Comparing Results of Classification Techniques Regarding Heart Disease Diagnosing

Benan Abdullah AL badr, Raghad Suliman AL ghezzi, ALjohara Suliman AL moqhem, Dr.Sarah Eljack.
381200736@s.mu.edu.sa, 381200973@s.mu.edu.sa, 381200760@s.mu.edu.sa, s.alshiekh@mu.edu.sa
College of Science, Department of Computer Science and information, Majmaah University 11952, Az Zulfi, KSA

Abstract
Despite global medical advancements, many patients are misdiagnosed, and more people are dying as a result. We must now develop techniques that provide the most accurate diagnosis of heart disease based on recorded data. To help immediate and accurate diagnose of heart disease, several data mining methods are accustomed to anticipating the disease. A large amount of clinical information offered data mining strategies to uncover the hidden pattern. This paper presents, comparison between different classification techniques, we applied on the same dataset to see what is the best. In the end, we found that the Random Forest algorithm had the best results.

Keywords:
Google Colab, classification technique, Random Forest, Python language, machine learning.

1. Introduction

The heart is the foremost imperative component of the human body because it is capable of pumping oxygen-rich blood through a network of arteries and veins to other body parts. [1-2] as stated in the world health organization report people die every year worldwide due to heart illness. [3-4] Predicting and diagnosing heart disease is the biggest challenge in the medical industry [5-6]. Coronary artery disease, congenital heart disease, arrhythmia, and other forms of heart illness exist. Heart disease manifests itself in a variety of ways, including chest discomfort, dizziness, and excessive sweating. [7] Nowadays, the healthcare industry has a significant amount of healthcare data, much of which is secret.[8] Conventional diagnoses and tests like ECG, MRI scans are used for checking whether the heart is healthy or not. But these tests are expensive as heavy imported machinery, skilled labor, etc. are involved. The main aim of the healthcare policy of any country is to provide cheap but effective healthcare to every citizen.[9] Nowadays, Data mining techniques like classification are playing an imperative part within the biomedical field as they are used to explore, analyze and extract medical data using complex algorithms to find unknown patterns. [10-11-12]

In medical care, there is a lot of information, but sometimes this information may not be accurate or the symptoms of the disease may be similar, which causes an error in diagnosis and this may lead to unsatisfactory results. So, the objective in this paper to predict possible heart disease using classification techniques, one of the most common forms of medical costs worldwide are diagnostic error so the correct diagnosis of the disease will help the doctor saving more lives, to helps avoid human biasness, to provide effective treatments to patients, and rapid diagnosis of the patient's condition. Heart disease refers to the group of diseases that affect the human heart.
heart disease includes:
Cardiomyopathy, heart valve disease, coronary artery disease, arrhythmias, heart failure. [13]
Symptoms of heart disease:
Shortness of breath, pain in the upper abdomen, throat, neck, back or jaw, chest discomfort, chest pressure, Chest pain and chest tightness, and weakness, coldness or numbness in your legs or arms. [14]
Heart diseases risk can increase by health condition and standard of living. These include:
Smoking, high stress and anxiety levels, dietary choices, diabetes, high cholesterol, a history of preeclampsia during pregnancy, age, low activity levels, sleep apnea, overweight and obesity, a family history of heart disease, leaky heart valves, a high intake of alcohol, and high blood pressure. [15]

2. Research Methodology

Machine learning

Machine learning procedures have been effectively applied in a wide variety of fields. Ionizing radiation is used to treat many cancer patients. Radiation therapy involves a complex set of processes that go beyond counseling and treatment to ensure patients receive the exact dose and respond well. Machine learning algorithms can enhance the safety and practice of radiotherapy, leading to excellent results.[16]

Manuscript received May 5, 2022
Manuscript revised May 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.5.20

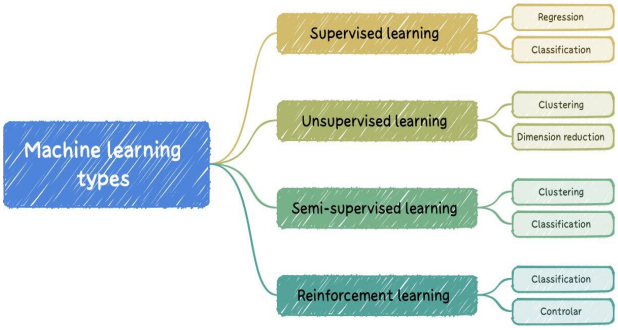

Figure 1 Machine learning types [17]

Types of machine learning:

1. Unsupervised Learning

Consider a set of data that only consists of input and look for a structure in it, such as collecting or installing data points. As a result, these algorithms learn from test data that has not been labeled or classified. Unsupervised learning algorithms identify commonalities in data and interact based on the presence or absence of such commonalities in each new segment of data.[18]

2. Reinforcement learning

Guiding machine learning models to adopt a decision. The agent of reinforcement learning figures out how to accomplish a goal in an unclear environment. [19]

3. Semi-supervised learning

Semi-supervised learning is a model that investigates how computers like humans learn in the presence of labeled and unlabeled data. Learning has also been studied in unsupervised or supervised models, where all data is labeled. The goal of semi-supervised learning is to understand how the combination of data affects learning behavior and to design algorithms that take advantage of this combination. [20]

4. Supervised Learning

Supervised learning trains a data sample that contains the correct classification from the data source. These methods are used in models, including the MLP model, which is characterized by one or more layers of covered-up neurons that are not a portion of the network's input, the nonlinearity reflected in neuronal activity can be distinguished and the network's interconnection model contains a tall degree of connectivity. These features will help to solve difficult problems. [19]

Classification Techniques

Classification is the process of identifying things and ideas and then understanding and categorizing them. In machine learning programs, classification is used to classify a set of future data into related categories with the help of training data sets that were previously classified. The most common use of the classification is to classify an email as spam or non-spam, classification helps to identify patterns.[21]

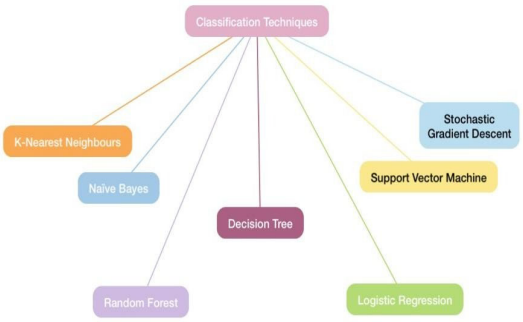

Figure 2 Classification Techniques

1: Logistic Regression

Logistic regression is a machine learning classification algorithm. A logistic function is used in this algorithm to the model probability that describes the possible outcomes of a single experiment.

Advantages: Designed for this purpose (classification), its usefulness lies in understanding the effect of independent variables on what is not a single outcome.

Disadvantages: Its modus operandi are limited. It only works if the variable is binary, all predictors must be independent of each other and all data must be complete. [22]

Logistic regression formula: [23]

(1)

$$Y = \frac{e^{b0+b1*x}}{(1+e^{b0+b1*x})}$$

Where:

- x is the input value
- y is the predicted output
- b0 is the bias or intercept term
- b1 is the coefficient for the single input value (x)

2: Naïve Bayes

Is based on a theory called Bayes, which assumes the independence between each pair of features. It is very good at classifying documents and spam

Advantages: One of the advantages is that it is very fast in classifying compared to other algorithms

Naïve bayes formula: [24]

(2)

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

Where:

- P(A|B) = Posterior Probability, Probability of A given Value of B.
- P (B|A) = Likelihood of B given A is True.
- P (A) = Prior Probability, Probability of event A.
- P (B) = Marginal Probability, Probability of event B.

3: Stochastic Gradient Descent

Is a straightforward and highly efficient method for fitting linear models. It is especially useful when the number of samples is large. It supports various loss functions and penalties for classification .

Advantages: Efficiency and ease of implementation .

Disadvantages: Requires a number of hyper-parameters and is sensitive to feature scaling.

Stochastic Gradient Descent formula: [25]

(3)

$$\theta_{n+1} = \theta_n - \alpha \frac{\partial}{\partial \theta_n} J(\theta_n)$$

Where:

- θ = Parameter vector
- J = Cost Function
- α = Slope Parameter

4: Nearest Neighbors

Neighbors-based classification is a type of lazy learning in that it does not attempt to build a general internal model but instead simply stores instances of the training data. The classification is determined by a simple majority vote of each point's k nearest neighbors .

Advantages: This algorithm is simple to implement, robust to noisy training data, and effective when training data is large .

Disadvantages: It is necessary to determine the value of K, and the computation cost is high due to the need to compute the distance of each instance to all of the training samples.

Nearest neighbors' formula: [26]

(4)

$$\sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Where:

- n = no of dimensions (2 for our data)
- q = datapoint from dataset
- p = new data point(to be predicted)

5: Decision Tree

Given a set of attributes and their classes, a decision tree generates a set of rules that can be used to classify the data .

Advantages: is simple to understand and visualize, requires little data preparation, and can handle both numerical and categorical data .

Disadvantages: Decision trees can generate complex trees that do not generalize well, and decision trees can be unstable because small variations in the data can result in a completely different tree being generated.

Decision Tree formula: [27]

(5)

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

Where:

- c is the number of classes
- pi is the probability associated with the class

6: Random Forest

A classifier is a meta-estimator that fits a number of decision trees on different sub-samples of datasets and uses average to improve predictive accuracy while controlling over-fitting. The original input sample size is always used as the sub-sample size, but the samples are drawn with replacement .

Advantages: Over-fitting is reduced, and random forest classifiers are more accurate than decision trees in most cases .

Disadvantages include slow real-time prediction, difficulty in implementation, and a complex algorithm.

Random forest simplified: [28]

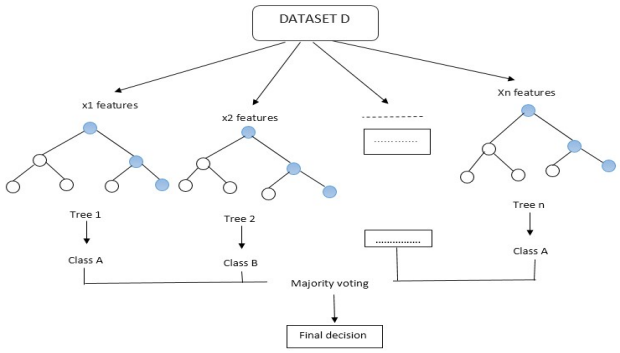

Figure 3 Random Forest

7: Support Vector Machine

Is a representation of the training data as points in space separated into categories by as wide a gap as

possible? New examples are then mapped into that same space and predicted to belong to one of the categories based on which side of the gap they fall on .

Advantages: It is effective in high-dimensional spaces and uses only a subset of training points in the decision function, making it memory efficient .

Disadvantages: The algorithm does not directly provide probability estimates; instead, these are obtained through an expensive five-fold cross-validation procedure .

Support vector machine simplified: [29]

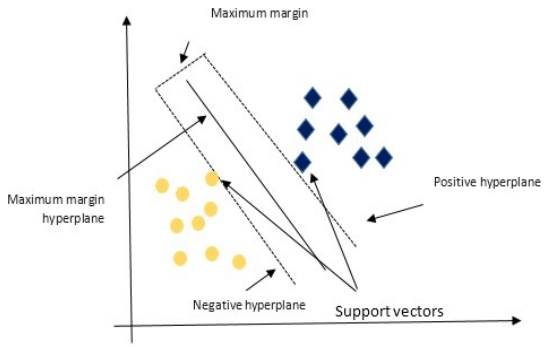

Figure 4 support vector machine

3. Proposed methodology

We are going to comparing the classification techniques, using the Colab program using Python language. And the dataset we will use is which include information about people regard stroke disease.

3.1 analysis datasets

The dataset we used it talks about stroke heart disease. It contains 5110 rows and 12 columns we can divide the data into categorical (for words) and numerical (for numbers) in this data set we have 7 numerical data which is (ID, age, hypertension, heart disease, avg glucose level, bmi, stroke) and we have 5 categorical data which is (gender, ever married, work type, residence type, smoking status) in this data set, the number of females is more than males, and the number of non-smoker is more than the number of smokers, also the percentage of married people is more, and the number of those who have private work is more than the number of those who have government job or self-employed. Here is an explain for each columns what it means; "ID" It means each person's private number, "gender" It means is it female or male? , "age" It means the age of the person, and the ages in this dataset range from 30 to 90 years, "hypertension" There is a number '0', which means that he did not have hypertension, and the number '1' means that the person has it, "heart disease" There is a number '0' and '1' and the number '0' means that the person does not suffer from heart disease, and the number '1' is that the person has heart disease, "ever married" It means, has he been married before or is he married, 'Yes, No' , "work type" What type of work is it a governmental or private job or self-employed ? , "residence type" What is meant by the type of residence is whether it is urban or rural? , "avg glucose level" What is meant by the level of glucose, the ratio in this dataset ranges from 80 to 300, "BMI" It means the body mass index, the ratio in this dataset ranges from 20 to 50, "smoking status" It means the state of smoking Is it 'never smoked or formerly smoked or smokes or unknown'? , "stroke" The numbers '0' and '1' , '0' means the person does not have a stroke but '1' means have a stroke. [30]

3.2 preprocessing datasets

We did some amendments to the dataset The first amendment is: Convert text data into digital data and the second amendment: the empty data we did “drop” and then deleted it.

3.3 Python language

Python is an interpreted, versatile language, widely used in many fields, such as building standalone programs using graphical interfaces and in web applications, and it can be used as a scripting language to control the performance of many programs [31]

3.5 Google Colab

Google Colab was developed by Google to provide free access to GPU’s and TPU’s to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook. [32]

Here we talked about each step of coding using Python language in Google Colab

A. Required libraries

Here are all the libraries that we used in Python that we need in our system, first library used to read csv files. Numby is data structure and every operation happen and we used it convert data to matrix. Sklearn model selection is to divide data to training set and test set. Sklearn pre-processing is to convert text type data to numbers because machine learning deals with numbers and considered as pre-processing. Sklearn metrics is formal matrix to measure the model using performance measures to give us results. SVM, onevsrest classifier, mlp classifier, knn, svm random forest, Logistic Regression, Decision Tree, SGD classifier are algorithms used in our Code. Recall, precision, f1 Score are our Code performance measures.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn import svm
from sklearn.multiclass import OneVsRestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import recall_score
from sklearn.metrics import precision_score
from sklearn.metrics import f1_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import RidgeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.ensemble import RandomForestClassifier
```

Figure 5 of the required libraries

B. Algorithms

After pre-processing stroke dataset, we applied SVM, Logistic Regression, Decision Tree, KNN, Naïve Bayes, SGD, Random Forest.

B1.SVM Algorithm
The code is:

```
clf = OneVsRestClassifier(svm.SVC(gamma=1, C=10 , kernel='rbf'))
clf_output = clf.fit(xtrain, ytrain)
y_score= clf_output.predict(xtest)
print("SVM kernel RBF")
print("accuracy="+str(accuracy_score(ytest, y_score)))
print("recall=" + str(recall_score(ytest, y_score, average='macro')))
print("precision=" + str(precision_score(ytest, y_score, average='macro')))
print("F1=" + str(f1_score(ytest, y_score, average='macro')))
```

The C (a hypermeter for error control), Gamma (a hypermeter for curvature weight) is also a hypermeter that is set before the training model.[33] Kernel Functions transform large sets of data into linear equations in a dimensional space that has a high number of dimensions.[34] As you can see, we used the type (RBF). is a popular and widely used kernel function. Nonlinear data is usually chosen for it,[35] the last section deals with finding performance measures (accuracy, recall, precision, f Score)
Then the output is :

```
SVM kernel RBF
accuracy=0.9626612355736592
recall=0.5
precision=0.4813306177868296
F1=0.4904877205119336
```

Table 1: the result of SVM Algorithm

accuracy	0.9626612355736592
recall	0.5
precision	0.4813306177868296
F1	0.4904877205119336

B2. Logistic Regression Algorithm
The code is:

```
logistic = LogisticRegression()
logistic=logistic.fit(xtrain, ytrain)
y_score = logistic.predict(xtest)
# accuracy on X_test
accuracy = logistic.score(xtest, ytest)
print("LogisticRegression")
print ("accuracy ="+str(accuracy) )
print("recall=" + str(recall_score(ytest, y_score, average='macro')))
print("precision=" + str(precision_score(ytest, y_score, average='macro')))
print("F1=" + str(f1_score(ytest, y_score, average='macro')))
```

We divided the data into two sections, first one is the training and second one is the test, the last section deals with finding performance measures (accuracy, recall, precision, f Score)
Then the output is:

```
LogisticRegression
accuracy =0.9626612355736592
recall=0.5
precision=0.4813306177868296
F1=0.4904877205119336
```

Table 2: the result of Logistic Regression Algorithm

accuracy	0.9626612355736592
recall	0.5
precision	0.4813306177868296
F1	0.4904877205119336

B3. Decision Tree Algorithm
The code is:

```
DecisionTreeClass = DecisionTreeClassifier()
DecisionTreeClass=DecisionTreeClass.fit(xtrain, ytrain)
y_score = DecisionTreeClass.predict(xtest)
# accuracy on X_test
accuracy = DecisionTreeClass.score(xtest, ytest)
print ("DecisionTreeClassifier")
print ("accuracy ="+str(accuracy) )
print("recall=" + str(recall_score(ytest, y_score, average='macro')))
```

```
print("precision=" + str(precision_score(ytest, y_score, ave
rage='macro')))
print("F1=" + str(f1_score(ytest, y_score, average='macro'
)))
```

As you can see there are 2 sections, first one is the training and second one is the test, the last section deals with finding performance measures (accuracy, recall, precision, f Score)

Then the output is:

DecisionTreeClassifier
accuracy =0.9192124915139172
recall=0.538601102705475
precision=0.527667493796526
F1=0.5314797837109659

Table 3: the result of Decision Tree Algorithm

Accuracy	0.9192124915139172
Recall	0.538601102705475
Precision	0.527667493796526
F1	0.5314797837109659

B4. KNN Algorithm

The code is:

```
knn = KNeighborsClassifier(n_neighbors=7)
knn=knn.fit(xtrain, ytrain)
y_score = knn.predict(xtest)
# accuracy on X_test
accuracy = knn.score(xtest, ytest)
print ("knn")
print ("accuracy ="+str(accuracy) )
print("recall=" + str(recall_score(ytest, y_score, average='
macro')))
print("precision=" + str(precision_score(ytest, y_score, ave
rage='macro')))
print("F1=" + str(f1_score(ytest, y_score, average='macro'
)))
```

As you can see, we chose 7 to be value for k randomly and the last section deals with finding performance measures (accuracy, recall, precision, f Score)

Then the output is:

knn
accuracy =0.9613034623217923
recall=0.4992947813822285
precision=0.4813052345343304
F1=0.490134994807892

Table 4: the result of KNN Algorithm

Accuracy	0.9613034623217923
Recall	0.4992947813822285
Precision	0.4813052345343304
F1	0.490134994807892

B5. Naïve Bayes Algorithm

The code is:

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
y_pred = gnb.fit(xtrain, ytrain).predict(xtest)
print ("Naive Bayes" )
print("Accuracy=" + str(accuracy_score(ytest, y_score)))
print("recall=" + str(recall_score(ytest, y_score, average='
macro')))
print("precision=" + str(precision_score(ytest, y_score, ave
rage='macro')))
print("F1=" + str(f1_score(ytest, y_score, average='macro'
)))
```

As you can see, we used one of Naive Bayes algorithms which was Gaussian Naive Bayes, and the last section deals with finding performance measures (accuracy, recall, precision, f Score).

Then the output is:

Naive Bayes
Accuracy=0.9613034623217923
recall=0.4992947813822285
precision=0.4813052345343304
F1=0.490134994807892

Table 5: the result of Naïve Bayes Algorithm

Accuracy	0.9613034623217923
Recall	0.4992947813822285
Precision	0.4813052345343304
F1	0.490134994807892

B6. SGD Algorithm

The code is:

```
clf = SGDClassifier(alpha=0.01)
clf_output=clf.fit(xtrain, ytrain)
y_score=clf_output.predict(xtest)
print("SGDClassifier")
print("accuracy="+str(accuracy_score(ytest, y_score)))
print("recall=" + str(recall_score(ytest, y_score, average='
macro')))
print("precision=" + str(precision_score(ytest, y_score, ave
rage='macro')))
print("F1=" + str(f1_score(ytest, y_score, average='macro'
)))
```

We have used alpha=0.01 in the first line then the last section deals with finding performance measures (accuracy, recall, precision, f Score).

Then the output is:

SGDClassifier
accuracy=0.9613034623217923
recall=0.4992947813822285
precision=0.4813052345343304
F1=0.490134994807892

Table 6: the result of SGD Algorithm

Accuracy	0.9613034623217923
Recall	0.4992947813822285
Precision	0.4813052345343304
F1	0.490134994807892

B7. Random Forest Algorithm
The code is:

```
clf = RandomForestClassifier(max_depth=100, random_state=0)
clf=clf.fit(xtrain, ytrain)
y_score=clf.predict(xtest)
print("RandomForestClassifier")
print("accuracy="+str(accuracy_score(ytest, y_score)))
print("recall=" + str(recall_score(ytest, y_score, average='macro')))
print("precision=" + str(precision_score(ytest, y_score, average='macro')))
print("F1=" + str(f1_score(ytest, y_score, average='macro')))
```

max_depth represents the depth of each tree in the forest. The deeper the tree, the more splits it has and it captures more information about the data, and the last section deals with finding performance measures (accuracy, recall, precision, f Score).
Then the output is:
RandomForestClassifier
accuracy=0.9626612355736592
recall=0.5087382997820233
precision=0.7316451393609789
F1=0.5080249949900713

Table 7: the result of Random Forest Algorithm

Accuracy	0.9626612355736592
Recall	0.5087382997820233
Precision	0.7316451393609789
F1	0.5080249949900713

4. Results and discussion

After pre-processing stroke dataset, we applied SVM, Logistic Regression, Decision Tree, KNN, Naïve Bayes, SGD, Random Forest, and use different evaluation metrics (accuracy, recall, precision, and F1-score) we did all the calculations in the Google Collab program using the Python language, and the results show that the Random Forest algorithm has the highest value of accuracy and precision, you can see that in the result here.

Table 8: the results of the Algorithms

Algorithms/measures	***accuracy***	***recall***	***precision***	***F1***
SVM	96.27%	50%	48.13%	49.04%
Logistic Regression	96.27%	50%	48.13%	49.04%
Decision Tree	91.92%	53.86%	52.76%	53.14%
KNN	96.13%	49.92%	48.13%	49.01%
Naïve Bayes	96.13%	49.92%	48.13%	49.01%
SGD	96.13%	49.92%	48.13%	49.01%
Random Forest	96.27%	50.87%	73.16%	50.80%

5. Concision

We proposed a diagnostic system for heart disease diagnosis in this paper. Our proposed method for the diagnostic system uses classification techniques to accurate detection of heart disease. The different evaluation metrics we have used are, accuracy, recall, precision, and F1-score. The classification algorithms we used for comparison are SVM, Logistic Regression, Decision Tree, KNN, Naïve Bayes, SGD, Random Forest. We tested these algorithms in Google Collab using Python language. We observed from the results that the Random Forest algorithm has the highest value of accuracy and precision followed by the decision tree algorithm which has the highest value in recall and F1-score. We can conclude that the use of classification methods in diagnosing heart diseases helps in improving the accuracy of diagnosis and making the right decision. Future research can improve the results of our research if appropriate dataset is used.

Acknowledgment

Great thankful for partnership at scientific research and computer science department and the staff.

References

[1] V. Krishnaiah, G. Narsimha, N. Subhash Chandra, Heart disease prediction system using data mining techniques and intelligent fuzzy approach: A review, Int. J. Comput. Appl. (2016).
[2] H. Guizhou, M.M. Root, Building prediction models for coronary heart disease by synthesizing multiple longitudinal research findings, Eur. Sci. Cardiol, (2005).
[3| AnimeshHazraArkomita Mukherjee, Amit Gupta, Asmita Mukherjee, "Heart disease diagnosis and prediction using machine learning and data mining techniques: A review" , Research Gate Publications, July 2017, pp.2137-2159.
[4]Cardiovascular-diseases-(CVDs).Retrieved-from, http://www.who.int/cardiovascular_diseases/en/; 2019, July 16.

[5] V. Manikantan & S.Latha,"Predicting the Analysis of Heart Disease Symptoms Using Medicinal Data Mining Methods", International Journal on Advanced Computer Theory and Engineering, Volume-2, Issue-2, pp.5-10, 2013.
[6] Uma.K, M.Hanumathappa, "Heart Disease Prediction Using Classification Techniques with Feature Selection Method", Adarsh Journal of Information Technology, Volume-5, Issue-2, pp.22-29, 2016
[7] Himanshu Sharma, M.A.Rizvi, "Prediction of Heart Disease using Machine Learning Algorithms:A Survey",International Journal on Recent and Innovation Trends in Computing and Communication,Volume5,Issue-8,pp.99-104, 2017.
[8] T. Mythili, Dev Mukherji, Nikita Padaila, Abhiram Naidu, A heart disease prediction model using SVM- decision trees-logistic regression (SDL, Int. J. Comput. Appl. 68 (2013).
[9] An optimized XGBoost based diagnostic system for effective prediction of heart disease
[10] A. Samuel, Some studies in machine learning using the game of checkers, IBM J.Res. Dev. 3 (1959) 210–229, https://doi.org/10.1147/rd.33.0210.
[11] P. Rajpurkar, J. Irvin, K. Zhu, et al., CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, 2017, < https://arxiv.org/abs/1711.05225 > (accessed 20 mar 2018)
[12] Z. Li, C. Wang, M. Han, et al., Thoracic Disease Identification and Localization with Limited Supervision, 2018. < https://arxiv.org/abs/1711.06373 > (accessed 20 mar 2018)
[13]heart-disease https://www.rxlist.com/heart_disease_slideshow_pictures_a_visual_guide/article.htm access at (17\11\2021 10:21AM)
[14]Causes-and-risk-factors https://www.medicalnewstoday.com/articles/237191#causes-and-risk-factors access at (17\11\2021 11:30 AM)
[15] Symptoms of heart disease https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118 access at (17\11\2021 10:55AM)
[16] Machine Learning in Radiation Oncology. Theory and Applications. Editors (view affiliations) Issam El Naqa, Ruijiang Li, Martin J. Murphy (https://link.springer.com/chapter/10.1007/978-3-319-18305-3_1)
[17] Figure of machine learning types https://www.researchgate.net/figure/Overview-of-machine-learning-techniques_fig1_348764759 access at (15\11\2021 9:45 AM)
[18] Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification R. Sathya Professor, Dept. of MCA, Jyoti Nivas College (Autonomous), Professor and Head, Dept. of Mathematics, Bangalore, India. Annamma Abraham Professor and Head, Dept. of Mathematics B.M.S.Institute of Technology, Bangalore, India (https://www.researchgate.net/publication/273246843_Comparison_of_Supervised_and_Unsupervised_Learning_Algorithms_for_Pattern_Classification) access at (18\11\2021 11:22AM)
[19] reinforcement learning https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/ access at (18\11\2021 11:46AM)
[20]semi-supervised-learning https://www.morganclaypool.com/doi/abs/10.2200/S00196ED1V01Y200906AIM006 access at (20\11\2021 9:50AM)
[21]classification techniques https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning#what_is_classification access at (20\11\2021 10:30AM)
[22]types of classification algorithms https://analyticsindiamag.com/7-types-classification-algorithms/ (20\11\2021 11:35AM)
[23] https://www.springboard.com/blog/data-science/what-is-logistic-regression/
[24] http://mineetha.com/2020/06/18/naive-bayes/
[25] Welcome to the Machine (Learning)! | Object Computing, Inc.
[26] https://www.24tutorials.com/machine-learning/knn-algorithm-case-study/
[27]https://stackoverflow.com/questions/42995958/how-to-find-entropy-of-split-points-when-building-decision-tree
[28] IJERPH | Free Full-Text | Rainfall-Induced Landslide Prediction Using Machine Learning Models: The Case of Ngororero District, Rwanda (mdpi.com)
[29] Support Vector Machine (SVM) Algorithm - Javatpoint access at (29/3/2022 1:17PM)
[30] Stroke Prediction Dataset | Kaggle
[31] Python language https://ar.m.wikipedia.org/wiki/%D8%A8%D8%A7%D9%8A%D8%AB%D9%88%D9%86_(%D9%84%D8%BA%D8%A9_%D8%A8%D8%B1%D9%85%D8%AC%D8%A9) (7\2\2022 8:45 PM)
[32] Google Colab https://www.scaler.com/topics/what-is-google-colab/ (7\2\2022 9:11 PM)
[33] https://medium.com/@myselfaman12345/c-and-gamma-in-svm-e6cee48626be
[34] https://www.geeksforgeeks.org/major-kernel-functions-in-support-vector-machine-svm/
[35] https://dataaspirant.com/svm-kernels/