Learning from Data

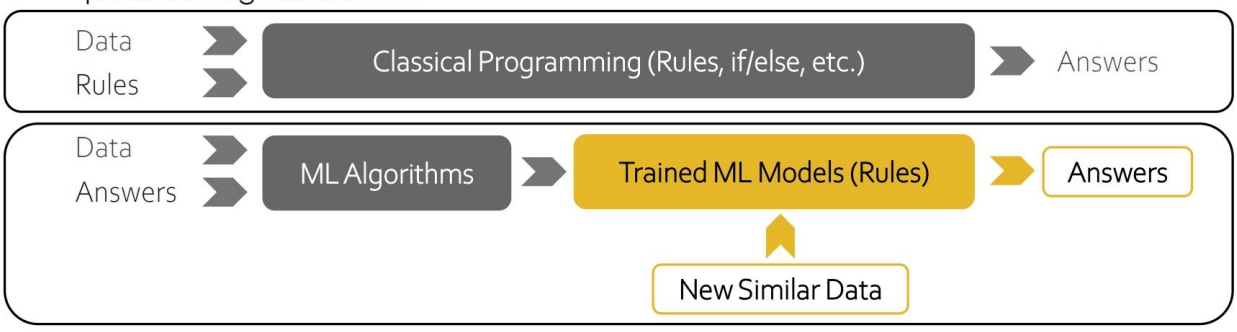
Outline

- Classical programming vs. machine learning.
- What is machine learning?
- Why use machine learning?
- Types machine learning?
- Example of a machine learning project.

Classical programming vs. machine Learning

Classical programming vs. machine learning.

- In **classical programming**, humans input rules (a program) and data to be processed according to these rules, and outcome answers.
- With **machine learning (ML)**, humans input data as well as the answers expected from the data, and outcome the rules. These rules can then be applied to new data to produce original answers.



Classical programming vs. machine learning

- An ML is **trained** rather than explicitly programmed. It's presented with many examples relevant to a task, and it finds statistical structure in these examples that eventually allows the system to come up with rules for automating the task.
- ML is tightly related to mathematical statistics, but it differs from statistics in several important ways.
  - ML tends to deal with large, complex datasets (such as a dataset of millions of images, each consisting of tens of thousands of pixels).

What is machine learning (ML)?

What is machine learning (ML)?

- **ML** is the science (and art) of programming computers so they can **learn from data**
- **ML** is the field of study that gives computers the ability to **learn without being explicitly programmed**.

  — Arthur Samuel, 1959
- 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

What is machine learning?

- ML is the process of **training** a piece of software, called a **model**, to make useful **predictions** from data. An **ML model** represents the **mathematical relationship** between the elements of data that an ML system uses to make predictions.

Why now?

- **Data**: Larger amounts of data, easy to produce, collect and store
- **Compute**: Powerful processing units, hardware acceleration
- **Algorithms**: ML frameworks, libraries, improved and more efficient techniques

Why use machine learning?

Why use machine learning?

- ML is great for:
  - Problems for which existing **solutions require long lists of rules**: one ML algorithm can often simplify code and perform better.
  - **Complex problems** for which there is no good solution (no known algorithm) using a traditional approach (e.g. speech recognition).
  - **Fluctuating environments**: ML systems can adapt to new data.
  - Getting insights about large amounts of data. ML can help discover patterns that were not immediately apparent. This is called **data mining**

Quiz

- The science of programming computers so they can learn from data is called ----------.

a) AI
b) machine learning
c) classical programming

- A mathematical relationship that an ML system uses to make predictions?

a) Data.
b) Feature.
c) Model.

Types of machine learning

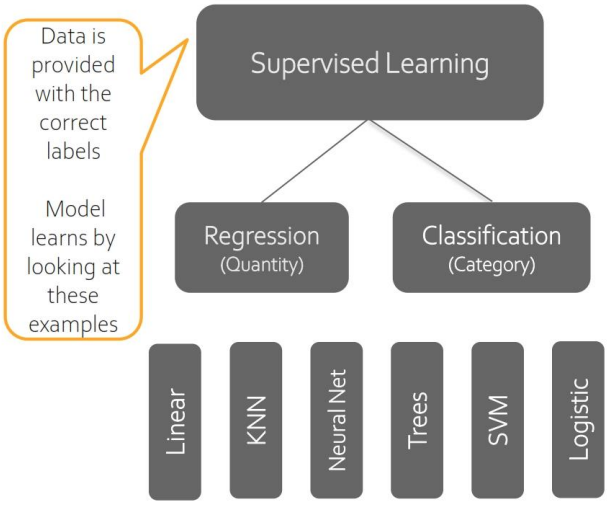
Types machine learning

- There are so many different types of ML systems that it is useful to classify them in broad categories based on:
  - Whether or not they are trained with **human supervision**.
  - Whether or not they can **learn incrementally** from a stream of incoming data (online versus batch learning).
  - Whether they work by simply **comparing new data points to known data points**, or instead detect patterns in the training data and **build a predictive model**.

These criteria are not exclusive; you can combine them in any way you like.

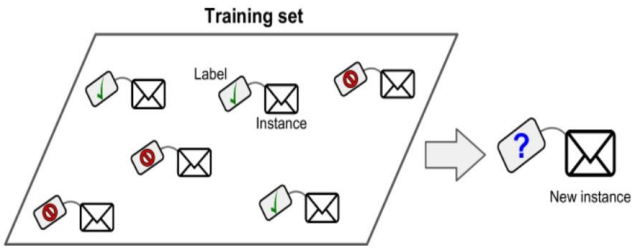
Supervised vs. unsupervised learning

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

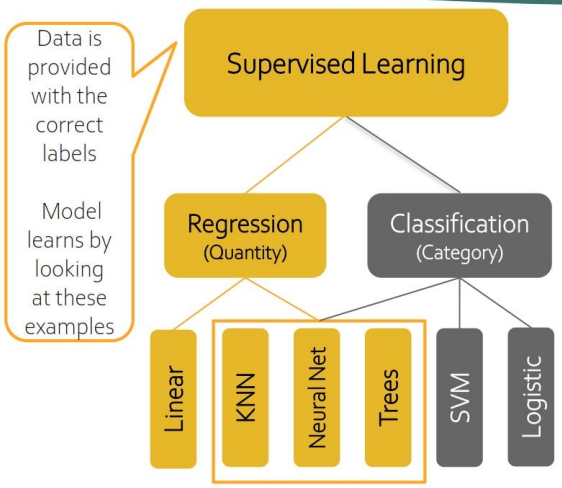


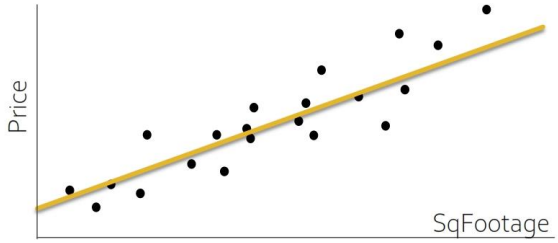
Supervised learning: Classification

- A typical supervised learning task is **classification** (discrete-valued output). The spam filter is a good example of this.



Supervised learning: Regression



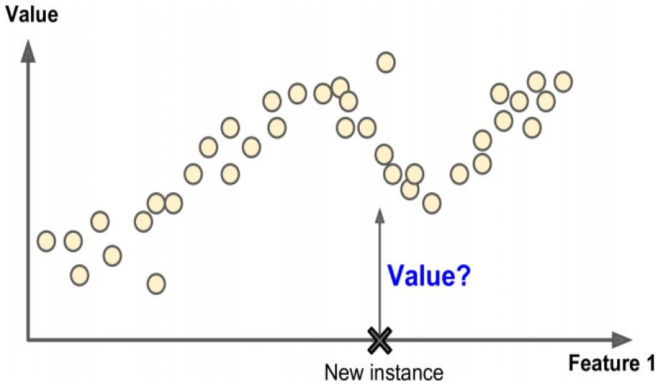


Label | Features

Price	Bedrooms	SqFootage	Age
280.000	3	3292	14
210.030	2	2465	6
...	...	...	...

Supervised learning: Regression

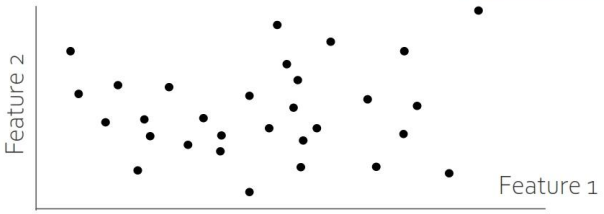
- Another typical task is to predict a target numeric value, such as the price of a car, given a set of features (mileage, age, brand, etc.) called predictors. This sort of task is called **regression** (continuous-valued output).



Supervised learning

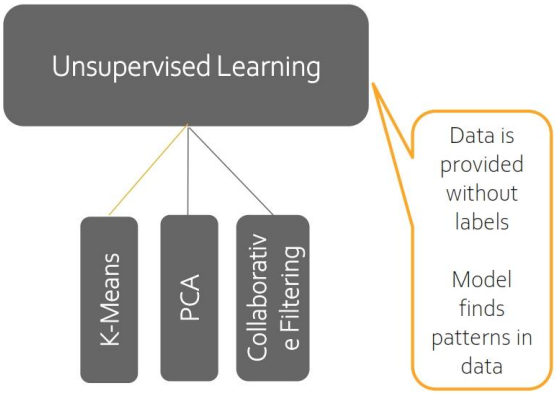
- Here are some of the most important supervised learning algorithms:
  - k-Nearest Neighbors
  - Linear Regression
  - Logistic Regression
  - Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

Unsupervised learning

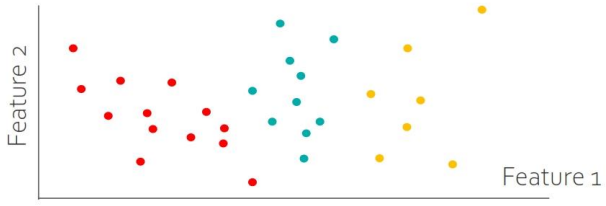


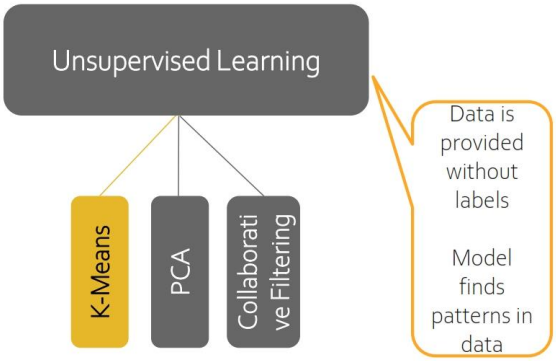
Features

Age	Music	Books
21	Classical	Practical Magic
47	Jazz	The Great Gatsby
...	...	...



Unsupervised learning: Clustering



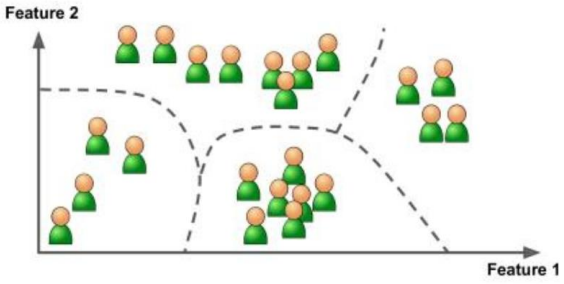


Features

Age	Music	Books
21	Classical	Practical Magic
47	Jazz	The Great Gatsby
...	...	...

Unsupervised learning

- In **unsupervised learning**, the training data is **unlabeled**. The system tries to learn without a teacher.
- Here are some of the most important unsupervised learning tasks and algorithms:
  - **Clustering** (K-Means, Hierarchical cluster analysis (HCA))

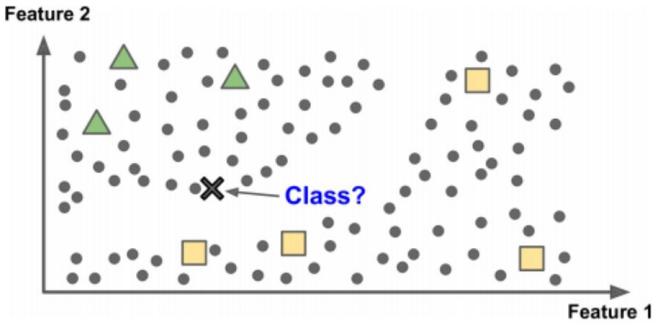


Unsupervised learning

- **Dimensionality reduction** (Principal component analysis (PCA))
- **Anomaly detection** (One-class SVM - Isolation Forest)

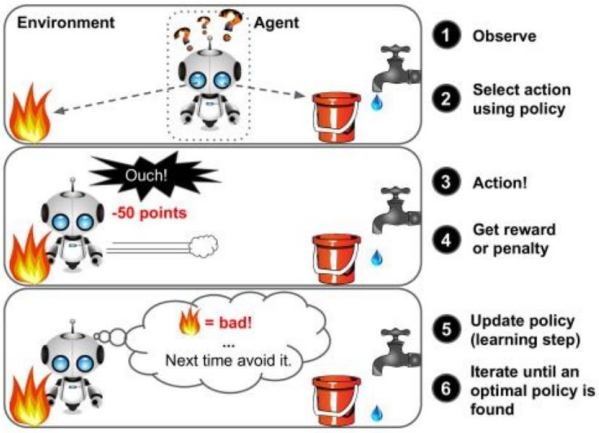
Semisupervised learning

- **Semisupervised learning** (Deep belief ntworks (DBNs)): Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data.
- Some photo-hosting services, such as Google Photos, are good examples of this.



Reinforcement learning

- **Reinforcement learning**: The learning system, called an **agent** in this context, can observe the environment, select and perform actions, and get **rewards** in return. It must then learn by itself what is the best strategy, called a **policy**, to get the most reward over time.



Batch and online learning

- Another criterion used to classify ML systems is whether or not the system can learn incrementally from a stream of incoming data.
- In **batch learning**, the system is incapable of learning incrementally: it must be trained using all the available data.
- 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**.

Batch and online learning

- In **online learning**, you train the system incrementally by feeding it data instances sequentially, either individually or by 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.
- **Online learning** algorithms can also be used to train systems on **huge datasets** that cannot fit in one machine's main memory (**out-of-core learning**).

Instance-based versus model-based learning

- One more way to categorize Machine Learning systems is by how they **generalize**.
- Having a good performance measure on the training data is good, but insufficient; the true **goal** is to **perform well on new instances**.
- Two main approaches to generalization: instance-based learning and model-based learning.

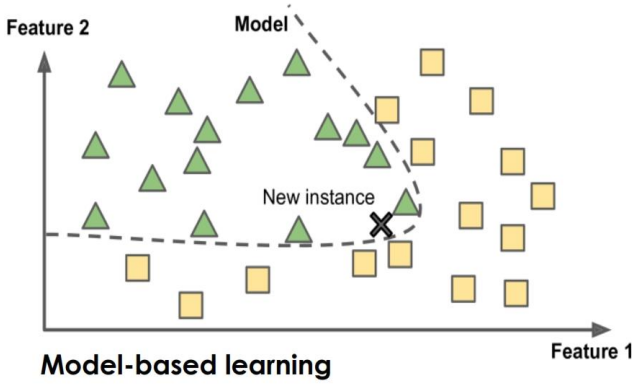
Instance-based learning

- System learns the **examples by heart**, then generalizes to new cases by comparing them to the learned examples (or a subset of them), using a **similarity measure**.

Instance-based learning

Model-based learning

- Another way to generalize from a set of examples is to **build a model** of these examples, then use that model to make predictions.



Model-based learning

Quiz

- If the training data that you feed to the ML algorithm is labeled, this is called ----------.
  a) supervised learning
  b) unsupervised learning
  c) reinforcement learning

- To train systems on huge datasets that cannot fit in machine's memory, ---------- learning algorithms can be used?
  a) online.
  b) offline.
  c) semisupervised.

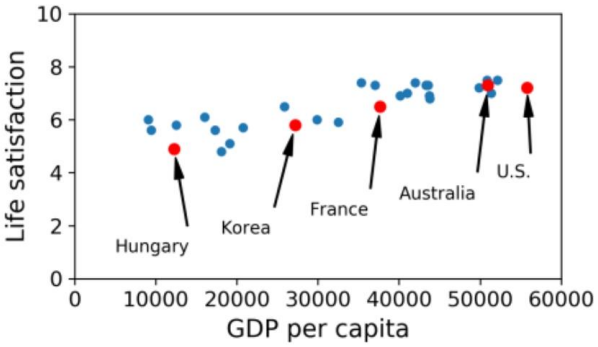
Example of a machine learning project

Example: Study the data

- To generalize from a set of examples, build a model of these examples, then use that model to make predictions (model-based learning).

Does money make people happier?

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

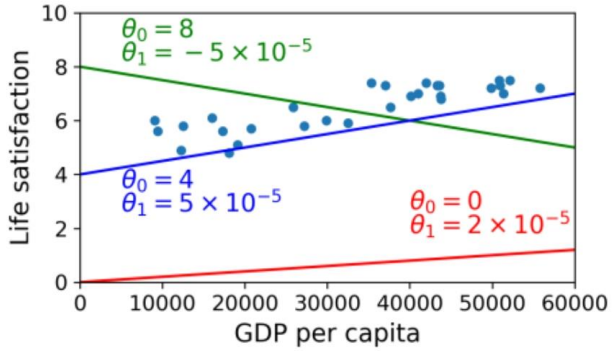


Example: Model selection

- you selected a linear model of life satisfaction with just one attribute, GDP per capita

$$life\_satisfaction = \theta_0 + \theta_1 \times GDP\_per\_capita$$

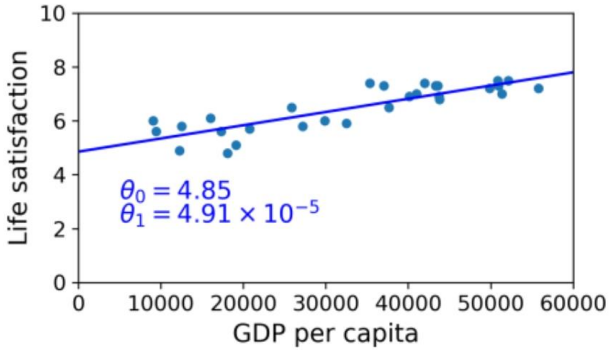
- This model has two model parameters, $\theta_0$and θ_1.



A few possible linear models

Example: Training the model

- A performance measure:
  - **Utility function** (or fitness function) that measures how good your model is.
  - **Cost function** that measures how bad it is.
- Training the model to find the parameters that make the model fit best to your data



The linear model that fits the training data best

Example: Make predictions

- You are finally ready to run the model to make predictions.
  - For example, say you want to know how happy Cypriots are, you can use your model to make a good prediction:

$$4.85 + 22587 \times 4.91 \times 10^{-5} = 5.96$$

Example of Model-based learning

- Study the data.
- Select a model.
- Train the model on the training data (i.e., the learning algorithm searched for the parameter values that minimize a cost function).
- Apply the model to make predictions on new cases (**inference**), hoping that this model will generalize well.

```
import matplotlib.pyplot as plt
import pandas as pd
import sklearn.linear_model

# Load the data
oecd_bli = pd.read_csv("oecd_bli_2015.csv", thousands=',')
gdp_per_capita = pd.read_csv("gdp_per_capita.csv",thousands=',',delimiter='\t',
                             encoding='latin1', na_values="n/a")
# Visualize the data
country_stats.plot(kind='scatter', x="GDP per capita", y='Life satisfaction')
plt.show()

# Select a linear model
model = sklearn.linear_model.LinearRegression()

# Train the model
model.fit(X, y)

# Make a prediction for Cyprus
X_new = [[22587]]  # Cyprus' GDP per capita
print(model.predict(X_new)) # outputs [[ 5.96242338]]
```

Thank you