भारतीय सूचना प्रौद्योगिकी संस्थान नागपुर

Indian Institute of Information Technology Nagpur

An Institution of National Importance By An Act of Parliament

Introduction to Machine Learning

For Semester VI, B. Tech. Computer Science & Engineering

By – Bharat S. Makhija

Course Material

- **Class notes, ppts.**
- **Text Books**
  1. Machine Learning, Tom Mitchell, McGraw Hill, 1997.
  2. Ethem Alpaydin, Introduction to Machine Learning, PHI, 2016
- **Reference Books**
  1. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2008.
  2. Christopher Bishop. Pattern Recognition and Machine Learning. 2e, 2006.
  3. Richard O. Duda, Peter E. Hart, David G. Stork. Pattern classification, Wiley, New York, 2001.

What is Learning?

Learning

The ability to improve behavior based on the experience.

Identify fruits in the image

What is Learning?

Learning

The ability to improve behavior based on the experience.

Identify fruits in the image

Apple Guava Custard Apple Strawberry Pineapple

What is Learning?

Identify fruits in the image

What is Learning?

Remember the names of the fruit

Ackee

Mangosteen

Rambutan

Horned Melon

Finger Lime

What is Learning?

Identify Fruits in the image

What is Learning?

How many Fruit names predicted correctly

	Student 1	Student 2	Student 3
Correct	2	3	4
Incorrect	3	2	1
Performance	40%	60%	80%

What is Machine Learning?

What we are expecting from Machine

Hey Machine!
Can you Learn
yourself?

Sure
Human!
Feed me
data

Machine Learning

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Machine Learning to a Layman

You had a task : Identify Fruit Names
You Experienced: Remembered name and image of the fruit
Performance : How many fruit names you correctly Identified

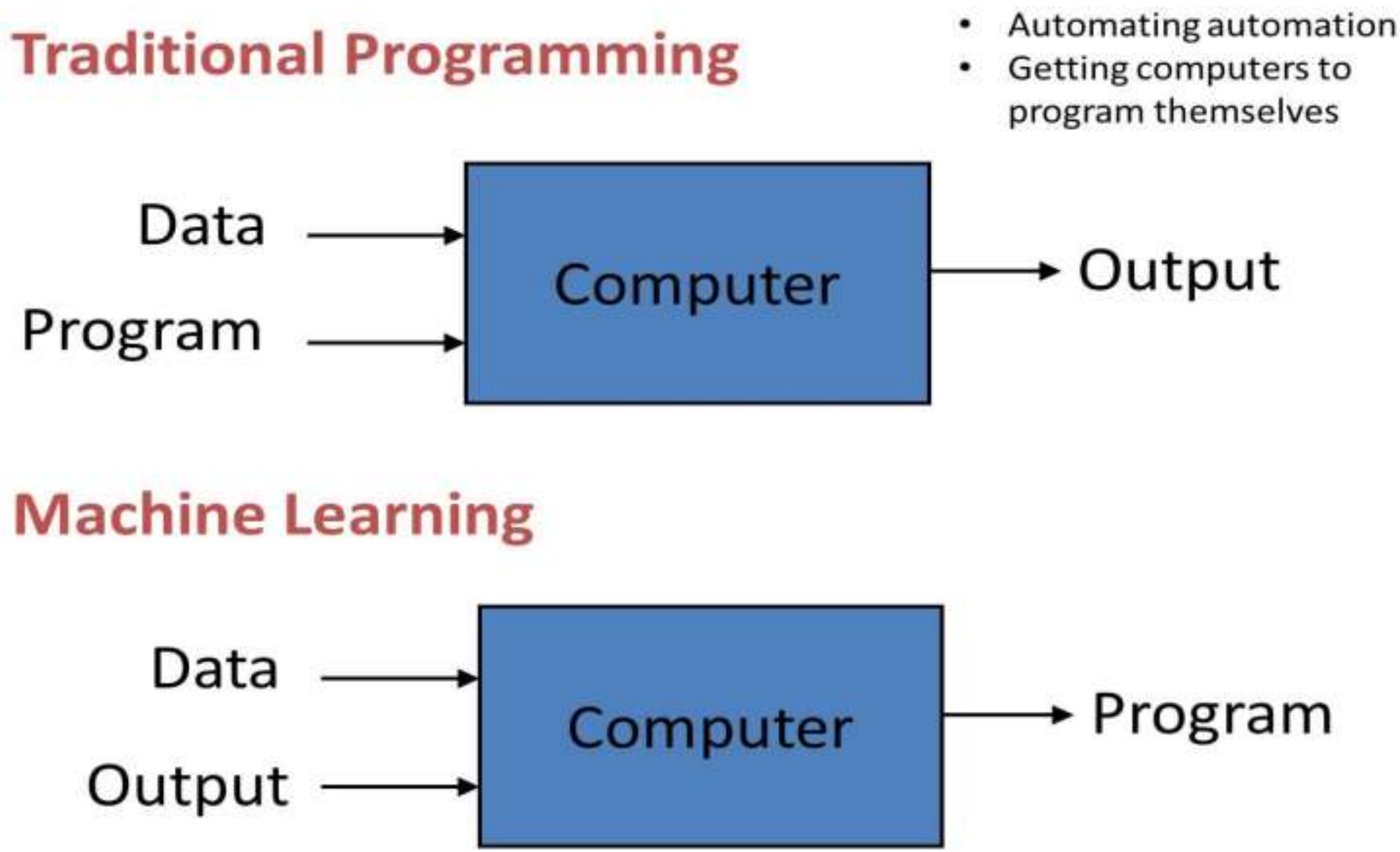
What is Machine Learning?

Machine Learning

Design of Algorithm that-

- Learn from data or build models using that data
- The learned model can be used to
  - Detect patterns/structures/themes/trends etc. in the data
  - Make predictions about future data and make decisions
- Modern ML algorithms are heavily "data-driven"
  - No need to pre-define and hard-code all the rules (usually infeasible/impossible anyway).
  - The rules are not "static"; can adapt as the ML algorithm ingests with more and more data.

Machine Learning vs Programming



When to Use Machine Learning?

- Human expertise is absent

  Example: navigating on mars

- Humans are unable to explain their expertise

  Example: vision, speech, language

- Requirements and data change over time

  Example: Tracking, Biometrics, Personalized fingerprint recognition

- The problem or the data size is just too large

  Example: Web Search

- When not to use it: If you can precisely/mathematically describe how to solve the task. Just program it.

Why Machine Learning?

- Machine Learning term first coined in 1959
- Computer Model based on Neural Network was created in 1943

Why Machine Learning?

- Machine Learning term first coined in 1959
- Computer Model based on Neural Network was created in 1943

Why Machine Learning?

**DAT
A**

OPTIMIZED ALGORITHMS

COMPUTING POWER

Why Machine Learning?

DATA

- **Structured Data**
- **Unstructured Data**

- “More than 300 million photos get uploaded **per day**.
- Every minute there are 510,000 comments posted and 293,000 statuses updated”
- “Over 2.5 quintillion bytes of data are **created every** single **day**, and it's only going to grow from there.
- By 2020, it is said that 1.7MB of data has been **created every** second for **every** person on earth”

More than 80% data is unstructured.

Why Machine Learning?

OPTIMIZED ALGORITHMS

Python
Libraries: Pandas, Numpy, Sklearn, Keras
TensorFlow, PyTorch, Theano

Less programming more science!

Why Machine Learning?

COMPUTING POWER

Powerful CPUs
GPU
Parallel and Distributed Computing

Jargon Difference!

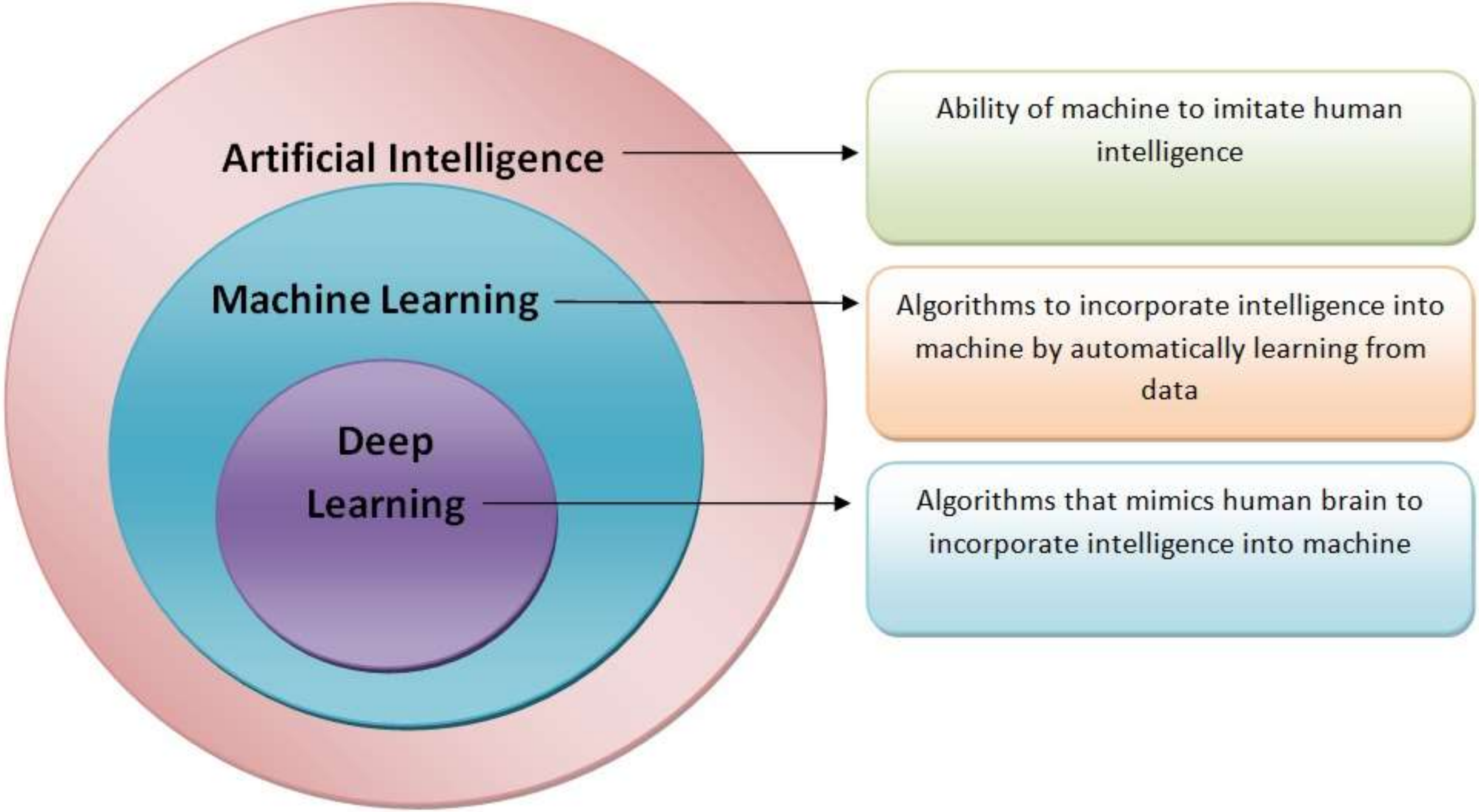
Machine Learning

Artificial Intelligence

Deep Learning

Data Science

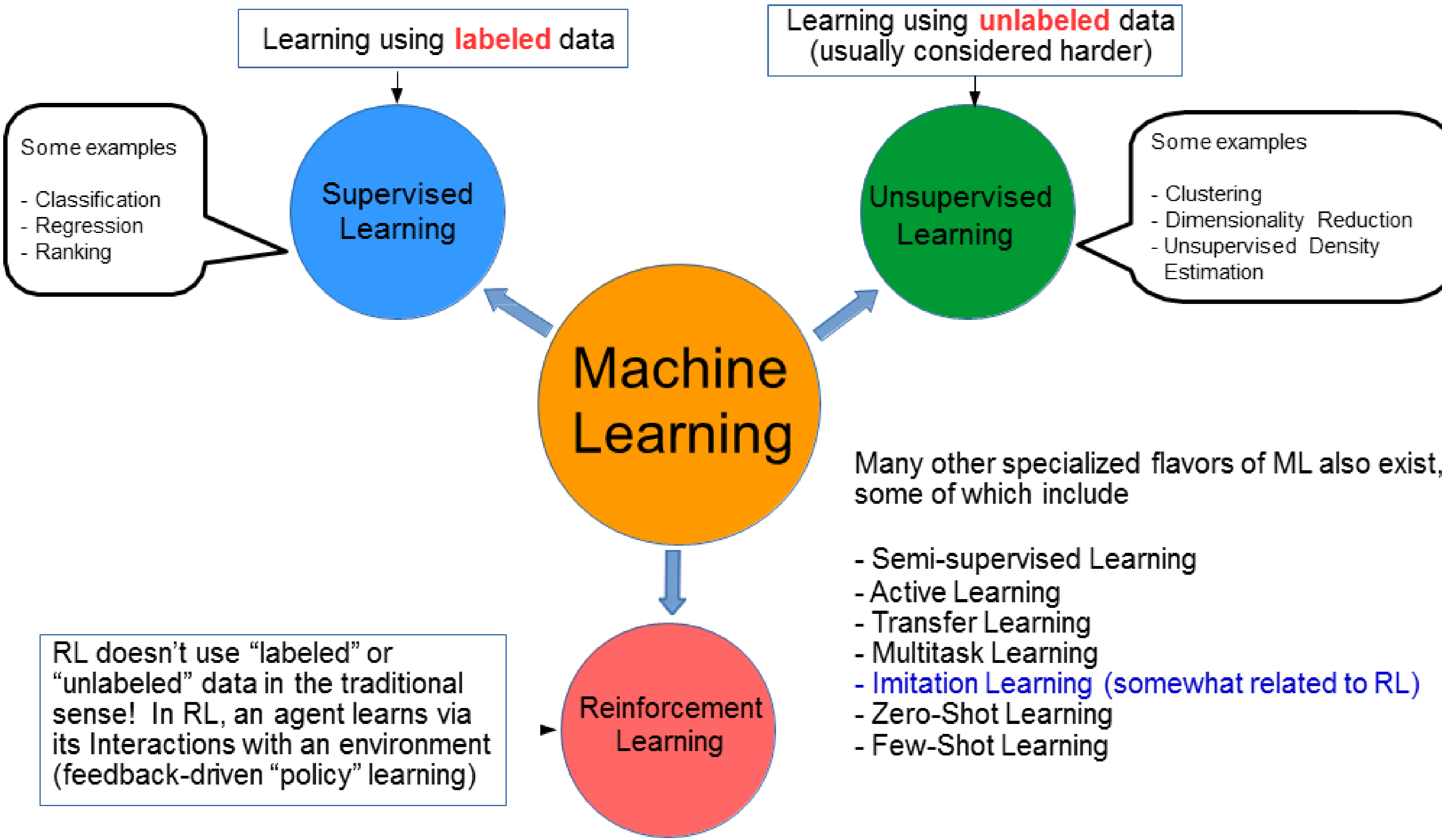
Jargon Difference!



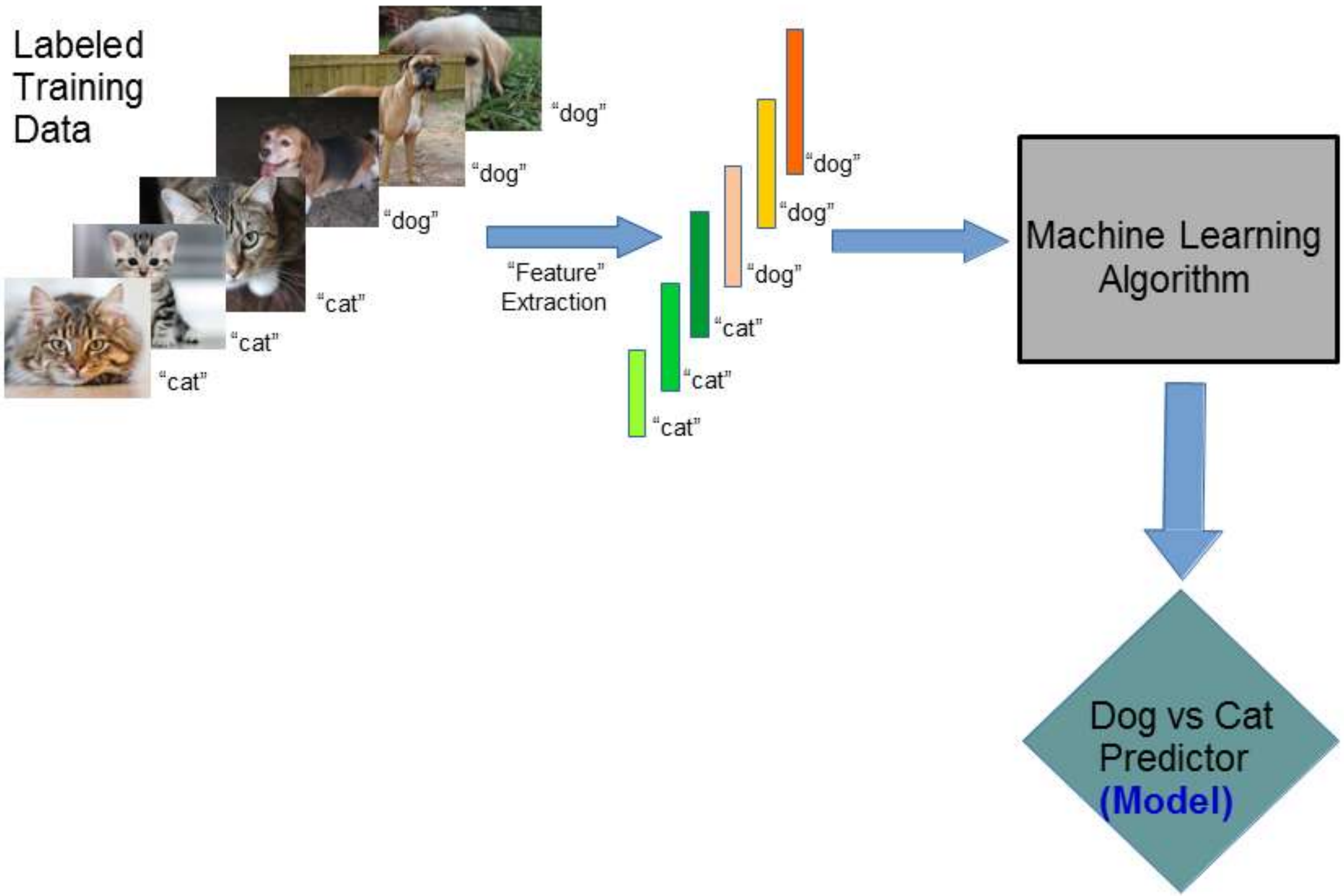
Types of Learning

- Supervised (inductive) learning: Training data includes desired outputs.
- Unsupervised learning: Training data does not include desired outputs, Find hidden/interesting structure in data.
- Semi-supervised learning: Training data includes a few desired outputs
- Reinforcement learning: the learner interacts with the world via “actions” and tries to find an optimal policy of behavior with respect to “rewards” it receives from the environment

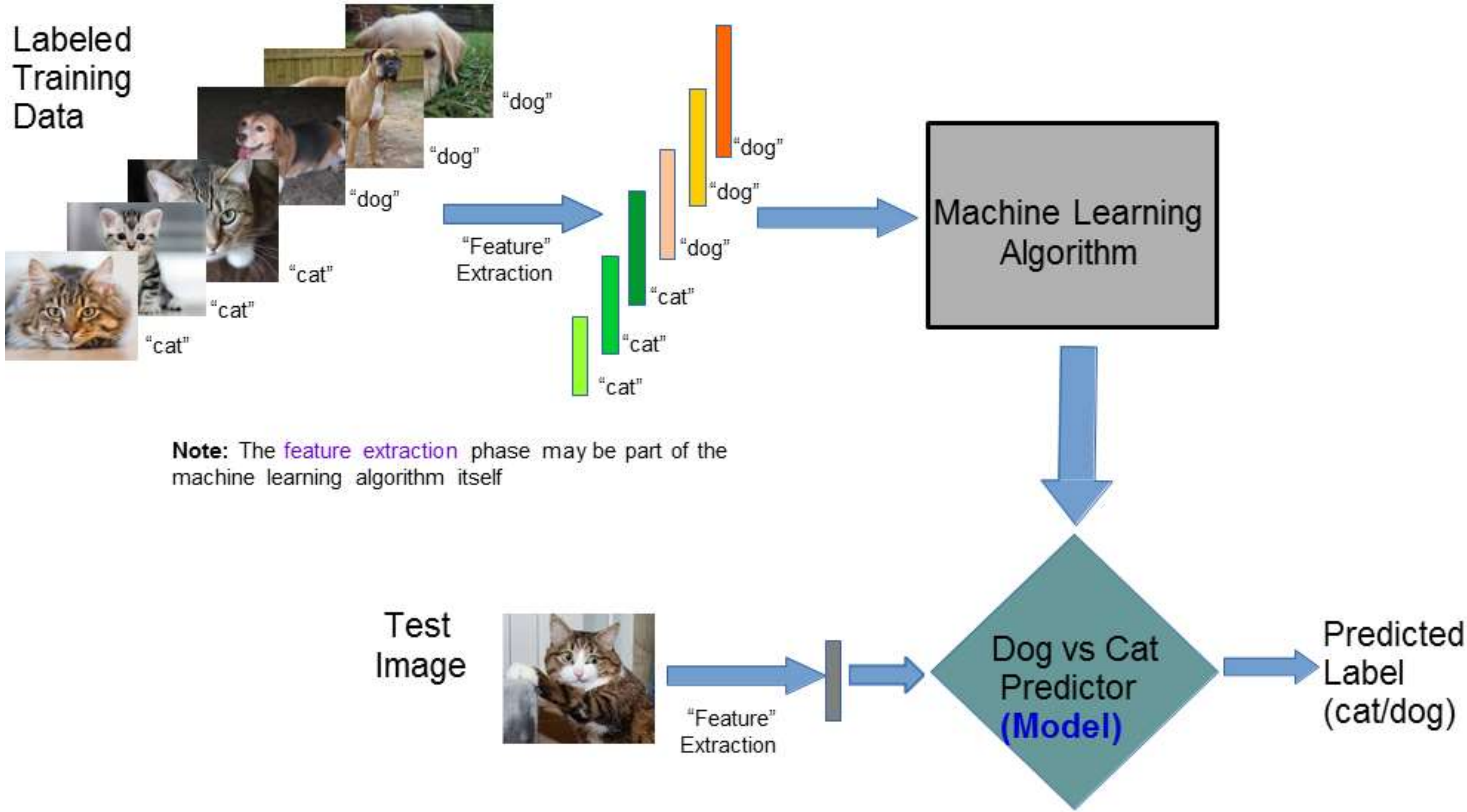
Types of Learning



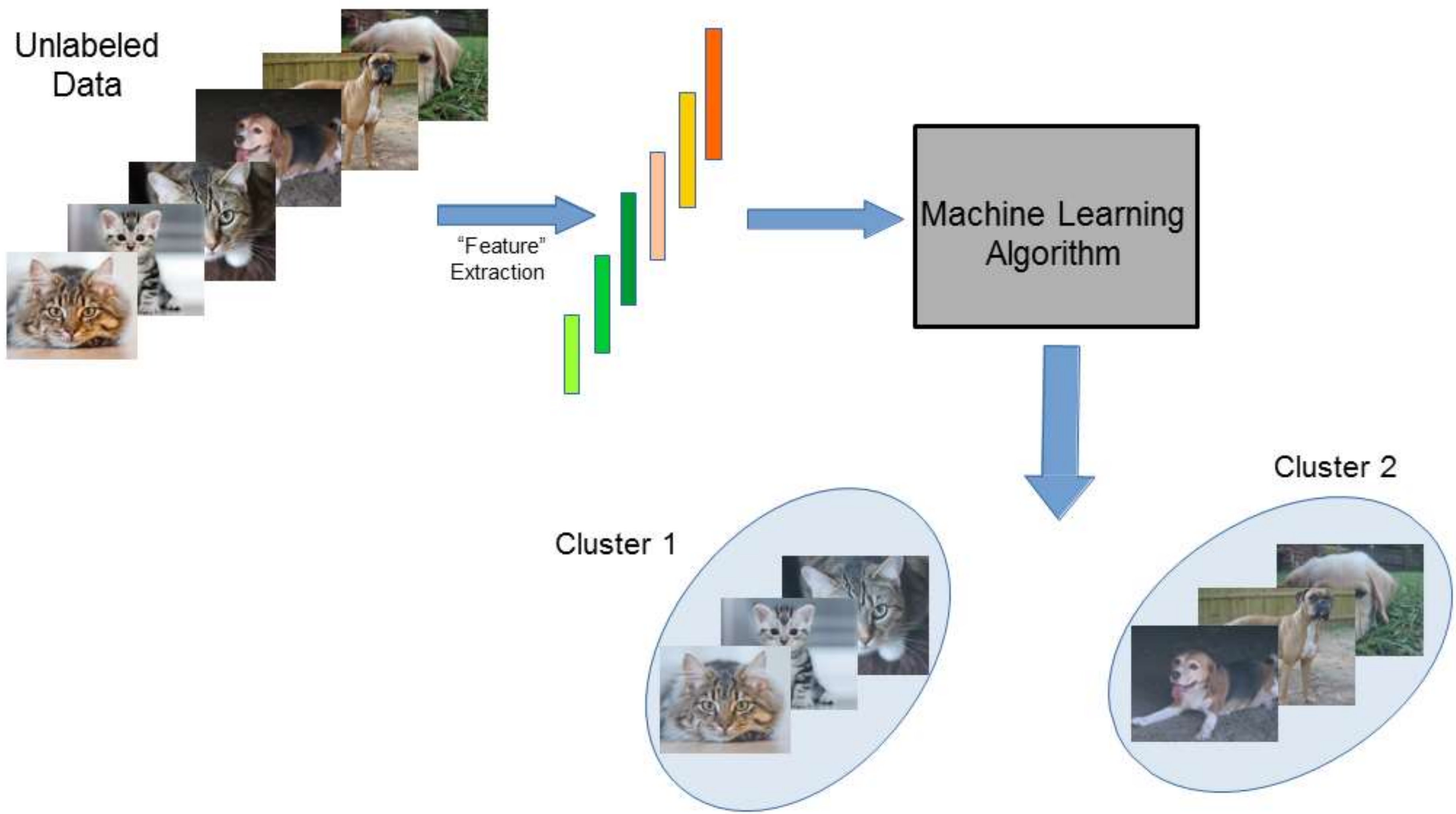
A Typical Supervised Learning Workflow (for Classification)



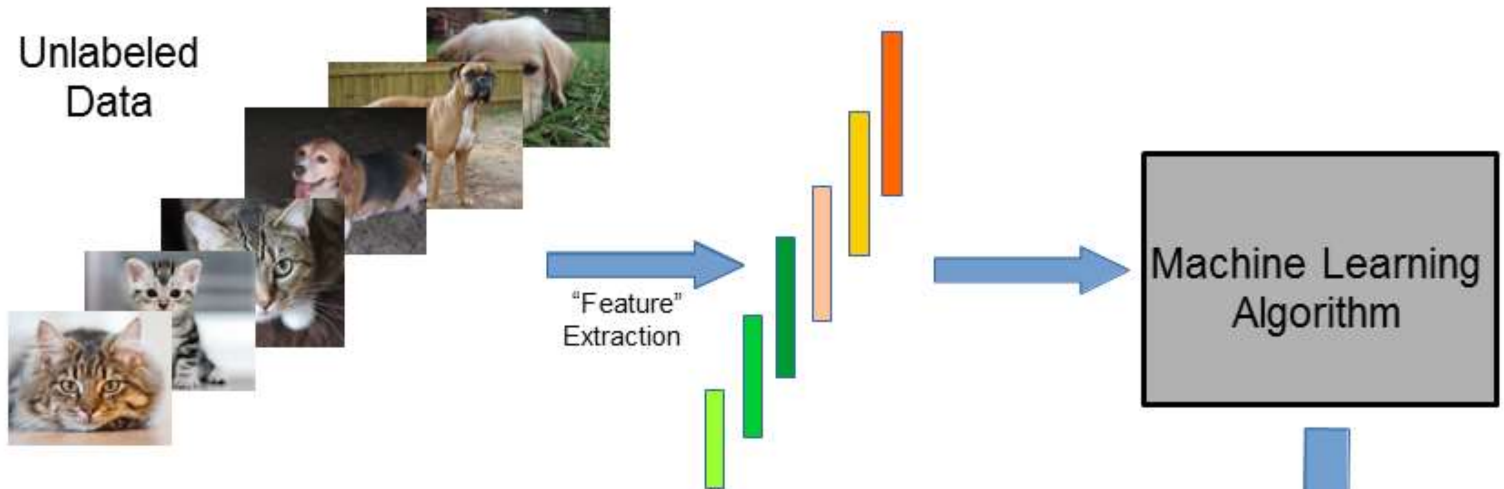
A Typical Supervised Learning Workflow (for Classification)



A Typical Un-supervised Learning Workflow (for

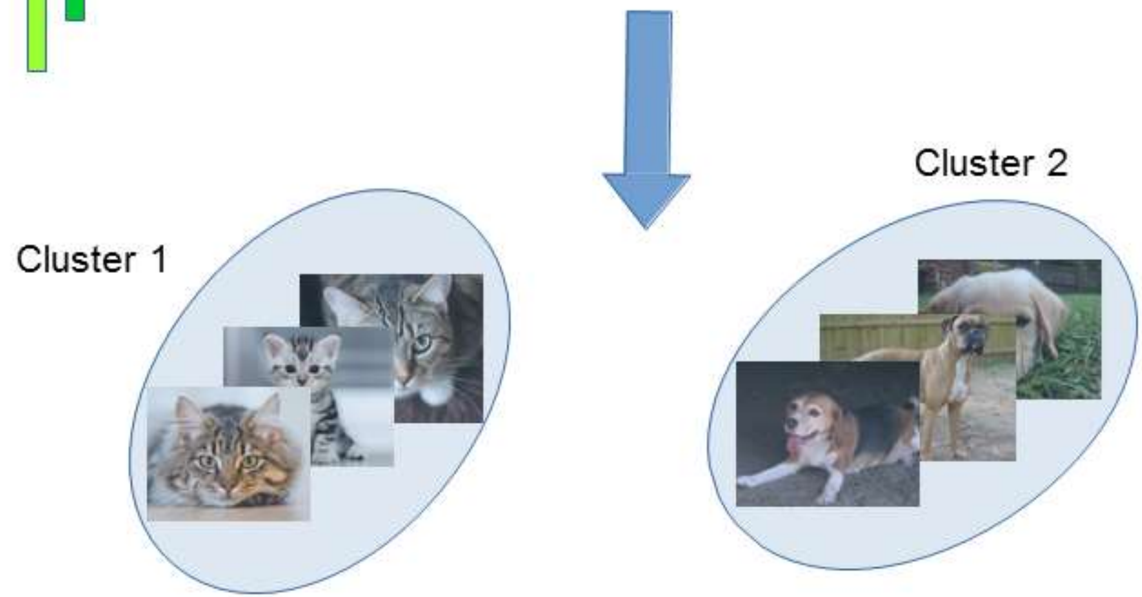


A Typical Un-supervised Learning Workflow (for

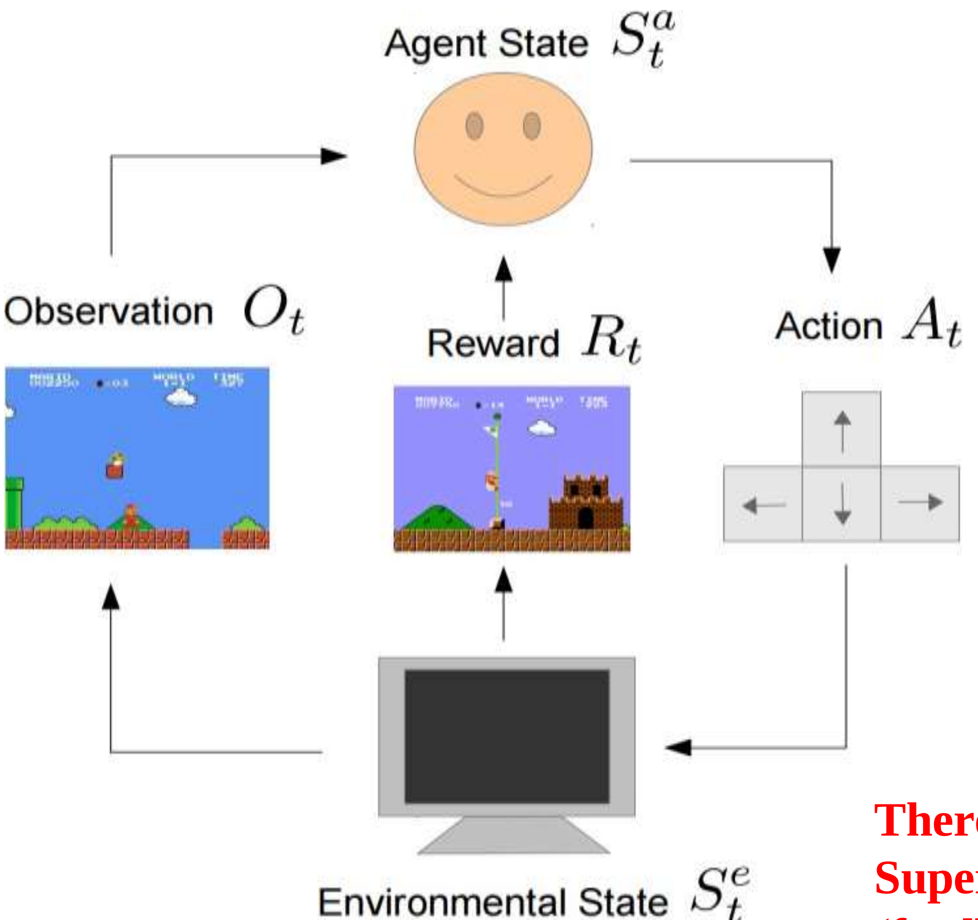


Note: Unsupervised Learning too can have (and often has) a "test" phase.
E.g., in this case, given a new cat/dog image, predict which of the two clusters it belongs to.

It can be done by assigning the image to the cluster with closer centroid



A Typical Reinforcement Learning Workflow



Agent's goal is to learn a policy for some task

Agent does the following repeatedly

- Senses/observes the environment
- Takes an action based on its current policy
- Receives a reward for that action
- Updates its policy

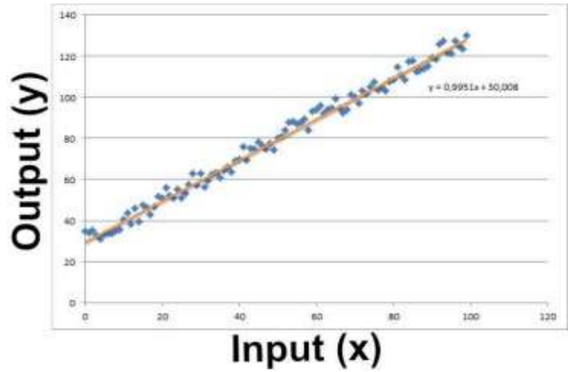
There IS supervision, not explicit (as in Supervised Learning) but rather implicit (feedback based)

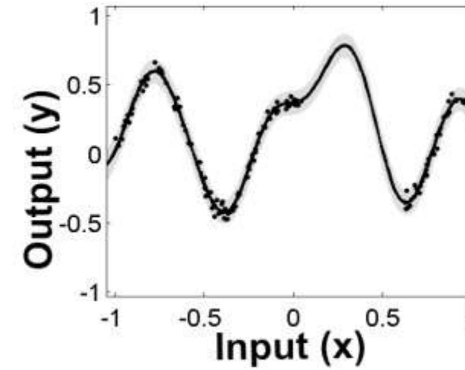
Geometric View of Some Basic ML Problems

Regression

Supervised Learning:
Learn a line/curve (the “model”) using training data consisting of Input-output pairs (each output is a real-valued number)

Use it to predict the outputs for new “test” inputs

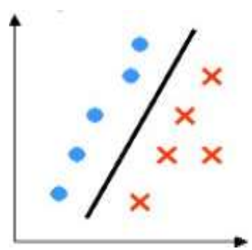




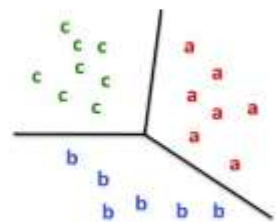
Classification

Supervised Learning: Learn a linear/nonlinear separator (the “model”) using training data consisting of input-output pairs

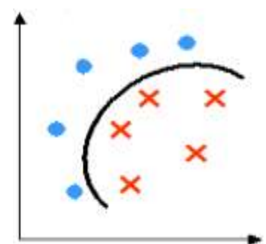
Use it to predict the labels for new “test” inputs



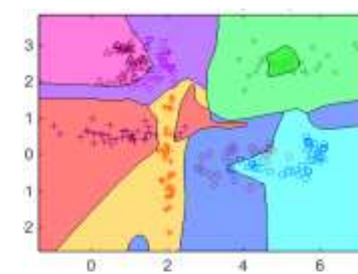
Two-Class (binary) Linear Classification



Multi-Class Linear Classification



Two-Class (binary) Nonlinear Classification

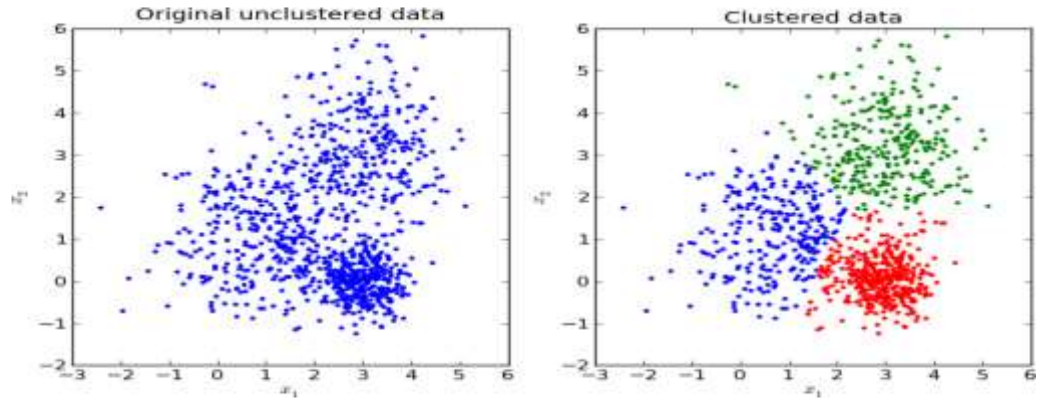


Multi-Class Nonlinear Classification

Geometric View of Some Basic ML Problems

Clustering

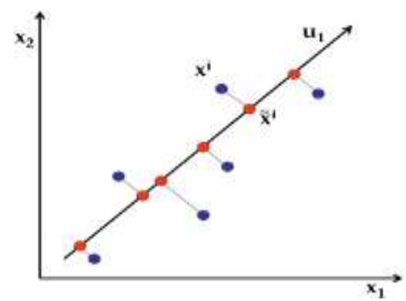
Unsupervised Learning: Learn the grouping structure for a given set of unlabeled inputs



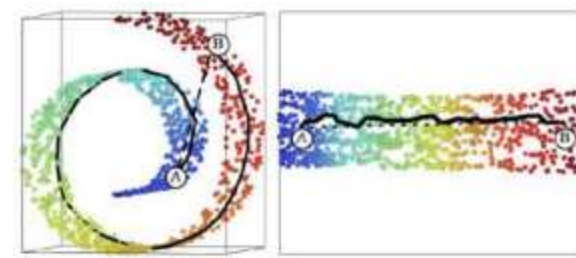
Dimensionality Reduction

Unsupervised Learning: Learn a Low-dimensional representation for a given set of high-dimensional inputs

Note: DR also comes in supervised flavors (supervised DR)



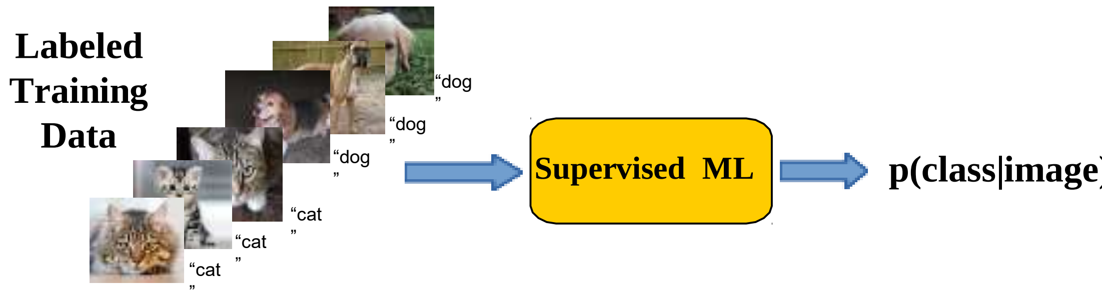
Two-dim to one-dim linear projection



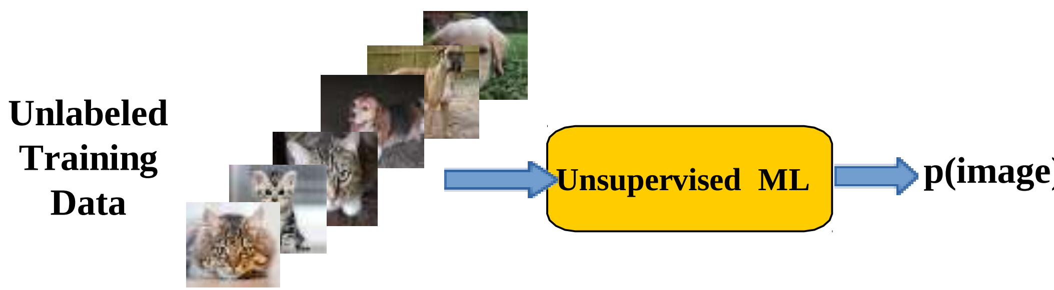
Three-dim to two-dim nonlinear projection (a.k.a. manifold learning)

Machine Learning = Probability Density Estimation

Supervised Learning (“predict *y* given ***x*** ”) can be thought of as estimating $p(y|\mathbf{x})$



Unsupervised Learning (“model x ”) can also be thought of as estimating p(x)

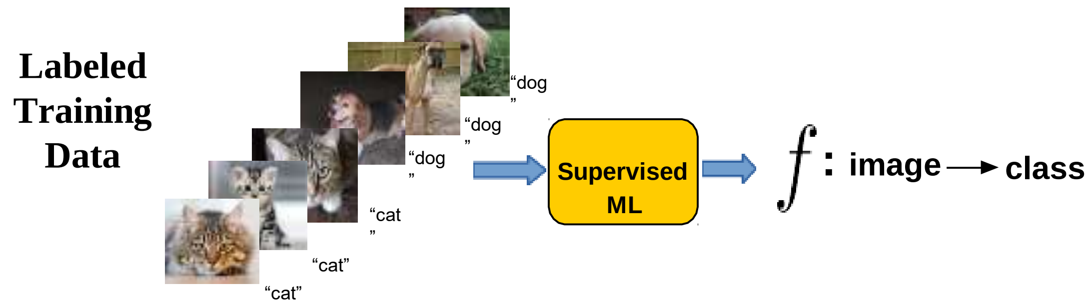


Harder for Unsupervised Learning because there is no supervision y

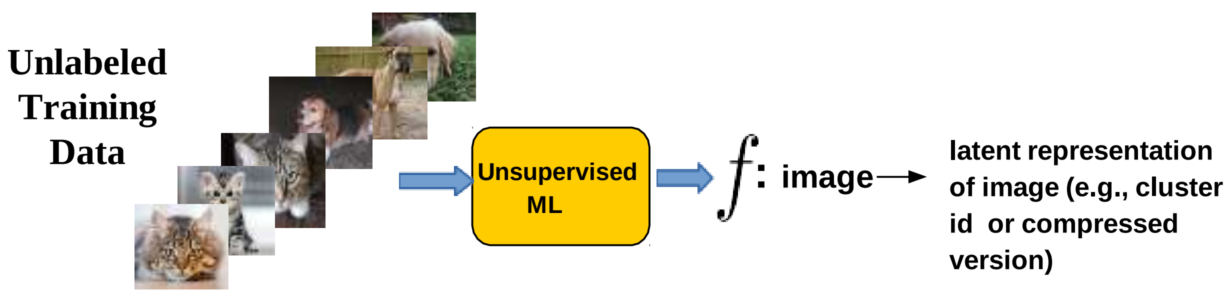
Other ML paradigms (e.g., Reinforcement Learning) can be thought of as learning prob. density

Machine Learning = Function Approximation

Supervised Learning (“predict y given x ”) can be thought learning a function that maps x to y



Unsupervised Learning (“model x ”) can also be thought of as learning a function that maps x to some useful latent representation of x

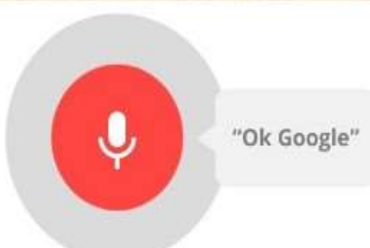


Harder for Unsupervised Learning because there is no supervision y

Other ML paradigms (e.g., Reinforcement Learning) can be thought of as doing function approx.

Machine Learning in the real-world

Broadly applicable in many domains (e.g., internet, robotics, healthcare and biology, computer vision, NLP, databases, computer systems, finance, etc.)

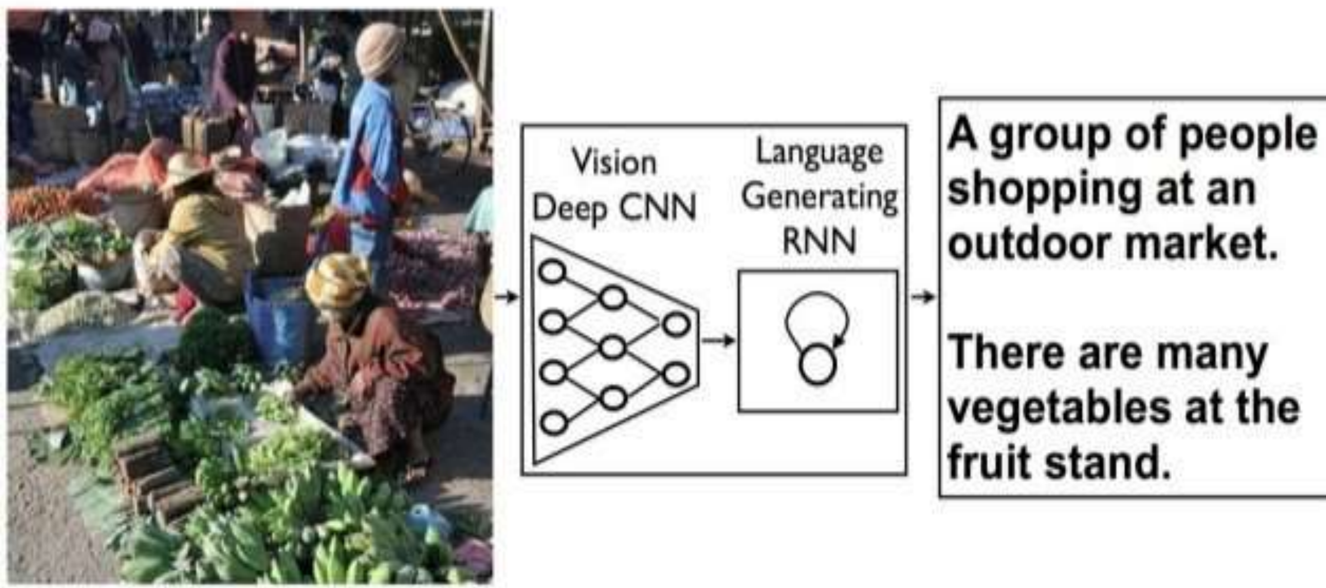



Predictive Policing


Online Fraud Detection

Machine Learning helps Computer Vision

ML algorithms can learn to generate captions for images



http://arxiv.org/abs/1411.4555 “Show and Tell: A Neural Image Caption Generator”

Machine Learning helps Computer

ML algorithms can learn to answer questions about images (Visual QA)

Machine Learning helps NLP

ML algorithms can learn to translate text

(even “transliterate”)

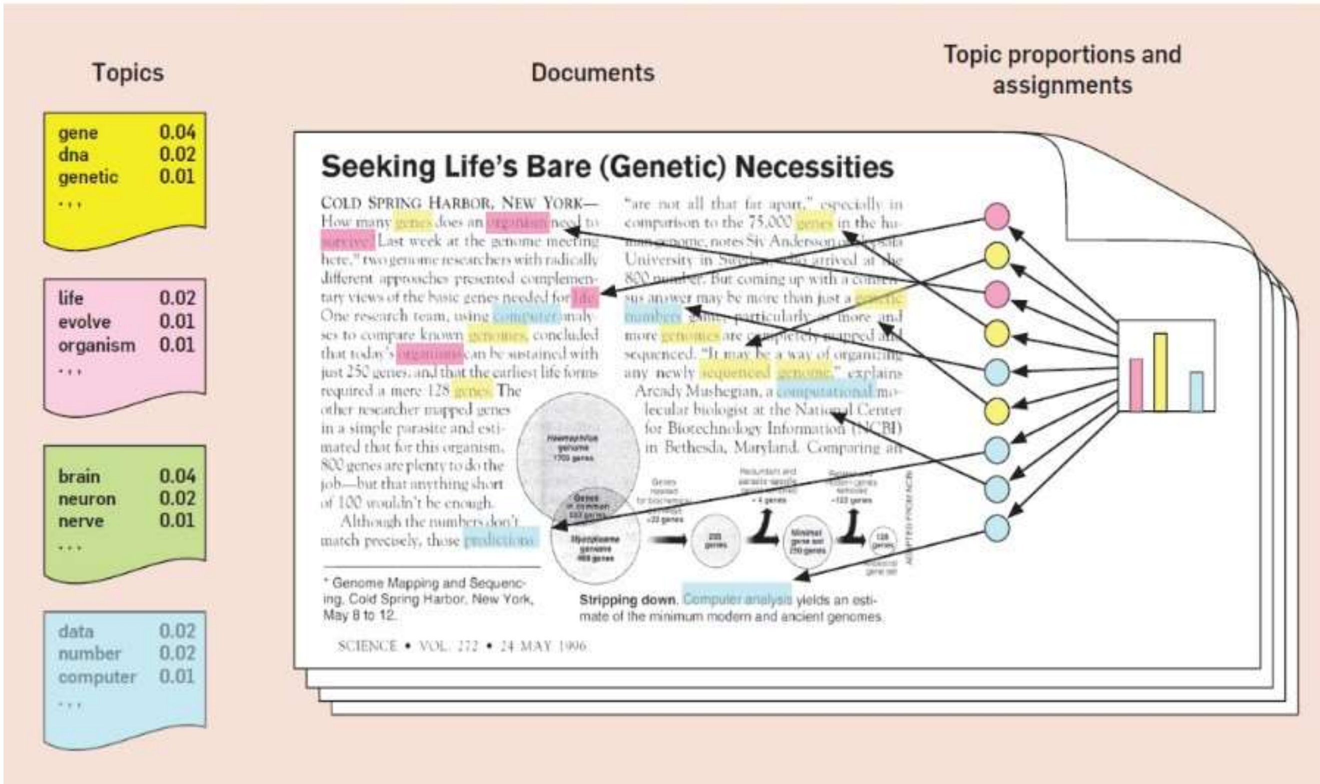
Machine Learning helps NLP

ML algorithms can learn to summarize text

Input: Article 1st sentence	Model-written headline
metro-goldwyn-mayer reported a third-quarter net loss of dlrs 16 million due mainly to the effect of accounting rules adopted this year	mgm reports 16 million net loss on higher revenue
starting from july 1, the island province of hainan in southern china will implement strict market access control on all incoming livestock and animal products to prevent the possible spread of epidemic diseases	hainan to curb spread of diseases
australian wine exports hit a record 52.1 million liters worth 260 million dollars (143 million us) in september, the government statistics office reported on monday	australian wine exports hit record high in september

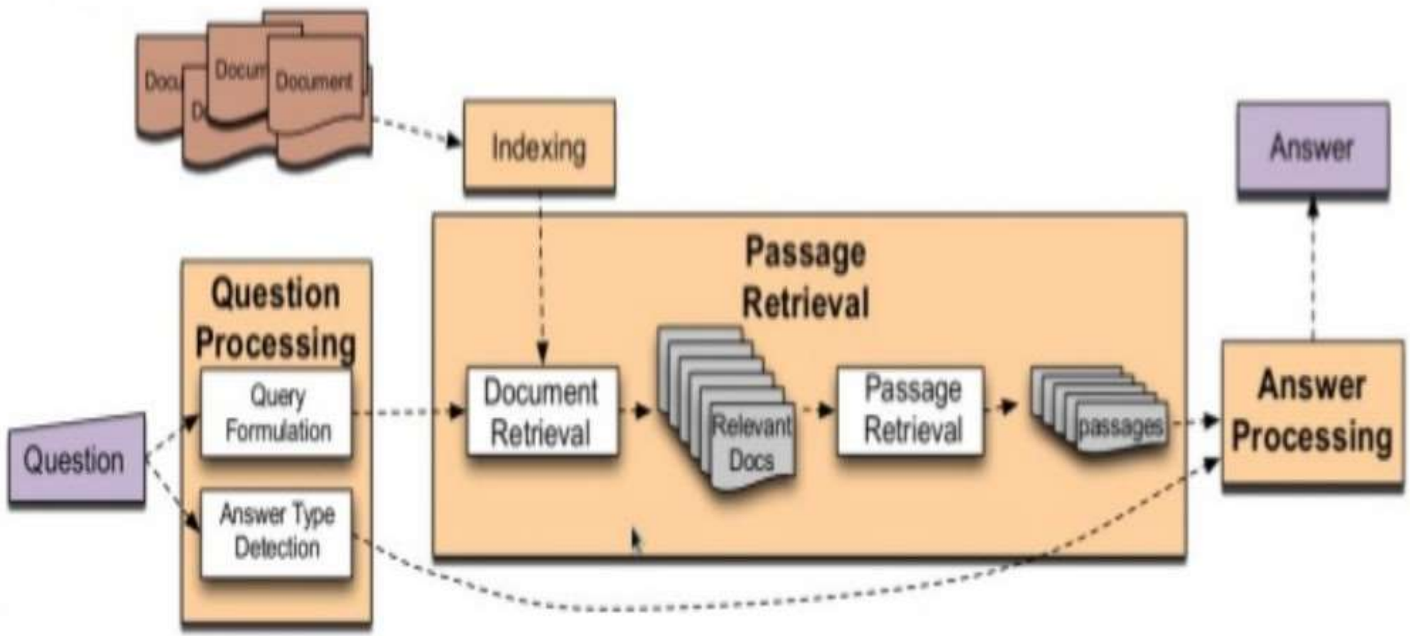
Machine Learning helps NLP

ML algorithms can learn the topics in a text corpus ("Topic Modeling")



Machine Learning helps NLP- Search and Info Retrieval

ML algorithms can learn to search for the answer to a given question from a large database of documents

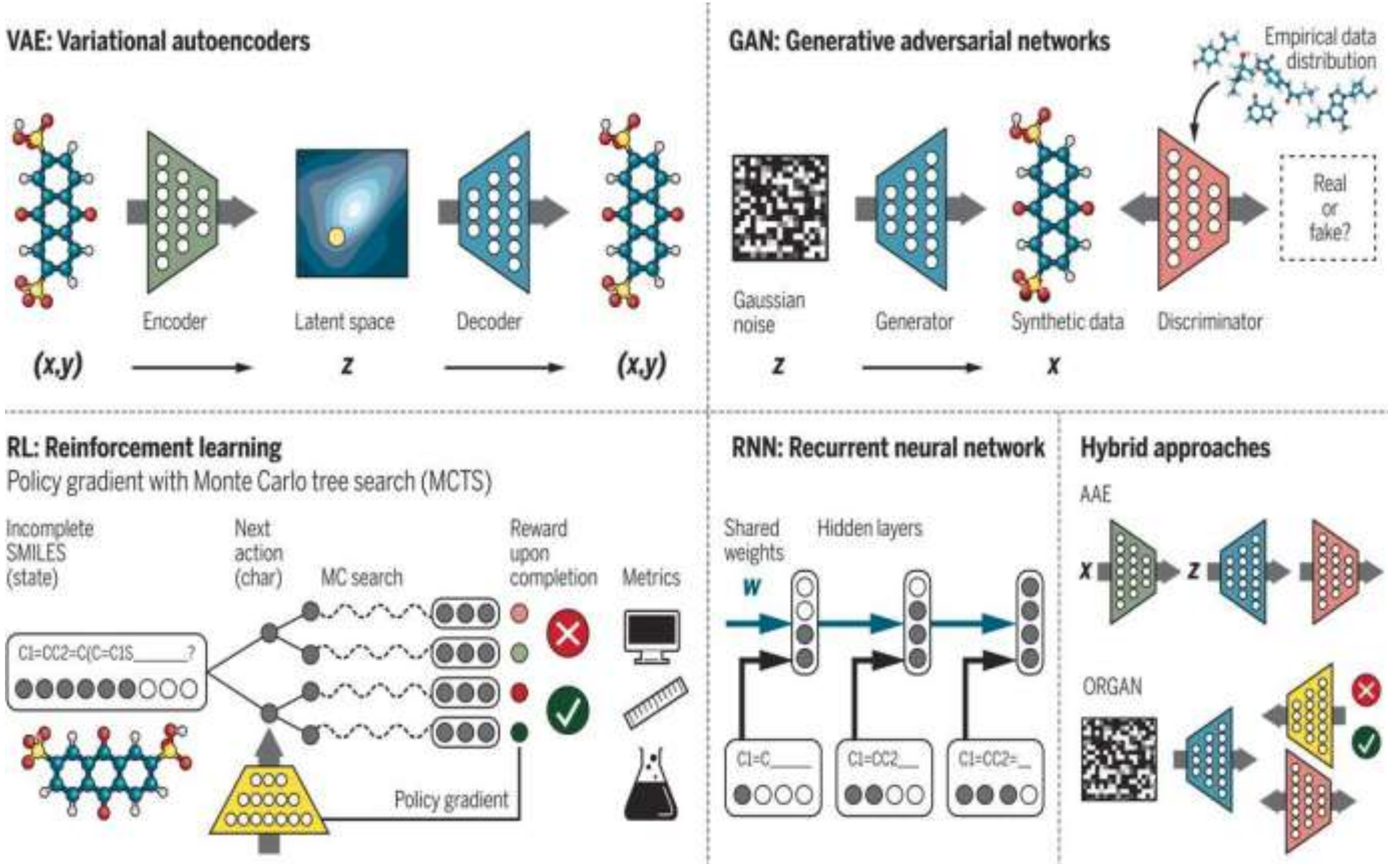


Machine Learning meets Speech Processing

ML algorithms can learn to translate speech in **real time**

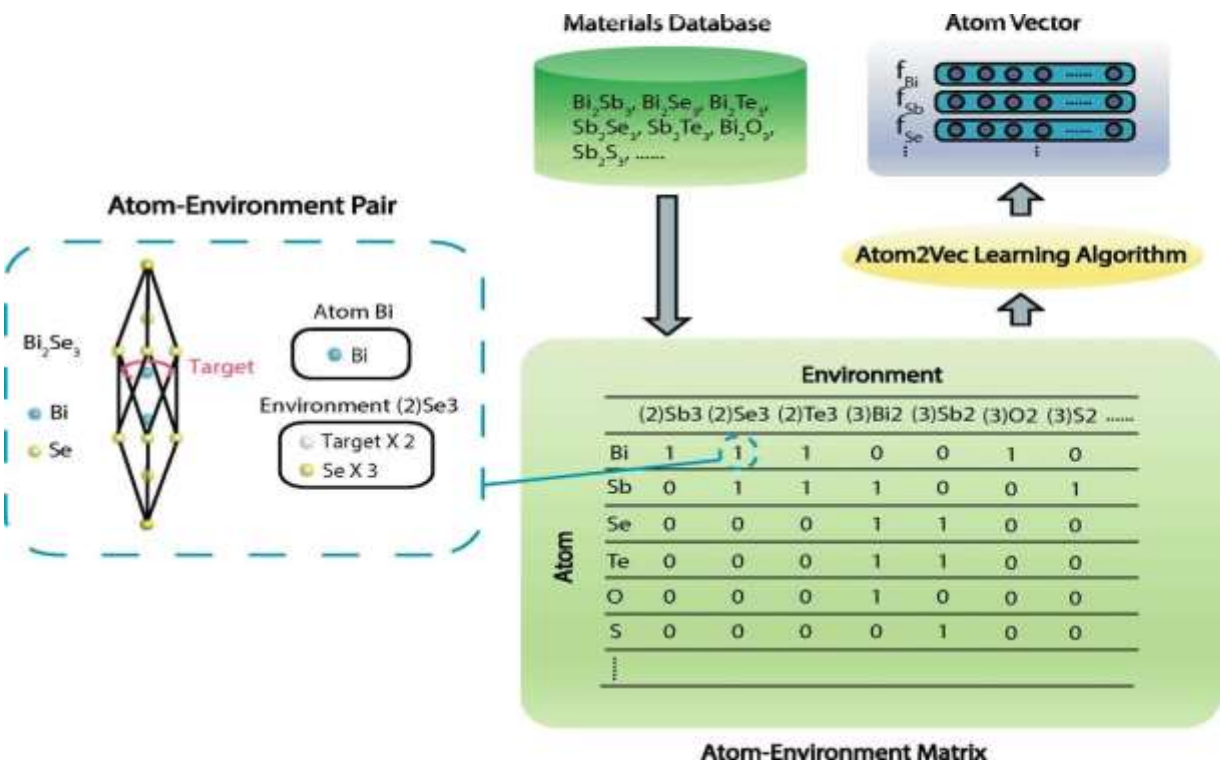
Machine Learning helps Chemistry

ML algorithms can understand properties of molecules and learn to synthesize new molecules

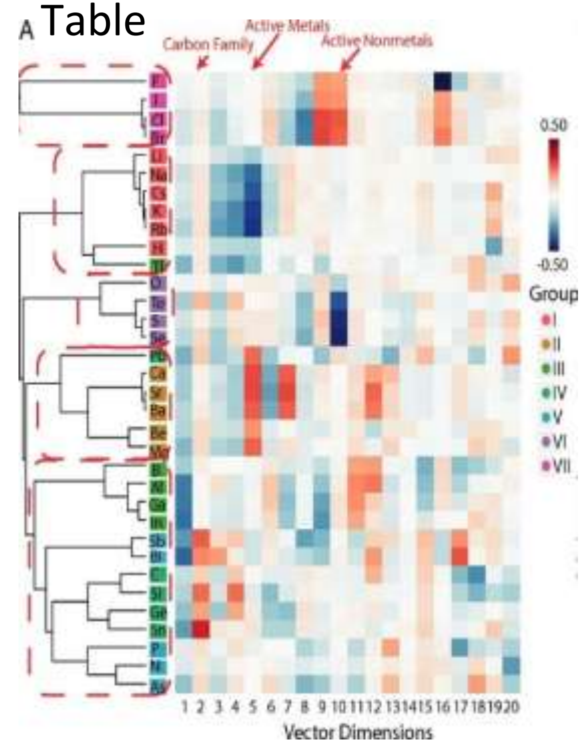


Machine Learning helps Chemistry

ML algorithms can “read” databases of matetials and recreate the Periodic Table within hours

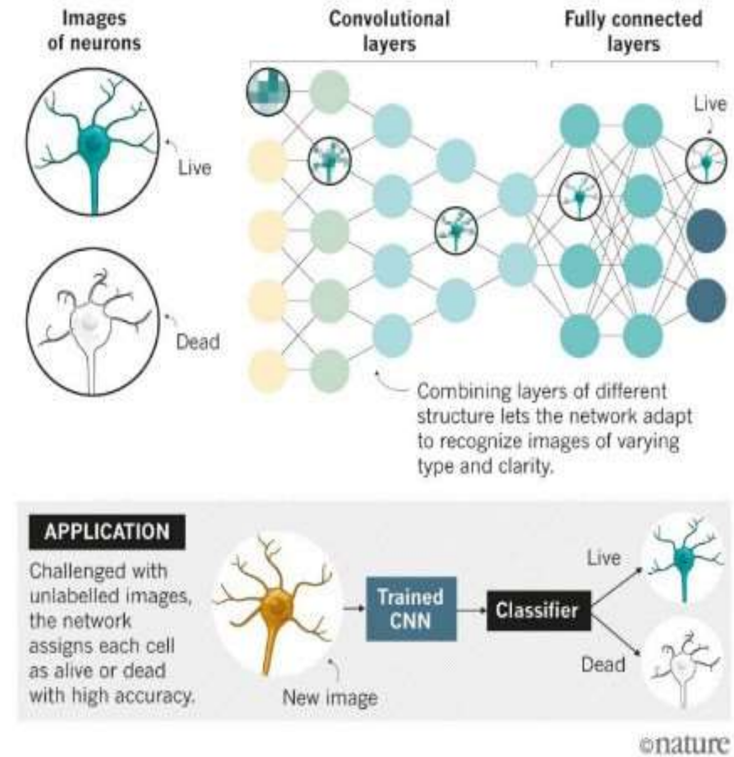


“Recreated” Periodic Table



Machine Learning helps in Biology, E-commerce

Biology



Source: Jeremy Linsley/Drew Linsley/Steve Finkbeiner/Thomas Serre

Finance

Inductive Learning

Inductive Learning

What is Inductive Learning Algorithm?

Inductive Learning Algorithm (ILA) is an iterative and inductive machine learning algorithm that is used for **generating a set of classification rules**, which produces rules of the form "IF-THEN", for a set of examples, producing rules at each iteration and appending to the set of rules.

- Also called as Deterministic Supervised Learning
- In this, first input x, (the verified value) given to a function f, and the output is f(x).
- Then we can give different set of inputs (raw inputs) to the same function f, and verify the output f(x).
- **By using the outputs we generate (learn) the rules**.

Need for Inductive Learning

- There are basically two methods for knowledge extraction firstly from domain experts and then with machine learning.
- For a very large amount of data, the domain experts are not very useful and reliable.
- So we move towards the machine learning approach for this work.

Inductive Learning

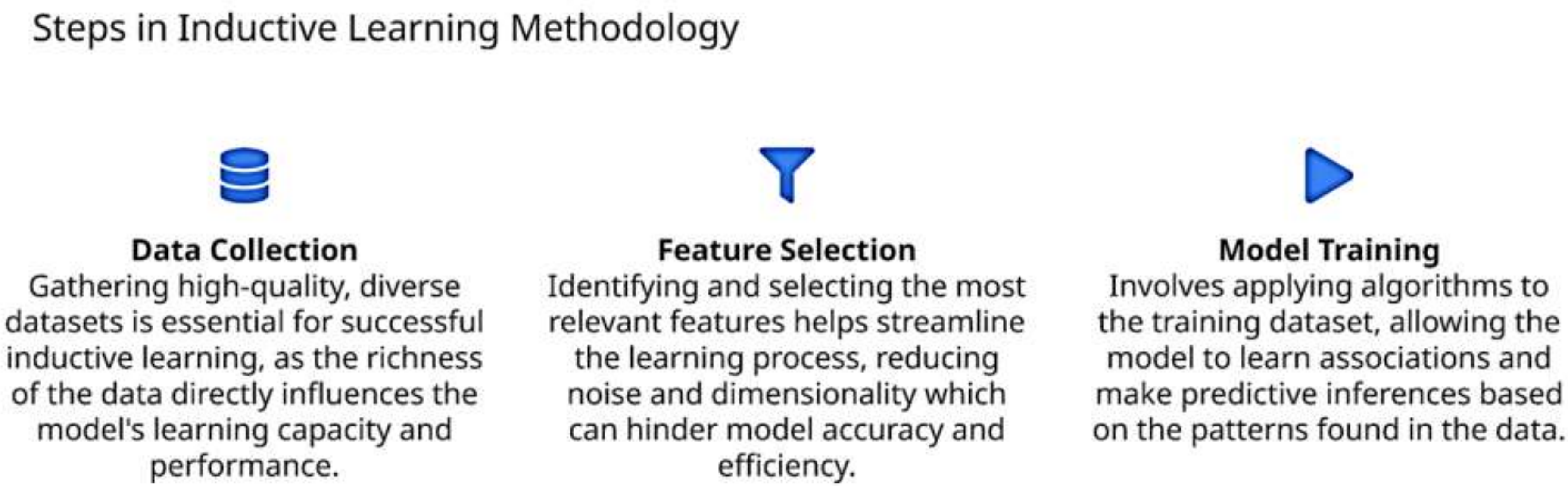
- Inductive learning, also known as discovery learning, is a process where the learner discovers rules by observing examples.
- We can often work out rules for ourselves by observing examples. If there is a pattern; then record it.
- We then apply the rule in different situations to see if it works.
- With inductive language learning, tasks are designed specifically to guide the learner and assist them in discovering a rule.

Inductive Learning

Definition of Inductive Learning
Inductive learning is a fundamental process by which models infer generalizable patterns from specific examples, enabling predictions about unseen data based on learned information.

Importance in Machine Learning
Inductive learning forms the backbone of many machine learning algorithms, allowing systems to adapt and improve from exposure to data, thereby enhancing decision-making processes.

Inductive Learning Process



Inductive Learning Process

Data Collection:

- Gather labeled examples representing the problem domain.
- Example: Emails labeled as spam or non-spam in a classification task.

Hypothesis Space:

- Define the set of possible hypotheses/models based on the chosen algorithm and inductive bias.

Hypothesis Generation:

- Construct potential hypotheses using observed examples.
- Analyze instance features to identify patterns or relationships.

Hypothesis Evaluation:

- Assess hypotheses using evaluation metrics or validation techniques.
- Test predictive accuracy on new, unseen examples.

Hypothesis Refinement:

- Refine hypotheses iteratively based on evaluation feedback.
- Update models to enhance performance and generalization.

Generalization:

- Apply the final model to classify or predict new, unseen data instances.

Inductive Learning

- **Inductive learning** or "**Prediction**":
  - **Given** examples of a function *(X, F(X))*
  - **Predict** function *F(X)* for new examples *X*

  This is the function which we are trying to learn.

- Classification

  F(X) = Discrete
- Regression

  F(X) = Continuous
- Probability estimation

  F(X) = Probability*(X):*

Why it is called Inductive learning?

We are given some data and we are trying to do induction to identify a function which can explain that data.

Classification Learning

Task T

Input

- A set of instances d1, d2, ..., dn
- An instance has a set of features
- We can represent an instance as a vector
- d = <x1,x2, x3, ..., xn>

Output

- A set of predictions y1, y2, y3, ..., yc
- One of the fixed set of constant values
- Eg: {+1, -1}

Performance P - How accurately model predicts the output

Experience E - A set of labeled examples (x,y) where y is the true label of x.

Basic Terminologies

Types of features

1. **Categorical** - It will have finite number of categories and classes.

   Example

   – Gender - Male, Female
   – Age group: (0-12) children, (13-19) teenagers, (20-30) adults, 31-60 working professionals, above 60 senior citizen,
   – Blood group: A, B, AB, O etc

2. **Integer Valued**

   Example: Number of words in a text

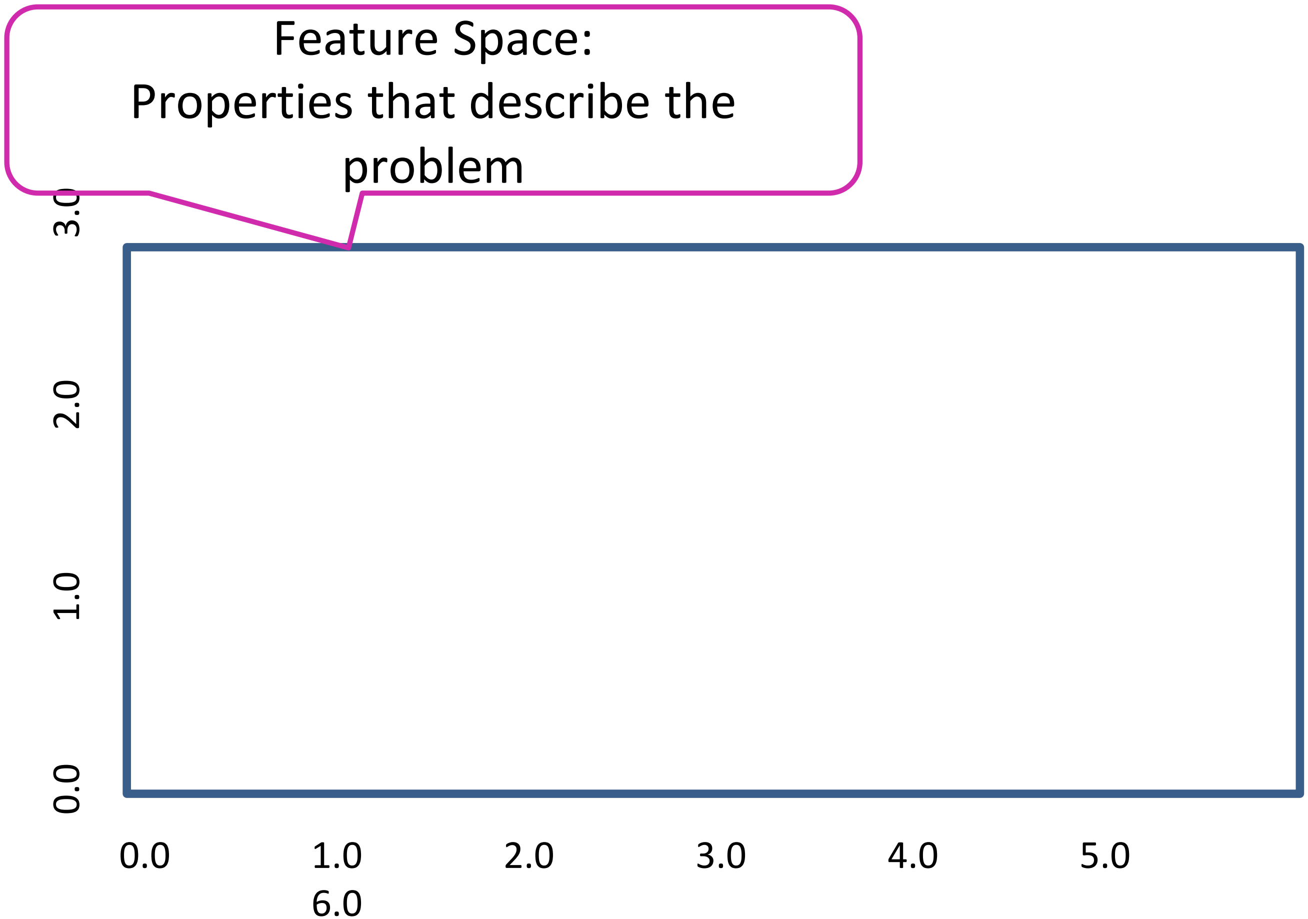
3. **Continuous** - Are those which can take INFINITE number of values.

   Example: Age, height, weight, price etc.

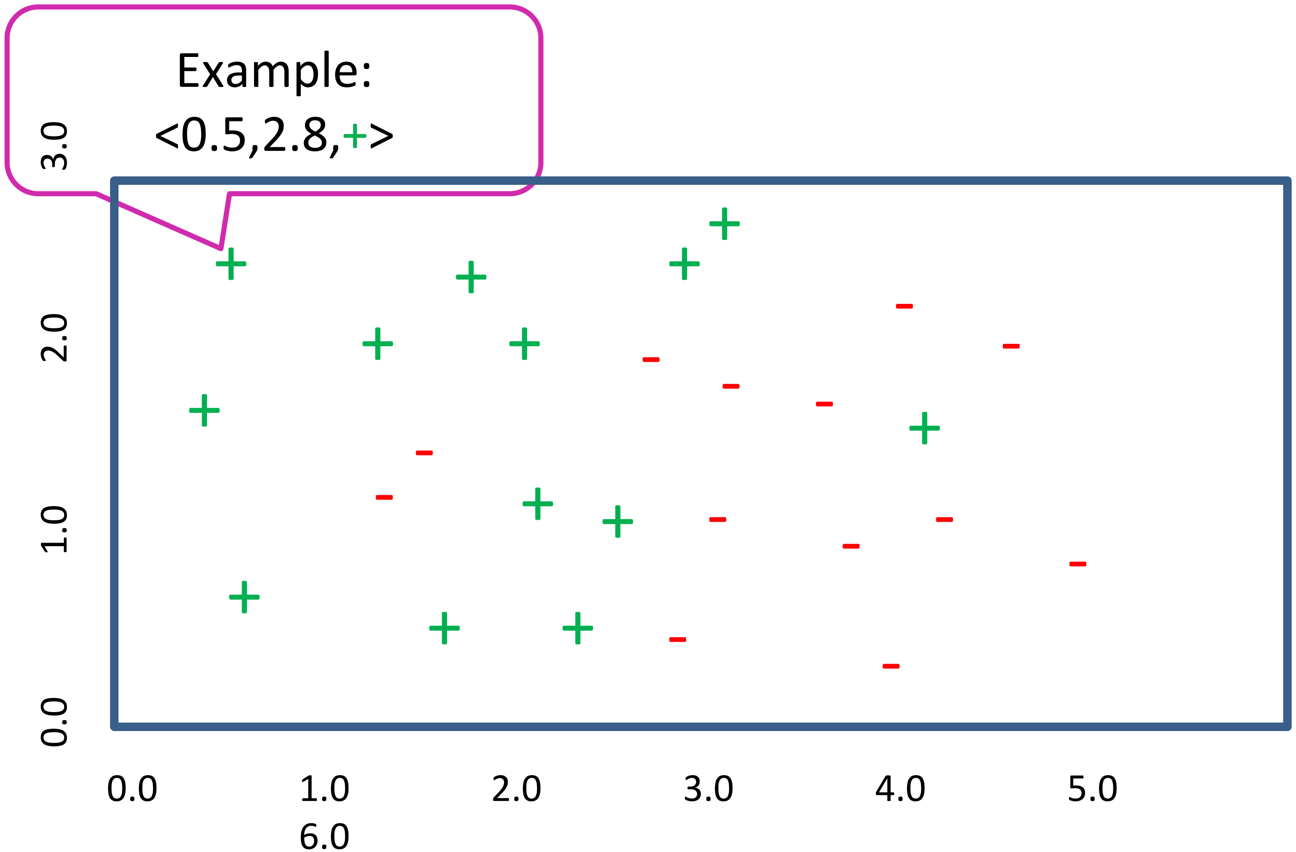
Basic Terminologies

- **Feature**: Distinct characters that can be used to describe each object in a quantitative manner.
- **Feature Vector**: n-dimensional vector of numerical features that represent some object.
- **Feature Space:**
  - Suppose we have two features x1 and x2
  - Two features will define two dimensional feature space
  - In general n-features will define n-dimensional feature space.
- **Instance Space X:** Set of all possible objects that can be described by features.
- **Target Function:** It is function we are trying to learn
- **Training data set:**
  - Collection of examples observed by learning algorithms
  - It is a used to discover potentially predictive relationship

Basic Terminologies



Basic Terminologies



Basic Terminologies

<u>Possible Functions</u>

<u>1. Slanted line</u> with 2 parameters

$y= mx +c$

-> We need to define both intercept and slope.

<u>2. Polynomial</u>

quadratic function: $ax^2 + bx +c$

a,b,c- 3 parameters

<u>3. Complex Function</u>

Note:- We are interested in a function which not only fit the training data but also works well with future or test data.

Basic Terminologies

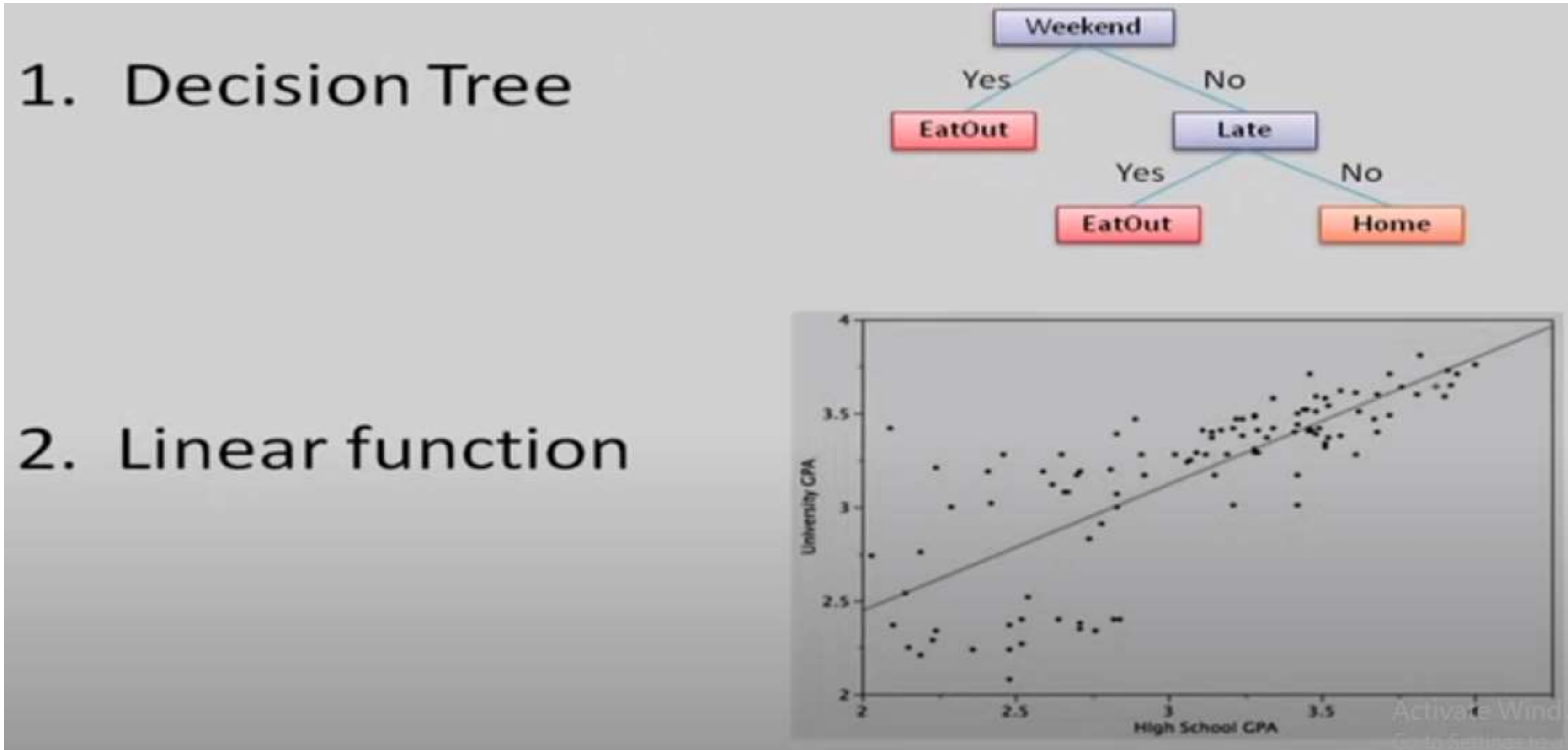
Representation of Function

- When we talk of representation of these hypothesis (or functions) then we have two things, one is features and the other is the function class.

- Function/ model/ hypothesis all are same things.

Basic Terminologies

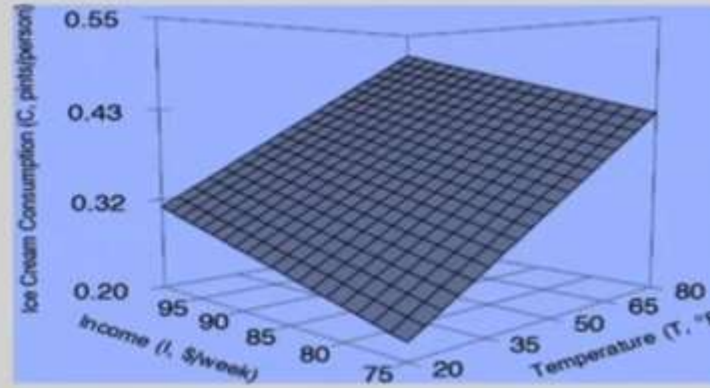
Representation

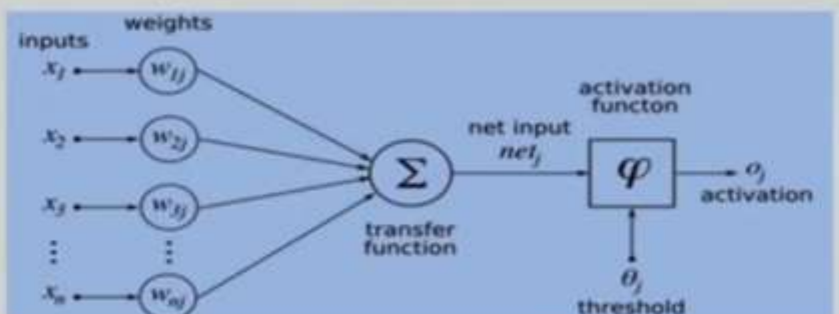


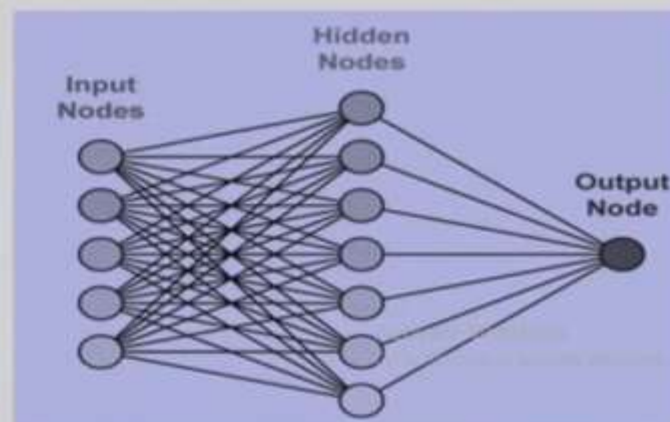
Basic Terminologies

Representation

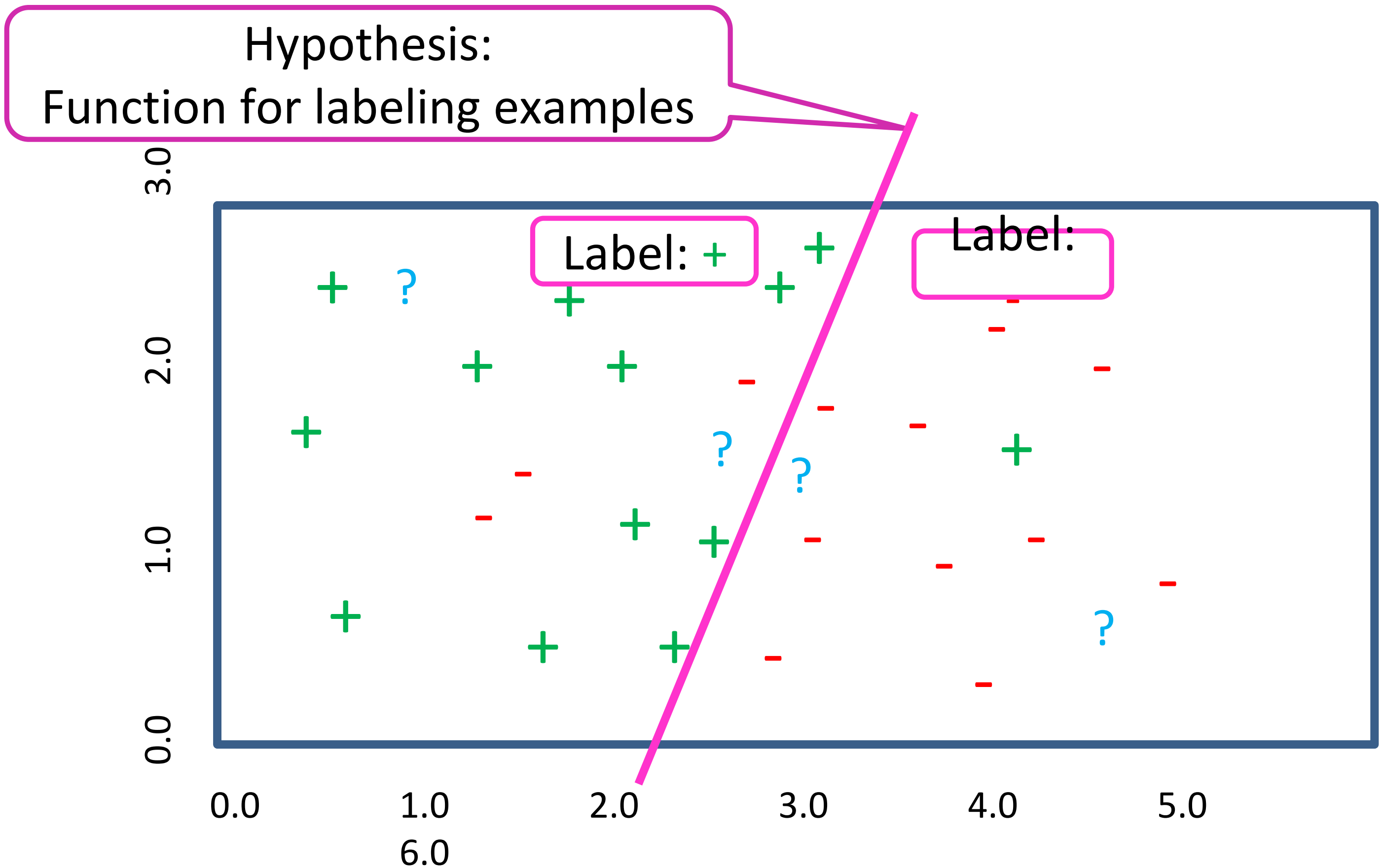
3. Multivariate linear function
4. Single layer perceptron
5. Multi-layer neural networks







Basic Terminologies



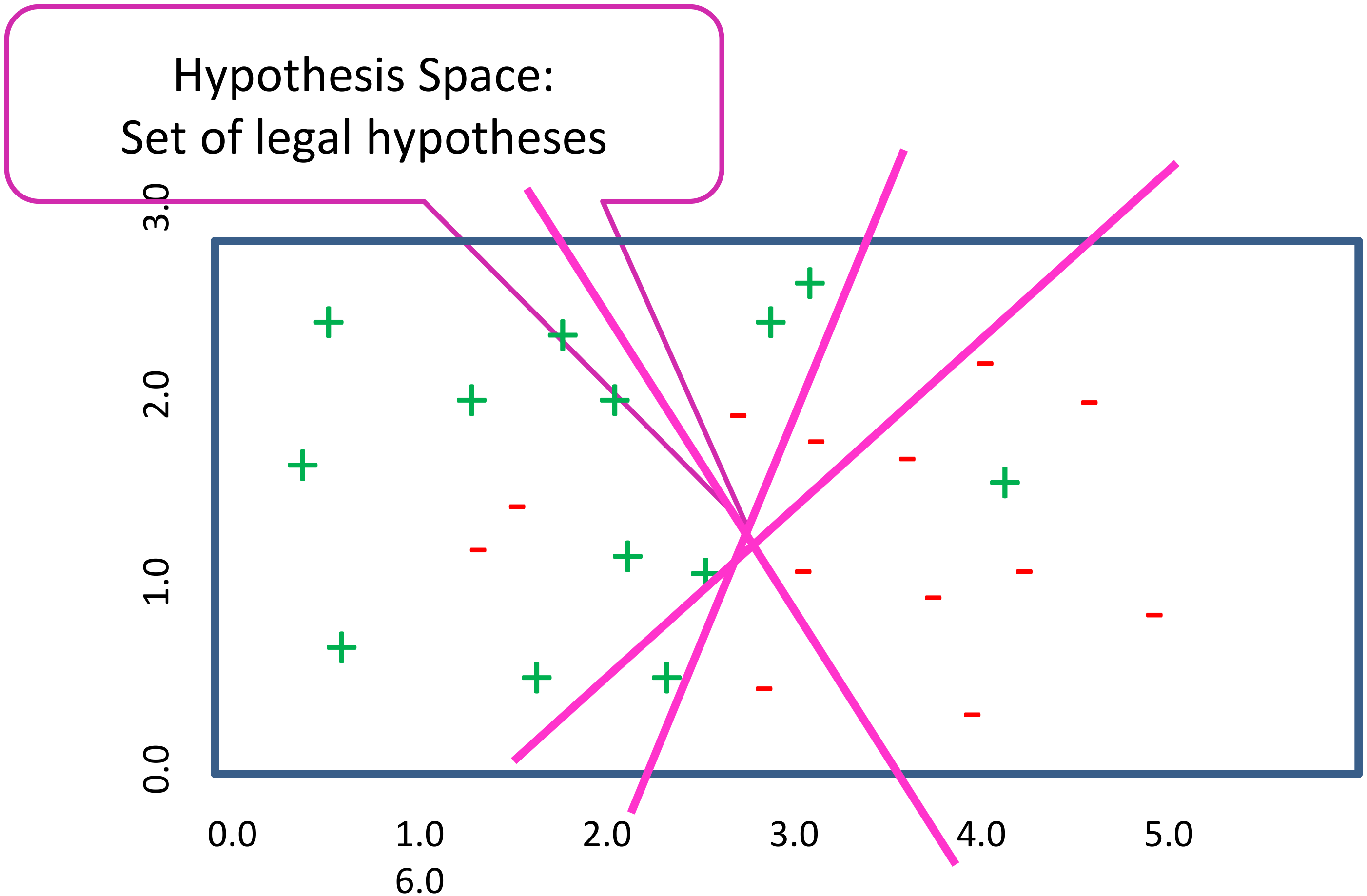
Hypothesis Space

- There could be many possible functions that explain the given training data as shown below:
- It is the set of legal hypothesis
- There could be multiple legal hypothesis or functions set of all such legal hypothesis is called as hypothesis space.
- Eg: class 1: +ve, class 2: -ve

✓ Our objective is to come up with best hypothesis
✓ We denote the hypothesis space by H
✓ Output of learning algorithm will be h where h $\in$ H

✓ One way to think about a supervised machine learning is as a device that explores a "Hypothesis Space“.

Hypothesis Space



Hypothesis Space

Target Function

It's a function which maps every input x to an output y, we denote it by f.

- Our objective is to come up with a hypothesis $h \in H$ that approximates "f" based on the training data.

Input and output of a learning algorithm

- Input - Training set, S
- Output- Hypothesis, h where $h \in H$.

Hypothesis Space

- If there are 2 Boolean input features then there are 2^2 possible instances.
- If there are 3 Boolean input features then there are 2^3 possible instances.
- If there are n Boolean input features then there are 2^n possible instances.

- If there are 2 Boolean input features then there are 2^4 possible Boolean functions
- If there are 3 Boolean input features then there are 2^8 possible Boolean functions.

- **For n variables, how many Boolean functions are possible?**

n variables = 2^{2^n} Boolean functions

- When there are no variables, there are two expressions.

False =0 True =1

Inductive Learning In General

- Inducing a general function from training examples.
- Constructs a hypothesis ***h*** to agree with all the training examples
- A hypothesis is consistent if it agrees (works will) with all training examples.
- A hypothesis is said to be generalized if it correctly predicts the value of ***y*** for new examples.

Inductive learning hypothesis (Rule)

- Any hypothesis used to approximate the target function well over a sufficiently large set of training examples will approximate the target function well over other unobserved examples-
  - If ***h*** works well on sufficiently large set of training example
  - Then it works well on observed data

Inductive Learning Algorithms

Key Computational Approaches

Decision Trees
A popular inductive learning algorithm that utilizes a tree-like model for decisions, providing interpretable outcomes while accommodating both numerical and categorical data.

Neural Networks
Inspired by biological neural networks, these models excel at capturing nonlinear relationships in data, making them suitable for complex tasks such as image and speech recognition.

Support Vector Machines
A robust classification technique that finds optimal hyperplanes to separate data points of different classes, known for their efficacy in high-dimensional spaces.

Applications of Inductive Learning

Real-World Use Cases

Natural Language Processing
Inductive learning techniques are fundamental in NLP tasks such as sentiment analysis and text classification, enabling models to interpret and generate human language effectively.

Image Recognition
Inductive learning enhances image classification and object detection, allowing machines to analyze and understand visual content based on learned features and patterns.

Predictive Analytics
These techniques are instrumental in forecasting future outcomes based on historical data, significantly impacting industries such as finance, marketing, and healthcare.

Challenges in Inductive Learning

Key Obstacles to Effective Induction

Overfitting
Occurs when a model learns patterns that are too specific to the training data, leading to poor generalization to new data, thereby reducing predictive accuracy.

Underfitting
Arises when a model is too simplistic, failing to capture important trends in the data, which results in inadequate performance on both training and test sets.

Bias-Variance Tradeoff
The balance between bias (error due to simplistic assumptions) and variance (error due to excessive complexity) is a critical factor affecting model performance.

Supervised Learning

Given: <x, f(x)> for some unknown function f

Learn: A hypothesis H, that approximates f

Example Applications:

- Disease diagnosis
  x: Properties of patient (e.g., symptoms, lab test results)
  f(x): Predict disease
- Automated steering
  x: Bitmap picture of road in front of car
  f(x): Degrees to turn the steering wheel
- Credit risk assessment
  x: Customer credit history and proposed purchase
  f(x): Approve purchase or not

Key Issues in Machine Learning

- **What are good hypothesis spaces?**
  Which spaces have been useful in practical applications and why?
- **What algorithms can work with these spaces?**
  Are there general design principles for machine learning algorithms?
- **How can we optimize accuracy on future data points?**
  This is sometimes called the "problem of overfitting".
- **How can we have confidence in the results?**
  How much training data is required to find accurate hypotheses? (the *statistical question*)
- **Are some learning problems computationally intractable?**
  (the *computational question*)
- **How can we formulate application problems as machine learning problems?** (the *engineering question*)

Learning = Representation + Evaluation + Optimization

- Combinations of just three elements

Representation	Evaluation	Optimization
Instances	Accuracy	Greedy search
Hyperplanes	Precision/Recall	Branch & bound
Decision trees	Squared error	Gradient descent
Sets of rules	Likelihood	Quasi-Newton
Neural networks	Posterior prob.	Linear progr.
Graphical models	Margin	Quadratic progr.
Etc.	Etc.	Etc.

Inductive BIAS

- As we can see that hypothesis space is very large. It is not possible to look at every hypothesis individually to choose the best hypothesis.
- So we put some restrictions on hypothesis.
- If we restrict the hypothesis, it reflects a bias of the learning algorithm

- **<u>Bias Could be of two types</u>**

1. Restricted bias: Limits the hypothesis space
2. Preference bias: Impose ordering on hypothesis space.

- Example of restriction bias: We may say that we are looking for a linear function or we are looking for 3rd degree polynomial.
- Example of preference bias: We may say that we are considering all possible polynomials but we will prefer a polynomial of lower degree.

Generalization & Error

Coming up with a general function from training examples

- When we do generalization some errors get introduced.
- There are two components of generalization error
  - Bias error
  - Variance error

Generalization & Error

Bias

- This is the error introduced due to simplifying assumptions made by a model
- Simplified assumptions limit the model's capacity to learn.

Low Bias

- Suggests less assumptions about the form of the target function.

High Bias

- Suggests more assumptions about the form of the target function.

Generalization & Error

<u>Variance</u>

- Variance tells that how much a random variable is different from its expected value.
- If the machine learning model performs well with the training dataset, but does not perform well with the test dataset, then variance occurs.

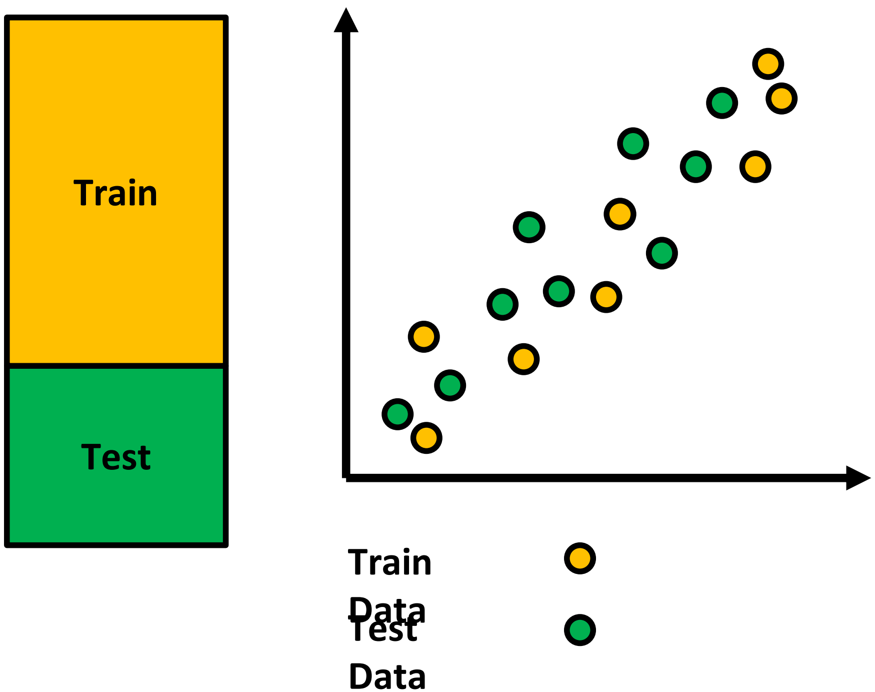
<u>Low variance.</u>

- Suggests small change to the estimated models with Changes to the training dataset.

<u>High variance.</u>

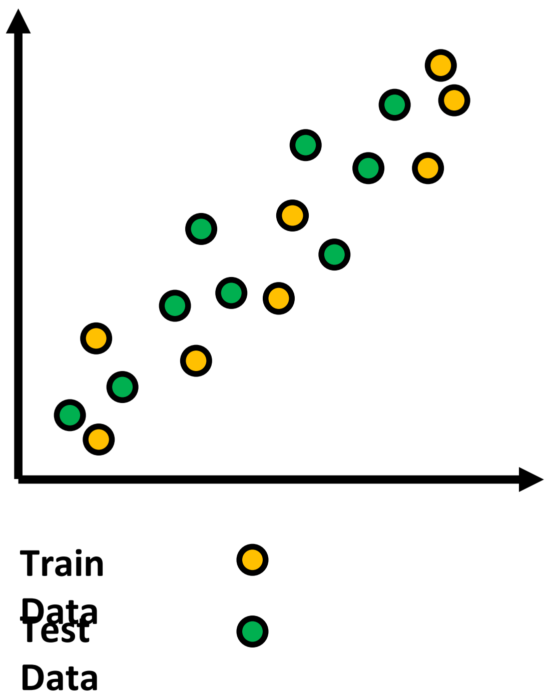
- Suggests large changes, to the estimated models with Changes to the training dataset.

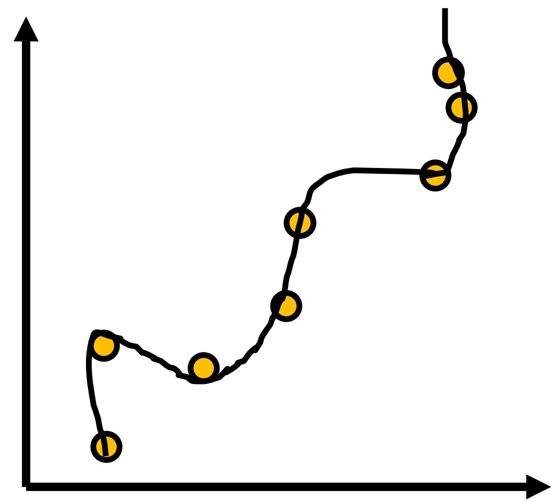
Bias Variance Trade-off



Note: This is a regression problem.
Data is divided into train and test set
Not classes

Bias Variance Trade-off





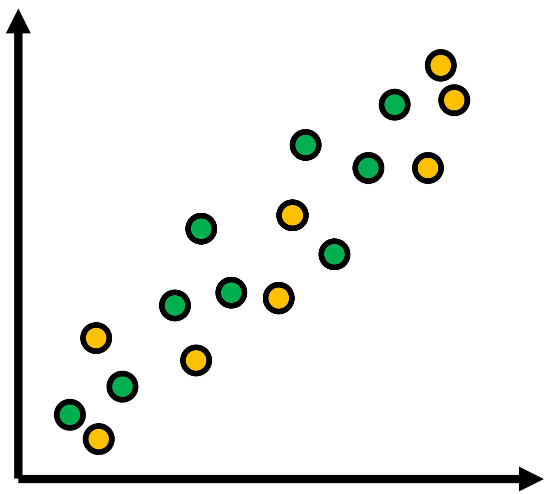
Overfit Model

Complex Model

No error on training data

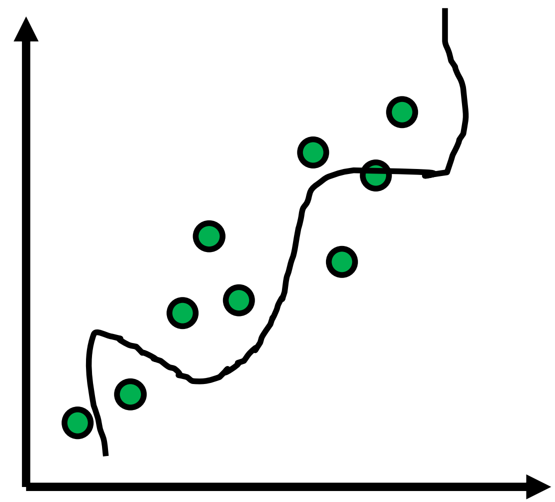
What about test set?

Bias Variance Trade-off



Train Data

Test Data



Overfit Model

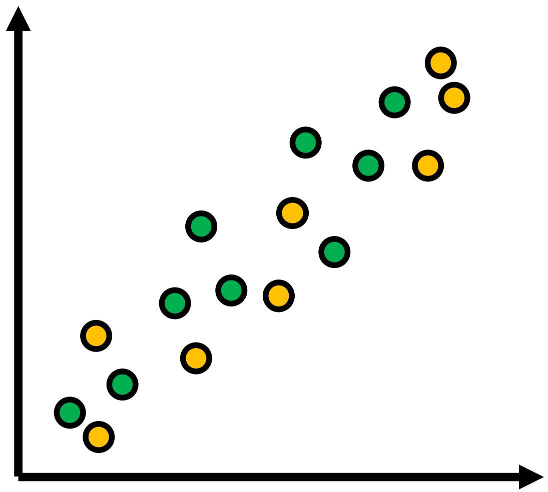
Complex Model

No error on training data

What about test set?

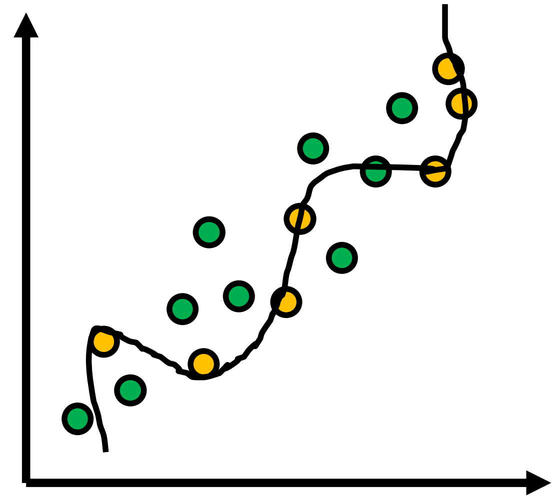
Model performed well on train Data but test error is high

Bias Variance Trade-off



Train Data

Test Data



Overfit Model

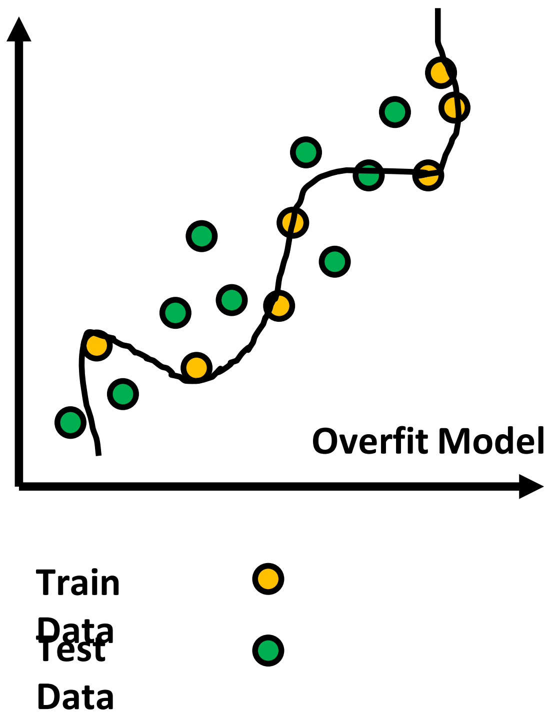
Complex Model

No error on training data

What about test set?

Model performed well on train Data but test error is high

Bias Variance Trade-off



Train error Low

Test Error High

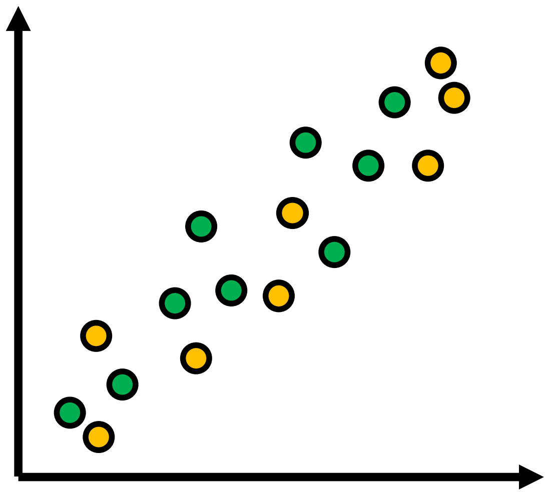
→ High Variance

Error difference in train and test set is more

Hence, we don’t want an Overfitting model

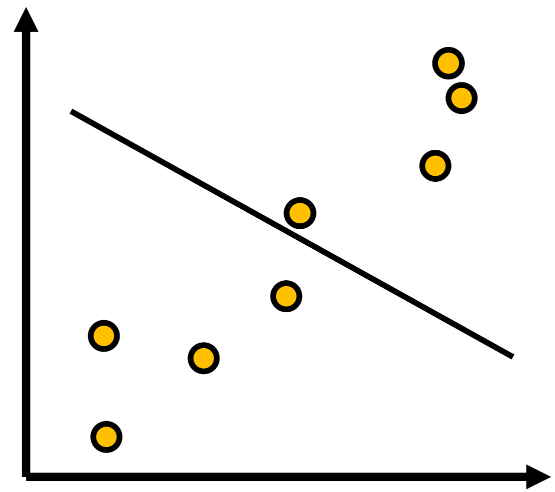
Overfit → High Variance

Bias Variance Trade-off



Train Data

Test Data



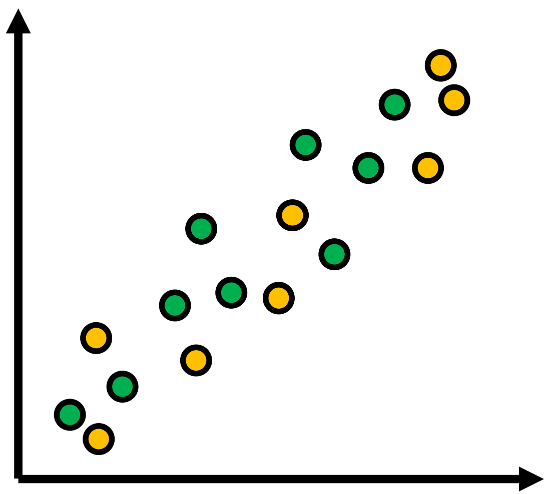
Underfit Model

Simple Model

More error on training data

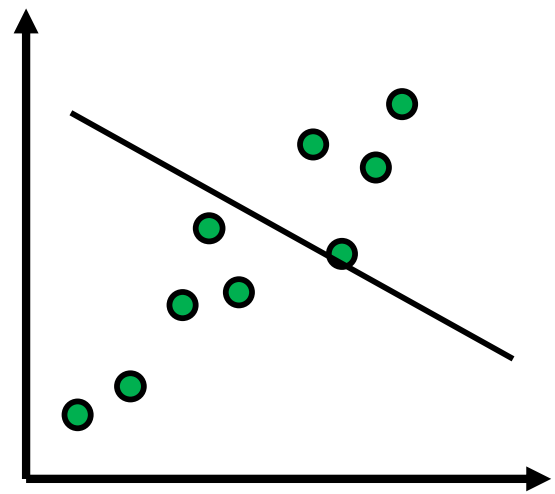
What about test set?

Bias Variance Trade-off



Train Data

Test Data



Underfit Model

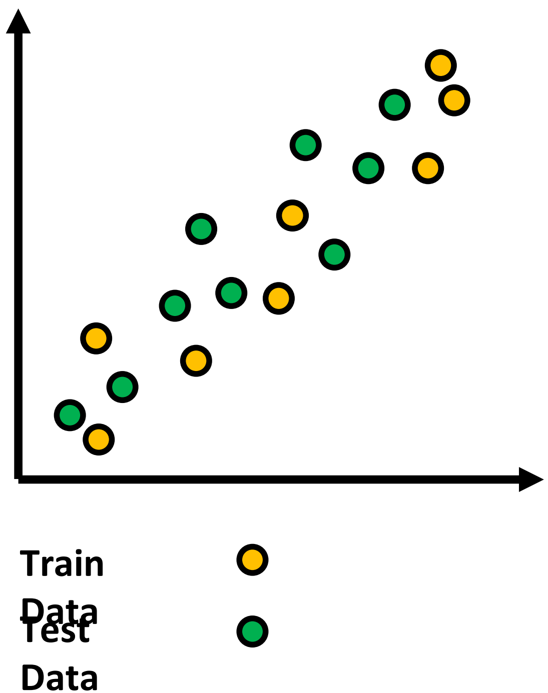
Simple Model

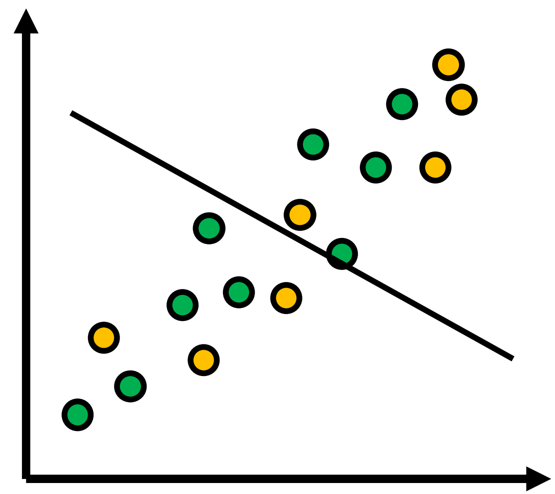
More error on training data

What about test set?

Model is neither performing Well on train data nor on Test

Bias Variance Trade-off





Underfit Model

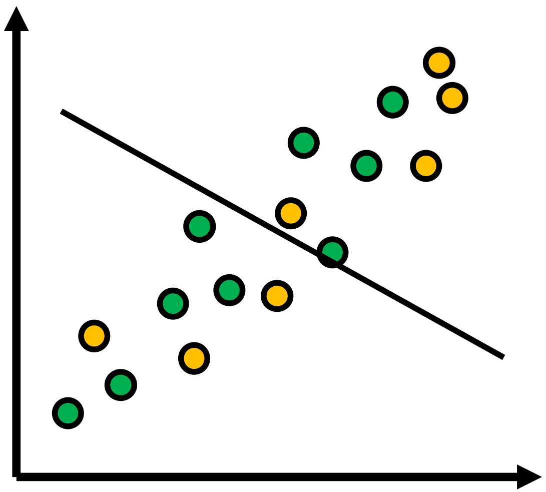
Simple Model

More error on training data

What about test set?

Model is neither performing Well on train data nor on Test

Bias Variance Trade-off



Train Data

Test Data

Train error High

Test Error High

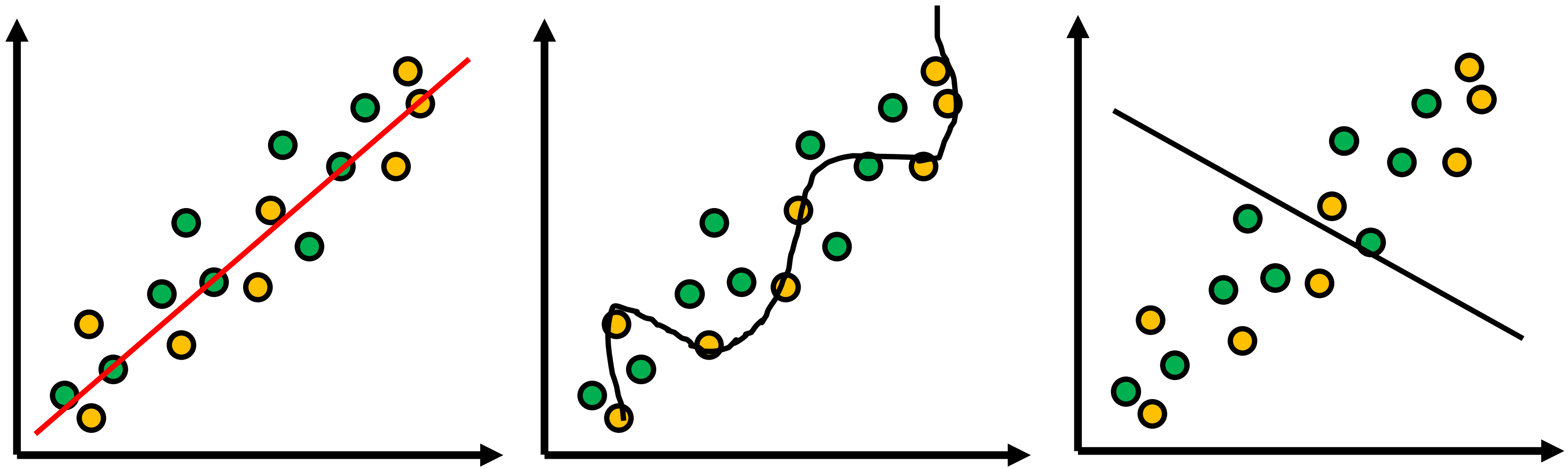
→ **High Bias**

Error difference in train and test set is Less

Hence, we don't want an Underfit model

Underfit → **High Bias**

Bias Variance Trade-off



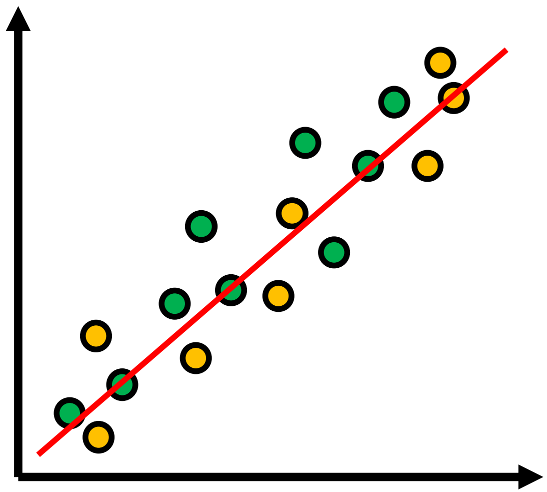
Train Data

Test Data

Overfit Model

Underfit Model

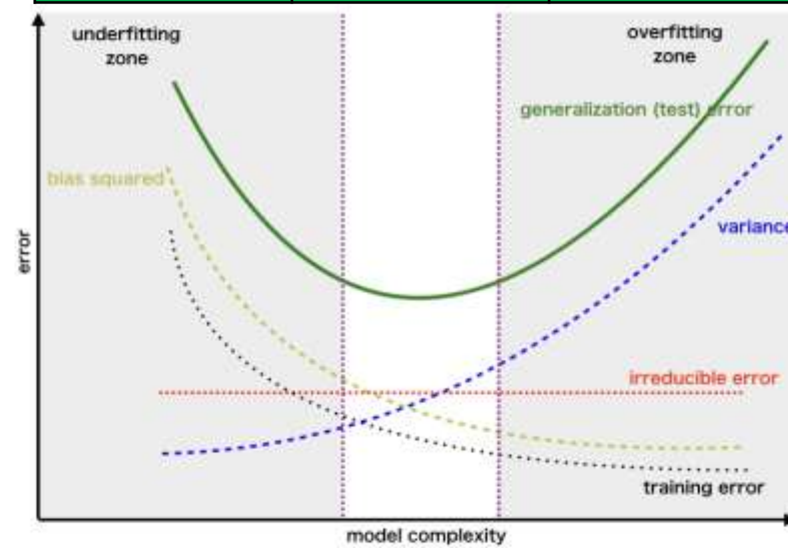
Bias Variance Trade-off



Train Data

Test Data

Train Error	**High**	**Low**
	High Bias	**Low Bias**
Test Error	High	Low
	High Variance	Low Bias



Over-fitting

- Over-fitting & under-fitting are two main errors/problems in the machine learning model, which cause poor performance in Machine Learning.
- Over-fitting occurs when the model fits more data than required, and **it tries to capture each and every data point fed to it**. Hence it starts capturing noise and inaccurate data from the dataset, which degrades the performance of the model.
- An over-fitted model doesn't perform accurately with the test/unseen dataset and can’t generalize well.
- An over-fitted model is said to have low bias and high variance.

How to avoid Overfitting

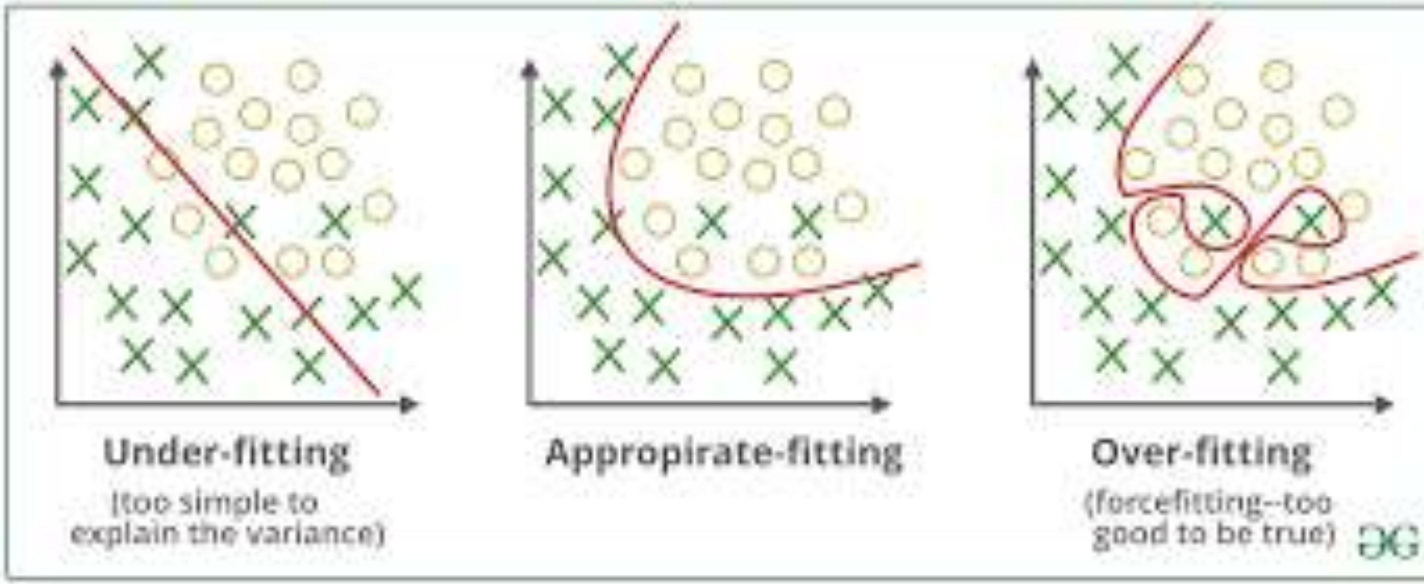
- Using cross-validation
- Using Regularization techniques
- Implementing Ensemble Techniques.
- Picking a less parameterized/complex model
- Training the model with sufficient data
- Removing features
- Early stopping the training

Under-fitting

- Model cannot create a mapping between the input and the target variable
- Under-observing features leads to a higher error in the training and unseen data samples.
- Under-fitting becomes obvious when the model is too simple and cannot create a relationship between the input and the output.

How to avoid Under-fitting

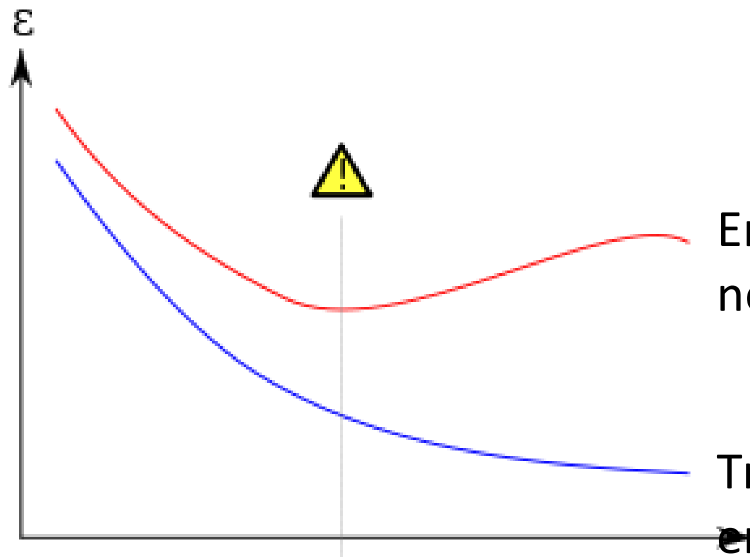
- Preprocessing the data to reduce noise in data
- More training to the model
- Increasing the number of features in the dataset
- Increasing the model complexity
- Increasing the training time of the model to get better results.


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

Over-fitting

Over-fitting during training

Model error



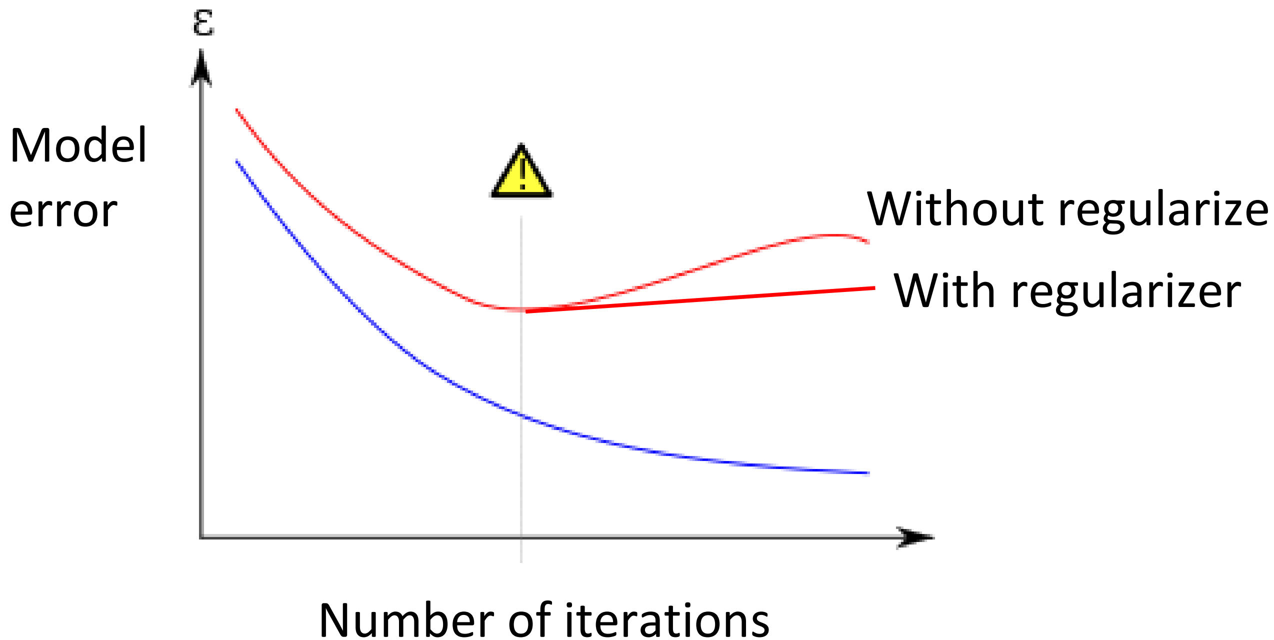
Error on new data

Training error

Number of iterations

Regularization and Over-fitting

Adding a regularizer:



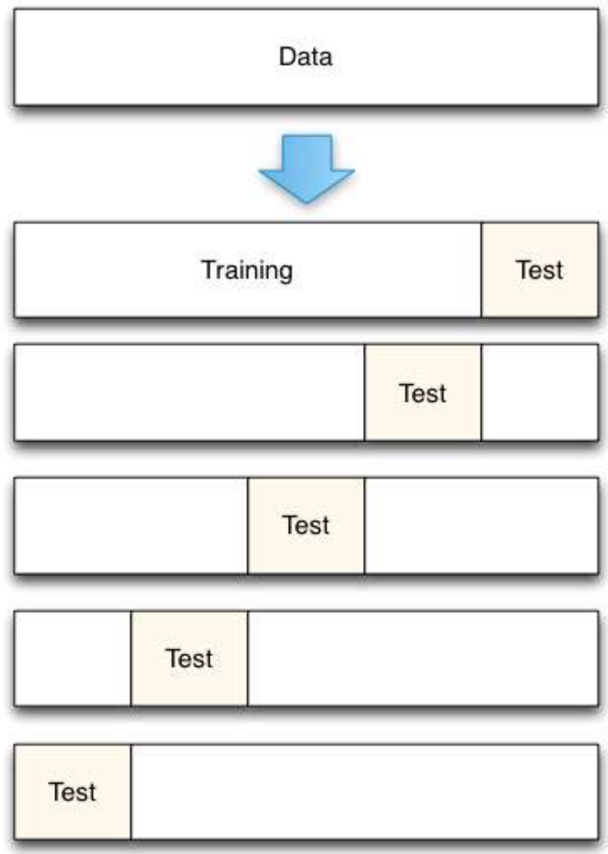
Cross-Validation

- Cross-validation involves **partitioning** your data into distinct **training** and **test** subsets.
- The test set **should never** be used to **train** the model.
- The test set is then used to **evaluate** the model after training.

K-fold Cross-Validation

- To get more accurate estimates of performance you can do this k times.
- Break the data into k equal-sized subsets A_i
- For each i in 1,…,k do:
  - Train a model on all the other folds $A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_k$
  - Test the model on A_i
- Compute the **average performance** of the k runs

5-fold Cross-Validation



Occam's Razer

- Classical example of Bias
- It says the simplest consistent hypotheses about the target function is actually the best.

Learning as a search

- Learning can be viewed as the task of searching through a large space of hypothesis implicitly defined the hypothesis representation.
- The goal of this search is to find the hypothesis that best fits the training examples and generalize well to unseen data.