INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 09 ISSUE: 03 | MARCH - 2025 SJIF RATING: 8.586 ISSN: 2582-3930

A Review on Various Machine Learning Algorithms

Priti K. Sahoo[1], M. Laxmi Parsanna[2], B. Gunasekhara[2], P. Shirisha[2], S. Yadav[2], Charanya[2], Archana Reddy[2], P.M. Sandhya[2], N. Shruthi[2], J. Kavya[2], P. Geethanjali[2], P. Nandhini[2], A. Saxena[2] and *S. Bose[2]

[1] *Apmosys Software Ltd., Navi Mumbai, India*
[2] *Dept. of ECE, Bharat Institute of Engineering and Technology, Ibrahimpatnam, Hyderabad, India*
Corresponding author (Email: drsrikanata@biet.ac.in, srknbose@gmail.com)

***Abstract*— Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It's a science that's not new – but one that has gained fresh momentum. Resurging interest in machine learning is due to the same factors that have made data mining and Bayesian analysis more popular than ever. Things like growing volumes and varieties of available data, computational processing that is cheaper and more powerful, and affordable data storage. All of these things mean it's possible to quickly and automatically produce models that can analyse bigger, more complex data and deliver faster, more accurate results – even on a very large scale. And by building precise models, an organisation has a better chance of identifying profitable opportunities – or avoiding unknown risks. By using algorithms to build models that uncover connections, organisations can make better decisions without human intervention. At its most basic, machine learning uses programmed algorithms that receive and analyze input data to predict output values within an acceptable range. As new data is fed to these algorithms, they learn and optimize their operations to improve performance, developing intelligence over time. Machine learning algorithms use computational methods to learn information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases.**

***Keywords*—Artificial Intelligence, Data mining, Machine learning, Machine learning algorithms, Pattern recognition**

I. INTRODUCTION

Machine learning (ML) is the ability of a system to automatically acquire, integrate, and then develop knowledge from large-scale data, and then expand the acquired knowledge autonomously by discovering new information, without being specifically programmed to do so. In short, the ML algorithms can find application in the following: (1) a deeper understanding of the cyber event that generated the data under study, (2) capturing the understating of the event in the form of a model, (3) predicting the future values that will be generated by the event based on the constructed model, and (4) proactively detect any anomalous behaviour of the phenomenon so that appropriate corrective actions can be taken beforehand. ML is an evolutionary area, and with recent innovations in technology, especially with the development of smarter algorithms and advances in hardware and storage systems, it has become possible to perform a large number of tasks more efficiently and precisely, which were not even imaginable a couple of decades before[1-3].

Questions have been asked with regards to computers if they are capable of learning on their own. Human beings have over the years created different tools to enable them solve various tasks which led to the invention and production of different machines. With the rapid developments, the difference between humans and machines has remained intelligence. A human brain analyses information and makes decision accordingly but machines are not able to analyze and take decisions. Automating tasks has generated high interest in the information technology field where some designs and operations can be handed over to machines. Recently, with the introduction of artificial intelligence, machines have been created to have the same level of intelligence as the human brains.

A machine is expected to learn whenever there is changes in the structure, program or data, this is based on the input or response to the external environment which improves its expected results therefore, machine learning can be defined as a part of artificial intelligence that explains that fact that machines can learn on their own when given the right data thereby solving a specific problem. With the help of mathematics and statistics, machine learning can perform intellectual tasks independently that are always generally performed by human beings,

Machine learning is a part of computer science that emanated from the study of pattern recognition and computational learning theory all in artificial intelligence. Algorithms are used to make predictions on data. Before now the field of machine learning was mainly algorithms and theory of optimization but recently machine learning covers several other disciplines which includes statistics, information theory, theory of algorithms, probability and functional analysis. Machine learning and computational statistics are always closely related because of their specialty in prediction making and mathematical optimization which brings about methods, theories and application to the field. In machine learning, strictly static

program instructions are not followed, rather, algorithms are used to build a model from input which are used to make data-driven prediction or decisions.

Machine learning is a branch of artificial intelligence that aims at enabling machines to perform their jobs skilfully by using intelligent software. The statistical learning methods constitute the backbone of intelligent software that is used to develop machine intelligence. Because machine learning algorithms require data to learn, the discipline must have connection with the discipline of database. Similarly, there are familiar terms such as Knowledge Discovery from Data (KDD), data mining, and pattern recognition. One wonders how to view the big picture in which such connection is illustrated in Fig. 1.

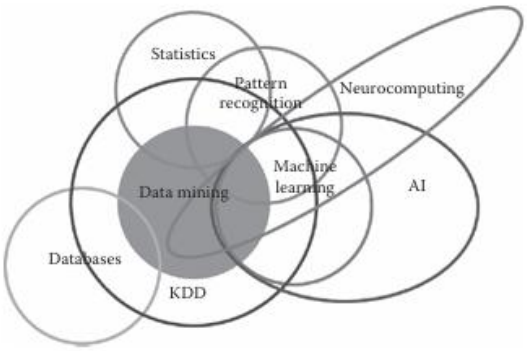

Fig.1: Different disciplines of knowledge and the discipline of machine learning

Recent times are witnessing rapid development in machine learning algorithm systems, especially in reinforcement learning, natural language processing, computer and robot vision, image processing, speech, and emotional processing and understanding. There are numerous applications of machine learning that have emerged or are evolving at present in several business domains, such as medicines and healthcare, finance and investment, sales and marketing, operations and supply chain, human resources, media and entertainment, and so on.

Of late, the applied ML systems in the industry are exhibiting some prominent trends. These trends will utilize the power of ML and artificial intelligence (AI) systems even further to reap benefits in business and society, in general. Some of these trends are as follows: (1) less volume of code and faster implementation of ML systems; (2) increasing use of light-weight systems suitable for working on the resource-constrained internet of things (IoT) devices; (3) automatic generation of codes for building ML models; (4) designing novel processes for robust management of the development of ML systems for increased reliability and efficiency; (5) more wide-spread adoption of deep-learning solutions into products of all domains and applications; (6) increased use of generative adversarial networks (GAN)-based models for various image processing applications including image searching, image enhancement, etc.; (7) more prominence of unsupervised learning-based systems that require less or no human intervention for their operations; (8) use of reinforcement learning-based systems; and finally, (9) evolution of few-shots, if not zero-shot learning-based systems.

This review presents a comprehensive study of various machine learning algorithms. The article is organized as follows: Section II presents motivation while Section III presents the objective. Section IV presents the theory whereas technique and tools is presented in Section V. Results and discussion is presented in Section VI. Future scope and conclusion are presented in sections VII and VIII respectively.

II. MOTIVATION

Machine Learning (ML) has revolutionized jobs, enhancing processes, using advanced systems, and elevating product quality. It has led to a heavy demand for AI and ML professionals with expertise in Artificial Intelligence, machine learning, and Natural Language Processing. Machine Learning is a great career option for those interested in computer science and mathematics. They can come up with new Machine Learning algorithms and techniques to cater to the needs of various business domains.ML is a skill of the future – Despite the exponential growth in machine learning, the field faces skill shortage. If we can meet the demands of large companies by gaining expertise in ML, we will have a secure career in a technology which is on the rise. All these factors motivated to prepare this review on machine learning algorithms. For simplicity, the figures wherever necessary, are taken from the refs[4-38] (as it is) without modifications.

III. OBJECTIVE

The objective of this review is to have knowledge on various machine learning algorithms which are in use today's rapidly growing Artificial Intelligence/Machine Learning (AI/ML) world.

IV. THEORY

Definition:

IBM says Machine learning(ML) is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Machine learning is a field of computer science that deals with the development of algorithms that can learn from data. Machine learning has revolutionized many areas of research over the past few decades- most notably in fields like natural language processing (NLP) and image recognition. It's because machine learning algorithms are able to improve efficiency and accuracy when it comes to tasks like predicting outcomes or interpreting data.

Machine learning algorithms are mathematical model mapping methods used to learn or uncover underlying patterns embedded in the data. Machine learning comprises a group of computational algorithms that can perform pattern recognition, classification, and prediction on data by learning from existing data (training set).

Time-line of Machine learning:

Machine learning earlier:

The early history of machine learning is shown in Fig.2.

Machine learning ahead:

The progress of machine learning from 1997-2017 is shown in Fig.3.

Present status of machine learning:

ML in Robotics:

Machine learning has been used in robotics for various purposes, the most common of which are classification, clustering, regression, and anomaly detection.

- In classification, robots are taught to distinguish between different objects or categories.
- Clustering helps robots group similar objects together so they can be more easily processed.
- Regression allows robots to learn how to control their movements by predicting future values based on past data.
- Anomaly detection is used to identify unusual patterns in data so that they can be investigated further.

ML in Healthcare:

Despite the challenges, machine learning has already made a significant impact in the healthcare industry. It is currently being used to diagnose and treat diseases, identify patterns and relationships in data, and help doctors make better decisions about treatments for patients. However, there is still much work to be done in order to realize the full potential of ML in healthcare.

ML in Education:

Machine learning is a process where computers are taught how to learn from data. This can be used in a variety of ways, one of which is in education.

- Track the progress of students and track their overall understanding of the material they are studying.

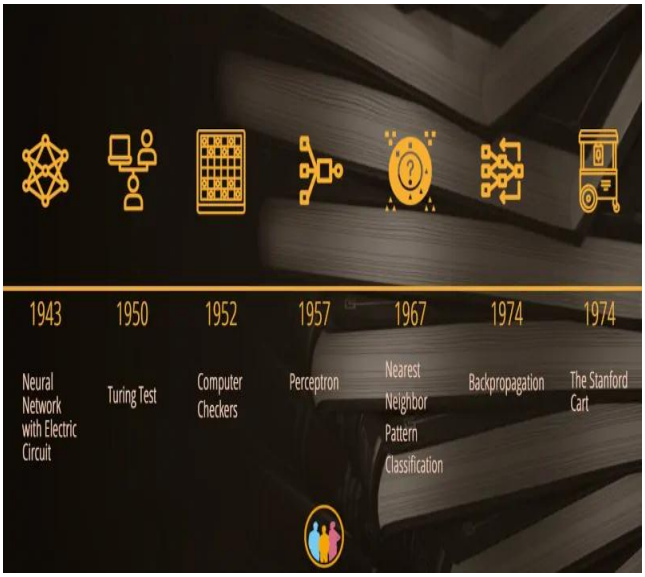

Fig.2: Early history of machine learning

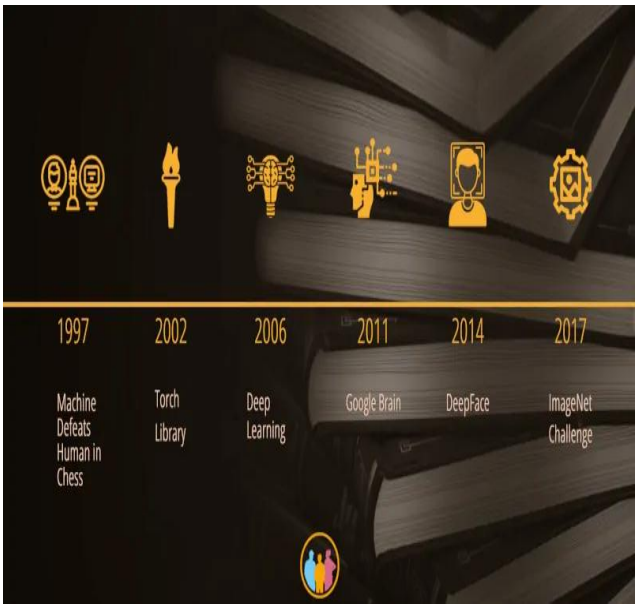

Fig.3: Machine learning from 1997-2017

- Personalize the educational experience for each student by providing personalized content and creating rich environments.
- Assess learners' progress, identify their interests in order to give appropriate support, and track learning progress to help students adjust their course.

Future of Machine Learning:

Quantum Computing:

A quantum computer is a device that uses the principles of quantum mechanics to process information in ways not possible with conventional computers. It has been said by some, including Elon Musk and Bill Gates, that quantum computing will have a huge impact on society, as it may be

the key to unlocking many of our existing problems and creating new ones. Quantum computers are exponentially more powerful than regular computers. They are able to process data at an incredible speed. This is because quantum computers can access information on a microscopic or atomic level, whereas conventional computers work with each piece of data as one whole item. Quantum computers are not yet being used for many tasks because scientists are still trying to figure out how to build them. Scientists have been able to create small quantum computers that can solve some problems, but they do not have the power to do much more.

AutoML:

AutoML is a machine learning algorithm that automates the process of training and tuning machine learning models. There's no doubt that AutoML has been making waves in the world of machine learning lately! Originally developed by Google, AutoML has since proven itself as an invaluable tool for businesses of all sizes – from small start-ups looking for ways to speed up their development processes, right up through larger organizations who need automated methods for dealing with large amounts of data or complex modelling problems. In short: if we're looking to automate any part (or all) of our ML workflow – whether it be developing/tuning our models manually; building features & datasets; or optimizing them – then AutoML is likely something we definitely want on our radar!

Time-line of machine learning algorithms:

The evolution of machine learning algorithms is shown in Fig. 4.

Fig.4 : Evolution of machine learning algorithms

Machine Learning relies on different algorithms to solve data problems. Data scientists like to point out that there is no single one-size-fits-all type of algorithm that is best to solve a problem. The kind of algorithm employed depends on the kind of problem you wish to solve, the number of variables, the kind of model that would suit it best and so on. Here is s a quick look at some of the commonly used algorithms in machine learning (ML) as shown in Fig.5

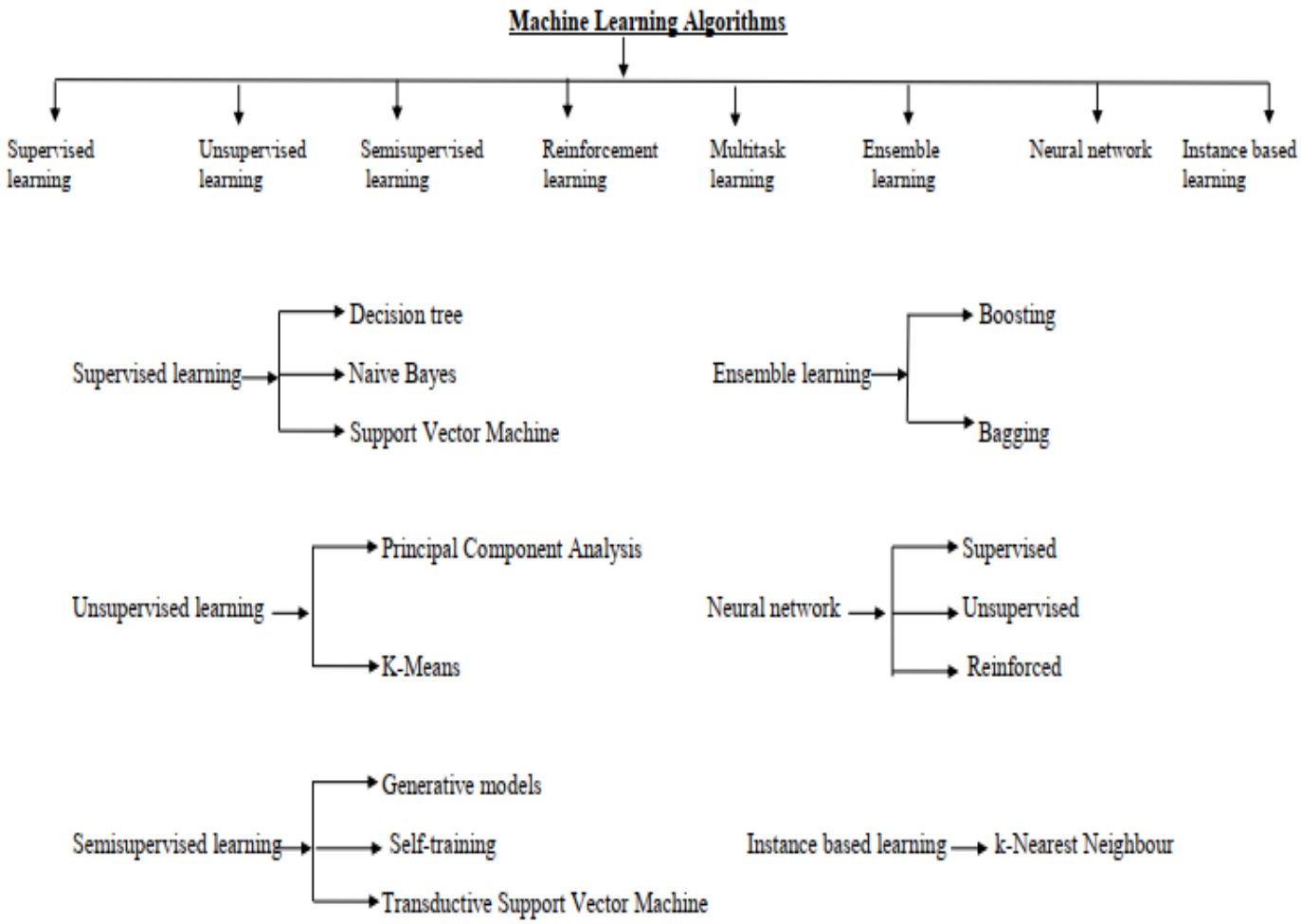

Fig.5: Commonly used algorithms in machine learning (ML)

We will describe each learning methods/algorithms briefly, henceforth.

1. **Supervised Learning:**

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. Supervised learning typically employs training data known as labelled data. Training data has one or more inputs and has a "labelled" output. Models use these labelled results to assess themselves during training, with the goal of improving the prediction of new data (i.e., a set of test data).Typically, supervised learning models focus on classification and regression algorithms. Classification problems are very common in medicine. In most clinical settings, diagnosing of a patient involves a doctor classifying the ailment given a certain set of symptoms. Regression problems tend to look at predicting numerical results like estimated length of stay in a hospital given a certain set of data like vital signs, medical history, and weight. The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. All algorithms learn some kind of patterns from the training dataset and apply them to the test dataset for prediction or classification. The workflow of supervised machine learning algorithm is given in Fig.6.

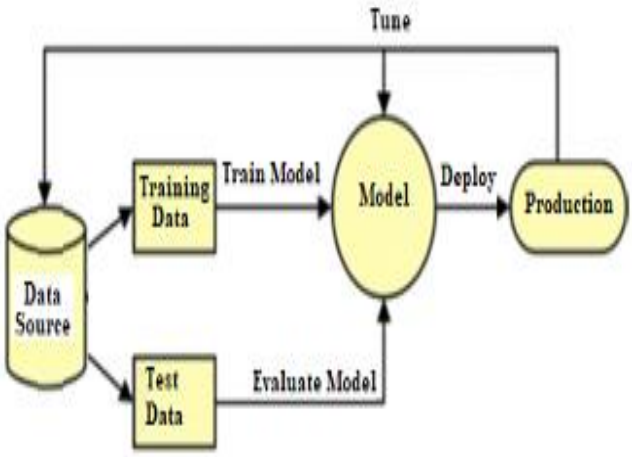

Fig.6(a)

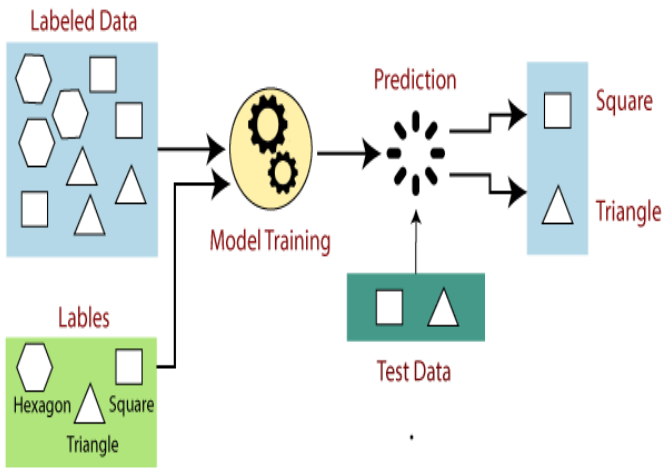

Fig.6(b)

Fig.6(a,b): Workflow of supervised learning

The various common algorithms used in supervised learning are briefly described below:

Decision Tree:

Decision tree is a graph to represent choices and their results in form of a tree. The nodes in the graph represent an event or choice and the edges of the graph represent the decision rules or conditions. Each tree consists of nodes and branches. Each node represents attributes in a group that is to be classified and each branch represents a value that the node can take. A typical graphical representation for decision tree along with its pseudo code are shown in Figs.7 and 8 respectively.

Navie Bayes:

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naïve Bayes mainly targets the text classification industry. It is mainly used for clustering and classification purpose depends on the conditional probability of happening. The mathematics involved in Navie Bayes along with its pseudo code are shown in Figs.9 and 10 respectively.

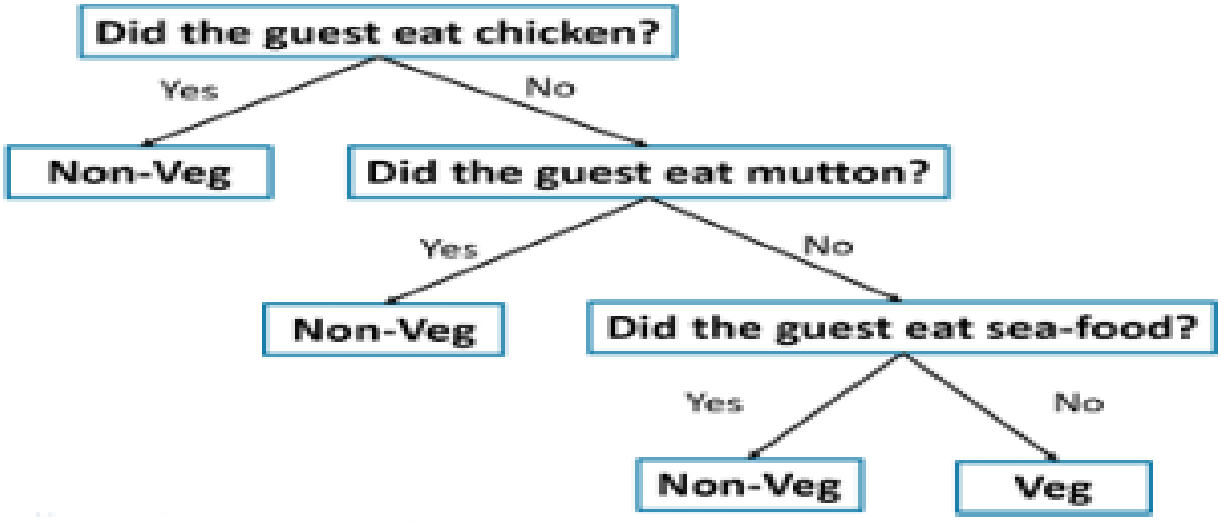

Fig.7: Graphical representation of decision tree

Decision Tree Pseudo Code:

```
def
decisionTreeLearning(examples, attributes,
parent_examples):
if len(examples) == 0:
return pluralityValue(parent_examples)
# return most probable answer as there is no training data
left
elif len(attributes) == 0:
return pluralityValue(examples)
elif (all examples classify the same):
return their classification
A = max(attributes, key(a)=importance(a, examples)
# choose the most promissing attribute to condition on
tree = new Tree(root=A)
for value in A.values():
exs = examples[e.A == value]
subtree = decisionTreeLearning(exs, attributes.remove(A),
examples)
# note implementation should probably wrap the trivial case
returns into trees for consistency
tree.addSubtreeAsBranch(subtree, label=(A, value)
return tree
```

Fig.8: Pseudo code of decision tree

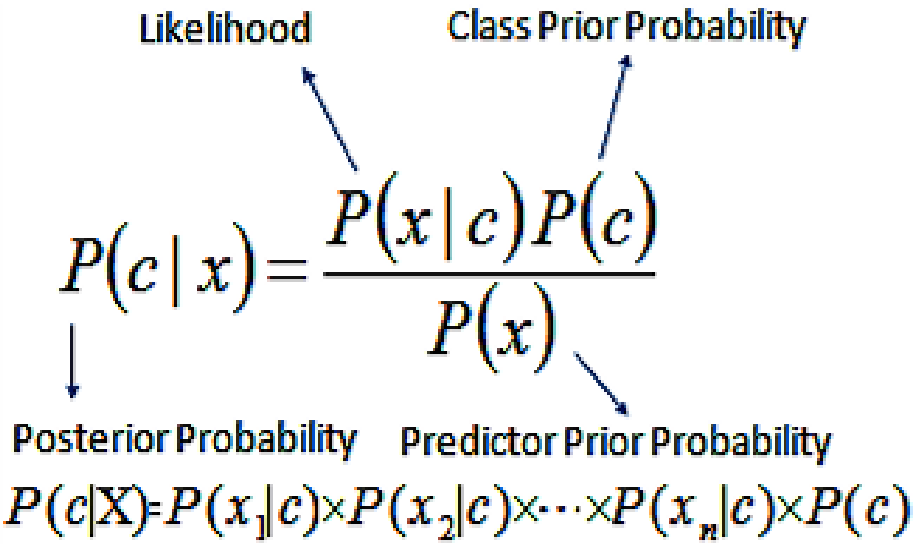

Fig.9: Mathematics involved in Navie Bayes

Pseudo Code for Navie Bayes:

Input:
Training dataset T,
F= (f1, f2, f3,.., fn) // value of the predictor variable in testing dataset.
Output: A class of testing dataset.
Steps:
1) Read the training dataset T;
2) Calculate the mean and standard deviation of the predictor variables in each class;
3) Repeat Calculate the probability of fi using the gauss density equation in each class; Until the probability of all predictor variables (f1, f2, f3,.., fn) has been calculated.
4) Calculate the likelihood for each class;
5) Get the greatest likelihood

Fig.10: Pseudo code of Naïve Bayes

Support Vector Machine:
Another most widely used state-of-the-art machine learning technique is Support Vector Machine (SVM). In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. 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. It basically, draw margins between the classes. The margins are drawn in such a fashion that the distance between the margin and the classes is maximum and hence, minimizing the classification error. The graphical representation along with the pseudo code for SVM are shown in Figs. 11 and 12 respectively.

Advantages of supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as fraud detection, spam filtering, etc.

Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is different from the training dataset.
- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of object.

2. Unsupervised Learning:

These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data. The unsupervised learning algorithms learn few features from the data. When new data is introduced, it uses the previously learned features to recognize the class of the data. It is mainly used for clustering and feature reduction. Unsupervised machine learning uses unlabeled data to find patterns within the data itself. These algorithms typically excel at clustering data into relevant groups, allowing for detection of latent characteristics which may not be immediately obvious. However, they are also more computationally intensive and require a larger amount of data to perform. The workflow of unsupervised machine learning algorithms is given in Fig.13.

The various common algorithms used in unsupervised learning are briefly described below:

Principal Component Analysis (PCA):

Principal component analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. In this the dimension of the data is reduced to make the computations faster and easier. It is used to explain the variance-covariance structure of a set of variables through linear combinations. It is often used as a dimensionality-reduction technique.

Principal Component Analysis is used to reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables, retaining most of the sample's information, and useful for the regression and classification of data and this is illustrated in Fig.14. Fig. 15 depicts pseudo code for PCA.

The following measures are maintained in PCA:

- Principal Component Analysis (PCA) is a technique for dimensionality reduction that identifies a set of orthogonal axes, called principal components that capture the maximum variance in the data. The principal components are linear combinations of the original variables in the dataset and are ordered in decreasing order of importance. The total variance captured by all the principal components is equal to the total variance in the original dataset.
- The first principal component captures the most variation in the data, but the second principal component captures the maximum variance that is orthogonal to the first principal component, and so on.
- Principal Component Analysis can be used for a variety of purposes, including data visualization, feature selection, and data compression. In data visualization, PCA can be used to plot high-dimensional data in two or three dimensions, making it easier to interpret. In feature selection, PCA can be used to identify the most important variables in a dataset. In data compression, PCA

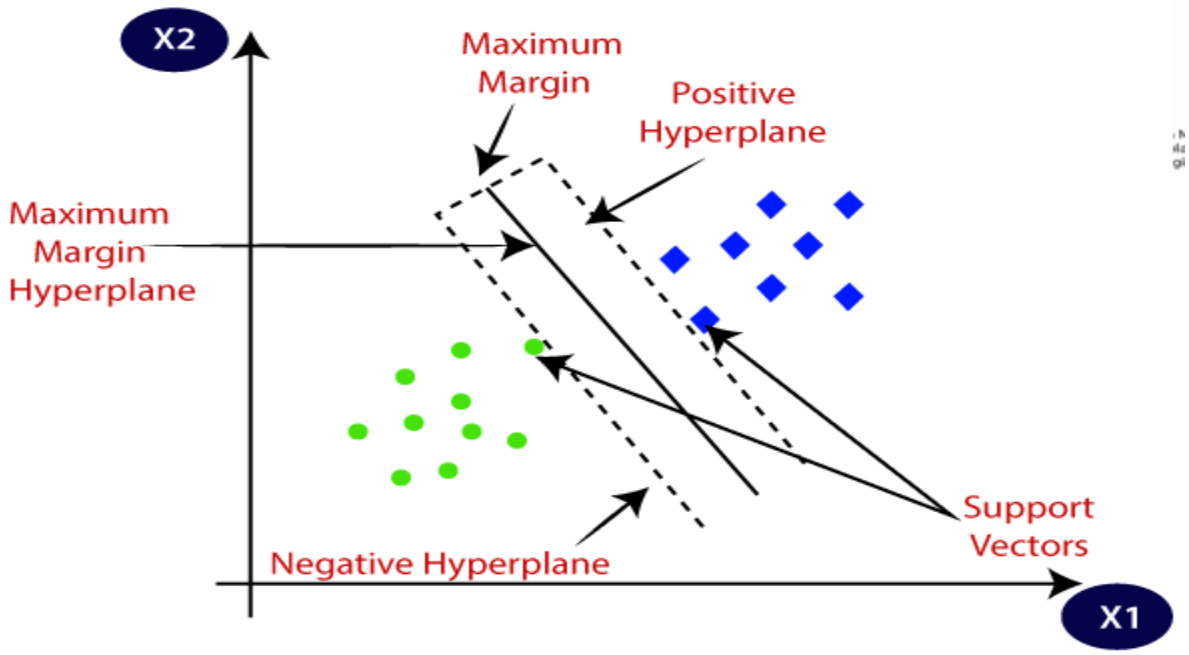

Fig. 11.: Graphical representation of SVM where two different classes (green dots and blue dots) are separated by a hyperplane

Pseudo Code for Support Vector Machine:

```
initialize Yi = YI for i ∈ I
repeat
compute svm solution vv , b for data set with imputed labels
compute outputs ii = (vv , xi) + b for all xi in positive bags
set yi = sgn(fi) for every i e i, yi = 1
for (every positive bag bi) end
if (liei(l + yi)/2 == 0)
compute i* = arg maxiei ii
set yi* = 1
end
while (imputed labels have changed)
output (vv, b)
```

Fig.12: Pseudo code of SVM

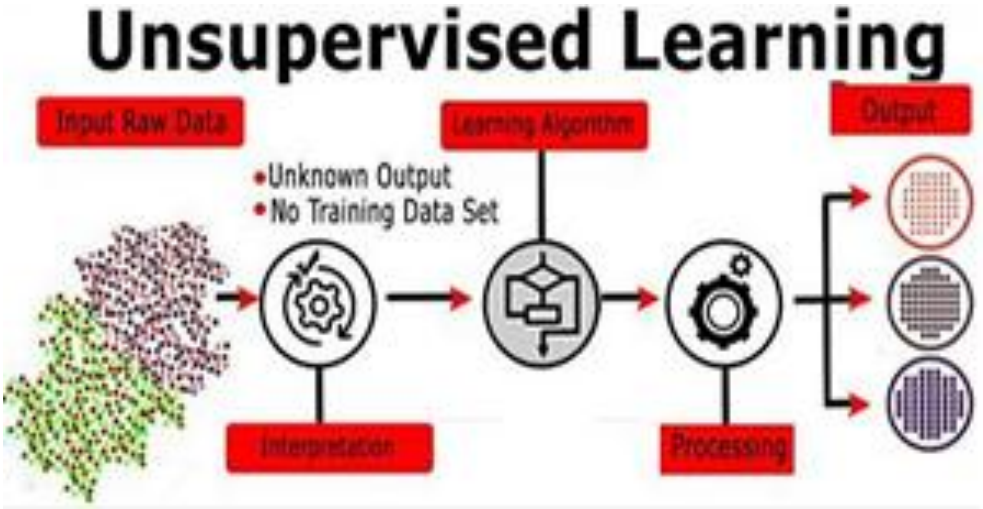

Fig.13(a)

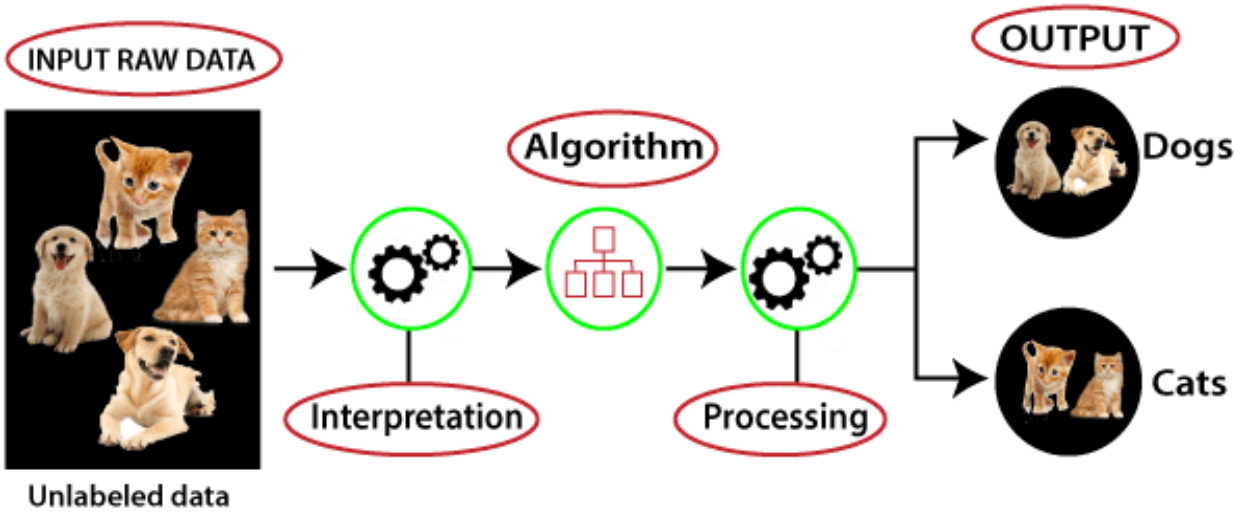

Fig.13(b)

Fig.13(a,b): Workflow of unsupervised learning

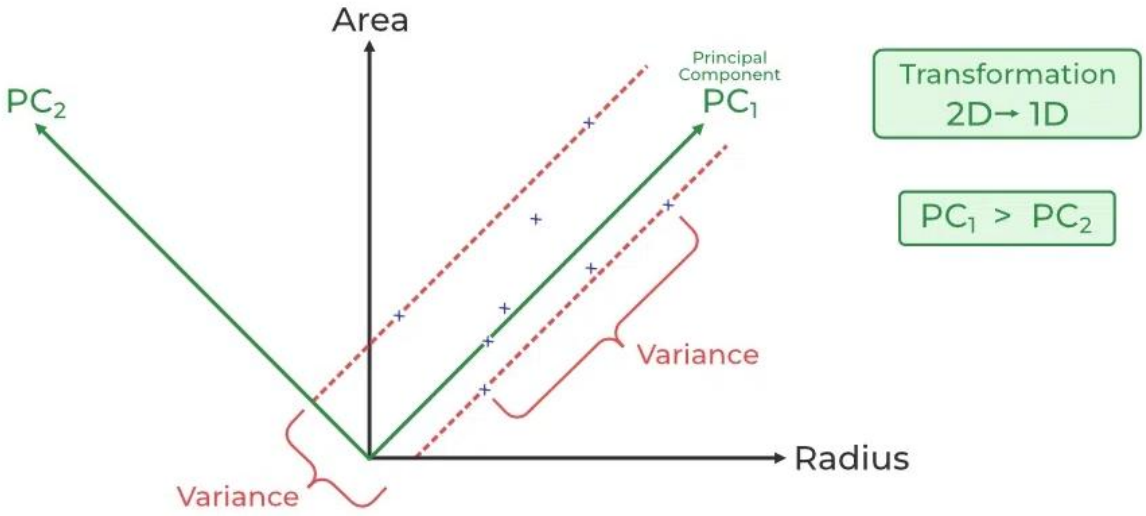

Fig. 14.: Reduction of dimensionality of dataset using principal component analysis

Pseudo code of PCA:

```
Data: Dataset as matrix: dataset
Result: PCA New Reduced Coordinates: reduced
FormDataAsMatrix (dataset, datamatrix)
AdjustMatrixToMean (datamatrix, meanadjusted)
TransposeMatrix (meanadjusted, meanadjustedtransposed)
Mult (meanadjusted, meanadjustedtransposed, correlation)
EigenvectorsAndEigenvalues (correlation, newcoords, eigenvalues)
FilterOutUnusedCoordinates (newcoords, eigenvalues, reduced)
return reduced
```

Fig.15: Pseudo code of PCA

can be used to reduce the size of a dataset without losing important information. Also, in PCA, it is assumed that the information is carried in the variance of the features, that is, the higher the variation in a feature, the more information that features carries.

K-Means Clustering:

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them is much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k

new centroids as bary-center of the clusters resulting from the previous step.
The working of K-Means along with its pseudo-code are shown in Figs. 16 and 17 respectively.

Advantages of Unsupervised Learning:

- Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labelled input data.
- Unsupervised learning is preferable as it is easy to get unlabelled data in comparison to labelled data.

Disadvantages of Unsupervised Learning:

- Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
- The result of the unsupervised learning algorithm might be less accurate as input data is not labelled, and algorithms do not know the exact output in advance.

3. Semi-supervised Learning:

Semi-supervised machine learning is a combination of supervised and unsupervised machine learning methods. It can be fruit-full in those areas of machine learning and data mining where the unlabeled data is already present and getting the labelled data is a tedious process. With more common supervised machine learning methods, you train a machine learning algorithm on a "labelled" dataset in which each record includes the outcome information. Some of Semi Supervise learning algorithms are discussed below:

Generative Models:

A generative model is a type of machine learning model that aims to learn the underlying patterns or distributions of data in order to generate new, similar data. In essence, it's like teaching a computer to dream up its own data based on what it has seen before. The significance of this model lies in its ability to create, which has vast implications in various fields, from art to science.
These models focus on understanding how the data is generated. They aim to learn the distribution of the data itself. For instance, if we're looking at pictures of cats and dogs, a generative model would try to understand what makes a cat look like a cat and a dog look like a dog. It would then be able to generate new images that resemble either cats or dogs. The workflow diagram of generative model and its pseudo code are shown in Figs. 18 and 19 respectively.

Self-Training:

In self-training, a classifier is trained with a portion of labelled data. The classifier is then fed with unlabeled data. The unlabeled points and the predicted labels are added together in the training set. This procedure is then repeated further. Since the classifier is learning itself, hence the name self-training.
The workflow diagram of Self-training and its pseudo-code are shown in Figs. 20 and 21 respectively.

Advantages of Semi-supervised learning techniques:

- Increased accuracy- As the semi-supervised learning technique uses labelled and unlabeled data, accuracy can be increased.
- Reduction in labelling cost- The process of labelling is tedious and expensive, but by using semi-supervised learning, we can use the combination of supervised and unsupervised learning techniques to reduce the labelling cost.
- Usage of both labelled and unlabeled data- Since the semi-supervised learning technique deal with the usage of labelled and unlabeled data, it can be considered an efficient utilization of data.

Disadvantages of Semi-supervised learning techniques:

- Not 100% reliable- Since semi-supervised learning is based on predictions, it might not be fully reliable. There are possibilities of errors too.
- Selection of unlabeled data- It is essential to select the correct unlabeled data as it might hamper the performance of the model.
- Accurate labelling- For labelled datasets, it is important to label them properly. For that, skilled data engineers are required.

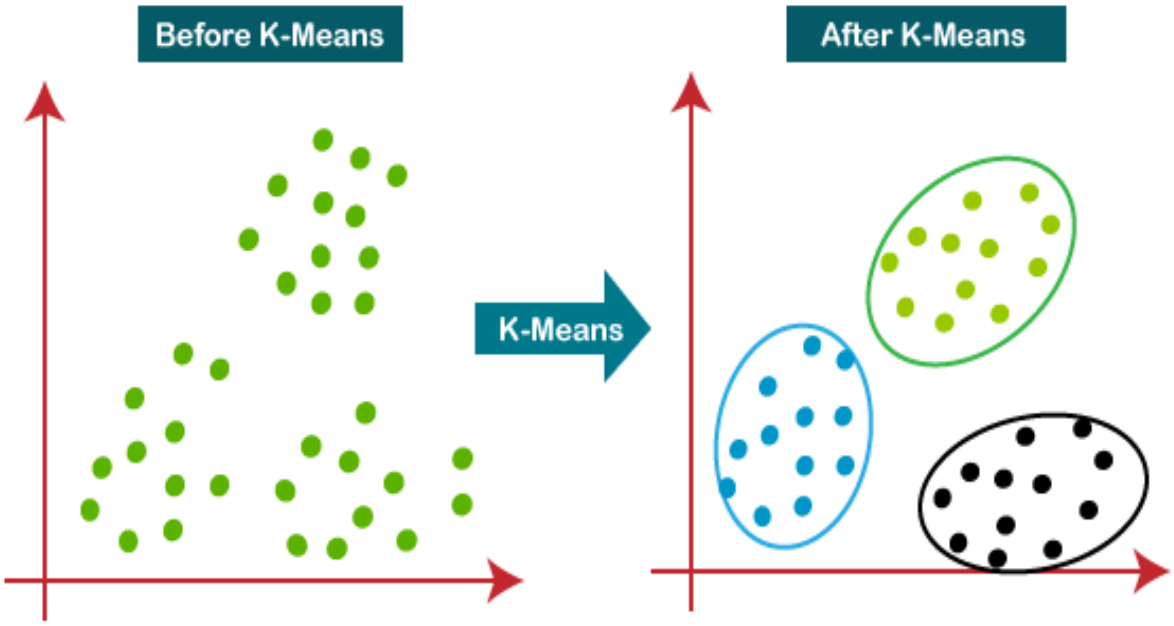

Fig.16: Working of K-Means Clustering

Pseudo code of K-Means Clustering:

Input:
D= {t1, t2, ... Tn } // Set of elements
K // Number of desired clusters
Output:
K // Set of clusters
K-Means algorithm:
Assign initial values for *m1, m2,.... mk*
repeat
assign each item tj to the clusters which has the closest mean; calculate new mean for each cluster;
until convergence criteria is met;

Fig.17: Pseudo code of K-Means Clustering

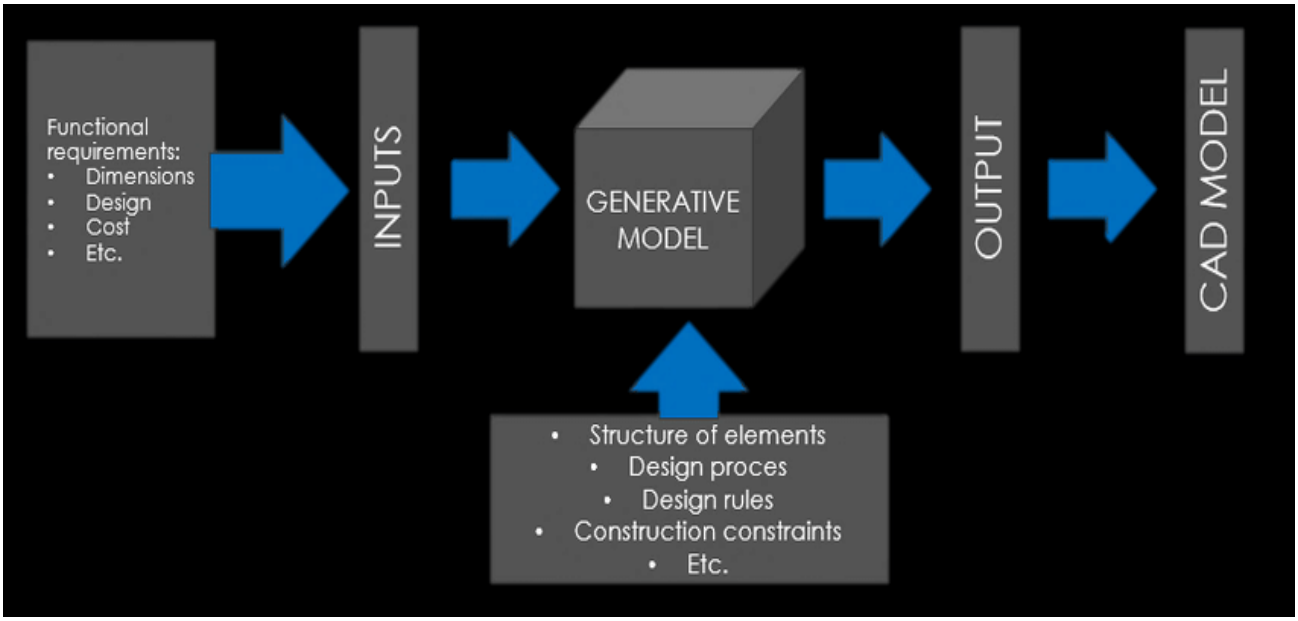

Fig.18: Workflow diagram for generative model

Pseudo code for Generative model:

for number of training iterations **do**
  for k steps **do**
    • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

Fig.19: Pseudo code of a typical generative model

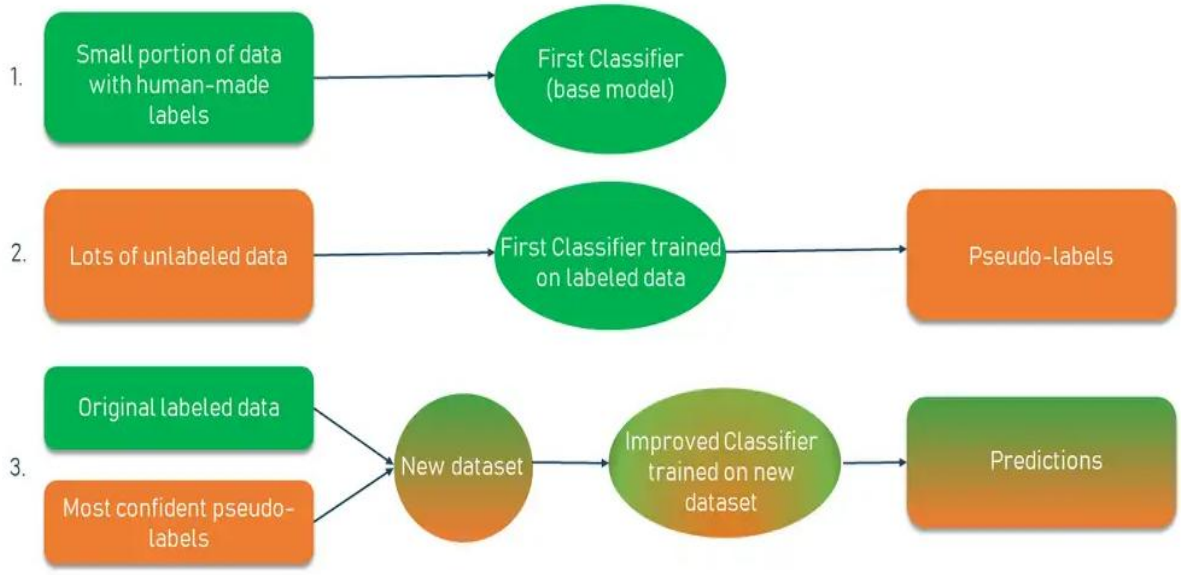

Fig.20: Workflow diagram of Self-training

Pseudo code for Self-training:

Input: labeled data (L), unlabeled data (U)
Output: A given trained classifier

```
Train a given classifier by using L;
while |U| ≠ 0
    Select a data set T from U, where T contains unlabeled samples with high confidence;
    Label all samples in T using the trained classifier;
    L=L ∪ T;
    U=U- T;
    Retrain the classifier using L;
end while
Output the trained classifier;
```

Fig.21: Pseudo code of self-training

4. Reinforcement Learning(RL):

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize some notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning.

Reinforcement Learning is the science of decision making. It is about learning the optimal behavior in an environment to obtain maximum reward. In RL, the data is accumulated from machine learning systems that use a trial-and-error method. Data is not part of the input that we would find in supervised or unsupervised machine learning.

Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next. After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral or incorrect. It is a good technique to use for automated systems that have to make a lot of small decisions without human guidance.

Reinforcement learning is an autonomous, self-teaching system that essentially learns by trial and error. It performs actions with the aim of maximizing rewards, or in other words, it is learning by doing in order to achieve the best outcomes.

The workflow diagram of Reinforcement learning and its pseudo code are shown in Figs. 22 and 23 respectively.

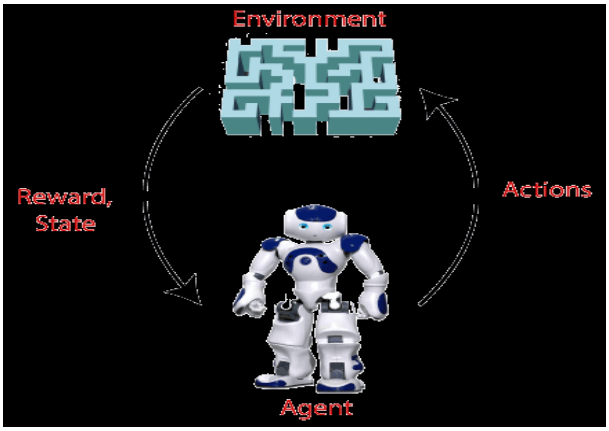

Fig.22: Workflow diagram of Reinforcement learning

Pseudo code for Reinforcement learning:

```
Initialize Q(s, a) arbitrarily
Repeat (for each episode):
   Initialize s
   Repeat (for each step of episode):
      Choose a from s using policy derived from Q
         (e.g., ε-greedy)
      Take action a, observe r, s'
      Q(s, a) <-- Q(s, a) + α [r + γ max_α Q(s', a') - Q(s, a)]
      s <-- s';
   until s is terminal
```

Fig.23: Pseudo code of a typical Reinforcement learning

Advantages of Reinforcement learning:

- Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.
- The model can correct the errors that occurred during the training process.
- In RL, training data is obtained via the direct interaction of the agent with the environment
- Reinforcement learning can handle environments that are non-deterministic, meaning that the outcomes of actions are not always predictable. This is useful in real-world applications where the environment may change over time or is uncertain.
- Reinforcement learning can be used to solve a wide range of problems, including those that involve decision making, control, and optimization.
- Reinforcement learning is a flexible approach that can be combined with other machine learning techniques, such as deep learning, to improve performance.

Disadvantages of Reinforcement learning:

- Reinforcement learning is not preferable to use for solving simple problems.
- Reinforcement learning needs a lot of data and a lot of computation
- Reinforcement learning is highly dependent on the quality of the reward function. If the reward function is poorly designed, the agent may not learn the desired behaviour.
- Reinforcement learning can be difficult to debug and interpret. It is not always clear why the agent is behaving in a certain way, which can make it difficult to diagnose and fix problems.

5. Multi-task Learning:

Multi-Task learning is a sub-field of Machine Learning that aims to solve multiple different tasks at the same time, by taking advantage of the similarities between different tasks. This can improve the learning efficiency and also act as a regularizer. Formally, if there are n tasks (conventional deep learning approaches aim to solve just 1 task using 1 particular model), where these n tasks or a subset of them are related to each other but not exactly identical, Multi Task Learning (MTL) will help in improving the learning of a particular model by using the knowledge contained in all the n tasks. Multi-Task Learning resembles the mechanism of human learning more closely than Single-Task Learning because we humans often learn transferable skills. For example, learning to ride a bicycle makes it easy for someone to learn to ride a motorbike later on, which builds upon similar concepts of body balance. Moreover, learning to ride a motorcycle with gears helps to learn to drive a car with manual transmission faster. This is referred to as the inductive transfer of knowledge. This mechanism of knowledge transfer is what allows humans to learn new concepts with only a few examples or no examples at all (which in Machine Learning is called "Few-Shot Learning" and "Zero-Shot Learning," respectively).
The workflow diagram of Multitask learning and its pseudo-code are shown in Figs. 24 and 25 respectively.

Advantages of Multi-task Learning:

- Data amplification
- Attribute selection,
- Eavesdropping
- Representation bias.
- Reduced inference time
- Solving a set of tasks jointly rather than independently
- Improved prediction accuracy
- Increased data efficiency

- Reduced training time

Disadvantages of Multi-task Learning:

- The quality of predictions is often observed to suffer when a network is tasked with making multiple predictions due to a phenomenon called "negative transfer."
- If there is no feature shared between the different objectives, this technique can yield sub-optimal models as the gradient will struggle to find a common path.
- The lack of a cross-domain formalism that would explain when tasks collaborate or compete.

6. Ensemble Learning:

Ensemble learning is the process by which multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular computational intelligence problem. Ensemble learning is primarily used to improve the performance of a model, or reduce the likelihood of an unfortunate selection of a poor one. Other applications of ensemble learning include assigning a confidence to the decision made by the model, selecting optimal features, data fusion, incremental learning, non-stationary learning and error-correcting.

Ensemble Learning is a method of reaching a consensus in predictions by fusing the salient properties of two or more models. The final ensemble learning framework is more robust than the individual models that constitute the ensemble because ensembling reduces the variance in the prediction errors

Ensemble Learning tries to capture complementary information from its different contributing models—that is, an ensemble framework is successful when the contributing models are statistically diverse.

In other words, models that display performance variation when evaluated on the same dataset are better suited to form an ensemble.

The workflow diagram of Ensemble learning and its pseudo-code are shown in Figs. 26 and 27 respectively. The most popular algorithms used in Ensemble learning are Boosting and Bagging, that are described below briefly:

Boosting:

Unlike many ML models which focus on high quality prediction done by a single model, boosting algorithms seek to improve the prediction power by training a sequence of weak models, each compensating the weaknesses of its predecessors. Boosting is based on the question posed by Kearns and Valiant "Can a set of weak learners create a single strong learner?" To understand Boosting, it is crucial to recognize that boosting is a generic algorithm rather than a specific model**.** Boosting needs you to specify a weak model (e.g. regression, shallow decision trees, etc.) and then improves it. The workflow diagram of Boosting and its pseudo-code are shown in Figs. 28 and 29 respectively.

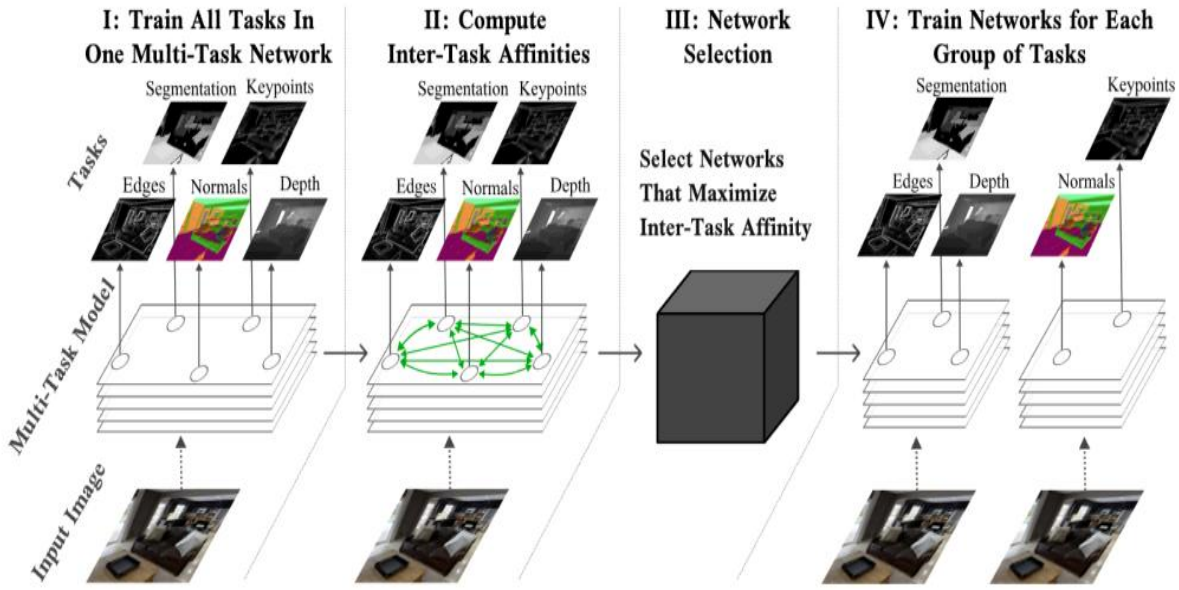

Fig.24: Workflow diagram of Multitask learning

Pseudo code for Multitask learning:

Input : n datasets which belong to the same drug target group (each dataset represents one drug target)
Output: Performance evaluation of RF models built for each of these datasets

1- Concatenate the n datasets into one big dataset;
2- Add an indicator variable TID to each example;
3- Perform the following using the big dataset;
for $i \leftarrow 1$ **to** 10 **do**
  Observe: the splits are stratified based on TID;
  - train set = 90% of the big dataset;
  - test set = 10% of the big dataset;
  - build RF using train set;
  - predict the test set (here we save MOL_ID, TID, actual and predicted values);
end
4- Evaluate using the saved predictions;
for $j \leftarrow 1$ **to** n **do**
  - filter predictions using jth TID;
  - compute and save RMSE for the jth drug target;
end

Fig.25: Pseudo code of Multitask learning

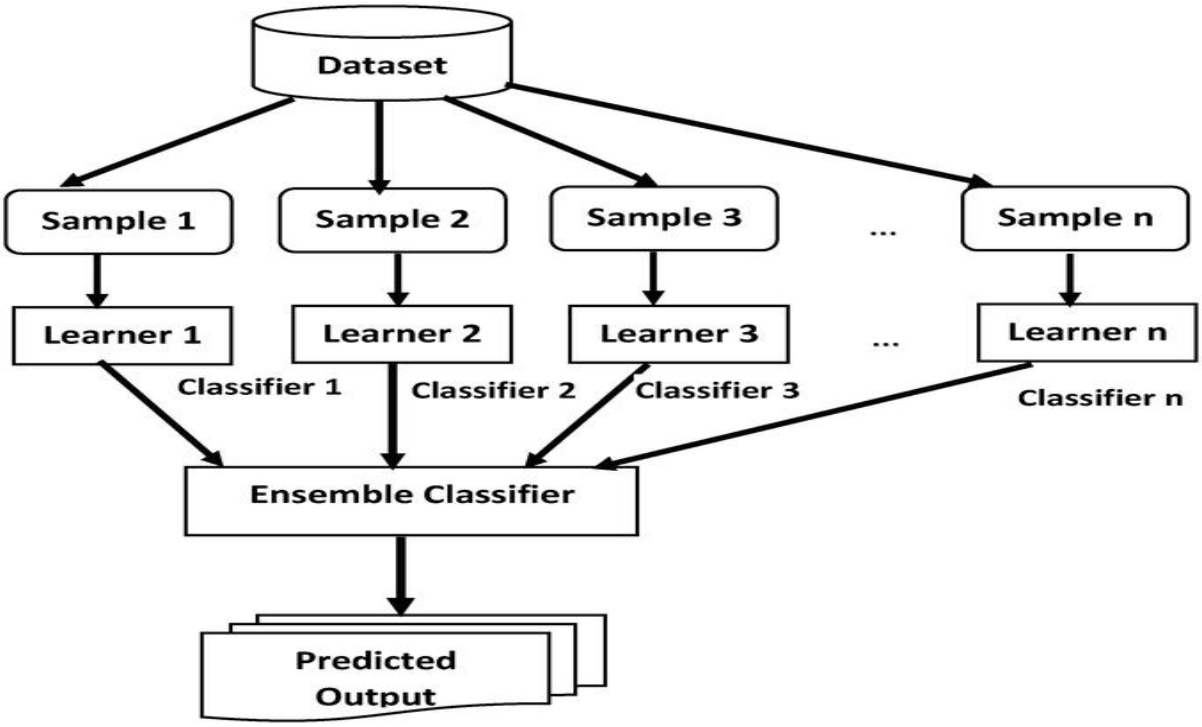

Fig.26: Workflow diagram of Ensemble learning

Pseudo code for Ensemble learning:

Input: Training dataset $\mathcal{D} = \{(x_1, c_1), (x_2, c_2), \ldots, (x_n, c_n)\}$
Base level classifiers $\mathcal{L}1, \ldots, \mathcal{L}_k$
Meta level classifier $\hat{\mathcal{L}}$
Output: Trained ensemble classifier $\hat{\mathcal{M}}$
BEGIN
Step 1: Train base learners by applying classifiers $\mathcal{L}_i$ to dataset $\mathcal{D}$
for $i = 1, \ldots, k$ **do**
$\mathcal{B}_i = \mathcal{L}_i(\mathcal{D})$
end for
Step 2: Construct new dataset of predictions $\hat{\mathcal{D}}$
for $j = 1, \ldots, n$ **do**
for $i = 1, \ldots, k$ **do**
% use $\mathcal{B}_i$ to classify training example x_j
$z_{ij} = \mathcal{B}_i(x_j)$
end for
$\hat{\mathcal{D}} = \{\mathcal{Z}_j, c_j\}$, where $\mathcal{Z}_j = \{z_{ij}, z_{2j}, \ldots, z_{nj}\}$
end for
Step 3: Train a meta level classifier $\hat{\mathcal{M}}$
$\hat{\mathcal{M}} = \hat{\mathcal{L}}(\hat{\mathcal{D}})$
Return $\hat{\mathcal{M}}$
END

Fig.27: Pseudo code of Ensemble learning

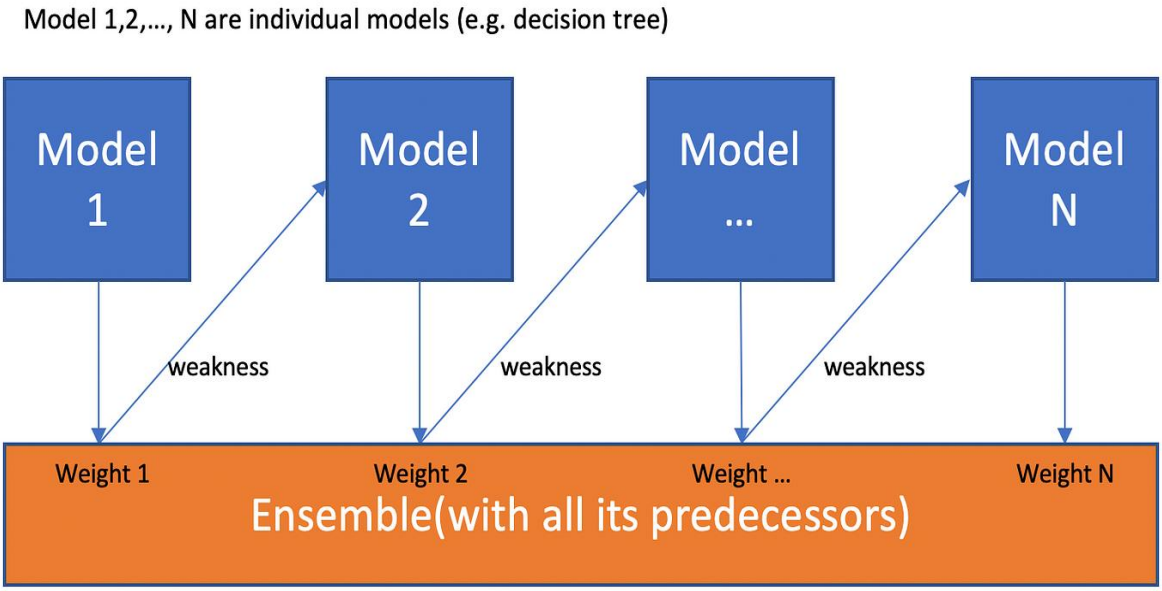

Fig.28: Workflow diagram of Boosting algorithm

Pseudo code for Boosting algorithm:

1: Input: Dataset $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$:
2: Algorithm of learning base L; Number of learning rounds T.
3: Process: $D_1(i) = 1/m$. %Initializes the distribution of weights.
4: For $t = 1, 2, \cdots, T : h_1 = L(D, D_t)$;
5: Train the learning base h_t for D using the D_t distribution
6: $\epsilon_i = Pr_{i \cong D_i}[h_t(x_i \neq y_i)]$;
7: Measures the error of h_t
8: $\alpha_t = \frac{1}{2} ln \frac{1-\epsilon_t}{\epsilon_t}$
9: Determines the weight of h_t
10: $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \begin{cases} exp(-\alpha_t) & if \quad h_t(x_i) = y_i \\ exp(\alpha_t) & if \quad h_t(x_i) \neq y_i \end{cases}$
11: Updates the distribution
12: $= \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ %Normalization factor allowing D_{t+1} to be a distribution
13: End.
14: Output: $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Fig.29: Pseudo code of Boosting algorithm

Bagging:
Bagging or bootstrap aggregating is applied where the accuracy and stability of a machine learning algorithm needs to be increased. It is applicable in classification and regression. Bagging also decreases variance and helps in handling overfitting. It is an ensemble learning technique that helps to improve the performance and accuracy of machine learning algorithms. It is used to deal with bias-variance trade-offs and reduces the variance of a prediction model. Bagging avoids overfitting of data and is used for both regression and classification models, specifically for decision tree algorithms.

The workflow diagram of Bagging and its pseudo-code are shown in Figs. 30 and 31 respectively.

Advantages of Ensemble Learning:

- More accurate prediction results: We can compare the working of the ensemble methods to the Diversification of our financial portfolios. It is advised to keep a mixed portfolio across debt and equity to reduce the variability and hence, to minimize the risk. Similarly, the ensemble of models will give better performance on the test case scenarios (unseen data) as compared to the individual models in most of the cases.
- Stable and more robust model: The aggregate result of multiple models is always less noisy than the individual models. This leads to model stability and robustness.
- Ensemble models capture linear as well as non-linear relationships in the data. It's accomplished by using 2 different models and forming an ensemble of the two.

Disadvantages of Ensemble Learning:

- Reduction in model interpret-ability: Using ensemble methods reduces the model interpret-ability due to increased complexity and makes it very difficult to draw any crucial business insights at the end.
- Computation and design time is high
- The selection of models for creating an ensemble is an art which is really hard to master.

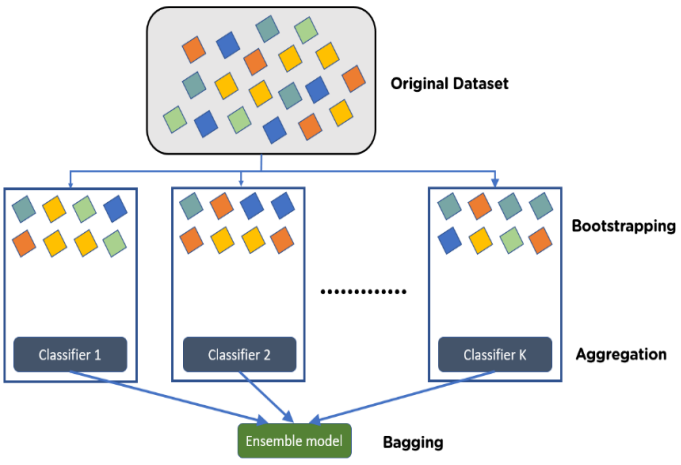

Fig.30: Workflow diagram of Bagging algorithm

Pseudo code for Bagging algorithm:

Input: Data set $TR = \{(x_1, y_1), (x_2, y_2), \ldots, (x_1, y_n)\}$;
Basc learning algorithm L;
Number of learning rounds K.

Process:
for $k = 1, \ldots, K$:
$TR_k = Bootsrap(TR)$; % Generate a bootstrap sample from TR
$h_k = L(TR)$ % Train a base learner h_k from the bootstrap sample
end.

Output:
$H(X) = argmax_{y \in Y} \sum_{k=1}^{K} I(y = h_k(x))$ % the value of $I(a)$ is 1 if a is *true* and 0 otherwise.

Fig.31: Pseudo code of Bagging algorithm

7. Neural Networks:

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems. Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another. Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google's search algorithm.
The workflow diagram of Neural networks and its pseudo-code are shown in Figs. 32 and 33 respectively.
In neural network, basically three types of algorithms are used:

Supervised Neural Network:
In the supervised neural network, the output of the input is already known. The predicted output of the neural network is compared with the actual output. Based on the error, the parameters are changed, and then fed into the neural network again. Supervised neural network is used in feed forward neural network. In supervised learning, the artificial neural network is under the supervision of an educator (say a system designer) who utilizes his or her knowledge of the system to prepare the network with labelled data sets. Thus, the artificial neural networks learn by receiving input and target the sets of a few observations from the labelled data sets. It is the process of comparing the input and output with the objective and computing the error between the output and objective. It utilizes the error signal through the idea of backward propagation to alter the weights that interconnect the network neuron with the point of limiting the error and optimizing performance. Fine-tuning of the network proceeds until the set of weights that limit the discrepancy between the output and the targeted output. The supervised learning process is used to solve classification and regression problems. The output of a supervised learning algorithm can either be a classifier or predictor. The application of this process is restricted when the supervisor's knowledge of the system is sufficient to supply the network's input and targeted output pairs for training.
The workflow diagram of supervised neural network and its pseudo-code are shown in Figs. 34 and 35 respectively.

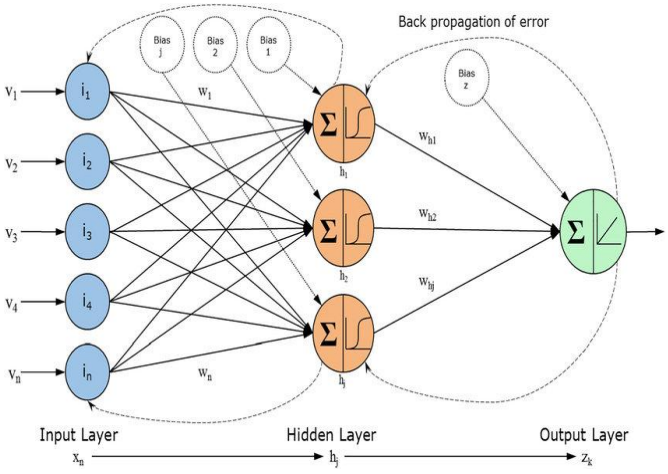

Fig.32: Workflow diagram of Neural network

Pseudo code for Artificial Neural Network:

```
1: procedure ANN (Input, Neurons, Repeat)
   Create input database
2:     Input← Database with all possible variable combinations
   Train ANNs
3:     for Input = 1 to End of input do          ▷ Change inputs for every run
4:         for Neurons = 1 to 20 do              ▷ Increase neurons for every run
5:             for Repeat = 1 to 20 do           ▷ Repeat run 20 times
6:                 Train ANN
7:                 ANN-Storage← save highest test R^2
8:             end for
9:         end for
10:        ANN-Storage← Save best predicting ANN depending on inputs
11:    end for
12:    return ANN-Storage ▷ Library with best predicting ANN for every variable
                          combination
13: end procedure
```

Fig.33: Pseudo code of Artificial Neural Network

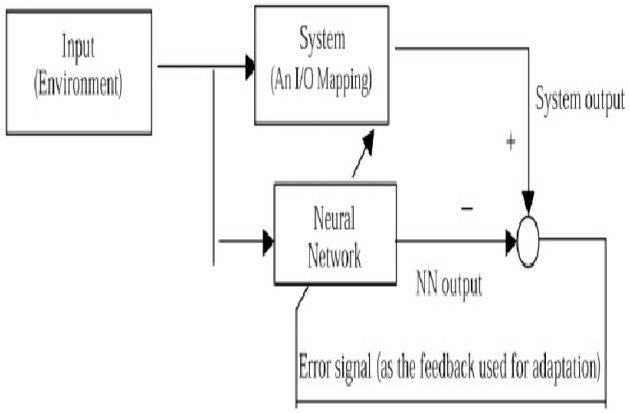

Fig.34: Workflow diagram of Supervised Neural network

Pseudo code for Supervised Neural Network:

1: choose an initial weight vector ~w
2: initialize minimization approach
3: **while** error did not converge **do**
4: **for all** (~x, ~d) ∈ D **do**
5: apply ~x to network and calculate the network output
6: calculate ∂e(~x)
7: **end for**
8: calculate ∂E(D)
9 for all weights summing over all training patterns
10 perform one update step of the minimization approach
11: **end while**

Fig.35: Pseudo code of Supervised Neural network

Unsupervised Neural Network:

The neural network has no prior clue about the output the input. The main job of the network is to categorize the data according to some similarities. The neural network checks the correlation between various inputs and groups them. The workflow diagram and pseudo-code for unsupervised neural network are shown in Figs. 36 and 37 respectively.

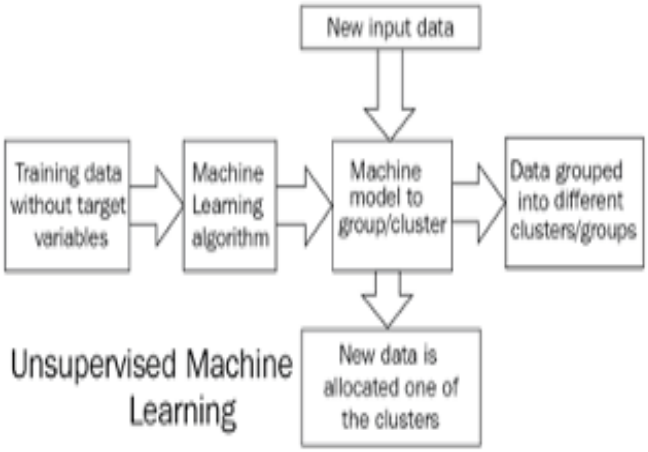

Fig.36: Workflow diagram of Unsupervised neural network

Pseudo code for Unsupervised Neural Network:

```
Procedure
   Initialize the weight matrix W, bias vectors
   a and b, momentum v.
   Set the states of visible unit v1 as the
   training vector
   While i < Max_Iter
     For j = 1, 2 …, m (all hidden units)
        Compute P(h1j = 1|v1) using equation (7)
        Gibbs Sampling h1j ∈ {0,1} from P(h1j|v1)
     End For
     For i = 1, 2 …, n (all visible units)
        Compute P(v2i = 1|h1) using equation (8)
        Gibbs Sampling v2i ∈ {0,1} from P(v2i|h1)
     End For
     For j = 1, 2 …, m (all hidden units)
        Compute P(h2j = 1|v2) using equation (7)
     End For
        //Update rule:
        W := W + ∈ (P(h1 = 1|v1) v1^T − P(h2 = 1|v2)v2^T)
        a := a + ∈ (v1 − v2)
        b := b + ∈ (P(h1 = 1|v1) − P(h2 = 1|v2))
        x := updation of momentum;
   End While
End Procedure
```

Fig.37: Pseudo code of Unsupervised Neural Network

Reinforced Neural Network:

Reinforcement learning refers to goal-oriented algorithms, which learn how to attain a complex objective (goal) or maximize along a particular dimension over many steps; for example, maximize the points won in a game over many moves. They can start from a blank slate, and under the right conditions they achieve superhuman performance. Like a child incentivized by spankings and candy, these algorithms are penalized when they make the wrong decisions and rewarded when they make the right ones – this is reinforcement. Reinforcement learning is a goal-directed computational approach where a computer learns to perform a task by interacting with an unknown dynamic environment. This learning approach enables the computer to make a series of decisions to maximize the cumulative reward for the task without human intervention and without being explicitly programmed to achieve the task.

The workflow diagram and pseudo-code for reinforced neural network are shown in Figs. 38 and 39 respectively.

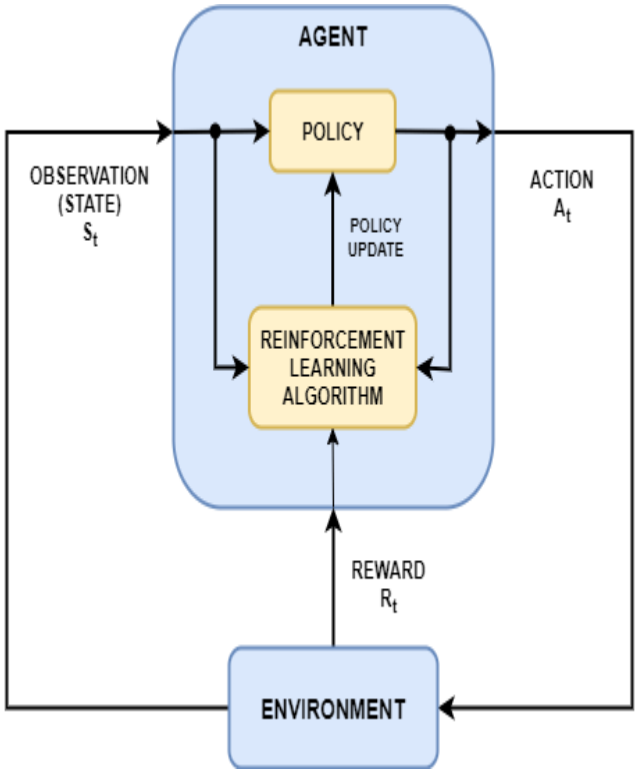

Fig.38: Workflow diagram of Reinforced Neural Network

Pseudo code for Reinforced Neural Network:

Initialize replay memory D to capacity N

Initialize action-value function Q with random weights θ

Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$

For episode = 1, M **do**

Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$

For t = 1,T **do**

With probability ε select a random action a_t

otherwise select $a_t = \text{argmax}_a Q(\phi(s_t), a; \theta)$

Execute action a_t in emulator and observe reward r_t and image x_{t+1}

Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$

Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in D

Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from D

Set $y_j = \begin{cases} r_j & \text{if episode terminates at step j+1} \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^{-}) & \text{otherwise} \end{cases}$

Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the network parameters θ

Every C steps reset $\hat{Q} = Q$

End For

End For

Fig.39: Pseudo code of Reinforced Neural Network

Advantages of neural networks:

- Neural networks can handle unorganized data:
  Neural networks can process large volumes of raw data, enabling them to tackle advanced data challenges. In fact, neural networks get better as they're fed more data. By comparison, traditional machine learning algorithms reach a level where more data doesn't improve their performance.
- Neural networks cam improve accuracy:
  Neural networks practice continuous learning, allowing them to gradually improve their performance after each iteration. This process gives machines the ability to build on past experiences and raise their accuracy over a given period.
- Neural networks can increase flexibility:
  Neural networks are able to adapt to different problems and environments, unlike more rigid machine learning algorithms. This makes it possible to apply neural networks to a wide range of areas, including natural language processing and image recognition.
- Neural networks can lead to faster workflows:
  Neural networks are capable of performing multiple actions simultaneously, speeding up workflows for machines and humans alike. And with computational power increasing exponentially over the years, neural networks can process even more data than before.

Disadvantages of neural networks:

- Neural networks are black boxes, meaning we can't know how much each independent variable is influencing the dependent variables.
- It is computationally very expensive and time consuming to train with traditional CPUs.
- Neural networks depend a lot on training data. It leads to the problem of over-fitting and generalization. The mode relies more on the training data and may be tuned to the data.

8. Instance-Based Learning:

Instance-based learning refers to a family of techniques for classification and regression, which produce a class label/predication based on the similarity of the query to its nearest neighbour(s) in the training set. In explicit contrast to other methods such as decision trees and neural networks, instance-based learning algorithms do not create an abstraction from specific instances. Rather, they simply store all the data, and at query time derive an answer from an examination of the queries nearest neighbour(s).It is called instance-based because it builds the hypotheses from the training instances. It is also known as memory-based learning **or** lazy-learning (because they delay processing until a new instance must be classified). The time complexity of this algorithm depends upon the size of training data. Each time whenever a new query is encountered, it's previously stores data is examined and assign to a target function value for the new instance.
The workflow diagram and pseudo-code for instance based learning are shown in Figs. 40 and 41 respectively.

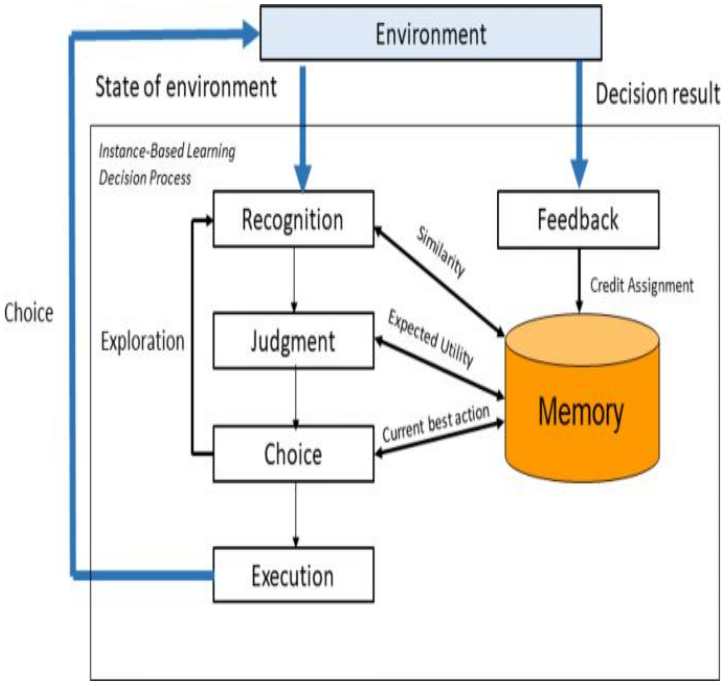

Fig.40: Workflow diagram of instance based learning

Pseudo code of Instance based learning:

Input: default utility x_0, a memory dictionary $\mathcal{M} = \{\}$, global counter $t = 1$, step limit L, a flag *delayed* to indicate whether feedback is delayed.

repeat
  Initialize a counter (i.e., step) $l = 0$ and observe state s_l
  while s_l *is not terminal and* $l < L$ **do**
    Execution Loop
      Exploration Loop $a \in A$ **do**
        Compute activation values $A_{i(s_l^i,a)t}$ of instances $((s_l^i, a), x_{i(s_l^i,a)t}, T_{i(s_l^i,a)t})$ by Eq. 1
        Compute retrieval probabilities $P_{i(s_l^i,a)t}$ by Eq. 2
        Compute blended values $V_{(s_l,a)t}$ corresponding to (s_l, a) by Eq. 3
      end
      Choose an action $a_l \in \arg\max_{a \in A} V_{(s_l,a)t}$
    end
    Take action a_l, move to state s_{l+1}, observe s_{l+1}, and receive outcome x_{l+1}
    Store t into instance corresponding to selecting (s_l, a_l) and achieving outcome x_{l+1} in $\mathcal{M}$
    If *delayed* is true, update outcomes using a *credit assignment* mechanism
    $l \leftarrow l + 1$ and $t \leftarrow t + 1$
  end
until *task stopping condition*

Fig.41: Pseudo code of Instance based learning

K-nearest neighbour is the most popular algorithm used for Instance-based learning and is briefly described below.

K-Nearest Neighbour(k-NN):

The k-nearest neighbours (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use

grows. KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. It is a robust and intuitive machine learning method employed to tackle classification and regression problems. By capitalizing on the concept of similarity, KNN predicts the label or value of a new data point by considering its K closest neighbours in the training dataset. It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data). We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

The workflow diagram and pseudo-code for k-NN are shown in Figs. 42 and 43 respectively.

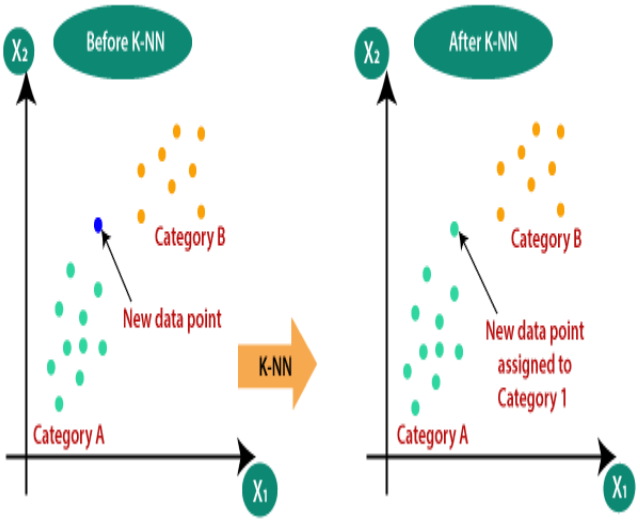

Fig.42: Workflow diagram of k-NN

Pseudo code for k-NN:

Input:

D: a set of training samples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
k: the number of nearest neighbors
$d(x, y)$: a distance metric
x: a test sample

1: **for each** training sample $(x_i, y_i) \in D$ **do**
2: Compute $d(x, x_i)$, the distance between x and x_i
3: Let $N \subseteq D$ be the set of training samples with the k smallest distances $d(x, x_i)$
4: **return** the majority label of the samples in N

Fig.43: Pseudo code of k-NN

Advantages of Instance-based learning:

- It makes no assumptions about the underlying data distribution, making it suitable for complex problems.
- It can also adapt quickly to changes, as it does not require retraining to incorporate new data.

Disadvantages of Instance-based learning:

- It requires a large amount of memory to store the training instances.
- It can also be computationally expensive, as it needs to compute the distance to all training instances for each new instance.
- It is sensitive to the choice of distance measure and the value of K in K-NN algorithm.
- Sensitivity to noise: Noisy or irrelevant instances in the training data can impact predictions significantly.
- Memory requirements: Storing all training instances can be memory-intensive, especially for large datasets.

V. TECHNIQUE AND TOOLS

Machine learning (algorithms) uses two techniques: supervised learning, which trains a model on known input and output data to predict future outputs, and unsupervised learning, which uses hidden patterns or internal structures in the input data.

The two techniques used by machine learning is shown in Fig. 44.

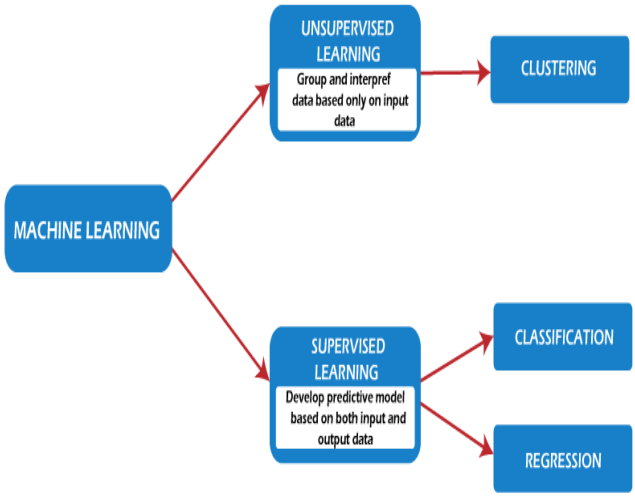

Fig. 44: Techniques used by machine learning

The various important machine learning tools are:

TensorFlow:

TensorFlow is one of the most popular open-source libraries used to train and build both machine learning and deep learning models. It provides a JS library and was developed by Google Brain Team**.** It is much popular among machine learning enthusiasts, and they use it for building different ML applications. It offers a powerful library, tools, and resources for numerical computation, specifically for large scale machine learning and deep learning projects. It enables data scientists/ML developers to build and deploy machine learning applications efficiently. For training and building the ML models, TensorFlow provides a high-level Keras API, which lets users easily start with TensorFlow and machine learning.

PyTorch:

PyTorch is an open-source machine learning framework, which is based on the Torch library. This framework is free and open-source and developed by FAIR(Facebook's AI Research lab). It is one of the popular ML frameworks, which can be used for various applications, including computer vision and natural language processing. PyTorch has Python and C++ interfaces; however, the Python interface is more interactive. Different deep learning software is made up on top of PyTorch, such as PyTorch Lightning, Hugging Face's Transformers, Tesla autopilot, etc.

Google Cloud ML Engine:

While training a classifier with a huge amount of data, a computer system might not perform well. However, various machine learning or deep learning projects requires millions or billions of training datasets. Or the algorithm that is being used is taking a long time for execution. In such a case, one should go for the Google Cloud ML Engine. It is a hosted platform where ML developers and data scientists build and run optimum quality machine, learning models. It provides a managed service that allows developers to easily create ML models with any type of data and of any size.

Amazon Machine Learning (AML):

Amazon provides a great number of machine learning tools, and one of them is Amazon Machine Learning or AML. Amazon Machine Learning (AML) is a cloud-based and robust machine learning software application, which is widely used for building machine learning models and making predictions. Moreover, it integrates data from multiple sources, including Redshift, Amazon S3, or RDS.

Accord.NET:

Accord.Net is .Net based Machine Learning framework, which is used for scientific computing. It is combined with audio and image processing libraries that are written in C#. This framework provides different libraries for various applications in ML, such as Pattern Recognition, linear algebra, Statistical Data processing. One popular package of the Accord.Net framework is Accord.Statistics, Accord.Math, and Accord.MachineLearning**.**

Apache Mahout:

Apache Mahout is an open-source project of Apache Software Foundation, which is used for developing machine learning applications mainly focused on Linear Algebra. It is a distributed linear algebra framework and mathematically expressive Scala DSL, which enable the developers to promptly implement their own algorithms. It also provides Java/Scala libraries to perform Mathematical operations mainly based on linear algebra and statistics.

Shogun:

Shogun is a free and open-source machine learning software library, which was created by Gunnar Raetsch and Soeren Sonnenburg in the year 1999. This software library is written in C++ and supports interfaces for different languages such as Python, R, Scala, C#, Ruby, etc., using SWIG(Simplified Wrapper and Interface Generator). The main aim of Shogun is on different kernel-based algorithms such as Support Vector Machine (SVM), K-Means Clustering, etc., for regression and classification problems. It also provides the complete implementation of Hidden Markov Models.

Oryx2:

It is a realization of the lambda architecture and built on Apache Kafka and Apache Spark. It is widely used for real-time large-scale machine learning projects. It is a framework for building apps, including end-to-end applications for filtering, packaged, regression, classification, and clustering. It is written in Java languages, including Apache Spark, Hadoop, Tomcat, Kafka, etc. The latest version of Oryx2 is Oryx 2.8.0.

Apache Spark MLlib:

Apache Spark MLlib is a scalable machine learning library that runs on Apache Mesos, Hadoop, Kubernetes, standalone, or in the cloud. Moreover, it can access data from different data sources. It is an open-source cluster-computing framework that offers an interface for complete clusters along with data parallelism and fault tolerance. For optimized numerical processing of data, MLlib provides

linear algebra packages such as Breeze and netlib-Java. It uses a query optimizer and physical execution engine for achieving high performance with both batch and streaming data.

Google ML kit for Mobile:

For Mobile app developers, Google brings ML Kit, which is packaged with the expertise of machine learning and technology to create more robust, optimized, and personalized apps. This tools kit can be used for face detection, text recognition, landmark detection, image labelling, and barcode scanning applications. One can also use it for working offline.

Weka:

Weka is a popular open-source machine learning tool that provides a collection of algorithms for data pre-processing, classification, regression, clustering, and visualization. It is widely used in academic and industrial settings and supports a variety of file formats.

BigML:

BigML is a cloud-based machine learning platform that allows users to build and deploy predictive models quickly and easily. With a user-friendly interface and powerful automation tools, BigML enables organizations to derive insights from their data and make better decisions.

Vertex AI:

Vertex AI is a cloud-based machine learning platform developed by Google. It allows developers and data scientists to build, deploy, and manage large-scale machine learning models. Vertex AI supports various popular machine learning frameworks and tools, including TensorFlow, PyTorch, and scikit-learn. Its features and tools are designed to streamline the machine learning workflow and help users achieve faster and more accurate results.

OpenNN:

OpenNN is an open-source software library for neural network development. It provides a high-performance implementation of various types of neural networks. It offers an easy-to-use interface with a wide range of customization options, making it suitable for beginners and advanced users. Additionally, it supports multiple operating systems and programming languages, and its computational speed is optimized for both CPU and GPU architectures.

IBM Watson:

Watson Machine Learning is an IBM cloud service that uses data to put machine learning and deep learning models into production. This machine learning tool allows users to perform training and scoring, two fundamental machine learning operations. Keep in mind, IBM Watson is best suited for building machine learning applications through API connections.

Microsoft Azure Machine Learning:

Azure Machine Learning is a cloud platform that allows developers to build, train, and deploy AI models. Microsoft is constantly making updates and improvements to its machine learning tools and has recently announced changes to Azure Machine Learning, retiring the Azure Machine Learning Workbench.

Various important tools for machine learning presently used are shown in Fig. 45.

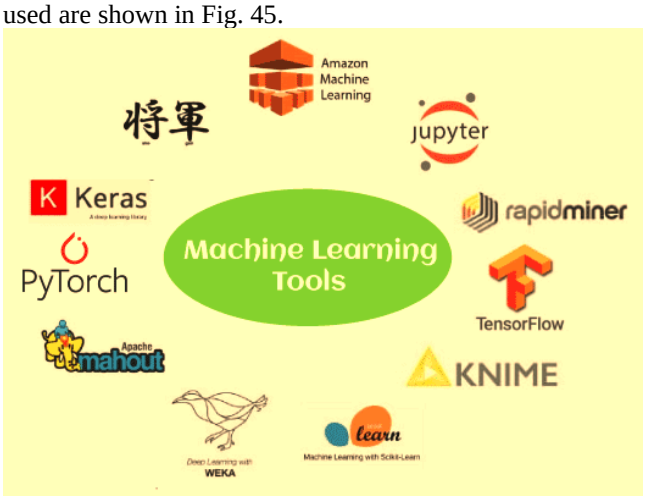

Fig.45(a)

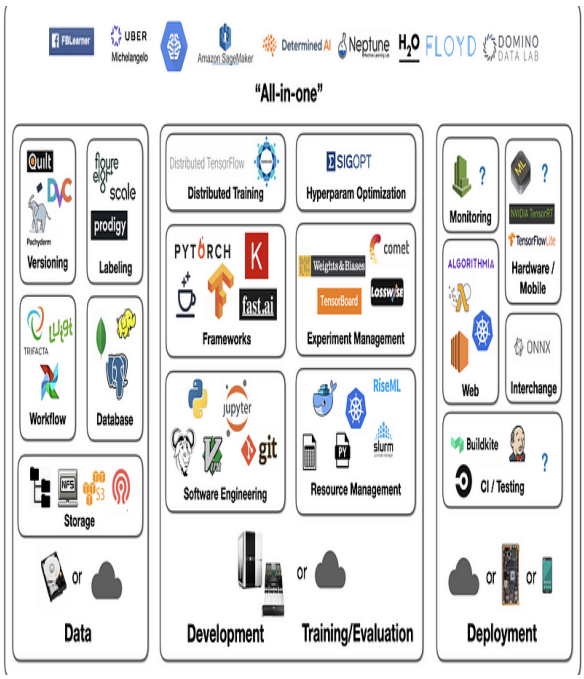

Fig.45(b)

Fig.45(a,b): Tools for machine learning (algorithms).

VI. RESULTS AND DISCUSSION

Machine Learning algorithms can predict patterns based on previous experiences. The overarching practice of Machine Learning includes both robotics (dealing with the real world) and the processing of data (the computer's equivalent of thinking). These algorithms find predictable, repeatable patterns that can be applied to ecommerce, Data Management, and new technologies such as driverless cars. The full impact of Machine Learning is just starting to be felt, and may significantly alter the way products are created, and the way people earn a living. Machine

Learning algorithms are trained with large amounts of data, allowing the "robot" to learn and anticipate problems and patterns. The Mars rover *Curiosity* uses a form of Machine Learning to traverse the Martian terrain, and there are plans to use the same algorithm for driverless cars. In the world of commerce "trend forecasting and analytics" rely on Machine Learning algorithms to anticipate shifts in purchasing behaviours, providing significantly better forecasts than had been done before the algorithm's development.

VII. FUTURE SCOPE

Machine learning algorithms will provide many beneficial applications in occupational safety and health. While algorithm-enabled systems and devices may reduce sources of human error and enhance worker safety and health, algorithms may also introduce new sources of risks to worker wellbeing. Using algorithms, we can divide a task into smaller parts, making it easier to complete. Algorithms are fundamental to computational thinking and problem-solving in many aspects of life, as people use them to complete tasks accurately and effectively. Algorithms streamline processes, optimize resource allocation, and minimize wastage, resulting in cost savings for businesses. The algorithms will also be helpful in document analysis, fraud detection, KYC processing, high-frequency trading, etc. When it comes to work prospects, Machine Learning has a higher scope than other career disciplines both in India and elsewhere in the world. A Number of over 2.3 million is estimated to be employed in the domain of Machine Learning and Artificial Intelligence by 2022 as predicted by Gartner. Additionally, the pay for a machine learning engineer is significantly higher than the pay for other job types. Therefore, future scope of machine learning algorithms is bright.

VIII. CONCLUSION

When using machine learning algorithms, determining what learning algorithm we use is based on the dimensionality, complexity, and the method which we collect our data. Semisupervised algorithms can be the most effective for different applications, but reinforcement algorithms cannot be substituted. Machine Learning can be a Supervised or Unsupervised. If we have lesser amount of data and clearly labelled data for training, opt for Supervised Learning. Unsupervised Learning would generally give better performance and results for large data sets. If we have a huge data set easily available, go for deep learning techniques. We also have learned Reinforcement Learning and Deep Reinforcement Learning. We now know what Neural Networks are, their applications and limitations. This report surveys various machine learning algorithms. Today each and every person is using machine learning knowingly or unknowingly from getting a recommended product in online shopping to updating photos in social networking sites.

This review gives an introduction to most of the popular machine learning algorithms and can be useful for the students or researchers as a reference while working on machine learning algorithms.

ACKNOWLEDGEMENT

The authors are thankful to the management committee of the respective organizations for encouraging and supporting the research works in the respective departments. Dr. S. Bose acknowledges the sincere support from the Chairman Ch. V. Reddy of BIET, in every aspects.

Bibliography

[1] J. Nilsson, "Introduction to Machine Learning an Early Draft of a Proposed Textbook Department of Computer Science." Machine Learning 56 (2): 387–399, 2005 http://www.ncbi.nlm.nih.gov/pubmed/21172442.

[2] W. Richert, L. P. Coelho, "Building Machine Learning Systems with Python", Packt Publishing Ltd., ISBN 978-1-78216-140-0

[3] J. M. Keller, M. R. Gray, J. A. Givens Jr., "A Fuzzy K Nearest Neighbor Algorithm", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-15, No. 4, August 1985

[4]https://www.clickworker.com/customer-blog/history-of-machine-learning/

[5]https://www.researchgate.net/figure/llustration-of-machine-learning-and-deep-learning-algorithms-development-timeline_fig2_359285491

[6]https://www.semanticscholar.org/paper/Analysis-of-Diabetic-Prediction-Using-Machine-on-H.N.-Reddy/fae2f73056b922a1a7710c09fde9eb3e9408b4fd

[7]https://www.nature.com/articles/s41598-024-65255-2

[8]https://discuss.boardinfinity.com/t/how-unsupervised-learning-works/4959

[9]https://www.geeksforgeeks.org/machine-learning/

[10]https://appinventiv.com/blog/machine-learning-algorithms-for-business-operations/

[11]https://www.researchgate.net/figure/The-flow-chart-of-the-Generative-Model_fig1_319888289

[12]https://www.altexsoft.com/blog/semi-supervised-learning/

[13]https://www.codingal.com/coding-for-kids/blog/difference-between-machine-learning-and-deep-learning/

[14]https://www.v7labs.com/blog/multi-task-learning-guide

[15]https://medium.com/@sid8491/ensemble-learning-rachel-greens-chic-guide-to-machine-learning-fashion-38a5e64e8ccc

[16]https://www.researchgate.net/figure/Workflow-diagram-of-the-artificial-neural-network-algorithm-developed-by-Lancashire-et_fig3_323312622

[17]https://pythonphd.com/2018/10/27/types-of-machine-learning-algorithms/

[18]https://www.researchgate.net/figure/The-structure-of-reinforcement-learning-block-diagram_fig1_378683794

[19]https://link.springer.com/article/10.3758/s13428-022-01848-x

[20]https://link.springer.com/chapter/10.1007/978-981-19-0898-9_12

[21]https://www.researchgate.net/figure/Machine-Learning-Techniques_fig1_382851235

[22]https://www.tpointtech.com/machine-learning-tools

[23]https://medium.com/data-science/moving-ai-to-the-real-world-e5f9d4d0f8e8

[24]S. Marsland, Machine learning: an algorithmic perspective. CRC press, 2015.

[25] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine learning techniques in cognitive radios," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1136–1159, Oct. 2012.

[26]https://en.wikipedia.org/wiki/Instance-based_learning

[27] R. S. Sutton, "Introduction: The Challenge of Reinforcement Learning", Machine Learning, 8, Page 225-227, Kluwer Academic Publishers, Boston, 1992

[28] P. Harrington, "Machine Learning in action", Manning Publications Co., Shelter Island, New York, 2012

[29]https://www.careerera.com/blog/machine-learning-career-and-future-scope

[30]https://www.google.com/search?sca_esv=588006686&rlz=1C1VDKB_enIN1079IN1080&q=machine+learning+tools&tbm=isch&source=lnms&sa=X&ved=2ahUKEwjwxrz_sfiCAxULSWwGHSn7CNUQ0pQJegQICxAB&biw=1600&bih=751&dpr=1#imgrc=uNlr98-Dx6baCM

[31]https://www.google.com/search?q=techniques+and+tools+in+machine+learning&rlz=1C1VDKB_enIN1079IN1080&oq=techniques+and+tools+in+machin&gs_lcrp=EgZjaHJvbWUqCQgBECEYChigATIGCAAQRRg5MgkIARAhGAoYoAEyCQgCECEYChigATIJCAMQIRgKGKABMgkIBBAhGAoYoAEyCggFECEYFhgdGB4yCggGECEYFhgdGB7SAQkxNDYyN2owajmoAgCwAgA&sourceid=chrome&ie=UTF-8

[32]https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[33]https://www.clickworker.com/customer-blog/history-ofmachinelearning/#:~:text=Who%20invented%20machine%20learning%3F,in%201959%20while%20at%20IBM.

[34]https://www.geeksforgeeks.org/instance-based-learning/

[35]https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94

[36]https://www.datacamp.com/blog/what-is-a-generative-model

[37]https://www.baeldung.com/cs/principal-component-analysis#:~:text=We%20list%20them%20here%20as,vector%20of%20each%20data%20point.

[38]https://www.researchgate.net/figure/Illustration-of-machine-learning-and-deep-learning-algorithms-development-timeline_fig2_359285491