Machine Learning Algorithms

Ashok Kumar Meghvanshi

Department of Computer Science and engineering

Lovely Professional University, Phagwara Punjab, India

Abstract – This term paper about machine learning. Diff-Diff types of machine learning algorithms have been discussed here. These ML algorithms are used in data mining, image processing, speech recognition etc. These algorithms easily classify a group of data and remove noisy data from a set of data once we have to teach the algorithm then it works automatically.

Keywords **– Machine learning, algorithm, pseudo code, programming code, applications, graph, diagram.**

I. INTRODUCTION

Machine learning is a branch of computer science and subfield of artificial intelligence (AI) that deals with the techniques of teaching the instructions to the machine so that machine is able to do work or recognize its task to do automatically and gives a good result with full of correct accuracy to the user.

Machines are able to handle the large datasets with the help of machine learning algorithms. If we put large datasets into machine then, that machine makes pattern between same data and extract some useful information from that data and remove noise data from datasets. Now, the time many industries and companies are using machine learning.

All the types of machine learning are explained in Section 2.

All types of learning given in Fig. 1.

Applications of Machine Learning: -

- Image and Speech Recognition.
- Traffic Prediction.
- Virtual Personal Assistant.

II. MACHINE LEARNING METHODS

❖ *Supervised Learning*

The supervised machine learning algorithms are depending on the concept of how teachers teaching and train their students. As same in this learning user have to input a well labeled and well-trained data to the machine then that machines are able to predict the correct answer with the help of ML algorithms.

Applications of Supervised Learning: -

- Face Recognition.
- Spam Classification.
- Fraud Detections.

❖ *Unsupervised Learning*

The unsupervised machine learning algorithms are opposite of supervised learning. In this learning user input unlabeled data to the machine then machine makes pattern, cluster and make group of same data and predict the correct output and organize in that manner so that user can be easily understand that unlabeled data.

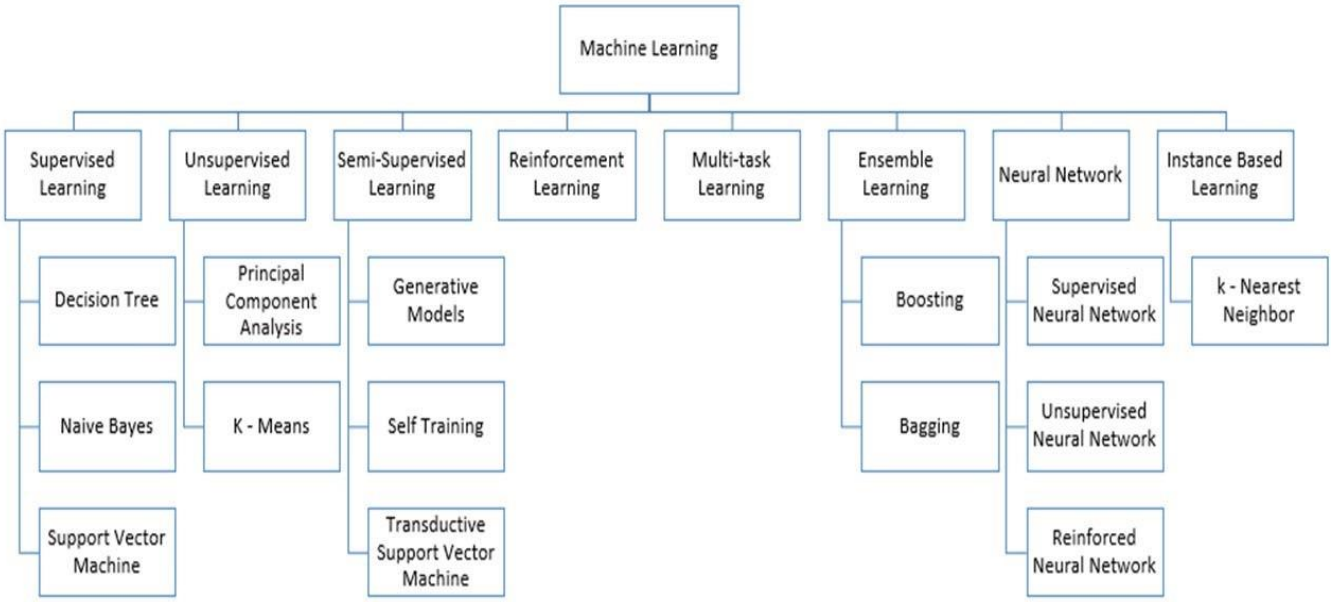

Fig. 1. Types of Learning [1] [1]

Applications of Unsupervised Learning: -

- Recommendation System.
- Item Categorization.
- Clustering of Customer.

❖ *Reinforcement Learning*

The reinforcement machine learning is totally different from supervised and unsupervised learning. In reinforcement learning our algorithm make so much mistake at the beginning but after some time our algorithm learns from its past experience and past behaviors. Reinforcement learning mainly focus on how our algorithm learn from feedback in every iteration in order to maximize its performance and give a better result to the user with full of accuracy.

Applications of Reinforcement Learning: -

- Playing Games.
- Self-Driving Cars.
- Robotics.

All the methods of supervised and unsupervised learning is explained in Section 3.

III. LEARNING METHODS

➢ *Supervised Learning*

1) Classification supervised learning

Classification is a method of supervised machine learning algorithms. In classification client input either labeled or unlabeled data to the machine with some algorithm then machine use those algorithms and categorize into different-different classes. In some datasets these classes are target function.

Classification method is given in Fig. 2.

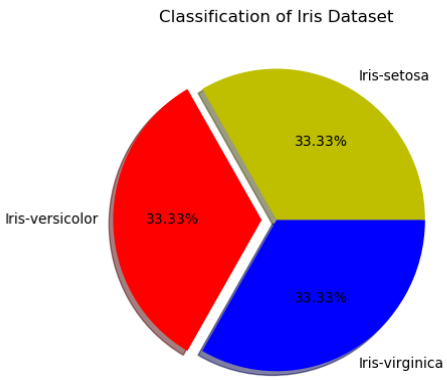

Fig. 2. Classification of Iris Dataset [3][1].

2) Regression supervised learning

Regression is a method of supervised machine learning algorithms. In Regression machine makes a relationship between dependent variable and independent variable. The main task of machine is that predict the correct value of dependent variable by putting different-different values of independent variable. In this we have some linear, quadratic, cubic equations.

Hypothesis function for linear regression is given in Fig. 3.

$$y = \theta_1 + \theta_2.x$$

Fig. 3. Equation of linear regression [3][2].

Regression method is given in Fig. 4.

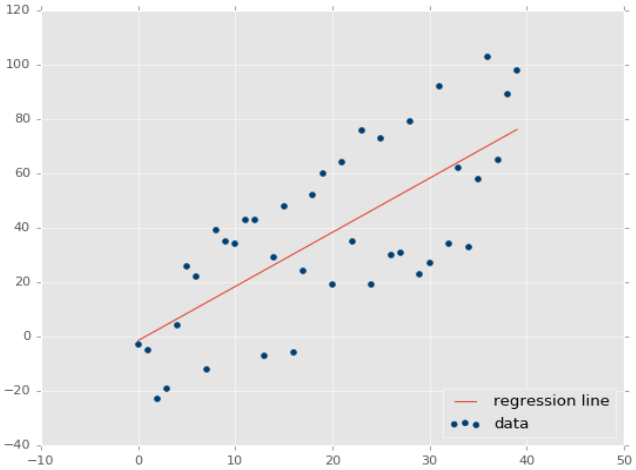

Fig. 4. Regression of Weather Dataset [3][3].

➢ *Unsupervised Learning*

1) Clustering unsupervised learning

Clustering is a method of unsupervised machine learning algorithms. The main task of this algorithm is to identify group of similar objects with the help of their same properties and same feature while separate dissimilar object in different group. This technique is also used for statistical data analysis. Mainly clustering techniques is mostly used in marketing so that companies easily identify similar type of customer.

Clustering method is given in Fig. 5.

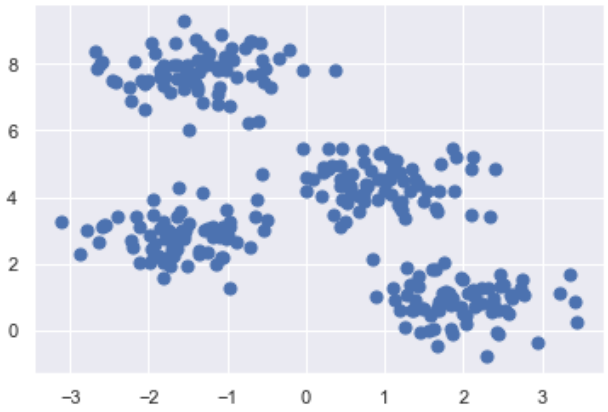

Fig. 5. Clustering of Walmart Dataset [3][4].

All most popular algorithm of supervised learning is discussed in Section 4.

IV. ALGORITHMS OF SUPERVISED LEARNING

1) *Neural Network*

Neural network is one of the important learning algorithms in supervised learning. Neural networks work on the basis of neurons and each neuron are connected to each other. Neural networks are associated with some weight and bias. Each neuron has activation function and those activation functions are multiply both input data and weight then calculate correct output. If we get any error in the output then our algorithm adds or subtract some bias in the output so that we can remove error from output and get a good result with full of accuracy.

There is different type of activation function like Linear, Binary, Bipolar, Sigmoid, Tanh, ReLU.

Activation Function is given in Fig. 6.

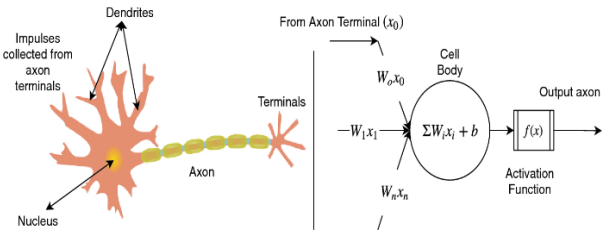

Fig. 6. Activation Function [4][1].

Activation function provide non-linearity in output of neurons. It converts the output in normalize form and fix the output of each neuron in range between 0 to 1 or -1 to 1.

Artificial neural network is given in Fig. 7.

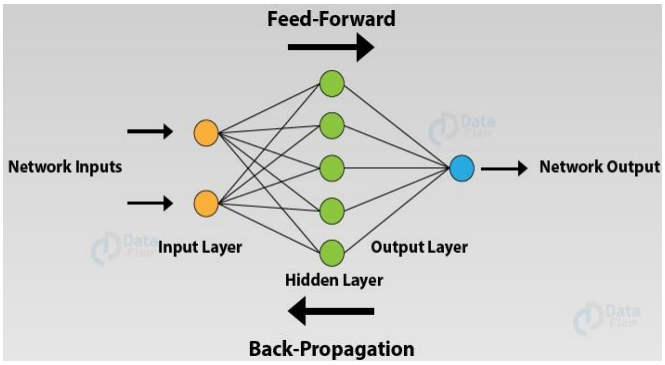

Fig. 7. Artificial neural network [4][2].

Programming code for neural network: -

```
import numpy
import random
import os

lr = 1
bias = 1

w1= random.random()
w2= random.random()
w3= random.random()
weights = [w1, w2, w3]

# lr = learning rate of neural network
# bais = value of bias
# w1,w2, w3 = weights of neural network

def Perceptron(input1, input2, output):
    outputP = input1*weights[0]+
              input2*weights[1]+
              bias*weights[2]

    if outputP > 0:
        outputP = 1
    else:
        outputP = 0

  error = output - outputP
  weights[0] += error * input1 * lr
  weights[1] += error * input2 * lr
  weights[2] += error * bias * lr

for i in range(50):
  Perceptron(1, 1, 1)
  Perceptron(1, 0, 1)
  Perceptron(0, 1, 1)
  Perceptron(1, 1, 0)

x = int(input())
y = int(input())
outputP = x*weights[0] + y*weights[1]
        + bias*weights[2]

if outputP > 0:
    outputP = 1
else:
    outputP = 0

print(x, "or", y, "is : ", outputP)

#sigmoid function
outputP = 1/(1+numpy.exp(-outputP))
```

2) *Decision Tree*

Decision tree is a popular algorithm in supervised learning. It is one of the most widely used and effective methods of supervised learning. Decisions is an advanced unreadable learning process that is used to perform segmentation and sorting tasks. The goal is to create a model that predicts the number of target variables by reading simple decision rules derived from data symbols.

The choice of the algorithm also depends on the type of the target variables. Let us look at some of the algorithms used in decision trees: -

ID3 → (extension of D3)
C4.5 → (Successor of ID3)
CART → (classification and regression tree)
CHAID Ch (Chi-Square Automatic Interaction Detection

performs multi-level partitions when computing classification trees)
MARS → (Multivariate Adaptive Regression Splines)

Decision tree is given in Fig. 8.

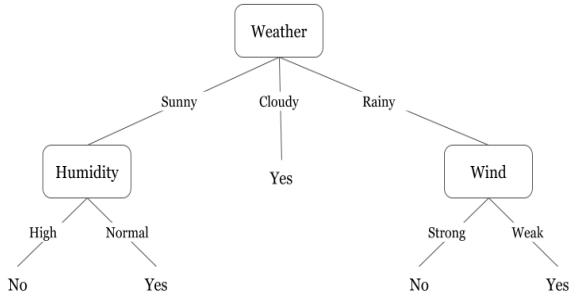

Fig. 8. Decision tree diagram [4][3].

Programming code for decision tree: -

```
import numpy as np
from sklearn import datasets
wine=datasets.load_wine()
x=wine.data
y=wine.target

print('======Standard Scaler======')
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_std=sc.fit_transform(x)

print('======Min Max Scaler======')
from sklearn.preprocessing import MinMaxScaler
mms=MinMaxScaler()
x_norm=mms.fit_transform(x)

print('======Train Test Split ======')
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_std,y,test_size
=0.3,random_state=0)

print('====Decision Tree Classifier ==')
from sklearn.tree import DecisionTreeClassifier
tr = DecisionTreeClassifier(criterion='gini',max_depth=3)
tr.fit(x_train,y_train)
y_pred=tr.predict(x_test)

from sklearn.metrics import accuracy_score
test_acc = accuracy_score(y_test,y_pred)
print('Testing Accuracy : ', test_acc*100)
```

3) Naive Bayes

Naive Bayes classifier is a popular algorithm in supervised learning. In Naive Bayes classifier the main motive of our algorithm is to calculate conditional probability with the help of Bayes Theorem. The Naive Bayes are subject conditions. It building trees as far as possible. These trees are also known as the Bayesian network. In Bayes Theorem our algorithm calculates conditional probability when event A is given and event B already occur.

Bayes Theorem formula is given in Fig. 9.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Fig. 9. Bayes Theorem formula [4][4].

A, B = events.
P(A|B) = probability of A given B is given(true).
P(B|A) = probability of B given A is given(true).
P(A), P(B) = the independent probabilities of A and B.

Programming code for Naive Bayes: -

```
import numpy as np
from sklearn import datasets
iris=datasets.load_iris()
x = iris.data
y = iris.target

print('======Train Test Split ======')
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_std,y,
                                test_size=0.3,random_state=0)

from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)

from sklearn import metrics
print("Gaussian Naive Bayes model accuracy(in %):")
print(etrics.accuracy_score(y_test, y_pred)*100))
```

4) Support Vector Machine

Support vector machine is a most important algorithm for classification problem. It is a part of supervised machine learning algorithm. It is used for classification as well as regression challenges. It works is to find a hyperplane in N-dimensional space that hyperplane classifies the given data point in different-different groups. SVM create a decision boundary between both groups of data.

Support Vector Machine is given in Fig. 10.

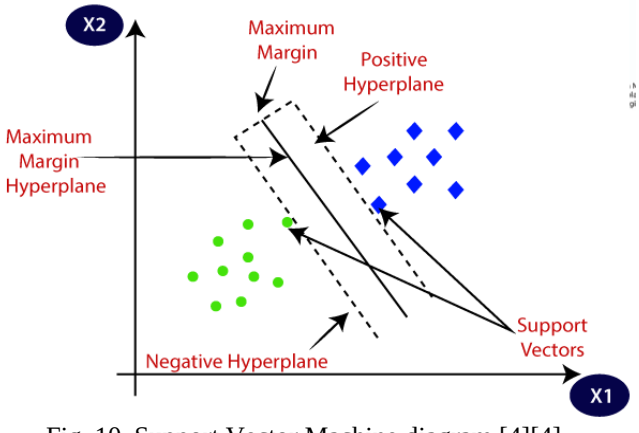

Fig. 10. Support Vector Machine diagram [4][4].

Programming code for Support Vector Machine: -

```
from sklearn import datasets
import numpy as np

iris=datasets.load_iris()
x=iris.data[:,[2,3]]
y=iris.target

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3
,random_state=0)

from sklearn.svm import SVC
svm=SVC(kernel='linear',C=1,random_state=0)
svm.fit(x_train,y_train)
y_pred=svm.predict(x_test)

print('misclassified sample: %d'%(y_test!=y_pred).sum())

from sklearn.metrics import accuracy_score
print('Accuracy:%.2f'%accuracy_score(y_test,y_pred))
```

IV. ALGORITHMS OF UNSUPERVISED LEARNING

1) *K-Means Clustering*

K-means are one of the simplest learning algorithms in unsupervised learning. The algorithm follows a simple and easy way to aggregate a given data set to a specific number of groups called clusters. K-means clustering algorithms classify a given organized data into a specific number (K) of clusters. So, they call K-means clusters. First a K number is called centroids. With the help of each centroid we can understand about the similar set of data.

K-Means clustering is given in Fig. 11.

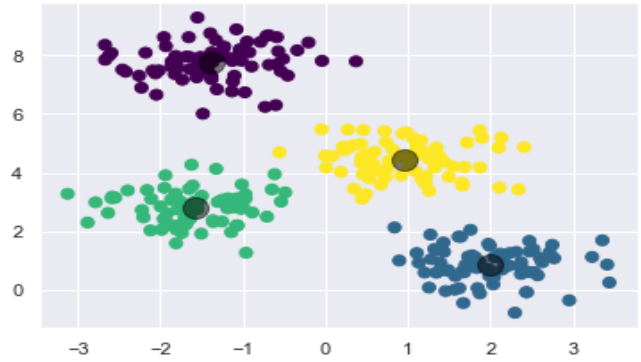

Fig. 11. K-Means clustering diagram [5][1].

There are four steps in how K-means integrates performance: -

1. Initialize Clusters of given data sets.
2. Assign data to clusters.
3. Update Cluster Clusters.
4. Repeat steps 2–3 until the stopping point is met.

2) *K-Medoids clustering*

K-Medoids select the medoids from the data points in the set. A medoid can be defined as a point in the cluster, with all other points in the cluster being at least minimized. It was proposed by Kaufman and Rousseau in 1987.A medoid is a point in the cluster, it's dissimilarities with other points in the cluster is minimum in nature. K-medoids attempts to reduce the mismatch between objects in the cluster and one of the objects designated as representative of that cluster. These representatives are known as Meadows.

Flow chart K-Medoids clustering is given in Fig. 12.

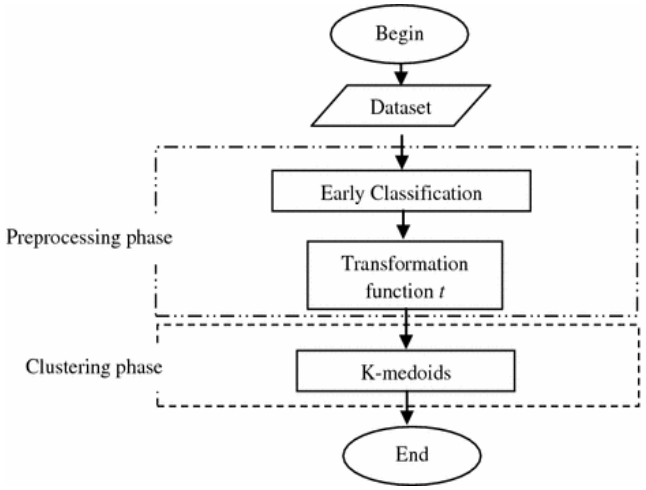

Fig. 12. Flow chart K-Medoids clustering diagram [5][2].

1) *Principle Component Analysis*

In PCA, we reduce the dimension of the data so that the computations speed of program is become faster and easier. It is an unsupervised statistical technique that examine the interrelation between a set of instances. It is statistical method that uses an orthogonal transformation which converts a group of correlated instances to a group of uncorrelated instances. PCA is used for exploratory data analysis in machine learning for predictive models.

Programming code for PAC Learning: -

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.datasets import load_breast_cancer
                                    cancer = load_breast_cancer()
df = pd.DataFrame(cancer['data'], columns =
                                        cancer['feature_names'])
df.head()

from sklearn.preprocessing import StandardScaler
scalar = StandardScaler()
scalar.fit(df)
scaled_data = scalar.transform(df)

from sklearn.decomposition import PCA
pca = PCA(n_components = 2)


pca.fit(scaled_data)
x_pca = pca.transform(scaled_data)
```

```
plt.figure(figsize =(8, 6))
plt.scatter(x_pca[:, 0], x_pca[:, 1], c =
                    cancer['target'], cmap ='plasma')

plt.xlabel('First Principal Component')
plt.ylabel('Second Principal Component')
```

PAC Learning is given in Fig. 13.

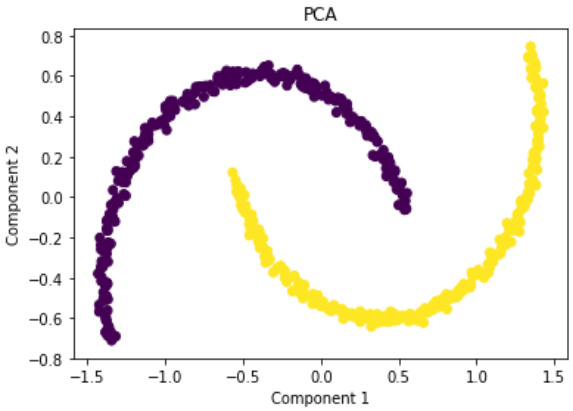

Fig. 13. PAC Learning diagram [5][3].

V. ALGORITHMS OF ENSEMBLE LEARNING

❖ *Ensemble Learning*

The Ensemble learning algorithms is an important algorithm in machine learning. It combines multiple models or algorithm and improve our model performance. It gives better result than a single model. It decreases the variance (bagging) and bias (boosting) and improve predictions(stacking). It mainly focuses on the majority voting of our data points

There are two types of Ensemble learning: -

1) *Bagging*

Bagging is a part of ensemble learning. It mainly uses the random forest method It is a meta Algorithm and it is used for classification as well as regression. It provides better stability and accuracy of our learning algorithm and it also decrease variance and avoid over-fitting. It gives equal weight to all model and combine same type of models. It creates a collection of weak learners and convert them to one strong learner. A weak learner is barely correlated with correct classification. On the other hand, a strong learner is strongly correlated with correct classification.

Pseudo code for Bagging given in Fig. 13.

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

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

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

2) *Boosting*

Boosting is also a part of ensemble learning. It mainly uses the gradient boosting. It modifies our data distribution and reduce both variance and bias. In boosting noise data and outliers can change our algorithm accuracy or performance. The main task of boosting to find strong learner that gives better and efficient performs. In Boosting training, our algorithm allocates higher weights to good classification learner that poor learner.

VI. CONCLUSION

This term paper is all about machine learning algorithms. Now in our world new technologies are coming every day. machine learning is a one of the most important technology from them. With the help of these algorithm We can design our website in that manner so that we can recommend product on our online shopping website. Apart from them we can also detect any face, image, sound etc. with the help of these ML algorithms.

VII. REFERENCES

[1] W. Richert, L. P. Coelho, "*Building Machine Learning Systems with Python*", Packt Publishing Ltd., ISBN 94-1-77716-170-0

[2] Allen B. Downey, "Think Stats", O'Reilly. ISBN 978-1-78216-140-0

[3] David Barber, "Bayesian Reasoning and Machine Learning"

[4] D. Lowd, P. Domingos, "Naïve Bayes Models for Probability Estimation"

[5] Shai shalev- Shwatz and Ben-David, "Understanding Machine Learning: From Theory to Algorithm", Cambridge University Press.

[6] https://en.wikipedia.org/wiki/Boosting_(machine_learning)

[7] https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm

[8] https://towardsdatascience.com/k-means-clustering-from-a-to-z-f6242a314e9a

[9] https://www.geeksforgeeks.org/ml-linear-regression/

[10] https://towardsdatascience.com/how-do-artificial-neural-networks-learn-773e46399fc7

[11] https://towardsdatascience.com/simple-guide-for-ensemble-learning-methods-d87cc68705a2

[12] https://www.geeksforgeeks.org/comparison-b-w-bagging-and-boosting-data-mining/

[13] https://www.researchgate.net/publication/220691633_Natural_Language_Processing_with_Python

[14] https://www.digitalocean.com/community/tutorials/an-introduction-to-machine-learning

[15] https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f

[16] https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf

[17] http://www.academia.edu/Documents/in/Machine_Learning

[18] https://www.kaggle.com/hammadzahidali/classification-tutorial-machine-learning