ML in summary

Saurav Sumit (17IE33011)
Sujabrata Mallick (17IE33015)
Arnab Moitra (17MI31008)

31st March 2021

Introduction

The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence, and stated that "it gives computers the ability to learn without being explicitly programmed".

And in 1997, Tom Mitchell gave a "well-posed" mathematical and relational definition that "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. Machine Learning is the latest buzzword floating around. It deserves to, as it is one of the most interesting sub-fields of Computer Science. So what does Machine Learning really mean?

Let's try to understand Machine Learning with a practical example. Consider you are trying to toss a paper into a dustbin. After the first attempt, you realize that you have put too much force into it. After the second attempt, you realize you are closer to the target but you need to increase your throw angle. What is happening here is basically after every throw we are learning something and improving the end result. We are programmed to learn from our experience.

This implies that the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?"

Within the field of data analytics, machine learning is used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data set(input).

Classification of Machine Learning

Machine learning implementations are classified into three major categories, depending on the nature of the learning "signal" or "response" available to a learning system which are as follows:-

1. **Supervised learning**: When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning. This approach is indeed similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize, and the student then derives general rules from these specific examples.

2. **Unsupervised learning**: Whereas when an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms. As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

3. **Reinforcement learning**: When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error. Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others. An interesting example of reinforcement learning occurs when computers learn to play video games by themselves. In this case, an application presents the algorithm with examples of specific situations, such as having the gamer stuck in a maze while avoiding an enemy. The application lets the algorithm know the outcome of actions it takes, and learning occurs while trying to avoid what it discovers to be dangerous and to pursue survival. You can have a look at how the company Google DeepMind has created a reinforcement learning program that plays old Atari's video games. When watching the video, notice how the program is initially clumsy and unskilled but steadily improves with training until

it becomes a champion.

4. **Semi-supervised learning**: where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing.

Basic Difference in ML and Traditional Programming?

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

Figure 1: Flowchart of Traditional Programming and Machine Learning

Supervised Learning

Supervised machine learning algorithms are designed to learn by example. The name "supervised" learning originates from the idea that training this type of algorithm is like having a teacher supervise the whole process.When training a supervised learning algorithm, the training data will consist of inputs paired with the correct outputs. During training, the algorithm will search for patterns in the data that correlate with the desired outputs. After training, a supervised learning algorithm will take in new unseen inputs and will determine which label the new inputs will be classified as based on prior training data. The objective of a supervised learning model is to predict the correct label for newly presented input data. At its most basic form, a supervised learning algorithm can be written simply as Y = f(x) Where Y is the predicted output that is determined by a

mapping function that assigns a class to an input value x. The function used to connect input features to a predicted output is created by the machine learning model during training.

Supervised learning can be split into two subcategories: **Classification** and **Regression**.

1. **Classification**: During training, a classification algorithm will be given data points with an assigned category. The job of a classification algorithm is to then take an input value and assign it a class, or category, that it fits into based on the training data provided.

   The most common example of classification is determining if an email is spam or not. With two classes to choose from (spam, or not spam), this problem is called a binary classification problem. The algorithm will be given training data with emails that are both spam and not spam. The model will find the features within the data that correlate to either class and create the mapping function mentioned earlier: Y=f(x). Then, when provided with an unseen email, the model will use this function to determine whether or not the email is spam. Classification problems can be solved with numerous algorithms. Whichever algorithm you choose to use depends on the data and the situation. A few popular classification algorithms are: **Linear Classifiers, Support Vector Machines, Decision Trees, K-Nearest Neighbors, Random Forest.**

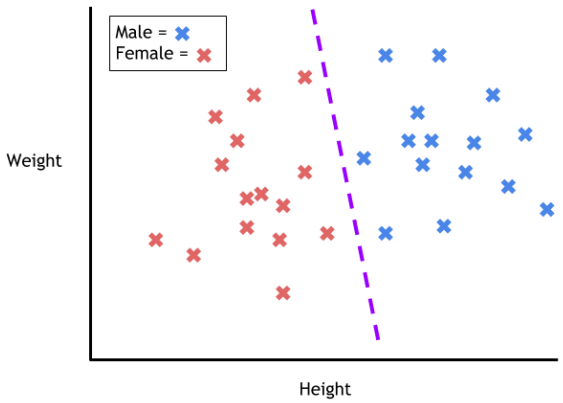

Figure 2: Example of classification of data points

2. **Regression**: Regression is a predictive statistical process where the model attempts to find the important relationship between dependent and independent variables. The goal of a regression algorithm is to predict a continuous number such as sales, income, and test scores. The equation

for basic linear regression can be written as so :

$$y = w[0] * x[0] + w[1] * x[1] + ...w[i] * x[i] + b$$

Where x[i] is the feature(s) for the data and where w[i] and b are parameters that are developed during training. For simple linear regression models with only one feature in the data, the formula looks like this

$$y = w * x + b$$

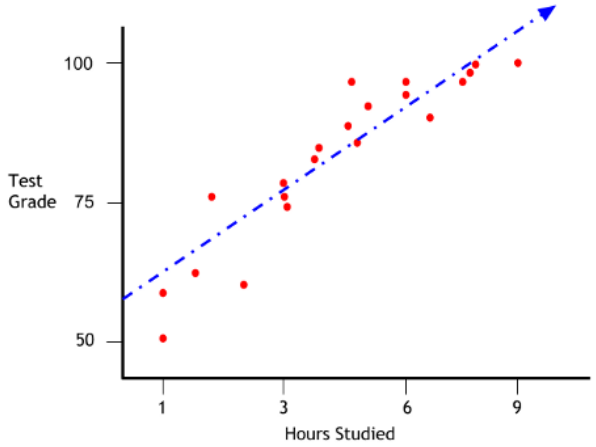

Figure 3: An example of best fit line on a dataset

There are many different types of regression algorithms. The three most common are listed below:

- Linear Regression
- Logistic Regression
- Polynomial Regression

Linear Regression
Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

Logistic Regression
Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model (a form of binary regression).

Polynomial Regression
Polynomial regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an nth degree polynomial in x. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y —x).

Decision Tree Learning

The Decision Tree Learning is the easiest model to understand and interpret. Following is the Algorithm in brief:

- Every time find the 'Best Attribute' and the value of that attribute which essentially forms the a node.
- Add the Node, and if the data is pure now; then we are done else we build upon the tree recursively.

Figure 4: Decision Tree Classifier

Here, the '**Best Attribute**' is '**Best**' in terms of Information Gain, this is essentially a **Greedy Algorithm** applied Recursively. A node is '**Pure**' when the remaining data is of only one class. Purity Now - Purity Before = Information gain
This model can be applied in varied scenarios and often helps us to understand the important attributes which are often very useful, however the model has some drawbacks as well especially when there are a huge number of attributes.

Support Vector Machine

The Support Vector Machine algorithm seeks to maximize the '**Margin**' between the +ve and -ve examples; and thus also known as Maximum Margin Classifier.

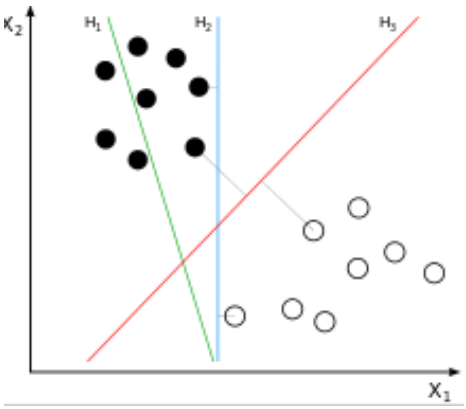

Figure 5: The Maximum Margin Classifier

In the adjacent diagram, H3 is the maximum margin classifier. The prime idea here is that the higher the margin , the lesser is the chance for mis-classification and this better the hypothesis. Now, this problem can be formulated as an Optimization problem, where we seek to maximize the Margin.

As it turns out that the Maximum Margin Separator doesn't really depend upon all the data points; but depends only on a few data points which essentially are the support vectors.

A data which is not linearly separable in 2 dimension, might become Linearly Separable once it's mapped o a higher dimensional space. SVM essentially does this through something known as '**Kernel Trick**'. The Kernel might project something to 2-dimensional space, 3-dimensional space or even infinite dimensional space [i.e. RBF Kernel].

Bayesian Classifier [Naive Bayes Algorithm]

One of the simplest algorithms in the Machine Learning domain is the Naive Bayes Algorithm, which performs fairly well in Natural Language Processing problems especially. This classifier works on the basis of Bayes' Theorem [and thus the name]. It assumes the simplest form of Bayesian Network, i.e. all the attributes are assumed to be independent.

Algorithm:
Let's say there are n classes, 1,2,3...n. Given a vector of attributes X, we shall assign X to class k, such that k = argmax (P(c=i |x=X)) over i .

P(c=Ci |x=X) is proportional P(c=Ci) * P(x=X |c=Ci). [By Bayes' Theorem]. P(c=Ci) and P(x=X |c=Ci) are calculated using **Maximum Likelihood Estimator**.

One of the prime issues for this algorithm is that we assume that the attributes are Pairwise independent which in practice is seldom true. The second thing, while calculating the required probabilities we might find that the number of instances might be 0, which shall make the probability 0, in that case we are to use the Prior Knowledge that we have.

K - Nearest Neighbors [Instance Based Learning]

The **k-nearest neighbors algorithm (k-NN)** is a non-parametric classification method. It is used for classification and regression. In both cases, the input consists of the k closest training examples in the data set. The output depends

on whether k-NN is used for classification or regression: The KNN algorithm essentially depends upon the distance metric, so often what is done is more importance is given to data points closer than those which are farther [i.e. The point is more similar to those which are closer to it].

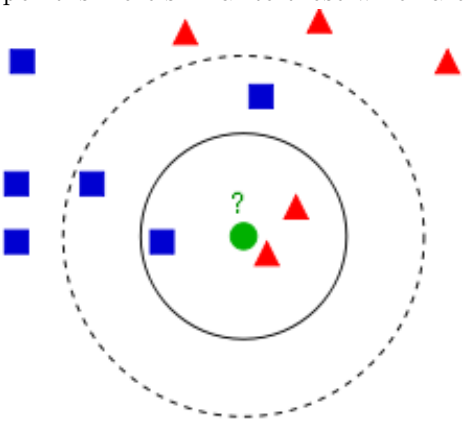

Figure 6: Classifying with varying K value

Algorithm:
For each test point, calculate the distance of each of the training points to that point.

Select the nearest K Point, and assign the class based on the majority class among those K points.

Similarly, for Regression predict based upon weighted distance values.

The issues for KNN are:

- Time Complexity : Addressed through Condensing and using advanced data structure such as.K-d trees.
- The choice of Distance Metric : The way we define our distance function essentially has a profound impact on the performance of the algorithm.
- Value of K : This has to do with the Generalization issues, i.e. Les value of K won't be able to generalize,and higher value of K would generalize too much.

This Algorithm however is pretty easy to implement, and works pretty well in some domains especially in the domain of anomaly detection.

Artificial Neural Network

Artificial Neural Network, is a very powerful tool in the field of Machine Learning, and is extensively used to understand images and a separate field of 'Deep Learning' has emerged. Artificial Neural Network is originally trying to mimic the way humans see or feel, i.e. we have Neurons which transmits signals to various parts of our body to and from our brain.

Parts of a Neural Network

- Input Layer
- Hidden Layer
- Output Layer

Then, besides these we also have an activation function as well, to be applied.

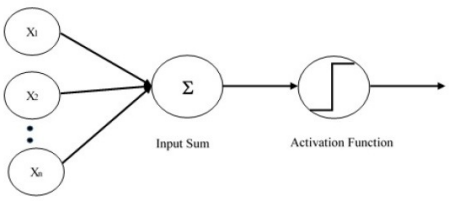

Figure 7: Single Layer Perceptron

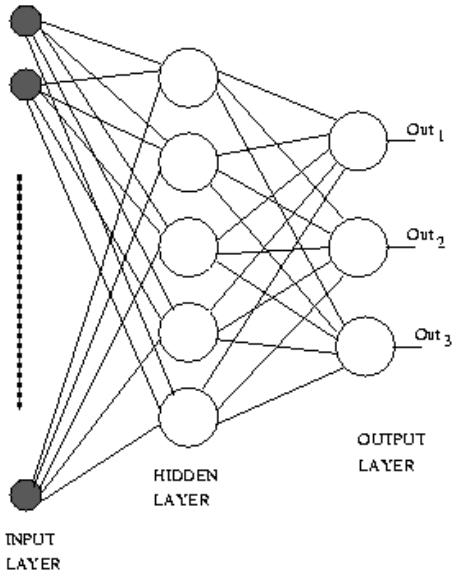

Figure 8: Multi -Layer Neural Networks

The Neural Networks are trained by a method known as **Back-Propagation**, where the errors are Back-Propagated, and the weights get updated based on the errors that occurred in the following layer. For training, there is a **Forward Pass**, followed by a **Backward pass**. Neural Networks are very powerful, however there are many hyper-parameters [No. of Hidden Layers, No. of Hidden Units, No. of epochs, Mini-Batch Size for instance] which need to be tuned.

Hypothesis Evaluation

Evaluation is one of the most important measures of the performance of a model in Machine Learning. The problem here obviously is the fact that, we never would be able to see the full picture, and still we have to make sure that our model has low Bias and it performs fairly well in out-of-sample data. For this, certain techniques are recommended, which again are applicable in certain scenarios. However, the following methods have been proposed for getting a good approximation of our variables of interest.

The data often is divided into Train, Test and Validation set. The model is trained on the Training Data, Hyper-parameters are tuned on the basis of its performance in the Validation data,and finally applied on the test data.

- **Accuracy Score** : It just finds the accuracy, applicable for classification problems
- **RMSE Score**: This is Root Mean Squared Error, applicable for Regression Problems

- **MAE Score**: This is Mean Absolute Error, applicable for Regression Problems

To, compare between the performance of 2 classifiers, we can use AUC [Area under the Curve] measure. The curve is obtained by changing the hyperparameter value.

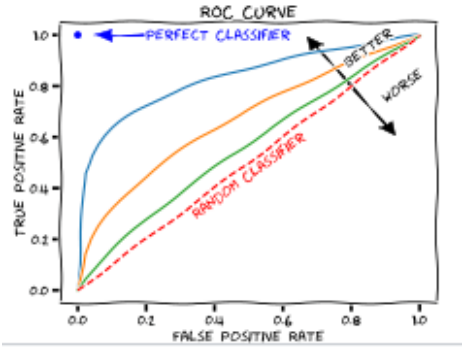

Figure 9: ROC Curve with TPR on Y axis and FPR on X axis

It is to be noted that we are trying to minimize the bias, and at the same time we want, to minimize the variance. For that, we want to do multiple runs for our model and like to get an overall idea about how it's performing, for that we perform K-fold Cross Validation , Leave One Out method etc. For various models, it's good to note that with fewer data points, our model is prone to overfitting, whilst with more data the bias would be high. The higher the AUC, the better is the classifier. The TPR, FPR is plotted and we want TPR ¿ FPR.

Unsupervised Learning

Unsupervised learning is a type of algorithm that learns patterns from untagged data. The hope is that through mimicry, the machine is forced to build a compact internal representation of its world. The main method used in unsupervised learning is clustering.

Clustering

Clustering is the process of grouping objects into subsets that have meaning in the context of a particular problem. The objects are thereby organized into an efficient representation that characterizes the population being sampled. Unlike classification, clustering does not rely on predefined classes. Clustering is referred to as an unsupervised learning method because no information is provided about the "right answer" for any of the objects. It can uncover previously undetected relationships in a complex data set. Many applications for cluster analysis exist. For example, in a business application, cluster analysis can be used to discover and characterize customer groups for marketing purposes.

Two types of clustering algorithms are nonhierarchical and hierarchical.
In **Partitional clustering**, such as the k-means algorithm, the relationship between clusters is undetermined. **Hierarchical clustering** repeatedly links pairs of clusters until every data object is included in the hierarchy. It's of 2 types: Divisive and Agglomerative.

With both of these approaches, an important issue is how to determine the similarity between two objects, so that clusters can be formed from objects with a high similarity to each other. Commonly, distance functions, such as the

Manhattan and Euclidean distance functions, are used to determine similarity. A distance function yields a higher value for pairs of objects that are less similar to one another. Sometimes a similarity function is used instead, which yields higher values for pairs that are more similar.

Another type of clustering is **Density Based Clustering**, which has certain pros and cons as well. For instance, it doesn't require a number of clusters to be defined apriori, and also can be used extensively for Anomaly Detection; since it's pretty great in identifying outliers. Again, different clustering algorithms work for different scenarios.

Other avenues of Machine Learning

With the advent of Neural Networks and related technical advancement, the field of **Deep Learning** has emerged which is largely used in understanding Image, Sound etc, Other than that the field of **Reinforcement Learning** has emerged as well to model much complex situations such as prediction of stock markets especially in a stochastic environment. There is a field of **Statistical Learning**, which focuses more on the Statistical aspects of Machine Learning, including some of the areas of Traditional Machine Learning.

In days, to come this field shall develop more and would attract some of the world's best research talents.