**Course Description**

This course provides a broad introduction to machine learning and statistical pattern recognition. Topics include: supervised learning (generative/discriminative learning, parametric/non-parametric learning, neural networks, support vector machines); unsupervised learning (clustering, dimensionality reduction, kernel methods); learning theory (bias/variance tradeoffs; VC theory; large margins); reinforcement learning and adaptive control. The course will also discuss recent applications of machine learning, such as to robotic control, data mining, autonomous navigation, bioinformatics, speech recognition, and text and web data processing.

**Prerequisites**

Students are expected to have the following background:

* Knowledge of basic computer science principles and skills, at a level sufficient to write a reasonably non-trivial computer program.
* Familiarity with the probability theory. (CS 109 or STATS 116)
* Familiarity with linear algebra (any one of Math 104, Math 113, or CS 205)

All REQUIRED COURSE MATERIALS ON : http://cs229.stanford.edu/materials.html

------------------------------------------------------------------------------------------------------------------------------------------

Hello guys. Roshan here, today I am gonna talk about the ML course provided by Stanford Uni (Andrew NG). So lets do it. Isn’t ML the most interesting topic? ML has large impact in our day to day lives including economy, science, engineering, health etc.

Q. What is Machine Learning?

* 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 .

Syllabus :

1. **Supervised learning.**

Intro to Machine Learning   
 Supervised learning setup. LMS.  
 Logistic regression. Perceptron. Exponential family.   
  Generative learning algorithms. Gaussian discriminant analysis. Naive Bayes.   
 Support vector machines.   
 Model selection and feature selection.   
 Ensemble methods: Bagging, boosting.   
  Evaluating and debugging learning algorithms.

1. **Learning theory.**

Bias/variance tradeoff. Union and Chernoff/Hoeffding bounds.   
VC dimension. Worst case (online) learning.   
Practical advice on how to use learning algorithms.

1. **Unsupervised learning.**

Clustering. K-means.   
EM. Mixture of Gaussians.   
Factor analysis.   
PCA (Principal components analysis).   
ICA (Independent components analysis).

1. **Reinforcement learning and control.**

MDPs. Bellman equations.   
Value iteration and policy iteration.   
Linear quadratic regulation (LQR). LQG.   
Q-learning. Value function approximation.   
Policy search. Reinforce. POMDPs.

Let’s Start then with some general idea about each of above topics ……………….

* **Supervised Learning**

**Supervised learning** is the [machine learning](https://en.wikipedia.org/wiki/Machine_learning) task of inferring a function from labeled training data.[[1]](https://en.wikipedia.org/wiki/Supervised_learning#cite_note-1) The [training data](https://en.wikipedia.org/wiki/Training_set) consist of a set of *training examples*. In supervised learning, each example is a *pair* consisting of an input object (typically a vector) and a desired output value (also called the *supervisory signal*). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

**Classes of Supervised Learning Problems:**

* **Regression Problem( Continuous Variables)**

Output can be guessed using two continuous variables range in regression problem. Example would be you selling a house in bay area, but you don’t know the actual selling price. So you can feed your algorithm a set of inputs including the price and specifications of previous houses sold, which acts as generalized formula generator to predict other output.

* **Classification Problem( Discrete Variables)**

Tumor can either be Malignant or Benign (harmless). So such output can be guessed using two Discrete variables 1 or 0. 1 means Malignant here. Such problems come under classification problem. In [machine learning](https://en.wikipedia.org/wiki/Machine_learning) and [statistics](https://en.wikipedia.org/wiki/Statistics), **classification** is the problem of identifying to which of a set of [categories](https://en.wikipedia.org/wiki/Categorical_data) (sub-populations) a new [observation](https://en.wikipedia.org/wiki/Observation) belongs, on the basis of a [training set](https://en.wikipedia.org/wiki/Training_set) of data containing observations (or instances) whose category membership is known. An example would be assigning a given email into ["spam" or "non-spam"](https://en.wikipedia.org/wiki/Spam_filtering) classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition).

In the terminology of machine learning,[[1]](https://en.wikipedia.org/wiki/Statistical_classification#cite_note-1) classification is considered an instance of [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning), i.e. learning where a training set of correctly identified observations is available.

* **Learning Theory**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), **computational learning theory** (or just **learning theory**) is a subfield of [Artificial Intelligence](https://en.wikipedia.org/wiki/Artificial_Intelligence) devoted to studying the design and analysis of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) algorithms

* **Unsupervised Learning**

**Unsupervised learning** is a type of machine **learning** algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common **unsupervised learning** method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data.

**Unsupervised machine learning** is the [machine learning](https://en.wikipedia.org/wiki/Machine_learning) task of inferring a function to describe hidden structure from "unlabeled" data (a classification or categorization is not included in the observations). Since the examples given to the learner are unlabeled, there is no objective evaluation of the accuracy of the structure that is output by the relevant algorithm—which is one way of distinguishing unsupervised learning from [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) .  
Unsupervised learning can be implemented in pattern recognition and sound analysis . Example: Andrew Ng’s experiment of (Uno,Dos,Tres…. + One,Two,Three = separate countings) . Also images pixel clustering is an example of UL where cluster of pixels are drawn as per the images provided . Note: That sound analysis part can be done using Matlab .

* **Reinforcement Learning**

Reinforcement learning , unlike one decision taken in supervised learning , here to get a job done a series of good or bad decisions are made which is then learnt by the machine to maximize the positives thereby minimizing the negatives and getting the result perfectly ! Eg : Training a dog requires both kind of Dog’s decision so that you give rewards and punishments accordingly and dog will learn that also accordingly taking its time which is Reinforcement Learning .

**Wiki says: Reinforcement learning** is an area of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) inspired by [behaviorist psychology](https://en.wikipedia.org/wiki/Behaviorism), concerned with how [software agents](https://en.wikipedia.org/wiki/Software_agent) ought to take [*actions*](https://en.wikipedia.org/wiki/Action_selection) in an *environment* so as to maximize some notion of cumulative *reward*.