**Definition of Machine Learning ?**

Field of study that gives computers the capability to learn without being explicitly programmed

**Example:** Training of students during exams.

Targeted Advertisement in FB,Instagram

Diagnosis of Cancer in Healthcare

Google Lens image text recognition

Spam detection in Gmail

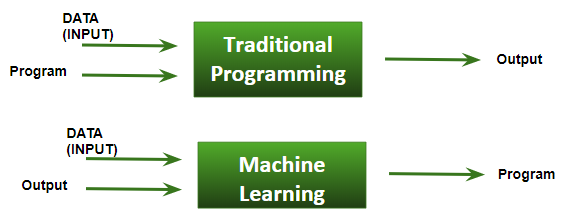
Web Search Engine

Face Recognition for photo tagging

**Basic Difference in ML and Traditional Programming?**

**Traditional Programming :** We feed in DATA (Input) + PROGRAM (logic), run it on machine and get output.

**Machine Learning :** We feed in DATA(Input) + Output, run it on machine during training and the machine creates its own program(logic), which can be evaluated while testing.

****

**How ML works?**

**Data Collection** : Gathering past data in any form suitable for processing.The better the quality of data, the more suitable it will be for modeling

**Data Processing** : Sometimes, the data collected is in the raw form and it needs to be pre-processed.  
 **Example:** Some tuples may have missing values for certain attributes, an, in this case, it has to be filled with suitable values in order to perform machine learning or any form of data mining.  
**Data Division** : Divide the input data into training,cross-validation and test sets. The ratio between the respective sets must be 6:2:2

**Model Building** : Building models with suitable algorithms and techniques on the training set.

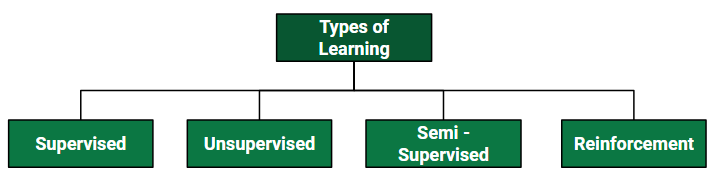
**Testing** : Testing our conceptualized model with data which was not fed to the model at the time of training

**Evaluation** : evaluating its performance using metrics such as F1 score, precision and recall.

**Prerequisites to learn ML:**

* Linear Algebra
* Statistics and Probability
* Calculus
* Graph theory
* Programming Skills - Python

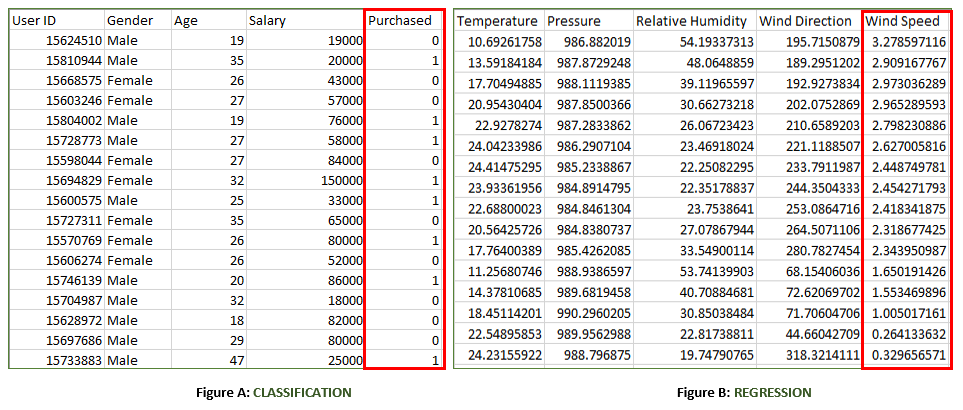
**Types of Machine Learning Algorithms:**

****

**Supervised Learning :**

Supervised learning is when the model is getting trained on a labelled dataset. Labelled dataset is one which has both input and output parameters.

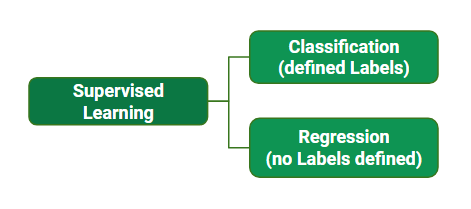
In this type of learning both training and validation datasets are labelled as shown in the figures below.



Both the above figures have labelled data set –

* **Figure A**: It is a dataset of a shopping store which is useful in predicting whether a customer will purchase a particular product under consideration or not based on his/ her gender, age and salary.  
  **Input** : Gender, Age, Salary  
  **Output** : Purchased i.e. 0 or 1 ; 1 means yes the customer will purchase and 0 means that customer won’t purchase it.
* **Figure B**: It is a Meteorological dataset which serves the purpose of predicting wind speed based on different parameters.  
  **Input** : Dew Point, Temperature, Pressure, Relative Humidity,Wind Direction  
  **Output** : Wind Speed

**Types of Supervised Learning:**

****

**Classification :** It is a Supervised Learning task where output is having defined labels(discrete value). For example in above Figure A, Output – Purchased has defined labels i.e. 0 or 1 ; 1 means the customer will purchase and 0 means that customer won’t purchase. The goal here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy.  
It can be either binary or multi class classification. In **binary** classification, model predicts either 0 or 1 ; yes or no but in case of **multi class** classification, model predicts more than one class.  
**Example:** Gmail classifies mails in more than one classes like social, promotions, updates, forum.

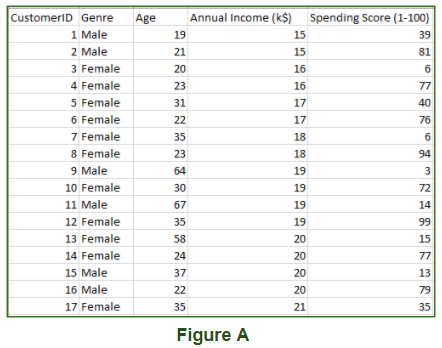
**Regression :** It is a Supervised Learning task where output is having continuous value.  
Example in above Figure B, Output – Wind Speed is not having any discrete value but is continuous in the particular range. The goal here is to predict a value as much closer to actual output value as our model can and then evaluation is done by calculating error value. The smaller the error the greater the accuracy of our regression model.

**Example of Supervised Learning Algorithms:**

* Linear Regression
* Nearest Neighbor
* Guassian Naive Bayes
* Decision Trees
* Support Vector Machine (SVM)
* Random Forest

**Unsupervised Learning :**

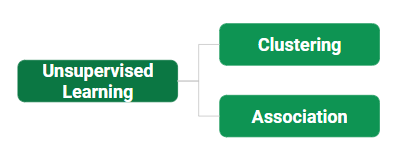
It’s a type of learning where we don’t give a target to our model while training i.e. the training model has only input parameter values. The model by itself has to find which way it can learn. Data-set in Figure A is mall data that contains information of its clients that subscribe to them. Once subscribed they are provided a membership card and so the mall has complete information about the customer and his/her every purchase. Now using this data and unsupervised learning techniques, mall can easily group clients based on the parameters we are feeding in.



Training data we are feeding is –

* **Unstructured data**: May contain noisy(meaningless) data, missing values or unknown data
* **Unlabeled data** : Data only contains value for input parameters, there is no targeted value(output). It is easy to collect as compared to the labelled one in Supervised approach.

**Types of Unsupervised Learning :-**

****

**Clustering:** Broadly this technique is applied to group data based on different patterns, our machine model finds. For example in the above figure we are not given output parameter value, so this technique will be used to group clients based on the input parameters provided by our data.

**Association:** This technique is a rule based ML technique which finds out some very useful relations between parameters of a large data set. For e.g. shopping stores use algorithms based on this technique to find out the relationship between sale of one product w.r.t to others sale based on customer behavior. Once trained well, such models can be used to increase their sales by planning different offers.

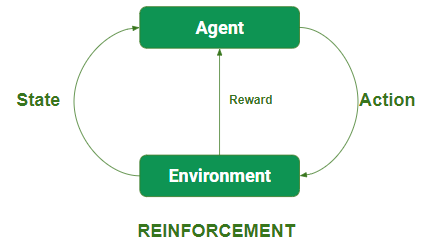
**Examples of Unsupervised Learning Algorithms:**

* K-Means Clustering
* DBSCAN – Density-Based Spatial Clustering of Applications with Noise
* BIRCH – Balanced Iterative Reducing and Clustering using Hierarchies
* Hierarchical Clustering

**Semi-supervised Learning:**

As the name suggests, its working lies between Supervised and Unsupervised techniques. We use these techniques when we are dealing with data which is a little bit labelled and a large portion of it is unlabeled. We can use unsupervised techniques to predict labels and then feed these labels to supervised techniques. This technique is mostly applicable in case of image data-sets where usually all images are not labelled.

**Reinforcement Learning:**

****

In this technique, the model keeps on increasing its performance using a Reward Feedback to learn the behavior or pattern. These algorithms are specific to a particular problem e.g. Google Self Driving car, AlphaGo where a bot competes with human and even itself to getting better and better performer of Go Game. Each time we feed in data, they learn and add the data to its knowledge that is training data. So, the more it learns the better it gets trained and hence experienced.

Agents observe input.

Agent performs an action by making some decisions.

After its performance, the agent receives reward and accordingly reinforces and the model stores a state-action pair of information.

**Examples of Reinforcement Algorithms:**

Temporal Difference (TD)

Q-Learning

Deep Adversarial Networks

**Terminologies of Machine Learning**

**Model :** A model is a **specific representation** learned from data by applying some machine learning algorithm. A model is also called a **hypothesis.**

**Feature :** A feature is an individual measurable property of our data. A set of numeric features can be conveniently described by a **feature vector**. Feature vectors are fed as input to the model.

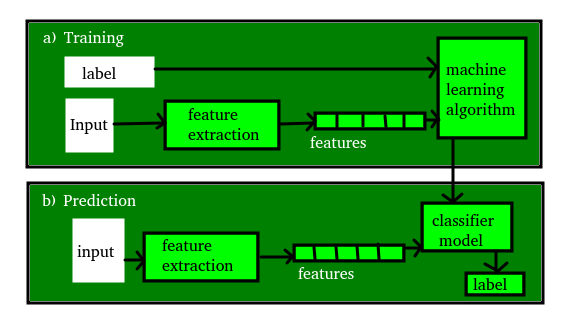
For example, in order to predict a fruit, there may be features like color, smell, taste, **etc.  
Note:** Choosing informative, discriminating and independent features is a crucial step for effective algorithms. We generally employ a **feature extractor** to extract the relevant features from the raw data.

**Target (Label)** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the features section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

**Training** The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

**Prediction** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

The figure shown below clears the above concepts:



**Data Handling:**

* <https://www.geeksforgeeks.org/ml-introduction-data-machine-learning/>
* <https://www.geeksforgeeks.org/ml-feature-scaling-part-1/>
* <https://drive.google.com/file/d/1K2SekZvwc9yqudeIziAvgxbDpTVPACrb/view?usp=sharing>

**Evaluation Measures:**

* <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>
* <https://www.geeksforgeeks.org/ml-log-loss-and-mean-squared-error/>
* <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>

**Machine Learning Algorithms:**

**Regression**

**1.Linear Regression:**

* ML | Linear Regression
  + <https://www.geeksforgeeks.org/ml-linear-regression/>
* Linear Regression (Python Implementation)
  + <https://www.geeksforgeeks.org/linear-regression-python-implementation/>
* Gradient Descent in Linear Regression
  + <https://www.geeksforgeeks.org/gradient-descent-in-linear-regression/>
* ML | Normal Equation in Linear Regression
  + <https://www.geeksforgeeks.org/ml-normal-equation-in-linear-regression/>
* Linear Regression Using Tensorflow
  + <https://www.geeksforgeeks.org/linear-regression-using-tensorflow/>

**2.Polynomial Regression:**

* Python | Implementation of Polynomial Regression
  + <https://www.geeksforgeeks.org/python-implementation-of-polynomial-regression/>
* Types of Regression Techniques
  + <https://www.geeksforgeeks.org/types-of-regression-techniques/>

**3.Support Vector Machine:**

* Classifying data using Support Vector Machines(SVMs) in Python
  + [https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python/](https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python)
* ML | Using SVM to perform classification on a non-linear dataset
  + <https://www.geeksforgeeks.org/ml-using-svm-to-perform-classification-on-a-non-linear-dataset/>
* ML | Non-Linear SVM
  + <https://www.geeksforgeeks.org/ml-non-linear-svm/>
* Train a Support Vector Machine to recognize facial features in C++
  + <https://www.geeksforgeeks.org/train-a-support-vector-machine-to-recognize-facial-features-in-c/>

**4.Decision Tree:**

* Decision Tree Introduction with example
  + <https://www.geeksforgeeks.org/decision-tree-introduction-example/>
* Python | Decision Tree Regression using sklearn
  + <https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/>
* Decision tree implementation using Python
  + <https://www.geeksforgeeks.org/decision-tree-implementation-python/>
* ML | Logistic Regression v/s Decision Tree Classification
  + <https://www.geeksforgeeks.org/ml-logistic-regression-v-s-decision-tree-classification/>

**5.Random Forest:**

* Random Forest Regression in Python
  + <https://www.geeksforgeeks.org/random-forest-regression-in-python/>

**Classification**

**1.Logistic Regression:**

* ML | Why Logistic Regression in Classification ?
  + <https://www.geeksforgeeks.org/ml-why-logistic-regression-in-classification/>
* ML | Cost function in Logistic Regression
  + <https://www.geeksforgeeks.org/ml-cost-function-in-logistic-regression/>
* ML | Logistic Regression using Python
  + <https://www.geeksforgeeks.org/ml-logistic-regression-using-python/>
* Understanding Logistic Regression
  + <https://www.geeksforgeeks.org/understanding-logistic-regression/>
* ML | Logistic Regression using Tensorflow
  + <https://www.geeksforgeeks.org/ml-logistic-regression-using-tensorflow/>

**2.K-Nearest Neighbors:**

* K-Nearest Neighbours
  + <https://www.geeksforgeeks.org/k-nearest-neighbours/>
* Implementation of K Nearest Neighbors
  + <https://www.geeksforgeeks.org/implementation-k-nearest-neighbors/>
* Project | kNN | Classifying IRIS Dataset
  + <https://www.geeksforgeeks.org/project-knn-classifying-iris-dataset/>

**3.Naive Bayes Classification:**

* Naive Bayes Classifiers
  + <https://www.geeksforgeeks.org/naive-bayes-classifiers/>
* Applying Multinomial Naive Bayes to NLP Problems
  + [https://www.geeksforgeeks.org/applying-multinomial-naive-bayes-to-nlp-problems](https://www.geeksforgeeks.org/applying-multinomial-naive-bayes-to-nlp-problems/)

**4.Decision Tree Classification:**

* Decision Tree Introduction with example
  + <https://www.geeksforgeeks.org/decision-tree-introduction-example/>
* Python | Decision Tree Regression using sklearn
  + <https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/>
* Decision tree implementation using Python
  + <https://www.geeksforgeeks.org/decision-tree-implementation-python/>
* ML | Logistic Regression v/s Decision Tree Classification
  + <https://www.geeksforgeeks.org/ml-logistic-regression-v-s-decision-tree-classification/>

**5.Random Forest Classification:**

* Random Forest Regression in Python
  + <https://www.geeksforgeeks.org/random-forest-regression-in-python/>

**Clustering**

**1.K-Means Clustering:**

* K-Means Clustering Introduction
  + <https://www.geeksforgeeks.org/k-means-clustering-introduction/>
* K-Means++ Introduction
  + <https://www.geeksforgeeks.org/ml-k-means-algorithm/>

**2.Mean Shift Clustering:**

* Mean Shift Clustering Introduction
  + <https://www.geeksforgeeks.org/ml-mean-shift-clustering/>

**3.Agglomerative Clustering:**

* Agglomerative Introduction
  + <https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divisive-clustering/>
  + <https://www.geeksforgeeks.org/hierarchical-clustering-in-data-mining/>

**Data Dimensionality**

**Introduction:**

* <https://www.geeksforgeeks.org/dimensionality-reduction/>

**Feature Selection:**

* **Correlation Matrix:**
  + <https://www.geeksforgeeks.org/parameters-feature-selection/>
* **Extra Tree Classifier:**
  + <https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/>
* **Chi Square intuition with KBest Method :**
  + <https://www.geeksforgeeks.org/ml-chi-square-test-for-feature-selection/>

**Feature Extraction:**

* **PCA :**
  + <https://www.geeksforgeeks.org/ml-principal-component-analysispca/>
* **t-SNE :**
  + <https://www.geeksforgeeks.org/ml-t-distributed-stochastic-neighbor-embedding-t-sne-algorithm/>

**Cross Validation:**

**Introduction :**

* <https://www.geeksforgeeks.org/cross-validation-machine-learning/>

**k-Fold Cross validation:**

* <https://www.geeksforgeeks.org/k-fold-cross-validation-in-r-programming/>

**Stratified k-Fold cross validation:**

* <https://www.geeksforgeeks.org/stratified-k-fold-cross-validation/>

**Association Mining:**

* Frequent Itemset in Data set (Association Rule Mining)
  + <https://www.geeksforgeeks.org/frequent-item-set-in-data-set-association-rule-mining/>
* Apriori Algorithm
  + <https://www.geeksforgeeks.org/apriori-algorithm/>
* ML | ECLAT Algorithm
  + <https://www.geeksforgeeks.org/ml-eclat-algorithm/>

**Natural Language Processing :**

* <https://www.geeksforgeeks.org/introduction-to-natural-language-processing/>
* <https://www.geeksforgeeks.org/applications-of-nlp/>
* [https://www.geeksforgeeks.org/processing-text-using-nlp-basics/](https://www.geeksforgeeks.org/applications-of-nlp/)