Introduction to Machine Learning

Machine Learning (Core-11) CS305

Overview of Course

1. Introduction
2. Linear Regression and Decision Trees
3. Instance based learning
   Feature Selection
4. Probability and Bayes Learning
5. Support Vector Machines
6. Neural Network
7. Introduction to Computational Learning Theory
8. Clustering

B. Tech. III (CSE) Semester – V
MACHINE LEARNING (CORE - 11)
CS305

	L	T	P	Credit
Scheme	3	0	2	04

1. Course Outcomes (COs): At the end of the course, students will be able to	
CO1	acquire knowledge of pattern recognition, regression, classification, clustering algorithms and statistics.
CO2	apply different classification, regression, machine learning algorithms and modelling.
CO3	analyze the data patterns and modelling for applying the learning algorithms.
CO4	evaluate the performance of an algorithm and comparison of different learning techniques.
CO5	design solution for real life problems like biometric recognition, natural language processing and its related applications using various tools and techniques of machine learning.

2. Syllabus

- **INTRODUCTION** **(09 Hours)**

  Pattern Representation, Concept of Pattern Recognition and Classification, Feature Extraction, Feature Selection, Basics of Probability, Bayes Decision Theory, Maximum-Likelihood and Bayesian Parameter Estimation, Error Probabilities, Learning of Patterns, Modelling, Regression, Discriminant Functions, Linear Discriminant Functions, Decision Surface, Learning Theory, Fisher Discriminant Analysis.

- **SUPERVISED LEARNING ALGORITHMS** **(09 Hours)**

  Linear Regression, Gradient Descent, Support Vector Machines, Artificial Neural, Networks, Decision Trees, ML and MAP Estimates, K-Nearest Neighbor, Naive Bayes, Bayesian Networks, Classification, Overfitting, Regularization, Multilayer Networks, Back-propagation, Bayes Classification, Nearest Neighbor Classification, Cross Validation and Attribute Selection, K Means Clustering, Agglomerative Hierarchical Clustering.

- **UNSUPERVISED LEARNING ALGORITHMS** **(09 Hours)**

  K-Means Clustering, Gaussian Mixture Models, Learning with Partially Observable Data, Expectation Maximization Approach. Dimensionality Reduction, Principal Component Analysis, Model Selection and Feature Selection.

- **TRANSFORM DOMAIN PATTERN ANALYSIS** **(06 Hours)**

  Signal Transformation, Frequency Domain Representation of Signal, Feature Extraction and

Analysis, Multiresolution Representation, Wavelet Transform, Discrete Cosine Transform.

- **APPLICATIONS** **(09 Hours)**

  Signal Processing Application, Image Processing, Biometric Recognition, Face and Speech Recognition, Information Retrieval, Natural Language Processing.

(Total Contact Time: 42 Hours + 28 Hours= 70 Hours)

---

3. Practical:

1. Implement classification and regression techniques.
2. Implement clustering and statistical modeling methods.
3. Implement various dimensionality reduction techniques.
4. Implement neural networks and non-parametric techniques.
5. Implement mini-project based on machine learning approaches.

4. Books Recommended:

1. Geoff Dougherty, "Pattern Recognition and Classification: An Introduction", 1st Edition, Springer, 2013.
2. Theodoridis and K.Koutroumbas, "Pattern Recognition", 4th Ed., Academic Press, 2009.
3. Christopher M. Bishop, "Pattern Recognition and Machine Learning", 1st Edition, Springer, 2006.
4. Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification", 2nd Edition, Wiley, 2001.
5. K. Fukunaga, "Introduction to Statistical Pattern Recognition", 2nd Edition, Academic Press, 2000.

ADDITIONAL REFERENCE BOOKS

1. Ranjjan Shinghal, "Pattern Recognition Techniques and Application", 1st Edition, Oxford university press, 2006.

Books and references

- Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
- Introduction to Machine Learning Edition 2, by Ethem Alpaydin
- Pattern Recognition and Machine Learning, by Christopher Bishop (optional)
- NPTEL Video Lectures
  - Prof. S. Sarkar, IIT Kharagpur
  - Prof. Balaraman Ravindran, IIT Madras

Introduction

- basics of the course,
- discuss the brief history of machine learning
- discuss what learning is about
- some simple applications of machine learning
- different types of learning

Machine Learning History

- 1950s:
  - Samuel's checker-playing program
- 1960s:
  - Neural network: Rosenblatt's perceptron
  - Pattern Recognition
  - Minsky & Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Expert systems and knowledge acquisition bottleneck
  - Quinlan's ID3
  - Natural language processing (symbolic)

Machine Learning History (Contd..)

- 1980s:
  - Advanced decision tree and rule learning
  - Learning and planning and problem solving
  - Resurgence of neural network
  - Valiant's PAC learning theory
  - Focus on experimental methodology
- 90's ML and Statistics
  - Support Vector Machines
  - Data Mining
  - Adaptive agents and web applications
  - Text learning
  - Reinforcement learning
  - Ensembles
  - Bayes Net learning

- 1994: Self-driving car road test
- 1997: Deep Blue beats Gary Kasparov

Machine Learning History (Contd..)

- Popularity of this field in recent time and the reasons behind that
  - New software/ algorithms
    - Neural networks
    - Deep learning
  - New hardware
    - GPU's
  - Cloud Enabled
  - Availability of Big Data

- 2009: Google builds self driving car
- 2011: Watson wins Jeopardy
- 2014: Human vision surpassed by ML systems

Machine Learning History (Contd..)

- A machine that is intellectually capable as much as humans has always fired the imagination of writers and also the early computer scientist who were excited about artificial intelligence and machine learning, but the first machine learning system was developed in the 1950s.
- In 1952, Arthur Samuel was at IBM. He developed a program for playing Checkers. The program was able to observe positions at the game and **learn a model that gives better moves for the machine player.**
- **Samuel coined the term machine learning and he defined learning as a field of study that gives computers the ability without being explicitly programmed.**

Machine Learning History (Contd..)

- In 1957, Rosenblatt proposed the perceptron. Perceptron is the simple neural network unit; it was a very exciting discovery at that time.
- But after 3 years, came up with the delta learning rule that is used for learning perceptron. It was used as a procedure for training perceptron. It is also known as the least square problem. The combination of these ideas created a good linear classifier.
- However, the work along these lines suffered a setback when Minsky in 1969 came up with the limitations of perceptron. He showed, that the problem could not be represented by perceptron.

Machine Learning History (Contd..)

- J.R. Quinlan, in 1986 came up with decision tree learning, specifically the ID3 algorithm.
- After ID3 many alternatives or improvement ID3 were developed such as cart, regression, trees and it is still one of the very popular topics in machine learning.
- At the same time, there was a resurgence of neural network. The multilayer perceptron was suggested by in 1981 and neural network specific back propagation algorithm was developed.
- Back propagation is the key ingredient of today's neural network architectures.

Machine Learning History (Contd..)

- In the 90s, machine learning embraced statistics to a large extent. It was during this time, that support vector machines were proposed by Vapnik and Cortes in 1995.
- Then, another strong machine learning model was proposed by Freund and Schapire in 1997, which was part of what we called ensembles to create a strong classifier from an ensemble of weak classifiers.
- The kernalized version of SVM was proposed near 2000s, which was able to exploit the knowledge of convex optimization, generalization and kernels.

Machine Learning History (Contd..)

- This rise of neural network began roughly in 2005 with the conjunction of many different discoveries for people by Hinton, LeCun, Bengio, Andrew and other researchers.

Machine Learning History (Contd..)

- Certain Applications:
  - certain applications where machine learning has come to the public forefront.
  - In 1994, the first self driving car made a road test;
  - in 1997, Deep Blue beat the world champion Gary Kasparov in the game of chess;
  - in 2009 we have Google building self driving cars;
  - in 2011, Watson, from IBM, won the popular game of Jeopardy;
  - In 2014, we see human vision surpassed by ML systems.
  - In 2014-15, 2015 we find, that machine translation systems driven by neural networks are very good and they are better than the other statistical machine translation systems where certain concepts and certain technology, which are making headlines.

Machine Learning History (Contd..)

- Now, in machine learning we have GPU’s, which are enabling the use of machine learning and deep neural networks.
- There is the cloud, there is availability of big data and the field of machine learning is very exciting now.

How machine learning solutions differs from programmatic solution?

How machine learning solutions differs from programmatic solution?

- When you have a program or algorithm to solve a problem, this is how you use the computer. This is your computer and you write a program. **The program takes data as the input and the program produces output.**
- On the other hand, when we are **using machine learning** and you have the computer, you are feeding the data, input as well as examples of output.
- So, you are putting examples of input, output data and you are getting a program or a model with which you can solve subsequent tasks. **So, this is what learning is about.**

formal definition of machine learning

- **learning is the ability to improve one's behavior with experience.** So, it is about building computer systems, that **automatically improve with experience** and we have to discuss what are the fundamental laws that governed the learning processes.
- **Machine learning** explores algorithms that learn from data, build models from data and this model can be used for different tasks.
- **For example**, model can be used for prediction, decision making or solving tasks.

formal definition of machine learning as given by Tom Mitchell

- "… said to learn from experience with respect to some class of tasks, and a performance measure P, if [the learner' s] performance at tasks in the class, as measured by P, improves with experience."

Tom Mitchell 1997.

formal definition (contd…)

- **we have to define learning with respect to a specific class of tasks**, it could be ***answering exams*** in a particular subject or it could be ***diagnosing patients*** of a specific illness.
- So we have to be very careful about defining the set of tasks on which we are going to define this learning.
- the second thing we need is of a performance measure P, if you want to be more clear about **measuring whether learning is happening or not** you first need to define some kind of a performance criteria.

formal definition (contd...)

- **example** (**performance criterion)**
  - if you talk about answering questions in an exam your **performance criterion** could very well be the number of marks that you get
  - or if you talk about diagnosing illness then your performance measure would be the number of patients who did not have adverse reaction to the drugs you gave them
  - there could be variety of ways of defining performance measures depending on what you are looking for **and the third important component here is experience**.

formal definition (contd...)

- **So with experience the performance has to improve and so what we mean by experience here** in the case of writing exams it could be writing more exams so the more the number of exams you write the better you write it, the better you get it.
- It could be a patient's in the case of diagnosing illnesses like the more patients that you look at the better you become at diagnosing illness.

formal definition (contd…)

- So these are the three components so **you need a class of tasks, you need a performance measure and you need some well-defined experience** so this kind of learning where you are learning to improve your performance based on experience is known as inductive learning.

machine learning paradigms

- Supervised Learning
  - Learn an input and output map
    - Classification: categorical output
    - Regression: continuous output
- Unsupervised Learning
  - Discover patterns in the data
    - Clustering: cohesive grouping
    - Association: frequent cooccurance
- Reinforcement Learning
  - Learning Control

machine learning paradigms (cont..)

- In **supervised learning** what you essentially do is on a mapping from this input ***(description of the patient)*** to the required output ***(patient has a certain disease or not)***.
- If the output that you are looking for happens to be a categorical output like whether he has a disease or does not have a disease or whether the answer is true or false then the ***supervised learning problem is called the classification problem***
- and if the output happens to be a continuous value like,
  - what is the expected rainfall tomorrow
  - how long will this product last before it fails
  - so these kinds of problems they would be called as **regression problems (output is a continuous value).**

machine learning paradigms (cont..)

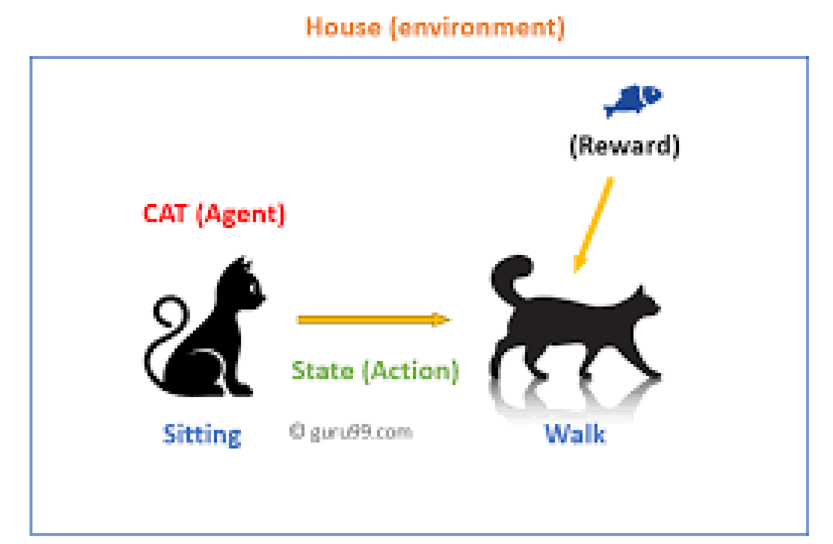
- **unsupervised learning** – there is no real desired output that we are looking for , we are more interested in finding patterns in the data.
- So clustering is one task of unsupervised learning where you are interested in finding cohesive groups among the input pattern,
  - for example I might be looking at customers who come to my shop **and I want to figure out if there are categories of customers like**, so may be
    - college students could be one category and
    - IT professionals could be another category and so on
- So when I'm looking at these kinds of grouping in my data, so I would call that a clustering task.

machine learning paradigms (cont..)

- the other **popular unsupervised learning paradigm is known as the Association rule mining** or **frequent pattern mining**
  - Where you are interested in finding a frequent co-occurrence of items in the data that is given to you so whenever A comes to my shop, B also comes to my shop. So those kinds of co-occurrence, where I see A then there is likely very likely that B is also in my shop
  - We can learn these kinds of associations between data.

machine learning paradigms (cont..)

- Third form of learning which is called **reinforcement learning** it is neither supervised nor unsupervised in nature and typically these are problems where you are learning to control the behavior of a system.
- we can say that "Reinforcement learning is **a type of machine learning method where an intelligent agent (computer program) interacts with the environment and learns to act within that**." How a Robotic dog learns the movement of his arms is an example of Reinforcement learning.



Machine Learning Tasks

Task	Measure
• Classification	error
• Regression	error
• Clustering	scatter/purity
• Associations	support/confidence
• Reinforcement Learning	cost/reward

Machine Learning Tasks

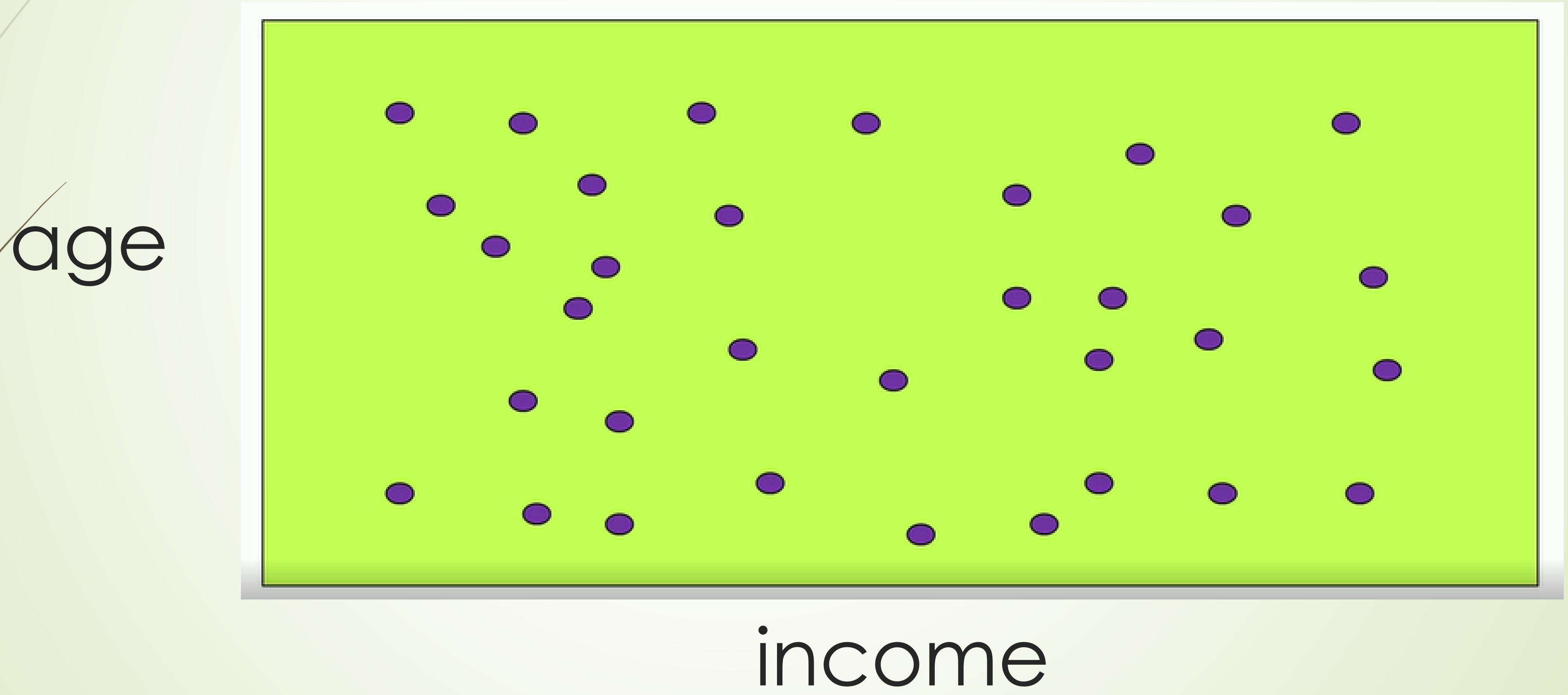
- So we need to have **some kind of a performance measure**,
- looking at classification **the performance measure is going to be classification error.**
- it's how many of the items or how many of the patients did I get incorrect
  - so how many of them who are not having the disease did I predict they had the disease
  - and how many of them that had the disease that I missed.

Challenges

- How good is a model?
- How do I choose a model?
- Do I have enough data?
- Is the data of sufficient quality?
  - Errors in data. Ex: Age=225; noise in low resolution images
  - Missing Values
- How confident can I be of the results?
- Am I describing the data correctly?
  - Are Age and Income enough? Should I look at Gender also?
  - How should I represent age? As a number, or as young, middle age, old?

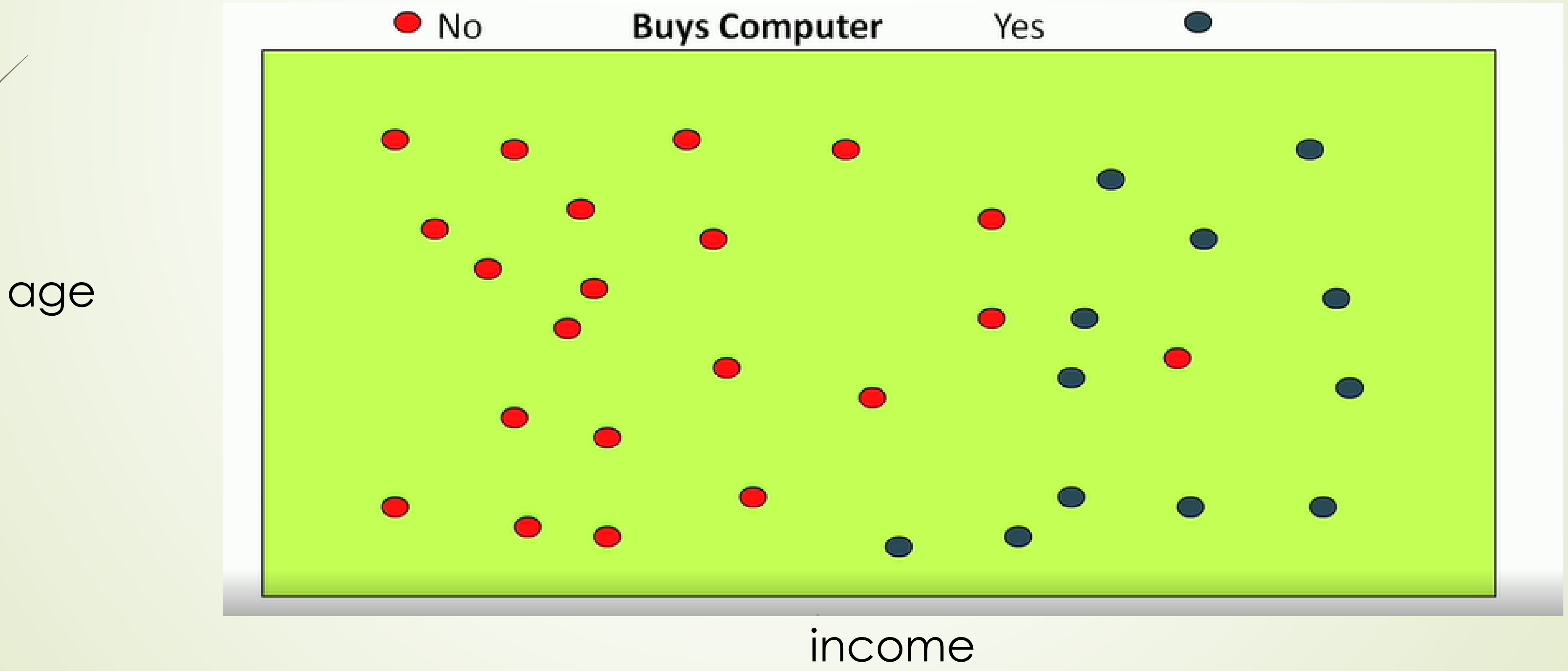
Introduction to Supervised Learning

Training data



A customer database describing by two attributes, age and income.

- You have each customer that comes to your shop and you know the age of the customers and the income level of the customers.
- And **goal is to predict** whether the customer will **buy a computer** or **not buy a computer.**
- So you have this kind of labeled data that is given for building a classifier.
- classification where **the output is a discrete value in this case it is yes or no**.



Labelled Training Data

Supervised Learning (Contd….)

- And so now the goal is to **come up with a function**,
  - come up with a mapping that will take the age and income as the input and it will give you an output
    - that says the person will buy the computer or not buy the computer.
- So there are many different ways in which you can create this function

Applications

- Credit Card fraud detection
  - Valid transaction or not
- Sentiment Analysis
  - Opinion mining; buzz analysis; etc.
- Churn prediction
  - Potential churner or not
- Medical diagnoses
  - Risk analysis

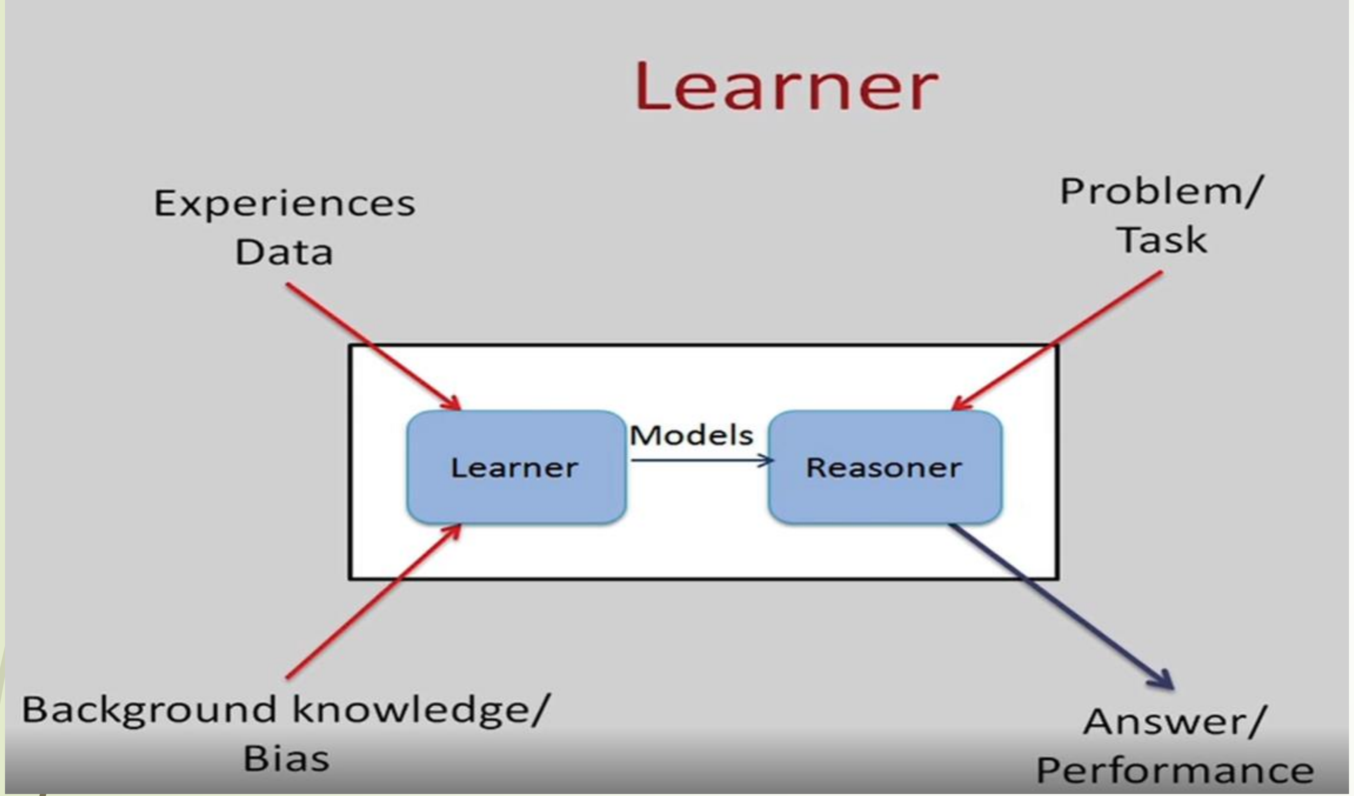
Applications

- **a fraud detection**, so we have some data where the input is a set of transactions made by a user and then you can flag each transaction as a valid transaction or not.
- **sentiment analysis** is called as opinion mining , where if I give you a piece of text or a review written about a product or a movie and then you tell me whether the movies review is positive or whether is negative and what are the negative points that people are mentioning about and so on this again a classification task.

Applications

- **churn prediction** where you are going to say whether a customer who is in the system is likely to leave your system or is going to continue using your product or using your service for a longer period of time,
  - So when a person leaves your services you call the person churner and you can label what the person is churner or not.

Learning System



- Learner takes input as experience data and background knowledge
- Learner builds models
- Models can be used by Reasoner

Learning System(contd..)

Set of steps:

- choose the training data
- choose the target function (that is to be learned)
  - For example, if you are trying to write a machine learning system to play the game of checkers, **the target function would be**, given a board position, what move to take. And then, we want to have the class of function
- choose how to represent the target function
  - what type of function we will use, whether we will use a linear function or some other representation.
- choose a learning algorithm to infer the target function
  - learning algorithm will explore the possible function parameters so that based on the training experience it can come up with the best function given its computational limitations.
  - So, what is very important in the designing of a learning algorithm is how to represent the target function.

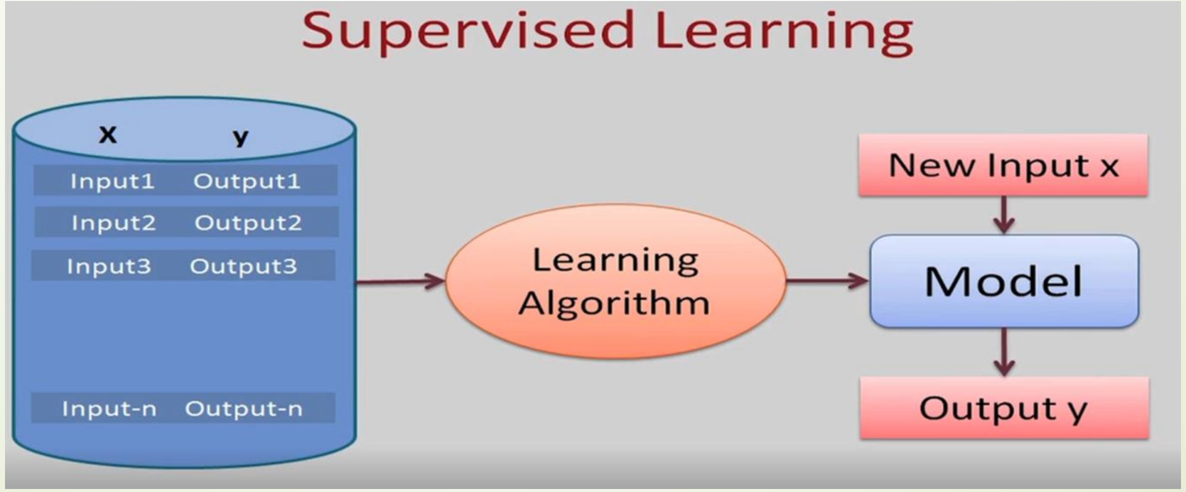
Different Types of Learning

Broad types of machine learning

- Supervised Learning
  - X,y (pre-classified training examples)
  - Given an observation x, what is the best label for y?
- Unsupervised learning
  - X
  - Given a set of x's, cluster or summarize them
- Reinforcement Learning
  - Determine what to do based on rewards and punishments.

Supervised Learning

- In supervised learning what we have? **We have data comprise of input) and the corresponding output.**
- **For every data instance we can have the input x and the corresponding output y.**
- And, from this the **machine learning system will build a model so that given a new observation x will try to find out what is the corresponding y.**
- This called supervised learning because for every instance we tell what is the output. So, this is called labeled data.



Supervised Learning (contd...)

Given:

- a set of input features $X_1, \ldots, X_n$
- A target feature Y
- a set of training examples where the values for the input features and the target features are given for each example
- a new example, where only the values for the input features are given

Predict the values for the target features for the new example.

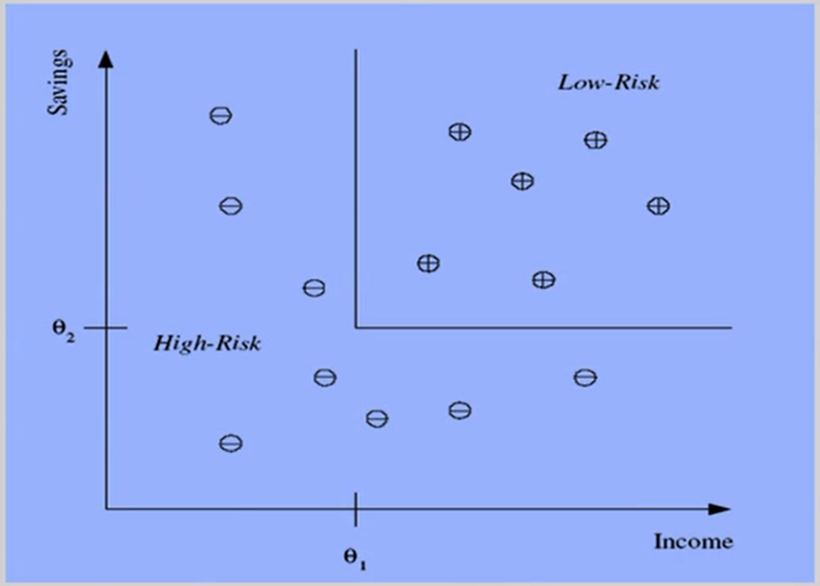
- classification when Y is discrete
- regression when **Y** is continuous

Supervised Learning (contd...)

Classification

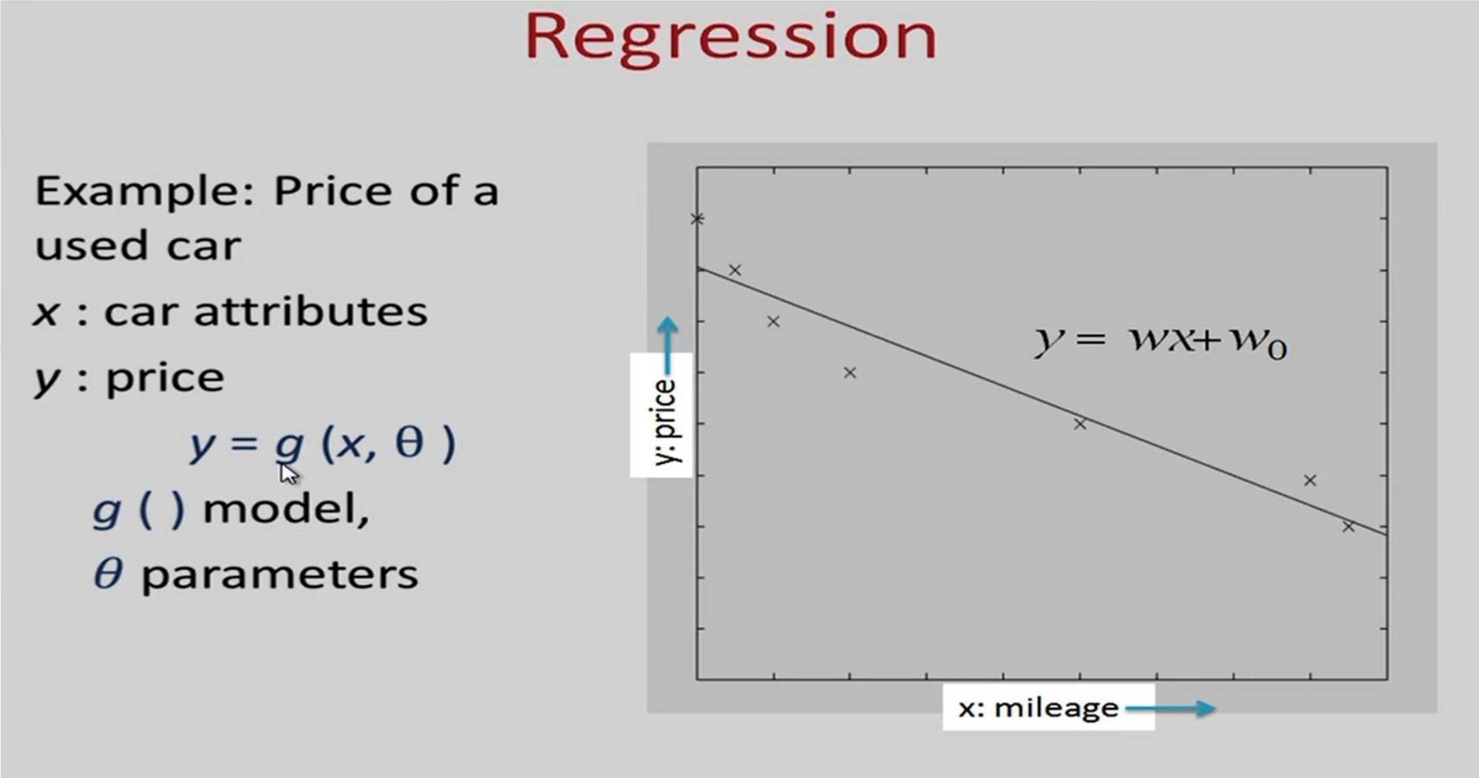
Example: Credit scoring

Differentiating between low-risk and high-risk customers from their *income* and *savings*



Discriminant: IF *income* > θ_1 AND *savings* > θ_2 THEN low-risk ELSE high-risk

- **Describing each input in terms of two features; income and savings**. We look at the income and the savings of a person and we want to predict whether this is a **high risk person or a low risk person.**
- In figure, **the high risk persons are labeled with minus and the low risk persons labeled with plus**.
- Now given this data you want to come up with a classifier which will predict whether that person is high risk or low risk.



- Regression problem: You want to find out the price of a used car and you use certain attributes of the car to predict its price. For example, let us say that **you have only looking at the mileage of the car and based on the mileage you have predicting the price.**
- And these different points they correspond to a particular car show what is the mileage and what is the corresponding price.
- **So given the mileage and price of several cars you have the data points and you can come up with the function so that given a new car whose mileage is given you can predict the price.**
- In regression you come up with a function which takes the input instance and the parameters of the model.

Supervised Learning (contd...)

Features

- Often, the individual observations are analyzed into a set of quantifiable properties which are called features. May be
  - categorical (e.g. "A", "B", "AB" or "O", for blood type)
  - ordinal (e.g. "large", "medium" or "small")
  - integer-valued (e.g. the number of words in a text)
  - real-valued (e.g. height)

Supervised Learning (contd…)

Example Data

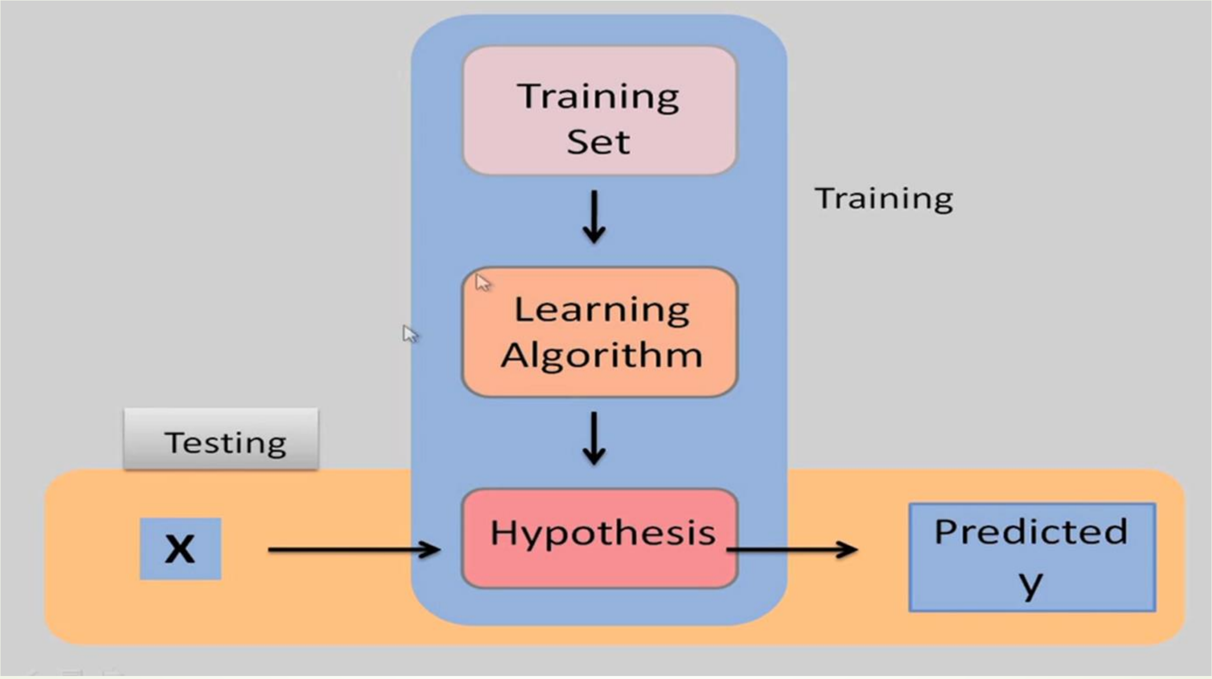
Training Examples:

	Action	Author	Thread	Length	Where
e1	skips	known	new	long	Home
e2	reads	unknown	new	short	Work
e3	skips	unknown	old	long	Work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

New Examples:

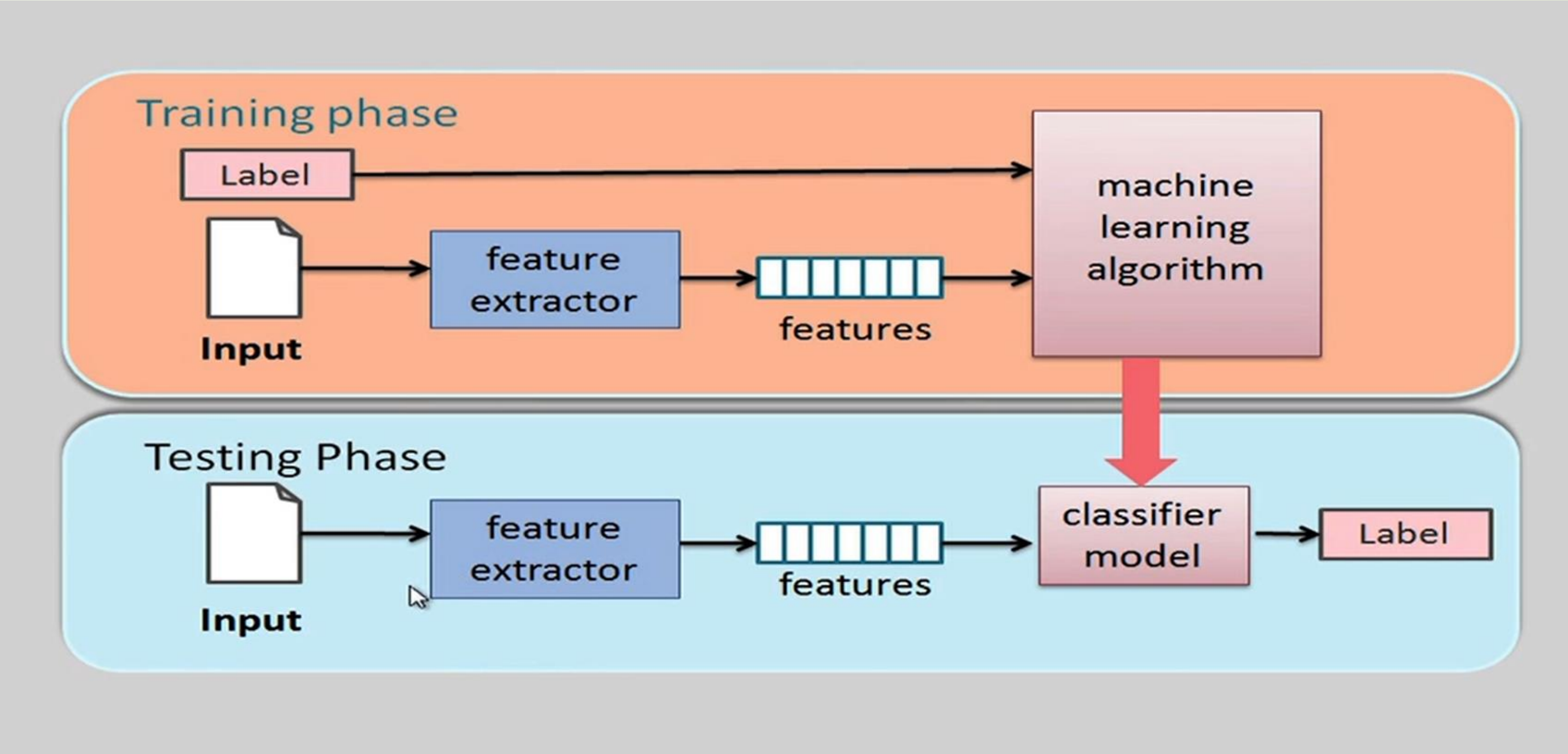
e7	???	known	new	short	work
e8	???	unknown	new	short	work

Schematic diagram in supervised learning



- You have the training set, the learning algorithm uses the training set to come up with a model or hypothesis.
- And in the testing phase given a new instance you use the hypothesis to predict the value of y.

Schematic diagram in supervised learning



Classification Learning

- Task *T:*
  - input: a set of *instances* $d_1,\ldots,d_n$
    - an instance has a set of *features*
    - we can represent an instance as a vector $\mathbf{d}=<x_1,\ldots,x_n>$
  - output: a set of *predictions* $\hat{y}_1,\ldots,\hat{y}_n$
    - one of a fixed set of constant values:
      - *{+1,-1}* or *{cancer, healthy}*, or *{rose, hibiscus, jasmine, …}*, or …
- Performance metric *P:*
- Experience *E:*

Classification Learning

Task	Instance	Labels
medical diagnosis	**patient record:** blood pressure diastolic, blood pressure systolic, age, sex (0 or 1), BMI, cholesterol	{-1,+1} = low, high risk of heart disease
finding company names in text	**a word in context:** capitalized (0,1), word-after-this-equals-Inc, bigram-before-this-equals-acquired-by, …	{first,later,outside} = first word in name, second or later word in name, not in a name
brain-human-interface	**brain state**: neural activity over the last 100ms of 96 neurons	{n,s,e,w,ne,se,nw,sw} = direction you intend to move the cursor
image recognition	**image:** 1920*1080 pixels, each with a code for color	{0,1} = no house, house

Classification Learning

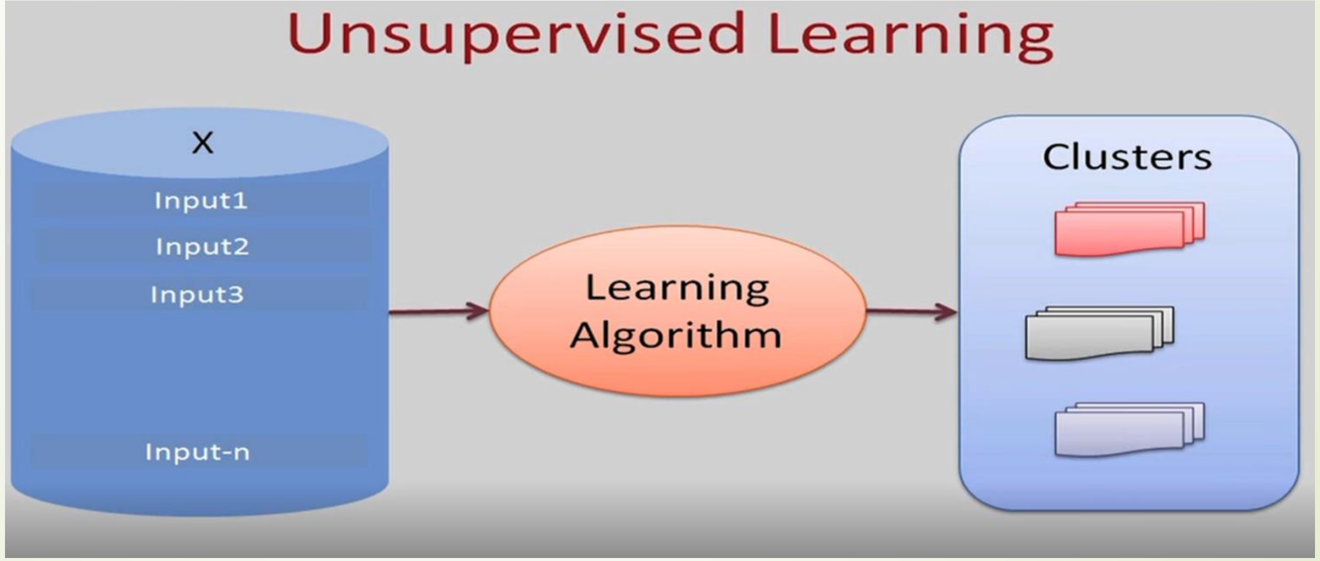
- Task *T:*
  - input: a set of *instances* $d_1,\ldots,d_n$
  - output: a set of *predictions* $\hat{y}_1,\ldots,\hat{y}_n$
- Performance metric *P:*
  - Prob (wrong prediction) on examples from *D*
- Experience *E:*
  - a set of *labeled examples (x,y)* where *y* is the true label for *x*
  - ideally, examples should be *sampled* from some fixed distribution *D*

we care about performance on the *distribution*, not the *training data*

- The performance metric is what is the probability of wrong prediction
- The experience is a set of labeled examples where y are the true labels for x for those training examples we have the ground truth data.

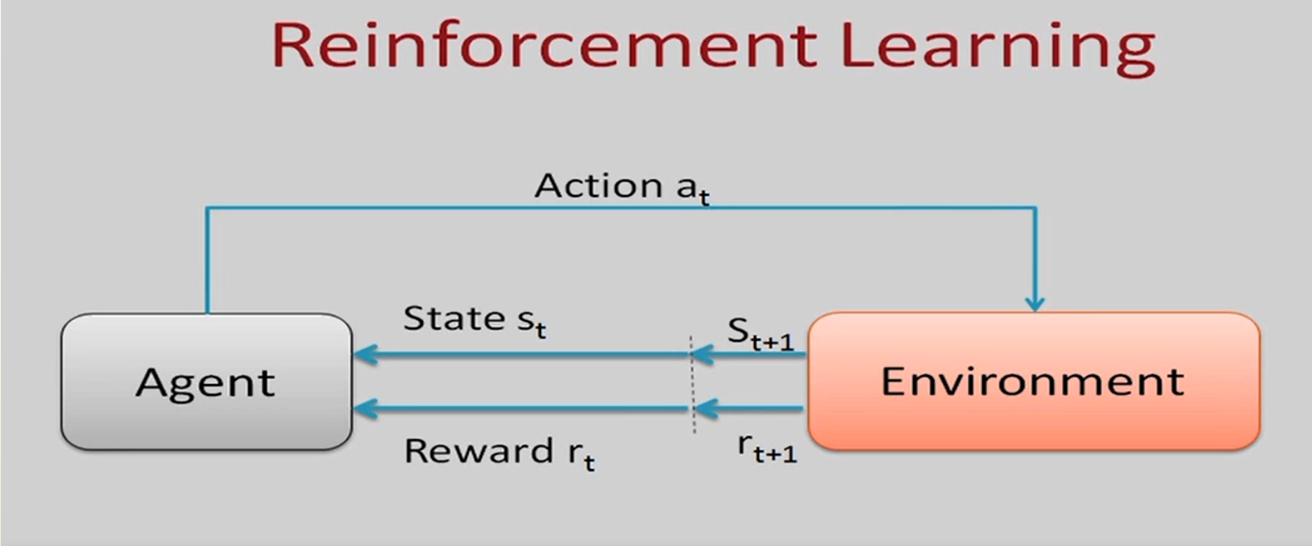
Unsupervised Learning

- In unsupervised learning you are only given x, there is no label to the data. And given the different data points you may like to summarize them or find some patterns in them.

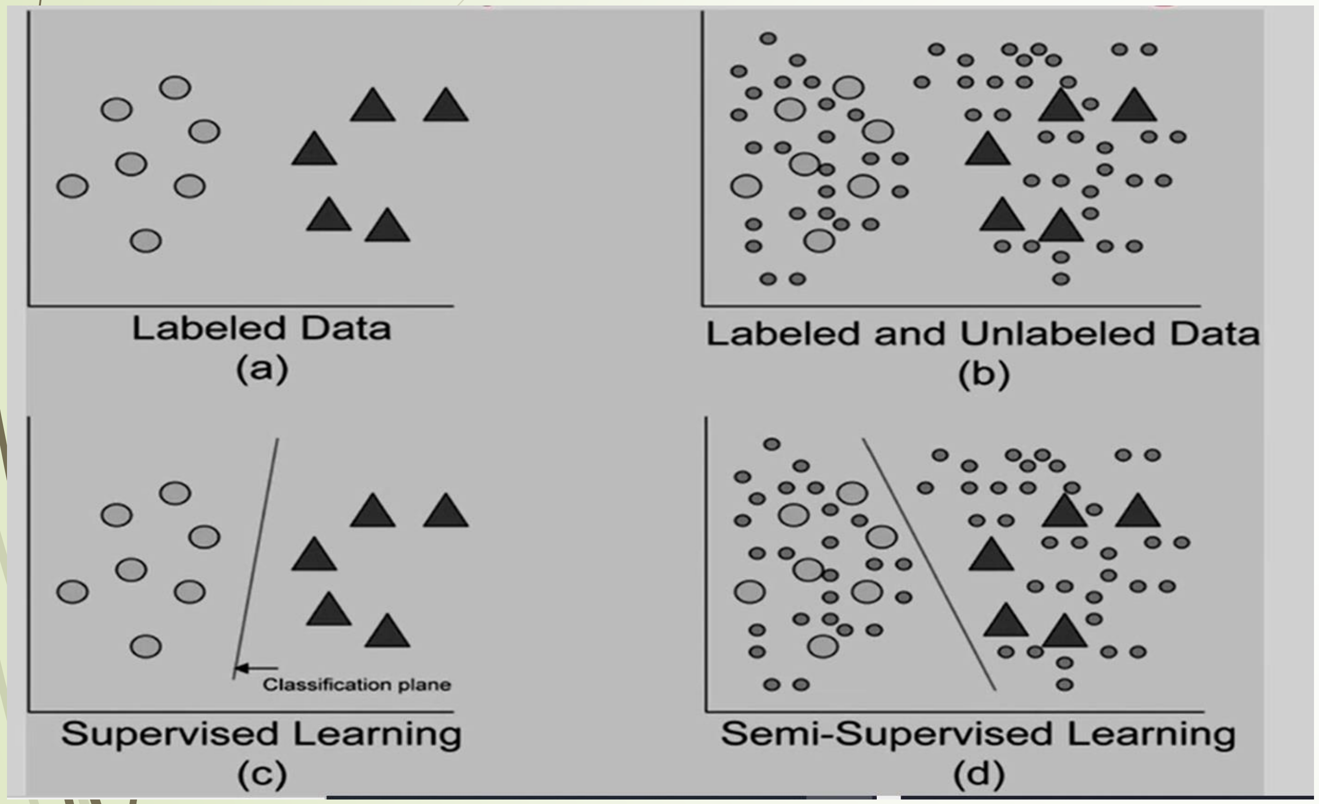


Reinforcement learning

- In reinforcement learning you have an agent who is acting in a environment, and you want to figure out what actions the agent must take at every step.
- The action that the agent takes is based on the rewards or penalty is that the agent gets in different states.
- The agent can take action and this action can impact the environment.
- In a particular stage, the agent takes an action and the environment goes to a new state and gives some reward to the agent, that reward may be a positive reward can be a negative reward or penalty or can be nothing at that particular time step. But the agent is continually acting in this world.



Semi-supervised Learning



- In semi-supervised learning, we have a combination of labeled data and unlabeled data.
- Fig (a) shows labeled data, the data which belong to two different classes, so one class is circle the other class is triangle;
- In semi-supervised learning, apart from having data from the two classes you also have unlabeled data which is indicated by the small circles in fig (b).
- Supervised data you will come up with some function and if you also have unlabeled data in addition to the labeled data you might try to come up with the better function.

Metrics for Performance Evaluation of Classifier

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes (Positive)	Class=No (Negative)
	Class=Yes (Positive)	a	b
	Class=No (Negative)	c	d

- The entries in the confusion matrix have the following meaning :
  - a is the number of **correct** predictions that an instance is **positive,**
  - *b* is the number of **incorrect** of predictions that an instance **negative**,
  - c is the number of **incorrect** predictions that an instance is **positive**, and
  - d is the number of **correct** predictions that an instance is **negative.**

Metrics for Performance Evaluation of Classifier

- The *accuracy* (AC)- is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

Contd...

- The *recall or true positive rate (TP)* is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$\mathrm{TP} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

- The *false positive rate (FP)* is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$\mathrm{FP} = \frac{c}{c+d} = \frac{FP}{FP+TN}$$

Contd...

- The *true negative rate (TN)* is defined as the proportion of negatives cases that were classified correctly, as calculated using the equation:

$$\mathrm{TN} = \frac{d}{d+c} = \frac{TN}{TN+FP}$$

- The *false negative rate (FN)* is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$\mathrm{FN} = \frac{b}{b+a} = \frac{FN}{FN+TP}$$

Contd.....

- *The* *precision (P)* is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$\mathrm{P} = \frac{a}{c+a} = \frac{TP}{FP+TP}$$

Example

- Suppose we train a model to predict whether an email is **Spam** or **Not Spam**. After training the model, we apply it to a test set of 500 new email messages (also labeled) and the model produces the contingency matrix below.

		True Class	
		Spam	Not Spam
Predicted Class	Spam	70	10
	Not Spam	40	380

(a) Compute the precision of this model with respect to the **Spam class.**

Precision with respect to SPAM = # correctly predicted as SPAM / # predicted as SPAM

= 70 / (70 + 10) = 70 / 80.

Contd…

(b) Compute the recall of this model with respect to the **Spam class.**

recall with respect to SPAM = # correctly predicted as SPAM / # truly SPAM

= 70 / (70 + 40) = 70 / 110.

- **High-precision and low recall with respect to SPAM:** whatever the model classifies as SPAM is probably SPAM. However, many emails that are truly SPAM are misclassified as NOT SPAM i.e <False Negative (False Acceptance)>
- **High recall and low precision with respect to SPAM:** the model filters all the SPAM emails, but also incorrectly classifies some genuine emails as SPAM i.e. <False Positive (False Rejectance)>.

End of the presentation