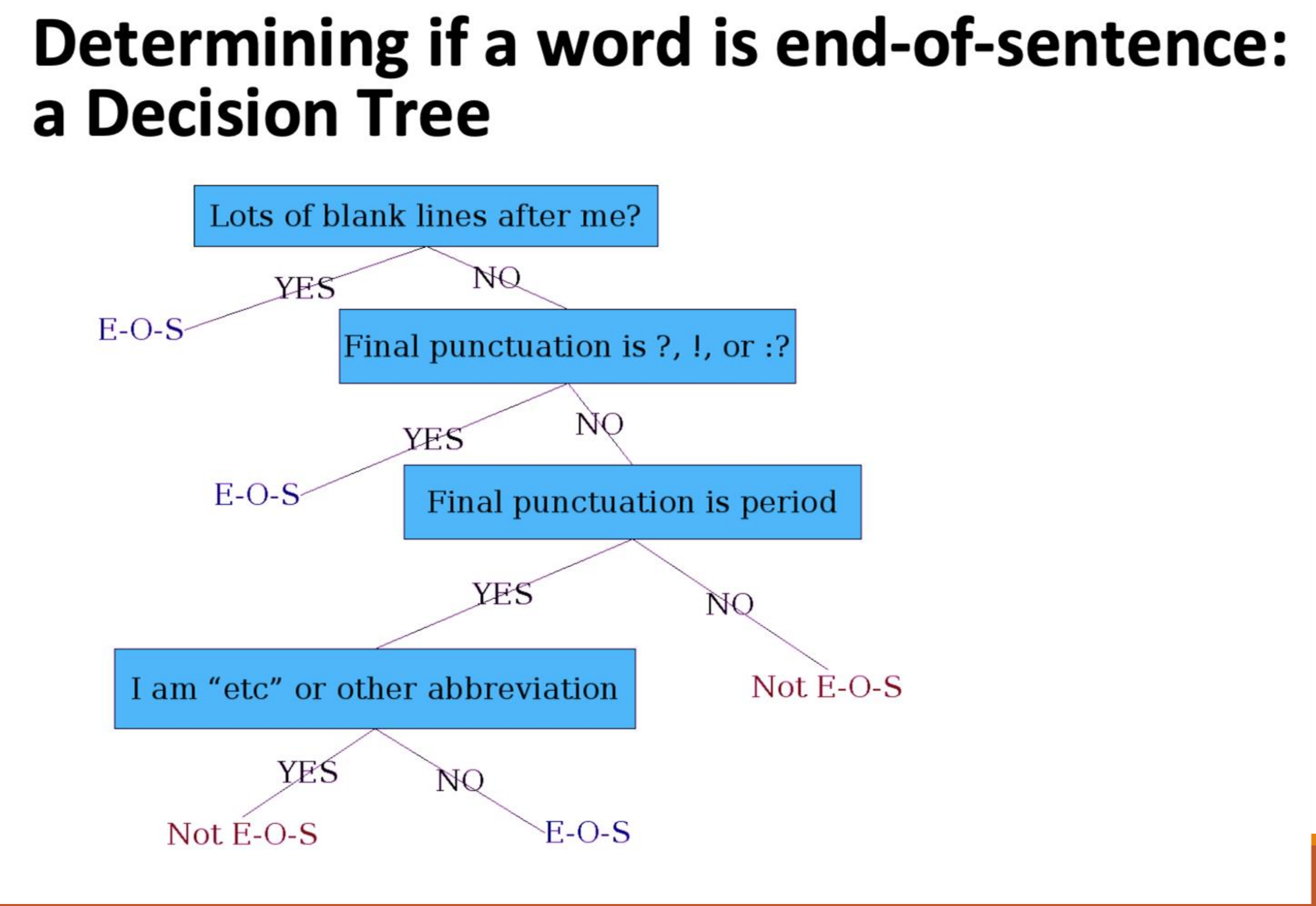
How?

Define a **loss function** ("error") and minimize it!



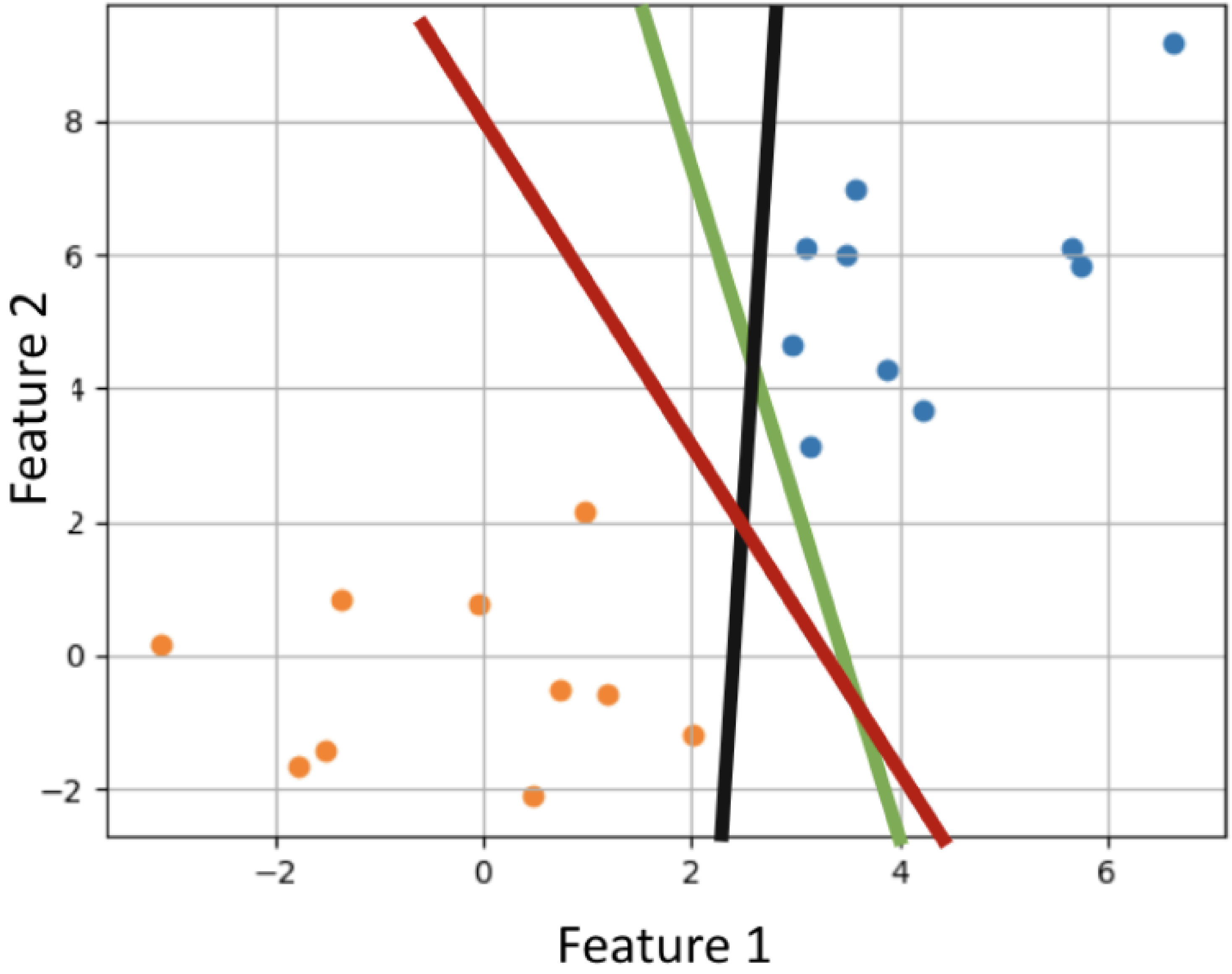
Other supervised classifiers

- **Decision tree**



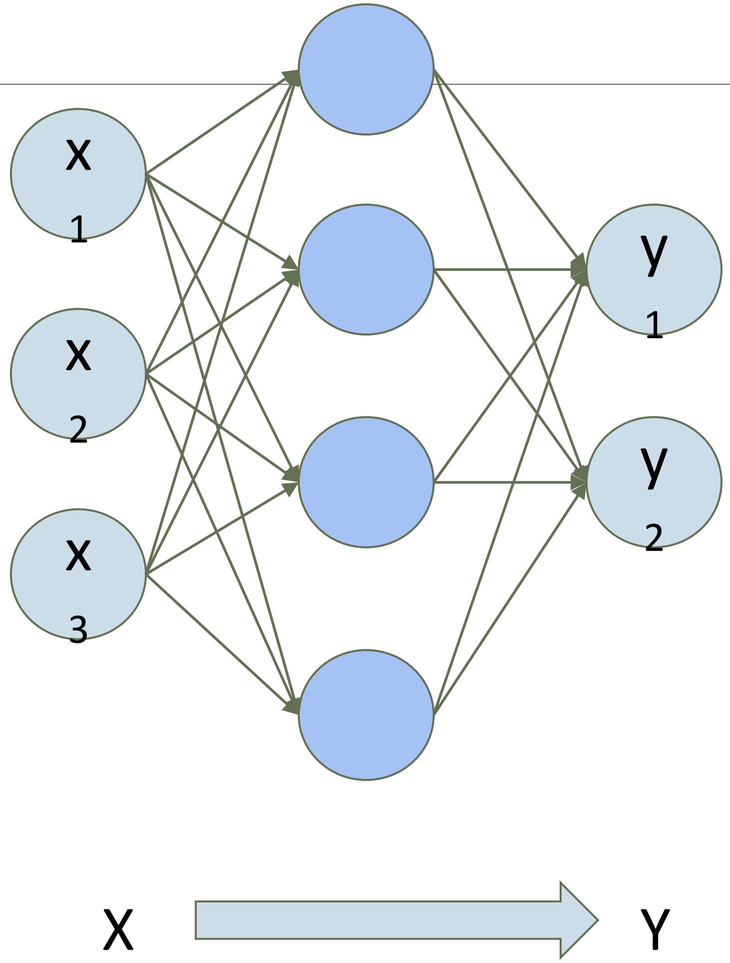
Other supervised classifiers

- **Decision tree**
- **Support Vector Machine**



Other supervised classifiers

- **Decision tree**
- **Support Vector Machine**
- **Neural Network**

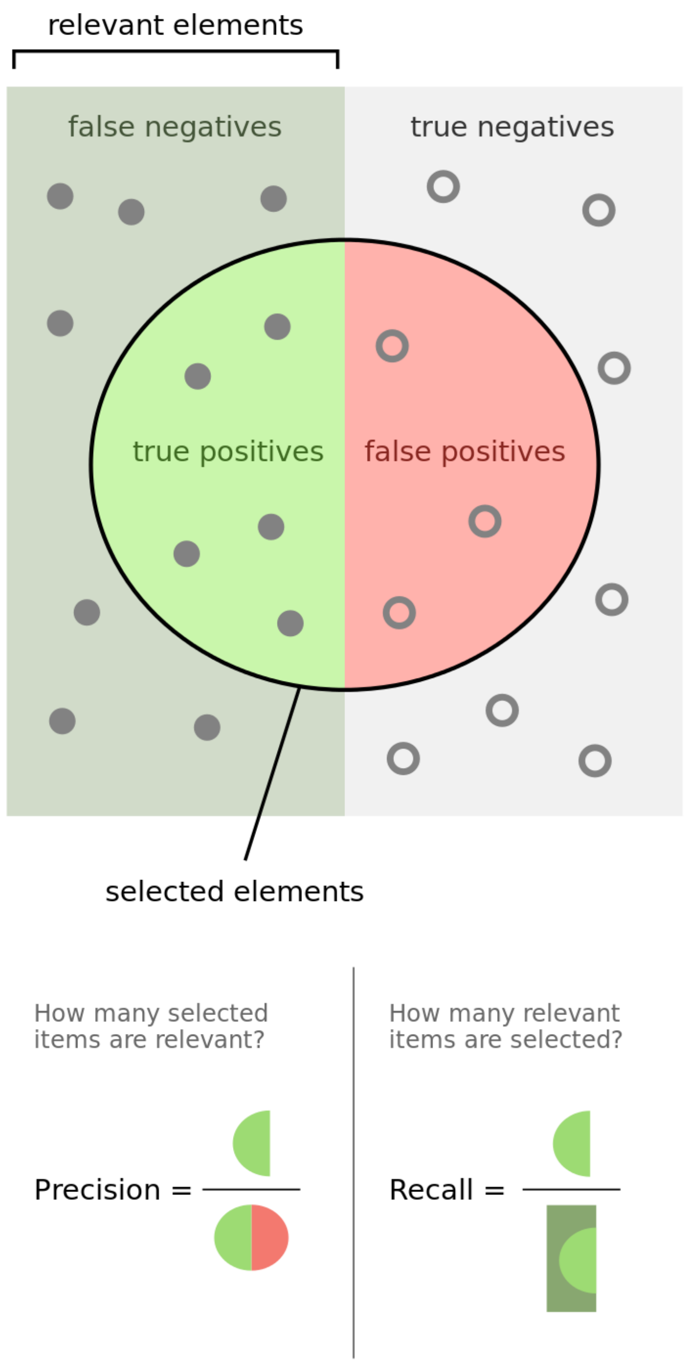


How good is the model?

We define a **metric** to measure and compare accuracy.

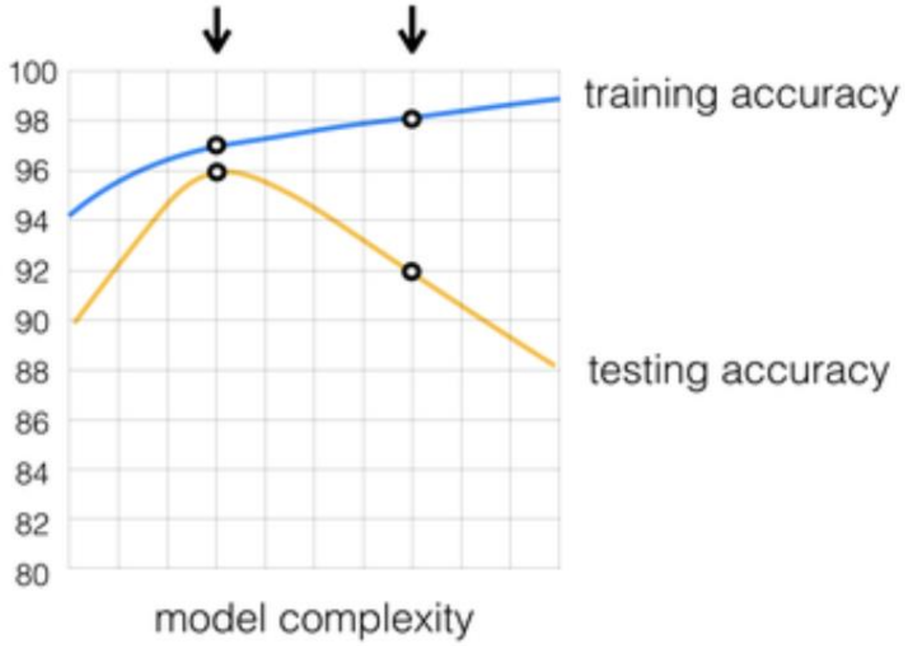
- Precision
  - Out of those tested positive, how many are truly positive?
  - TP / (TP + FP)
- Recall
  - Out of those truly positive, how many tested positive?
  - TP / (TP + FN)
- F1

$$\frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}$$



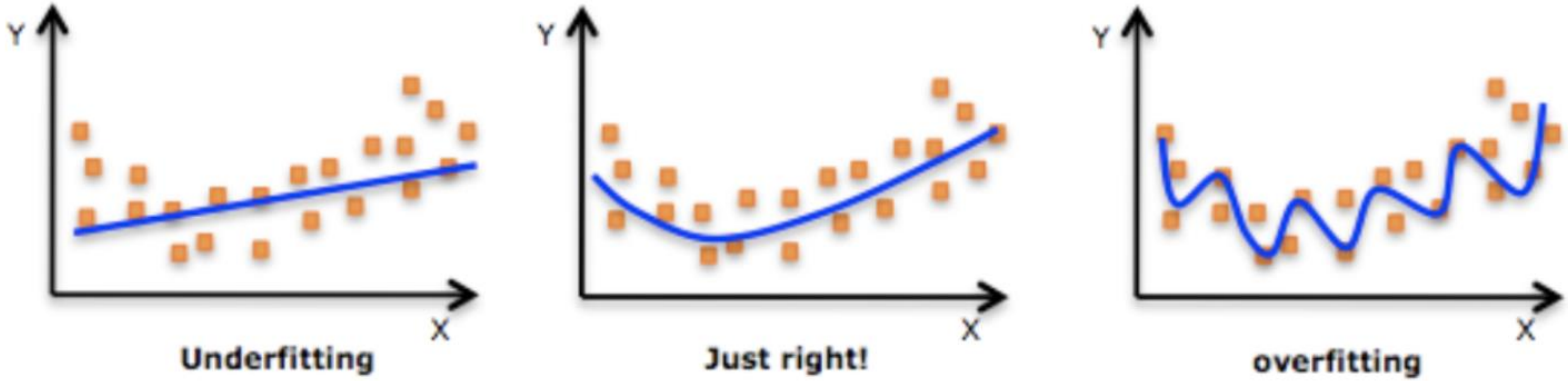
https://en.wikipedia.org/wiki/F1_score

Overfitting



Solution: **Cross-Validation**

Split the training data into two non-overlapping sets. Train on one set, and measure performance on the other. Pick the model that does well on the data that you *didn't* train on.



Supervised models

Model	When to use it?
KNN	➢ Little / no training time, large prediction time ➢ Given small / medium dataset size
Linear / polynomial regression	➢ Linear / polynomial relationship between input and output ➢ Small training time ➢ Given small / medium dataset size
Logistic regression, SVM, decision tree	➢ Categorical output ➢ Given small / medium training time
Neural network	➢ Large training time, large computer ➢ Given large dataset size

Note / life tip

- **Don’t re-implement it yourself!**
  - Unless you are doing research on the method itself, you are trying to learn how it works, or you are coding in an obscure language where it isn’t already implemented
  - The already implemented versions are widely used and tested

Note / life tip

- **Don't re-implement it yourself!**
  - Unless you are doing research on the method itself, you are trying to learn how it works, or you are coding in an obscure language where it isn't already implemented
  - The already implemented versions are widely used and tested
- **Use these common tools:**
  - Scikit-learn has most supervised and unsupervised methods you might need
  - If you want to build a custom neural network, try using Pytorch or Tensorflow
  - There are many task-specific libraries

Session Agenda

1. What is Machine Learning
2. What is the ML Process
3. Problem Formulation
4. Loading the raw data
5. Data Pre-processing:
   - Cleaning
   - Distributions
   - Vizualizations
   - Transformations
   - Feature Engineering
6. Splitting the data
7. Running Regression
8. Evaluation Metrics

What is Machine Learning (ML) in a nutshell

Machine learning is the science (and art) of programming computers so they can learn from data” by Aurélien Géron book (Hands-On Machine Learning with Scikit-Learn and TensorFlow)

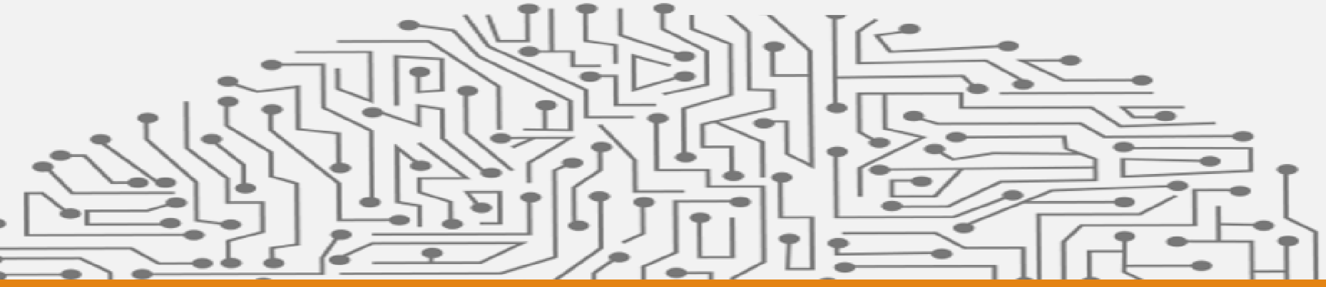
- ML uses statistical models and algorithms to perform tasks like predictions & classifications without explicit instructions

- ML is a subset of Artificial Intelligence

The Machine Learning Process

1. Problem Formulation

- What question are we trying to answer?



The Machine Learning Process

1. Problem Formulation

- What question are we trying to answer?

2. Raw Data Gathering

- Identifying the data sources we need to answer the question

3. Data Pre-processing

- Exploratory Data Analysis (EDA)
- Cleaning & aggregating
- Joins & Transformations
- Distributions, normalization & scaling
- Converting categorical values to numeric representation
- Feature Selection
- Feature Engineering

4. Splitting the Data

- Break the data into Train (70%) – Test (20%) – Evaluation (10%)

- Separate Xs & y variable(s)

5. ML Model Selection & Training

- **Regression models** when predicting a continues number. Examples: LR, RFR, SVR, NNR, etc.
- **Classification models** when predicting a class. Examples: Logistic Regression, Naïve Bayes, Decision Trees, RF, Knn, SVM, NN, etc
- **Unsupervised models** when investigating relationships / clustering. Examples: K-means, hierarchical clustering, PCA, etc.
- **Time Series models** when predicting the future values of time series variable. Examples: Arima, Auto-Arima, exponential smoothing, prophet, etc

- Train the model on 70% of the data

6. Model Evaluation

- Evaluate the model on the 20% of the data

- Metrics for Regression: R^2, MSE, RMSE, etc.

- Metrics for Classification: Accuracy, Log Loss, Confusion Matrix, AUC, etc.

- Metrics for Unsupervised: Inertia, Adjusted Rand Index, etc.

7. Parameter Tuning

Trying to improve the Evaluation Metrics of the model by adjusting their hyperparameters.

Examples: activation function, regularization parameter C, different weights or distributions, penalty parameters, gamma function, training steps, learning rate, etc.

Problem Formulation

- We want to understand the factors that affect car prices
- We want to be able to predict car prices based on our data/variables

Raw data: https://www.kaggle.com/datasets/shaistashaikh/carprice-assignment