A Project Report on

Brain Tumor Prediction using Machine Learning

Submitted in partial fulfillment of the requirement for the award of the degree of

B.TECH- CSE-BAO-1

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No 14 of 2011)

Under the supervision of:

Mr. Deependra Rastogi

Submitted By

Ujjwal Pandey-19SCSE1210019
Shubham Shami– 19SCSE1210016

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGGALGOTIASUNIVERSITY, GREATER NOIDA
INDIASEPTEMBER2021

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **19SCSE1210016 – SHUBHAM SHAMI, 19SCSE1210019 – UJJWAL PANDEY** has been held on _[redacted] and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:

Place:

ACKNOWLEDGEMENT

Deependra Rastogi for their valuable inputs, able guidance, encouragement, whole-hearted cooperation, and constructive criticism throughout our project.

We deeply express our sincere thanks to our Head of Department for encouraging and allowing us to present the project on the topic "**Brain Tumor Prediction Using Machine Learning**" at our department SCSE to partially fulfill the requirements leading to the award of B.Tech Degree. We pay our respects and love to our parents and all other family members and friends for their love and encouragement throughout our careers. Last but not least, we thank our friends for their cooperation and support.

1. UJJWAL PANDEY 19SCSE1210019

2. SHUBHAM SHAMI
19SCSE1210016

Abstract

Brain Tumor is one of the major threats confronted by many people around the world. As per the International Agency of Research on Cancer (IARC) more than one million people are diagnosed with brain tumors per year around the world, with an increased fatal rate. During brain tumor studies, the occurrence of the abnormal tissues is easily detectable most of the time, still, accurate segmentation and characterization of these abnormalities are not genuine. In the present scenario, the radiologists have to manually study the tumors with the available medical imaging tools and generate a report. The process is time-consuming. Although much signs of progress have been made, segmentation of brain tumors from MR Images in a quick, accurate, authentic, and reproductive way is still a challenging issue. To overcome this problem a system that will detect the tumor and will classify them as benign and malignant has been proposed in this paper by using image processing in integration with machine learning. Which will help to detect the tumor and classify them into benign and malignant in quick time. In this work step by step procedure for image pre-processing, segmenting brain tumors using morphological operations, extracting tumor features using DWT, and classification of the tumor using SVM is accomplished with the actual clinical data

CHAPTER NO. **Table of Contents**

Abstract

1 Introduction

1 .1History

The objective of the project

Use of the project

2 Theoretical Background

2.1 Intro to Brain Tumor

2.2Methodology

3 System Analysis and planning

3.1 steps in the requirement analysis process

4 System Design flow-chart data-flow diagram Activity diagram

5 System Implementation Details

5.1 Modules

6 Conclusion and suggestion

Reference

CHAPTER-1 Introduction

HISTORY

Machine Learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

Recommendation engines are a common use case for machine learning. Other popular uses include fraud detection, spam filtering, malware threat detection, business process automation (BPA), and predictive maintenance.

1.1.1 Types of Machine Learning

- Supervised learning:

  In this type of machine learning, data scientists supply algorithms withlabelledtrainingdataanddefinethevariablestheywantthealgorithmtoassess for correlations. Both the input and the output of the algorithm are specified.

- Unsupervised learning:

  This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as predictions or recommendations they output are predetermined.

- Reinforcement learning:

  Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined

  rules. Data scientists program an algorithm to complete at ask and gives it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

1.2 Brain Tumor

A brain tumor is one of the most rigorous diseases in medical science. Effective and efficient analysis is always a key concern for the radiologist in the premature phase of tumor growth.

Histological grading, based on a stereotactic biopsy test, is the gold standard and the convention for detecting the grade of a brain tumor. The biopsy procedure requires the neurosurgeon to drill a small hole into the skull from which the tissue is collected. There are many risk factors involving the biopsy test, including bleeding from the tumor and brain causing infection, seizures, severe migraine, stroke, coma, and even death. But the main concern with the stereotactic biopsy is that it is not 100% accurate which may result in a serious diagnostic error followed by wrong clinical management of the disease. Tumor biopsy being challenging for brain tumor patients, non-invasive imaging techniques like Magnetic Resonance Imaging (MRI) have been extensively employed in diagnosing brain tumors. Therefore, the development of systems for the detection and prediction of the grade of tumors based on MRI data has become necessary. But at first sight of the imaging modality like in Magnetic Resonance Imaging (MRI), the proper visualization of the tumor cells and its differentiation with its nearby soft tissues is a somewhat difficult task which may be due to the presence of low illumination in imaging modalities or its large presence of data or several complexity and variance of tumorslike unstructured shape, viable size, and unpredictable locations of the tumor. Automated defect detection in medical imaging using machine learning has become the emergent field in several medical diagnostic applications. Its application in the detection of brain tumors in MRI is very crucial as it provides information about abnormal tissues which is necessary for planning treatment. Studies in the recent literature have also reported that automatic computerized detection and diagnosis of the disease, based on medical image analysis, could be a good alternative as it would save radiologists time and also obtain a tested accuracy. Furthermore, if computer algorithms can provide robust and quantitative measurements of tumor depiction, these automated measurements will greatly aid in the clinical management of brain tumors by freeing physicians from the burden of the manual depiction of tumors. The machine learningbased approaches like Deep Convey Nets in radiology and other medical science fields play an important role to diagnose the disease in a much simpler way as never done before and hence providing a feasible alternative to surgical biopsy for brain tumors. In this project, we attempted at detecting and classify the brain tumor and compare results of binary and multi-class classification of brain tumors with and without Transfer Learning (use of pre-trained Keras models like VGG16, ResNet50, and Inception v3) using Convolutional Neural Network (CNN) architecture.

CHAPTER-2 Literature survey

Krizhevsky et al. 2012 achieved state-of-the-art results in image classification based on transfer learning solutions upon training a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, he achieved top1 and top-5 error rates of 37.5% and 17.0% which was considerably better than the previous state-of-the-art. He also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry. The neural network, which had 60 million parameters and 650,000 neurons, consisted of five convolutional layers, some of which were followed by maxpooling layers, and three fully-connected layers with a final 1000-way Soft-max. To make training faster, he used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers he employed a recently-developed regularization method called —dropoutl that proved to be very effective.

Simonyan& Zisserman 2014 they investigated the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. These findings were the basis of their ImageNet Challenge 2014 submission, where their team secured the first and the second places in the localization and classification tracks respectively. Their main contribution was a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the priorart configurations can be achieved by pushing the depth to 16–19 weight layers after training smaller versions of VGG with fewer weight layers.

Pan & Yang 2010's survey focused on categorizing and reviewing the current progress on transfer learning for classification, regression, and clustering problems. In this survey, they discussed the relationship between transfer learning and other related machine learning techniques such as domain adaptation, multitask learning, and sample selection bias, as well as covariate shift. They also explored some potential future issues in transfer learning research. In this survey article, they reviewed several current trends of transfer learning.

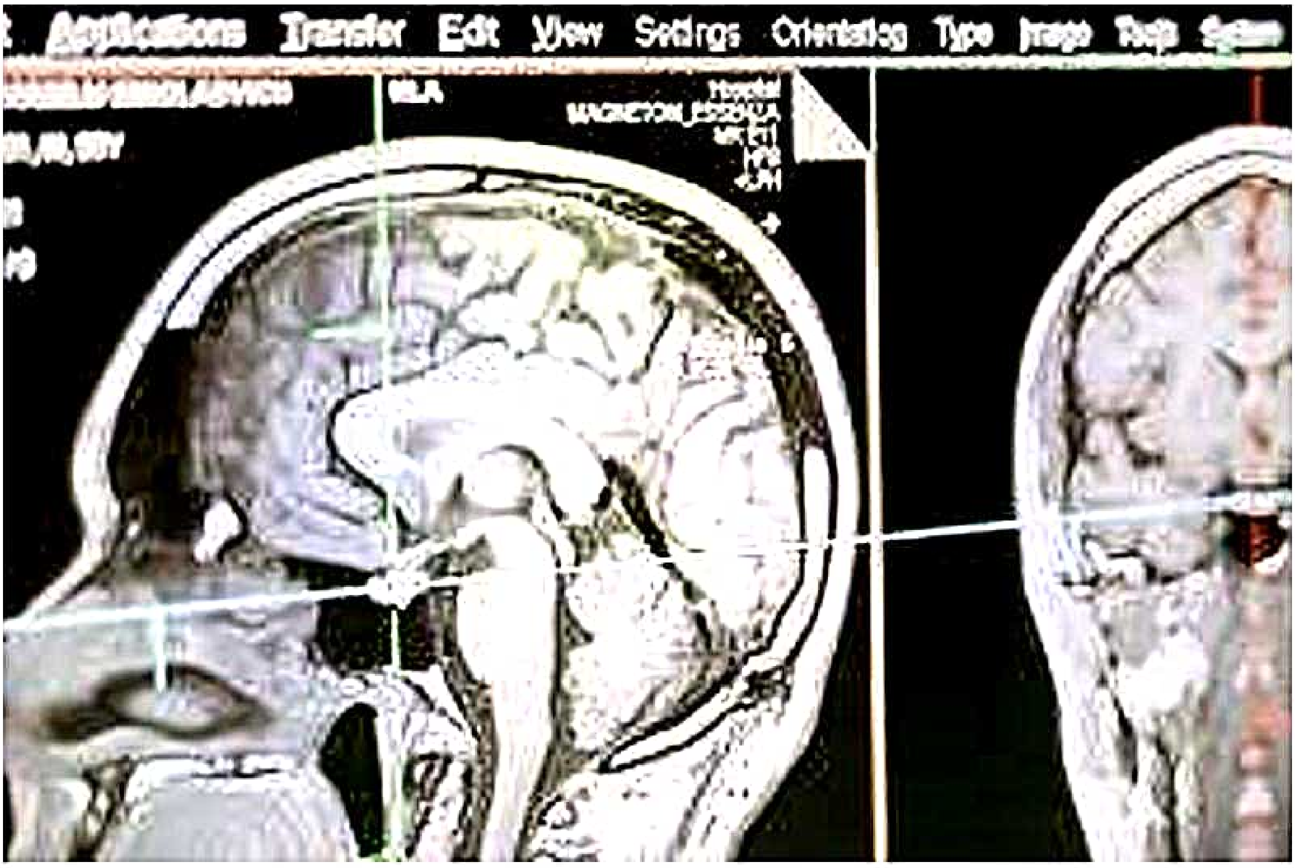


Fig. Working scenario for Brain Tumour Predication using machine learning

Methodology

The two ANN and CNN techniques used in the brain tumor database and their functions are open image separation is analyzed. The steps to follow in applying ANN to the brain tumor database are

1. Import the required packages
2. Enter the data folder

3. Read pictures, give photo labels (Set a photo with Brain Tumor as 1 and the picture may not be with a brain tumor as 0), and store them in Data Frames. 4. Resize images as 256x256 by reading images individually.
5. Make the image normal
6. Divide preset data into train, validation, and test sets
7. Create a model
8. Assemble the model
9. Add the model to the train set.
10. Test the model by using it in a test set.

The ANN model used here has seven layers. The first layer is a flat layer that converts 256x256x3 images into a single-dimensional column. The next five layers are dense layers that have the function of making it work like a relax and the number of neurons in each layer is 128,256,512,256 and 128 respectively. These five layers act as hidden layers and dense final layer with the function of the sigmoid is an outgoing layer with 1 neuron representing two categories.

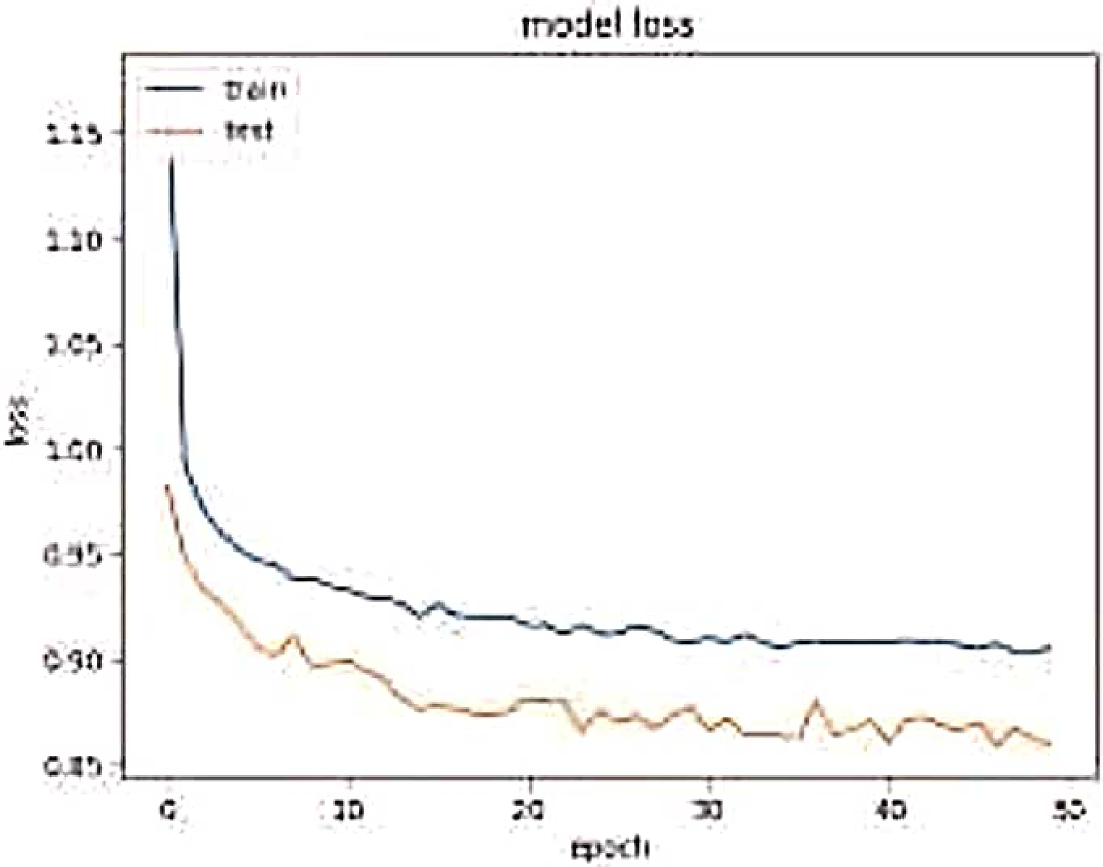
The model is combined with a well-functioning adam and a cross-entropy binary loss function. The model is manufactured and trained by providing training images and validation images. Once a model is trained, tested using a set of test images. Next, the same database is given the CNN method. Steps followed using CNN in the brain tumor database is available

1. Import the required packages
2. Enter the data folder (Yes and No)
3. Set photo labels in class (1 for Brain Tumor and 0 for No Brain Tumor)
4. Convert images (256X256)
5. Make the image normal
6. Separate images on the train, verification, and photo set.
7. Create a sequential model.
8. Assemble the model.
9. Use it on the training database (use the verification set to check training performance).
10. Evaluate the model using test images.
11. Arrange the graph and compare the accuracy of the training and verification.
12. Draw the confusion matrix to find the actual output against the predicted output

CHAPTER-3

SYSTEM ANALYSIS AND PLANNING

Image data is added to the data variant which is an array data type. Photo class labels are also produced and stored in variable data_ target which is also an array. Photos are then uploaded to the data frame. The image database is divided into training, verification, and evaluation data set. Figure 3 represents accuracy and loss obtained when the ANN model is used in the training and certification database When the ANN model is used in training 50 times the accuracy of the training received is 97.13% and the verification accuracy is 71.51%. Same as used in test data provides 80.77% accuracy.



CHAPTER-4

SYSTEM DESIGN

Software design is a process of problem-solving and planning for a software solution. After the purpose and specifications of software are determined, software developers build a design or employ designers to develop a plan for a solution. It includes low-level component and algorithm implementation issues as well as the architectural view. Software design can be considered as putting a solution to the problem(s) in hand using the available capabilities. Hence the main difference between software analysis and design is that the output of the analysis of a software problem will be smaller problems to solve and it should deviate so much even if it is conducted by different team members or even by entirely different groups. But since design depends on the capabilities, we can have different designs for the same problem depending on the capabilities of the environment that will host the solution. The solution will depend also on the used development environment.

4.1 Flow Chart

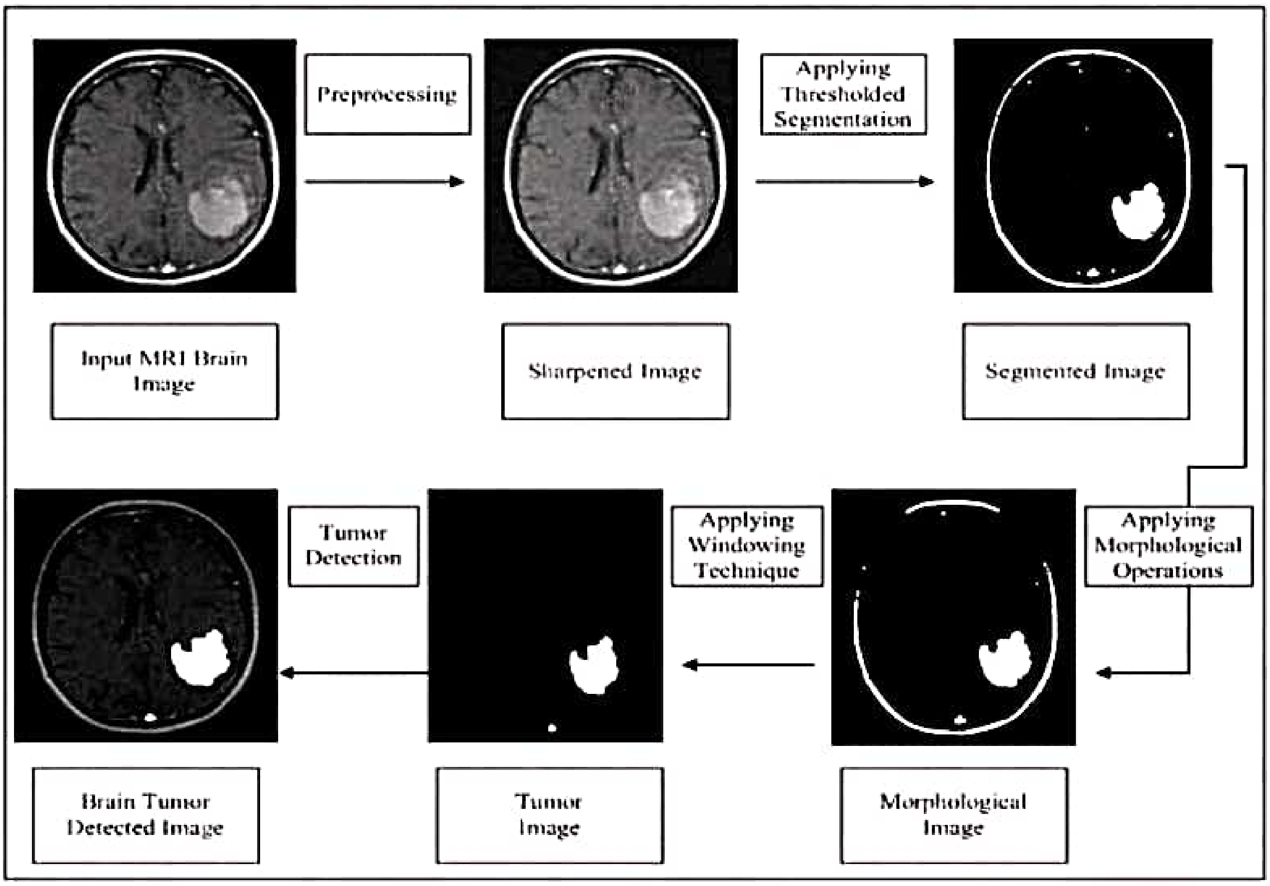

Preprocessing
Applying Thresholded Segmentation
Input MRI Brain Image
Sharpened Image
Segmented Image
Applying Morphological Operations
Applying Windowing Technique
Tumor Detection
Brain Tumor Detected Image
Tumor Image
Morphological Image

Fig: Flow Chart of Brain Tumor Predication using machine learning

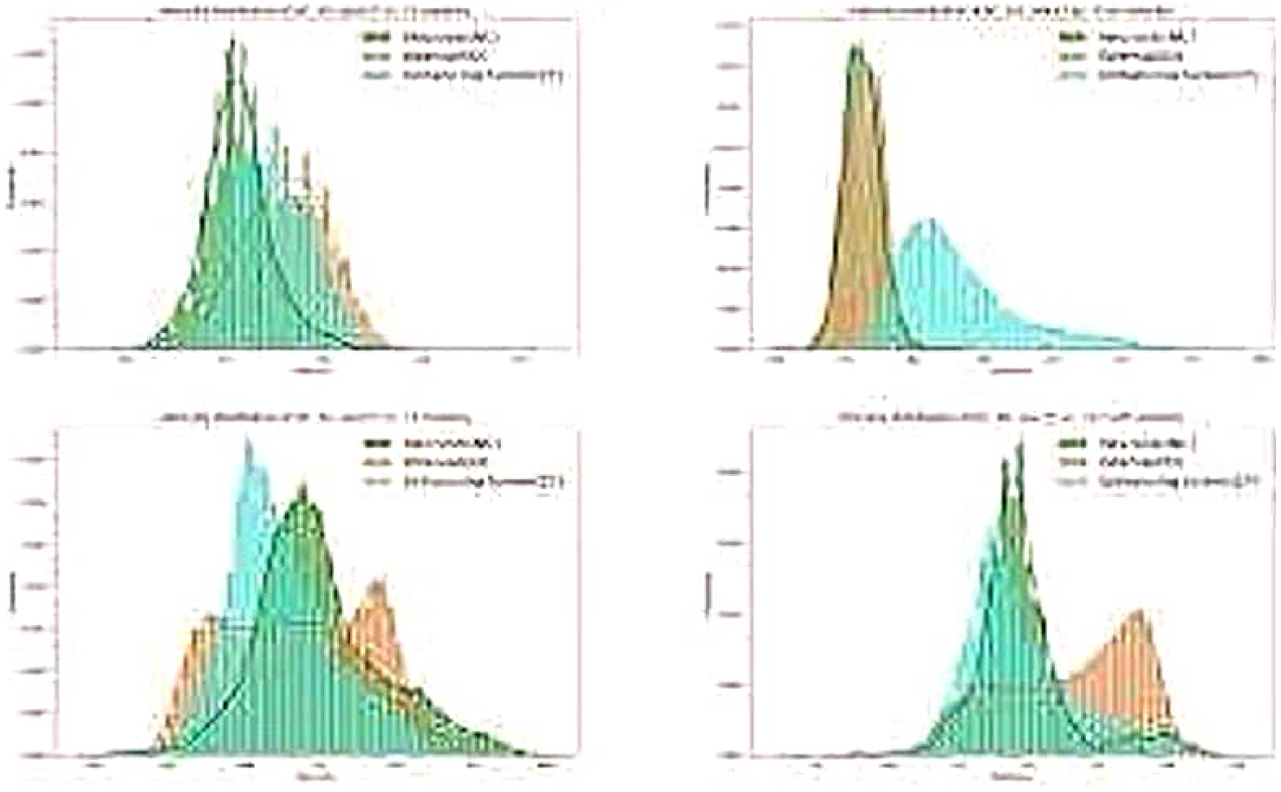


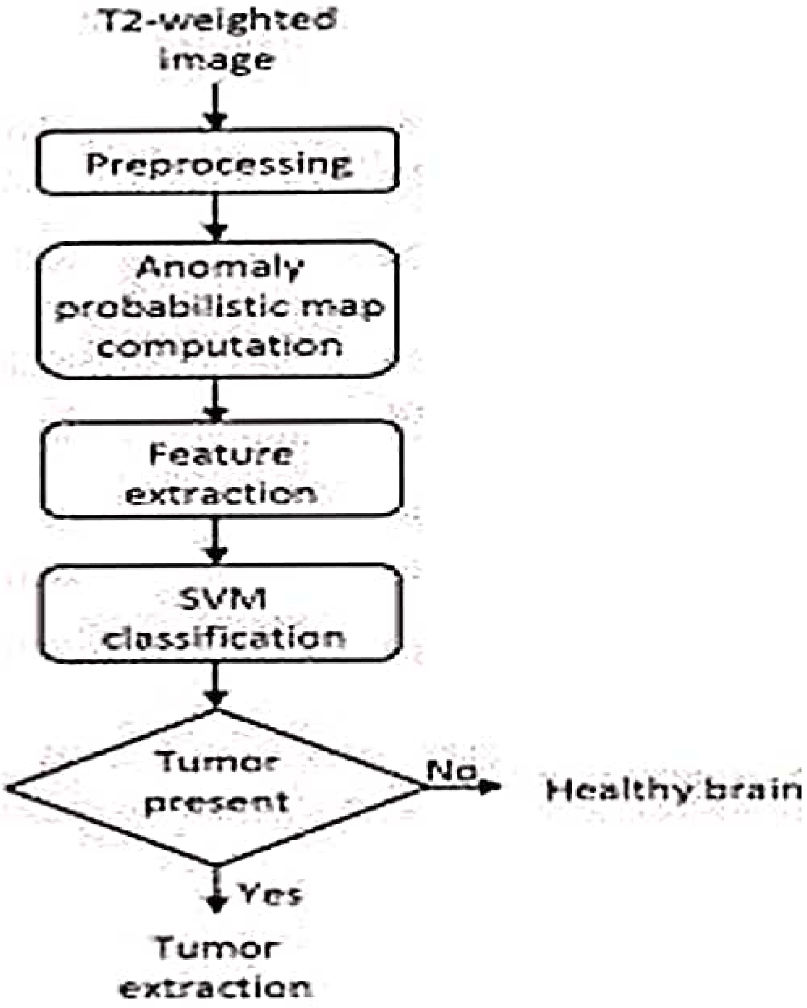
Fig: Flow Chart of Brain Tumor Predication using machine learning

In this study, a cascade CNN model has been proposed that combines both local and global information from across different MRI modalities. Also, a distance-wise attention mechanism is proposed to consider the effect of the brain tumor location in four input modalities. This distance-wise attention mechanism successfully applies the key location feature of the image to the fully-connected layer to overcome over-fitting problems using many parallel convolutional layers to differentiate between classes like the self-co-attention mechanism47. Although many CNN-based networks have been employed for similar multi-modality tumor segmentation in prior studies, none of them uses a combination of an attention-based mechanism and an areaexpected approach.

4.1 Data Flow Diagram

DFD is used to show how data flows through the system and the processes that transform the input data into output. Data flow diagrams are a way of expressing system requirements graphically. DFD represents one of the most ingenious tools used for structured analysis. It is also known as a bubble chart.

The DFD at the simplest level is referred to as a CONTEXT ANALYSIS DIAGRAM. These are expended by level, each explaining its process in detail. Processes are numbered for easy identification and are normally labeled in block letters.



Activity Diagram

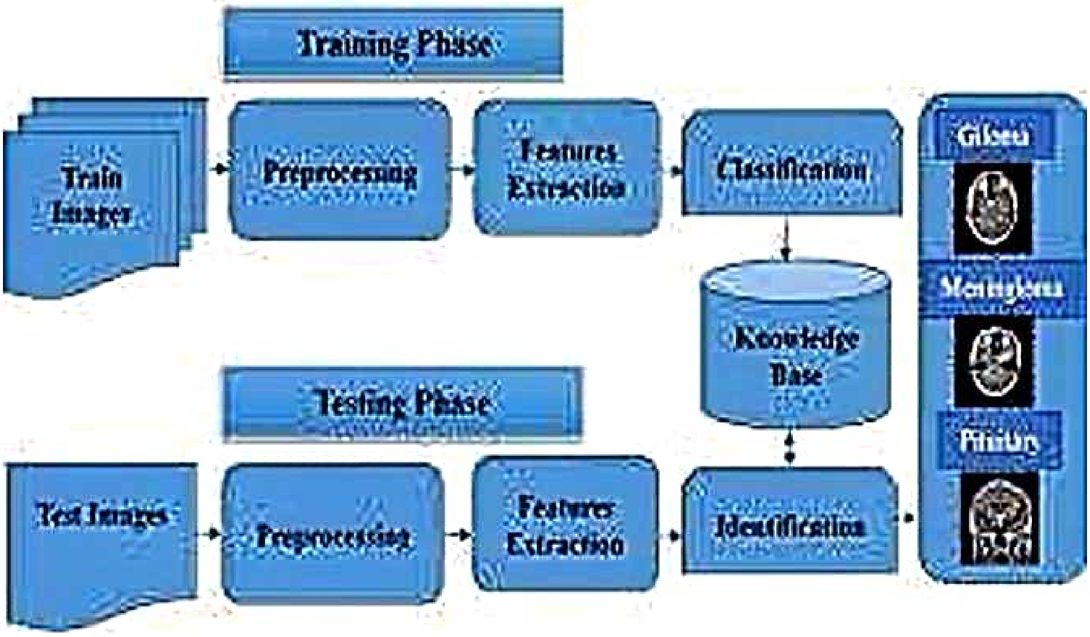
Activity diagrams are a loosely defined diagram technique for showing workflows of stepwise activities and actions, with support for choice, iteration, and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. They consist of:

- Initial node.

- Activity final node.

- Activities

The starting point of the diagram is the initial node, and the activity final node is the ending.



CHAPTER 5

FUTURE WORK

Applying the concept of a newly trained database machine can be used with a more accurate guessing system. Accounts that can be created for each user and then referring to the preferences history of the user's mood can be monitored to see if there is an improvement or if the situation has deteriorated. Now a day's the healthcare industry plays an important role in the treatment of patients' diseases so this is often the case great help in the healthcare industry to inform the user and helpful to the user in the event of his own he does not want to go to the hospital or other clinics, so to include symptoms and everything some useful information within the form of a user who can identify the disease he or she is suffering from therefore the healthcare industry can also find enjoyment in this process by asking for symptoms from the user and log in within the system and in a few seconds, they will tell you the accuracy and arrival to some extent accurate diseases. If the health industry accepts this project it is the job of the doctor they are often reduced and can easily predict a patient's illness. Disease forecast says to provide forecasts for a variety of common and uncommon diseases that, if not reexamined sometimes neglect can turn into a deadly disease and cause many problems for the patient. We can update this project in the future by adding more attributes to the database and more interaction with users and this can be done as an android or ios app. We will fix the system by connecting it to the hospital database.

5 Conclusions & Suggestions

Algorithms for analysing and classifying medical images have gained a great level of attention recently. The experiments we present in this work show that after pre-processing MRI images, neural network classification algorithm was the best. Lazy-IBk did very well and came in second. Na¨ıve Bayes and J48 decision tree came in last. Much higher accuracy can be achieved by gaining a better dataset with high-resolution images taken directly from the MRI scanner. Moreover, classifier boosting techniques can be used to raise the accuracy even higher and reach the level that will allow this tool to be a significant asset to any medical facility dealing with a brain tumor.

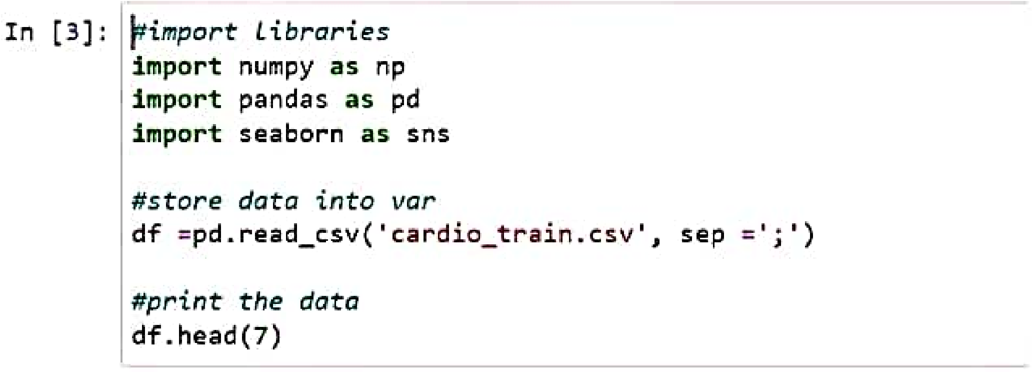
Reference

[1] N. N. Gopal and M. Karnan. Diagnose brain tumor through MRI using image processing clustering algorithms such as Fuzzy C Means along with intelligent optimization techniques. In IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pages 1–4, Dec 2010.

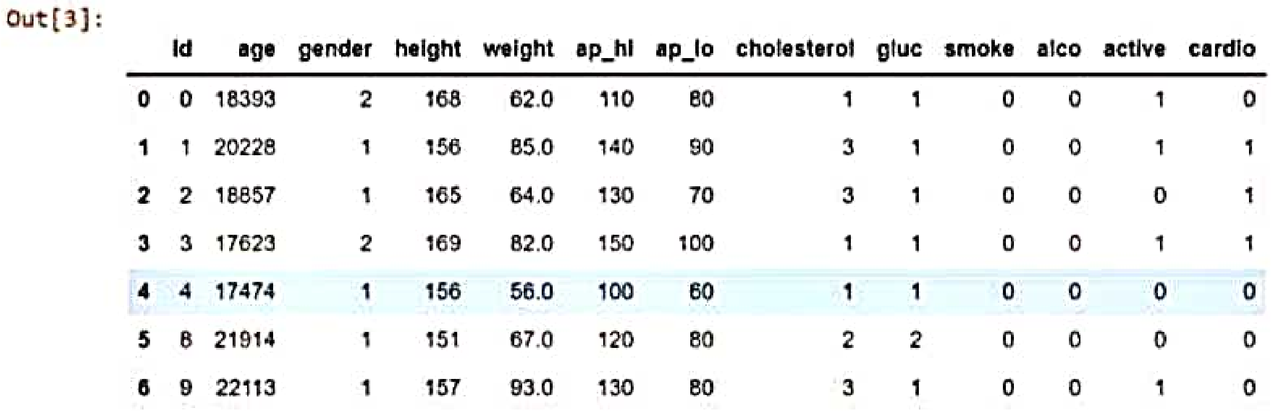
[2] H. Najadat, Y. Jaffal, O. Darwish, and N. Yasser. A classifier to detect abnormality in CT brain images. In The 2011 IAENG International Conference on Data Mining and Applications, pages 374–377, Mar 2011. [3] M. F. Othman and M. A. M. Basri. Probabilistic neural network for brain tumor classification. In Second International Conference on Intelligent Systems, Modelling and Simulation (ISMS), pages 136–138, Jan 2011.

[4] D. F. Specht. Probabilistic neural networks. Neural

SOURCE CODE

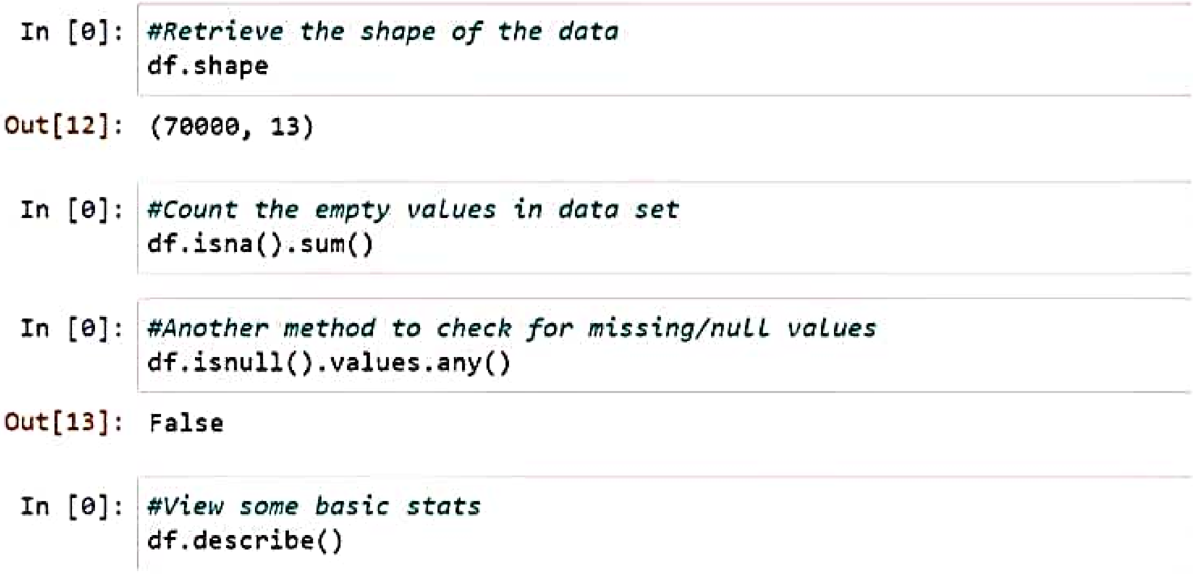


```
In [3]: #import libraries
        import numpy as np
        import pandas as pd
        import seaborn as sns

        #store data into var
        df =pd.read_csv('cardio_train.csv', sep =';')

        #print the data
        df.head(7)
```

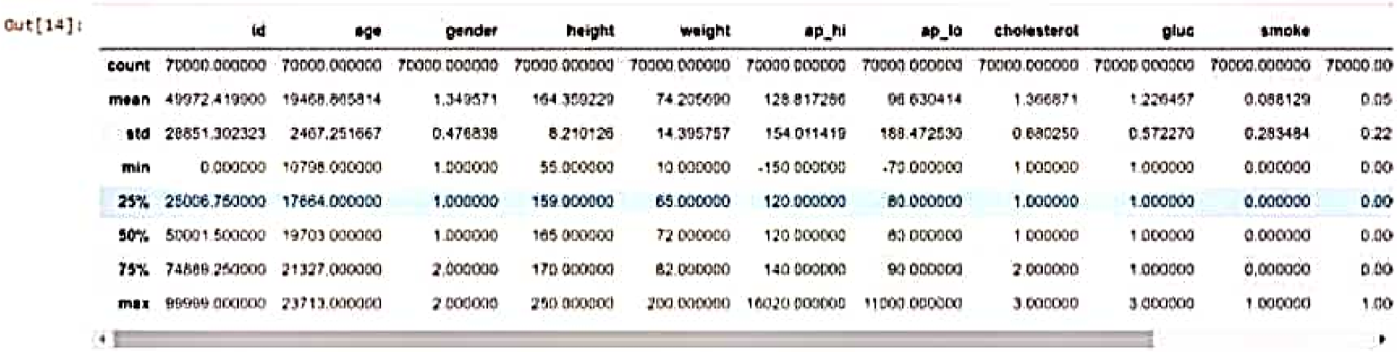


Out[3]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0
5	8	21914	1	151	67.0	120	80	2	2	0	0	0	0
6	9	22113	1	157	93.0	130	80	3	1	0	0	1	0

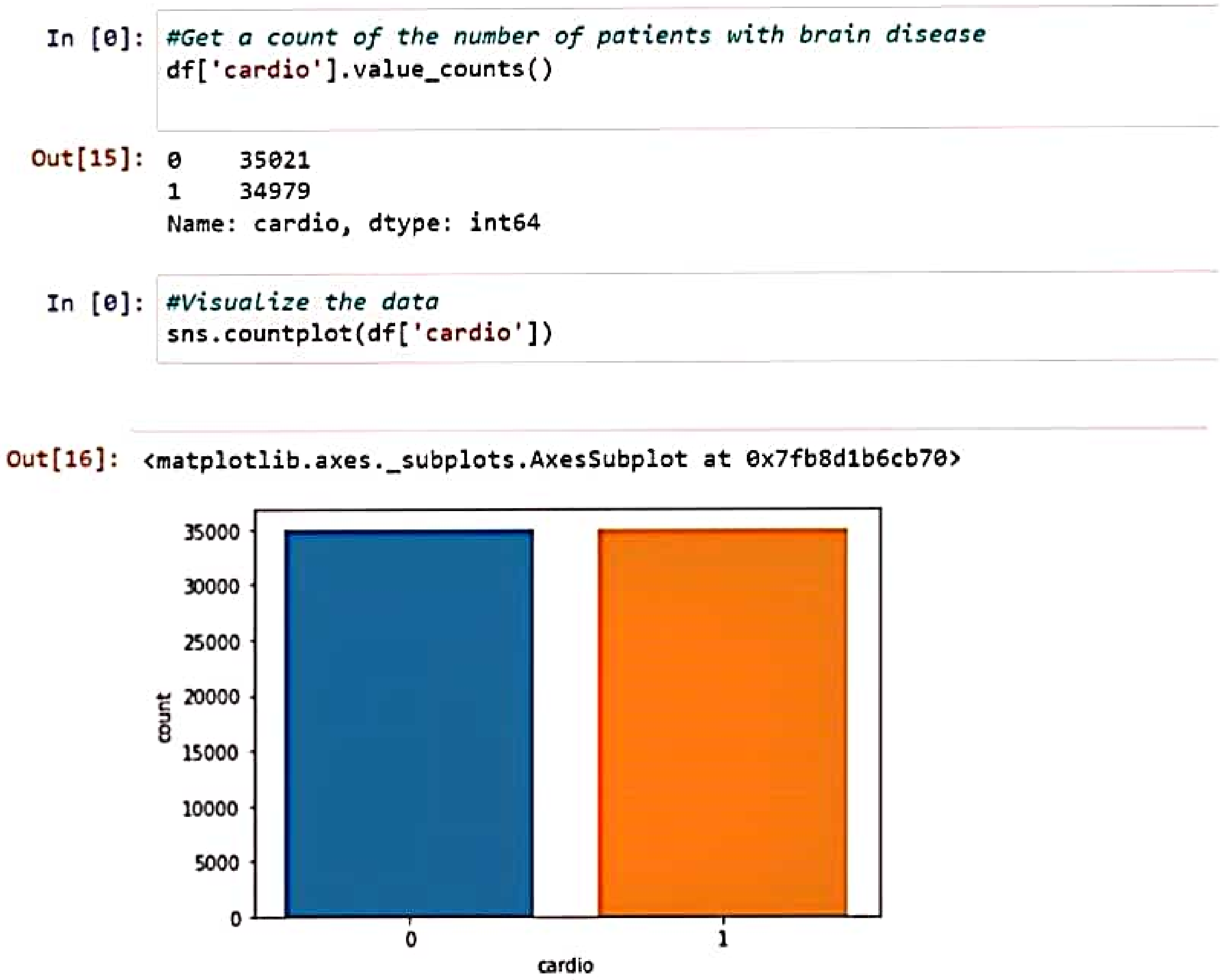


```
In [0]: #Retrieve the shape of the data
        df.shape

Out[12]: (70000, 13)

In [0]: #Count the empty values in data set
        df.isna().sum()

In [0]: #Another method to check for missing/null values
        df.isnull().values.any()

Out[13]: False

In [0]: #View some basic stats
        df.describe()
```

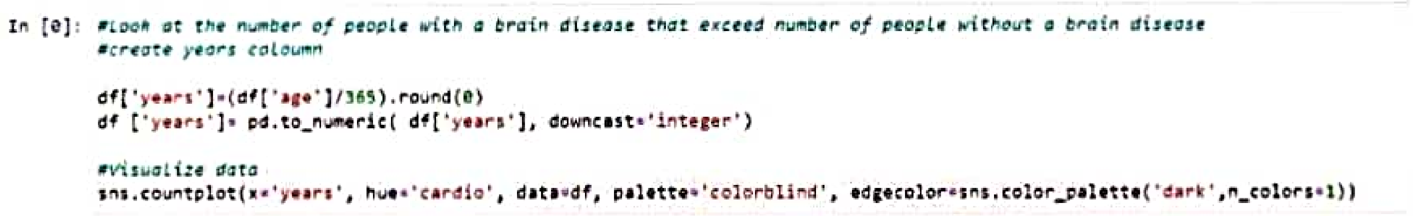


Out[14]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	
count	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.00
mean	49972.419900	19468.865814	1.349571	164.359229	74.205690	128.817286	96.630414	1.366871	1.226457	0.088129	0.05
std	28851.302323	2467.251667	0.476838	8.210126	14.395757	154.011419	188.472530	0.680250	0.572270	0.283484	0.22
min	0.000000	10798.000000	1.000000	55.000000	10.000000	-150.000000	-70.000000	1.000000	1.000000	0.000000	0.00
25%	25006.750000	17664.000000	1.000000	159.000000	65.000000	120.000000	80.000000	1.000000	1.000000	0.000000	0.00
50%	50001.500000	19703.000000	1.000000	165.000000	72.000000	120.000000	80.000000	1.000000	1.000000	0.000000	0.00
75%	74889.250000	21327.000000	2.000000	170.000000	82.000000	140.000000	90.000000	2.000000	1.000000	0.000000	0.00
max	99999.000000	23713.000000	2.000000	250.000000	200.000000	16020.000000	11000.000000	3.000000	3.000000	1.000000	1.00

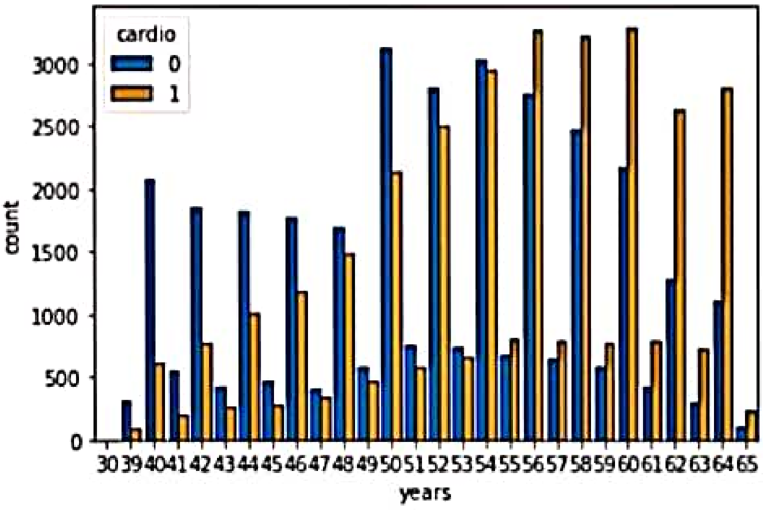


```
In [0]: #Get a count of the number of patients with brain disease
        df['cardio'].value_counts()

Out[15]: 0    35021
         1    34979
         Name: cardio, dtype: int64

In [0]: #Visualize the data
        sns.countplot(df['cardio'])

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb8d1b6cb70>
```



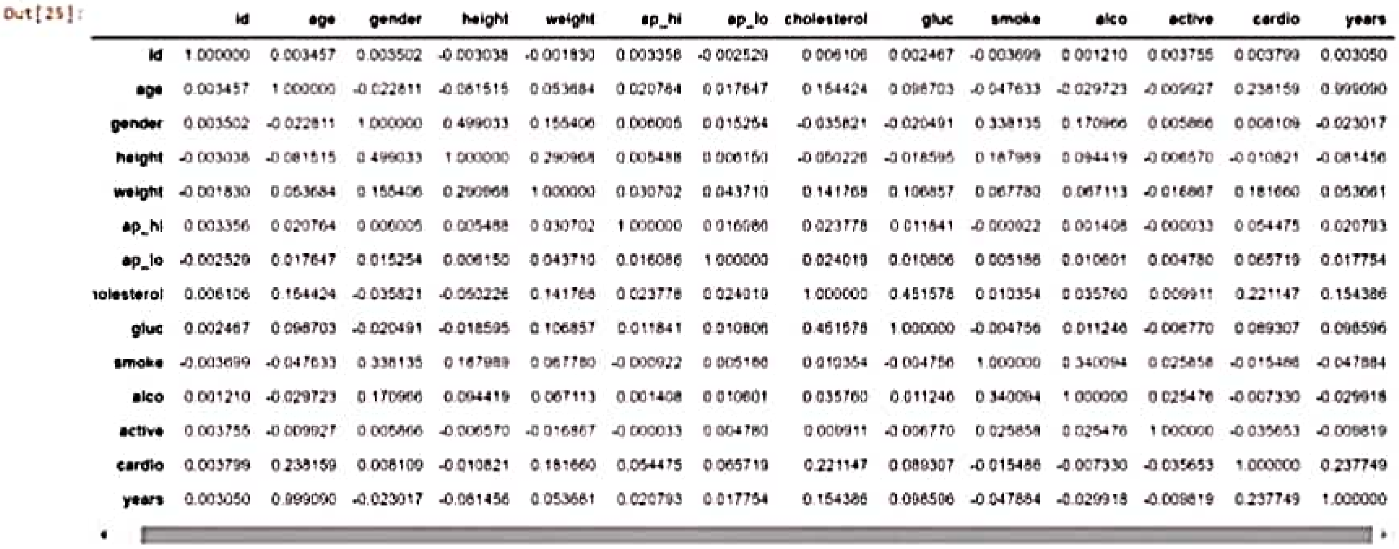
```
In [0]: #Look at the number of people with a brain disease that exceed number of people without a brain disease
        #create years coloumn

        df['years']=(df['age']/365).round(0)
        df ['years']= pd.to_numeric( df['years'], downcast='integer')

        #Visualize data
        sns.countplot(x='years', hue='cardio', data=df, palette='colorblind', edgecolor=sns.color_palette('dark',n_colors=1))
```

```
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb8ce434c88>
```



```
In [0]: # Get the correlation of the coloumns
        df.corr()
```

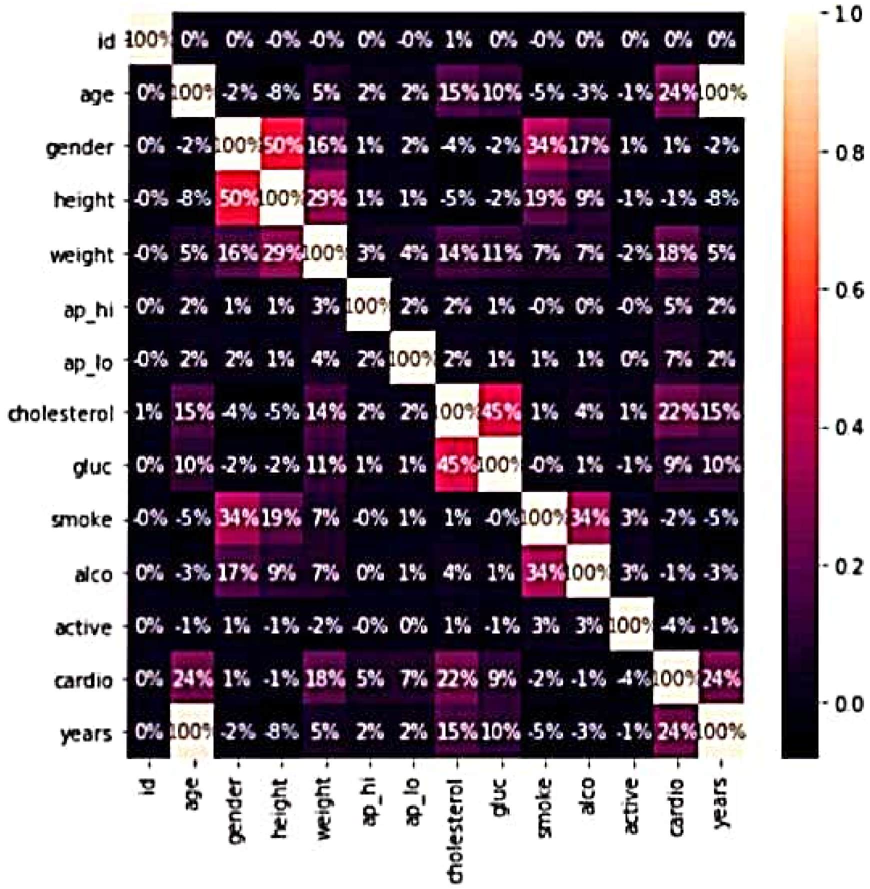
Out[25]:



	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio	years
id	1.000000	0.003457	0.003502	-0.003038	-0.001830	0.003356	-0.002529	0.006106	0.002467	-0.003699	0.001210	0.003755	0.003799	0.003050
age	0.003457	1.000000	-0.022811	-0.081515	0.053684	0.020764	0.017647	0.154424	0.098703	-0.047633	-0.029723	-0.009927	0.238159	0.999090
gender	0.003502	-0.022811	1.000000	0.499033	0.155406	0.006005	0.015254	-0.035821	-0.020491	0.338135	0.170966	0.005866	0.008109	-0.023017
height	-0.003038	-0.081515	0.499033	1.000000	0.290968	0.005488	0.006150	-0.050226	-0.018595	0.187989	0.094419	-0.006570	-0.010821	-0.081456
weight	-0.001830	0.053684	0.155406	0.290968	1.000000	0.030702	0.043710	0.141768	0.106857	0.067780	0.067113	-0.016867	0.181660	0.053661
ap_hi	0.003356	0.020764	0.006005	0.005488	0.030702	1.000000	0.016086	0.023778	0.011841	-0.000922	0.001408	-0.000033	0.054475	0.020793
ap_lo	-0.002529	0.017647	0.015254	0.006150	0.043710	0.016086	1.000000	0.024019	0.010806	0.005186	0.010601	0.004780	0.065719	0.017754
holesterol	0.006106	0.154424	-0.035821	-0.050226	0.141768	0.023778	0.024019	1.000000	0.451578	0.010354	0.035760	0.009911	0.221147	0.154386
gluc	0.002467	0.098703	-0.020491	-0.018595	0.106857	0.011841	0.010806	0.451578	1.000000	-0.004756	0.011246	-0.006770	0.089307	0.098596
smoke	-0.003699	-0.047633	0.338135	0.187989	0.067780	-0.000922	0.005186	0.010354	-0.004756	1.000000	0.340094	0.025858	-0.015486	-0.047884
alco	0.001210	-0.029723	0.170966	0.094419	0.067113	0.001408	0.010601	0.035760	0.011246	0.340094	1.000000	0.025476	-0.007330	-0.029918
active	0.003755	-0.009927	0.005866	-0.006570	-0.016867	-0.000033	0.004780	0.009911	-0.006770	0.025858	0.025476	1.000000	-0.035653	-0.009819
cardio	0.003799	0.238159	0.008109	-0.010821	0.181660	0.054475	0.065719	0.221147	0.089307	-0.015486	-0.007330	-0.035653	1.000000	0.237749
years	0.003050	0.999090	-0.023017	-0.081456	0.053661	0.020793	0.017754	0.154386	0.098506	-0.047884	-0.029918	-0.009819	0.237749	1.000000

```
In [0]: #Visualize data to gain insight on correlations and trends
        import matplotlib.pyplot as plt

        #Output heatmap to see the dataset
        plt.figure(figsize=(7,7))
        sns.heatmap(df.corr(), annot=True, fmt='.0%')
```

```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb8ce42d470>
```



	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio	years
id	100%	0%	0%	-0%	-0%	0%	-0%	1%	0%	-0%	0%	0%	0%	0%
age	0%	100%	-2%	-8%	5%	2%	2%	15%	10%	-5%	-3%	-1%	24%	100%
gender	0%	-2%	100%	50%	16%	1%	2%	-4%	-2%	34%	17%	1%	1%	-2%
height	-0%	-8%	50%	100%	29%	1%	1%	-5%	-2%	19%	9%	-1%	-1%	-8%
weight	-0%	5%	16%	29%	100%	3%	4%	14%	11%	7%	7%	-2%	18%	5%
ap_hi	0%	2%	1%	1%	3%	100%	2%	2%	1%	-0%	0%	-0%	5%	2%
ap_lo	-0%	2%	2%	1%	4%	2%	100%	2%	1%	1%	1%	0%	7%	2%
cholesterol	1%	15%	-4%	-5%	14%	2%	2%	100%	45%	1%	4%	1%	22%	15%
gluc	0%	10%	-2%	-2%	11%	1%	1%	45%	100%	-0%	1%	-1%	9%	10%
smoke	-0%	-5%	34%	19%	7%	-0%	1%	1%	-0%	100%	34%	3%	-2%	-5%
alco	0%	-3%	17%	9%	7%	0%	1%	4%	1%	34%	100%	3%	-1%	-3%
active	0%	-1%	1%	-1%	-2%	-0%	0%	1%	-1%	3%	3%	100%	-4%	-1%
cardio	0%	24%	1%	-1%	18%	5%	7%	22%	9%	-2%	-1%	-4%	100%	24%
years	0%	100%	-2%	-8%	5%	2%	2%	15%	10%	-5%	-3%	-1%	24%	100%

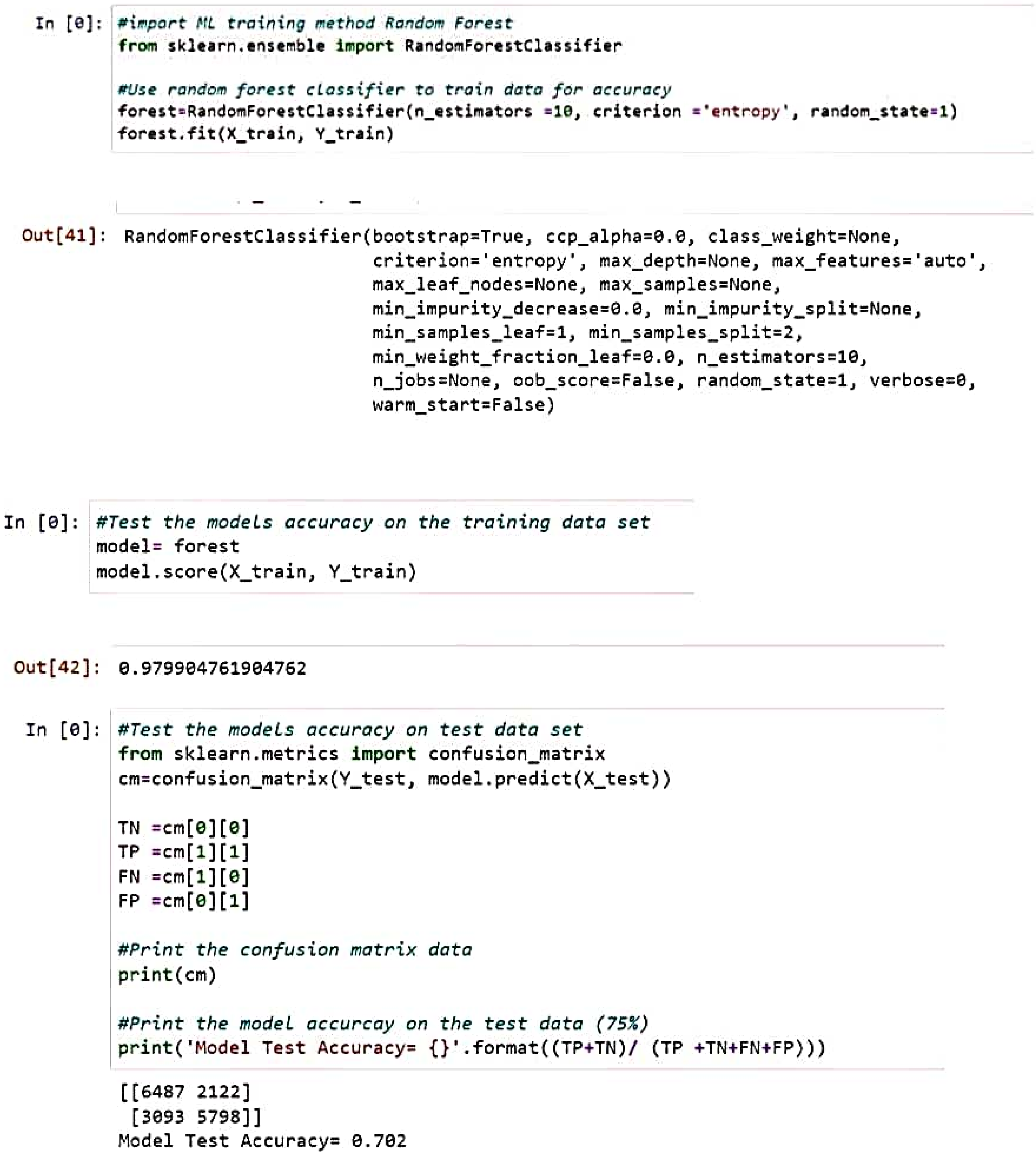
```
In [0]: # Remove or drop the years coloumn
        df=df.drop('years', axis=1)
```

```
In [0]: #Remove the id coloumn as it will deter accuracy due when training model
        df=df.drop('id', axis=1)
```

```
In [0]: #Split the data into feature data and target data
        X= df.iloc[:, :-1].values
        Y= df.iloc[:,-1]. values
```

```
In [0]: #Split the data again, into 75% trainig and 25% testing
        from sklearn.model_selection import train_test_split

        X_train, X_test, Y_train, Y_test=train_test_split(X,Y, test_size=0.25,random_state=1)
```

```
In [0]: #Feature scaling
        #Scale the values in data to be values between 0 and 1 inclusive
        from sklearn.preprocessing import StandardScaler

        sc =StandardScaler()
        X_train = sc.fit_transform(X_train)
        X_test =sc.transform(X_test)
```



```
In [0]: #import ML training method Random Forest
        from sklearn.ensemble import RandomForestClassifier

        #Use random forest classifier to train data for accuracy
        forest=RandomForestClassifier(n_estimators =10, criterion ='entropy', random_state=1)
        forest.fit(X_train, Y_train)
```

```
Out[41]: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                                criterion='entropy', max_depth=None, max_features='auto',
                                max_leaf_nodes=None, max_samples=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=10,
                                n_jobs=None, oob_score=False, random_state=1, verbose=0,
                                warm_start=False)
```

```
In [0]: #Test the models accuracy on the training data set
        model= forest
        model.score(X_train, Y_train)
```

```
Out[42]: 0.979904761904762
```

```
In [0]: #Test the models accuracy on test data set
        from sklearn.metrics import confusion_matrix
        cm=confusion_matrix(Y_test, model.predict(X_test))

        TN =cm[0][0]
        TP =cm[1][1]
        FN =cm[1][0]
        FP =cm[0][1]

        #Print the confusion matrix data
        print(cm)

        #Print the model accurcay on the test data (75%)
        print('Model Test Accuracy= {}'.format((TP+TN)/ (TP +TN+FN+FP)))
```

```
[[6487 2122]
 [3093 5798]]
Model Test Accuracy= 0.702
```