**INTRODUCTION**

**MACHINE LEARNING**

According to Arthur Samuel machine learning is the computer’s ability to learn without being specially programmed. He wrote a checker’s game playing program, And the amazing thing about this checker’s game playing program was that Arthur Samuel himself was not a very good checkers player. But, what he did was, he had to write a program for it to play a lot of games against itself, And by watching the winning and losing positions, the checkers playing program learnt over time what winning and losing positions. And eventually learnt to play checkers game better than Arthur Samuel himself was able to. Because the computer has the patience to play a number of games itself it was able to get so much checkers-playing experience that it became a better player.

According to Tom Mitchell,

**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.**

For the checkers example the experience e, will be the experience of program playing a lot of games against itself. The task t, will be the task of playing checker’s game, And the performance measure p, will be the probability that it wins and not loses the next game of checkers against some opponent.

There are different types of machine learning algorithms. The main two types are

1. Supervised learning

2. Unsupervised learning.

In supervised learning, we teach the computer how to do something, whereas in unsupervised learning we let it learn by itself.

**SUPERVISED LEARNING ALGORITHM**

In supervised learning we give the algorithm a data set in which the "right answers" were given.



For example, let's say we want to predict housing prices. We collect data sets and plot a data set and it looks like above. On the horizontal axis, the size of different houses is in square feet, and on the vertical axis, the price of different houses is in dollars. So, given this data, let's say we want to know the price of size, say 750 square feet. So how can the learning algorithm help you? One thing a learning algorithm can do is to put a straight line through the data and, based on that, it looks like the house can be sold for about $150,000. But there might be a better learning algorithm. For example, we might decide that it's better to fit a quadratic function to this data. Then, maybe we can sell the house for about $200,000. So, we gave it a data set of houses in which for every example, we told it what is the right price so the task of the algorithm was to just give more of right answers . This is also called a **regression problem** and by regression we mean we try to predict a continuous value output.

But for some learning problems, we don’t want to use three or five features. But we want to use an infinite number of features so that the learning algorithm has lots of features with which to make correct predictions. So how do you even store an infinite number of features on the computer when your computer runs out of memory. Using **Support Vector Machine**, there is a mathematical trick that allows a computer to store an infinite number of features.

**UNSUPERVISED LEARNING**

In Unsupervised Learning, we're given the data that looks quite different than data that looks similar. So we're given this data set and we are not told what we should do with this. Given this data set, the Unsupervised Learning algorithm can decide that the data lives in two or more different clusters. Supervised Learning algorithm will however, break this data into two separate clusters. So, this is called a **clustering algorithm**.

Clustering is used is in **Google News** . Google News does looks at tens of thousands of new stories on the web and it clusters them into group of news stories. For example:



The URLs here are the link to different stories about the BP Oil Well story. So, on clicking on one of these we get a Wall Street Journal article about, the BP Oil Well stories of "BP Kills Macondo". If we click on a different URL from that group of story then we get the different story. If we click on a third link, then we get a different story. Here's the UK Guardian story about the BP Oil Well Spill. So, the news stories that are about the same topic get clustered together.

Here's one example on understanding the genomics. It is an example of DNA microarray data.



We put a group of different individuals and then for each of them, we measure what quantity they do or do not have a particular gene. So these colors, show the degree up to which the different individuals do or do not have a particular gene. Then run a clustering algorithm to group these individuals into different categories. So this is how Unsupervised Learning is used, because we do not tell the algorithm about the type of people. We don't know who's and what type of data is given. We don't even know what and how many the different types of people are, but can we automatically find certain structure in the data. Because we do not give the algorithm the right answer for the examples in given data set, this is Unsupervised Learning. Unsupervised Learning, also called clustering is used for a lot of other applications too.

**2.LITERATURE REVIEW**

Some of the most popular Machine Learning techniques in the market today are Bayesian classification(the oldest one) , k-NN, ANNs, Artificial immune system,SVMs, and Rough sets. We review some of them in this section.

**2.1 Bayes classification method**

It was proposed in 1988. Bayesian classifier works on the dependent events and the probability of an event occurring in the future that can be detected from its previous occurring. Words probabilities play the main rule in this. If some words occur often in spam emails but do not occur in ham, then that e-mail is probably spam. The statistic we are interested in for a particular token is its 'spamminess' (spam rating), which can be calculated as follows:



Where ,  
CSpam(T)=number of spam messages

Cham(T) = number of ham messages containing token T.

Now, to calculate the possibility for a message M to be a spam with tokens {T1,......,TN}, we need to combine the individual token's spamminess to calculate the overall message's spamminess. A naive way to make classifications is to calculate product of individual token's spamminess and compare it with the product of individual token's hamminess .The message is classified spam if the overall spamminess product 'S[M]' is greater than the hamminess product 'H[M]'.

ALGORITHM :

1.) -Separate a message into tokens

-Calculate a probability for each token T :

S[T] = Cspam(T) / ( Cham(T) + Cspam(T) )

Store all the spamminess values in a file.

2.) Filtering :

-For each message M ,

-while (M does not end) do

-scan message for next token 'Ti '

-query the earlier file for spamminess S(Ti)

-calculate total spam and ham message probabilities: S[M] and H[M] -calculate the message filtering indication by: I[M] = f(S[M] , H[M]) ; where f is a filter function, like :



If I[M] > threshold ; message is marked as spam otherwise it is marked as non-spam.

**K-nearest neighbour classifier method**

The k-nearest neighbour (K-NN) classifier is considered an example-based classifier, that means that the training documents are used for comparison rather than an explicit category representation, such as the category profiles used by other classifiers. As such, there is no real training phase. When a new document needs to be categorized, the k most similar documents (neighbours) are found and if a large enough proportion of them have been assigned to a certain category, the new document is also assigned to this category, otherwise not . Additionally, finding the nearest neighbours can be quickened using traditional indexing methods. To decide whether a message is spam or ham, we look at the class of the messages that are closest to it. The comparison between the vectors is a real time process. This is the idea of the k nearest neighbor algorithm:

Stage1. Training Store the training messages.

Stage2. Filtering Given a message x, determine its k nearest neighbours among the messages in the training set. If there are more spam's among these neighbours, classify given message as spam. Otherwise classify it as ham.

The use here of an indexing method in order to reduce the time of comparisons which leads to an update of the sample with a complexity O(m), where m is the sample size. As all of the training examples are stored in memory, this technique is also referred to as a memory-based classifier [6]. Another problem of the presented algorithm is that there seems to be no parameter that we could tune to reduce the number of false positives. This problem is easily solved by changing the classification rule to the following l/k-rule:

If l or more messages among the k nearest neighbours of x are spam, classify x as spam, otherwise classify it as legitimate mail.

**2.3 Artificial Neural Networks classification method**

An artificial neural network , or a "Neural Network" , is a computation model based on neural networks inside brain. It consists of an interconnected system of artificial neurons. An artificial neural network is a dynamic system which changes its structure based on information that goes through the network during the learning phase. It is based on the principle of learning by examples. There are 2 kinds of the neural networks, the simple perceptron and the multilayer perceptron. Let us look at the perceptron algorithm.

The idea of the perceptron algorithm is to find the feature vector f(x) = pTx + c such that f(x)> 0 for members of one class , and f(x)< 0 for members of the other class. Here p = (p1p2,…pm) is the vector of coefficients of the function, and c is the bias. Let us denote the classes by +1 and -1,so we search for a decision function d(x)= (sign )(pTx + c). The learning is done with an iterative algorithm. It starts with randomly chosen parameters (p0,c0) of the decision and updates them iteratively. On the n-th iteration of the algorithm a training sample (x,b) is chosen such that the current decision function does not classify it correctly (i.e. sign (pnx + cn) ≠ b). The parameters (pn,cn) are updated using the rule:

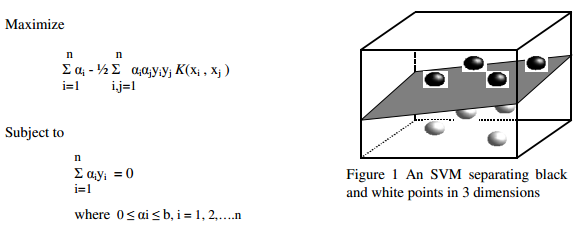
pn+1= pn+ bx

cn+1 = cn+ b

The algorithm stops when a decision function is found such that it correctly classifies all the training examples.

**Support Vector Machines classifier method**

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships, the SVM modeling algorithm finds an optimal hyperplane with the maximal margin to separate two classes, which requires solving the following optimization problem.



Where αi is the weight of training samplex1. If αi > 0, x1is called a support vector bis a regulation parameter used to trade-off the training accuracy and the model complexity so that a superior generalization capability can be achieved. Kis a kernel function, which is used to measure the similarity between two samples. A popular radial basis function (RBF)kernel functions.



To determine the values of < γ, b >, a cross validation process is usually conducted on the training dataset . Cross validation is also used to estimate the generalization capability on new samples that are not in the training dataset. A k-fold cross validation randomly splits the training dataset into k approximately equal-sized subsets, leaves out one subset, builds a classifier on the remaining samples, and then evaluates classification performance on the unused subset. This process is repeated k times for each subset to obtain the cross validation performance over the whole training dataset. If the training dataset is large, a small subset can be used for cross validation to decrease computing costs.