19ECCN1702 - Machine Learning

UNIT 1- INTRODUCTION

Unit I INTRODUCTION **9 Hours**

Introduction to Machine Learning - Types of Machine Learning systems - Challenges in Machine Learning - Overfitting and Under fitting - Testing and Validating the model - Bias and Variance

Unit II MACHINE LEARNING FRAMEWORK **9 Hours**

Problem Formulation - Get the data - analyze and visualize the data - Prepare the data for ML algorithms - sample complexity - Hypothesis space - Model evaluation and Improvement: Cross validation - Grid search - Evaluation Metrics - Kernel functions

Unit III SUPERVISED LEARNING **9 Hours**

Linear and Logistic Regression – Eigen Values and Eigen vectors - Naïve Bayes Classifier: Maximum Likelihood, Minimum Description Length – Gradient Descent - Decision Trees - Ensembles of Decision Trees - Support Vector Machine(SVM)

Unit IV UNSUPERVISED LEARNING 9 Hours

Clustering: k-Means clustering- Agglomerative Clustering - DBSCAN- Gaussian Mixtures- precision and recall - Collaborative filtering and Content Filtering

Unit V NEURAL NETWORK AND DEEP LEARNING 9 Hours

Biological Neuron - Logical computation with Neuron - Perceptron - Sigmoid and softmax functions - Multi Layer Perceptron(MLP) with Back propagation - Regression MLPs - Classification MLPs - Fine Tuning NN models - Convolutional Neural Network: Architecture of Visual cortex - Convolutional Layers - Stacking Multiple Feature Maps- CNN architectures

Course Outcomes	**Cognitive Level**
At the end of this course, students will be able to:	
CO1:Describe the types and challenges in Machine learning for exploring the machine learning concepts	Understand
CO2:Illustrate the machine learning framework for implementation of machine learning projects	Apply
CO3:Interpret the supervised learning techniques for classification	Apply
CO4:Demonstrate the un-supervised learning methods for clustering and classification	Apply
CO5:Construct the Neural network and deep learning models for classification	Apply

Text Book(s):

T1. AurélienGéron," Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow", Second edition, O'Reilly Media, Inc,2019

T2. Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python A Guide for Data Scientists", First Edition,O'Reilly,2017

Reference Book(s):

R1. Ethem Alpaydin, "Introduction to Machine Learning 3e (Adaptive Computation and Machine Learning Series)", 3rd Edition, MIT Press, 2014

R2. Jason Bell, "Machine learning - Hands on for Developers and Technical Professionals",1st Edition, Wiley, 2014

R3. Peter Flach, "Machine Learning: The Art and Science of Algorithms that Make Sense of Data", 1st Edition, Cambridge University Press, 2012.

Web References:

1. https://www.kaggle.com/kanncaa1/machine-learning-tutorial-for-beginners
2. https://nptel.ac.in/courses/106/106/106106139/
3. https://archive.ics.uci.edu/ml/datasets.php

What is artificial intelligence?

Artificial intelligence is a broad field, which refers to the use of technologies to build machines and computers that have the ability to mimic cognitive functions associated with human intelligence, such as being able to see, understand, and respond to spoken or written language, analyze data, make recommendations, and more.

What is machine learning?

Machine learning is a subset of artificial intelligence that automatically enables a machine or system to learn and improve from experience.

Instead of explicit programming, machine learning uses algorithms to analyze large amounts of data, learn from the insights, and then make informed decisions.

How are AI and ML connected?

- AI is **the broader concept** of enabling a machine or system to sense, reason, act, or adapt like a human
- ML is **an application of AI** that allows machines to extract knowledge from data and learn from it autonomously

The Fundamentals of Machine Learning

Machine Learning is the science (and art) of programming computers so they can learn from data.

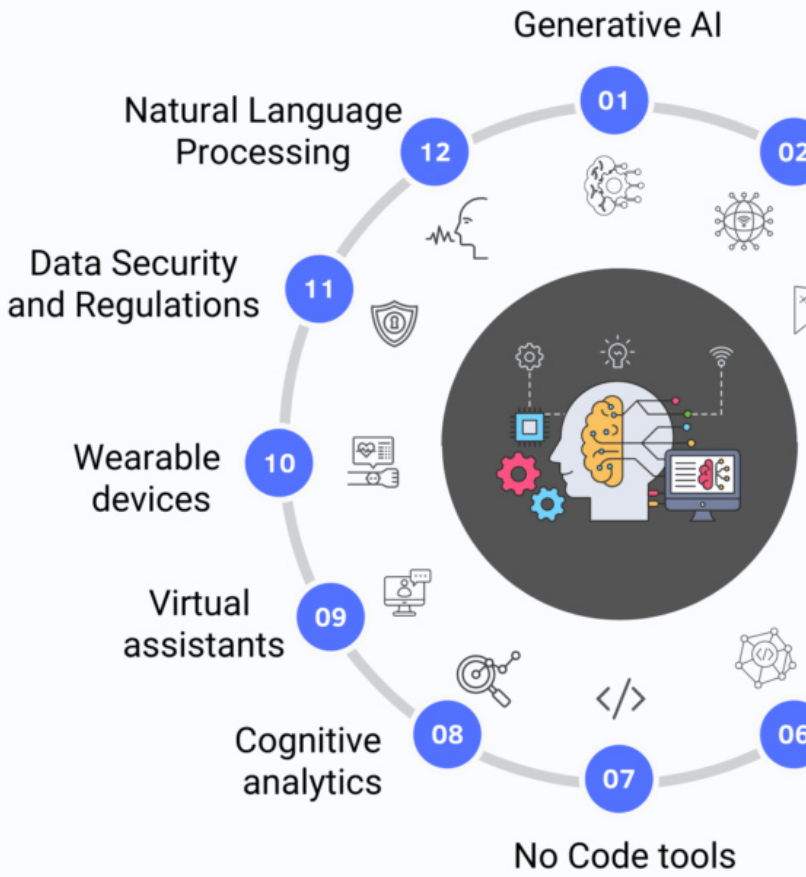
Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.

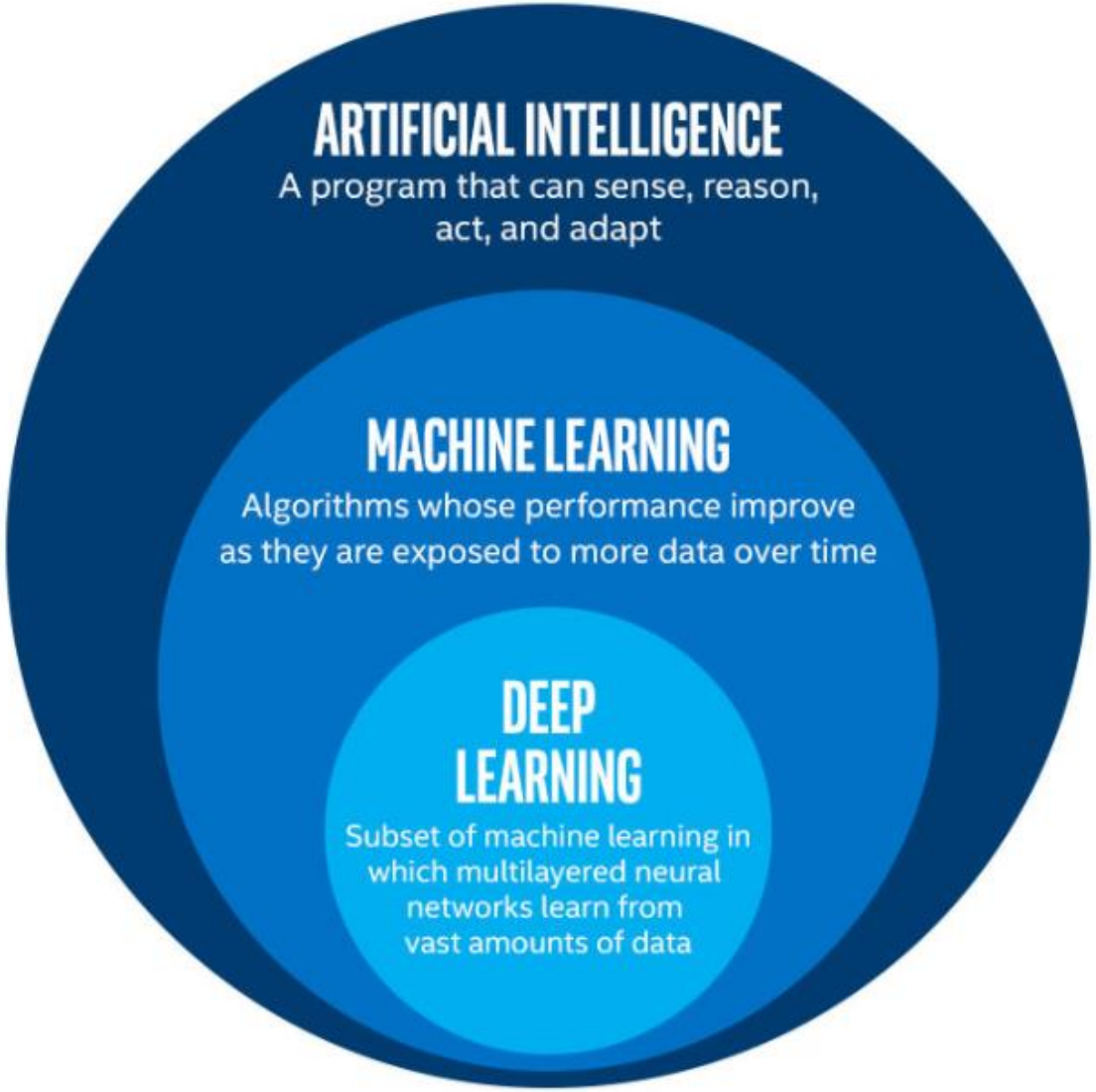
- —Arthur Samuel, *1959*

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*

2023 Emerging
AI and Machine Learning Trends



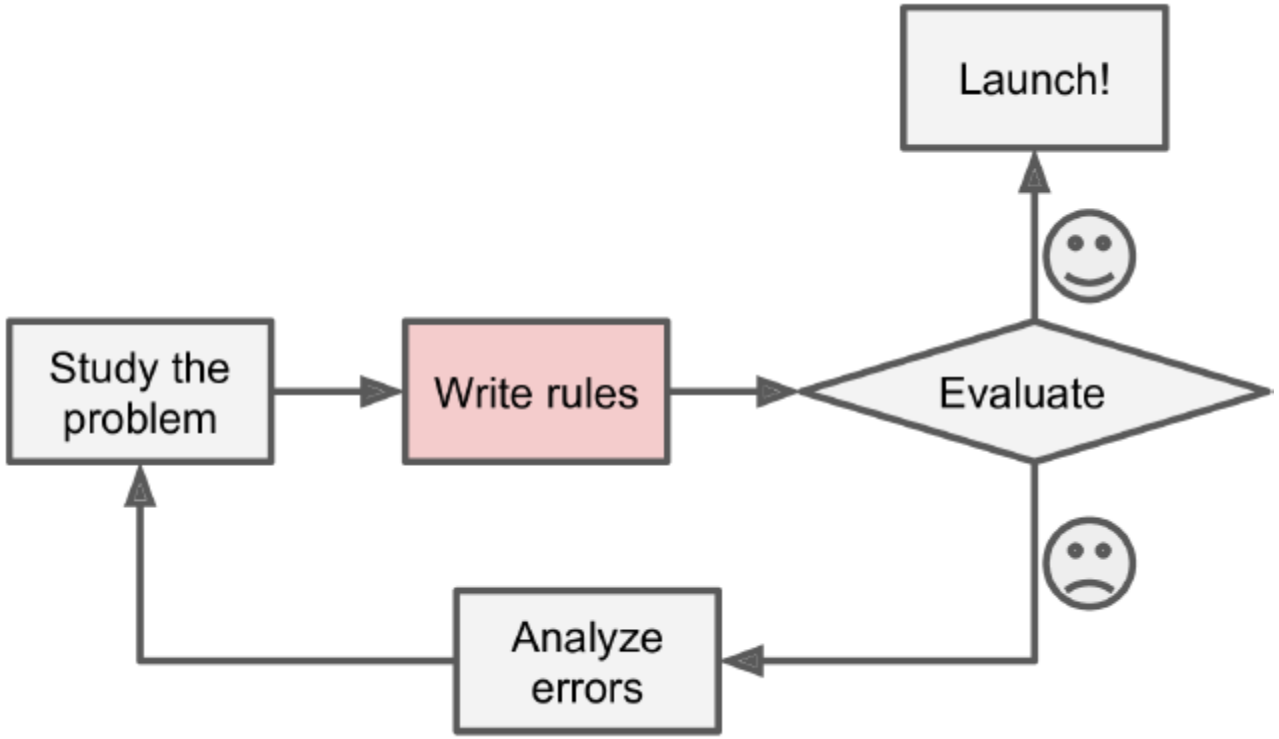

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Why Use Machine Learning?

Example

- Spam Filter Design in traditional approach
- Spam Filter Design in machine learning approach

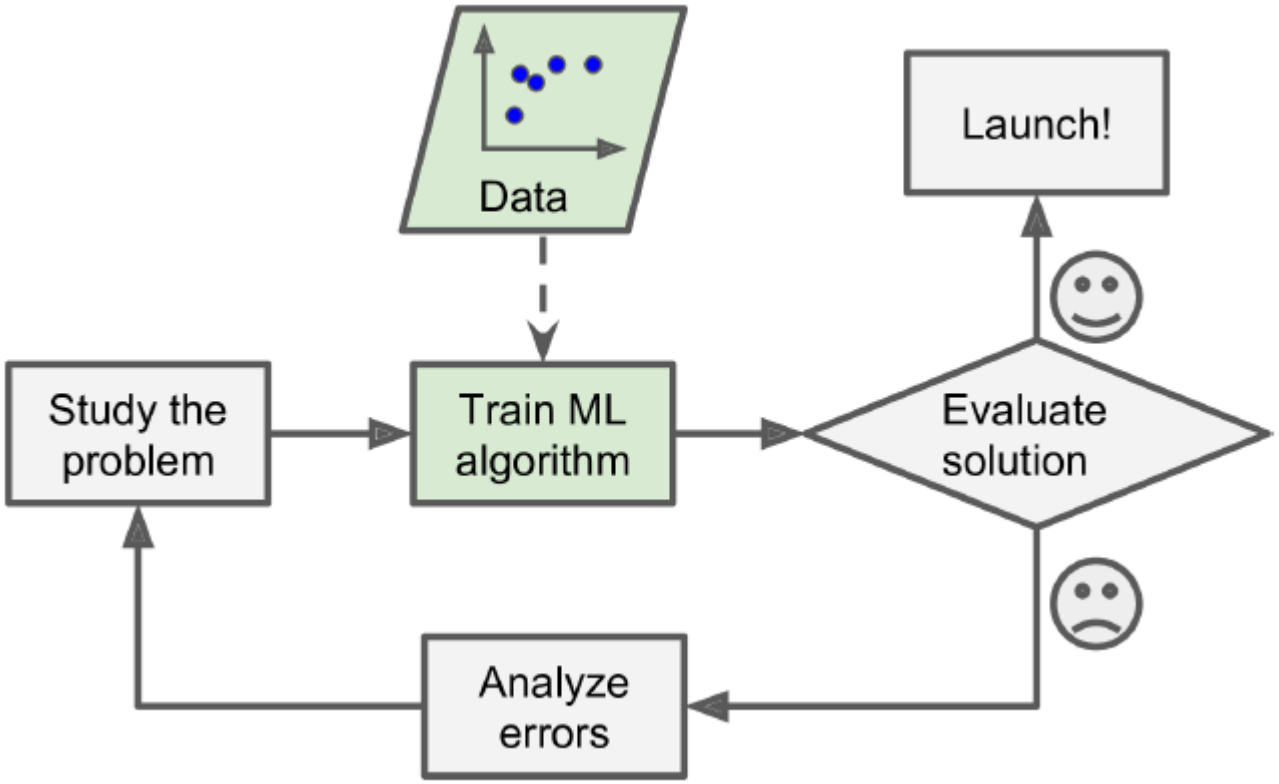
Traditional approach



Writing a detection algorithm for each of the Known spam pattern and once these patterns are detected the mail will be flagged as spam

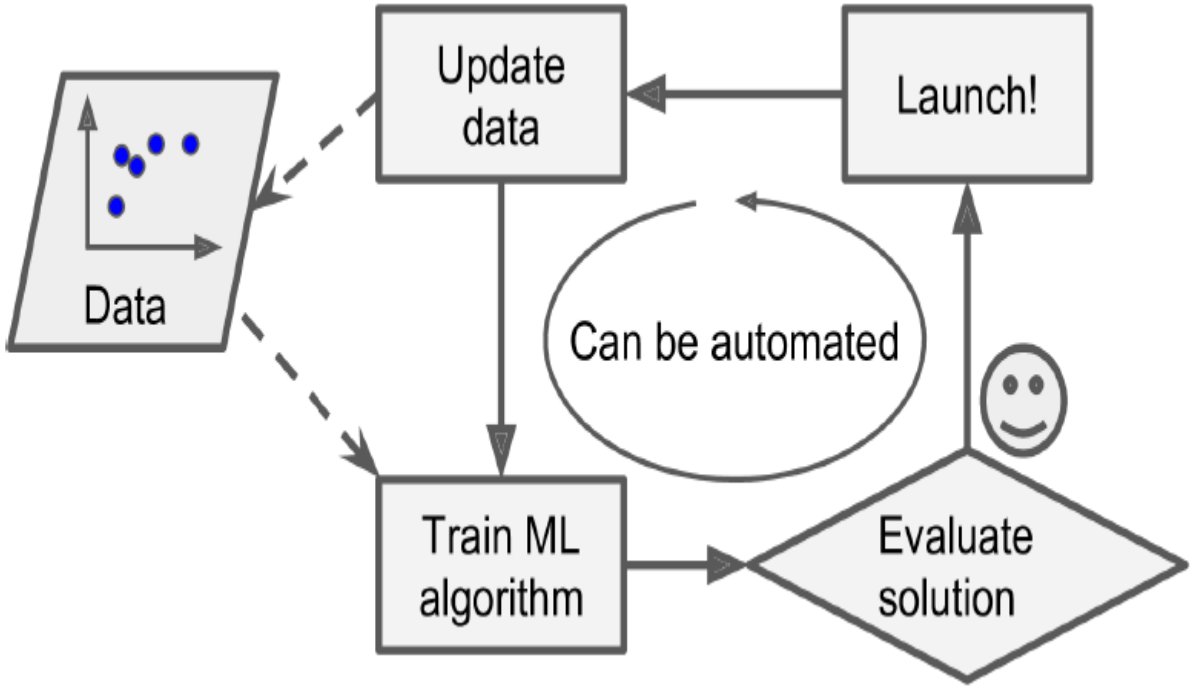
```
#  4U
# FOR U     -- ????
```

Machine Learning approach



- In contrast, a spam filter based on Machine Learning techniques automatically learns which words and phrases are good predictors of spam by detecting unusually frequent patterns of words in the spam examples compared to the ham examples.
- The program is much shorter, easier to maintain, and most likely more accurate.

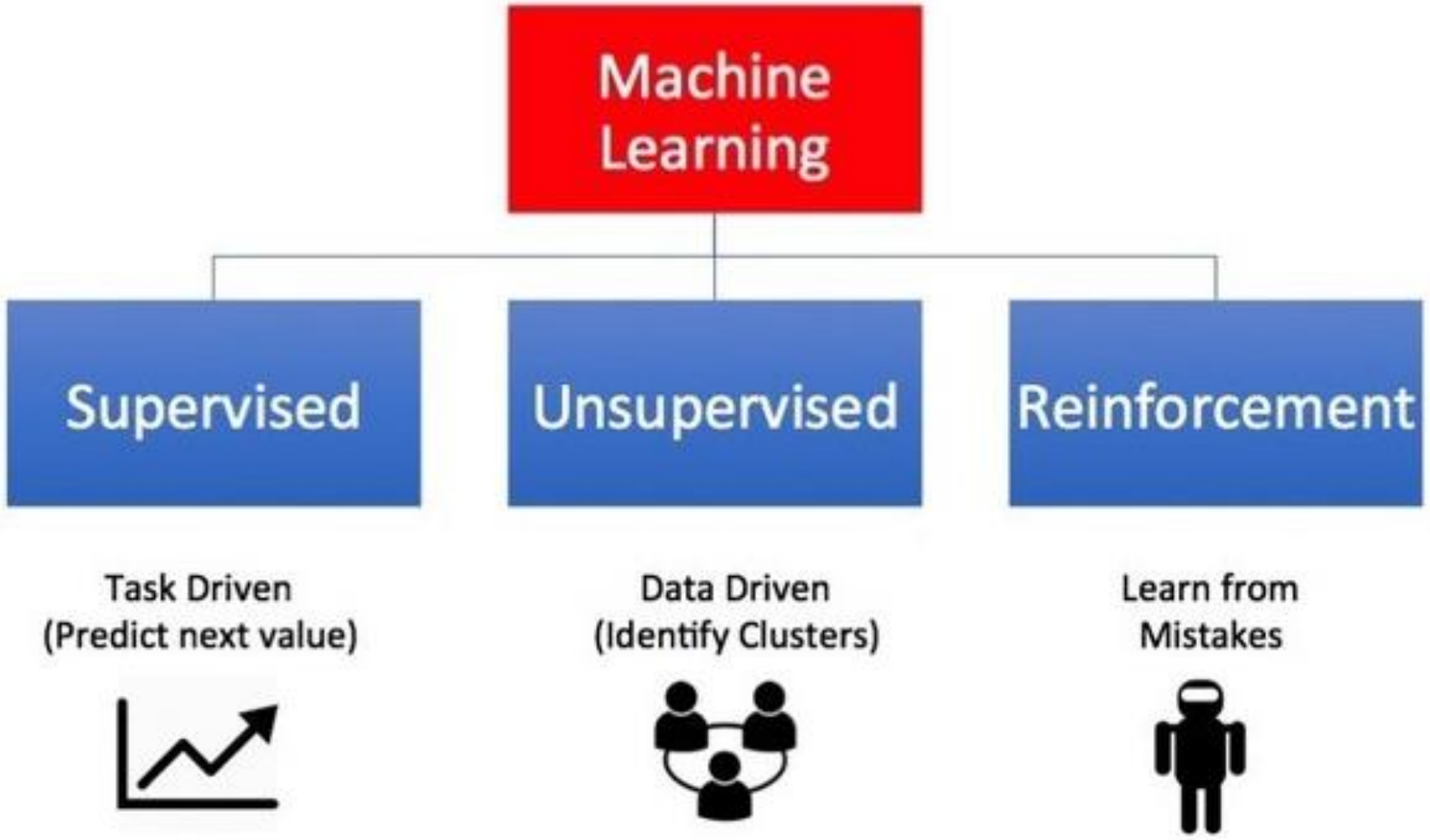
Machine Learning approach



spam filter based on Machine Learning techniques automatically notices that “For U” has become unusually frequent in spam flagged by users, and it starts flagging them without your intervention

To summarize, Machine Learning is great for:

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



Source: Google Images – Machine Learning Types

Types of Machine Learning Systems

Supervised, Unsupervised, semi supervised, and Reinforcement Learning

- Whether or not they are trained with human supervision

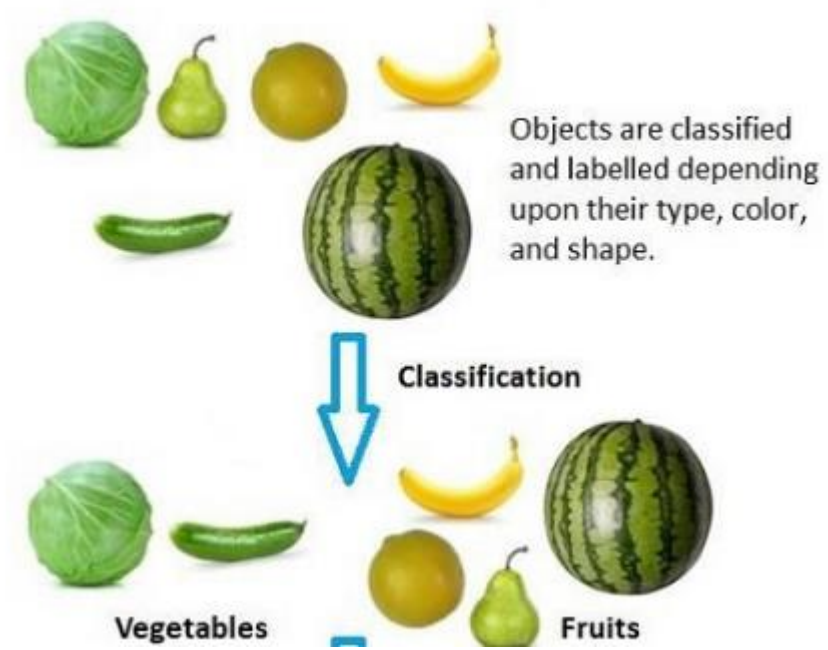
Online versus batch learning

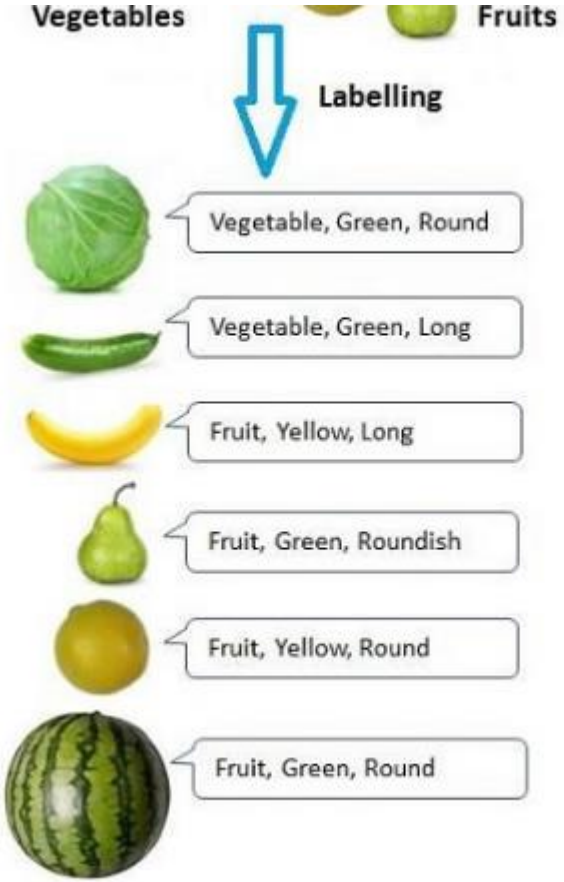
- Whether or not they can learn incrementally on the fly

Instance-based versus model-based 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, much like scientists do

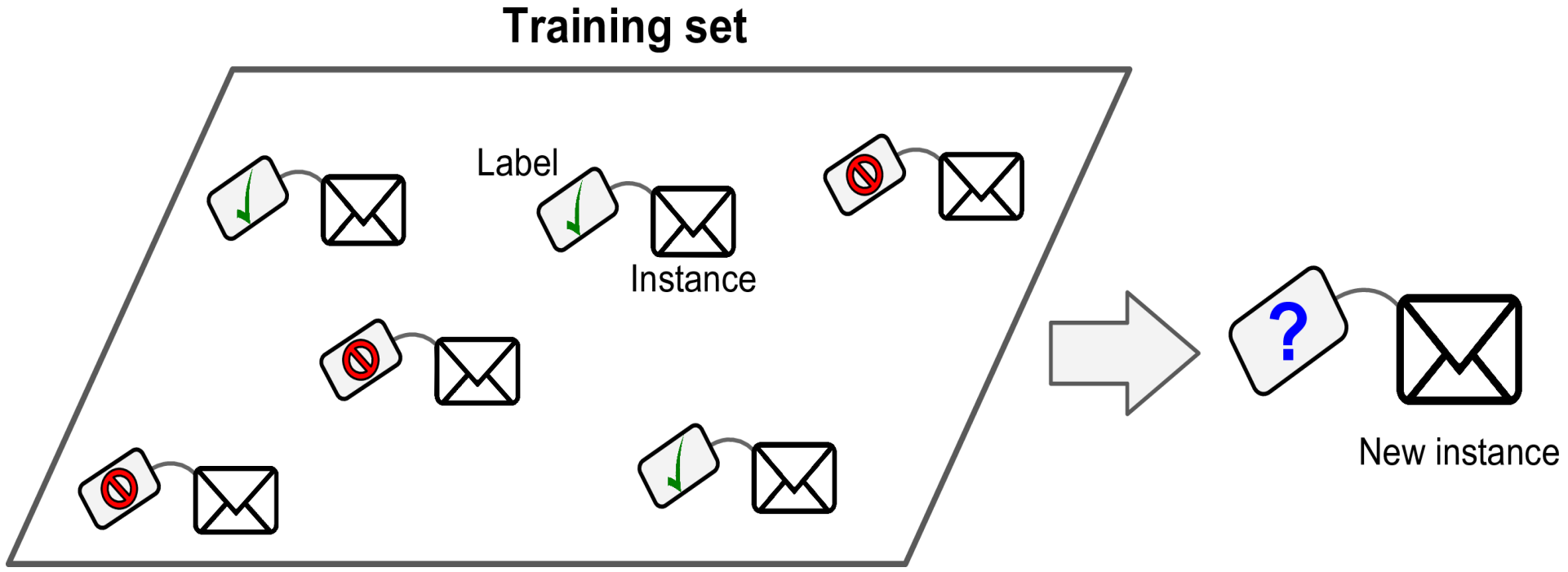
Supervised Learning





Supervised learning

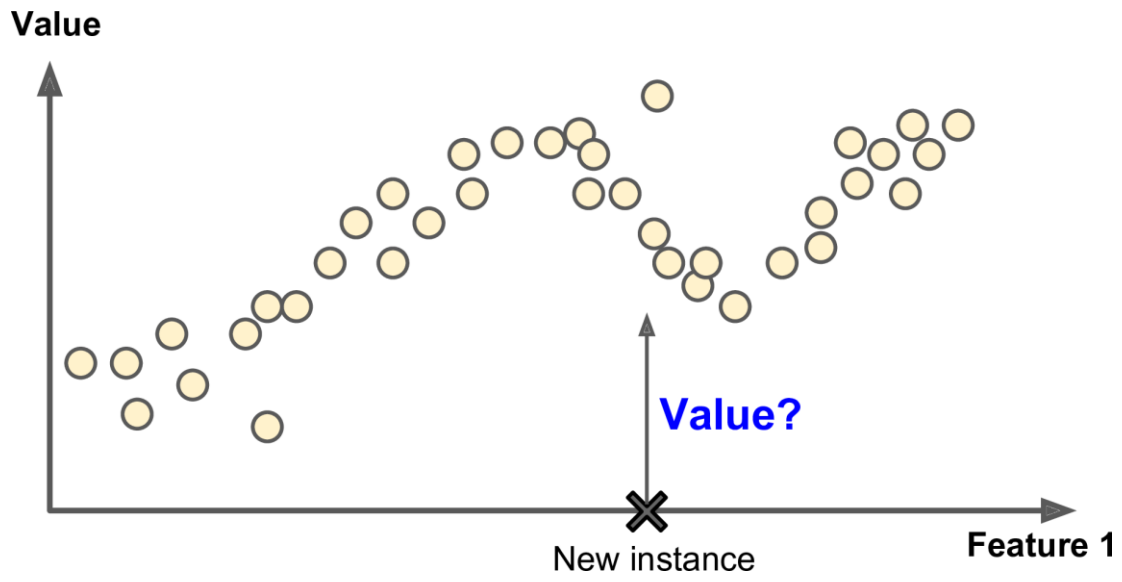
In *supervised learning*, the training data you feed to the algorithm includes the desired solutions, called ***labels***



e.g., spam classification

Tasks in Supervised learning

- **Classification**
- Prediction of target numerical value with given set of features – called **Regression**



some regression algorithms can be used for classification as well, and vice versa. For example, *Logistic Regression*

Figure 1-6. Regression

Supervised learning algorithms

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

Unsupervised learning

- the training data is unlabeled, The system tries to learn without a teacher.

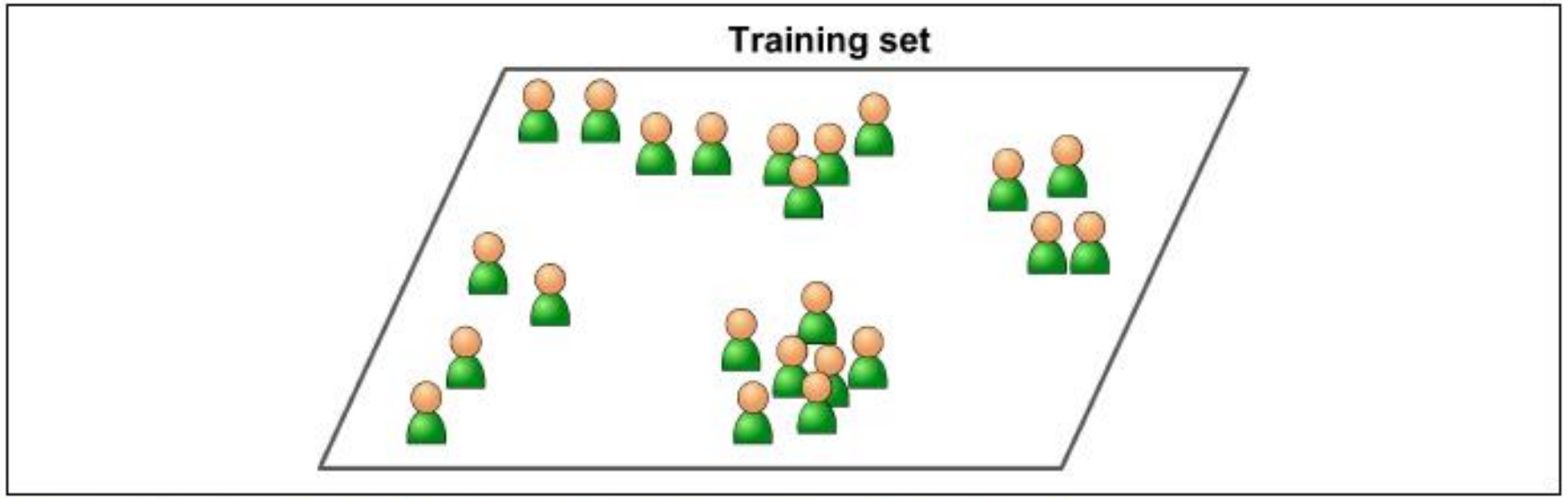


Figure 1-7. An unlabeled training set for unsupervised learning

unsupervised learning algorithms

- Clustering
  - K-Means
  - DBSCAN
  - Hierarchical Cluster Analysis (HCA)
- Anomaly detection and novelty detection
  - One-class SVM
  - Isolation Forest

unsupervised learning algorithms

- Visualization and dimensionality reduction
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally-Linear Embedding (LLE)
  - t-distributed Stochastic Neighbor Embedding (t-SNE)
- Association rule learning
  - Apriori
  - Eclat

Unsupervised learning

- Clustering – try to detect a group of similar inputs
- *Visualization* -preserve as much structure as algorithm can
- *dimensionality reduction -simplify the data - feature extraction*-increase running speed
- *anomaly detection*- automatically removing outliers from a dataset before feeding

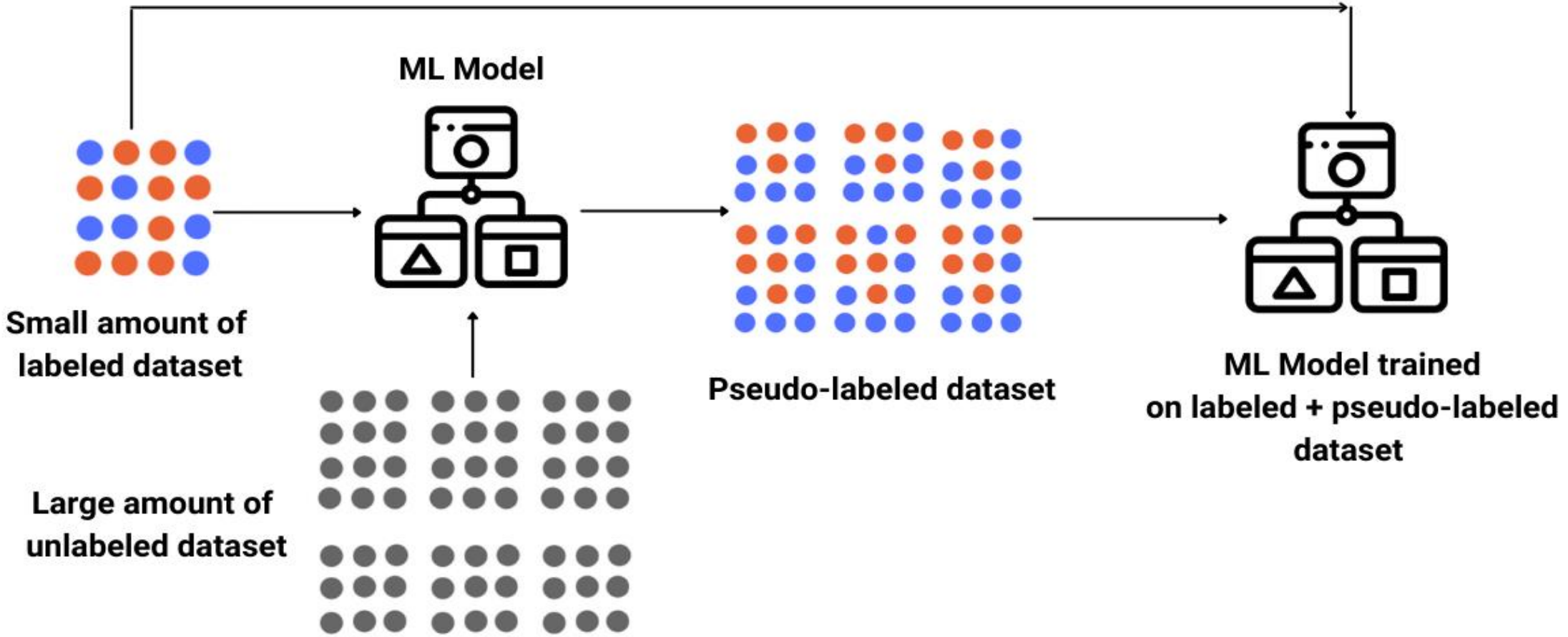
Semisupervised learning

- Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data. This is called *semisupervised learning.*

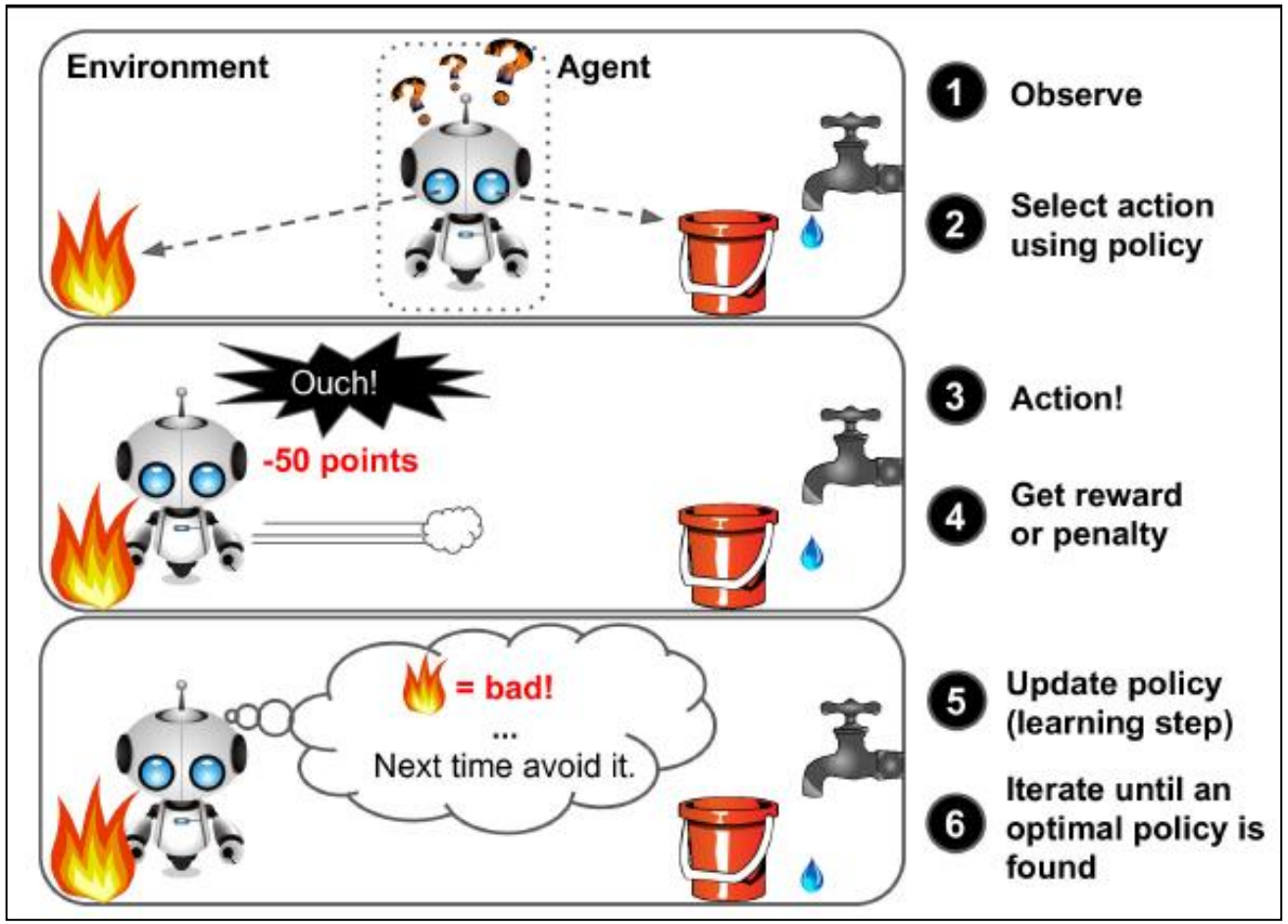
  Eg: Google photos

- The basic procedure involved is that first, the programmer will cluster similar data using an unsupervised learning algorithm and then use the existing labeled data to label the rest of the unlabelled data.

Semi-supervised learning use-case



Reinforcement Learning



- Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself
- whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task.
- In the absence of a training dataset, it is bound to learn from its experience.

Example

Carname	Color	Age	Speed	AutoPass
BMW	red	5	99	Y
Volvo	black	7	86	Y
VW	gray	8	87	N
VW	white	7	88	Y
Ford	white	2	111	Y
VW	white	17	86	Y
Tesla	red	2	103	Y
BMW	black	9	87	Y
Volvo	gray	4	94	N
Ford	white	11	78	N
Toyota	gray	12	77	N
VW	white	9	85	N
Toyota	blue	6	86	Y

[99,86,87,88,111,86,103,87,94,78,77,85,86]

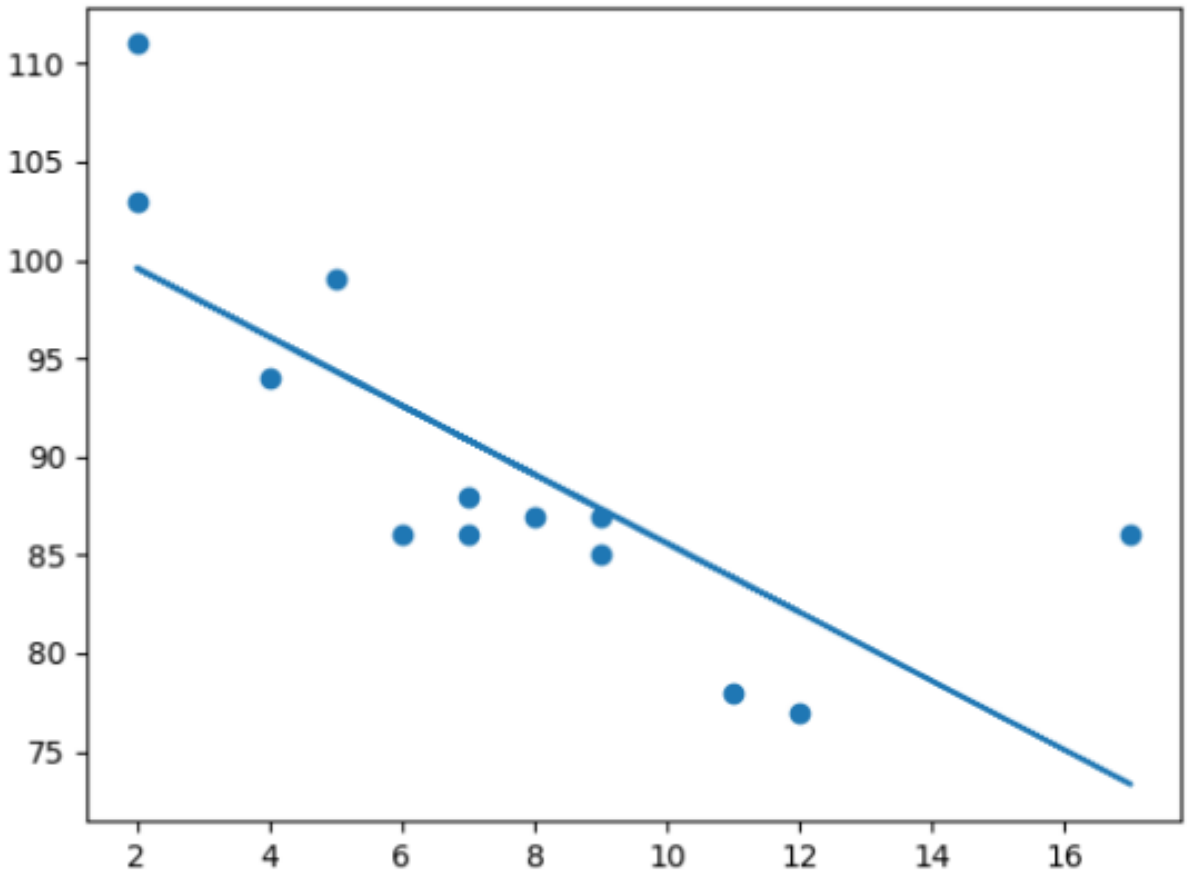
Mean - The average value
Median - The mid point value
Mode - The most common value

```
import numpy

speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]

x = numpy.mean(speed)

print(x)
```

mean

Example

```
#Three lines to make our compiler able to draw:
import sys
import matplotlib
matplotlib.use('Agg')

import matplotlib.pyplot as plt
from scipy import stats

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

slope, intercept, r, p, std_err = stats.linregress(x, y)

def myfunc(x):
  return slope * x + intercept

mymodel = list(map(myfunc, x))

plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()

#Two  lines to make our compiler able to draw:
plt.savefig(sys.stdout.buffer)
sys.stdout.flush()
```



Example

```
from scipy import stats

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

slope, intercept, r, p, std_err = stats.linregress(x, y)

def myfunc(x):
  return slope * x + intercept

speed = myfunc(10)

print(speed)
```

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.

- 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.

- In online learning, you train the system incrementally by feeding it data instances sequentially, either individually or by small groups called mini-batches.

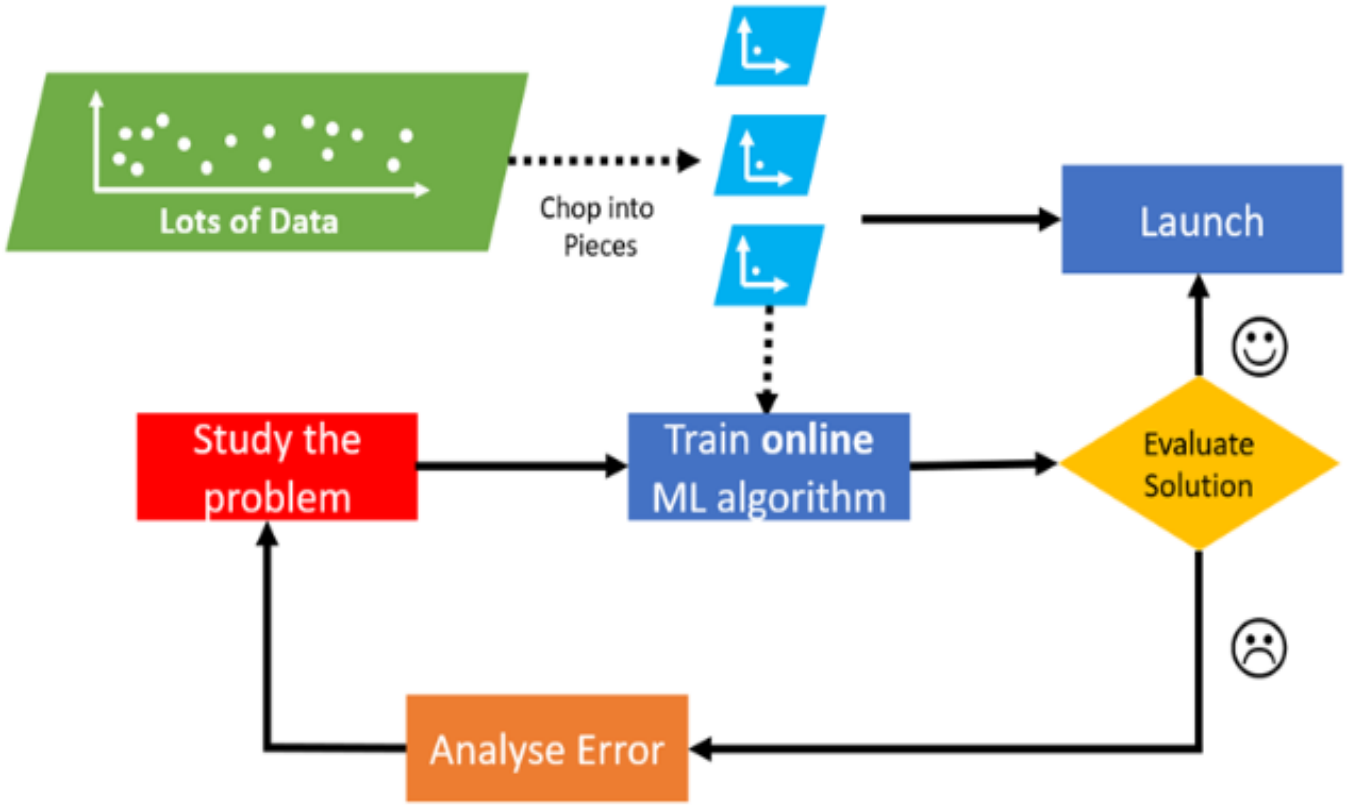
Why online learning vs batch or offline learning?

- Volume: The data comes in large volumes. This would thus require IT infrastructures, software systems, and appropriate expertise and experience to do the data processing.

- Velocity: As like in the case of the high volume of data, the data coming at high speed (for example, tweets) can also become key criteria

- Variety: Similar to volume and variety, the data can become of a different variety. For example, data for aggregator services such as Uber

Online Learning



Online learning algorithms can also be used to train systems on huge datasets that cannot fit in one machine's main memory (this is called out-of-core learning).

A big challenge with online learning is that if bad data is fed to the system, the system performances will gradually decline.

To reduce this risk, you need to monitor the systems closely and promptly switch learning off and possibly you want to revert to a previous working state if you detect a drop-in performance.

Instance-Based Versus Model-Based Learning

There are two main approaches to generalization.

This is called instance-based learning: the 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.

*A (very basic) **similarity measure** between two emails could be to count the number of words they have in common. The system would flag an email as spam if it has many words in common with a known spam email.*

Model-based learning

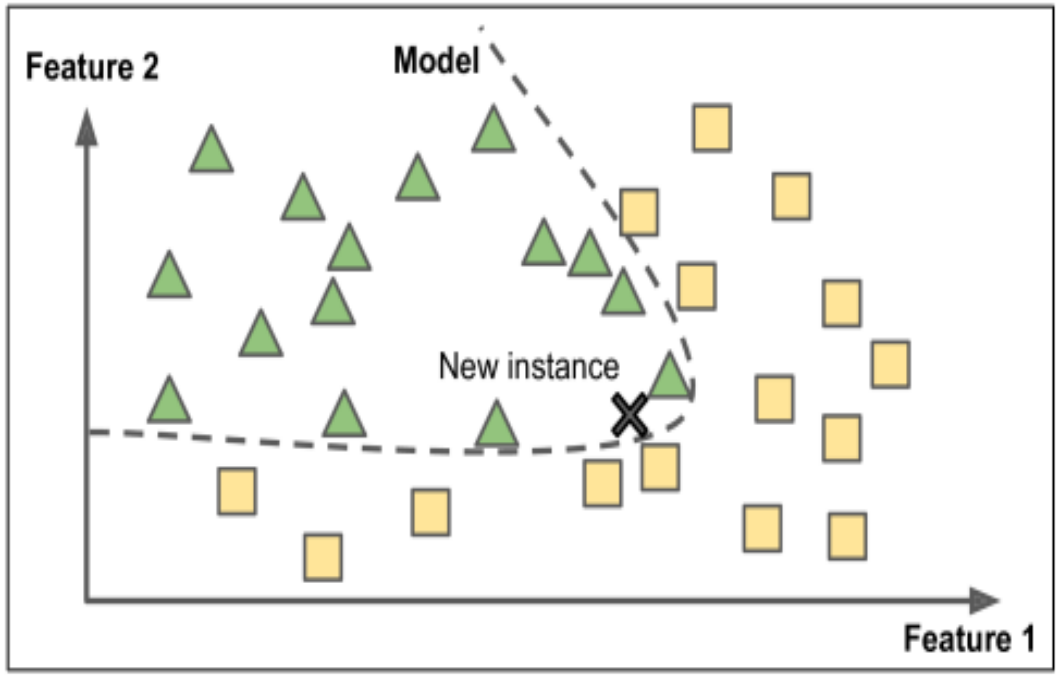


Figure 1-16. Model-based learning

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

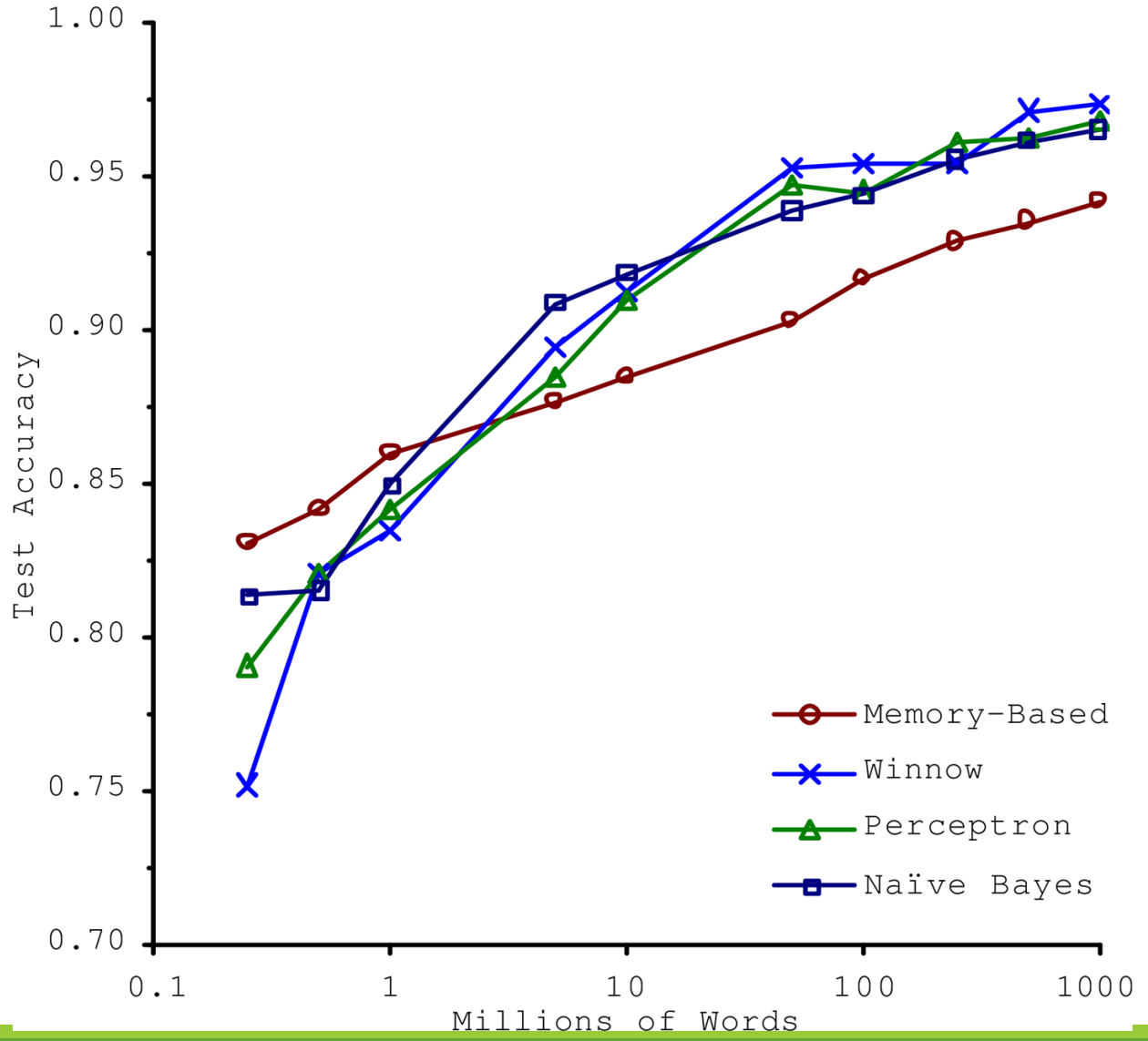
Main Challenges of Machine Learning

- Insufficient Quantity of Training Data
- Non representative Training Data
- Poor-Quality Data
- Irrelevant Features
- Over fitting the Training Data
- Under fitting the Training Data

Insufficient Quantity of Training Data

- Machine Learning is not quite there yet; it takes a lot of data for most Machine Learning algorithms to work properly.
- Even for very simple problems you typically need thousands of examples, and for complex problems such as image or speech recognition you may need millions of examples

The importance of data versus algorithms



Non representative Training Data

If the sample is too small, you will have sampling noise, which is the non representative data as a result of chance, but even large samples can be non representative if the sampling method is flawed. This is called S**ampling Bias.**

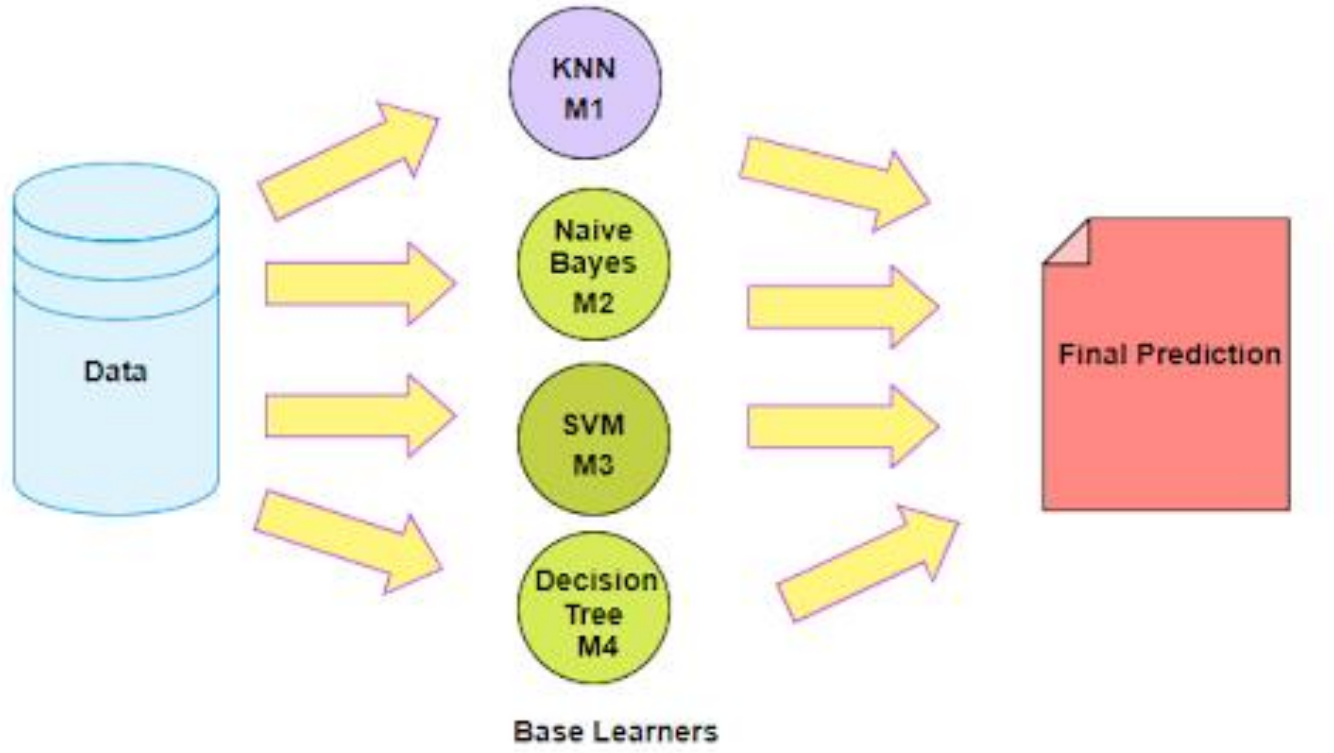
Non representative Training Data

The samples might tend to favor a select portion of the population, and thus might not accurately represent the true population. This is also popularly known as the presence of **Skewness** in data, and the data can be either right-skewed or left-skewed.

Second, there can also be a case (especially in surveys), where enough people might not answer the questions, and do not get represented in the data. This is also called as the presence of **nonresponse bias** in data

Non representative Training Data

Model Complexity - Ensemble learning



It is defined as several learners are combined to obtain a better performance than any individual learners.

It is often used to improve classification and prediction

Transfer Learning

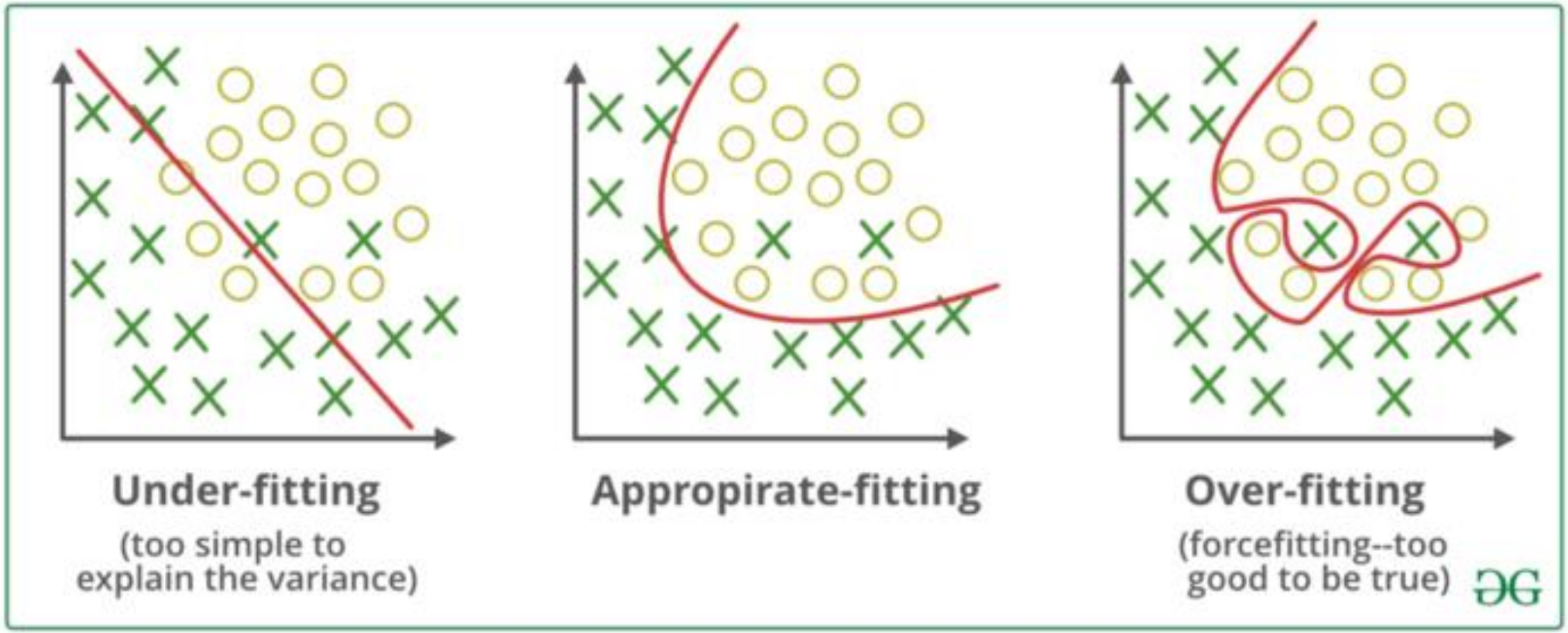
Transfer Learning is used in the case of **Deep Learning and Neural Networks**. It uses a pre-built model, which is then tweaked on the small dataset that you have.

Irrelevant Features

Feature Selection: Selecting the most useful features to train on among existing features. This can be done by using methods such as Lasso Regression.

Feature Extraction: combining existing features to create a new, more useful feature that can have a higher importance in model.

Creating new features by **gathering new data**


Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)

Bias: Assumptions made by a model to make a function easier to learn. It is actually the error rate of the training data.

Variance: The difference between the error rate of training data and testing data is called variance. Usually, we want to make a low variance for generalized our model.

Underfitting

A statistical model or a machine learning algorithm is said to have under fitting **when it cannot capture the underlying trend of the data**, i.e., it only performs well on training data but performs poorly on testing data.

It usually happens when we have **fewer data to build an accurate model** and also when we try to build a linear model with fewer non-linear data.

Reasons for Underfitting

- High bias and low variance
- The size of the training dataset used is not enough.
- The model is too simple.
- Training data is not cleaned and also contains noise in it.

Techniques to reduce Underfitting

- Increase model complexity
- Increase the number of features, performing feature engineering
- Remove noise from the data.
- Increase the number of epochs or increase the duration of training to get better results.

Overfitting

A statistical model is said to be over fitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set

Then the model does not categorize the data correctly, because of too many details and noise.

Techniques to reduce overfitting

- Increase training data.
- Reduce model complexity.
- Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
- Ridge Regularization and Lasso Regularization
- Use dropout for neural networks to tackle overfitting