iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** V **Month of publication:** May 2021

DOI: https://doi.org/10.22214/ijraset.2021.34397

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

AI and ML based Machine in HealthCare Predicting Diseases, Suggesting nearby Hospitals & Providing Emergency Medical Assistance

Siddharth Nalwaya[1], Dixita Bhargava[2], Mr. Narasimhayya B E[3]
[1, 2]*Department of (CSE-CTMA), SET - Jain University, Bengaluru, India*
[3]*Assistant Professor, Dept of CSE (IOT), FET- Jain University, Bengaluru, India*

***Abstract:** In recent years, AI and ML have been developing rapidly in hardware implementations, software algorithms and a large number of areas in various applications. Because of which many people worry that AI doctors will take over the human physician for the betterment. We believe that AI doctors will not replace human physicians but help them to make better decisions. There are a various number of successful applications now due to the increased availability of healthcare data and development of big data analytics. Before our AI system can be deployed for use in healthcare, it needs to be trained well enough through the data that is generated from various clinical activities like diagnosis, treatment assignment and so on, this way it can learn the similar associations between subject features and outcome, group of subjects. We often come across terms like algorithm and model, but we don't really know the key difference between them. An algorithm is a mathematical technique or equation (that is, a framework). In other words, it's some sort of framework, but it's not concrete, it will have some parameters, but those parameters have not assigned an actual value.*

I. INTRODUCTION

Artificial intelligence in healthcare is a commonly used term to describe the use of machine learning algorithms or artificial intelligence, to work like humans for the analysis and presentation and comprehension of complex medical and healthcare data. It is the ability of computer algorithms which we provide to approximate conclusions based only on the input data.

The ability to gather lots of data, process it and provide a well defined output to the user is what makes AI technology different from the traditional technologies. It is commonly done using various machine learning algorithms and deep learning. What these algorithms do is, recognize patterns in behavior and create its own logic according to the data that has been provided to it. A vast amount of input data is used to train the machine learning model to gain useful insights and predictions.

II. BACKGROUND

Artificial Intelligence in healthcare has been developing in the past few years very quickly. We can discover key areas of patient care that require any improvements by quickly obtaining patient insights. We have found out that few applications like Nuance provide AI powered solutions to help doctors cut documentation time and improve reporting quality. Service acceleration that suggests the best next step to be taken so that the customers' needs are met and Churn reduction on the other hand uses ML and NLP. It understands the patient, studies the patterns and behavioral trends. Another application uses speech recognition, nlp and wireless integration with medical devices such as blood pressure cuffs to provide assistance to patients known as "Sensely". It has the following features: (1) Self care (2) Clinical advice (3) Scheduling an appointment (4) Nurse line (5) ER Direction.

III. PROPOSED RESEARCH

In general, when there is an emergency case people rush to the nearby hospital and get treated. But there are times where the hospital doesn't have enough facilities for the treatment and asks them to visit another hospital. There can be times when the hospital cheats or gives the wrong treatment mistakenly. To avoid such times, we plan to design a machine which would help us in these emergency cases. In common, when you visit a hospital, you are asked to fill in a form, do all the formalities and then they start with your treatment. What if you had an AI based machine which would assess you and suggest the best hospital? The AI based machine would ask a few questions and analyse and suggest you with the best hospital you need to visit as soon as possible. It would also suggest few medicines which could be taken at that particular time to keep things in control.

If the nearby pharmacy does not have the medicine the ambulance on its way can get the medicine and the situation can be taken in control. To study the above, we have worked on a few applications which help us in gaining knowledge and a better understanding to work with it. We have learned about the various ML models such as Multi layer perceptron , Multi class classification , Reinforcement - Q-learning ,Naive Bayes .Stacking, K mean clustering, Image Data Generator which will help us in the main machine boosting and working.

IV. PLANNING

We plan on learning a hardware system with various software's which would distinguish between physical and mental illness initially with the help of various algorithms and applications which will be taught to the hardware system. Then it would assess the patient quickly by asking a few questions without much medical terms. It would then take the data and put it with the testing set and give a reply to the user. For gaining better knowledge, we have learned about various models and algorithms in Machine Learning, Artificial Intelligence, Deep learning such as Logistic Regression, Decision Trees, Random Forest Classifier, SVM, Gaussian NB, K Neighbors classifier, ANN, Sequential Model we also worked on various applications such as Heart Disease Detection, Breast Cancer Detection, Chronic Kidney Disease Prediction, Diabetes Prediction, Malaria Disease Detection which use the above techniques.

A. Logistic Regression

Any type of regression is a statistical process for estimating the relationships among variables, commonly used to make a prediction about some outcome. Linear regression is one type of regression which is used when we have a continuous target variable. Logistic regression is another type of the regressions where the target variable or the value we are trying to predict is binary i.e. just 0 or 1 or True or False. A general confusion is with where to use linear and logistic regression. If we use a linear regression for a binary target then it will take a lot of time to come with a best fit line. Linear regression will try to fit in all the data and it will end up predicting negative values or values above 1 which is impossible. Logistic regression on the other hand is built using a logistic or sigmoid curve, which has a S shape. This will always be in the range 0 and 1 and makes it a much better fit for any binary classification problem.

Logistic regression is one of the most transparent algorithms, in the sense that you can gauge the importance of individual predictors by what's called odds ratio. Logistic regression doesn't do really good with messy data so you should consider it when you

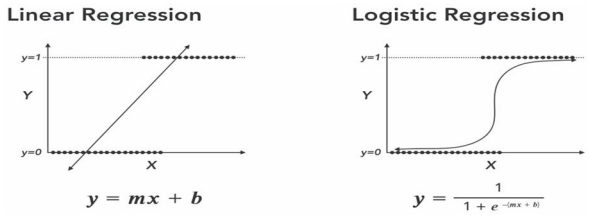


Fig 1 : Difference between Linear Regression and Logistic Regression.

have fairly well-behaved data. The C hyperparameter is a regularization parameter that controls how closely the model fits to the training data. Regularization is a technique which is used to reduce overfitting, which occurs when a model fits too closely to the training data. So regularization combats this overfitting by discouraging overly complex models in some way. Now calling C a regularization parameter is actually slightly misleading as it's actually one over Lambda where Lambda is actually the regularization parameter.

B. Decision Trees

For predictive modeling machine learning, Decision Trees are an important type of algorithm. Decision Tree algorithms that can be used for classification or regression predictive modeling problems are referred to as CART (Classification and Regression Trees). We have something known as gini coefficient, which is a number used to measure the degree of inequality in a given distribution. Gini impurity is a measure of how many times a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. Decision Trees generally are used when we talk about classifying data into two parts. For example, it could have numerical data, ranked data, multiple choices data, continuous variables.

In general, the decision tree asks for a question and then classifies the person based on the answer; it's no big deal. A decision tree could be made up of a yes or no question but it is just as easy to build a tree from data which is numeric. If a person has a really high resting heart rate then he should see a doctor, if not then he is doing okay. A decision tree could be ranked too. The classification can be categories or numeric. For the most part, decision trees are pretty congenital to work with. We start classifying at the top and go our way down till we get a point where we can not go any further and that is how we classify a sample. The root node or the root is the top of the tree. The nodes inside the tree which are the branches of the tree are called the internal nodes or just nodes, they have arrows pointing towards them and from them. Lastly, we have the external nodes or the child nodes or the leaves, they have arrows pointing to them but not away from them. For any dataset, we start building a decision tree with every variable or column and keep a track. Because none of the leaf nodes would be 100%, they are considered to be impure. We need a way to measure and compare the gini impurity to find out which was the best separation. There are a bunch of ways to measure impurity, but we will focus on the most popular one called "Gini". The gini impurity = 1 - (yes probability)^2 - (no probability)^2. The tree with the least impurity is considered.

C. Random Forest Classifier

Random Forests are made up of several decision trees. As we know, Decision Trees are easy to build, use and interpret. To quote from The Elements of Statistical Learning "Trees have only one aspect that prevents them from being the ideal tool for predictive learning, namely inaccuracy". In other words, they are not flexible when it comes to classifying new samples but they work great with data used to create them. Random Forests always combine the simplicity of decision trees with a lot of flexibility resulting in a vast improvement in the model's accuracy. For creating a random forest we need to first create a bootstrapped dataset, a bootstrapped dataset is a dataset that is the same size as the original dataset, but in this we just randomly select samples from the original dataset. The important detail is we are allowed to pick the same sample multiple number of times. After creating a bootstrapped dataset, we need to create a random forest but only using a random subset of variables or columns at each step. Any two columns are selected at random for the root node to be selected. Just like the root node, other nodes are selected at random from the bootstrapped dataset. Considering a random subset of variables at each step, we build the tree as usual. Similarly we build more trees by creating more random bootstrapped datasets considering a subset of variables at each step.

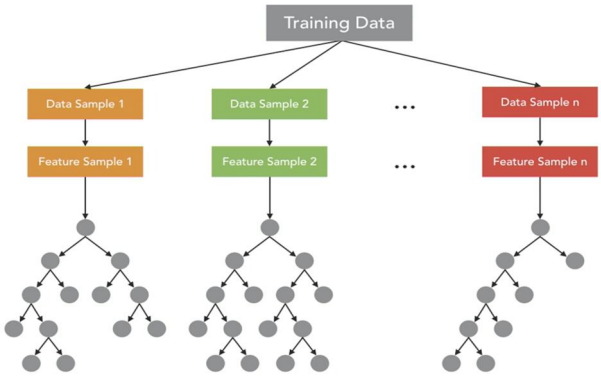


Fig 2 Example of how random forest classifiers work.

Using a bootstrap sample of each tree and considering only a subset of the variables at each step results in a wide variety of trees which we can further use. The variety is what makes the random forests different and more effective from an individual decision tree. Once we have the random forest, let's see how we use it. When we have new data we take in all the data in all the random forests and keep a track of that. Once we have considered all the random forests we see the maximum number of votes and accordingly conclude the result. Bagging is bootstrapping the data and using the aggregate to make a decision. Sometimes the bootstrapped dataset does not include all the entries from the original dataset, we allow duplicate entries in the bootstrapped dataset. Some of the original data is not present in the bootstrapped dataset as it is duplicated, this is called the "Out-Of-Bag Dataset". We see that the Out-Of-Bag data was not used to create the tree, so now we can run it through again and again and see if it correctly classifies the sample.

By the proportion of Out Of Bag samples that were correctly classified by the Random Forest we can measure how accurate our random forest is. The proportion of Out-Of-Bag samples that were not correctly classified is known as "Out-Of-Bag Error". Now we can compare the Out-Of-Bag Error for a random forest built using only 2 variables per step to a random forest built using 3 variables per step and choose the most accurate random forest by testing a bunch of different settings. In other words, we create and design a random forest and estimate the accuracy of a random Forest, then change the number of variables used per step then we do this for a bunch of times and then choose the one that is most accurate. Typically, we start by using the square of the number of variables and then try a few settings above and below the value. Random Forests consider 2 types of missing data : (i) Missing data in the original dataset used to create the random forest (ii) Missing data in a new sample that you want to categorize

D. SVM

Support vector machines(SVM) is a classifier that finds an optimal hyperplane that maximizes the margin between two classes. Taking a very basic example with only two dimensions, X1 and X2, we want a line to separate the red squares from the blue circles. So any line shown on this plot is really a feasible option. They would all perfectly separate the red squares from the blue circles but there has to be an optimal line or decision boundary. We want a line that is evenly spaced between these two classes to give a little bit of buffer for each class. Mathematically the process is defined as maximizing the margin between your decision boundary and the closest points.. Support vector is the name for the perpendicular line from your decision boundary to your closest points in both classes. On the right, it's the perpendicular line that goes from that green optimal hyperplane to those filled end points, that is filled in a blue circle and then filled in red squares. The goal is to maximize the length of those support vectors. Now in the example we can visually see that none of these are likely to be the ideal decision boundary. So a hyperplane is just a generalized term to identify your decision boundary in an N-dimensional space. So in two dimensions, that's just a line. In three dimensions, it becomes a plane and so on. We're using a straight line to separate out examples in two-dimensional space, and we're using a flat hyperplane to separate examples in a three-dimensional space. We're not using any curved lines or curved hyperplanes. So what happens when we have data that can't be separated by a straight line or a hyperplane? That leads us to this really creative trick called the kernel trick which transforms data that is not linearly separable in N-dimensional space into a higher dimension where it is linearly separable. So we have this two-dimensional data where we're trying to identify the presence of cancer based on gene X and gene Y. This data clearly is not linearly separable in two dimensions. We couldn't draw a straight line here that would separate the blue from the red. We would need a circle in order to do that. However, if we were to project this data from two-dimensional space into three dimensional space like this, all of a sudden, now we can use a flat hyperplane to split our data very nicely to identify the presence of cancer. So that's the kernel trick and it's a really powerful tool for SVM.

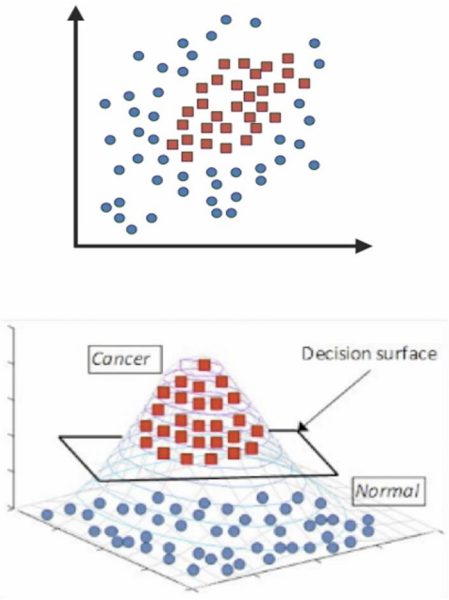


Fig 3: SVM Working

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

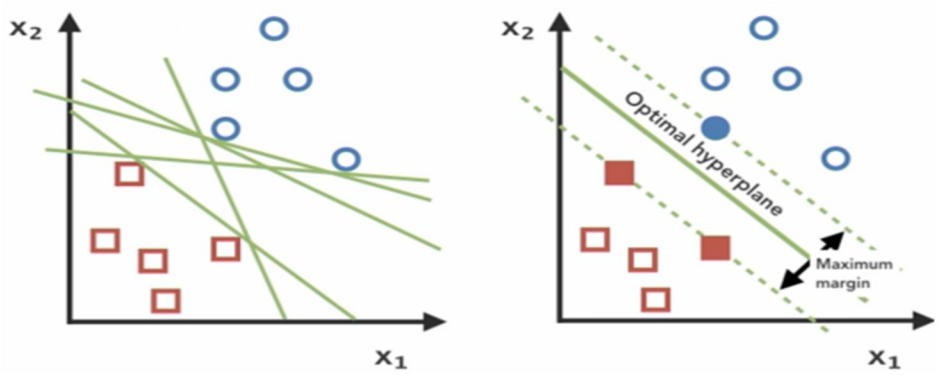


Fig 4 : Use of SVM

E. Q-learning

With Q-learning we are reinforcing certain ways that we want the machine to have. In Q-learning, there are set environments or states, there are also possible actions that can respond to these states. In Q-learning, we want the machine to improve the quality of the outcome. This is represented with the letter Q.

F. Keras

Keras is a framework from Deep Learning, used to build artificial neural networks. It's commonly used in companies like Google and Facebook. Keras acts as a front end which helps the programmer to write a few lines of code and build a model, with using tensorflow or theano in the backend. It is designed to build complex applications with ease in coding.

G. Sequential Model in ANN

Sequential Model is one of the commonly used models in the Artificial Neural Network. It consists of Dense Layers which inturn consist of the number of neurons required, the input_dim, the activation function, the kernel initialiser and many more attributes. A sequential model can have N numbers of Dense layers as any neural network can work with a huge number of layers. They are called sequential because of the way they perform their task. The first layer takes input from the input layer and assigns specific weight to it randomly and with the help of various mathematical functions it processes the data and passesit to the next layer and the networking continues. This networking is done in a sequence which we specify.

H. Recurrent Neural Network

RNN(Recurrent Neural Networks) is a type of Neural Network where the current step is based on the previous output to get a result. In the old and traditional Neural Networks, all the inputs and outputs were independent of each other, but in cases like when it is required to predict the next word of a given sentence, the previous words are required in predicting them and hence there is a need to remember the previous words. When RNN came into existence, it solved this issue with the help of something known as Hidden Layer which is the main and most important feature of RNN, the Hidden state, which remembers some information about a sequence used. RNN remembers all information about what has been calculated in its memory. The same parameters are used for each input as it performs the same task on all the inputs or hidden layers to produce the output. Unlike other neural networks, this reduces the complexity of parameters.

V. APPLICATIONS

A. Heart Disease Detection

Cardiovascular Disease is also known as Heart Disease, it is generally referred to the condition that involves a blocked blood vessel or a narrowed blood vessel that can lead to heart attack or chest pain or a stroke other heart conditions such as those that affect your hearts muscle valves or rhythm also are considered forms as heart disease and many forms of heart diseases can be prevented or treated with healthy lifestyle choices. In this, we have built an application which classifies if a person has cardiovascular disease or not with help of python and machine learning. Python is a programming language which is easy to understand by everyone.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

Machine Learning is a branch of Artificial Intelligence through which machines can simulate human minds in learning and analysis and thus can work in problem solvings. I believe it is pretty fascinating that we might be able to detect if a person has any heart disease and help them out. We use various libraries such as numpy, pandas, seaborn, from train_test_split, StandardScaler, LogisticRegression, accuracy_score, DecisionTreeClassifier, confusion_matrix, classification_report, KFold, GaussianNB, RandomForestClassifier, cross_val_score, SVC, KNeighboursClassifiers and matplotlib.pyplot. We have taken the dataset from kaggle, kaggle is an online platform where you can upload or search for datasets/notebooks. This dataset has an usability rate of 7.6 consisting of 303 rows and 75 columns in which we use only 14 columns. After looking at various datasets for cardiovascular disease we found that this was giving the best outcomes.
Data exploration is a very important aspect of data analysis and model building. Data exploration takes a lot of time in a data science project consisting of data cleaning and preprocessing.

Dataset attribute information:

1) Age refers to age in years.
2) Sex is denoted as 0s and 1s where 0 denotes female and 1 denotes male.
3) There are 4 types of chest pains such as - typical angina, atypical angina, non - anginal pain and asymptomatic. Typical angina pain is defined as substernal chest pain precipitated by physical exertion or emotional stress and relieved with rest or nitroglycerin, Angina pectoris which does not have associated classical symptoms of chest pain. Symptoms may include weakness, nausea, or sweating is known as atypical angina. Non-anginal pain is a term used to describe chest pain that resembles angina in patients who do not have heart disease. The pain typically is felt behind the breast bone (sternum) and is described as oppressive, squeezing or pressure-like.
4) Trestbps is resting blood pressure (in mm Hg on admission to the hospital)
5) serum cholesterol in mg/dl, A serum cholesterol level is a measurement of certain elements in the blood, including the amount of high and low density lipoprotein cholesterol in a person's blood.
6) fasting blood sugar should be greater than 120 mg/dl
7) There are 3 types of resting electrocardiographic results: Value 0 is normal, Value 1 is having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), Value 2 is showing probable or definite left ventricular hypertrophy by Estes' criteria
8) Thalach refers to maximum heart rate achieved
9) Exang refers to exercise induced angina (1 = yes; 0 = no)
10) oldpeak refers to ST depression induced by exercise relative to rest
11) Slope refers to the slope of the peak exercise ST segment which consists of 3 values - value 1: upsloping, value 2: flat, value 3: downsloping
12) Ca refers to number of major vessels (0-3) colored by fluoroscopy
13) thal: 3 = normal; 6 = fixed defect; 7 = reversible defect
14) Num refers to diagnosis of heart disease (angiographic disease status) : Value 0: < 50% diameter narrowing , Value 1: > 50% diameter narrowing (in any major vessel: attributes 59 through 68 are vessels)

The dataset does not contain any null values so no column had to be deleted for data cleaning. From the heatmap We observe positive correlation between target and cp, thalach,slope and also negative correlation between target and sex, exang,ca,thai,oldpeak. We plot all the attributes by taking 2 attributes at a time to see which has the most chances of being affected. We explore and visualise the data by plotting them in different types and checking which is better.
A heat map is a 2D representation of data in which values are represented by colors. A simple heat map provides an immediate visual summary of the information whereas a more elaborate heat map allows the viewer to understand complex data sets.
Data scaling is done so that the ML models get a better understanding of the data and work with them easier. Our data was in human readable format earlier, this converts our data to the ML language.
After Scaling the data, we break it down to training and testing datasets and check its prediction and accuracy score with different ML model such as Logistic Regression,
Decision Tree, KNeighbour Classifier, GaussianNB, RandomForestClassifier and SVC. It is found that Logistic Regression has the best accuracy score followed by GaussizanNB and Decision Trees.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

```
KNN : 0.8351648351648352
lr : 0.9230769230769231
dtc : 0.9010989010989011
rfc : 0.8681318681318682
NB:  0.9010989010989011
SVC : 0.8791208791208791
```

Fig 5: Acuracy Score.

B. Breast Cancer Detection

Breast cancer is a type of cancer that develops because of the breast tissues. Signs of breast cancer might include a lumps in the breast, change of shape of the breast, dimpling of the skin, fluid from the nipple, a newly-inverted nipple, or a red or scaly patch of skin. Breast Cancer is a common type of cancer around the world, and early detection of this can greatly improve prognosis and survival chances by promoting clinical treatment to patients early. After going through the different types of dataset, we have used the Breast Cancer Wisconsin Data Set from the Kaggle website with a usability of 8.5. The dataset consists of 10 real-valued features are computed for each cell nucleus in 3 types: - mean, se and worst, which makes a total of 30 features for breast cancer detection.

There are several features such as id, diagnos, radius, texture, perimeter, area, smoothness, compactness, concavity and concave points

After exploring we have 569 rows of rows and 33 columns, out of which one column has empty values so we drop that column and format and clean the data. We create a count plot and visualize the counts, by getting a count on the numbers of patients with benign and malignant cells. Columns are encoded by seeing the data types. We notice that all the columns are numerical data except for the 'diagnosis' column which is of categorical data which is represented as an object in the python language. We transformed the categorical data. We use 7 types of models to train our data to check which shows the best result. We use Logistic Regression, Decision Trees, Random Forest Classifier, KNeighborsClassifier, SVC linear, SVC rbf, and GaussianNB.

```
Logistic Regression Training Accuracy:  0.9906103286384976
Decision Tree Training Accuracy:  1.0
Random Forest Classifier Training Accuracy:  0.9953051643192489
K Nearest Neighbor Training Accuracy: 0.9765258215962441
Support Vector Machine (Linear Classifier) Training Accuracy: 0.9882629107981221
Support Vector Machine (RBF Classifier) Training Accuracy: 0.9835680751173709
Gaussian Naive Bayes Training Accuracy: 0.9507042253521126
```

Fig 6 : Accuracy Score.

Below is the confusion matrix and the accuracy of the models on the testing dataset. The confusion matrix tells us the correct and incorrect diagnosis by the model i.e. the true positives and true negatives. After training the model we have tested and found the following predictions were made by the model.

```
[1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 0 1 0 0
 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0
 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1]

[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 0 1 0 1
 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0
 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1]
```

Fig 7 : Difference between Testing Dataset and Original Dataset

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

C. Chronic Kidney Disease Prediction

The program classifies patients as having chronic kidney disease or not using artificial neural networks.

We import libraries such as Sequential, load model from keras, numpy, pandas, LabelEncoder, MinMaxScaler, matplotlib.pyplot, model selection for sklearn and dense layer from the keras layers. This would use Tensorflow in the backend. It's pretty neat that we might be able to identify these patients as having kidney disease or not based on just data and using artificial intelligence so now the patient's you know what's it once this disease is identified we can then take preparations to monitor it or prevent it getting it any worse or prepare for surgery. It can really save people's life using data using machine learning and artificial intelligence and the medical field could use many many data scientists and programmers to do things like this. The dataset has the following attributes:

1) Age refers to the age of the patient
2) current blood pressure is bp
3) specific gravity is sg
4) albumin, which is a protein found in the blood is al
5) sugar is su
6) red blood cells is rbc
7) pus cell is pc
8) pus cell clumps is pcc
9) bacteria is ba
10) blood glucose random is bgr
11) blood urea is bu
12) serum creatinine is sc
13) sodium is sod
14) potassium is pot
15) hemoglobin is hemo
16) packed cell volume is pcv
17) white blood cell count is wc
18) red blood cell count is rc
19) Hypertension is htn
20) diabetes mellitus is dm
21) coronary artery disease is cad
22) appetite is appet
23) pedal edema is pe
24) anemia is ane
25) classification is class

We have 400 patients in our dataset and 26 data attributes to each one of them. We first clean our data by dropping a few columns which have missing values or empty values. To manipulate our data we transform all non-numeric data to numeric using the LabelEncoder.

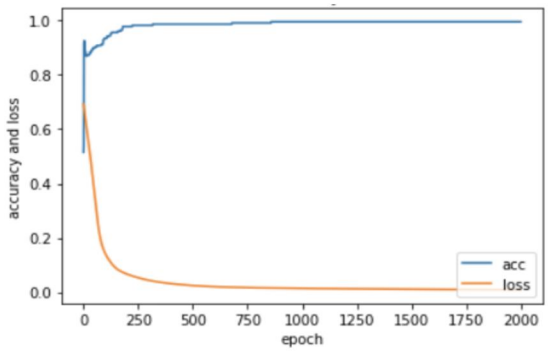


Fig 8 : Model Accuracy & Loss.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

Finally, we build our ANN using the Sequential model and adding dense layers to it. We add a kernel initialiser to the first layer along with the neuron size, input parameter, and activation function which is relu. The kernel initializer helps us in generating tensors with a random distribution. To test the initial random weights of the Keras layer we use an initialiser. We add another layer which uses an activation method called hard sigmoid with only one neuron. We compile the model with a loss function of binary cross entropy which is used for binary classes or two classes. So in this case we have two classes, either a person has a chronic kidney disease or he doesn't. While training the model we have given the epoch size as 2000, epoch is the number of iterations over the entire dataset to train on and we give it a batch size, which is the number of samples per gradient update before training. We give it a value which is the number of patients in our dataset. After training and saving our model, we visualize the models loss and accuracy. Once we have trained our model, we test it to make predictions and we get the following results.

```
Model file:  ckd.model
2/2 [==============================] - 0s 6ms/step - loss: 0.0572 - accuracy: 0.9655

Original  : 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1
Predicted : 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1
Scores    : loss =  0.05716724693775177  acc =  0.9655172228813171
------------------------------------------------------------
```

Fig 9 : Testing Dataset Results.

D. Diabetes Prediction

The program classifies patients as having diabetes or not using random forest classifiers.

We import libraries such as numpy, pandas, seaborn, imputer from sklearn, matplotlib.pyplot. It's pretty neat that we might be able to identify these patients as having diabetes or not based on just data and using random forest classifiers. So now you know what's it once this disease is identified we can then take preparations to monitor it or prevent it getting it any worse or prepare for surgery. It can really save people's life using data using machine learning and the medical field could use many many data scientists and programmers to do things like this. The dataset has the following attributes:

1) Number of pregnancies
2) Glucose level of two hours
3) BP
4) Skin thickness
5) Two hours of serum insulin
6) BMI
7) Function of Diabetes pedigree
8) Patient's age
9) Outcome - whether the patient is diabtic or not

We see the correlation between different types of attributes and then change the diabetes column data from boolean to number. The model is trained and tested with the testing data and the accuracy score is displayed.

```
array([0.72727273, 0.77922078, 0.67532468, 0.67532468, 0.7012987 ,
       0.74025974, 0.76623377, 0.76623377, 0.77631579, 0.80263158])
```

Fig 10 : Accuracy Score.

E. Malaria Disease Detection

Malaria is a very dangerous disease which usually happens because of the mosquitoes in most of the developing and under developing countries these diseases exist. It mostly occurs because of bad water, say sewage or drainage water or rain water accumulated in small areas. This dataset is all about blood tests, so we will be having microscopic images, scanned images of blood and have a look at malaria infected blood. In the dataset we have a training and a testing dataset, out of which the training datasheet contains two folders which are Parasitized and Uninfected. Parasitized refers to a group of blood samples infected by malaria.

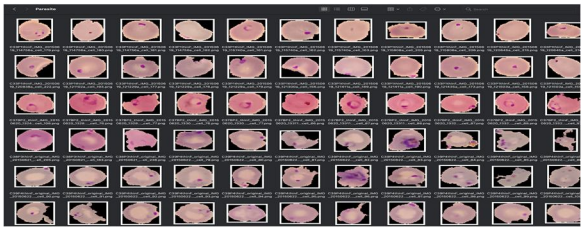


Fig 11 : Images of Parasitized Dataset.

Uninfected refers to a group of blood samples which aren't infected by malaria.

Fig 12 : Images of Uninfected Dataset.

We will be using VVG19 which consists of 1 SoftMax layer, 5 MaxPool layers, 16 convolution layers and 3 Fully connected layers. In the front end we have used a deep learning framework called keras and at the backend tensorflow is being used. Image Data Generator helps us in generating more images using the data augmentation. Initially we will be taking image sizes as (224,224). We add a preprocessing layer of VVG19 which has an input of the image size plus the rgb colors along with weight. We drop the last two rows and then add another layer to replace it using the softmax function. Once we have created the layer we compile the model and check if things are good to go. We then use the Image Data Generator to import pictures from the dataset. The function has various types of attributes so we use few of them to resize and rescale the image. We further fit and train the model and plot and check the loss and accuracy.

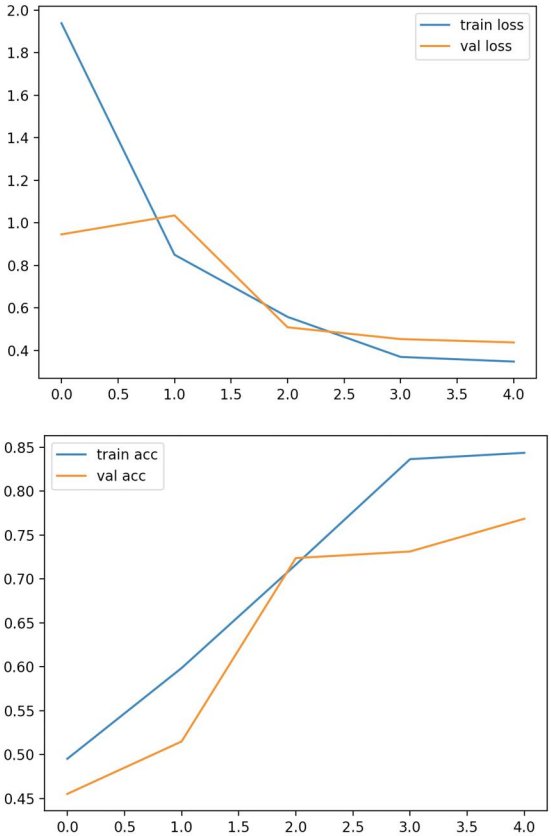


Fig 13 : Checking the loss and accuracy.

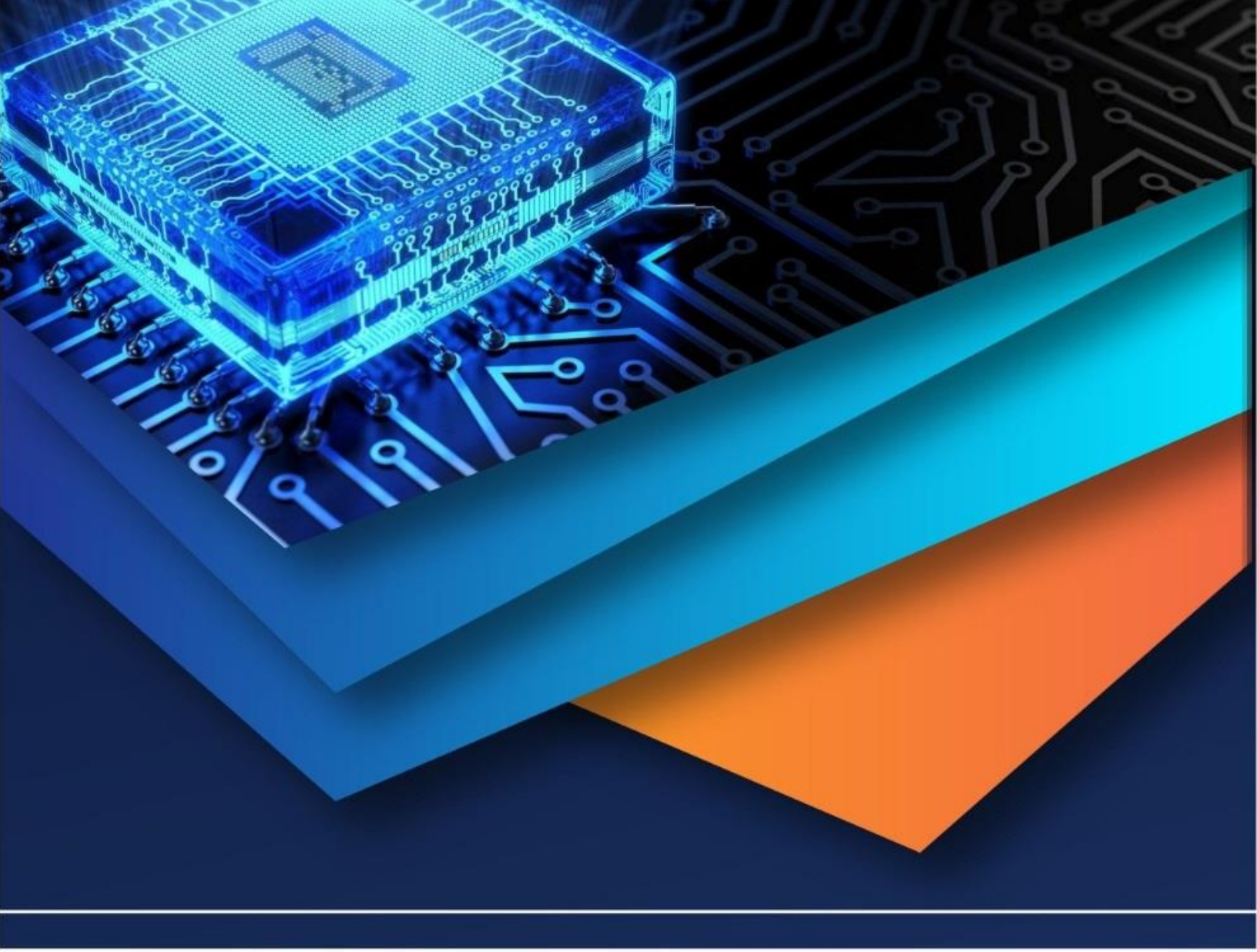
VI. WORKING

The main working could be done by using neural networks and other algorithms which take in user input and analyse and provide the appropriate solution. Using Tensorflow, we can give our desired inputs to train and then add convolutional layers if needed and then a dropout layer for better results. Depending on the input the machine gets in the initial stages, we can improve our algorithm and machine and provide it with better accuracy and results. For providing medical assistance we can have a database where we can store all the medicines names along with their molecules and when it should be taken so that when required we can search and suggest the medicine to the patient. Similarly, it can be done for the hospitals, the data would be provided in the initial stage itself, the model will be improved with time and as we get more data we can update it and let the machine do the working. Suggesting doctors needs to be a very clear task, so we can ask people for the data to create a database, we can get in touch with all the doctors, talk to them about our plan, ask them to provide with data, make a google form and circulate among people through social media and collect information. This way we can collect a lot of data for our training and testing.

REFERENCES

[1] https://www.sciencedirect.com/science/article/pii/S2095809919301535
[2] https://www.researchgate.net/publication/317880442_Artificial_intelligence_in_healthcare_past_present_and_future
[3] https://arxiv.org/pdf/2001.09778.pdf
[4] https://en.wikipedia.org/wiki/Artificial_intelligence_in_healthcare
[5] https://seaborn.pydata.org
[6] https://www.kaggle.com/merishnasuwal/breast-cancer-prediction-dataset
[7] https://www.kaggle.com/uciml/breast-cancer-wisconsin-datahttps://www.kaggle.com/uciml/breast-cancer-wisconsin-data
[8] https://www.slideshare.net/mathupuji/medical-imaging-techniques-for-breast-cancer-a-study
[9] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isna.html
[10] https://seaborn.pydata.org/generated/seaborn.countplot.html
[11] https://www.kaggle.com/piotrgrabo/breastcancerproteomes
[12] https://www.sciencedirect.com/science/article/pii/S1350453302001947
[13] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html
[14] https://seaborn.pydata.org/generated/seaborn.heatmap.html
[15] https://en.wikipedia.org/wiki/Heat_map
[16] https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/canjclin.37.5.258
[17] https://www.forbes.com/sites/bernardmarr/2018/07/27/how-is-ai-used-in-healthcare-5-powerful-real-world-examples-that-show-the-latest-advances/?sh=fd74a915dfbe
[18] https://www.youtube.com/watch?v=j6EB9HO6acE&t=1183s
[19] https://github.com/yzhao062/PyHealth
[20] https://healthitanalytics.com/news/top-5-use-cases-for-artificial-intelligence-in-medical-imaging
[21] https://www.internationalsos.com/client-magazines/in-this-issue-3/how-ai-is-transforming-the-future-of-healthcare
[22] https://seaborn.pydata.org
[23] https://www.kaggle.com/sulianova/cardiovascular-disease-dataset
[24] https://www.mayoclinic.org/diseases-conditions/high-blood-pressure/expert-answers/pulse-pressure/faq-20058189
[25] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isna.html
[26] https://seaborn.pydata.org/generated/seaborn.countplot.html
[27] https://seaborn.pydata.org/tutorial/color_palettes.html
[28] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html
[29] https://seaborn.pydata.org/generated/seaborn.heatmap.html
[30] https://www.newgenapps.com/blog/random-forest-analysis-in-ml-and-when-to-use-it/
[31] https://en.wikipedia.org/wiki/Heat_map
[32] https://towardsdatascience.com/predicting-presence-of-heart-diseases-using-machine-learning-36f00f3edb2c
[33] https://www.kaggle.com/uciml/pima-indians-diabetes-database
[34] https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118
[35] https://www.kaggle.com
[36] https://www.healthline.com/health/chest-pain#TOC_TITLE_HDR_1
[37] https://archive.ics.uci.edu/ml/datasets/heart+disease
[38] serum cholesterol meaning
[39] st-t abnormality
[40] https://towardsdatascience.com/data-exploration-and-analysis-using-python-e564473d7607
[41] https://en.wikipedia.org/wiki/Heat_map
[42] https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35
[43] https://www.youtube.com/watch?v=yIYKR4sgzI8&t=336s
[44] Master Machine Learning Algorithms 2016
[45] https://www.sciencedirect.com/topics/medicine-and-dentistry/logistic-regression-analysis

[46] https://www.statisticssolutions.com/what-is-logistic-regression/?__cf_chl_jschl_tk__=81e581ca07e9495c62d64063c7785cbf03a93292-1612786059-0-AZ1dDWa3oqCF_ZB5GSCVUm02nqBa9qe7xt5ewFpmVBi9DIE8G6xWEShd6mvj-arqzYroQ8QDeyvOqKsf-i0PZkl6TSoUXuvhyDfrtCJZDIipzWUq25XA1yiL0fC3GIudxM3F626nS0gm9D-bXFSjWRJSaXf_o9aciToF8Uq63OEAygiJzPUOQ_I54BP0BQiQuFTzlH3up2d-R8J84cMLfUrkxuqwhjxBbFSdBEYq_rP78N_8EZedEBMZVX2W_TJlVHgab_GLj4APu_LJ4FQ6jAFxd0eSS9fD3NwNhpjkfZXv9Tv3I37WEe4L5sDt5fsF1CD5RJvJnNITfrvPFa3s5bo
[47] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[48] https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397
[49] https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/
[50] https://www.sciencedirect.com/science/article/abs/pii/S0169260715303369
[51] https://link.springer.com/chapter/10.1007/978-3-642-16239-8_8
[52] https://www.scirp.org/html/6-9101686_31887.htm
[53] https://www.sciencedirect.com/science/article/pii/S2214317316300099
[54] https://link.springer.com/chapter/10.1007/0-387-25465-X_9
[55] https://www.sciencedirect.com/science/article/pii/B9781558602472500358
[56] https://link.springer.com/article/10.1007/BF00116251
[57] https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/
[58] https://en.wikipedia.org/wiki/Support-vector_machine
[59] https://scikit-learn.org/stable/modules/svm.html
[60] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
[61] https://www.geeksforgeeks.org/q-learning-in-python/
[62] https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56
[63] https://keras.io
[64] https://en.wikipedia.org/wiki/Keras
[65] https://en.wikipedia.org/wiki/Recurrent_neural_network
[66] https://builtin.com/data-science/recurrent-neural-networks-and-lstm
[67] https://www.mayoclinic.org/diseases-conditions/chronic-kidney-disease/symptoms-causes/syc-20354521
[68] https://en.wikipedia.org/wiki/Chronic_kidney_disease
[69] https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY