Lecture 6

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 **hypothesis**.

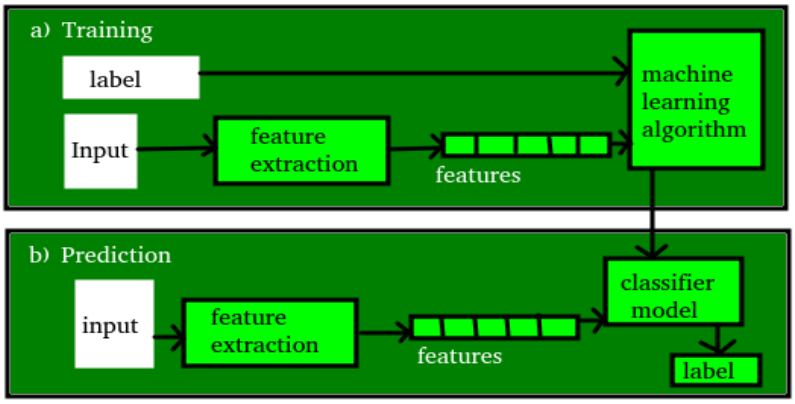
- **Feature**
  A feature is an individual measurable property of our data. A set of numeric features can be conveniently described by a **feature vector**. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, **etc. 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:



We can use the canonical problem of spam detection as a running example to illustrate some basic definitions and to describe the use and evaluation of machine learning algorithms in practice. Spam detection is the problem of learning to automatically classify email messages as either spam or non-spam.

Examples: Items or instances of data used for learning or evaluation. In our spam problem, these examples correspond to the collection of email messages we will use for learning and testing.

Features: The set of attributes, often represented as a vector, associated to an example. In the case of email messages, some relevant features may include the length of the message, the name of the sender, various characteristics of the header, the presence of certain keywords in the body of the message, and so on.

Labels**:** Values or categories assigned to examples. In classification problems, examples are assigned specific categories, for instance, the spam and non-spam categories in our binary classification problem. In regression, items are assigned real-valued labels.

Training sample**:** Examples used to train a learning algorithm. In our spam problem, the training sample consists of a set of email examples along with their associated labels.

Validation sample**:** Examples used to tune the parameters of a learning algorithm when working with labeled data. Learning algorithms typically have one or more free parameters, and the validation sample is used to select appropriate values for these model parameters.

Test sample**:** Examples used to evaluate the performance of a learning algorithm. The test sample is separate from the training and validation data and is not made available in the learning stage. In the spam problem, the test sample consists of a collection of email examples for which the learning algorithm must predict labels based on features. These predictions are then compared with the labels of the test sample to measure the performance of the algorithm.

Loss function**:** A function that measures the difference, or loss, between a predicted label and a true label.

Hypothesis set**:** A set of functions mapping features (feature vectors) to the set of labels *Y*. In our example, these may be a set of functions mapping email features to *Y = {*spam, non-spam*}*.

We now define the learning stages of our spam problem. We start with a given collection of labeled examples. We first randomly partition the data into a training sample, a validation sample, and a test sample. The size of each of these samples depends on a number of different considerations. For example, the amount of data reserved for validation depends on the number of free parameters of the algorithm.

Also, when the labeled sample is relatively small, the amount of training data is often chosen to be larger than that of test data since the learning performance directly depends on the training sample. Next, we associate relevant features to the examples. This is a critical step in the design of machine learning solutions. Useful features can effectively guide the learning algorithm, while poor or uninformative ones can be misleading. Although it is critical, to a large extent, the choice of the features is left to the user. This choice reflects the user's *prior knowledge* about the learning task which in practice can have a dramatic effect on the performance results.
Now, we use the features selected to train our learning algorithm by fixing different values of its free parameters. For each value of these parameters, the algorithm selects a different hypothesis out of the hypothesis set. We choose among them the hypothesis resulting in the best performance on the validation sample. Finally, using that hypothesis, we predict the labels of the examples in the test sample. The performance of the algorithm is evaluated by using the loss function associated to the task.

***

A free parameter is one that can be adjusted to make the model fit the data. If I make a model that says A is proportional to B, there is one free parameter, the proportionality constant. If my model has a specific value of the proportionality constant, there are no free parameters. If I say that A is a quadratic function of B, there are three free parameters, a,b,c in $A=aB^2+bB+c$. That makes it easier to fit the data, even if my model is not correct, so it is less impressive.