International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-6, March 2020

Machine Learning Techniques: The Need of the Hour

Parag Nijhawan, Vinod Kumar Bhalla, Jyoti Gupta, Manish Kumar Singla

Abstract : Scientists, researchers and business analyst need to process large volume of data to convert the data to meaningful information. Historically all the professionals are doing this manually and later with the invention of computers by developing algorithms and coding programs. Eventually size of data is growing with the advancement of social economic developments and advancement of technological sectors. This posed greater challenge to the technocrats. So there was demand to automatically process and analyze the data. Scientists started working towards the direction of artificial intelligence (AI). They developed the branch of AI as machine learning (ML). Large number of areas are using these techniques and getting benefit of these models. Financial data analyst, business analyst, medical researchers, software professionals are applying machine learning to increase the speed of large amount of data for decision making by drawing patterns with minimal and no human intervention. The purpose of this paper is to discuss a few widely publicized ML algorithms and their advantages and benefits.

Keywords: Machine learning, pseudo code, algorithm, learning's.

I. INTRODUCTION

Machine learning (ML) is an area in computer science which involves teaching computers to do thing naturally by learning through experience. This means that the computer system is turning data into information [1]. Machine learning algorithms derive information from data without the help of any model. The algorithms adaptively improve their performance as we increase for example, the number of tuples in the dataset, it is equivalent to the fact that learning happens with experience. Initially supervised machine learning methods were developed. Success of the supervised methods encouraged researchers to develop semi supervised techniques.

Manuscript received on March 15, 2020.
Revised Manuscript received on March 24, 2020.
Manuscript published on March 30, 2020.
* Correspondence Author
Parag Nijhawan*, Associate Professor, Department of Electrical and Instrumentation Engineering, Thapar Institute of Engineering and Technology, India
Vinod Kumar Bhalla, Assistant Professor, Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, India
Jyoti Gupta, PhD Scholar, Department of Electrical and Instrumentation Engineering, Thapar Institute of Engineering and Technology, India.
Manish Kumar Singla, PhD Scholar, Department of Electrical and Instrumentation Engineering, Thapar Institute of Engineering and Technology, India.

Research in the field of ML has entered in the era of unsupervised learning methods to automatically analyzed the data and automatically build the model by learning from the data samples to work on the actual data to predict the results. Machine learning techniques are categorized shown in figure 1.

Major Learning Methods

A. Supervised Learning

The human intervention based algorithms are known as supervised machine learning. These algorithms require assistance from the human programmer. First, input data is selected for specific domain. Then data is filter for noise and unwanted elements. Second step involves extract of features from these data set which are specially identified for the problem in hand. In third step data set is divided into training data used to train the model and the test data to evaluate the performance of model. This model can be developed for prediction, recognition or classification. [4]. As shown in figure 2 the popular supervised algorithms are discussed.

Decision Tree:

This method is mainly used for classification purpose. The technique implied here is grouping of attributes through sorting depending on their initial values. Tree consist node which represent the attributes that is to be categorized and the branch is used for value that node can have. [4] as shown in Figure 3. The Decision tree pseudo code is discussed in figure 5.

Figure 1 Machine learning types [2-3]

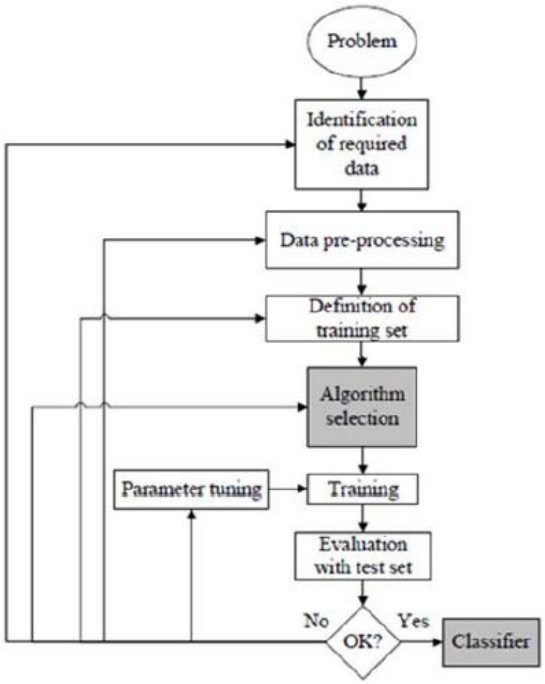

Figure 2 Machine learning flow chart [4]

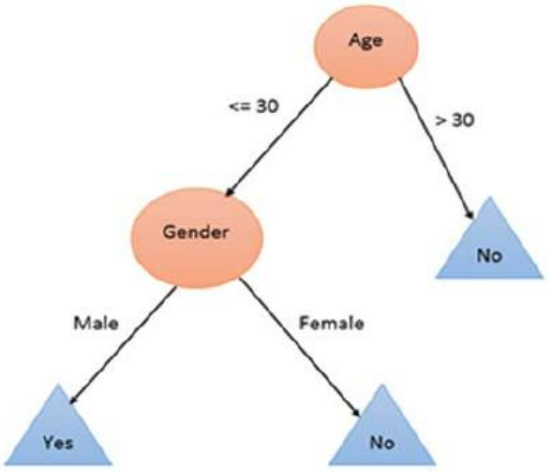

Figure 3 Decision tree [5]

Naive Baye's (NB):

The naive Bayes classifier, or simple Bayesian Classifier as shown in figure 4 and pseudo code in figure 6, works as follows: Let D is set of tuples where each tuple is an n-dimensional attribute vector given by x: (x_1, x_2, x_3 ….x_n) [6]. Let there be m classes: C_1, C_2, C_3 ….C_m. Naïve Bayes classifier predicts X belongs to class C_i if

$$P\left(\frac{C_i}{X}\right) > P\left(\frac{C_j}{X}\right) \; for \; 1 \le j \le m, j, i$$

The maximum Posteriori Hypothesis is given as

$$P\left(\frac{C_i}{X}\right) = P\left(\frac{X}{C_i}\right)\frac{P(C_i)}{P(X)}$$

This reduces to this

Maximize $P\left(\frac{X}{C_i}\right)P(C_i)$ as $P(X)$ is constant

With many attributes, it is computationally expensive to evaluate $P(X/C_i)$. For n attributes, we follow the naïve assumption of "class conditional independence", which is given eq.

$$P\left(\frac{X}{C_i}\right) = \prod_{k=1}^{n} P\left(\frac{x_k}{C_i}\right)$$

$$P\left(\frac{X}{C_i}\right) = P\left(\frac{x_1}{C_i}\right) \times P\left(\frac{x_2}{C_i}\right) \times \ldots \times P\left(\frac{x_n}{C_i}\right)$$

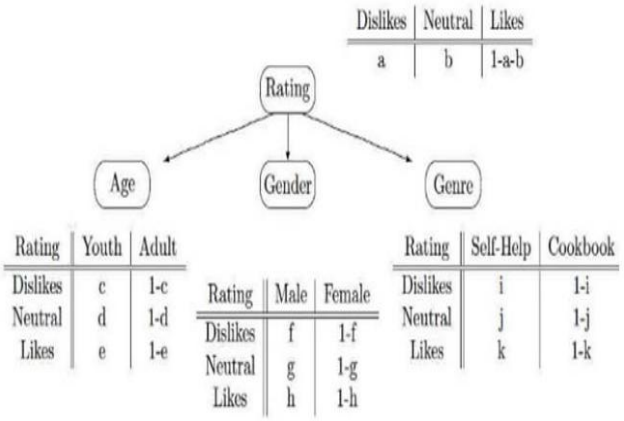

Figure 4 Example of NB [7]

Support Vector Machine (SVM)

Support vector machines was introduced in early 1990 and falls under category of supervised machine learning algorithms. The basic idea of support vector machine is to find the optimal separating hyperplane which maximizes the margin of the training data.Basically SVM uses the mapping function that transforms data in input space to data in feature space in such a way as to render a problem linearly separable. The working model and pseudo code of support vector machine is shown in figure 7 and 8. The SVMs then automatically discover the optimal separating hyperplane, which is nothing but a complex decision surface.

procedure $DTInducer(S, A, y)$
1: $T = TreeGrowing(S, A, y)$
2: Return TreePruning(S,T)
procedure $TreeGrowing(S, A, y)$
1: Create a tree T
2: **if** One of the Stopping Criteria is fulfilled **then**
3: Mark the root node in T as a leaf with the most common value of y in S as the class.
4: **else**
5: Find a discrete function $f(A)$ of the input attributes values such that splitting S according to $f(A)$'s outcomes $(v_1, \ldots, v_n)$ gains the best splitting metric.
6: **if** best splitting metric $\geq$ treshold **then**
7: Label the root node in T as $f(A)$
8: **for** each outcome v_i of f(A) **do**
9: $Subtree_i = TreeGrowing(\sigma_{f(A)=v_i} S, A, y)$.
10: Connect the root node of T to $Subtree_i$ with an edge that is labelled as v_i
11: **end for**
12: **else**
13: Mark the root node in T as a leaf with the most common value of y in S as the class.
14: **end if**
15: **end if**
16: Return T
procedure $TreePruning(S, T, y)$
1: **repeat**
2: Select a node t in T such that pruning it maximally improve some evaluation criteria
3: **if** $t \neq \emptyset$ **then**
4: $T = pruned(T, t)$
5: **end if**
6: **until** t=Ø
7: Return T

Figure 5 Pseudo code of decision tree [5]

Unsupervised Learning

In unsupervised learning, the data is unlabeled, so the learning algorithm is left to find commonalities among its input data. The goal of unsupervised learning is to discover hidden patterns within dataset, which allows the computational machine to automatically discover the representations that are needed to classify raw data. Mostly, cluster analysis technique is used in these techniques and is shown through example workflow as shown in fig 9.

Clustering is the process of grouping together data objects into multiple sets or clusters, so that objects within a cluster have high similarity as compared to objects outside of it. The partitioning of cluster is done with help of algorithm. The most commonly used algorithms for clustering are:

INPUT: training set T, hold-out set H, initial number of components k_0, and convergence thresholds δ_{EM} and δ_{Add}.

Initialize M with one component.
$k \leftarrow k_0$
repeat
  Add k new mixture components to M, initialized using k random examples from T.
  Remove the k initialization examples from T.
  repeat
    E-step: Fractionally assign examples in T to mixture components, using M.
    M-step: Compute maximum likelihood parameters for M, using the filled-in data.
    If $\log P(H|M)$ is best so far, save M in M_{best}.
    Every 5 cycles, prune low-weight components of M.
  until $\log P(H|M)$ fails to improve by ratio δ_{EM}.
  $M \leftarrow M_{best}$
  Prune low weight components of M.
  $k \leftarrow 2k$
until $\log P(H|M)$ fails to improve by ratio δ_{Add}.
Execute E-step and M-step twice more on M_{best}, using examples from both H and T.
Return M_{best}.

Figure 6 Pseudo code of NB [6]

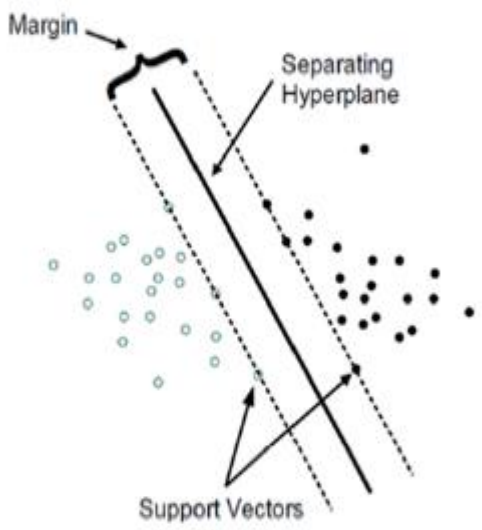

Figure 7 Support vector machine working model [8]

1) K-Means Clustering*:* The k-means is most well-known clustering algorithm [11]. The main concept is to define k cluster centers. This algorithm aims at minimizing an objective function know as squared error function, which is given by

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} (\|x_i - v_j\|)^2$$

Where $\|x_i - v_j\|$ is the Euclidean distance between x_i and v_j.

c_i is the number of data points in the i^{th} cluster.

c is the number of cluster centers.

The objective function aims for high intra-cluster similarity and low inter-cluster similarity. This function tries to make the resulting k clusters as compact and as separate as possible. To overcome the prohibitive computational cost for exact solution, greedy approaches are often used in practice clustered data looks like figure 10 and figure 11 discuss the algorithm for k-means.

INPUT: S, λ, T, k

INITIALIZE: Choose $\mathbf{w}_1$ s.t. $\|\mathbf{w}_1\| \leq 1/\sqrt{\lambda}$

FOR $t = 1, 2, \ldots, T$

  Choose $A_t \subseteq S$, where $|A_t| = k$

  Set $A_t^+ = \{(\mathbf{x}, y) \in A_t : y \langle \mathbf{w}_t, \mathbf{x} \rangle < 1\}$

  Set $\eta_t = \frac{1}{\lambda t}$

  Set $\mathbf{w}_{t+\frac{1}{2}} = (1 - \eta_t \lambda)\mathbf{w}_t + \frac{\eta_t}{k} \sum_{(\mathbf{x},y) \in A_t^+} y\mathbf{x}$

  Set $\mathbf{w}_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+\frac{1}{2}}\|}\right\} \mathbf{w}_{t+\frac{1}{2}}$

OUTPUT: $\mathbf{w}_{T+1}$

Figure 8 Pseudo code of SVM [9]

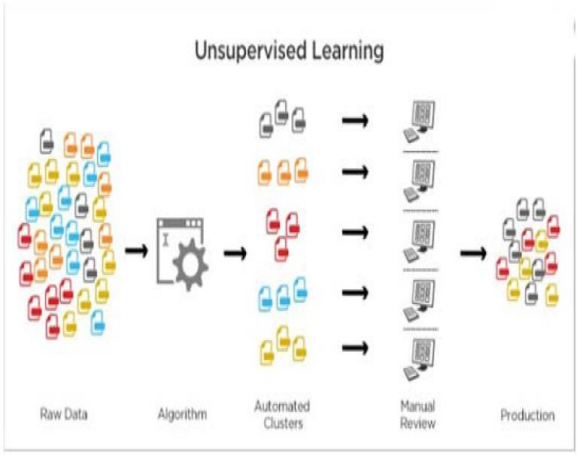

Figure 9 Unsupervised learning example [10]

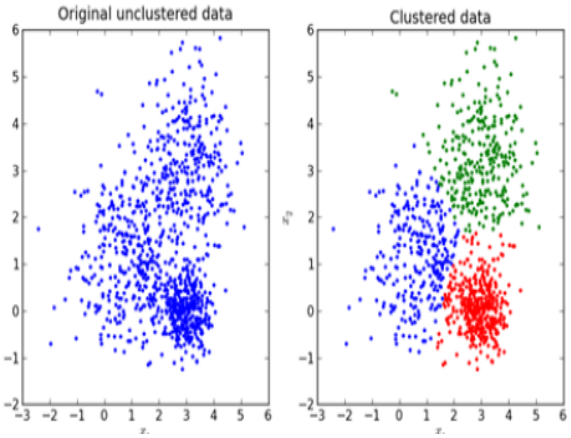

Figure 10 clustered data [12]

function Direct-k-means()

Initialize k prototypes $(w_1, \ldots, w_k)$ such that $w_j = i_l,\ j \in \{1, \ldots, k\},\ l \in \{1, \ldots, n\}$

Each cluster C_j is associated with prototype w_j

Repeat

for each input vector i_l, where $l \in \{1, \ldots, n\}$, *do*

Assign i_l to the cluster C_{j*} with nearest prototype w_{j*} (i.e., $| i_l - w_{j*} | \leq | i_l - w_j |, j \in \{1, \ldots, k\}$)

for each cluster C_j, where $j \in \{1, \ldots, k\}$, *do*

Update the prototype w_j to be the centroid of all samples currently in C_j, so that $w_j = \sum_{i_l \in C_j} i_l / | C_j |$

Compute the error function:

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} | i_l - w_j |^2$$

Until E does not change significantly or cluster membership no longer changes

Figure 11 pseudo code clustering K-means [13]

PCA learning

PCA learning is also known as PAC learning is "probably approximate learning". The analysis will be easier if we build it around a few simplifying assumptions. First, examples considered are noise free. Second, all attributes are discrete. Third, the classifier acquires the form of a logical expression of attribute values. And, finally there exist at least one expression that correctly classifies all training examples. Example of PCA is explained in figure 12. The pseudo code for PCA is given in figure 13.

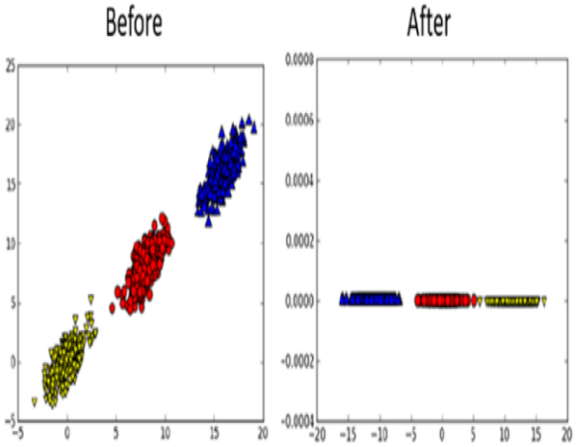

Figure 12 example-PCA [11]

$\mathbf{R} \leftarrow \mathbf{X}$
for$(k = 0, \ldots, K-1)$ do
{
$\lambda = 0$
$\mathbf{T}^{(k)} \leftarrow \mathbf{R}^{(k)}$
for$(j = 0, \ldots, J)$ do
{
$\mathbf{P}^{(k)} \leftarrow \mathbf{R}^T \mathbf{T}^{(k)}$
$\mathbf{P}^{(k)} \leftarrow \mathbf{P}^{(k)} \left\| \mathbf{P}^{(k)} \right\|^{-1}$
$\mathbf{T}^{(k)} \leftarrow \mathbf{R}\mathbf{P}^{(k)}$
$\lambda' \leftarrow \left\| \mathbf{T}^{(k)} \right\|$
if$(|\lambda' - \lambda| \leq \varepsilon)$ then break
$\lambda \leftarrow \lambda'$
}
$\mathbf{R} \leftarrow \mathbf{R} - \mathbf{T}^{(k)} (\mathbf{P}^{(k)})^T$
}
return T, P, R

Figure 13 pseudo code for PCA [14]

Semi - Supervised Learning

Combination of other two learning i.e. supervised learning and unsupervised learning defines semi-supervised learning. It is one of recent emerged part of artificial learning. Main aim of this technique is to identify the unlabeled data through the corresponding labeled data set [15-16]. Google expander is one of the prime examples of semi supervised learning. A few semi supervised learning methods as discussed for your reference.

1) Generative Models*:* Generative models is based on the following formula and it considered as oldest of all semi-supervised learning method. *P(a,b) = p(b)p(a|b) :p(a|b)* is a mixed distribution. The mixed components models could be identified using labeled data. In this learning technique one labeled training example for each component is enough to generate the mixted distribution.

2) Self-Training:- In this learning method, the model is trained with a some samples of labeled data and for this purpose a classifier is used. Further the classifier can be trained with the help of unlabeled data. The unlabeled data and predicted lables are combined repeatedly to generate a refined model own its own. This justifies the name self training and learning.

3) TSVM known as Transductive support vector machine. This method is developed by extending the capabilities of Support vector machine (SVM). Both labeled and unlabeled data sets are used in this technique. Finding the proper solution through TSVM is a NP-hard problem. So it uses labled and unlabeled data to find the maximum margin.

Reinforcement Learning

Reinforcement learning method is concerned inspired with behaviorist psychology. It is inspired by the concept of reward if the sw agents take action in manner to gain the maximum cumulative reward.

in an environment. The problem of *Reinforcement Learning* is studied in other fields also, statistics and genetic game theory, , information theory, multi-agent systems, control theory, simulation-based optimization, operations research and swarm intelligence[17]. It is also called the dynamic programming in the area of operations research and control.. The reinforcement learning general purpose model [18] is shown in figure 14.

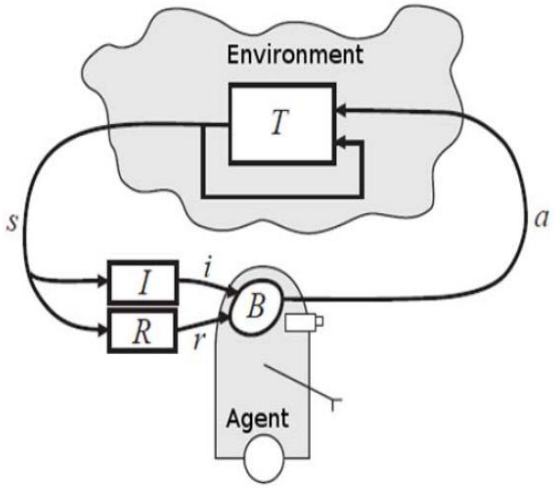

Figure 14 Model of reinforcement learning [18]

the SW agent gets the input as *i*, the current state as *s*, the state transition as *r* and the operational input function as *I* from the environment in which agent operates. On the basis of these values it generates a behavior as *B* and it is forced to take an action as *a* this is implied as an outcome.

E. Multitask Learning

Main idea of Multitask learning is to help every learner to improve or to perform more efficiently. Each and every step applied in solving a particular task is remembered, which is biggest assets of multitask learning. Latter these steps can be used by algorithm to solve the similar types of problems in future. Inductive transfer mechanism term is defined as using same codes by other algorithm. If each learner shares its experiences with other learner, learners can learn from each other and change quickly, rather than individually [19].

F. Ensemble Learning

In this method strategically generated multiple models, classifiers and experts, are combined to solve a complex intelligence domain of problem in hand. This process improves the performance of a model, and reduce the chance of an unwanted poor selections [20]. selecting optimal features, error-correcting, data fusion, assigning a confidence to the decision, non-stationary learning and incremental learning othe rmajor applications of ensemble learning [21].

1) Boosting:

Boosting is one of technique of ensemble learning, mainly function to reduce variance and bias. It obtains the one strong learner by training the collection of weak learner. A weak learner can be defined as a form of classifier which cannot be related with any defined classification whereas strong learner is vice versa of weak learner. Figure 15 describe pseudo code of the most popular example of boosting. Known as Ada Boost.

Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds T.

Process:
$D_1(i) = 1/m$.
for $t = 1, \cdots, T$:
$h_t = \mathcal{L}(\mathcal{D}, D_t)$;
$\epsilon_t = \Pr_{i \sim D_i}[h_t(\boldsymbol{x}_i) \neq y_i)]$;
$\alpha_t = \frac{1}{2}\ln \frac{1-\epsilon_t}{\epsilon_t}$;

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) \neq y_i \end{cases} = \frac{D_t(i)\exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$$

end.

Output: $H(\boldsymbol{x}) = \text{sign}\,(f\,(\boldsymbol{x})) = \text{sign}\ \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$

Figure 15 pseudo code-Boosting [21]

2) Bagging:

Bagging or bootstrap is another technique of ensemble learning. Wherever stability and accuracy of the problem is the prime objective there bagging or bootstrap can be used [22]. It can be applied in regression and classification type of problem. It is also very effective in reducing variance. Figure 16 is used to describe pseudo code corresponding to bagging algorithm.

Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds T.

Process:
for $t = 1, \cdots, T$:
$\mathcal{D}_t = Bootstrap(\mathcal{D})$;
$h_t = \mathcal{L}(\mathcal{D}_t)$
end.

Output: $H(\boldsymbol{x}) = \text{argmax}_{y \in \mathcal{Y}} \sum_{t=1}^{T} 1(y = h_t\,(\boldsymbol{x}))$

Figure 16 pseudo code-Bagging [23]

G. Neural Network Learning

A neural network's ability to perform computations is based on the hope that we can reproduce some of the flexibility and power of the human brain by artificial means. It tries to mimic the structure and function of our nervous system. An artificial neural network (ANN) is used as a methodology for information processing and the method got its inspiration from biological nervous system. Five parts of neuron are dendrites, soma, nucleus, synaptic terminals and axon as shown in figure 17.

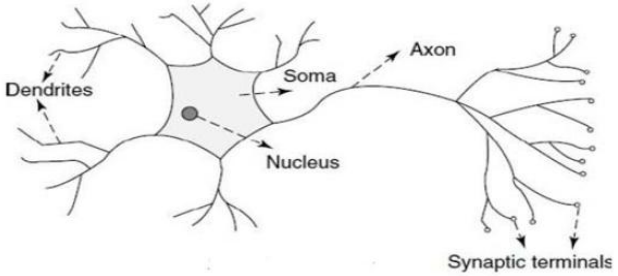

Figure 17 Structure of neuron [24]

Dendrites are structures used for collecting input for a neuron. It collects and sums up the inputs and if the result is greater than its firing threshold, the neuron fires else it inhibits. When a neuron fires, it sends an electrical impulse from its nucleus to synaptic terminals. The synaptic terminals can then network to more neurons via connections called synapses.

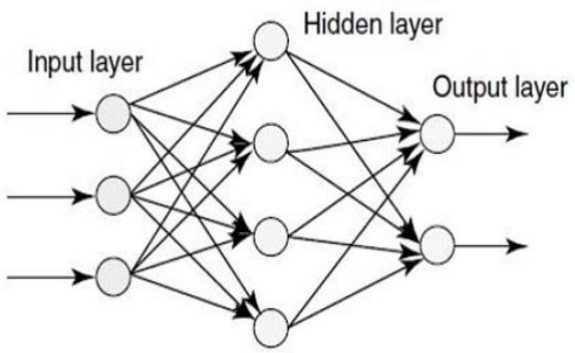

Figure 18 Hidden layer

Learning takes place by changing the effectiveness of the synapses so that the influence of one neuron on another changes. An artificial neural network can be divided in two three parts. First is input layer, as the name suggests it accepts the input data. Second is hidden layer, it processes the data received from input layer and third is output layer, it displays the processed data as sown in figure 18. Three types of artificial neural network (ANN) are discussed in literature known as supervised(S), unsupervised(U) and reinforcement(R) [25].

1) Supervised Neural Network:

Supervised learning, is employed by using each input vector associated with the output which is desired. The input vector and the corresponding output vector results in a training pair. Here, the network knows what should be the output. During training, the input vector is given is given to the network to produce an output. This output is the actual output. Then this actual output is checked whether it is same as the desired output. The block diagram of supervised learning algorithm is shown in figure 19. The difference between the actual and desired output is considered as the error signal and is generated by the network. This error signal can be used to adjust the weights of the network layers so that for all training pair the actual output becomes the desired output.

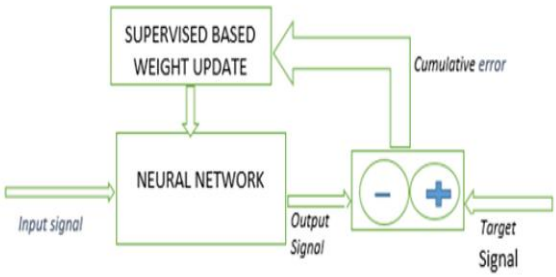

Figure 19 Schematic of supervised network [25]

2) Unsupervised Neural Network:

Unsupervised learning, the inputs of a similar category are grouped together without the help of any training. The network clubs together the similar input patterns to form clusters in the training process. When a new input is applied, the network gives an output response indicating the class to which it belongs. If an input does not belong to any cluster, *a* new cluster is formed. The schematic diagram is shown in figure 20.

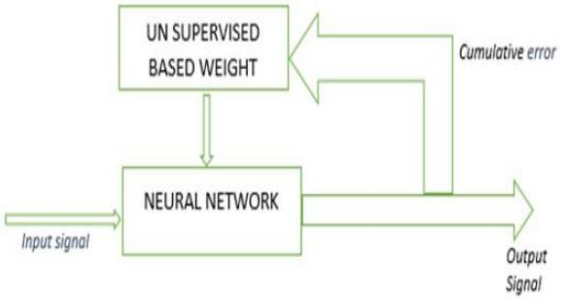

Figure 20 Schematic of unsupervised learning [25]

3) Reinforced Neural Network*:* Reinforcement learning is similar to supervised learning in that information is available. However, in case of reinforcement learning, only critical information is available. The exact information needs to be obtained from this critical information. The process of extracting real information from critical information is termed as reinforcement learning. Reinforced neural network is represented in figure 21.

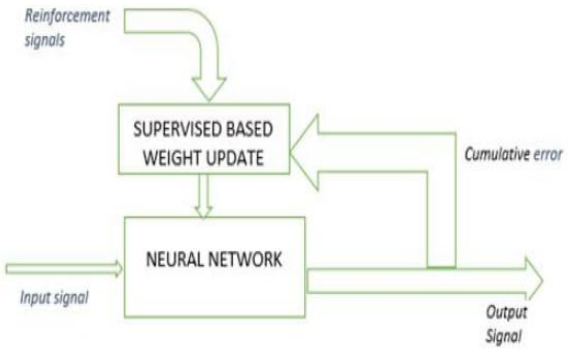

Figure 21 Schematic of reinforced learning [25]

H. Instance-Based Learning

A unique type of pattern is followed in instant-based learning which is different from other type of learning. As the name suggests the similar type of pattern is applied to every data. In this technique complexity of the algorithm also increases as size of data increases. Sometime it is also considered lazy method because it takes long intervals to iterate the test data with the training data [26].

1) K-Nearest Neighbor (KNN):

The k-nearest neighbor, or KNN, is non-parametric, and slow learning algorithm. KNN deploy data set to separate the data points into many categories to predict the class of a test case. KNN model structure is determined from the data. This algorithm is not parametric because no assumptions are made from input data distribution [27]. In many applications data does not comply with the theoretical assumptions. So when there is no prior information about the distribution of data, KNN becomes the obvious option to classify the data.

Figure 22 is used to describe the KNN pseudo code.

Let $W = \{x_1, x_2, \cdots, x_n\}$ be a set of n labeled samples. The algorithm is as follows:

```
BEGIN
  Input y, of unknown classification.
  Set K, 1 ≤ K ≤ n.
  Initialize i = 1.
  DO UNTIL (K-nearest neighbors found)
    Compute distance from y to x_i.
    IF (i ≤ K) THEN
      Include x_i in the set of K-nearest neighbors
    ELSE IF (x_i is closer to y than any previous nearest
      neighbor) THEN
      Delete farthest in the set of K-nearest neighbors
      Include x_i in the set of K-nearest neighbors.
    END IF
    Increment i.

  END DO UNTIL
  Determine the majority class represented in the set of K-
  nearest neighbors.
  IF (a tie exists) THEN
    Compute sum of distances of neighbors in each class
    which tied.
    IF (no tie occurs) THEN
      Classify y in the class of minimum sum
    ELSE
      Classify y in the class of last minimum found.
    END IF
  ELSE
    Classify y in the majority class.
  END IF
END
```

Figure 22 KNN-Pseudo code [28]

II. CONCLUSION

The primary aim of this paper is to survey and concisely describe various machine learning algorithms and techniques for easy reference & learning. It has wide range of applications in large number of fields with success. Scientists developed number of supervised, semi supervised and unsupervised techniques for data analysis and decision making. Aim of these techniques is to reduce the human intervention. Main idea behind ML methods is to learn from the existing data, identify the patterns and draw conclusion based on automated build data model. In supervised methods features are manually extracted where as in unsupervised methods features are automatically extracted from the input data samples and hence reduce human efforts. The ability of ML is to automatically apply complex mathematical calculations, to huge amount of data again and again with more faster and efficient manner with minimal human intervention. Today, every person is using machine learning knowingly or unknowingly. It ranges from email classification, webpage classification for easy search and retrival of information to design chat boat and assistant for online shopping site. This paper attempted to summarize the machine learning for benefit of the society at large.

REFERENCES

1. Coelho, L. P., Richert, W., & Brucher, M. (2018). Building Machine Learning Systems with Python: Explore machine learning and deep learning techniques for building intelligent systems using scikit-learn and TensorFlow. Packt Publishing Ltd.
2. Welling, M. (2011). A first encounter with Machine Learning. Irvine, CA.: University of California, 12.
3. Bowles, M. (2015). Machine learning in Python: essential techniques for predictive analysis. John Wiley & Sons.
4. Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160, 3-24.
5. Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 35(4), 476-487..
6. Lowd, D., & Domingos, P. (2005, August). Naive Bayes models for probability estimation. In Proceedings of the 22nd international conference on Machine learning (pp. 529-536).
7. https://webdocs.cs.ualberta.ca/~greiner/C651/Homework2_Fall2008.html
8. Meyer, D. (2004). Support vector machines: The interface to libsvm in package e1071.
9. Shalev-Shwartz, S., Singer, Y., Srebro, N., & Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for svm. Mathematical programming, 127(1), 3-30.
10. http://www.simplilearn.com/what-is-machine-learning-and-why-itmatters- article.
11. Harrington, P. (2012). Machine learning in action. Manning Publications Co..
12. http://pypr.sourceforge.net/kmeans.html.
13. Alsabti, K., Ranka, S., & Singh, V. (1997). An efficient k-means clustering algorithm.
14. Andrecut, M. (2009). Parallel GPU implementation of iterative PCA algorithms. Journal of Computational Biology, 16(11), 1593-1599.
15. Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1), 1-130.
16. Zhu, X. J. (2005). Semi-supervised learning literature survey. University of Wisconsin-Madison Department of Computer Sciences..
17. Dey, A. (2016). Machine learning algorithms: a review. International Journal of Computer Science and Information Technologies, 7(3), 1174-1179..
18. Wolpert, D. H., Sill, J., & Turner, K. (2000). Reinforcement learning in distributed domains: Beyond team games.
19. Caruana, R. (1997). Multitask learning. Machine learning, 28(1), 41-75.
20. Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. Journal of artificial intelligence research, 11, 169-198.
21. Zhou, Z. H. (2009). Ensemble Learning. Encyclopedia of biometrics, 1, 270-273.
22. https://en.wikipedia.org/wiki/Boosting_(machine_learning).
23. https://en.wikipedia.org/wiki/Bootstrap_aggregating.
24. Dey, A. (2016). Machine learning algorithms: a review. International Journal of Computer Science and Information Technologies, 7(3), 1174-1179.
25. Hiregoudar, S. B., Manjunath, K., & Patil, K. S. (2014). A survey: research summary on neural networks. International Journal of Research in Engineering and Technology, 3(15), 385-389.
26. https://en.wikipedia.org/wiki/Instance_based_learning.
27. Harrington, P. (2012). Machine learning in action. Manning Publications Co.
28. Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. IEEE transactions on systems, man, and cybernetics, (4), 580-585.

AUTHOR'S PROFILE

[1,3,4] EIE Department, Thapar Institute of Engineering and Technology, Patiala (INDIA).
[2]CSE Department, Thapar Institute of Engineering and Technology, Patiala (INDIA).

Dr. Parag Nijhawan, is presently Associate Professor in the Electrical and Instrumentation Engineering Department at Thapar Institute of Engineering and Technology, India. He received his B.E. and M.E. degrees in Electrical Engineering from the Punjab Technical University and Punjab Engineering College in India, respectively. He did his PhD. in Electrical Engineering from National Institute of Technology, Kurukshetra. He has more than 19 years of work experience that includes teaching and research. His research focus includes renewable energy sources, power quality improvement, grounding and FACTS devices.
E-mail: parag.nijhawan@rediffmail.com

Dr. Vinod Kumar Bhalla is currently working as an Assistant Professor in the Computer Science and Engineering Department at Thapar Institute of Engineering and Technology, India. He has experience of more than 15 years in both Industry and Academia. He has taught many courses to UG/PG in his area of expertise in web technologies. He has guided many thesis leading to M.Tech, and M.E. of different streams.
E-mail:
vkbhalla@thapar.edu

Jyoti Gupta is PhD Scholar in the Electrical and Instrumentation Engineering Department at Thapar Institute of Engineering and Technology, India. She received her B.E. and M.E. degrees in Electrical Engineering from the Punjab Technical University and Thapar Institute of Engineering and Technology, in India, respectively. Her current fields of interest include power systems, Artificial intelligence, Renewable Energy.
E-mail: jg118207@gmail.com

Manish Kumar Singla is PhD Scholar in the Electrical and Instrumentation Engineering Department at Thapar Institute of Engineering and Technology, India. He received his B.E. and M.E. degrees in Electrical Engineering from the Punjab Technical University and Thapar Institute of Engineering and Technology, in India, respectively. His current fields of interest include Power Systems, Artificial intelligence, High Voltage Engineering, Fuel Cell.
E-mail: msingla0509@gmail.com