BRAIN TUMOR DETECTION USING MACHINE LEARNING TECHNIQUE

SWATHI DAMODARAN*1, SUDHARSHANA MAHALAKSHMI2, Mr. VENKATESH GAJENDRAN3

*1,2UG Scholars,Department of Electronics and Communication Engineering, Panimalar Engineering College, Chennai,India

3Assistant.Professor, Department of Electronics and Communication Engineering, Panimalar Engineering College ,Chennai,India

Abstract

Brain Tumor which is otherwise known as brain cancer. Growth of abnormal cell in brain will leads to brain cancer. The common tricky method to detect the cancer is MRI because it does not involves any radiations . It gives clear detailed images of the inner structure of our body than any other techniques. Brain tumor is detected by using machine learning algorithm. proposed work presents the comprehensive review of method and techniques that is used to detect the brain tumor. Brain tumor is detected by using Support Vector Machine (SVM) classification and SVM algorithm. SVM algorithm gives accurate results. Metastasis, astrocytoma are some types of the tumor. Different types of tumor are classified by using Multi-SVM. MRI images are used in proposed work because it is non-ionised and it has good spatial & temporal resolution.

Keywords: *Astrocytoma, Brain Tumor, Machine learning algorithm, Metastasis, MRI, SVM*

1.Introduction

The detection of brain cancer is very important because the survival rate of patients who are affected by brain tumor is low(12.8%). In human body , brain is the very important organ because it command and control every functions of our body . The brain tumor is caused by the growth of abnormal cells in brain. In the year of 2021 about 3,540 children at the age of 15 are diagonized due to brain tumor. One of the most crucial task in brain tumour detection system is the isolation of abnormal tissue from normal brain tissue. Gliomas are tumor that affects central nervous system. Prognosis of subjects with gliomas are low. There are many methods to detect brain tumor like CT, MRI, X-Rays the best one among this is MRI. The result of the detection performed in this paper is to find whether the brain is affected or not.

Classification and regression analysis are made by using Support Vector Machine (SVM). In high or dimensional space, hyper planes are constructed by SVM. The generalization error of the classifier in Support Vector Machine (SVM) is low. SVM is used not only in classification but also in regression problem. This supervised learning algorithm (SVM) is more popular. It is capable for separating n-dimensional space into classes. Creating the best line or decision boundary is the goal of SVM algorithm. Support vectors are the extreme point/vector which is choosen by the SVM in creating hyperplane. SVM gives the accurate result for the problem.

TYPES OF SVM:

- Linear
- Non-linear

2.Literature survey

Afshar, P.[1] classified the extracted feature using artificial neural network (ANN). Gray-Level Co-occurrence Matrix (GLCM) technique is used to extract a feature . The algorithm used is artificial intelligence algorithms. Different aspects, element of learning were motivated by ANN. ANN is inspired by biological nervous system and it is a mathematical problem.

W. S. Emmanuelet al [2] used ANN technique for tumor stage classification. Filter applied for pre-processing is weiner2 and median2 . TKFCM algorithm is used for segmentation. Brain is classified as normal and tumor by SVM classifier. MRI images are used as dataset. Normal brain, benign tumor , malignant tumor are the data used for each MRI image. Adjustment , adaptive threshold and histogram are the pre – processing techniques used in this proposed work . In this work both SVM, ANN techniques are used .

Anaraki, Aet al [4] detected brain tumor using deep learning technique . MRI images are used as dataset . Classification is done by using Artificial Neural Network (ANN), Convolution Neural Network (CNN) . 2065 MRI images are used in proposed work for detection. For final prediction (2 D, 2.5 D, 3 D) different dimensional data are applied. In image analysing voxel-wise classification and efficient patch-wise evaluation are used through CNN. Segmentation method is based on learning techniques.

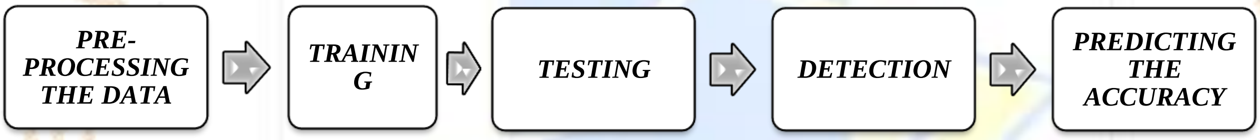
Bunevicius, Aet al [5] hybrid model is made for CNN and SVM model . Benign and malignant tumor are classified from public dataset. Segmentation is based on threshold. The technique used for categorization of brain is Conventional Machine Learning (ML). More effective techniques for classification is provided from hybrid model.

Table 1: comparison of literature survey

AUTHOR	TOPIC	METHODS	TECHNIQUES
Afshar, P. et al[1]	ANN approach based on back propagation network and probabilistic neural network to classify brain cancer	Artificial intelligence	Gray-level co-occurrence matrix
W. S. Emmanuel et al [2]	Classification of tumor and its stage in MRI brain using support vector machine and artificial neural network	Deep learning and Machine learning	Support vector machine and Artificial neural network
Anaraki, Aet al [4]	Brain tumor detection from MRI images using deep learning techniques	Deep learning	Artificial neural network and Convolution neural network
Bunevicius, A[5]	A hybrid CNN-SVM threshold segmentation approach for tumor detection of MRI brain	Deep learning, Machine learning	Conventional machine learning, CNN, SVM

3.Proposed Work:

BLOCK DIAGRAM:



In this proposed work, brain tumor detection and classification was made by using support vector machine algorithm. Dependencies are installed to avoid runtime error. CSV file contains the dataset required for detection. Dataset is taken from github.com .

NUMPY:numpy is a python library , used for integrating a strong array object, mathematical, and statistical tool with other language's code.
PANDAS: It provides fast , flexible and expensive data structure design to form working with labelled data and it is a python package .
MATPLOTLIB: It is a plotting library for python . matplotlib.pyplot is used in this proposed work.
SEABORN: In pandas library it works easily with data frame . it is also used for data visualization and Exploratory data analysis.
LOGICAL REGRESSION METHOD: used to predict dependent target variable.
DECISION TREE METHOD: ID3 (by Quinlan) is the basic algorithm used in decision tree method.
RANDOM FOREST CLASSIFIER METHOD: it improves the predictive accuracy.
SUPPORT VECTOR CLASSIFIER METHOD: classify the objects into different methods.

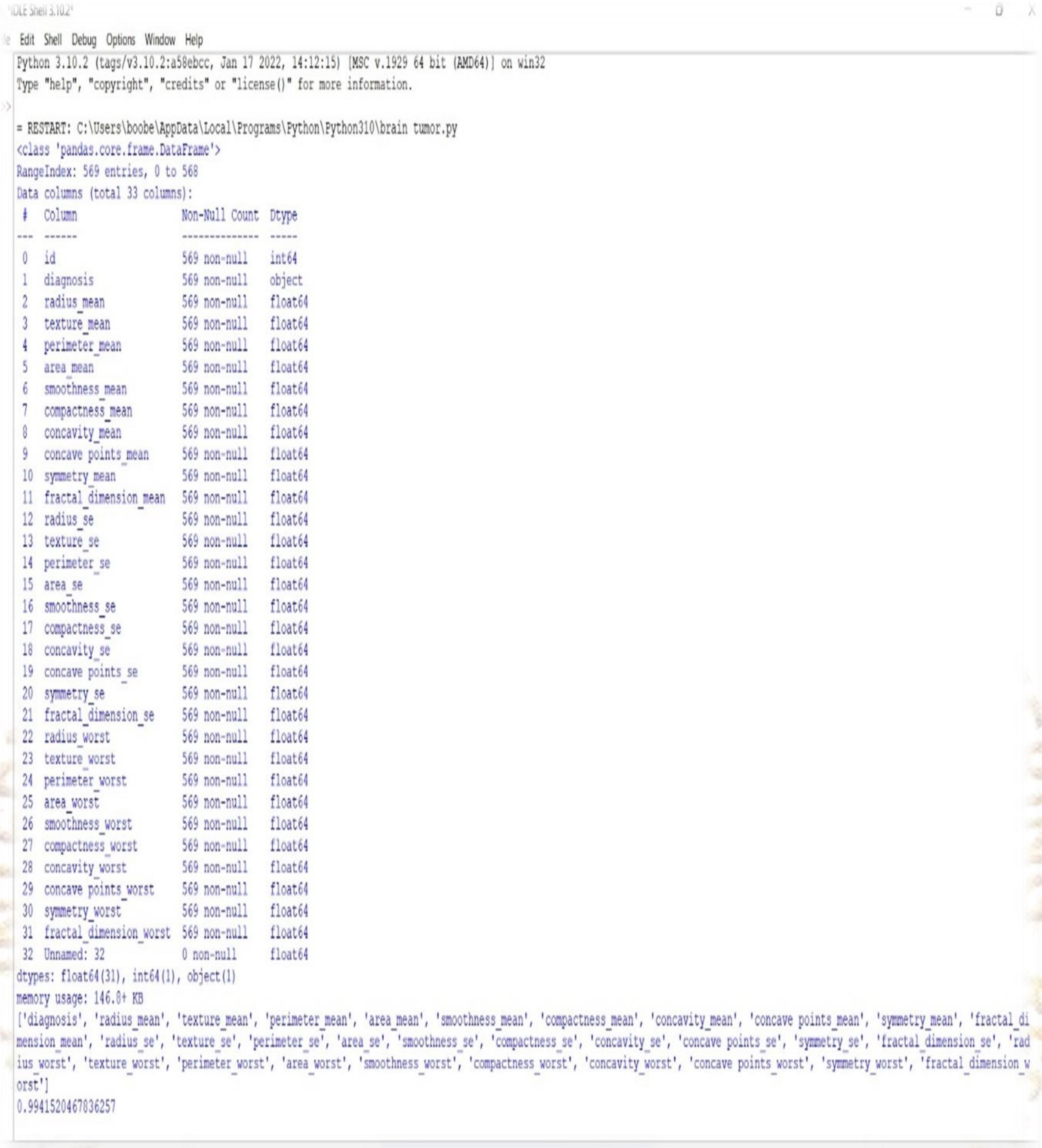
In this project the malignant and begin stage of brain tumor is detected. Features are categorized as mean, se and worse. The categorized features are radius, texture, smoothness, compactness, concavity, area, perimeter, concave points ,symmentry and fractal dimension. Range index are 569 entries, 0 to 568. 33 data columns are used Datatypes used are object-1, integer-1 and float-64. Data was pre- processed. Diagonisis field verifies the malignant and begin values. The graph is plotted between malignant and begin values to count by using count() method. Width of the graph = 0.1, height of the graph = 0.5. The graphical representation for features are made by using seaborn and matplotlib tools. Support vector machine is used for training and it is also used for classification and detection. Machine learning methods are used for testing the data set.

4.Experimental Results:

Table 2: comparison of algorithm

ALGORITHM	ACCURACY
Logical Regression Method	0.994152
Decision Tree Method	0.982456
Random Forest Classifier Method	0.964912
Support Vector Classifier Method	0.989473

A. Output For Logical Regression Method

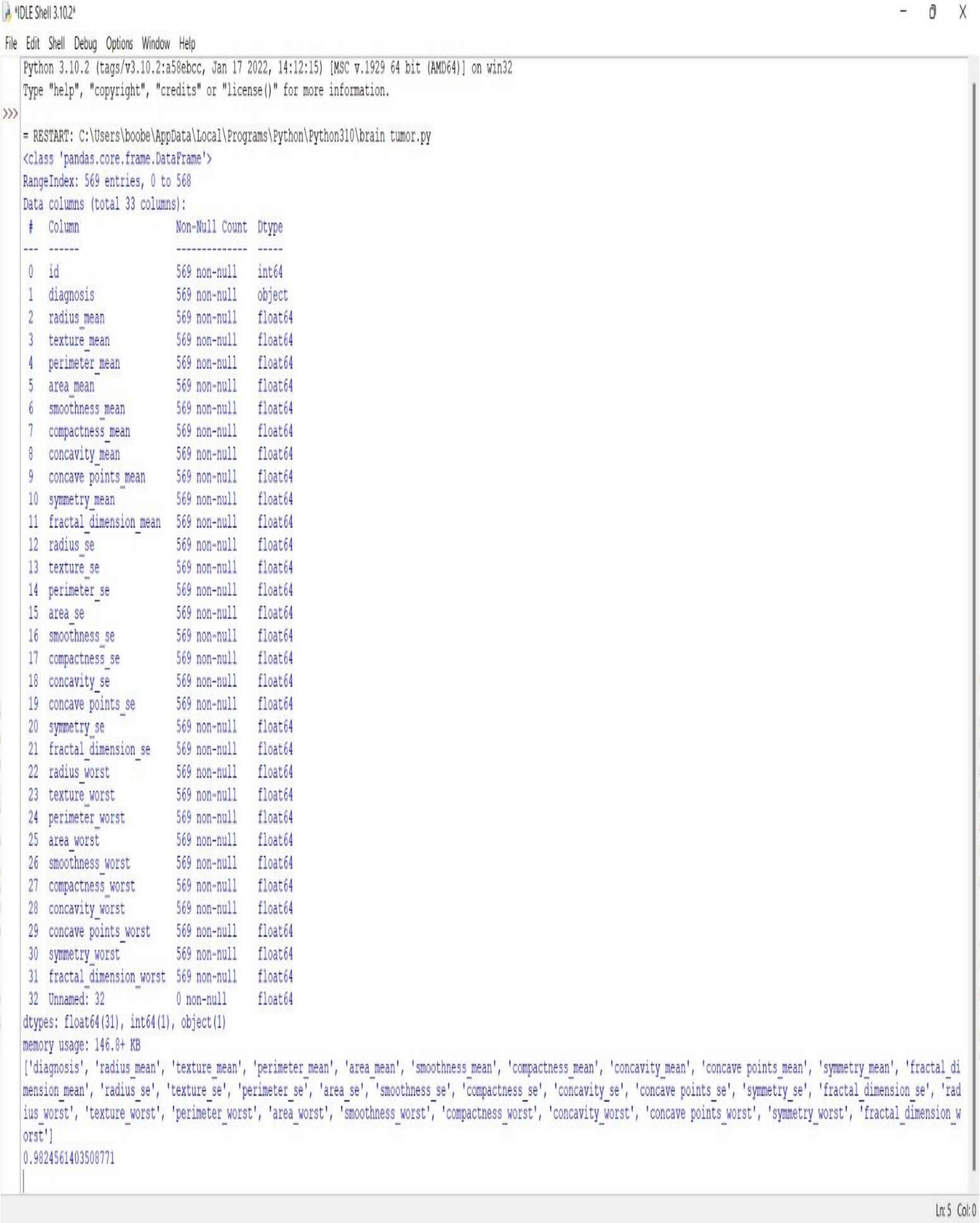


```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:\Users\boobe\AppData\Local\Programs\Python\Python310\brain tumor.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_di
mension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'rad
ius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_w
orst']
0.9941520467836257
```

Figure 1: Output

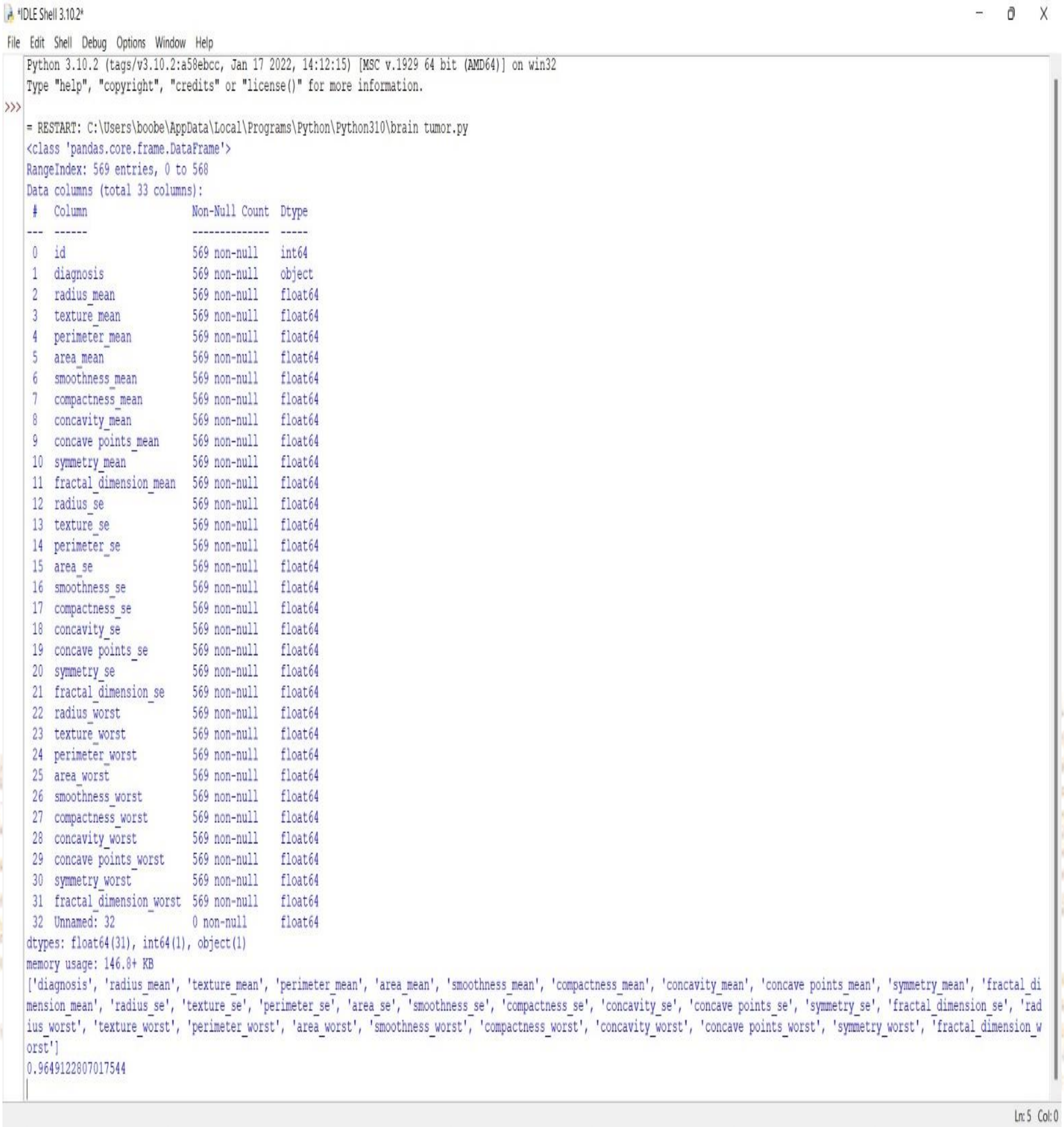
B. Output For Decision Tree Method



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\boobe\AppData\Local\Programs\Python\Python310\brain tumor.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_di
mension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'rad
ius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_w
orst']
0.9824561403508771
```

Figure 2 : Output

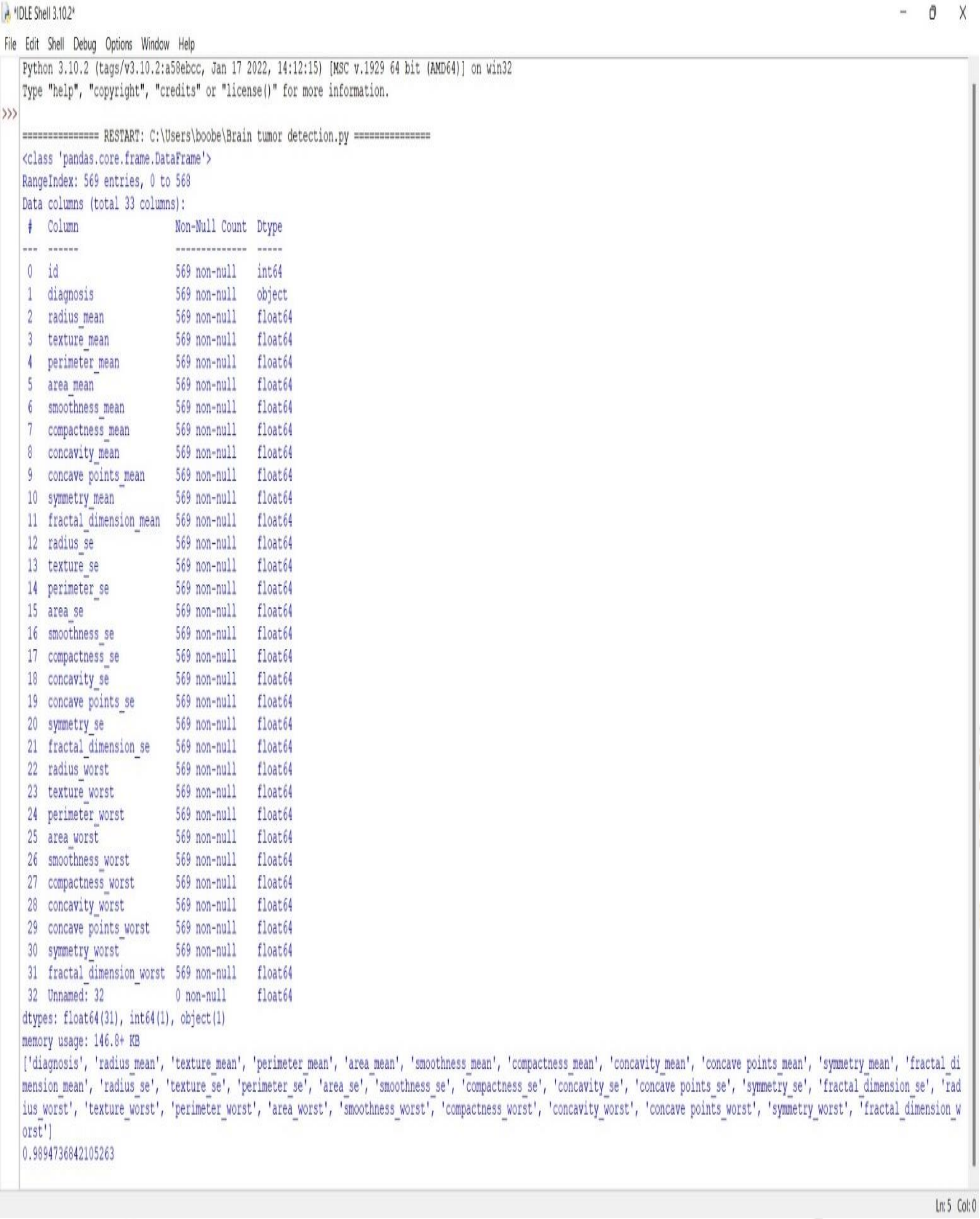
C. **Output For Random Forest Classifier Method**:



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\boobe\AppData\Local\Programs\Python\Python310\brain tumor.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_di
mension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'rad
ius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_w
orst']
0.9649122807017544
```

Figure 3 : Output

D. Output For Support Vector Classifier Method



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\boobe\Brain tumor detection.py ==============
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_di
mension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'rad
ius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_w
orst']
0.9894736842105263
```

Figure 4: Output

E. Graph Between Malignant And Begin Values To Count:

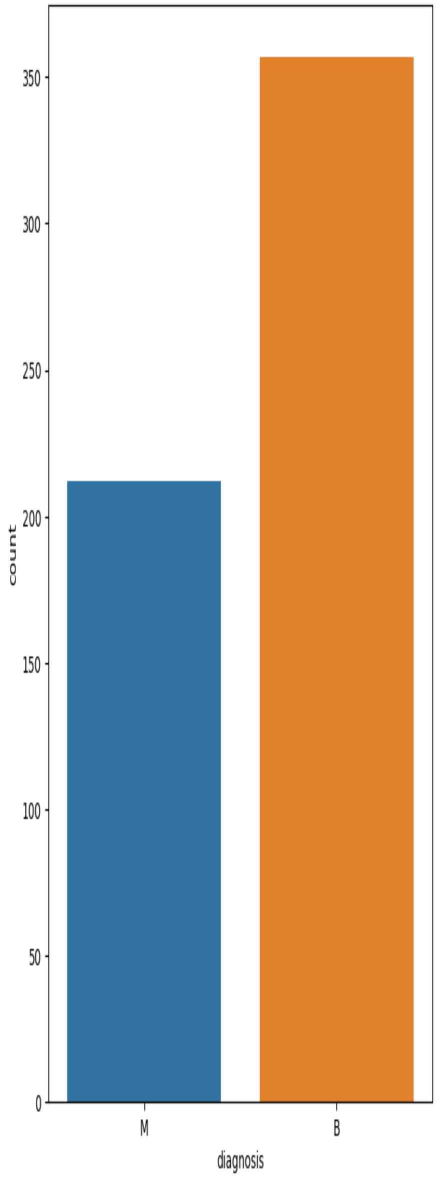


Figure 5 : Output

F. Graphical Representation For Feature

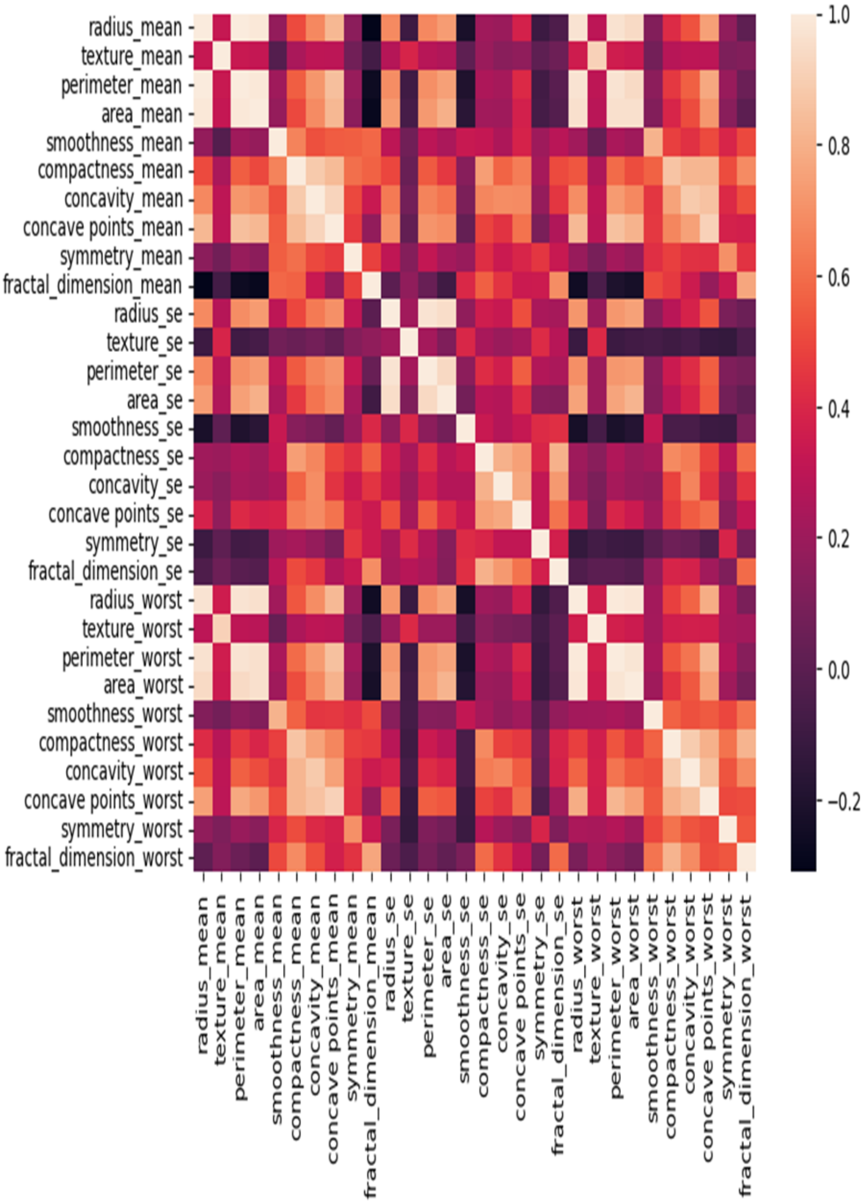


Figure 6 : Output

6.Conclusion

In this project the malignant and beginning stage of brain tumor was detected. The detection was made more efficiently and accurately by using support vector machine algorithm and machine learning methods. Many techniques was analyzed and found efficient and accurate result providing technique. Computational time was less. The accuracy obtained from proposed work is 0.994% (logical regression method), 0.989% (support vector classifier method), 0.982% (decision tree method), 0.964% (random forest classifier method). From this it was concluded that logical regression and support vector classifier are the more accurate result providing methods. Changing the dataset will also give the accurate results. Hence detecting the brain cancer in beginning stage provides the possibilities of treating the cancer and increasing the rate of survival.

References

[1] Afshar, P., Mohammadi, A., and Plataniotis, K. N. (2018). “Brain tumor type classification via capsule networks,” in in 2018 25th IEEE International Conference on Image Processing (ICIP), (New Jersey: IEEE), 3129–3133. doi: 10.1109/ICIP.2018.8451379

[2] A. Deepa and W. S. Emmanuel, "An efficient detection of brain tumor using fused feature adaptive firefly backpropagation neural network", *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 11799-11814, 2019.

[3] S. Kumar, A. Negi and J. Singh, "Semantic segmentation using deep learning for brain tumor MRI via fully convolution neural networks", *Proc. Inf. Commun. Technol. Intell. Syst.*, pp. 11-19, 2019.

[4] Anaraki, A. K., Ayati, M., and Kazemi, F. (2019). Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. Biocybern. Biomed. Eng. 39, 63–74. doi: 10.1016/j.bbe.2018.10.004

[5] Bunevicius, A., Schregel, K., Sinkus, R., Golby, A., and Patz, S. (2020). Review: MR elastography of brain tumors. NeuroImage Clin. 25:102109. doi: 10.1016/j.nicl.2019.102109

[6] G. S. Tandel et al., "A review on a deep learning perspective in brain cancer classification", *Cancers*, vol. 11, no. 1, pp. 1-32, 2019.