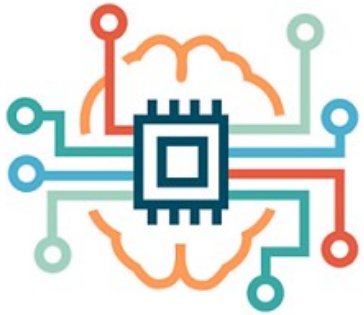
COMP1804 Applied Machine Learning

Lecture 3: ML Landscape

Dr. Dimitrios Kollias



Outline

- Types of Machine Learning System
- Dataset Preparation
- Main Challenges of Machine Learning
- Process of Creating a Machine Learning System

Types of Machine Learning System

Types of Machine Learning systems

- Supervised learning
  - Classification: binary, multi-class, multi-label
  - Regression
- Unsupervised learning
  - Clustering
  - Dimensionality Reduction
  - Association
- Semi-Supervised learning
- Reinforcement Learning

Supervised Learning (I)

In *Supervised Learning* we have:

- input variables (X)

and

- output variables (Y)

We use a ML algorithm to learn the mapping function, f, from the inputs to the outputs:

$$Y = f(X)$$

based on example input-output pairs, i.e., the *labelled/annotated training dataset*

Supervised Learning (II)

The goal is to approximate the mapping function so well that when you have new input data (X) you can predict the output variables (Y) for that data.

The majority of practical machine learning uses supervised learning.

Supervised Learning (III)

So: we have input training data/samples and we also have their labels/ground truth.

It is called supervised learning because the process of a ML algorithm learning from the labelled training dataset can be thought of as a student (i.e., the ML algorithm) learning while being supervised by the teacher (i.e., the labels of the training dataset).

The teacher provides the correct answers (labels), the algorithm iteratively makes predictions on the training data and is corrected by the teacher.

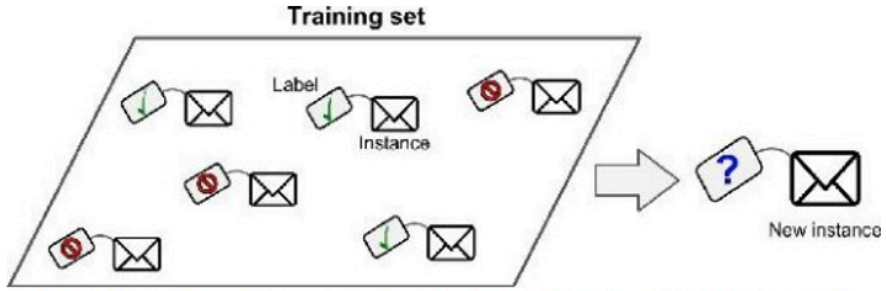
Learning stops when the algorithm achieves an acceptable level of performance.

Supervised Learning (IV)

Objectives:

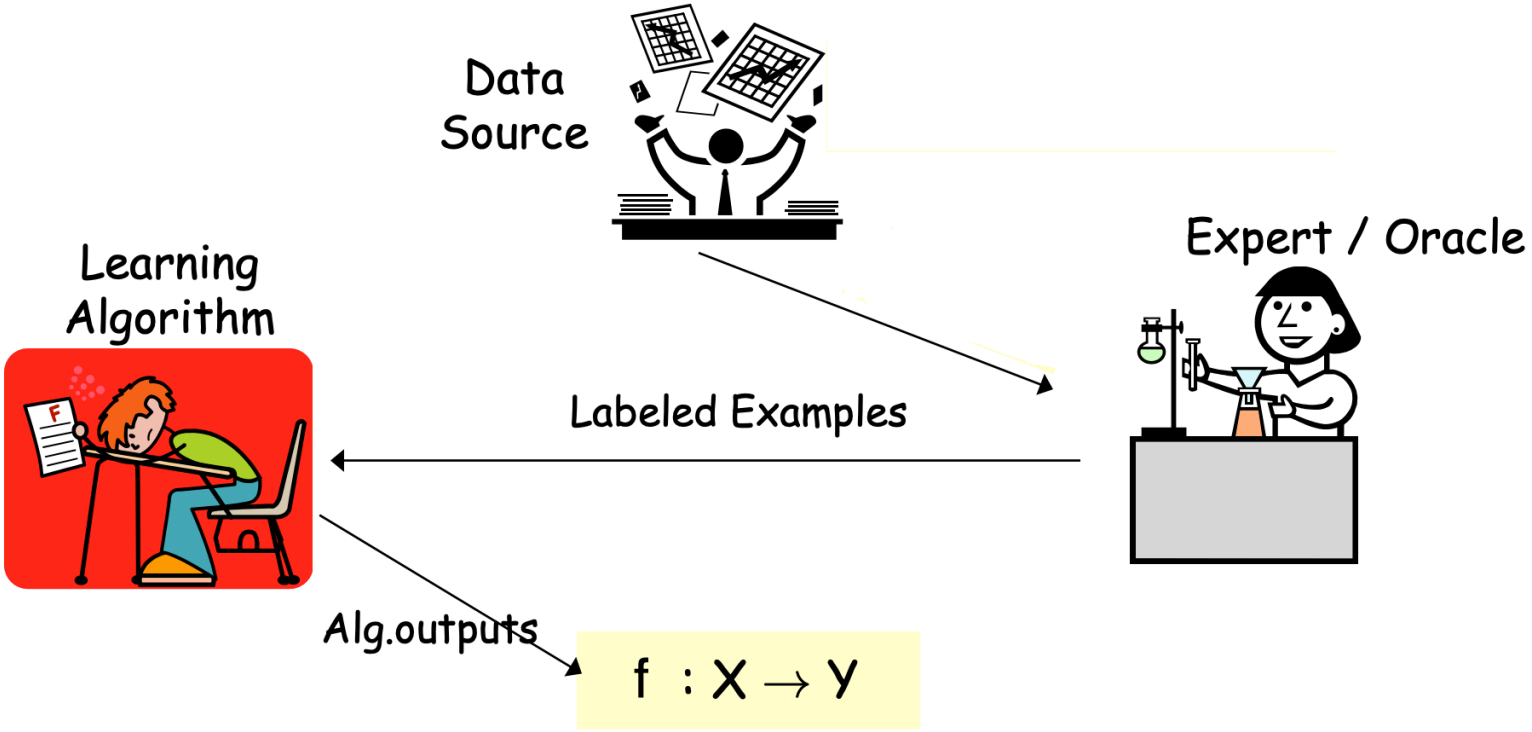
Based on the training data, the ML model aims to:

- Generalise from the training data so as to correctly predict unseen test cases in a "reasonable" way
- Understand which inputs affect the outcome, and how
- Assess the quality of the predictions



A labeled training set for supervised learning (e.g., spam classification)

Supervised Learning (V)



Supervised Learning (VI)

Supervised learning can be split into two subcategories:

- **Classification:** when the output variable is a category, such as "red" or "blue" ; "disease" or "no disease".

and

- **Regression:** when the output variable is a real value, such as "amount of pounds" or "weight".

Classification (I)

During training, a classification algorithm will be fed with input data samples and their label, i.e., an assigned category/class.

The job of the classification algorithm is to learn to assign the input samples to the class/category that they fit into.

Classification (II)

Binary classification:
A classification task with two classes to choose from;
e.g., classify emails if they are spam or not spam.

Multiclass Classification:
A classification task with more than two classes; e.g., classify a set of images of fruits which may be oranges, apples, or pears.

Multi-class classification makes the assumption that each sample is assigned to one and only one label (mutually exclusive classes): a fruit can be either an apple or a pear but not both at the same time.

Classification (III)

Multi-label classification:
In the multi-label classification there is no constraint on how many classes the training data can be assigned to (as opposed to multi-class classification).

Classification (IV)

Classification problems can be solved with a numerous amount of algorithms. Whichever algorithm you choose to use depends on the data and the situation.

Here are a few popular classification algorithms:

- ***K-Nearest Neighbor***
- ***(Artificial & Deep) Neural Networks***
- Linear Classifiers
- Support Vector Machines
- Decision Trees and Random Forests

Regression

The goal of a regression algorithm is to predict a continuous number such as sales, income, and test scores.

The three most common types of regression algorithms are:

- ***Linear Regression***
- Logistic Regression
- Polynomial Regression

Unsupervised Learning (I)

In *Unsupervised Learning* we only have input data (X) and neither corresponding output variables nor labels .

The goal of unsupervised learning is to model the underlying structure or distribution in the data in order to search for interesting/useful characteristics in the data, e.g.,

- find groups of samples that exhibit similarity in some sense
- find subset(s) of features that behave similarly
- find combinations of features with the greatest variation

Unsupervised Learning (II)

Unsupervised learning problems can be further grouped into:

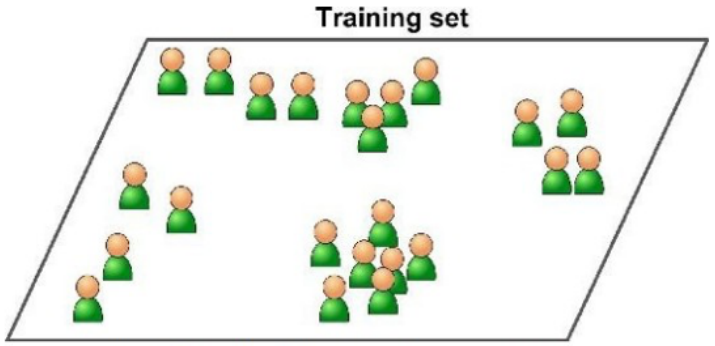
Clustering: We use clustering algorithms to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

Dimensionality reduction: We use dimensionality reduction algorithms when the number of input variables becomes very/quite large.

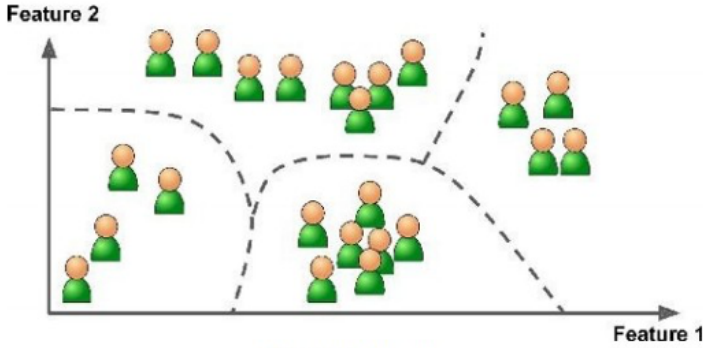
Association: We use association rule learning so as to discover rules that describe large portions of data, such as people that buy X also tend to buy Y.

Unsupervised Learning (III)

- Clustering
  - *K-means*
  - Hierarchical cluster analysis (HCA)
  - Expectation Maximisation (EM)
- Visualisation and dimensionality reduction
  - *Principle Component Analysis (PCA)*
  - Kernel PCA
  - Locally-Linear Embedded (LLE)
  - T-distributed Stochastic Neighbor Embedding (t-SNE)
- Association rule learning
  - Apriori
  - Eclat



An unlabeled training set for unsupervised learning



Clustering

Semi-Supervised Learning (I)

Problems in which we have a large amount of input data (X) and partially labelled data (i.e., only some of the data is labelled (Y)) can be tackled by semi-supervised learning algorithms.

Semi-Supervised Learning sits between supervised and unsupervised learning.

A good example is a photo archive where only some of the images are labelled, (e.g. dog, cat, person, location) and the majority are unlabelled.

Semi-Supervised Learning (II)

Many real world machine learning problems fall into this area.

This is because it can be expensive or time-consuming to label data as it may require assistance by many domain experts. In today's big data era, unlabelled data is cheap and easy to collect and store.

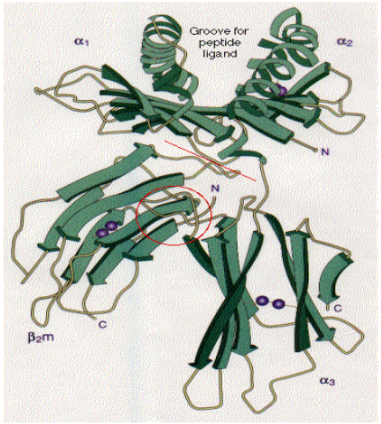
Semi-Supervised Learning (III)

We can use unsupervised learning techniques to discover and learn the structure in the input variables.

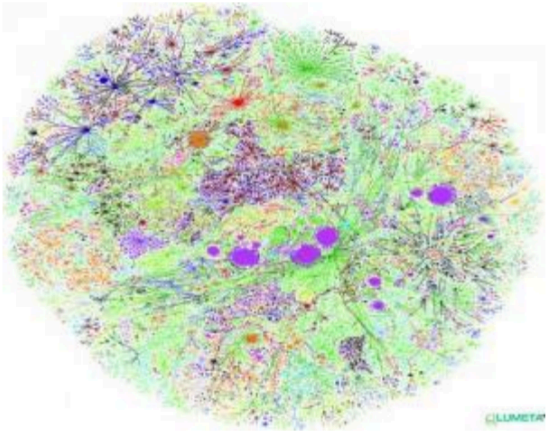
We can also use supervised learning techniques to make best predictions for the unlabelled data, feed that data back into supervised learning algorithm as training data and use the model to make predictions on new unseen data.

Semi-Supervised Learning (IV)

Modern applications: massive amounts of raw data. Only a tiny fraction can be annotated by human experts



Protein sequences



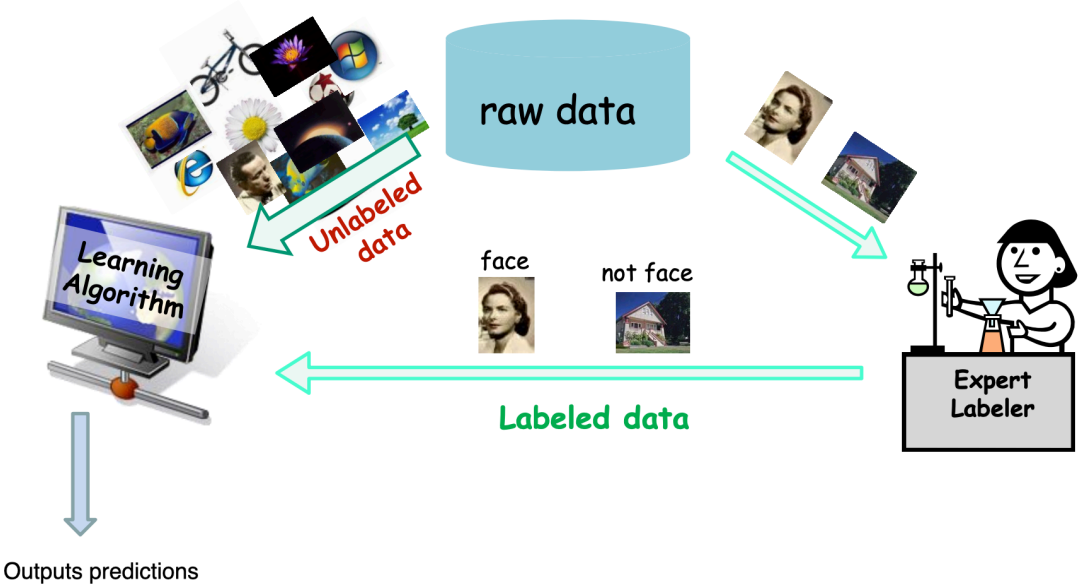
Billions of webpages

Images

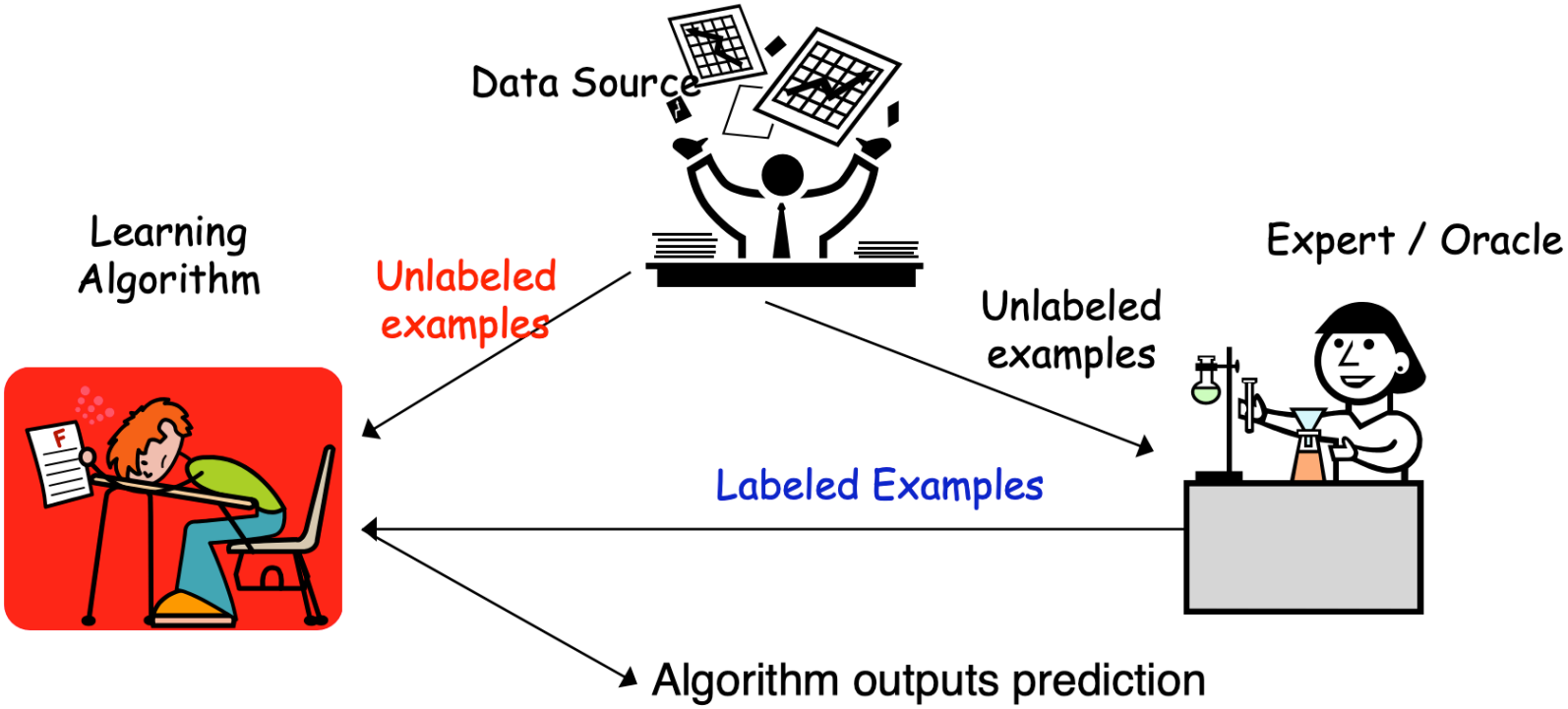
Semi-Supervised Learning (V)

Modern applications: massive amounts of raw data

Techniques that best utilize data, minimizing need for expert/human intervention

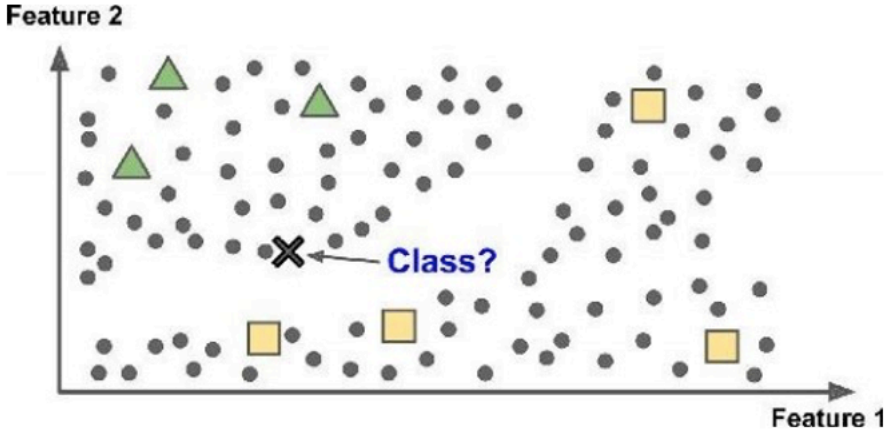


Semi-Supervised Learning (VI)



Semi-Supervised Learning (VII)

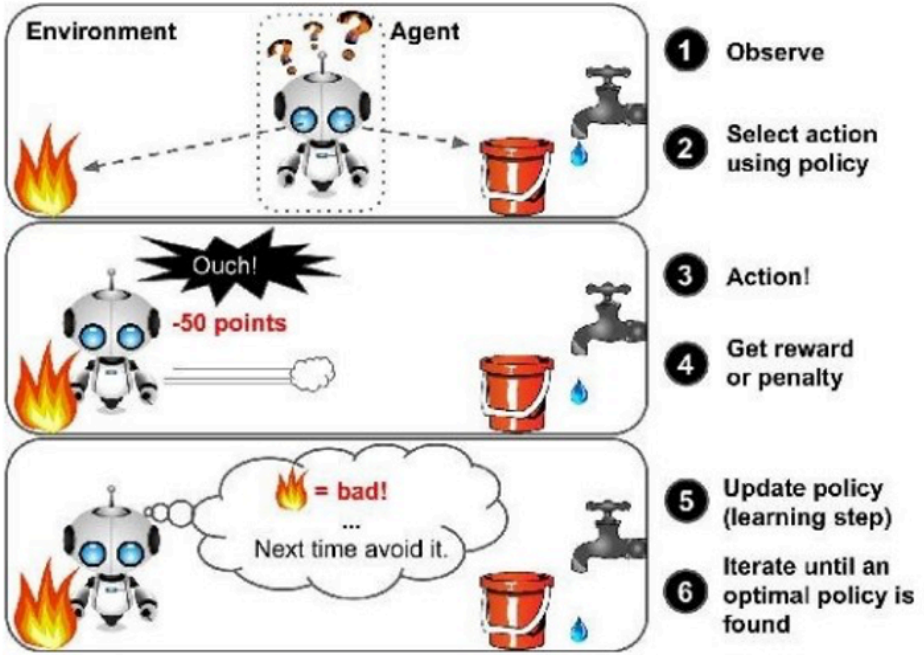
Algorithms are combinations of unsupervised and supervised ones.



Semisupervised learning

Reinforcement Learning (I)

- Learning system called agent, observes the environment, selects and perform actions, and gets rewards in return
- Learn the best strategy called a policy
- Example: Deepmind's AlphaGo vs. Lee Sedol at Go world champion.



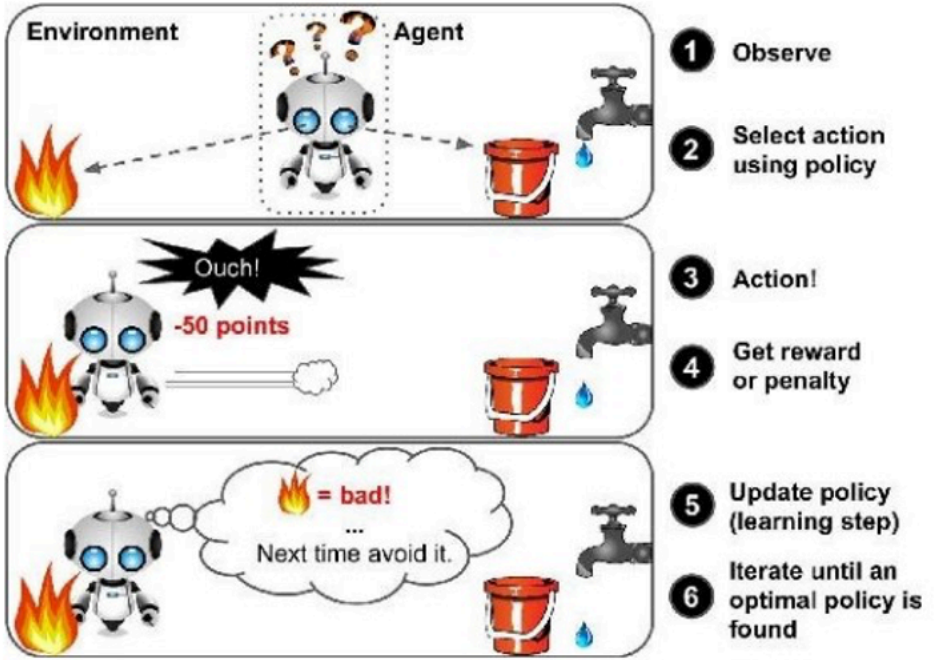
Reinforcement Learning

Reinforcement Learning (II)

Typical applications:

- Game playing
- Robot in a maze
- Credit assignment problem

etc.



Reinforcement Learning

Dataset Preparation

Training Dataset

Training Dataset:

The actual dataset that we use to train the model.

The model *sees* and *learns* from this data.

Validation Dataset

Validation Dataset:

The validation set is used to evaluate a given model.

We use this data to fine-tune the model hyperparameters.
Hence the model occasionally *sees* this data, but it never does "*Learn*" from this.

The validation set is also known as the Development set. This makes sense since this dataset helps during the "development" stage of the model.

Testing Dataset

Testing Dataset:

The sample of data used to provide an evaluation of the final model fit on the training dataset.

It is only used once a model is completely trained (using the train and validation sets).

The test set is generally well curated. It contains carefully sampled data that span the various classes that the model would face, when used in the real world.

Main Challenges of Machine Learning

Main Challenges of Machine Learning

- Problems that arise due to bad algorithm or data
  - Overfitting
  - Underfitting

Main Challenges of Machine learning

- Insufficient quantity of training data
- Non-representative training data
- Poor-quality data
- Irrelevant features
- Overfitting the training data
- Underfitting the training data

Overfitting Problem (I)

- task is to predict if an image shows a balloon or not
- train a model using a dataset containing many blue coloured balloons (and other irrelevant objects)
- test the model on the original dataset: it gives 99% accuracy!
- test the model on a new ("unseen") dataset containing yellow coloured balloons: it gives 20% accuracy!

Our model doesn't *generalise* well from our training data to unseen data. This is known as overfitting.

Overfitting Problem (II)

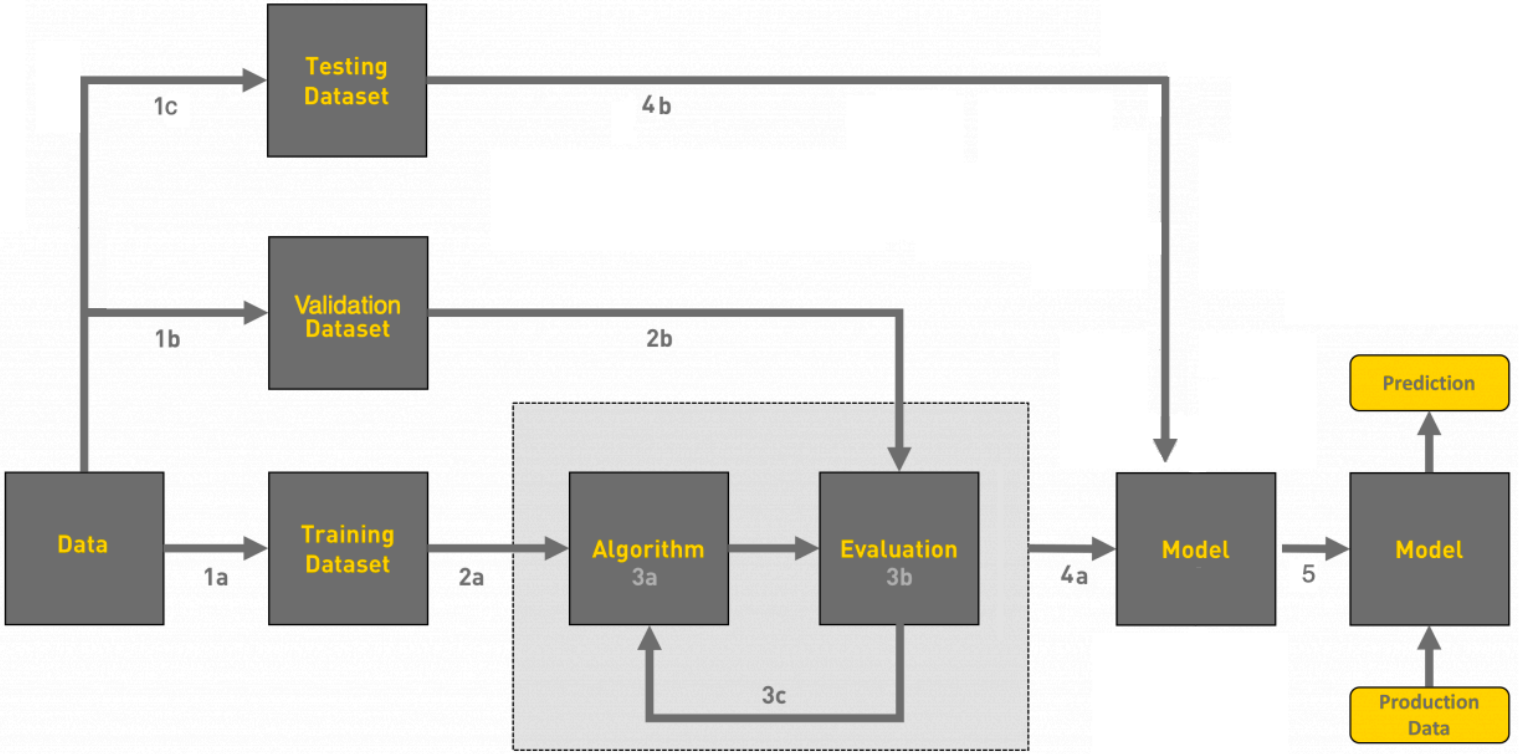
A model that has learned the noise instead of the signal is considered "overfit" because it fits the training dataset but has poor fit with new datasets.

Underfitting Problem

Underfitting happens when a machine learning model is not complex enough to accurately capture relationships between the input features and the target variable.

Process of Creating a Machine Learning System

Process of creating a ML system



Some Quick Overview

- ML is to make machines better by learning from data, instead of having to explicitly code rules
- Different types of ML systems: supervised, unsupervised, semi-supervised, reinforcement learning
- Dataset preparation: split into training, validation, testing set
- ML system will not perform well if training set is small, not representative, noisy, polluted with irrelevant features
- Model needs to be neither too simple (underfit) nor too complex (overfit)

Thank you for the attention