Disorder classification using Artificial IntelligenceTechniques

A Major Project Report

Submitted in the partial fulfillment of the requirements for

the award of the degree of

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

By

Name	Id no
U. Naga Pujitha	180030889
B. Lalasa	180030882

Under the Esteemed Guidance of

Mrs. M.V.B.T Santhi

Assistant Professor

Koneru Lakshmaiah Education Foundation

(Deemed to be University estd., u/s 3 of UGC Act 1956)

Greenfields, Vaddeswaram, Guntur (Dist.), Andhra Pradesh - 522502

November 2021

K L University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

The project report entitled "**Disorder classification using Artificial Intelligence Techniques"** is a record of bonafide work done and submitted by U.Naga Pujitha-180030889 and B.Lalasa-180030882 in partial fulfilment for the award of Bachelor of Technology in Department of Computer Science Engineering to the K L University. The results embodied in this report have not been copied from any other Departments/University/Institute.

180030889	U.Naga Pujitha
180030882	B.Lalasa

K L University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "**Disorder classification using Artificial Intelligence Techniques**" is a bonafide work done and submitted by by U.Naga Pujitha-180030889 and B.Lalasa-180030882 in partial fulfilment for the award of Bachelor of Technology in Department of Computer Science Engineering to the K L University is a record of bonafidework carried out under our guidance and supervision. The results embodied in this report have not been copied from any other Departments /University / Institute.

Signature of the Supervisor

(Assistant Professor)

Signature of the HOD

Mr. V. Hari Kiran

Signature of the External Examiner

ACKNOWLEDGEMENT

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our sincere thanks to all those who have assisted us in one way or the other for the completion of my project.

Our greatest appreciation to our guide **"MRS. MVBT Santhi"** Associate Professor, Department of Computer Science and Engineering which cannot be expressed in words for her tremendous support, encouragement, and guidance for this project.

We express our gratitude to **Mr. V. Hari Kiran,** Head of the Department for Computer Science and Engineering for providing us with adequate facilities, ways, and means by which we are able to complete this project.

We thank all the members of the teaching and non-teaching staff members, and also who have assisted us directly or indirectly in the successful completion of this project.

ABSTRACT

Medical services rehearse incorporate gathering a wide range of patient information which would help the specialist accurately analyse the medical issue of the patient. The information we incorporate like detailed examination of test consequences of the patient or breaking down the indications of that patient for the therapy of the individual in a suitable clinical therapy for recuperation level of the patients as information records. Thus, these information records are used for investigation by a specialist who at that point discovers the sickness utilizing his/her own clinical mastery. By analyzing the information and making an AI framework that groups the sickness of the patient i.e., Cancer, Diabetes, and so on from the grouping, we investigate distinctive AI strategies incorporate ML, and profound learning procedures are determined and looked at. The re-enactments are made by some pre-pre-processing methods and order procedures on the dataset. The results of the Machine Learning and Deep Learning techniques i.e., application of different algorithms on a dataset are measured, as well as the nature and complexity of the algorithms.

Keywords: Machine Learning, SVM, Naïve Bayes, Random Forest, K-NN, Multi-Layer Perceptron. Deep Learning, Neural Networks.

TABLE OF CONTENTS

List of Figures:

List of Tables:

1. INTRODUCTION

In advanced times, there is a great deal of progress in the human way of life, and it affected an enormous effect on the wellbeing and life examples of individuals. The new logical methodologies in the field of clinical consideration and fix are making earlier finding the illnesses by utilizing logical methodologies like information mining, AI instruments including Machine learning strategies, and profound learning on the records of the patients on different boundaries of their clinical records.

The main aim of the work is to predict the disease of the patients with the feature extraction and feature selection of the features i.e., applying Dimensionality reduction on the data record. As most of the real-world problems have more features for determining, but in contrast to it as the features increase, the classification can become redundant, for that using dimensionality reduction algorithms, of which PCA (Principal Component Analysis) is used to reduce the dimension of the data records. The prediction of the diseases namely Cancer, Diabetes, Mentality, Heart problems using various Al methodologies, as AI is the combination of Machine Learning classification algorithms and Deep Learning i.e., ANN (Artificial Neural Network) contrasting the classifications based on the accuracy metrics over testing records of the data.

Algorithm	Definition	Formula
SVM	The objective of the support vector machine algorithm is to seek out a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the info points.	K (x, xi) = sum (x ∗xi)
Naive Bayes	Naive Bayes classifiers are a set of classification algorithms supported Bayes' Theorem. It is not one algorithm but a family of algorithms where all of them share a standard principle, i.e., every pair of features being classified is independent of every other.	P(A|B) =P(B|A) P(A) / P(B)
Random Forest	Random forest may be a supervised learning algorithm which is employed for both classifications also as regression. But however, it is mainly used for classification problems. As we all know that a forest is formed from trees and more trees mean more robust forest.	$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(fi - yi)^2$
Multi-Layer Perceptron	A multilayer perceptron (MLP) might be a class of feed forward fake neural organization (ANN). MLP uses a directed learning strategy called backpropagation for preparing. Its numerous layers and non-straight actuation recognize MLP from a direct perceptron.	y = f(WxT + b).

K-Nearest Neighbors	The main objective of K-Nearest Neighbors algorithm is to makea new data to fit over which category based on the k-closest Neighbors of the labeled features by formulating Euclidean distance or Hamming Distance.	$P(y = j\|X = x) = \frac{1}{K}\sum_{i \in A} I(y^{(i)} = j)$
Principal Component analysis (PCA)	The main objective of the principal component Analysis is to make dimensionality reduction with features extracted i.e., making features from high dimensional to low dimensional features.	$\Rightarrow \vec{v}(A - \lambda I) = 0,$

Table 1: Brief on Algorithms

2. LITERATURE SURVEY

The Intrusion discovery frameworks are the current new moving ones that will abnormality distinguish the examples from the records. However, while Anomaly discovery frameworks are which can recognize the known and obscure assault designs from the information and the difficulties over to cause frameworks to be Anomaly was introduced by · Dhruba Kr Bhattacharyya et al. [1]

In the present-day way of life refreshes, there were wide assaults on wellbeing. An exact forecast

dependent on manifestations turns out to be excessively hard for the specialists. The right forecast of infection is the most difficult errand. To beat this difficult information mining assumes a significant part to anticipate the infection. The illness order should be possible through KNN and ANN, and correlation over it was made the most appropriate through their work results by Dhiraj Dahiwade et al. [2]

A large portion of the families is influenced by a typical issue i.e. diabetes mellitus. Most of the patients make an earlier wellbeing cognizance of the danger factors .HanWu et al [3] has proposed a more pre-prepared methodology to expand the exactness of the model and therefore, there was an increment of 3.04% over the aftereffects of different analysts.

Information mining is a developing field that changes pieces of information into helpful data. This procedure helps the approved individual make educated choices and take right choices for their betterment [4].

It is used to comprehend, anticipate and manage future conduct dependent on the secret examples among colossal datasets. It prompts refer apparatuses for robotized gaining from the historical backdrop of information and creating models to find the results of future situations. There are different apparatuses for information mining AI calculations to recognize and anticipate the different sickness as far as relapse, choice tree, and Bayesian organization [5].

The finding of infection required distinctive test brings about an assortment of situations concerning the specific patient. By applying information mining, the idea for information investigation number of tests will be decreased. It assumes an indispensable part in information examination to improve the presentation and efficient [6].

The forecast exactness of the k-implies calculation is upgraded utilizing both class and group strategy and causing it to adjust to different datasets [7].

A gathering of grouping calculations barring arbitrary woods calculation is applied on diabetes information to analyse the danger. On looking at the exhibition of each strategy, the result shows that Random Forest was performed well in both precision and ROC curve [8,9,10].

The exhibition report shows the execution measurements of the precision estimation for each class information independently. Essentially, it has prepared our three different class information with the irregular woodland calculation. It has determined the exactness of the outcomes, and this calculation gives the three classes independently utilizing the disarray framework. The exhibition report shows the execution measurements of the exactness estimation for each class information exclusively. Interior model boundaries refreshed through age, each age contains one or more groups. The age can be applied until limit the mistakes in datasets [11,12].

SVM classifier has no scourge of dimensionality since it can oversee meager information in high-dimensional datasets [13]. The hyper plane is optimized by increasing the margin. The margin is the distance between boundary and nearest point of each class. These points nearest to the boundary are called support vectors [14].

Random Forest algorithm is a blend of tree indicators to such an extent that each tree relies upon the estimations of an arbitrary vector examined autonomously and with a similar appropriation for all trees in the timberland. The speculation blunder for woodlands meets a.s. as far as possible as the quantity of trees in the backwoods turns out to be huge [15].

3. PROPOSED WORK

3.1 Background and Related Work:

Machine learning is a type of artificial intelligence when computers are programmed to learn information without human intervention. In machine learning, the underlying algorithms rely on computational statistics. Computers are provided data and then the computers learn from that data and finding its complex patterns and underlying algorithms. ML in medical services is getting even more generally utilized and is helping patients and clinicians from numerous points of view. The most widely recognized medical services use cases for AI are robotizing clinical charging, clinical decision support, and the advancement of clinical consideration rules. The Machine Learning algorithms used for classification of the disease are SVM (Support Vector Machine), Naive Bayes Algorithm, Random Forest, ANN or Multiple Layer Perceptron and K-NN algorithm.

Requirements:

1. Python (version 3.7 or higher).
2. Google Colab

3.2 Methodology:

Data Collection:

The medical records collected from medical results from the laboratory and doctor records are collected for the different diseases like Cancer, Diabetes, Mentality, and Heart respectively. The detection systems can act an important role in the absence of a doctor. To make it intellective to diagnose the patient, it should not be hardcoded, as a prior AI is required, and it can be developed through python modules.

Data Pre-processing:

The data pre-processing will handle missing null values, label encoding, grouping, and outlier removal in the data records. As, in data pre-processing, we are using feature engineering techniques that extract data features prior such that feature reduction will be using PCA algorithm.

Disease Classification:

The data records are being pre-processed and are applied to many classification models i.e SVM, Naive Bayes, Random Forest, K-NN, and Artificial Neural Network. These classification models work on their

math principle fit the data by understanding hidden patterns in the data such that the model will become an anomaly.

3.2.1 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is an extremely good, supervised learning algorithm. SVM algorithm creates decision boundaries that can separate n-dimensional space into different classes so that new points can be placed in the correct classism is used in real life for text, image classification, face detection etc.

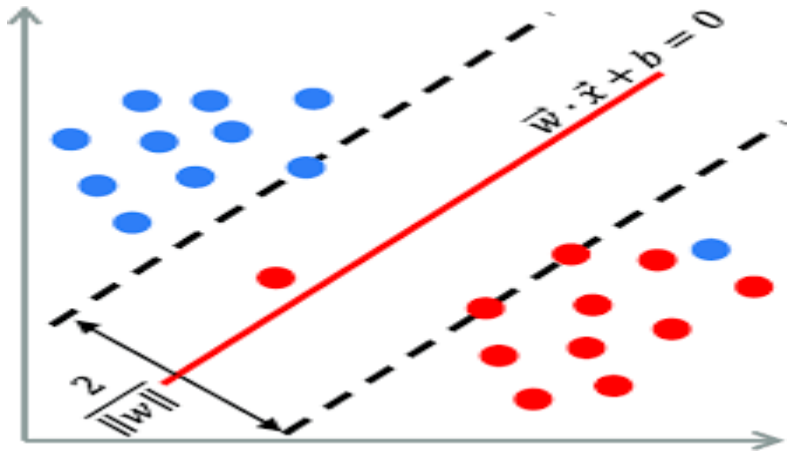
Support Vector Machine is a supervised machine learning algorithm capable ofperforming classification. SVM is of two types:

1) Linear SVM

2)Non-Linear SVM

Linear SVM

The linear Support Vector Machine classifier works by drawing a line between the two classes



Dataset divides into classes in Linear SVM

There are some important concepts to know in the SVM algorithm to divide the dataset into classes. They are Support Vectors, Hyperplane, Margin.

Hyperplane

The SVM model is fundamentally a portrayal of various classes in a hyperplane in multidimensional space. The hyperplane will be created in an iterative way by SVM so the

error can be limited. SVM or Support vector machine is the classifier that supports the edge. The target of a classifier in our model underneath is to find a line or (n-1) estimation hyper-plane that disconnects the two classes present in the n-dimensional space.The objective of SVM is to separate the datasets into classes to locate a most maximum marginal hyperplane (MMH). The hyperplane which has maximum margin is called the optimal hyperplane.

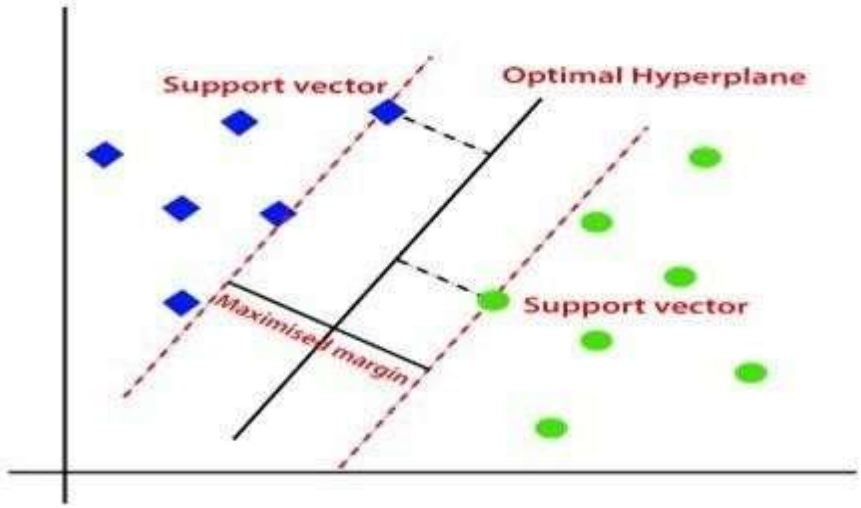
Support Vectors

Data Points that are nearest to the hyperplane are called support vectors. Lines separating them will be defined with the help of data points. With the help of datapoints, separating lines will be defined.

Margin

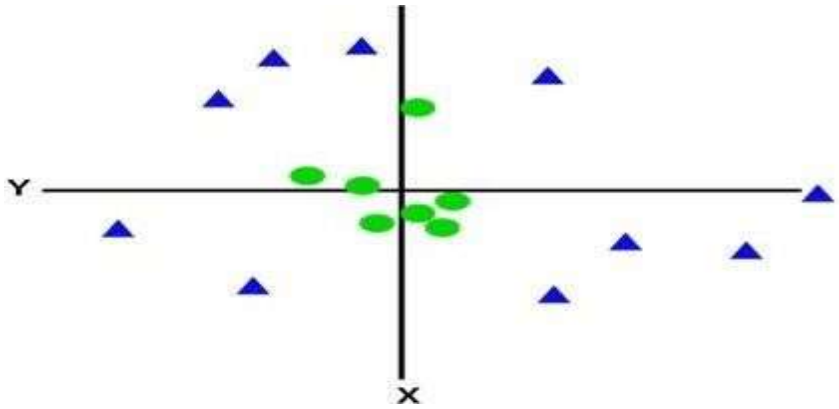
The gap between two lines or vectors on the closet data points of different classes is called Margin. which can be said as the perpendicular distance from the line to the support vectors. Large margin is said to be a good margin and a small margin is a bad margin.

- Support vector Machine will generate hyperplanes iteratively that segregates the instructions withinside the quality way.
- Then, it will choose the hyperplane that separates the classes correctly.



Non-Linear SVM

For Non-linear data, we cannot draw the straight line for data points of the non-inear svm



To separate these dimensions we need to add one more dimension, for linear data we use two dimensions for non-linear data, we use three dimensions. So the SVM divides the data points in the following way

Advantages of SVM:

- SVM classifiers can give good accuracy even in high dimensional space.
- SVM classifiers use very less memory because it uses a small part of training points. uses very less memory.
- SVM is effective when the number of dimensions is greater than the number of samples.
- SVM works properly with a clean margin of separation.

Disadvantages of SVM:

- SVM classifiers do not work well with overlapping classes.
- SVM does not carry out nicely while we take huge datasets due to the fact the training time is high.

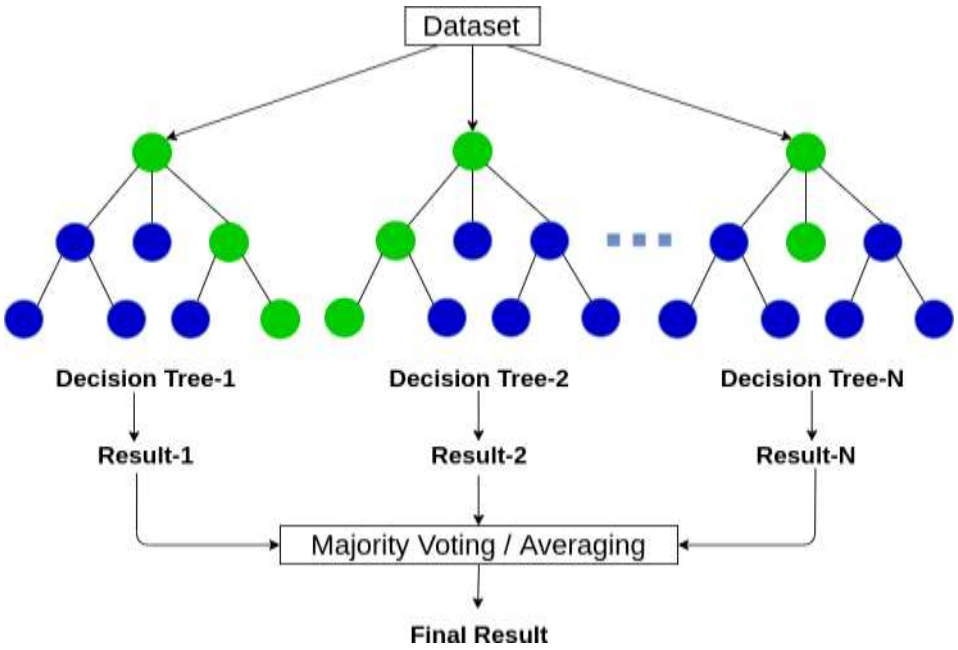
3.2.2 Random Forest

Random Forest is a supervised learning technique that can be used for Classification problems in Machine Learning. It is built based on a concept called ensemble learning, where multiple classifiers are combined to solve a complex problem and improve amodel's performance.

Random Forest is a classifier which contains many decision trees built on a given dataset and it takes average of output to improve accuracy. It uses an ensemble method which. gives better results than a single decision tree because over-fitting is reduced by averaging the figure.

The number of trees is proportional to accuracy and also prevents the problem of overfitting.

The following diagram describes about the working of Random Forest.



Random Forest works in two phases, one is to create a random forest by predicting the Ndecision trees and the second is to predict each decision tree created in the first phase.

Step-1: Selecting k data points randomly from the training set. Step-2: Creating the decision trees using the selected data points.Step-3: Choose the N decision trees to create for the model.

Step-4: Repeat step-1& step-2

Step-5: For new data points, find the predictions of each decision tree, and assign the newdata points to the category that wins the majority votes.

The random forest algorithm combines multiple trees to predict which class the output belongs to. Some decision trees might predict correct output while some may not. But all together, all the trees predict the right output. Therefore, below are two assumptions for a Random forest classifier:

- In feature variables, actual values should be present to get better accuracy.
- The predictions from each tree must have very low relations.

Advantages of Random Forest

- Random Forest overcomes the problem of overfitting by averaging or combiningthe results of different decision trees.
- Random Forest gives high accuracy and is very flexible.
- Random Forest has less variance than one decision tree.
- Random Forest works well for large datasets than one single decision tree
- Data scaling is not required in Random Forest.
- Random Forest maintains high accuracy when data is missing.

Disadvantages of Random Forest

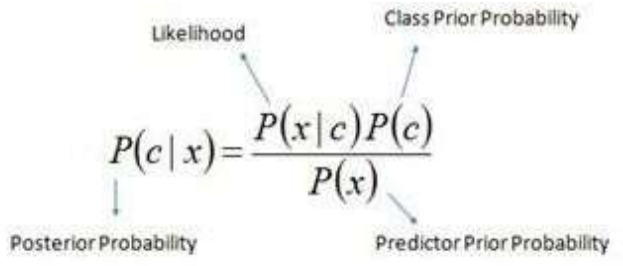
- Random Forests are not more suitable for regression tasks.
- Complexity is more for Random Forest because to create decision trees is muchharder for the algorithm and time complexity is also more.

- More Computational resources are required for the Random Forest Algorithm.
- Prediction is more difficult for Random Forest and very time consuming whencompared to other algorithms.

3.2.3 NAIVE BAYES

Naive Bayes is a supervised learning algorithm, which depends on Bayes hypothesis. In real life it is used in text classification. Naïve Bayes Classifier is one of the basic and best Classification calculations which helps in building the quick AI models that can make accurate predictions. It is based on the probability of an event occurring. So, it is called a probabilistic classifier. Real life examples of Naïve Bayes Algorithms are Sentimental analysis, spam filtration, classification of articles etc. It is called Naïve on the grounds that it expects that the event of a specific component is independent of the event of different features.

Naive Bayes is a machine learning model dependent on the Bayes hypothesis. It is a straightforward classification method; however it has high usefulness. It is useful when the dataset is large. Bayes hypothesis gives a method of computing the likelihood, P(c|x), from P(c), P(x), and P(x|c). Bayes classifiers accept that the impact of the estimation of an indicator (x) on a given class (c) is free of the estimations of different features.



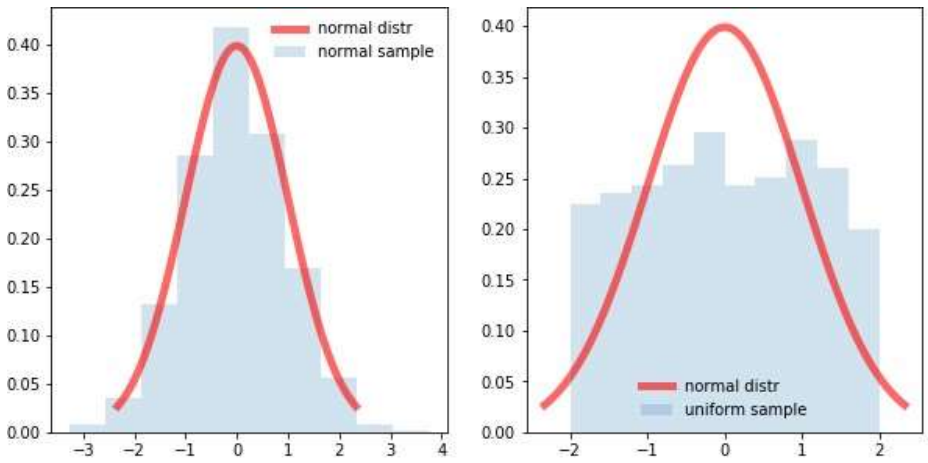
$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

P(c|x) is the likelihood of class (target) given indicator (property).

P(c) is the earlier likelihood of class.

P(x|c) is the probability which is the likelihood of the indicator

given class.P(x) is the earlier likelihood of indicator.



Bayes' Theorem finds the likelihood of an occasion happening given the likelihood of another occasion that has just happened. It is used for calculating conditional probabilities. Bayes hypothesis is expressed numerically as the accompanying condition: Naïve bayes has three types of models they are Gaussian, Bernoulli, Multinomial.

Gaussian:

The Gaussian model follows the features of normal distribution. In this model,

predictors take continuous values instead of discrete values. At that point the model accepts thatthese values are sampled from the Gaussian distribution.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Bernoulli:

The Bernoulli classifier is like the Multinomial classifier, however the predictor variables are the independent Booleans variables. This model can be utilized for classification tasks.

Multinomial:

The Multinomial Naive Bayes classifier is utilized when the information is multinomial distributed. It is fundamentally utilized for classification. The classifier utilizes the frequency of words for the predictors.

Advantages of Naive Bayes

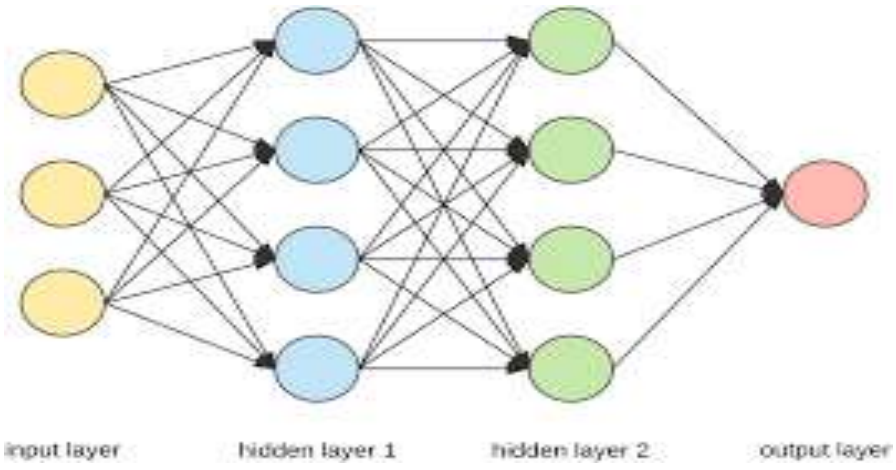
- Naive Bayes is easy and fast to predict the class of test data set. It also performswell in the multi class prediction.
- It can be used for Binary class predictions as well as multi class predictions and itperforms well for multiclass predictions as compared to other algorithms.
- This algorithm is mainly used for text classification problems.

Disadvantages of Naive Bayes

- Naive Bayes algorithm feels that every feature in the data set is independent andunrelated, so it doesn't learn relationships between the features.
- Naive Bayes is also known as a bad estimator because it doesn't take theprobability of the predictors too seriously.

3.2.4 Multi-Layer Perceptron

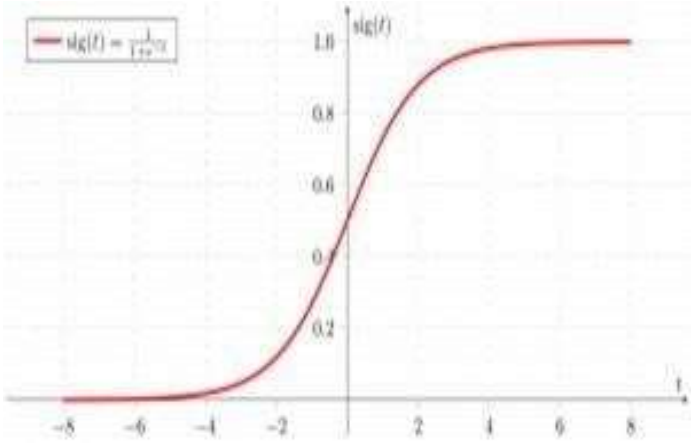
A Multi-Layer Perceptron (MLP) or Multi-Layer Neural Network contains one or additional hidden layers (apart from one input and one output layer). whereas one layer perceptron will solely learn linear functions, a multi-layer perceptron also can learn non – linear functions.



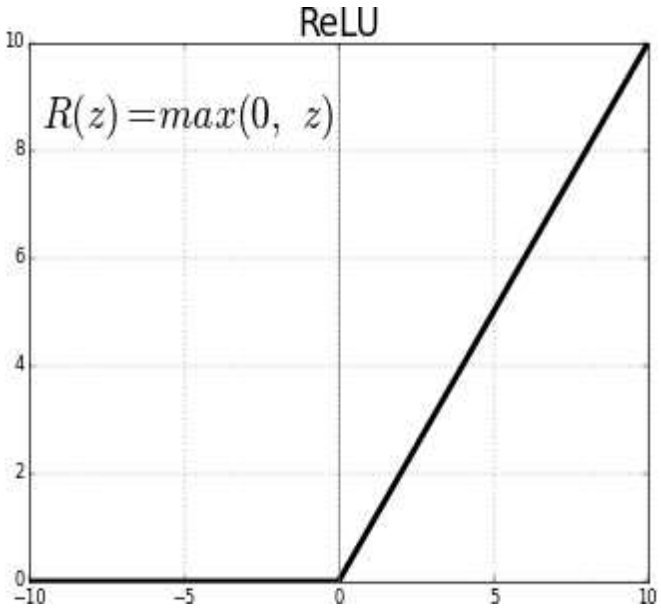
This nerve cell takes as input x1,x2,…., xn (and a +1 bias term), and outputs f(summed inputs +bias), where f(.) known as the activation perform. the most perform of Bias is to supply each node with a trainable constant price (in addition to the conventional inputs that the node receives). each activation perform (or non-

linearity) takes one range and performs a particular fastened computing thereon. There is area unit many activation functions you will encounter in practice:

Sigmoid: takes real-valued input and squashes it to vary between zero and one.



ReLu: ReLu stands for corrected Linear Units. It takes real-valued input and thresholds it to zero (replaces negative values to zero).



Dataset and Setup

The information we gathered from genuine expectation esteems i.e wearable gadgets, forecast and clinical trial results. The Data records are regarding medical aspects of Cancer, Coronary Heart, Cancer, Mentality. The Data records are with respect to clinical issues of Cancer, Coronary Heart, Cancer, Mentality patients.

- **Diabetes Dataset**

This dataset is initially from the National Institute of Diabetes and Digestive and Kidney Diseases. The target of the dataset is to indicatively anticipate whether a patient has diabetes, because of certain demonstrative estimations remembered for the dataset. A few imperatives were put on the

choice of these occurrences from a bigger data set. Specifically, all patients here are females at any rate 21 years of age of Pima Indian legacy. The datasets comprise of a few clinical indicators (autonomous) factors and one objective (subordinate) variable, Outcome. Autonomous factors incorporate the quantity of pregnancies the patient has had, their BMI, insulin level, age, etc.

- **Coronary heart disease**

The dataset has been gathered from four nations, which are functional outcomes in which 76 features of which 14 of them are prior to characterizing the objective boundary of the coronary illness of the patient.

- **Cancer Patients Dataset**

Numerous individuals' lives are sliced short because of malignant growth. Notwithstanding, because of the time of large information, we can battle this pernicious sickness i.e. Cancer. The dataset is the way of life style of different malignancy patients.

- **Mentality Dataset**

In the cutting-edge way of life, responding to a circumstance can characterize the disposition and attitude of an individual. The information records are about the activities of individuals in their way of life that characterized the attitude of the individual is ordinary or not.

4. BLOCK DIAGRAM

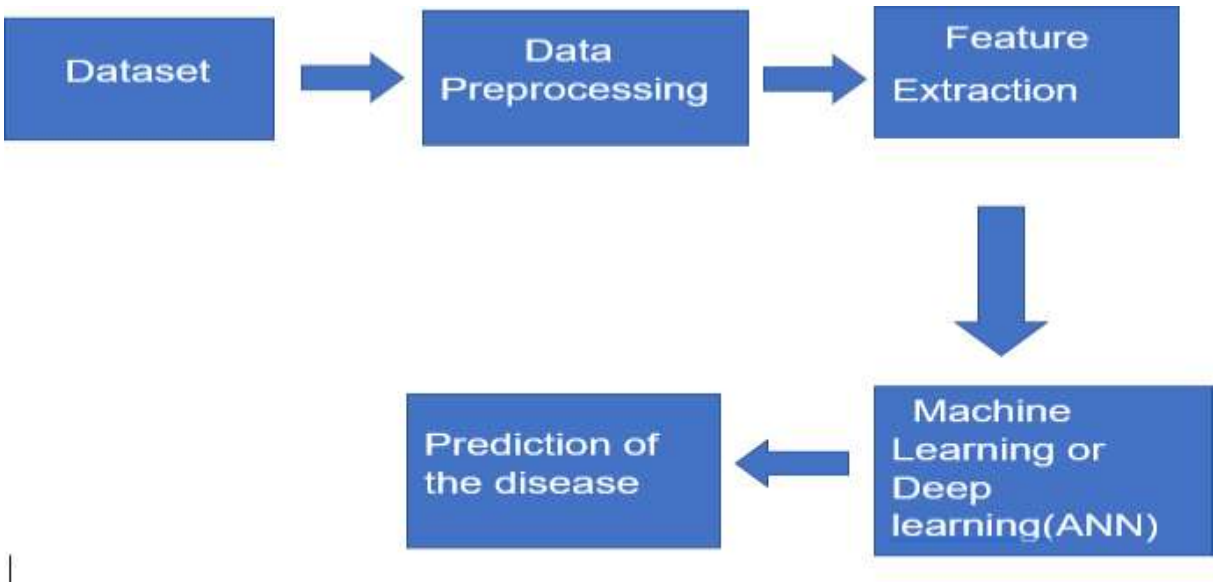


Fig.1.Block Diagram of Disease Prediction

5. IMPLEMENTATION

Disease classification using Artificial Intelligence Techniques

```
from google.colab import drive
drive.mount("/content/gdrive")
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
import joblib
data_heart=pd.read_csv("/content/gdrive/My Drive/heart.csv")


data_heart.shape


data_heart.head()


f=list(data_heart['age'].unique())


f.sort()
print(f[0],f[-1])
Xheart=data_heart.iloc[:, :-1].values
Yheart=data_heart.iloc[:,-1].values
import scipy
z_scores = scipy.stats.zscore(Xheart)

abs_scores = np.abs(z_scores)
fe = (abs_scores < 3).all(axis=1)
```

```
Xheart_z = Xheart[fe]

Yheart_z=Yheart[fe]


Xheart.shape

Xheart_z.shape


Yheart_z.shape

from collections import Counter

c=Counter(Yheart_z)

print(c)

from sklearn.decomposition import PCA

pca_pw=PCA(n_components=13)

pca_dpw=pca_pw.fit_transform(Xheart_z)

pve= pca_pw.explained_variance_ /np.sum(pca_pw.explained_variance_)

c_v_e=np.cumsum(pve)

plt.figure(1,figsize=(6,4))

plt.clf()

plt.plot(c_v_e,linewidth=2)

plt.axis('tight')

plt.grid()

plt.xlabel('n_components')

plt.ylabel('Cummulative_explained_varience')

plt.show()




from sklearn.model_selection import train_test_split

Xtrain_hh,Xtest_hh,Ytrain_hh,Ytest_hh=train_test_split(Xheart_z,Yheart_z,test_size=0.2)


from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()
Xtrain_hh[:,[0,3,4,7,9]]=sc.fit_transform(Xtrain_hh[:,[0,3,4,7,9]])
Xtest_hh[:,[0,3,4,7,9]]=sc.transform(Xtest_hh[:,[0,3,4,7,9]])
from sklearn.decomposition import PCA
pca_actual=PCA(n_components=4)
Xtrain_pca=pca_actual.fit_transform(Xtrain_hh)
Xtest_pca=pca_actual.transform(Xtest_hh)



from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(Xtrain_pca, Ytrain_hh)
y_pred_h = classifier.predict(Xtest_pca)

y_pred_h
from sklearn.metrics import accuracy_score
accuracy_score(Ytest_hh,y_pred_h)
y_pred_h_train = classifier.predict(Xtrain_pca)
from sklearn.metrics import accuracy_score
accuracy_score(Ytrain_hh,y_pred_h_train)
from sklearn.ensemble import RandomForestClassifier
cr_rm = RandomForestClassifier(n_estimators = 50, criterion = 'entropy', random_state = 0)
cr_rm.fit(Xtrain_pca, Ytrain_hh)
ypred_h_raf=cr_rm.predict(Xtest_pca)

from sklearn.metrics import  accuracy_score
```

```
accuracy_score(Ytest_hh,ypred_h_raf)

ypred_h_raf_t=cr_rm.predict(Xtrain_pca)


from sklearn.metrics import accuracy_score

accuracy_score(Ytrain_hh,ypred_h_raf_t)


from sklearn.svm import SVC

csvm = SVC(kernel = 'linear', random_state = 0)

csvm.fit(Xtrain_pca, Ytrain_hh)

y_pred_svm = classifier.predict(Xtest_pca)


from sklearn.metrics import accuracy_score

accuracy_score(Ytest_hh,y_pred_svm)


from sklearn.neighbors import KNeighborsClassifier

cl2 = KNeighborsClassifier(n_neighbors = 3, metric = 'minkowski', p = 2)

cl2.fit(Xtrain_pca, Ytrain_hh)

y_pred_knn_ = cl2.predict(Xtest_pca)

from sklearn.metrics import accuracy_score

accuracy_score(Ytest_hh,y_pred_knn_)

data_dbs=pd.read_csv("/content/gdrive/My Drive/diabetes.csv")

data_dbs.head()


data_dbs.isnull().sum()

for i in data_dbs.columns:

print(data_dbs[i].unique)

Xdbs=data_dbs.iloc[:,0:-1].values
```

```
Ydbs=data_dbs.iloc[:,-1].values
Ydbs
 import scipy
z_scores_db = scipy.stats.zscore(Xdbs)
abs_scores_db = np.abs(z_scores_db)
fe1 = (abs_scores_db < 3).all(axis=1)
Xheart_z_db = Xdbs[fe1]
Yheart_z_db=Ydbs[fe1]
Xheart_z_db.shape
from sklearn.decomposition import PCA
pca_pw_db=PCA(n_components=8)
pca_dpw=pca_pw_db.fit_transform(Xheart_z_db)
pve_db= pca_pw_db.explained_variance_ /np.sum(pca_pw_db.explained_variance_)
c_v_e_db=np.cumsum(pve_db)
plt.figure(1,figsize=(6,4))
plt.clf()
plt.plot(c_v_e_db,linewidth=2)
plt.axis('tight')
plt.grid()
plt.xlabel('n_components')
plt.ylabel('Cummulative_explained_varience')
plt.show()
Xheart_z_db
from sklearn.model_selection import train_test_split
Xtrain_db,Xtest_db,Ytrain_db,Ytest_db=train_test_split(Xheart_z_db,Yheart_z_db,test_size=0.2)



from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()
Xtrain_db[:,:]=sc.fit_transform(Xtrain_db[:,:])
Xtest_db[:,:]=sc.transform(Xtest_db[:,:])
from sklearn.decomposition import PCA
pca_actual_db=PCA(n_components=5)
Xtrain_db=pca_actual_db.fit_transform(Xtrain_db)
Xtest_db=pca_actual_db.transform(Xtest_db)

from sklearn.naive_bayes import GaussianNB
classifier_db = GaussianNB()
classifier_db.fit(Xtrain_db, Ytrain_db)

y_pred_db = classifier_db.predict(Xtest_db)
from sklearn.metrics import accuracy_score
accuracy_score(Ytest_db,y_pred_db)
from sklearn.ensemble import RandomForestClassifier
cr_rm_db = RandomForestClassifier(n_estimators = 90, criterion = 'entropy', random_state = 0)
cr_rm_db.fit(Xtrain_db, Ytrain_db)
y_pred_db_rm = cr_rm_db.predict(Xtest_db)
from sklearn.metrics import accuracy_score
accuracy_score(Ytest_db,y_pred_db_rm)
from sklearn.svm import SVC
csvm_db = SVC(kernel = 'linear', random_state = 10)
csvm_db.fit(Xtrain_db, Ytrain_db)
y_pred_db_svm = csvm_db.predict(Xtest_db)



from sklearn.metrics import accuracy_score
accuracy_score(Ytest_db,y_pred_db_svm)
```

```
from sklearn.neighbors import KNeighborsClassifier
cl23 = KNeighborsClassifier(n_neighbors = 10, metric = 'minkowski', p = 2)
cl23.fit(Xtrain_db, Ytrain_db)
y_pred_knn_1 = cl23.predict(Xtest_db)

from sklearn.metrics import accuracy_score
accuracy_score(Ytest_db,y_pred_knn_1)
import numpy as np
import matplotlib.pyplot as plt
data = [[97, 100, 80, 78],
[100, 100, 100, 74],
[100, 100, 80, 78],
[100,100,93,76],
[100,20,90,71]]
f, ax = plt.subplots(figsize=(18,10))
labels=["cancer","mentality","heart","diabetes"]
plt.bar([0,3,5,7], data[0],color='b',width=0.25,align='center')
plt.bar([0.25,3.25,5.25,7.25], data[1],color='r',width=0.25, align='center')
plt.bar([0.5,3.5,5.50,7.50], data[2],color='g',width=0.25,align='center')
plt.bar([0.75,3.75,5.75,7.75],data[3],color='y',width=0.25, align='center')
plt.bar([1.0,4.0,6.0,8],data[4],color='black',width=0.25,align='center')



plt.xticks([0.5,3.5,5.50,7.50],labels,fontsize=15)
plt.xlabel('Diseases',fontsize=12)
plt.ylabel('Accuracy(%)',fontsize=12)
plt.legend(["NaveBayes","Random Forest",'SVM',"KNN","ANN"])
plt.savefig('books_read.png')
```

```
import tensorflow as tf
ann = tf.keras.models.Sequential()
ann.add(tf.keras.layers.Dense(units=4, activation='relu'))
ann.add(tf.keras.layers.Dense(units=16, activation='relu'))

ann.add(tf.keras.layers.Dense(units=60, activation='relu'))
ann.add(tf.keras.layers.Dense(units=6, activation='relu'))

ann.add(tf.keras.layers.Dense(units=1,  activation='sigmoid'))
ann.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
ann.fit(Xtrain_pca, Ytrain_hh, batch_size = 32, epochs = 100)

d=[]
y_pred_ann_Heart = ann.predict(Xtest_pca)
for i in y_pred_ann_Heart:
  if i>=0.5:
    d.append([1])
  else:
    d.append([0])

d



from sklearn.metrics import accuracy_score
accuracy_score(Ytest_hh,d)
ann_db= tf.keras.models.Sequential()
ann_db.add(tf.keras.layers.Dense(units=5, activation='relu'))
```

```
ann_db.add(tf.keras.layers.Dense(units=25, activation='relu'))

ann_db.add(tf.keras.layers.Dense(units=50, activation='relu'))

ann_db.add(tf.keras.layers.Dense(units=100, activation='relu'))
ann_db.add(tf.keras.layers.Dense(units=10, activation='relu'))



ann_db.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
ann_db.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

ann_db.fit(Xtrain_db, Ytrain_db, batch_size = 40, epochs = 300)
d1=[]
y_pred_ann_Heart = ann_db.predict(Xtest_db)
for i in y_pred_ann_Heart:
  if i>=0.5:
    d1.append([1])
  else:
    d1.append([0])
 from sklearn.metrics import accuracy_score
accuracy_score(Ytest_db,d1)
  from google.colab import drive



drive.mount("/content/gdrive")
import pandas as pd
import numpy as np
```

```
import seaborn as sb
import matplotlib.pyplot as plt
import joblib

data_cancer=pd.read_excel('/content/gdrive/My Drive/cancer patient data sets.xlsx')
data_cancer
data_cancer.columns
data_cancer.isnull().sum()
data_cancer['Level'].unique()
Xcan=data_cancer.iloc[:, 1:24].values
Ycan=data_cancer.iloc[:,-1].values



from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
Ycan_lb=lb.fit_transform(Ycan)

import scipy
z_scores_db = scipy.stats.zscore(Xcan)
abs_scores_db = np.abs(z_scores_db)
fe1 = (abs_scores_db < 3).all(axis=1)
Xcan = Xcan[fe1]
Ycan=Ycan[fe1]
from sklearn.decomposition import PCA
pca_pw_db=PCA(n_components=23)
```

```
pca_dpw=pca_pw_db.fit_transform(Xcan)
pve_db= pca_pw_db.explained_variance_ /np.sum(pca_pw_db.explained_variance_)
c_v_e_db=np.cumsum(pve_db)
plt.figure(1,figsize=(6,4))
plt.clf()
plt.plot(c_v_e_db,linewidth=2)
plt.axis('tight')
plt.grid()
plt.xlabel('n_components')
plt.ylabel('Cummulative_explained_varience')
plt.show()
# Ycan_ohe = np.reshape(Ycan, (-1, 1))
Ycan=np.reshape(Ycan, (-1, 1))
from sklearn.preprocessing import OneHotEncoder
o=OneHotEncoder()
Y_ohe=o.fit_transform(Ycan).toarray()
Y_ohe

Ycan_lb=np.reshape(Ycan_lb, (-1, 1))
from sklearn.model_selection import train_test_split
Xtrain_db,Xtest_db,Ytrain_db,Ytest_db=train_test_split(Xcan,Ycan_lb,test_size=0.2)

Ytrain_db
from sklearn.decomposition import PCA
pca_actual_db=PCA(n_components=18)



Xtrain_db=pca_actual_db.fit_transform(Xtrain_db)
Xtest_db=pca_actual_db.transform(Xtest_db)
```

```
from sklearn.naive_bayes import GaussianNB
classifier_db = GaussianNB()
classifier_db.fit(Xtrain_db, Ytrain_db)

y_pred_db = classifier_db.predict(Xtest_db)


from sklearn.metrics import accuracy_score
accuracy_score(Ytest_db,y_pred_db)

y_pred_db_train = classifier_db.predict(Xtrain_db)

from sklearn.metrics import accuracy_score
accuracy_score(Ytrain_db,y_pred_db_train)

from sklearn.ensemble import RandomForestClassifier
cr_rm_db = RandomForestClassifier(n_estimators = 100, criterion = 'entropy', random_state = 10)
cr_rm_db.fit(Xtrain_db, Ytrain_db)
y_pred_db_rm = cr_rm_db.predict(Xtest_db)

y_pred_db_rm_train = cr_rm_db.predict(Xtrain_db)
from sklearn.metrics import accuracy_score
accuracy_score(Ytrain_db,y_pred_db_rm_train)
from sklearn.metrics import accuracy_score



accuracy_score(Ytest_db,y_pred_db_rm)
from sklearn.svm import SVC
csvm_db = SVC (kernel = 'linear', random_state = 10)
```

```
csvm_db.fit(Xtrain_db, Ytrain_db)


y_pred_db_svm = csvm_db.predict(Xtest_db)
y_pred_db_svm_train = csvm_db.predict(Xtrain_db)
from sklearn.metrics import accuracy_score
accuracy_score(Ytest_db,y_pred_db_svm)
from sklearn.metrics import accuracy_score
accuracy_score(Ytrain_db,y_pred_db_svm_train)
from sklearn.neighbors import KNeighborsClassifier
cl = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
cl.fit(Xtrain_db, Ytrain_db)
y_pred_knn = cl.predict(Xtest_db)
from sklearn.metrics import accuracy_score
accuracy_score(Ytest_db,y_pred_knn)


data_mh=pd.read_csv("/content/gdrive/My Drive/mental_health.csv")
data_mh.head()

data_mh=data_mh.astype('category')
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
for i in range(len(data_mh.columns)-1):
  data_mh[data_mh.columns[i]]=lb.fit_transform(data_mh[data_mh.columns[i]].astype(str))



data_mh.columns

data_mh.shape
```

```
data_mh.var()

Xmh=data_mh.iloc[:, :-1].values
Ymh=data_mh.iloc[:,-1].values
import scipy
z_scores = scipy.stats.zscore(Xmh)

abs_scores = np.abs(z_scores)
fe = (abs_scores < 3).all(axis=1)
Xmh_z = Xmh[fe]
Ymh_z=Ymh[fe]
Xmh_z.shape
from sklearn.decomposition import PCA
pca_pw_db=PCA(n_components=24)
pca_dpw=pca_pw_db.fit_transform(Xmh_z)
pve_db= pca_pw_db.explained_variance_ /np.sum(pca_pw_db.explained_variance_)
c_v_e_db=np.cumsum(pve_db)
plt.figure(1,figsize=(6,4))
plt.clf()
plt.plot(c_v_e_db,linewidth=2)
plt.axis('tight')
plt.grid()
plt.xlabel('n_components')
plt.ylabel('Cummulative_explained_varience')



plt.show()
from sklearn.preprocessing import LabelEncoder
```

```
lb=LabelEncoder()
Ymh_z=lb.fit_transform(Ymh_z)


from sklearn.model_selection import train_test_split
Xtrain_hh,Xtest_hh,Ytrain_hh,Ytest_hh=train_test_split(Xmh_z,Ymh_z,test_size=0.2)
from sklearn.decomposition import PCA
pca_actual=PCA(n_components=4)
Xtrain_hh_pca=pca_actual.fit_transform(Xtrain_hh)
Xtest_hh_pca=pca_actual.transform(Xtest_hh)
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(Xtrain_hh_pca, Ytrain_hh)

y_pred_h = classifier.predict(Xtest_hh_pca)
from sklearn.metrics import accuracy_score
accuracy_score(Ytest_hh,y_pred_h)
from sklearn.ensemble import RandomForestClassifier
cr_rm = RandomForestClassifier(n_estimators = 1, criterion = 'entropy', random_state = 0)
cr_rm.fit(Xtrain_hh_pca, Ytrain_hh)
ypred_h_raf=cr_rm.predict(Xtest_hh_pca)

from sklearn.metrics import accuracy_score
accuracy_score(Ytest_hh,ypred_h_raf)
from sklearn.svm import SVC
csvm = SVC(kernel = 'linear', random_state = 0)
```

```
csvm.fit(Xtrain_hh_pca, Ytrain_hh)

y_pred_svm = classifier.predict(Xtest_hh_pca)

from sklearn.metrics import accuracy_score

accuracy_score(Ytest_hh,y_pred_svm)

from sklearn.neighbors import KNeighborsClassifier

cl2 = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)

cl2.fit(Xtrain_hh_pca, Ytrain_hh)

y_pred_knn_ = cl2.predict(Xtest_hh_pca)

from sklearn.metrics import accuracy_score

accuracy_score(Ytest_hh,y_pred_knn_)


from sklearn.model_selection import train_test_split

Xtrain_db2,Xtest_db2,Ytrain_db_ohe,Ytest_db_ohe=train_test_split(Xcan,Y_ohe,test_size=0.2)



import tensorflow as tf


ann_cr= tf.keras.models.Sequential()

ann_cr.add(tf.keras.layers.Dense(units=18, activation='relu'))

ann_cr.add(tf.keras.layers.Dense(units=100, activation='relu'))

ann_cr.add(tf.keras.layers.Dense(units=180, activation='relu'))

ann_cr.add(tf.keras.layers.Dense(units=90, activation='relu'))

ann_cr.add(tf.keras.layers.Dense(units=50, activation='relu'))

ann_cr.add(tf.keras.layers.Dense(units=25, activation='relu'))

ann_cr.add(tf.keras.layers.Dense(units=13, activation='relu'))









ann_cr.add(tf.keras.layers.Dense(units=3, activation='softmax'))
```

```
ann_cr.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])


ann_cr.fit(Xtrain_db2, Ytrain_db_ohe, batch_size = 40, epochs = 300)
d1=[]
y_pred_ann_cancer = ann_cr.predict(Xtest_db2)

y_pred_ann_cancer
y_pred_ann_=o.inverse_transform(y_pred_ann_cancer)
y_test_ann_=o.inverse_transform(Ytest_db_ohe)



from sklearn.metrics import accuracy_score
accuracy_score(y_test_ann_,y_pred_ann_)



import tensorflow as tf

ann_my= tf.keras.models.Sequential()


ann_my.add(tf.keras.layers.Dense(units=4, activation='relu'))
ann_my.add(tf.keras.layers.Dense(units=16, activation='relu'))
ann_my.add(tf.keras.layers.Dense(units=25, activation='relu'))
ann_my.add(tf.keras.layers.Dense(units=50, activation='relu'))

ann_my.add(tf.keras.layers.Dense(units=100, activation='relu'))
ann_my.add(tf.keras.layers.Dense(units=200, activation='relu'))


ann_my.add(tf.keras.layers.Dense(units=75, activation='relu'))
ann_my.add(tf.keras.layers.Dense(units=15, activation='relu'))
```

```
ann_my.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))

ann_my.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

ann_my.fit(Xtrain_hh_pca, Ytrain_hh, batch_size = 40, epochs = 300)

d1k=[]

y_pred_ann_mentailty = ann_my.predict(Xtest_hh_pca)



y_pred_ann_mentailty

from sklearn.metrics import accuracy_score

accuracy_score(Ytest_hh,y_pred_ann_mentailty)
```

6. RESULTS AND ANALYSIS

The Datasets are set up with 80% preparing information while 20% of approval information i.e., for testing of the model. Various models are tried by utilizing the overall execution and announced the exactness of the models on different datasets through the bar outline in the underneath graph. In genral,Random Forest has best suited the data records i.e., Cancer,Mentality,heart and Diabetes records, whereas the run time for all the algorithms are similar. The execution time general, Random Forest has most appropriate the information records i.e., Cancer, Mentality, heart, and Diabetes records, of the models is a lot comparable. When all is said in done, Random Forest has most appropriate the information records i.e., Cancer, Mentality, heart and Diabetes records. The resulting information of the model is the person affected by a particular disease or not. As categorical one has yes or no for the datasets excluding cancer Dataset.where as in cancer dataset it was multi-categorical data.

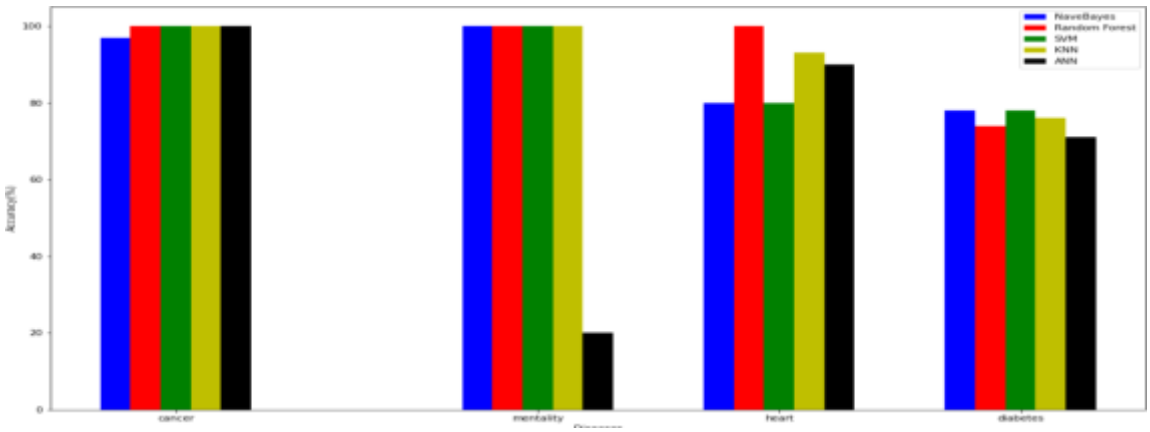


Fig2. Accuracies of Different models on different datasets

During the training phase, the model is trained using the IMDB Dataset. You will get different accuracy rates based on the number of iterations that you perform on the model. During every iteration, the model will compare the specified Review. Table.1

Algorithm	Cancer	Mentality	Heart	Diabetes
Nave Bayes	**97%**	**100%**	**80%**	**78%**
Random Forest	**100%**	**100%**	**100%**	**74%**
SVM	**100%**	**100%**	**80%**	**78%**
KNN	**100%**	**100%**	**93%**	**76%**
ANN	**100%**	**20%**	**90%**	**71%**

Table 1: Accuracy metrics.

7.CONCLUSION AND FUTURE SCOPE

The study of Artificial Intelligence, which contains ML and Deep learning supervised algorithms are subjected to classify the diseases of the patient whereas random forest algorithm as prior which was giving more accuracy to the respective datasets. As remaining algorithms i.e., SVM, ANN, Naive Bayes, and K-NN algorithm results are acceptable but predominantly the Random Forest algorithm is giving prior results for all the data records.

There are many ways for future enhancements. As the data records were taken for the building model are restricted to medical test results, but the model can't handle the patient inputs as a prior one and the model is not user-friendly without the medical test results. That data can lively as updated data can make models based on time series learning and. usage of hybrid algorithms i.e ensemble learning can make models more.

8. ACKNOLEDGMENT

I would like to express my special thanks of gratitude to my Guide M.V.B.T Shanthi .as well as our HOD Hari Kiran Vege who gave me the golden opportunity to do this wonderful project.

9. REFERENCES

1)Ren F, Hu L, Liang H, Liu X, Ren W (2008) Using density-based incremental clustering for anomaly detection. In: International Conference on Computer and Software Engineering, IEEE, pp 986–989.

2) Dahiwade D, Patle G, Meshram E (2019) Designing disease prediction model using machine learning approach. In: Third IEEE International Conference on Computing Methodologies and Communication (ICCMC), 2019

3) Wu H, Yang S, Huang Z, He J, Wang X (2018) Type 2 diabetes mellitus prediction model based on data mining. Inform Med Unlocked 10:100–107.

4) Babu S, Vivek EM, Famina KP, Fida K, AswathiP, Shanid M, Hena M (2017) Heart disease diagnosis using data mining technique. In: International Conference on Electronics, Communication, and Aerospace Technology, ICECA2017

5) Sampaul TGA, Robinson YH, Julie EG, Shanmuganathan V, Nam Y, Rho S (2020) Diabetic retinopathy diagnostics from retinal images based on deep convolutional networks. Preprints. https://doi. Org/10.20944/preprints202005. 0493.v1.

6) Vimal S et al (2020) Deep learning-based decision-making with WoT for smart city development. In: Jain A, Crespo R, Khari M (eds) Smart inno.

7) Long NC, Meesad P, Unger H (2015) A highly accurate frefy-based algorithm for heart disease prediction. Expert Syst Appl 42:8221–8231.

8) Sarwar A, Sharma V (2012) Intelligent Naïve Bayes approach to diagnose diabetes type-2. In: Special Issue of International Journal of Computer Applications on Issues and Challenges in Networking, Intelligence and Computing Technologies, November 2012

9) Esteghamati A, Hafezi-Nejad N, Zandieh A, Sheikhbahaei S, Ebadi M, Nakhjavani M (2014) Homocysteine and metabolic syndrome: from clustering to additional utility in prediction of coronary heart disease. J Cardiol 64:290–296

10) Lee BJ, Kim JY (2016) Identifications of type 2 diabetes risk factors using phenotypes consisting of anthropometry and triglycerides based on machine learning. IEEE J Biomed Health Inform 20(1):39–46.

11) Nai-arna N, Maungmagan R (2015) Comparison of classifiers for the risk of diabetes prediction. In: 7th International Conference on Advances in Information Technology Procedia Computer Science, vol 69, pp 132 –142

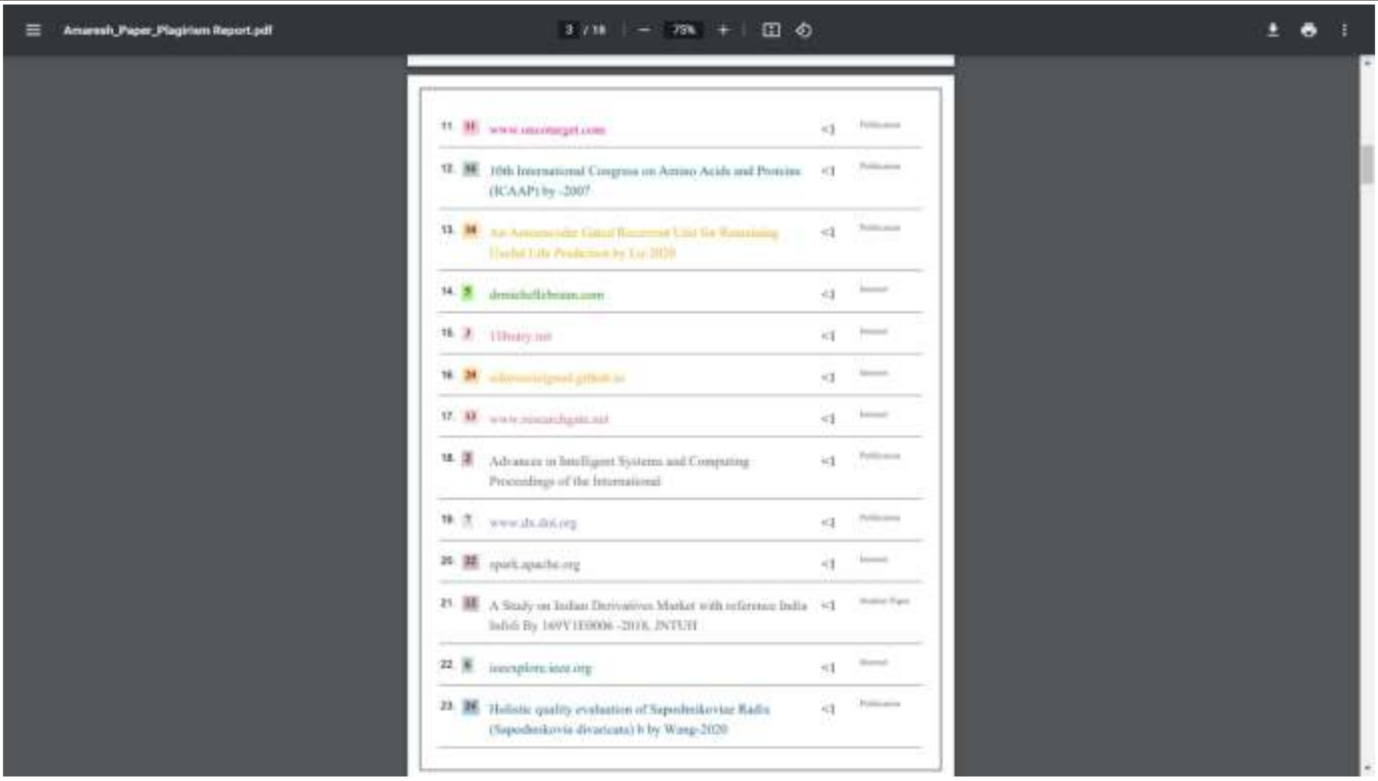
12) Zhang Z, Shen H (2005) Application of online-training SVMs for real time intrusion detection with different considerations. Compute Commun 28(12):1428–1442

13) Ramamurthy M, Krishnamurthi I, Vimal S, Harold Y (2020) Robinson deep learning-based genome analysis and NGS-RNA LL identification with a novel hybrid model.

14) Pradeepa S, Gayathri P, Nishmitha P, Vimal S, Oh-Young S, Usman T, Raheel N (2020) IoT based health-related topic recognition from emerging online health community: med help using machine learning technique. Electronics 9(9):1469

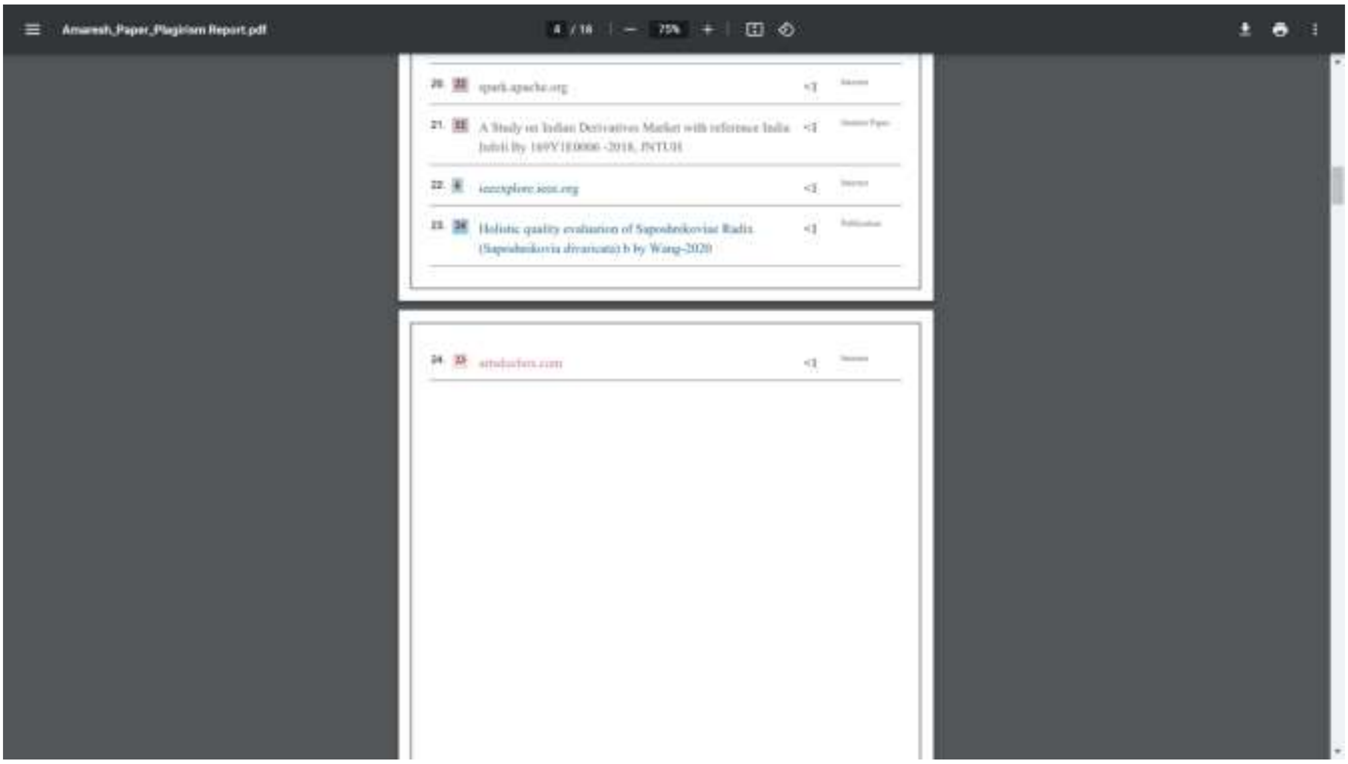
15) Breiman, L., Random Forests, Machine Learning 45(1), 5-32, 2001.

10. Plagiarism Report



Amaresh_Paper_Plagiarism Report.pdf 3 / 18 75%

11.	31	www.oncotarget.com	<1
12.	16	10th International Congress on Amino Acids and Proteins (ICAAP) by -2007	<1
13.	34	An Autoencoder Gated Recurrent Unit for Remaining Useful Life Prediction by Lu 2020	<1
14.	5	drmichellebeam.com	<1
15.	3	1library.net	<1
16.	24	[illegible]	<1
17.	13	www.researchgate.net	<1
18.	2	Advances in Intelligent Systems and Computing Proceedings of the International	<1
19.	7	www.dx.doi.org	<1
20.	22	spark.apache.org	<1
21.	12	A Study on Indian Derivatives Market with reference India Infoli By 16VY1E0006 -2018, JNTUH	<1
22.	6	ieeexplore.ieee.org	<1
23.	26	Holistic quality evaluation of Saposhnikoviae Radix (Saposhnikovia divaricata) b by Wang-2020	<1



Amaresh_Paper_Plagiarism Report.pdf 4 / 18 75%

20.	22	spark.apache.org	<1
21.	12	A Study on Indian Derivatives Market with reference India Infoli By 16VY1E0006 -2018, JNTUH	<1
22.	6	ieeexplore.ieee.org	<1
23.	26	Holistic quality evaluation of Saposhnikoviae Radix (Saposhnikovia divaricata) b by Wang-2020	<1
24.	23	amdocfeirs.com	<1