International Journal of Research in Engineering, Science and Management
Volume-1, Issue-12, December-2018
www.ijresm.com | ISSN (Online): 2581-5792

Introduction to Data Mining and Machine Learning Algorithms

Ashish Ernest Rahul[1], Satyavani Narukulla[2]
[1]*Software Engineer, Department of Data Analytics, Accenture, Bangalore, India*
[2]*Tech. manager, Department of Data Analytics, Accenture, Bangalore, India*

***Abstract*: Data analytics (DA) is the process of examining data sets in order to draw conclusions about the information they contain, increasingly with the aid of specialized systems and software. Data analytics technologies and techniques are widely used in commercial industries to enable organizations to make more- informed business decisions and by scientists and researchers to verify or disprove scientific models, theories and hypotheses.**

***Keywords*: Terminology, Hypothesis and data analysis**

1. Introduction

At a high level, data analytics methodologies include exploratory data analysis (EDA), which aims to find patterns and relationships in data, and confirmatory data analysis (CDA), which applies statistical techniques to determine whether hypotheses about a data set are true or false. Data analytics can also be separated into quantitative data analysis and qualitative data analysis. The former involves analysis of numerical data with quantifiable variables that can be compared or measured statistically. The qualitative approach is more interpretive it focuses on understanding the content of non-numerical data like text, images, audio and video, including common phrases, themes and points of view. More advanced types of data analytics include data mining, which involves sorting through large data sets to identify trends, patterns and relationships; predictive analytics, which seeks to predict customer behavior, equipment failures and other future events; and machine learning, an artificial intelligence technique that uses automated algorithms to churn through data sets more quickly than data scientists can do via conventional analytical modelling. Big data analytics applies data mining, predictive analytics and machine learning tools to sets of big data that often contain unstructured and semi-structured data. Text mining provides a means of analyzing documents, emails and other text-based content. Data analytics initiatives support a wide variety of business uses. For example, banks and credit card companies analyze withdrawal and spending patterns to prevent fraud and identity theft. E-commerce companies and marketing services providers do clickstream analysis to identify website visitors who are more likely to buy a particular product or service based on navigation and page-viewing patterns. Mobile network operators examine customer data to forecast churn so they can take steps to prevent defections to business rivals; to boost customer relationship management efforts, they and other companies also engage in CRM analytics to segment customers for marketing campaigns and equip call center workers with up-to-date information about callers. Healthcare organizations mine patient data to evaluate the effectiveness of treatments for cancer and other diseases. Data analytics applications involve more than just analyzing data. Particularly on advanced analytics projects, much of the required work takes place upfront, in collecting, integrating and preparing data and then developing, testing and revising analytical models to ensure that they produce accurate results. Once the data that's needed is in place, the next step is to find and fix data quality problems that could affect the accuracy of analytics applications. That includes running data profiling and data cleansing jobs to make sure that the information in a data set is consistent and that errors and duplicate entries are eliminated. Additional data preparation work is then done to manipulate and organise the data for the planned analytics use, and data governance policies are applied to ensure that the data hews to corporate standards and is being used properly.

Data profiling, also called data archeology, is the statistical analysis and assessment of data values within a data set for consistency, uniqueness and logic. The data profiling process cannot identify inaccurate data; it can only identify business rules violations and anomalies. The insight gained by data profiling can be used to determine how difficult it will be to use existing data for other purposes. It can also be used to provide metrics to assess data quality and help determine whether or not metadata accurately describes the source data. Profiling tools evaluate the actual content, structure and quality of the data by exploring relationships that exist between value collections both within and across data sets. For example, by examining the frequency distribution of different values for each column in a table, an analyst can gain insight into the type and use of each column. Cross-column analysis can be used to expose embedded value dependencies and inter-table analysis allows the analyst to discover overlapping value sets that represent foreign key relationships between entities.

A. Data quality

Data quality is a perception or an assessment of data's fitness

to serve its purpose in a given context. The quality of data is determined by factors such as accuracy, completeness, reliability, relevance and how up to date it is. As data has become more intricately linked with the operations of organizations, the emphasis on data quality has gained greater attention. Poor-quality data is often pegged as the source of inaccurate reporting and ill- conceived strategies in a variety of companies, and some have attempted to quantify the damage done. Economic damage due to data quality problems can range from added miscellaneous expenses when packages are shipped to wrong addresses, all the way to steep regulatory compliance fines for improper financial reporting. The demon of poor data quality was particularly common in the early days of corporate computing, when most data was entered manually. Even as more automation took hold, data quality issues rose in prominence. For a number of years, the image of deficient data quality was represented in stories of meetings at which department heads sorted through differing spreadsheet numbers that ostensibly described the same activity. Aspects, or dimensions, important to data quality include: accuracy, or correctness; completeness, which determines if data is missing or unusable; conformity, or adherence to a standard format; consistency, or lack of conflict with other data values; and duplication, or repeated records. As a first step toward data quality, organizations typically perform data asset inventories in which the relative value, uniqueness and validity of data can undergo baseline studies. Established baseline ratings for known good data sets are then used for comparison against data in the organization going forward. Methodologies for such data quality projects include the Data Quality Assessment Framework (DQAF), which was created by the International Monetary Fund (IMF) to provide a common method for assessing data quality. The DQAF provides guidelines for measuring data dimensions that include timeliness, in which actual times of data delivery are compared to anticipated data delivery schedules.

B. Meta data

Metadata is data that describes other data. Meta is a prefix that in most information technology usages means "an underlying definition or description." Metadata summarizes basic information about data, which can make finding and working with particular instances of data easier. For example, author, date created and date modified and file size are examples of very basic document metadata. Having the ability to filter through that metadata makes it much easier for someone to locate a specific document. In addition to document files, metadata is used for images, videos, spreadsheets and web pages. The use of metadata on web pages can be very important. Metadata for web pages contain descriptions of the page's contents, as well as keywords linked to the content. These are usually expressed in the form of Meta tags. The metadata containing the web page's description and summary is often displayed in search results by search engines, making its accuracy and details very important since it can determine whether a user decides to visit the site or not. Meta tags are often evaluated by search engines to help decide a web page's relevance, and were used as the key factor in determining position in a search until the late 1990s. The increase in search engine optimisation (SEO) towards the end of the 1990s led to many websites "keyword stuffing" their metadata to trick search engines, making their websites seem more relevant than others. Since then search engines have reduced their reliance on Meta tags, though they are still factored in when indexing pages. Many search engines also try to halt web pages' ability to thwart their system by regularly changing their criteria for rankings, with Google being notorious for frequently changing their highly-undisclosed ranking algorithms. Metadata can be created manually, or by automated information processing. Manual creation tends to be more accurate, allowing the user to input any information they feel is relevant or needed to help describe the file. Automated metadata creation can be much more elementary, usually only displaying information such as file size, file extension, when the file was created and who created the file.

C. Data scrubbing

Data scrubbing, also called data cleansing, is the process of amending or removing data in a database that is incorrect, incomplete, improperly formatted, or duplicated. An organization in a data-intensive field like banking, insurance, retailing, telecommunications, or transportation might use a data scrubbing tool to systematically examine data for flaws by using rules, algorithms, and look-up tables. Typically, a database scrubbing tool includes programs that are capable of correcting a number of specific type of mistakes, such as adding missing zip codes or finding duplicate records. Using a data scrubbing tool can save a database administrator a significant amount of time and can be less costly than fixing errors manually.

D. Machine learning

Machine learning is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to "learn" from data, without being explicitly programmed. Machine Learning facilitates the progressively improve performance on a specific task. The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders, and computer vision. Within the field of data analytics, machine learning is a method 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.

2. Algorithm

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve. Supervised and semi-supervised learning: Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and a desired output, also known as a supervisory signal. In the case of semi-supervised learning algorithms, some of the training examples are missing the desired output. In the mathematical model, each training example is represented by an array or vector, and the training data by a matrix. Through the process of iteration, supervised learning algorithms develop and optimize a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification. Unsupervised learning: Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms therefore learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses other domains involving summarizing and explaining data features. Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity. Reinforcement learning: Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov Decision Process (MDP). Many reinforcement learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

A. C4.5

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S=S_1,S_2,...$ of already classified samples. Each sample si consists of a p-dimensional vector (x1,i, x2,i,...,xp,i), where the xj represent attribute values or features of the sample, as well as the class in which si falls. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalised information gain (difference in entropy). The attribute with the highest normalised information gain is chosen to make the decision. The C4.5 algorithm then recurses on the partitioned sublists.

B. K-means

K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. The problem is computationally difficult (NP-hard). This algorithm uses cluster centers to model the data. K-means clustering tends to find clusters of comparable spatial extent, while the expectation- maximization mechanism allows clusters to have different shapes. The algorithm has a loose relationship to the k-nearest neighbor classifier, a popular machine learning technique for classification that is often confused with k-means due to the k in the name. Given a set of observations (x1, x2, ..., xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition

the n observations into k (≤ n) sets S = {S1, S2, ..., Sk} so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \|\mathbf{x}-\boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i|\,\mathrm{Var}\,S_i$$

where μ_i is the mean of points in S_i. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

$$\sum_{\mathbf{x}\in S_i} \|\mathbf{x}-\boldsymbol{\mu}_i\|^2 = \sum_{\mathbf{x}\neq\mathbf{y}\in S_i} (\mathbf{x}-\boldsymbol{\mu}_i)(\boldsymbol{\mu}_i-\mathbf{y})$$

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\mathbf{x},\mathbf{y}\in S_i} \|\mathbf{x}-\mathbf{y}\|^2$$

The equivalence can be deduced from identity: Because the total variance is constant, this is also equivalent to maximizing the sum of squared deviations between points in different clusters (between-cluster sum of squares, BCSS), which follows easily from the law of total variance. The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the k-means algorithm.

Given an initial set of k means $m_1(1),...,m_k(1)$, the algorithm proceeds by alternating between two steps:

- *Assignment Step:* Assign each observation to the cluster whose mean has the least squared Euclidean distance, this is intuitively the "nearest" mean.

$$S_i^{(t)} = \left\{x_p : \left\|x_p - m_i^{(t)}\right\|^2 \leq \left\|x_p - m_j^{(t)}\right\|^2 \ \forall j, 1 \leq j \leq k\right\}$$

- where each x_p is assigned to exactly one S(t), even if it could be assigned to two or more of them.
- *Update Step:* Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

The algorithm has converged when the assignments no longer change. There is no guarantee that the optimum is found using this algorithm. The algorithm is often presented as assigning objects to the nearest cluster by distance. Using a different distance function other than (squared) Euclidean distance may stop the algorithm from converging.

C. Initialization methods

Commonly used initialization methods are Forgy and Random Partition. The Forgy method randomly chooses k observations from the data set and uses these as the initial means. The Random Partition method first randomly assigns a cluster to each observation and then proceeds to the update step, thus computing the initial mean to be the centroid of the cluster's randomly assigned points. The Forgy method tends to spread the initial means out, while Random Partition places all of them close to the centre of the data set.

D. Support vector machines

In machine learning, support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non- linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. When data is unlabeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups.

Apriori(T, ϵ)
  $L_1 \leftarrow \{\text{large } 1-\text{itemsets}\}$
  $k \leftarrow 2$
  while $L_{k-1} \neq \emptyset$
    $C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k-1\} \not\subseteq L_{k-1}\}$
    for transactions $t \in T$
      $D_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$
      for candidates $c \in D_t$
        $count[c] \leftarrow count[c] + 1$
    $L_k \leftarrow \{c \mid c \in C_k \wedge \ count[c] \geq \epsilon\}$
    $k \leftarrow k + 1$
  return $\bigcup_k L_k$

E. Apriori

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis. Each transaction is seen as a set of items, known as an item set. Given a threshold C, the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. The pseudo

code for the algorithm is given below for a transaction database T, and a support threshold of ε. Usual set theoretic notation is employed, though note that T is a multiset. Ck is the candidate set for level k. At each step, the algorithm is assumed to generate the candidate sets from the large item sets of the preceding level, heeding the downward closure lemma. count [c] accesses a field of the data structure that represents candidate set c, which is initially assumed to be zero. Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies. The algorithm scans the database too many times, which reduces the overall performance. Due to this, the algorithm assumes that the database is Permanent in the memory. Also the time and space complexity of this algorithm is very high.

F. Expectation-maximization (EM)

The Expectation–maximization (EM) algorithm is an iterative method to find maximum likelihood, or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter- estimates are then used to determine the distribution of the latent variables in the next E step. The EM algorithm is used to find (local) maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. Typically these models involve latent variables in addition to unknown parameters and known data observations. That is, either missing values exist among the data, or the model can be formulated more simply by assuming the existence of further unobserved data points. For example, a mixture model can be described more simply by assuming that each observed data point has a corresponding unobserved data point, or latent variable, specifying the mixture component to which each data point belongs. Finding a maximum likelihood solution typically requires taking the derivatives of the likelihood function with respect to all the unknown values, the parameters and the latent variables, and simultaneously solving the resulting equations. In statistical models with latent variables, this is usually impossible. Instead, the result is typically a set of interlocking equations in which the solution to the parameters requires the values of the latent variables and vice versa, but substituting one set of equations into the other produces an unsolvable equation. The EM algorithm proceeds from the observation that there is a way to solve these two sets of equations numerically. One can simply pick arbitrary values for one of the two sets of unknowns, use them to estimate the second set, then use these new values to find a better estimate of the first set, and then keep alternating between the two until the resulting values both converge to fixed points. It's not obvious that this will work, but it can be proven that in this context it does, and that the derivative of the likelihood is (arbitrarily close to) zero at that point, which in turn means that the point is either a maximum or a saddle point. In general, multiple maxima may occur, with no guarantee that the global maximum will be found. Some likelihoods also have singularities in them, i.e., nonsensical maxima. For example, one of the solutions that may be found by EM in a mixture model involves setting one of the components to have zero variance and the mean parameter for the same component to be equal to one of the data points. Given the statistical model which generates a set X of observed data, a set of unobserved latent data or missing values Z, and a vector of unknown parameters θ, along with a likelihood function $L(\theta;X,Z)=p(X,Z|\theta)$, the maximum likelihood estimate (MLE) of the unknown parameters is determined by maximizing the marginal likelihood of the observed data.

$$L(\theta;X)=p(X|\theta)=\int p(X,Z|\theta)dZ$$

However, this quantity is often intractable, i.e., if Z is a sequence of events, so that the number of values grows exponentially with the sequence length, making the exact calculation of the sum extremely difficult. The EM algorithm seeks to find the MLE of the marginal likelihood by iteratively applying these two steps:

Expectation step (E step): Define $Q(\theta|\theta(t))$ as the expected value of the log likelihood function of θ, with respect to the current conditional distribution of Z given X and the current estimates of the parameters θ(t):

$$Q(\theta|\theta(t))=EZ|X,\theta(t)[\log L(\theta;X,Z)]$$

Maximization step (M step): Find the parameters that maximize this quantity: $\theta^{(t+1)} = \arg\max Q(\theta|\theta^{(t)})$

G. Pagerank

The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value. A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50% chance" of something happening. Hence, a PageRank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to the document with the 0.5 PageRank. Assume a small universe of four web pages: A, B, C and D. Links from a page to itself be ignored. Multiple outbound links from one page to another page are treated as a single link. PageRank is initialized to the same value for all pages. In the original form of PageRank, the sum of PageRank over all pages was the total number of pages on the web at that time, so each page in this example would have an initial value of 1. However, later versions of PageRank, and the remainder of this section, assume

a probability distribution between 0 and 1. Hence the initial value for each page in this example is 0.25.

The PageRank transferred from a given page to the targets of its outbound links upon the next iteration is divided equally among all outbound links. If the only links in the system were from pages B, C, and D to A, each link would transfer 0.25 PageRank to A upon the next iteration, for a total of 0.75.

PR(A) = PR(B) + PR(C) + PR(D) .

Suppose instead that page B had a link to pages C and A, page C had a link to page A, and page D had links to all three pages. Thus, upon the first iteration, page B would transfer half of its existing value, or 0.125, to page A and the other half, or 0.125, to page C. Page C would transfer all of its existing value, 0.25, to the only page it links to, A. Since D had three outbound links, it would transfer one third of its existing value, or approximately 0.083, to A. At the completion of this iteration, page A will have a PageRank of approximately 0.458.

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}$$

In other words, the PageRank conferred by an outbound link is equal to the document's own PageRank score divided by the number of outbound links L ( ).

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}$$

In the general case, the PageRank value for any page u can be expressed as:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$

H. Adaboost

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithm is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner. When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative 'hardness' of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.

AdaBoost refers to a particular method of training a boosted classifier. A boost classifier is a classifier in the form:

$$F_T(x) = \sum_{t=1}^{T} f_t(x)$$

Where each f_t is a weak learner that takes an object x as input and returns a value indicating the class of the object. For example, in the two class problem, the sign of the weak learner output identifies the predicted object class and the absolute value gives the confidence in that classification. Similarly, the Tth classifier is positive if the sample is in the positive class and negative otherwise. Each weak learner produces an output hypothesis, h(xi)h(xi), for each sample in the training set. At each iteration t, a weak learner is selected and assigned a coefficient αt such that the sum training error Et of the resulting t-stage boost classifier is minimized.

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)]$$

Here Ft−1(x) is the boosted classifier that has been built up to the previous stage of training, E(F) is some error function and ft(x)=αth(x)ft(x)=αth(x) is the weak learner that is being considered for addition to the final classifier. At each iteration of the training process, a weight wi,t is assigned to each sample in the training set equal to the current error E(Ft−1(xi)) on that sample. These weights can be used to inform the training of the weak learner, for instance, decision trees can be grown that favor splitting sets of samples with high weights.

3. Deep learning

Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Deep learning is a class of machine learning algorithms that:

- Use a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.
- Learn in supervised (e.g., classification) and/or unsupervised (e.g., pattern analysis) manners.
- Learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

Most modern deep learning models are based on an artificial neural network, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in deep belief networks and deep Boltzmann machines. In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the

fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own. The "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial credit assignment path (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one. For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited. No universally agreed upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth > 2. CAP of depth 2 has been shown to be a universal approximator in the sense that it can emulate any function. Beyond that more layers do not add to the function approximator ability of the network. Deep models (CAP > 2) are able to extract better features than shallow models and hence, extra layers help in learning features. For supervised learning tasks, deep learning methods obviate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures that remove redundancy in representation. Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabelled data are more abundant than labeled data. Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief networks.

4. Applications of machine learning

A. Virtual personal assistants

Siri, Alexa, Google Now are some of the popular examples of virtual personal assistants. As the name suggests, they assist in finding information, when asked over voice. For answering, your personal assistant looks out for the information, recalls your related queries, or send a command to other resources (like phone apps) to collect info. Machine learning is an important part of these personal assistants as they collect and refine the information on the basis of your previous involvement with them. Later, this set of data is utilized to render results that are tailored to your preferences.

B. Prediction while commuting

Traffic Predictions: We all have been using GPS navigation services. While we do that, our current locations and velocities are being saved at a central server for managing traffic. This data is then used to build a map of current traffic. While this helps in preventing the traffic and does congestion analysis, the underlying problem is that there are less number of cars that are equipped with GPS. Machine learning in such scenarios helps to estimate the regions where congestion can be found on the basis of daily experiences. Online Transportation Networks: When booking a cab, the app estimates the price of the ride. When sharing these services, how do they minimize the detours? The answer is machine learning. In the entire cycle of the services, ML is playing a major role.

C. Social media services

From personalizing your news feed to better ads targeting, social media platforms are utilizing machine learning for their own and user benefits. People You May Know: Machine learning works on a simple concept: understanding with experiences. Facebook continuously notices the friends that you connect with, the profiles that you visit very often, your interests, workplace, or a group that you share with someone etc. On the basis of continuous learning, a list of Facebook users are suggested that you can become friends with.

- *Face Recognition:* You upload a picture of you with a friend and Facebook instantly recognizes that friend. Facebook checks the poses and projections in the picture, notice the unique features, and then match them with the people in your friend list. The entire process at the backend is complicated and takes care of the precision factor but seems to be a simple application of ML at the front end.
- *Similar Pins:* Machine learning is the core element of Computer Vision, which is a technique to extract useful information from images and videos. Pinterest uses computer vision to identify the objects (or pins) in the images and recommend similar pins accordingly.

D. Email spam and malware filtering

There are a number of spam filtering approaches that email clients use. To ascertain that these spam filters are continuously updated, they are powered by machine learning. When rule-based spam filtering is done, it fails to track the latest tricks adopted by spammers. Multi-Layer Perceptron, C 4.5 Decision Tree Induction are some of the spam filtering techniques that are powered by ML.

E. Search engine refinement

Google and other search engines use machine learning to improve the search results for you. Every time you execute a search, the algorithms at the backend keep a watch at how you respond to the results. If you open the top results and stay on the web page for long, the search engine assumes that the results it displayed were in accordance to the query. Similarly, if you reach the second or third page of the search results but do not open any of the results, the search engine estimates that the results served did not match requirement. This way, the algorithms working at the backend improve the search results.

F. Product recommendations

You shopped for a product online few days back and then you keep receiving emails for shopping suggestions. If not this, then you might have noticed that the shopping website or the

app recommends you some items that somehow matches with your taste. Certainly, this refines the shopping experience but did you know that it's machine learning doing the magic for you? On the basis of your behavior with the website/app, past purchases, items liked or added to cart, brand preferences etc., the product recommendations are made.

5. Conclusion

We haven't tapped into the full potential of what Machine Learning has to offer. It a domain with lots of promise. Everyday something new takes place in this field. Scientists and engineers alike push the boundaries of what they know and what they can achieve using ML techniques. ML could be the foundation of a Sci-Fi world. There is a lot of scope for research in this domain.

References

[1] Wu, X., Kumar, V., Ross Quinlan, J. et al., "Top 10 algorithms in data mining," in Knowl Inf Syst (2008) 14: 1
[2] S. Yogasudha, K. Mounika, P. R. Namitha, K. Merlin Rathina Priya "Deep Learning", International Journal of Research in Engineering, Volume-1, Issue-10, October-2018.
[3] Hand DJ, Manila H, Smyth P. Principles of data mining. Cambridge (MA): The MIT Press, 2001
[4] https://en.wikipedia.org/wiki/Deep_learning
[5] https://en.wikipedia.org/wiki/Machine_learning