International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Mathematics & Techniques Related to Machine Learning (ML)

Miss. Vrnika Jain

Student, Department of Artificial Intelligence and Machine Learning, Chandigarh Engineering College, Jhanjeri (Punjab Technical University), CGC Jhanjeri, Mohali – 140307, Punjab, India

ABSTRACT

Since the hundreds of decade we humans are trying to figure out "how we think" and what is the good judgment at the back of our questioning, knowledge and predictions [1]. The field of Artificial Intelligence is somehow same trying to put into effect that intelligence in equipments. AI: Artificial Intelligence, as the name says a lot about itself it is human made thinking ability.

ML: Machine Learning, the ability of learning the trends and patterns when given to a machine is Machine Learning.

DL: Deep Learning, when we embed neural networking (like human brain neurons) into machine and drill down deeper into patterns is Deep Learning.

When we look into these terms we observe that these technologies working mechanisms and progress are partially based on roots of Mathematical concepts which include statistics, probability theory, calculus, linear algebra, [2] optimization methods, and information theory. Understanding these foundations is essential for innovators and researchers who want to contribute to this dynamic field [3].

This paper describes the mathematics that is hidden behind the Artificial Intelligence, Machine Learning and Deep Learning that we are using today, which are not the unusual milestones it has done thus far. The implementation is available on the presented GitHub repository, https://github.com/Vrnika-Jain/ML-Algos .

Keywords: Artificial Intelligence, Machine Learning, Deep Learning, optimization, model.

INTRODUCTION

Artificial Intelligence is a branch of computer science which deals with building the intelligence in machines [4]. Basically, it is a study that enables computers to do the stuff that can make them look intelligent to the humans. Artificial Intelligence plays a very vital role to perform intelligent behaviour, for learning, demonstrating and giving advices to the user [6]. In a broader view Artificial Intelligence is a truss of learning, problem solving, adjust the new solutions and perception to the system [3].

Artificial Intelligence consists of two types,

1. **Weak Artificial Intelligence:**

Computers do not have thinking ability but behave as per the programming and calculations. For example, chess game.

1. **Strong Artificial Intelligence:**

Machines which work on its own and think as powerfully as humans. For example, humanoid robots.

Machine Learning is a division of Artificial Intelligence where machines are expected to learn whenever there are changes in the structure, program or data and improve the estimated results [5]. Here, strict statistical programming and instructions are never followed as we work with the algorithms to construct the models. There are three taxonomies of machine learning.

1. Supervised Learning:

[7] Here the mapping takes place between the data labels and values to train the data sample from data source.

There are two types of Supervised Learning Techniques, which are:

- Regression: Regression is a strategy to determine the association between target variables (dependent variable) and predictor variables (independent variable) to predict steady outcomes [7].
  - It consists of algorithms like Linear Regression, Naïve Bayes Regression models, Lasso Regression, Ridge Regression, Random Forest Regression, Net Elastic Regression, K Nearest Neighbor, Decision Tree Regression models etc.
- Classification: As the term is speaking for itself, classification is a technique to categorise the data into two or more categories based on training data.
  - It consists of algorithms like Support Vector Machine, Logistic Regression, Random Forest , Naïve Bayes Classifiers, Decision Tree models etc.

2. Unsupervised Learning:

Here the machine tends to learn using the unlabeled data i.e. without human interruptions to train the data sample from data source.

There are two types of Unsupervised Learning Techniques, which are:

- Clustering: Cluster means bunch of something, here in Machine Learning, Clustering means grouping the data points which are having similar characteristics into one group which is calculated using metrics like Euclidean distance, Cosine similarity, Manhattan distance, etc. [3]
  - We see the various algorithms here, Principal component analysis, K Means Clustering, Hierarchical clustering, PAM Clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), etc.
- Association Rule: Association Rule is known for its ability to recognise the patterns inside the data.
  - It uses the algorithms like Apriori Algorithm and FP Growth Algorithm [9].

3. Reinforcement Learning:

Involves training a machine just like a kid (termed as agent technically), to take actions in the environment and learns from the feedback provided/observed for its actions (add to database if it's a reward and discard if punished) [8]. This reinforcement learning has been classified into two types:

1. Positive:

Here the system performs some operations in the environment and it has a positive effect on the behaviour and increase the strength of the system.

2. Negative:

Here the system performs some operations in the environment and it has a negative effect on the behaviour and might can decrease the strength of the system, which needs to be avoided or stopped.

Deep Learning is a folk of machine learning family [5], that is used to analyze and work on bunk of data which includes images as well with the help of neural networking [10]. There are various taxonomy of deep learning and we are still having the gates opened for more new types,

- Convolutional Neural Network: CNNs are feedforward neural networks (neurons moving in one i.e. forward direction) where using the convolution operations, neurons covers the outlying units within the convolution kernel (a sliding 2-D matrix used in input data to determine its behaviour) and has excellent performance in large image processing [11].

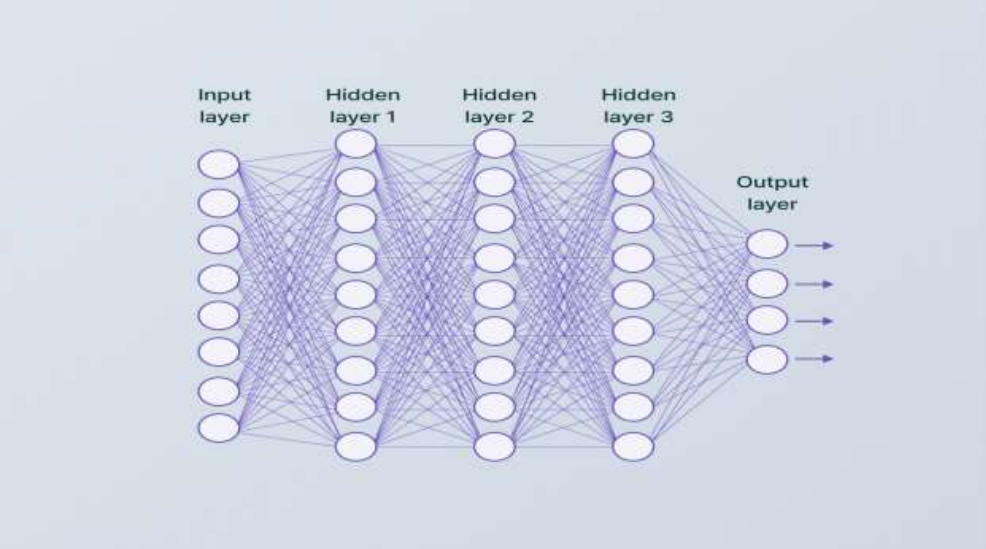


Figure: Convolutional Neural Networks

- Recurrent Neural Networks: RNNs are directed connections among neurons which form a cycle and allows the network to take inputs within the sequence and depends on the prior elements from the sequence [12] using the "memory".

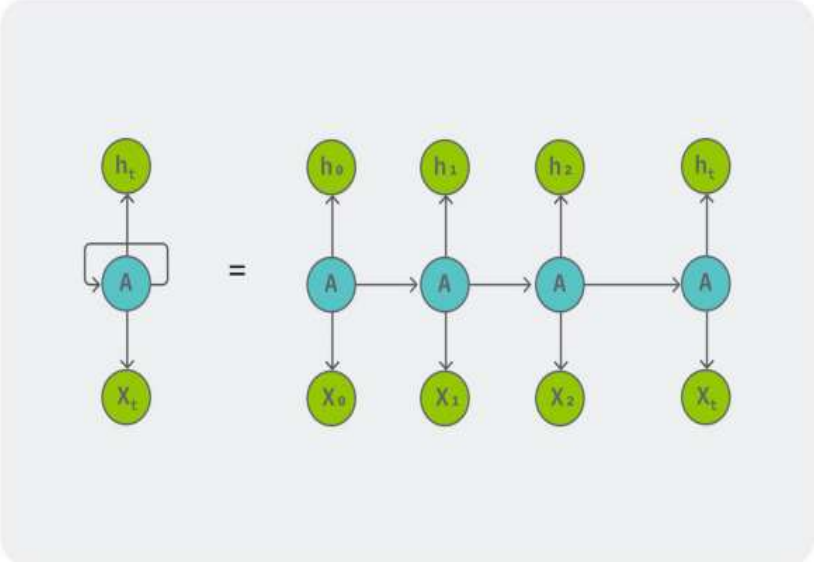


Figure: Recurrent Neural Network

- Generative Adversarial Networks: GANs are the basic of generative AI where we claim to create the new content that uses the training data as a reference. GANs are composed of two words which are

generator helps to create data (can be wrong or correct) and *discriminator* to distinguish between the data authenticity.

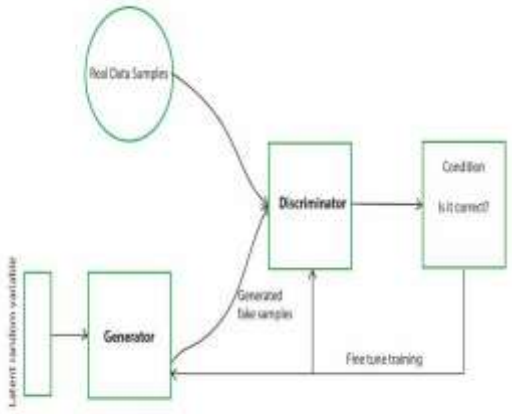


Figure: Generative Adversarial Networks

- Restricted Boltzmann Machines: RBMs learn from the probability distribution in the given input data and mainly used for dimension reduction, regression and classification, topic modelling.

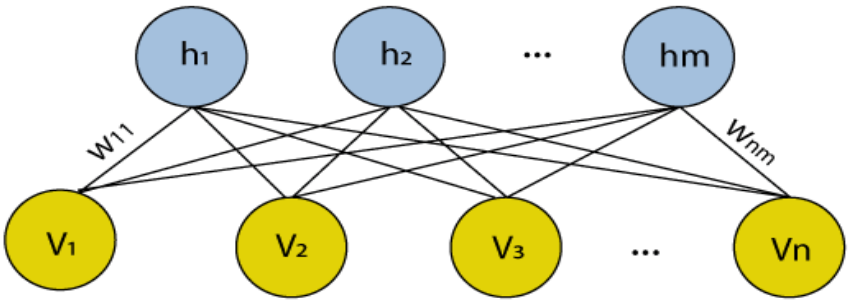


Figure: Restricted Boltzmann Machines

LITERATURE REVIEW: MACHINE LEARNING MATHMETICS

1. Supervised Learning:

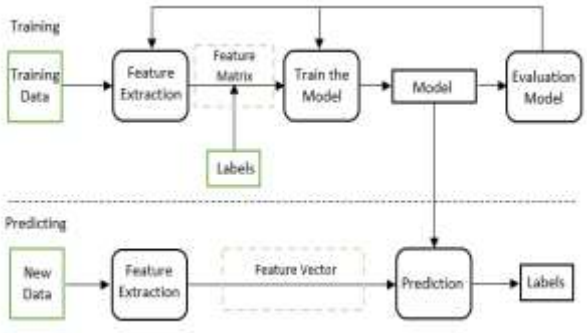


Figure: Workflow of Supervised Learning Technique

1.1. ***Univariate linear regression*:**

A beeline in the form of $\mathbf{o_w(i)= w_1 i+w_0}$ [2],where ,
i is the input,
o_w is the output,
w_0 and w_1 are the (weights) real valued coefficients.
To achieve an accomplishment of linear regression we need to firstly, fetch the values for the weights [w_0 ,w_1] which will further minimize the drops when the given equation will derivate partially $\sum_{j=1 \text{ to } N} (p_j - (w_1 i_j + w_0))^2$ forms w_0 and w_1 as zero, also p is a variable which will often change with the change in weights values [5].
The final solution comes out to be:

$$w_1 = [N(\sum i_j p_j) - (\sum i_j)(\sum p_j)]/[N(\sum i_j^2) - (\sum i_j)^2];$$
$$w_0 = (\sum p_j - w_1(\sum i_j))/N;$$

1.2. ***Multivariate Linear Regression*:**

It is like a wing of linear regression which we use to predict the target variable based on the values of two or more predictor variables [3]. It is in the form of $o_{sw}(i_j)=w_0 + w_1 i_{j,1} + \ldots\ldots + w_n i_{j,n} = w_0 + \sum_k w_k i_{j,I}$, but here $w_0, w_1 .. w_n$ are the coefficient to indicate the contribution of each predictor variable to predict target variable, and $i_{j,0} = 1$ to tell the strength of changes, at last, o is dot product of inputs and weights and further forms the equation as:

$$o_{sw}(j) = \sum_{k=0 \text{ to } n} w_k x_{j,I}$$

,that needs to be minimized using squared loss error loss function which can reach one and the updated equation will be:

$$w_k \leftarrow w_k + \alpha \sum_j i_{j,i}(p_j - o_w(i_j))$$

1.3. ***Logistic Regression*:**

It happens to predict the taxonomy of target variables using the given predictor values that results into the probabilistic values i.e. value between 0 and 1 [6].

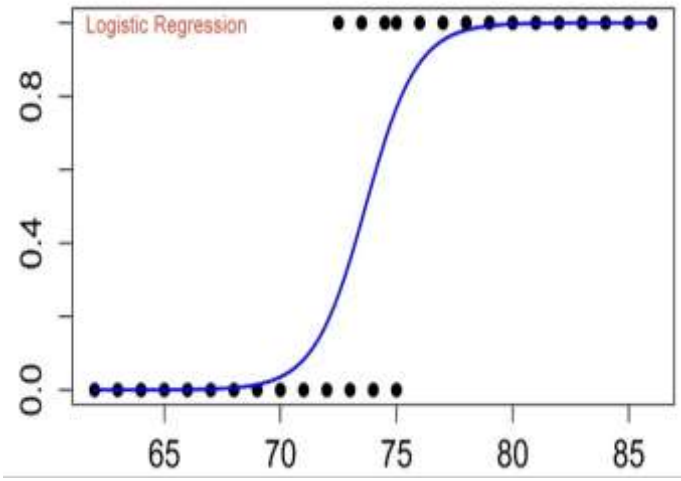


Figure: Logistic Regression

Here as the two functions (slopes) seems very similar in shape, therefore the logistic function is:

$$o_w(i)=\text{Logistic}(w.i)=1/[1+e^{-w.i}] ,$$

where, (w.i) can be claimed as z as a short form.

1.4. ***Support Vector Machine*:**

SVM is a well known algorithm which is used to solve both the classification as well as Regression problems by determining a *hyperplane* in an n-dimensional (N-D) space that can specifically classify required data points. A different approach which carries out the optimized solution for problem is called alternative representation.

$$\sum_j \alpha_j - (1/2) \sum_{j,k} \alpha_j \alpha_k y_j y_k (x_j x_k), \text{ where } \alpha_j >= 0 \text{ and } \sum_j \alpha_j y_j = 0$$

SVMs build a separator taking maximum margin to distinguish and generalize the model well which subsequently forms a decision boundary with comprehensive possible gap till the example points [9].

Once the optimized α_j has been determined, the following property is applicable for the separator also;

$$h(x) = \text{sign}(\sum_j \alpha_j y_j (x.x_j) - b) ,$$

b is here to keep intercept safe from involving into data points i.e. as a separate parameter.

1.5. ***Lasso Regression*:**

It is also known as L1 Regularization, where we try to shrink the overfitting in data and enhance the correctness of predictions in data. We achieve this by adding a penalty term to the main function [14]. Penalty term is the absolute sum of absolute values magnitude of the coefficient vector **w,**

$$\text{L1 Penalty} = \lambda \sum_{j=1}^{p} |w_j|$$

It is combined with the Residual Sum of Square (RSS) which measures how a model is fitting in the data and is formulated as:

$$(1/2n) \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

and after combining the equations we get the L1 Regression equation:

$$\min_w \{(1/2n) \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda \sum_{j=1}^{p} |w_j|\}$$

here,

$x_i = (x_i, x_i, \ldots, x_i)^T$ is feature vector for observations.

y_i is the response variable for the observation.

$w = (w_1, w_2, \ldots, w_p)^T$ are the coefficients used for penalty calculation.

n stands for the total number of observations observed.

p are the total number of features.

λ controls strength for the penalty in the regularization [14].

1.6. ***Ridge Regression*:**

It is also known L2 or Tikhonov Regularization, which also helps to prevent the overfitting and enhance the accuracy of predictions by shrinking the coefficients but instead of taking Penalty term as the total of absolute values magnitude for the coefficient vector w, we add penalty term as squares of absolute values of the coefficient vector w [15];

$$\text{L2 Penalty} = \lambda \sum_{j=1}^{p} (w_j)^2$$

The overall L2 Regression equation looks something like this, which is combined with OLS cost to make sure that the penalty/squared error is minimized as L2 alone may separate out everything far away from its actual position.

$$\min_w\{(1/2n)\sum_{i=1}^{n}(y_i - w^T x_i)^2 + \lambda\sum_{j=1}^{p}(w_j)^2\}$$

This method is a way bit more effective then L1 regularization as we do not shrink the coefficients to exact zero [15] and therefore it is only used when we need to reduce the effect of some unimportant features being used in the model construction.

1.7. ***Elastic Net Regression*:**

The combination of L1 regression and L2 regression is known as Elastic Net Regression. It basically handles scenarios where Lasso regression faces issue of correlation between features, than Elastic Net regression performs well on variable selection and overfitting at the same time [15][16]. Elastic Net adds both L1 and L2 penalties into the loss function. Objective function for minimizing the following expression is :

$$\min_w\{(1/2n)\sum_{i=1}^{n}(y_i - w^T x_i)^2 + \lambda_1\sum_{j=1}^{p}(w_j)^2 + \lambda_2\sum_{j=1}^{p}|w_j|\}$$

The parameters λ_1 and λ_2 plays a very important role here as they are the key to balance the model fitting and penalties that can be caused in large coefficients.

1.8. ***K Nearest Neighbors*:**

It is surpassing Machine Learning Algorithms which help in generalizing learning and not instance based learning, i.e. it tends to memorize the training instances and do not support learning the whole model explicitly. The idea behind the KNN working is that the data points which are nearby and have similar characteristics will be putted into one category and labelled by one name [9]. We tend to calculate the distance using equations like Euclidean distance;

$$f(m,n) = \sqrt{\sum_{i=1 \text{ to } n}(m_i - n_i)^2}$$

and sometimes we also use Manhattan distance;

$$f(m,n) = \sum_{i=1 \text{ to } n}|m_i - n_i|$$

and then we proceed further with using classification decision rule $n^{hat} = \text{argmax}_c \sum_{i\in NNk(m)} 1\{n_i=c\}$; where,

$NN_k(m)$ represents the near neighbours of m,

n_i = labels,

c stands for each classes which are going to get classified.

When the points gets to get classified into classes then we implement the Prediction rule on the set of data points, by using $n^{hat} = (1/k)\sum_{i\in NNk(m)} n_i$ and $n^{hat} = (\sum_{i\in NNk(m)} mini) / (\sum_{i\in NNk(m)} wi)$, when it is a weighted KNN, and w acts as a weight associated with the i^{th} data point.

1.9. ***Decision Tree*:**

This technique is used for both regression and classification tasks where we observe the data getting split into subsets till we reach to an end point and grab an answer according to the requirement. We achieve

this by using recursive method to partition the data and also predicting the target value [5]. Decision tree consists of concepts like;

Entropy is used to measure the impure data in dataset so that we can treat them to discourage any disorder in results. **Entropy**; $H(d)=-p_1\log_2(p_1)-p_2\log_2(p_2)$; here, p_1 and p_2 are class 1 and class 2 propositions in dataset d.

Information Gain is termed to measure the effective feature that should be given priority while each internal level of node splitting in data.

Information Gain; $IG(d,P)=H(d)-\sum_{v\in values(P)} H(d_v)_{|dl \text{ to } |dv|}$; here d stands for dataset, v symbolise the value and P is the feature on which we are working, d_v is the subset of dataset [6].

Gini Index is similar to entropy, i.e. it measures the impurities in data, the only difference here is the formulation.

Gini Index; $G(d) = 1-(p_1^2+p_2^2)$, p1 and p2 are class 1 and class 2 propositions in dataset d. Gini Gain is used for determining the best feature to split the data, similar to Information Gain [6][5].

$GG(d,P) = G(d)-\sum_{v\in values(P)} G(d_v)_{|dl \text{ to } |dv|}$; d is supporing the dataset, v stands for the value and P is the feature on which we are working, dv is the subset of dataset.

1.10. ***Random Forest*:**

Random Forest is a bunch of Decision Trees for the sake of overall accuracy and strength in the choosen data points. Here the multiple decision tree models work together to produce better overall results [7]. We firstly apply bagging by randomly sampling N instances in d dataset as d_i to train the separate decision trees. After which we train the decision trees and use them for classification and regression. For classification we use; $y\hat{} = mode(\{y_1\hat{},y_2\hat{},\ldots,y_b\hat{}\})$, where b is total number of decision trees. For regression we use; $y\hat{} = (1/b)\sum_{i=1to\ b}(y_i\hat{})$

Once, every tree is trained on the bagging sample (bootstrap sample), we are left with approximately $1/3^{rd}$ amount of data to do the testing part and calculate the error which is known as out of bag error (OOB) which is calculated by [7]; OOB Error = $(1/N)\sum_{i=1 \text{ to } N} L(y_i,y\hat{}_{OOB}(x_i))$; where; L is the loss function, $y\hat{}_{OOB}(x_i)$ is the estimation of OOB for i and y_i is the true label from the data.

2. Unsupervised Learning:

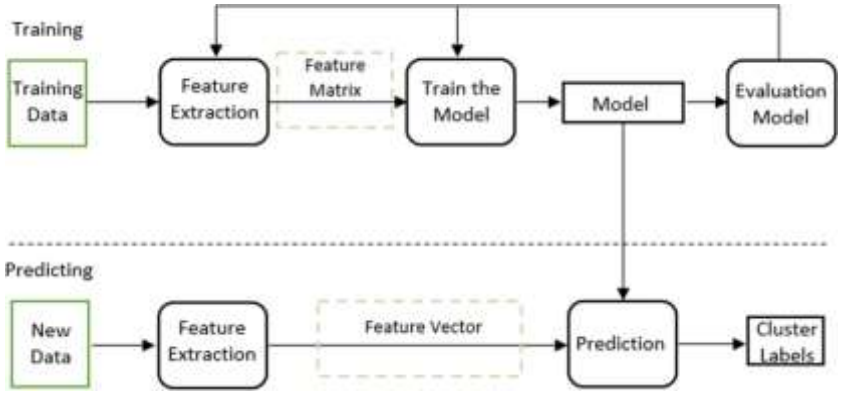


Figure: Workflow of Unsupervised Learning Technique

2.1. ***K Means Clustering*:**

It happens to be an iterative algorithm where we tend to work with the unlabeled data and divide the data points into one cluster after analysing the similarity between those points [9]. It is a centroid based algorithm where each and every cluster carries a central point in them.
It is represented as:

$$\min_c \sum_{k=1 \text{ to } K} \sum_{j \in Ck} \|m_j - \mu_k\|^2; \text{ where,}$$

K is the number of clusters, m_j is the data point, μ_k is the centroid for the different set of clusters $C=\{C_1,C_2,...,C_k)$ and $\|m_j - \mu_k\|^2$ is the Euclidean distance calculated between the data points of the respective cluster.

2.2. ***Hierarchical Clustering*:**

Hierarchical Clustering is also a technique in which we pack the data points into one cluster also, the method to analyse the cluster is not partitioning but analyzing using dendrogram (hierarchy of clusters) [8]. We find two division of hierarchical clustering:

- **Bottom-Up Method**

It is known as *Agglomerative Clustering* in technical terminology, where pairing starts from the data points considering every as a separate and single cluster and after analysing, closest clusters [8] got paired up into a single cluster iteratively till the cluster become rigorous to their characteristics.

- **Top-Down Method**

It is known as *Divisive Clustering* in technical terminology. It is somehow just opposite of Bottom-Up Method, i.e. every data point falls under one cluster and then based on the closeness and characteristics they iteratively splits till the data points become rigorous enough.
To measure the distance it uses methods like Euclidean Distance and Manhattan Distance which we have discussed K Nearest Neighbour under Supervised Learning part as well.
How the choice will choose which data point/cluster to pick depends on the linkage criteria [9], which are as follows:

Single Linkage: minimum distance between the data points;
$d(C_p,C_q) = \min_{m \in Cp, n \in Cq} d(m,n)$
Average Linkage: average distances between the data points;
$d(C_p,C_q) = 1/(|C_p||C_q|) \sum_{m \in Cp} \sum_{n \in Cq} d(m,n)$
Centroid Linkage: a distance between the centroids from each cluster;
$d(C_p,C_q) = d(\bar{m}_{Cp},\bar{m}_{Cq})$
Complete Linkage: maximum distance between data points;
$d(C_p,C_q) = \max_{m \in Cp, n \in Cq} d(m,n)$

2.3. ***Principle Component Analysis (PCA)*:**

It approaches as a dimensionality reduction methodology that is known to minimize the variables from the dataset while preserving the information as much as possible. Principal Components are the modification of initial variables into further series of uncorrelated variables which can be ordered by variance [8].
Primary thing to do for PCA is to standardize the data to have mean value as zero and standard deviation value as one.

$$Z_{ij} = (x_{ij} - \bar{x}_j)/s_j \text{ ;here,}$$

$\bar{x}_j$ acts as mean of j feature and s_j symbolises standard deviation for j feature.

Now, to calculate the covariance matrix C of the data which we have standardized in first step we use;

$$C = (1/(n-1))\ Z^T Z$$

each element C_{ij} represents the covariance between i and j points.

After this there is a need to construct eigenvectors (represents the directions of principal components) and eigenvalues (represents the variance occurred by principal component) [3] following that we sort them

$$Cv = \lambda v$$

results in n eigenvalues and λi and correlated eigenvectors.

Calculate the principal component by projecting Z on eigenvectors using PC=ZV; where V is matrix of eigenvectors.

2.4. ***Apriori Algorithm*:**

It is an Association rule part which is used to mine the items which are occurring often and is operated on transactional database [10]. Some concepts we need to know about Apriori Algorithm are:

Support: calculates how often of an item is occurring in dataset.

Support(m) = (Number of transactions dealing with m)/(Total transactions occured)

Confidence: checks for the items containing n that are occurring frequently in transactions

Confidence(m → n) = Support(m U n)/ Support(m)

Lift: measures the probability of n most likely of getting purchased when x is purchased.

Lift(m → n) = Confidence(m → n)/Support(n)

3. Reinforcement Learning:

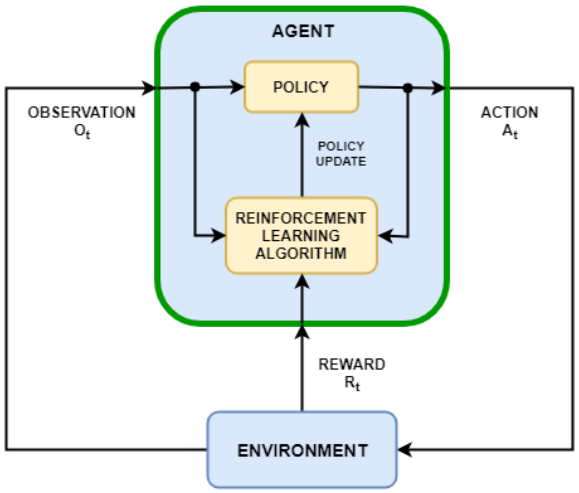


Figure: Workflow of Reinforcement Learning Technique

Unlike Supervised learning, here we are not provided with the whole labelled dataset, but here the systems tries to learn after exploring the environment on its own, making mistakes and learning from them, just like us humans[8]. There are some terminologies which are required to understand firstly to understand the Reinforcement Learning [8].

- **Agent:** performs the learning part.
- **Environment:** the external place where agent interacts and learns.
- **Action:** options for the agent to do in environment, represented by *a*.

- **Reward:** feedback an agent receives after performing the action, represented by *r*.
- **State:** current circumstances of the agent, represented by *s*.
- **Value function:** expected reward for an action, represented by *V*.
  $V^{\pi}(s)=E_{\pi}[\sum_{k=0 \text{ to } \infty} \gamma^{k} r(s_k,a_k)|s_0=s]$
- **Policy:** planning for the next task based on the current task and state, represented by π.
  $\pi(a|s)=P(a_k=a|s_k=s)$
- **Q-Function:** expected reward to take action in a state under some policy, represented by q.
  $q^{\pi}(s,a)=E_{\pi}[\sum_{k=0 \text{ to } \infty} \gamma^{k} r(s_k,a_k)|s_0=s,a_0=a]$

For implementing Reinforcement Learning, we formulate (MDP) Markov Decision Processes where,
s: series of states,
a: series of actions,
P(s′|s,a): Transitional probability from state s to state s′ to perform an action a,
r(s,a): Reward function to tip the agent,
γ: Discount factor, $0 \leq \gamma < 1$
We also observe Bellman equation playing a major role on affecting the results. Bellman equation writes that "*the working action equals to reward coming from the current action combined and the expected reward from the future actions.* [13]*"*
For the Value Function: $V^{\pi}(s)=\sum_{a}\pi(a|s)\sum_{s'}P(s'|s,a)[r(s,a)+\gamma V^{\pi}(s')]$
For Q-Function: $q^{\pi}(s,a) =r(s,a)+\gamma\sum_{s'} P(s'|s,a) \sum_{a'} \pi(a'|s')q^{\pi}(s',a')$
We also have an upgraded policy π^* that have the capability to maximize the return while optimizing the value function and Q-Function which also satisfies the Bellman equations. To represent the optimal values we use * with the functions.
For Value Function: $V^*(s)=\max_{a}\sum_{s'} P(s'|s,a)[r(s,a)+\gamma V^*(s')]$
For Q-Function: $q^*(s,a)=r(s,a)+\gamma\sum_{s'} P(s'|s,a) \max_{a'} q^*(s',a')$

ARTIFICIAL NEURAL NETWORKING MATHEMATICS

ANN simply says constructing a mathematical model like a human brain to mimic the brain activity (thinking ability) in the similar way. We all are aware from our biology books, since school, times that a human brain works properly because of formation of an electro-chemical which we call neurons [10], likewise, in Artificial Intelligence and Machine Learning, [11] we are trying to form these neuron net-working structure and working in an artificial manner in computers.

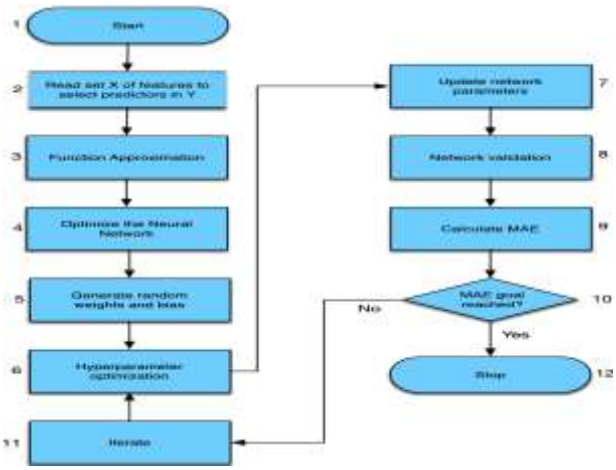


Figure: Workflow of Artificial Neural Networks

These training can be associated under Supervised, Unsupervised or Reinforcement Learning depending on the data and requirements. Every link (that connects the two neurons with each other) has a weight w_{kj} linked with it [11]: $in_j = \sum_{k=0}^{n} w_{kj} a_k$ which then applies an activation function for the following output:

$$a_j = f(in_j) = f(\sum_{k=0}^{n} w_{kj}\, a_k)$$

sometimes we call it perceptron and **sigmoid perceptron**. The error vector $y\text{-}h_w(x)$ for any weight's loss is,

δ Loss(w)/ $\delta w = \sum_k (\delta/\delta w(y_q - a_q)^2)$, where, the index q ranges over nodes in final (output) layer.

STATISTICAL MODELING MATHMETICS

It is representation of information and thesis afterwards we construct an audit to infer any relationships between predictor and target variables. Bayesian model is one of the most effective and widely used model for building statistical models which are fundamentally based on Bayes' Theorem;

$$p(o_i|d) = \alpha\, p(d|o_i) \backslash p(o_i),$$

where d stands for an/the observed value from all the represented data. And the most frequently used Bayesian model in machine learning is **naïve bayes models [7]** which acts as a probabilistic classifier and tries to solve classification tasks. Here the probability of each case is classified as;

$$p(C|m_1,\ldots,m_n) = \alpha\, p(C)\, \Pi_i\, p(m_i|C),$$

here C is the case variable which has to be predicted and $m_1, m_2, \ldots, m_n$ are the already interposed.

An another model to calculate the continuous trends such as **Gaussian Naïve Bayes model** claims itself as an extension of Naïve Bayes [1] in which we substitute the probability density of distribution in the subsequent equation where we use μ as a mean and σ as the variance for variable m;

$$p(m) = 1/(\sqrt{\pi\sigma})e^{-(m-\mu)sq/(2\sigma sq)}$$, here π is the mean and σ is the variance

that needs to be calculated independently here by using the following formulas:

$$\mu = \sum_j m_j/N \text{ , and , } \sigma = \text{sqrt.}[(\sum(m_j-\mu)^2/N]$$

PROBABILISTIC MODELS MATHMETICS

Probabilistic models are central to Machine Learning and Artificial Intelligence. It is considered as a statistical technique majorly used to predict the likely future outcome of incidents using randomized events or actions. Probabilistic models contain three model specified categories in them, [16] which are as follows:

1.1. ***Hidden Markov models (HMM)*:**

It is used to explain the evolvement of components that tend to be not straight noticeable, in simple words, operate the characteristics the probability of any random process.

Let X_k be a discrete state of a variable whose values can be denoted by integers 1,2, 3,…,S, and S stands for the number of possible states that can take place.

The transitional model $p(X_k|X_{k-1})$ becomes SxS matrix T, where $T_{ij} = p(X_k=j \mid X_{k-1}=i)$, T_{ij} is the change in the probability from state i to state j.

Here to resolve a transformation we usually put sensory models in the matrix form where e_k is the evidence variable at time k, that needs to be specified for each condition using $p(e_k|X_k=i)$ for each condition i keeping the (O_k),i^{th} diagonal entry $p(e_k|X_k=i)$ and other values 0.

After using column vectors in the model, the forward and backward equations come out to be as follows respectively;

$$f_{1:k+1} = \alpha\, O_{k+1}\, T^k f_{1:k} \text{ and,}$$

$b_{u+1:k} = TO_{u+1}b_{u+2:k}$

1.2. *Kalman Filter*:

The optimize estimator with linear dynamic schema and Gaussian noise models is known to be Kalman filter:

$p(X_{k+\Delta}=x_{k+\Delta}|X_k=x_k,X_k=x_k)=N(x_k+x_k\Delta,\sigma^2)(x_{k+\Delta})$,

where Δ is the time taken between the observations, constant velocity during interval, and $X_{k+\Delta} = FX_k + X\Delta$ is the transition model of the state where F is the state transition matrix and $X\Delta$ is the noice created in the process. The discussed equation can be further updated according to the problem and the requirements.

1.3. *Dynamic Bayesian Networks (DBN)*:

DBN acts as an another form of bayesian networks which plays a significant role to relate different variables from an adjacent time laps with each other [16]. To build a DBN, we require three basic information;

$p(X_0)$: distribution for the state variable,

$p(X_{k+1}|X_k)$: transitional model,

$p(e_k|X_k)$: sensory model.

APPLICATIONS

Artificial Intelligence, Machine Learning and Deep Learning are predominantly leading as a chief source of innovation [3]. Every Computer scientists and mathematicians are using Artificial Intelligence and Machine Learning to solve and suggest new mathematical solutions in the most complex fields i.e. knot theory and representation theory. Machine Learning is basically used here to assist all the work of analysis of exacerbate datasets and suggest possible lines of bombard the unproven proposals in mathematics [2]. There are a plenty of taxonomy of applications related to this field such as:

1. **Drug Discovery:**

It can be estimated how mixing some complex compound drugs will behave and if they have the potential to work in a progressive manner or not [14].

2. **Medical Imaging:**

We use technologies like Convolutional Neural Networks (CNN) to identify different anomalies and diseases that have the potential of taking in someone's body [4].

3. **Virtual Health assistant:**

All around the world, 24/7, AI chatbots and virtual machines are available for people to access them and ask for basic health advice which they might can also use in case of emergencies when meeting a doctor in person is not possible.

4. **Algorithmic Trading:**

There are AI systems available in market which are used for analysing the markets [2] and there trends and predict the movements on which one can generate profits.

5. **Visual Search:**

Using Deep Learning the users can directly use the images and web or application search for something or someone they want to know about [11].

6. **Automation:**

AI and Automation allows us to do the tasks with just a single click without any human interruptions which can save humans time and can help big MNCs to use their time on innovations.

7. **Route Optimization:**

We use Machine Learning and Data Structures together to shorten the routes between two places [14]. For example; Google Maps, Apple Maps.

8. **Content Creation:**

As in today's life we are watching ourselves we are using the Generative AI to create new content which includes images, text, videos and audios as well [11].

9. Recommendation Systems:

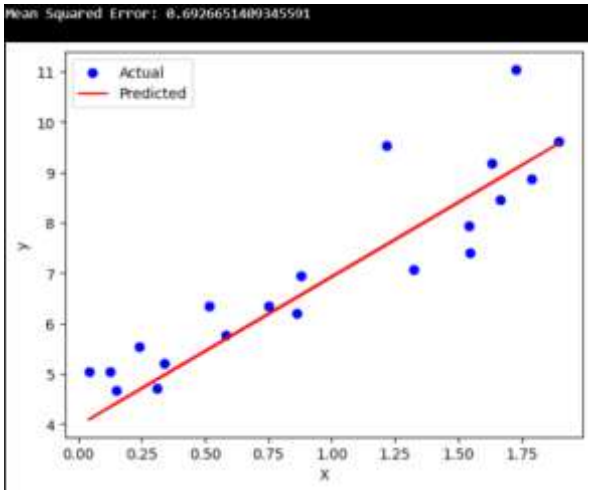
We see some platforms like Netflix [8] are using recommendation system in order to grab the watching pattern of different people around the globe and individual interests to recommend the next content one would like to watch next.

10. Energy Forecasting:

AI and ML predicts the resources and planning that can be in needed in future after analysing the energy production and consumption patterns.

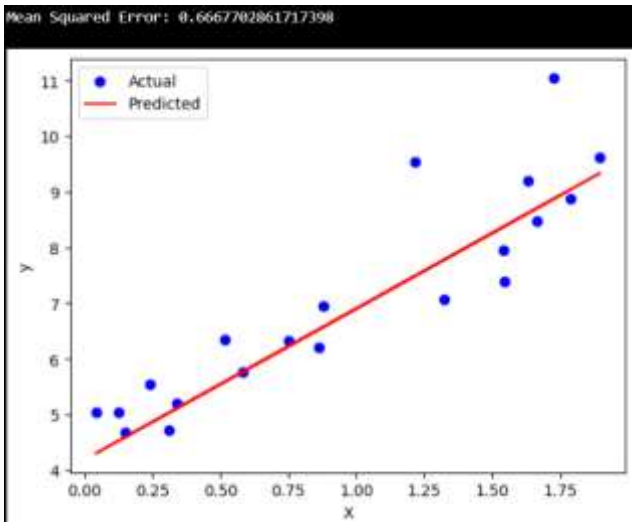
RESULTS:

1. Linear Regression

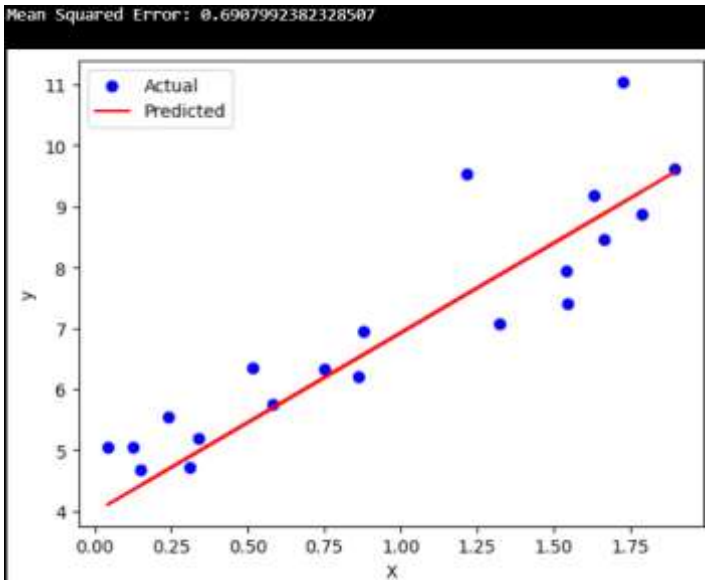


2. Logistic Regression

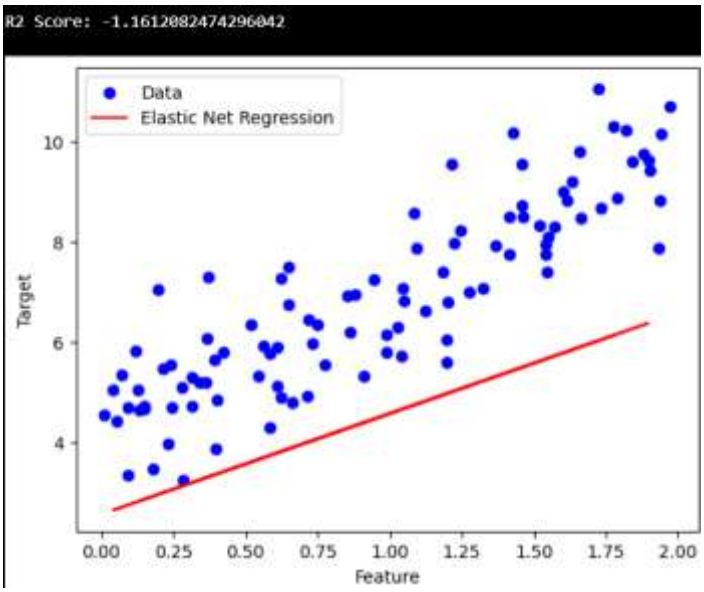
3. Lasso Regression



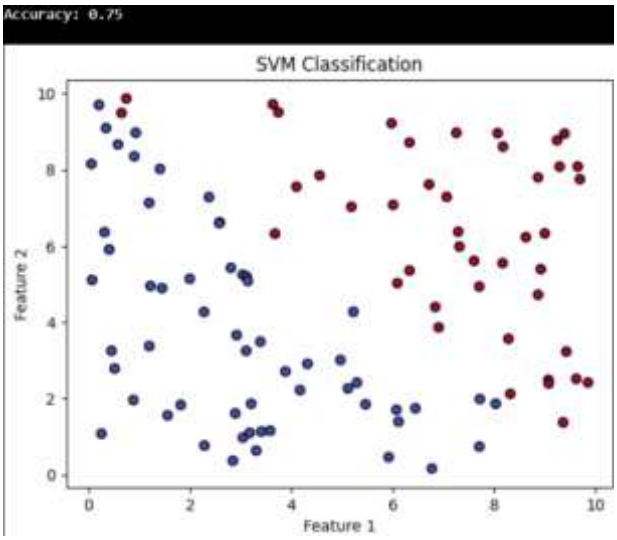
4. Ridge Regression



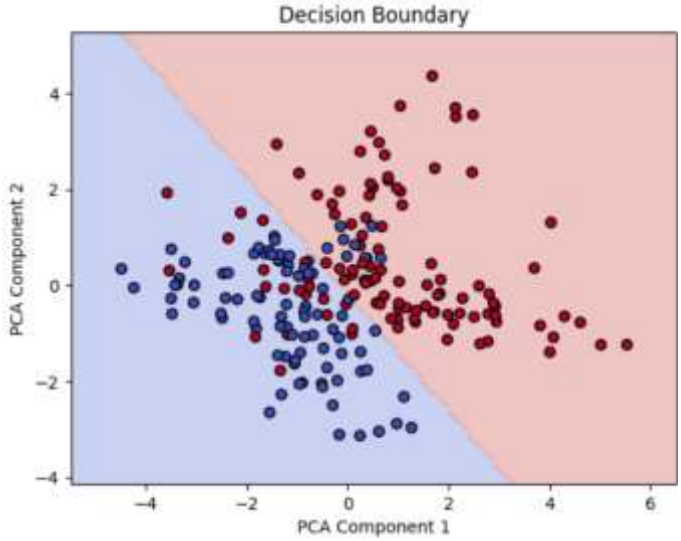
5. Elastic Net Regression



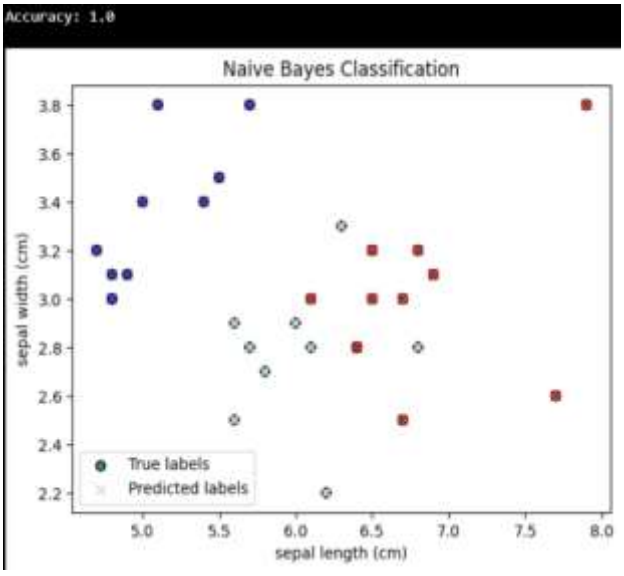
6. Support Vector Machine



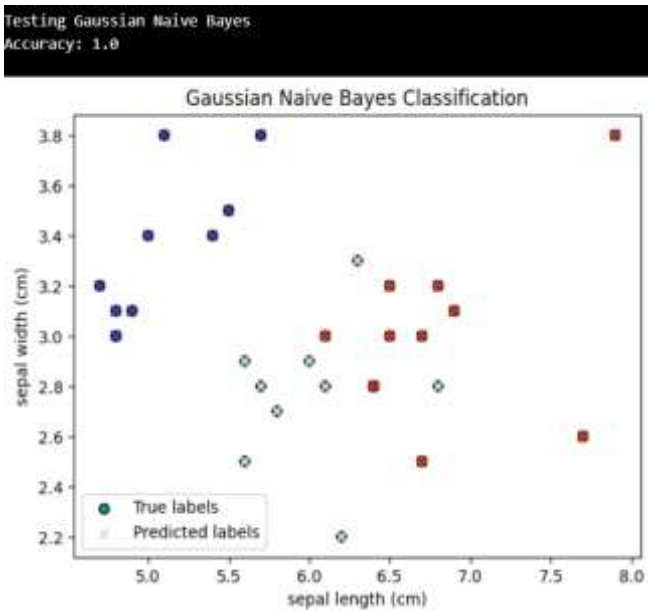
7. Principle Component Analysis



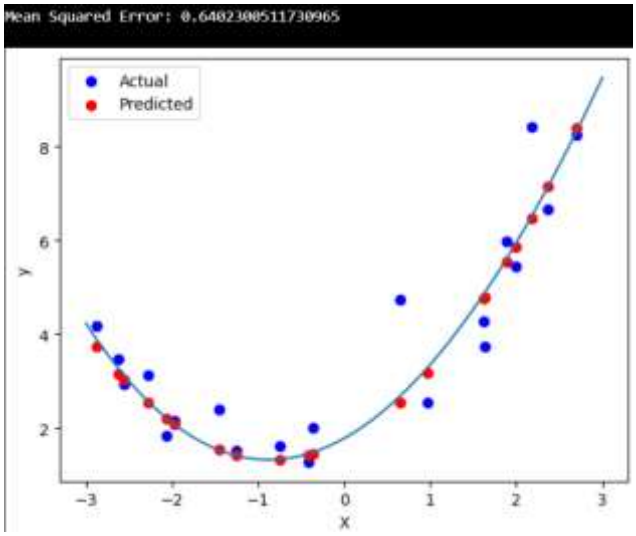
8. Naive Bayes Classification



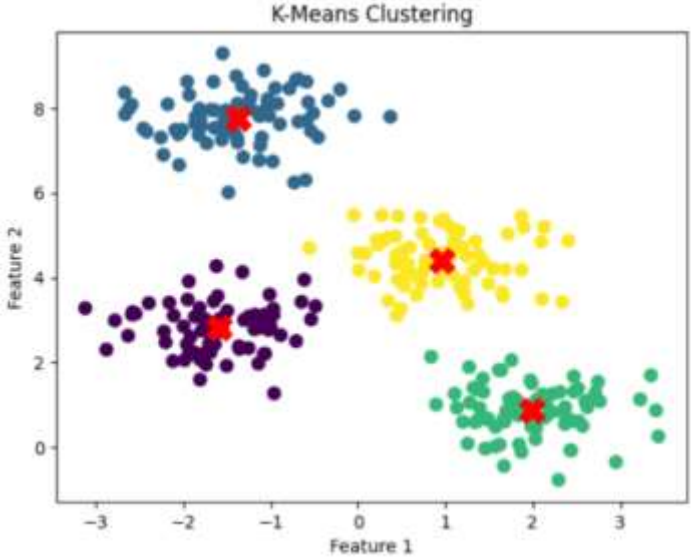
9. Gausian Naive Bayes Classification



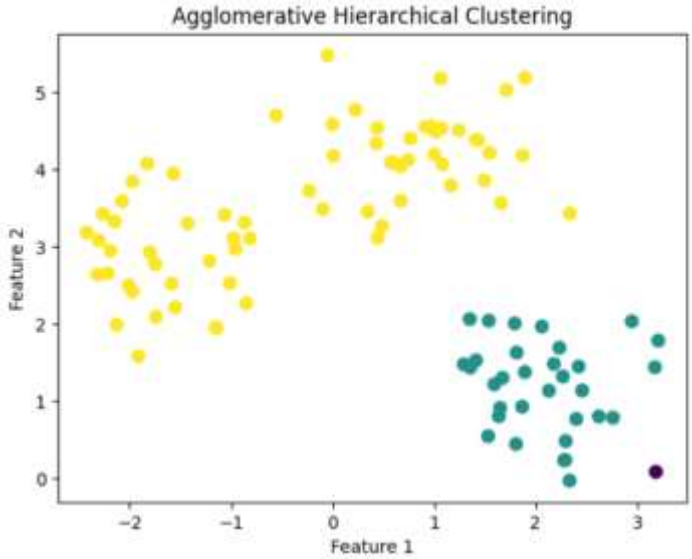
10. Polynomial Regression



11. K Means Clustering



12. Hierarchial Clustering

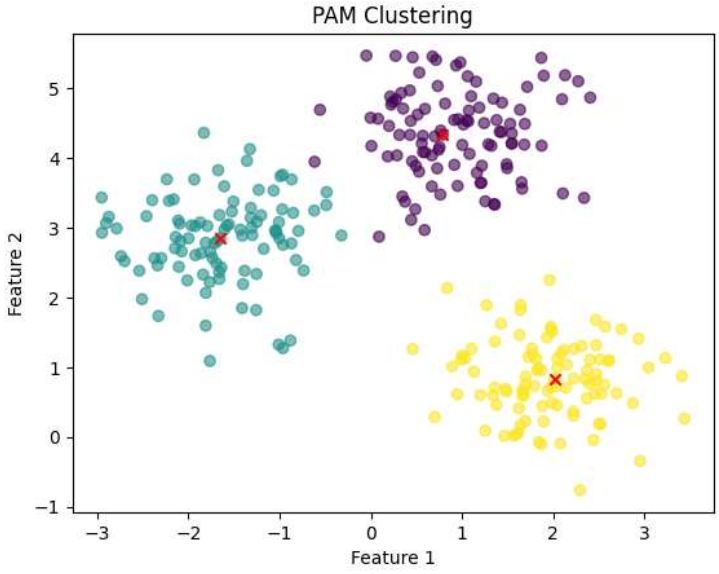


13. Apriori Algorithm

```
Frequent Itemsets:
frozenset({'bread'})
frozenset({'eggs'})
frozenset({'milk'})
frozenset({'eggs', 'milk'})
frozenset({'bread', 'eggs'})
frozenset({'bread', 'milk'})

Support Values:
frozenset({'bread'}): 0.8333333333333334
frozenset({'eggs'}): 0.6666666666666666
frozenset({'milk'}): 0.6666666666666666
frozenset({'eggs', 'milk'}): 0.5
frozenset({'bread', 'eggs'}): 0.5
frozenset({'bread', 'milk'}): 0.5
```

14. PAM Clustering



CONCLUSION:

Continuous advancements in the Artificial Intelligence, Machine Learning and Deep Learning has poised to evolve numerous sectors and fields in multiple industries including healthcare, finance, energy, entertainment and gaming [2][3][8] which can be further taken forward in future and can break the distance between virtual reality and augmented reality. Mathematics provides the basic but needful tools and framework which is used to analyze, predict and solve complex problems. Looking at the current working of mathematics having the capabilities to learn patterns from data, predict the requirements and adapt the trends, we expect more advancements in mathematical techniques which can push technological boundaries in multiple domains.

FUTURE SCOPE:

1. **Personalised Medicines:**

We can use AI and ML in order to prescribe medicines to the people who are at some emergency or may not have enough amount to pay on every visit to doctors for every small and big diseases.

2. **Robotic Surgery:**

In future, we might achieve a point where we will be able to see robots performing surgeries and operations on the humans standing on the doctor place.

3. **Advanced Fraud Detection:**

Even after having so many systems to monitor and detect the online frauds, still there are multiple frauds taking place outside we might achieve a goal of the least amount of frauds taking place in the world.

4. **Artificial Reality:**

Even though we are having multiple platforms which provide us the feature of trying the products virtually before buying them, it is expected to see more realistic and friendly changes in future.

5. **Air Mobility:**

AI will allow us to develop and operate the flying vehicles including cars, trains and buses for transportation in future.

6. **Standard Control:**

Using Real-Time data to ensure high accuracy and achieve high standards by performing the real time inspections on the system.

7. **Enhanced Gaming:**

We will see more realistic and personalized gaming experience in future which might have adaptive characters in them.

8. **Quantum Computing:**

It can open the gates to solve many unsolved questions and unfold number of paradoxes scientists have been facing since ages.

REFERENCES:

1. Theodore Grether-Murray (2022), "The math behind A.I: From Machine Learning to Deep Learning", www.medium.com
2. Sopan Talekar (2023) ,"THE ROLE OF MATHEMATICS IN MACHINE LEARNING", ResearchGate.
3. Elvir Cajic (2024), "Application of mathematics in artificial intelligence", ResearchGate.

4. Prof. Neha Saini (2023), "RESEARCH PAPER ON ARTIFICIAL INTELLIGENCE & ITS APPLICATIONS", IJRTI
5. SN Computer Science (2021), "Machine Learning: Algorithms, Real-World Applications and Research Directions", A Springer Nature Journal
6. Mahind Rupali, Patil Amit (2017), "A Review Paper on General Concepts of 'Artificial Intelligence and Machine Learning'", International Advanced Research Journal in Science, Engineering and Technology.
7. Bing Liu, "Supervised Learning", Department of Computer Science, University of Illinois at Chicago (UIC), 851 S. Morgan Street, Chicago.
8. Zoubin Ghahramani, "Unsupervised Learning", Gatsby Computational Neuroscience Unit, University College Lond Unsupervised", "Genetic Learning Algorithms", "Reinforcement Learning and Control", Department of Computer Science, Stanford University,450 Serra Mall, CA 94305, USA.
9. Artificial Intelligence: A Modern Approach, Stuart J. Russell, Peter Norvig, 2016
10. Deep Learning and Mathematical Intuition: A Review of (Davies et al. 2021)
11. Zhiying Hao (2019), Deep learning review and discussion of its future development, MATEC Web of Conferences 277(C):02035
12. IBM webpage, www.ibm.com/topics/deep-learning
13. Richard E. Bellmanm, Wikipedia, https://en.wikipedia.org/wiki/Bellman_equation
14. Lars Mescheder, Andreas Geiger, Sebastian Nowozin (2018), "Which Training Methods for GANs do actually Converge?", International Conference on Machine Learning 2018
15. Corinna Cortes, Mehryar Mohri, Afshin Rostamizadeh (2012), "L2 Regularization for Learning Kernels", Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI2009)
16. Jonathan Ho, Ajay Jain, Pieter Abbeel (2020), "Denoising Diffusion Probabilistic Models", NIPS Papers