**1. INTRODUCTION**

Breast Cancer is the most often identified cancer among women and major reason for increasing mortality rate among women. As the diagnosis of this disease manually takes long hours and the lesser availability of systems, there is a need to develop the automatic diagnosis system for early detection of cancer. While the cure for this cancer is now available in almost all first world and some of the third world nations, the main dilemma takes place when the cancer cannot be correctly identified at the very initial stages. Due to varying nature of breast cancers symptoms, patients are often subjected to a barrage of tests, including but not limited to mammography, ultrasound and biopsy, to check their likelihoods of being diagnosed with breast cancer. Biopsy is the most indicative among these procedures, which involves extraction of sample cells or tissues for examination. The sample of cells is obtained from a breast fine needle aspiration (FNA) procedure and then sent to a pathology laboratory to be examined under a microscope. Numerical features, such as radius, texture, perimeter and area, can be measured from microscopic images. Data, later on, obtained from FNA are analyzed in combination with various imaging data to predict probability of the patient having malignant breast cancer tumor. An automated system here would be hugely beneficial in this scenario. Data mining techniques contribute a lot in the development of such system. For the classification of benign and malignant tumor we have used classification techniques of machine learning in which the machine is learned from the past data and can predict the category of new input.

* 1. **Scope**

The number and the size of databases recording medical data are increasing rapidly. Medical data, produced from measurements, examinations, prescriptions, etc., are stored in different databases on a continuous basis. This enormous amount of data exceeds the ability of traditional methods to analyze and search for interesting patterns and information that is hidden in them. Therefore, new techniques and tools for discovering useful information in these data depositories are becoming more demanding. Analyzing these data with new analytical methods in order to find interesting patterns and hidden knowledge is the first step in extending the traditional function of these data sources.

With the evolution of different machine learning techniques it is easier to integrate the techniques with the data set and predict the presence of breast cancer in women. This helps in faster diagnosis so that the patient can be quick in getting medical attention. An automated system here would be hugely beneficial in this scenario. Breast cancer risk prediction models used in clinical practice have low discriminatory accuracy (0.53–0.64). Machine learning (ML) offers an alternative approach to standard prediction modeling that may address current limitations and improve accuracy of those tools. The diagnosis of this disease manually takes long hours and the lesser availability of systems, there is a need to develop the automatic diagnosis system for early detection of cancer.

* 1. **Existing System**

In the present system, this cancer is detected using mammograms. The patient has to go a diagnostic centre and undergo the various tests. The doctor analysis the report and provides a conclusion. This is a time taking process. However, mammogram images sometimes have a risk of false detection that may endanger the patient's health. If the tools and mammograms are unavailable then it increases the risk. The patient can also experience pain and will be exposed to radiation while undergoing other tests for confirmation and accuracy.

The drawbacks of the existing system are as follows:

* it is a time taking process
* can’t be detected early
* risk of false detection.
  1. **Proposed System**

The proposed system works include the integration of mixed data and machine learning technique i.e. Random Forest Classifier. We use the data set from UCI Machine Repository (Breast Cancer Wisconsin (Original)) in the proposed system. Using the random forest classifier, we will be able to identify the type of tumor i.e. whether it is malignant or benign tumor. The presence of malignant tumor refers to the presence of cancer cells and thus the cancer is detected.

The proposed system has been tested for several cases. In each case the accuracy measured is almost 94%. It helps in early detection and has greater accuracy values. It helps in overcoming the risk of false prediction. It will likely expedite the process and enhance the accuracy of the doctor’s predictions.

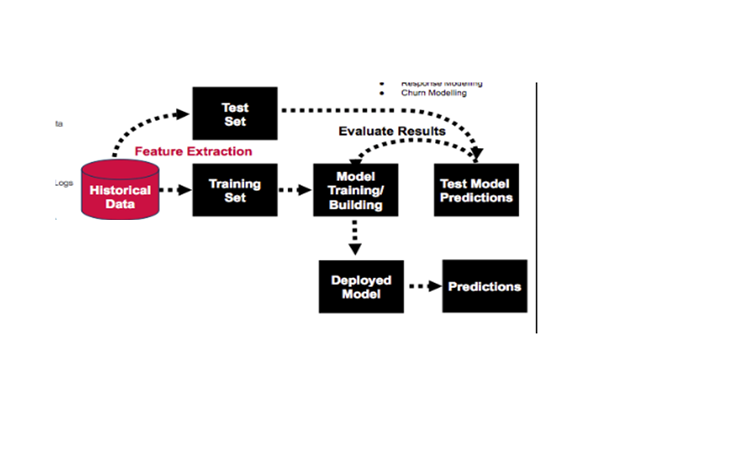


Fig 1.3: Proposed System

**2. SYSTEM ANALYSIS**

This System Analysis is closely related to [requirements analysis](http://en.wikipedia.org/wiki/Requirement_analysis). It is also "an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made."This step involves [breaking down](http://en.wikipedia.org/wiki/Work_breakdown_structure) the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

**2.1 Functional Requirement Specification**

The System after careful analysis has been identified to be present with the following modules.

* 1. **Administrator Module:**

Administrator Module is responsible for training data . Administrator is the one who add all the contour values to the excel sheet . Administrator collects all the training data set and test data .

* 1. **User Module :**

User Need to pass the test data to the model . User need to pass the contour values to the final Module , such that database Manager will predict the output.

* 1. **Database Manager Module :**

Data base Manager will predict output and based on training data and test data he will predict even accuracy.

**2.2 Performance Requirements**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

**2.3 Software Requirements**

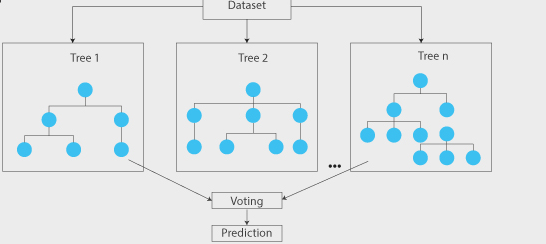
* OS – Windows,Linux
* Python-3.6
* Anaconda-python 3.6(optionally)

**2.4 Hardware Requirements**

* Ram-4GB
* CPU-Intel i3/i5/i7 recommended
* ROM-20GB HDD Free Space

**3. SYSTEM DESIGN**

**3.1 System Architecture**

****

**Fig 3.1 System Architecture**

**Random forests** or **random decision forests** are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks that operates by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of [over fitting](https://en.wikipedia.org/wiki/Overfitting) to their training set

Decision trees are a popular method for various machine learning tasks. Tree learning "come[s] closest to meeting the requirements for serving as an off-the-shelf procedure for data mining", say [Hastie](https://en.wikipedia.org/wiki/Trevor_Hastie) *et al.*, "because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable models. However, they are seldom accurate".

In particular, trees that are grown very deep tend to learn highly irregular patterns: they [overfit](https://en.wikipedia.org/wiki/Overfitting) their training sets, i.e. they have [low bias, but very high variance](https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff). Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

**Data set for Breast Cancer Prediction**

* The data set used for the training and testing the model is taken from UCI Machine Repository named as Breast Cancer Wisconsin.
* This consists of 18 features in total depending on the type of tumor.
* The tumors are classified into ‘B’ meaning benign and ‘M’ meaning Malignant.
* The above data set is trained and tested for accurate prediction
* The accuracy is also calculated for the details provided by the user.

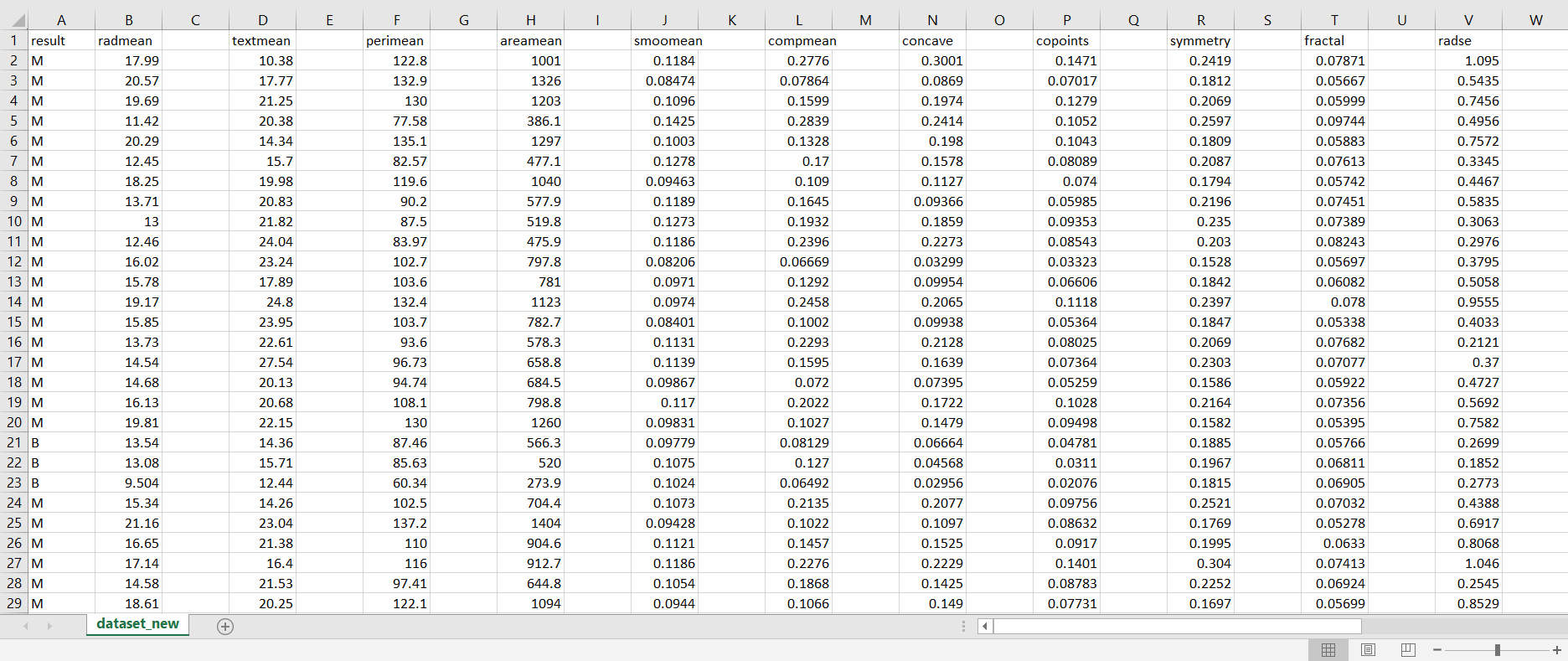


Fig 3.1.1: Data set containing the attributes

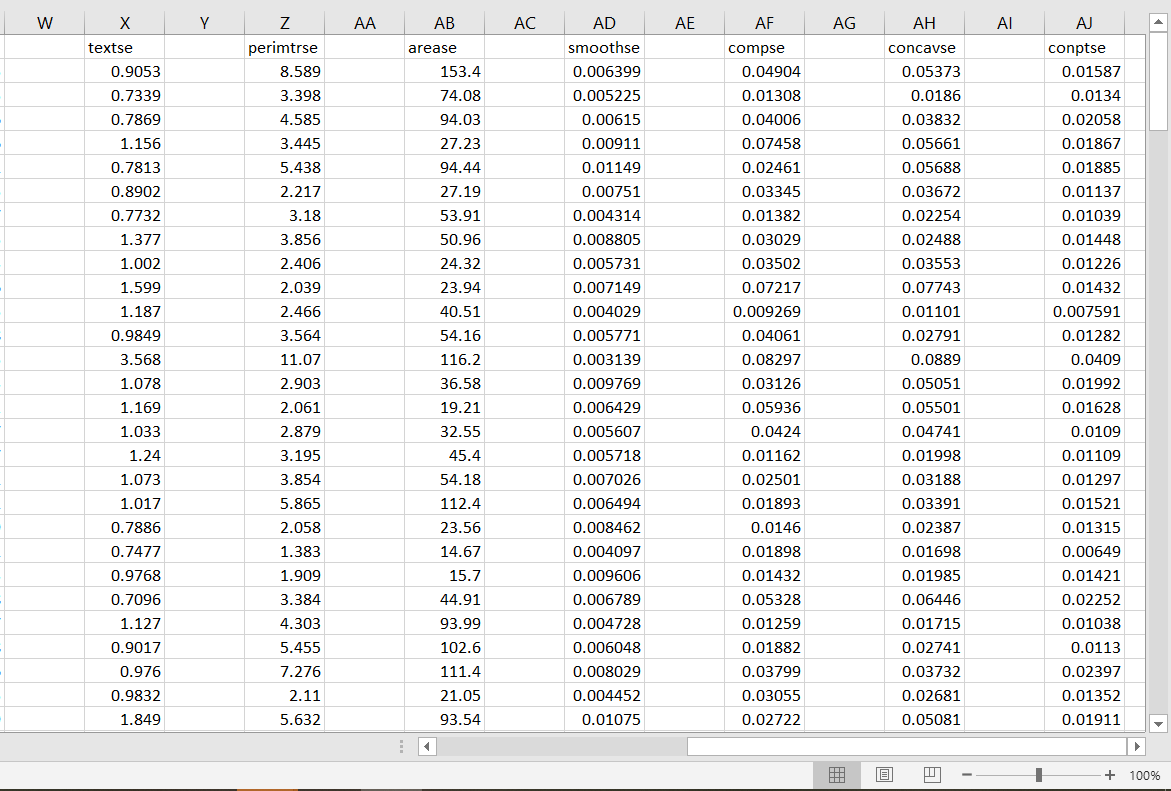
****

Fig 3.1.2: Data set containing the attributes (remaining 7 attributes)

The dataset consists of 18 attributes namely radius mean, texture mean, perimeter mean, area mean, smoothness mean, comp mean, concave mean, concave points mean, symmetry mean, fractal mean, radius standard error, texture standard error, perimeter standard error, perimeter standard error, area standard error, smoothness standard error, concave standard error and concave points error. The 80% of the data the data trained and the remaining 20% is tested.

**3.2 Modules**

**Front End:** In front-end we have only the test information i.e. the various features of tumor and the result is observed in output screen.

**Back End:** In back end we have the training set and testing set and the excel sheet is created and stored with data with the help of the algorithm which divides the tumor into malignant or benign.

**Database:** And the main part of any prediction projects is the database. For this we create excel sheet with the help of a program and pass it as a database.A level-set based algorithm will be applied to detect. The data set is collected from UCI Machine Repository.

**3.3 UML DIAGRAMS**

UML Diagrams for the proposed system are as follows :

**3.3.1 Use Case Diagram**

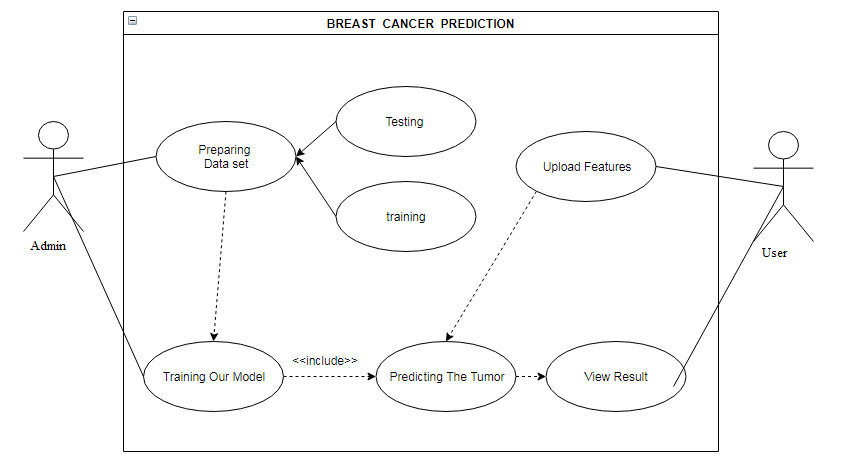


Fig 3.3.1: Use Case Diagram Of Proposed System

The Fig 3.3.1 refers to the use case diagram depicts the functionality of User , Administrator , Database Manager Administrator is responsible for training data . Administrator is the one who add all the contour values to the excel sheet. Administrator collects all the training data set and test data. User Need to pass the test data to the model . User need to pass the contour values to the final Module, such that database Manager will predict the output. Data base Manager will predict output and based on training data and test data he will predict even accuracy.

**3.3.2 Class Diagram**

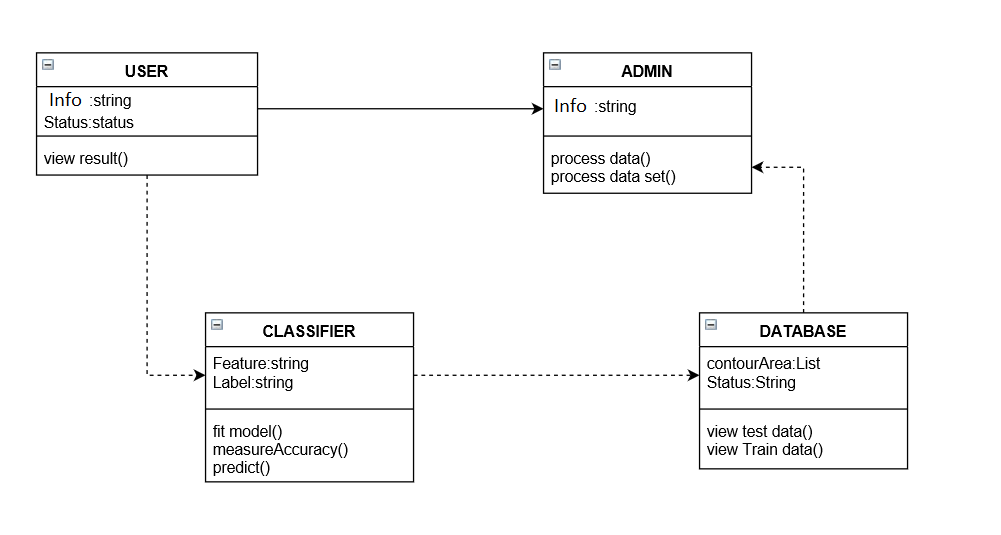
****

Fig 3.3.2: Class Diagram Of Proposed System

In Breast Cancer Prediction, we mainly identified three classes namely Admin , User , Database Manager. We also identify the attributes and operations for these classes. These classes along with their respective attributes and operations are depicted in above class diagram.

* 1. **SYSTEM IMPLEMENTATION**

The implementation stage of any project is a true display of the defining moments that make a project a success or a failure. The implementation stage is defined as the system or system modifications being installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. This phase continues until the system is operating in production in accordance with the defined user requirements.

**4.1 Code for Breast Cancer Prediction**

import pandas as pd

import numpy as np

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn import tree

import cv2

import numpy as np

import pandas as pd

import csv

df = pd.read\_csv('dataset\_new.csv')

features = df[['radmean','textmean','perimean','areamean','smoomean','compmean','concave','copoints','symmetry','fractal',

'radse','textse','perimtrse','arease','smoothse','compse','concavse','conptse']].values

labels = df[['result']].values

a=float(input("enter radius\_mean"))

b=float(input("enter texture\_mean"))

c=float(input("enter perimeter\_mean"))

d=float(input("enter area\_mean"))

e=float(input("enter smoothness\_mean"))

f=float(input("enter comp\_mean"))

g=float(input("concave\_mean"))

h=float(input("enter concavepoints\_mean"))

i=float(input("enter symmetry\_mean"))

j=float(input("enter fractal\_mean"))

k=float(input("enter radius standard error"))

l=float(input("enter texture standard error"))

m=float(input("enter perimeter standard error"))

n=float(input("enter area standard error"))

o=float(input("enter smoothness se"))

p=float(input("enter comp\_standard error"))

q=float(input("enter concave standard error"))

r=float(input("enter concave points error"))

#'''used for prediction'''

rf=RandomForestClassifier()

rf.fit(features,labels)

#'''used to train the model'''

pred\_rf = rf.predict([[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r]])

#'''to predict'''

print("\n Random Forest\_sample 1 : ",pred\_rf)

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

X\_trainset, X\_testset, y\_trainset, y\_testset = train\_test\_split(features, labels, test\_size=0.2, random\_state=3)

print(" ")

rf=RandomForestClassifier(n\_estimators = 199)

rf.fit(X\_trainset,y\_trainset)

pred\_rf = rf.predict(X\_testset)

print("Random\_Forest's Accuracy: ", metrics.accuracy\_score(y\_testset, pred\_rf)

**Explanation for prediction and generating accuracy:**

* The above section 4.1 contains the code for predicting the breast cancer.
* We need to install the required packages via command prompt.
* Here we need to provide the values for required prediction to excel sheet.
* We can execute this module with the packages mentioned at the beginning of the program.
* In this case, the labels refers to ‘B’ meaning benign and ‘M’ meaning malignant.
* The features are the 18 attributes that are present in the data set.
* The method fit() is used to train the model.
* We can find the Accuracy using metrics which is a module in sklearn.
* The accuracy of the proposed system is as high as 94%.
* More the no of information in the dataset, the more accurate is the prediction.
  1. **OUTPUT SCREENSHOTS**

Output Screens of various functionalities in our application are as follows:

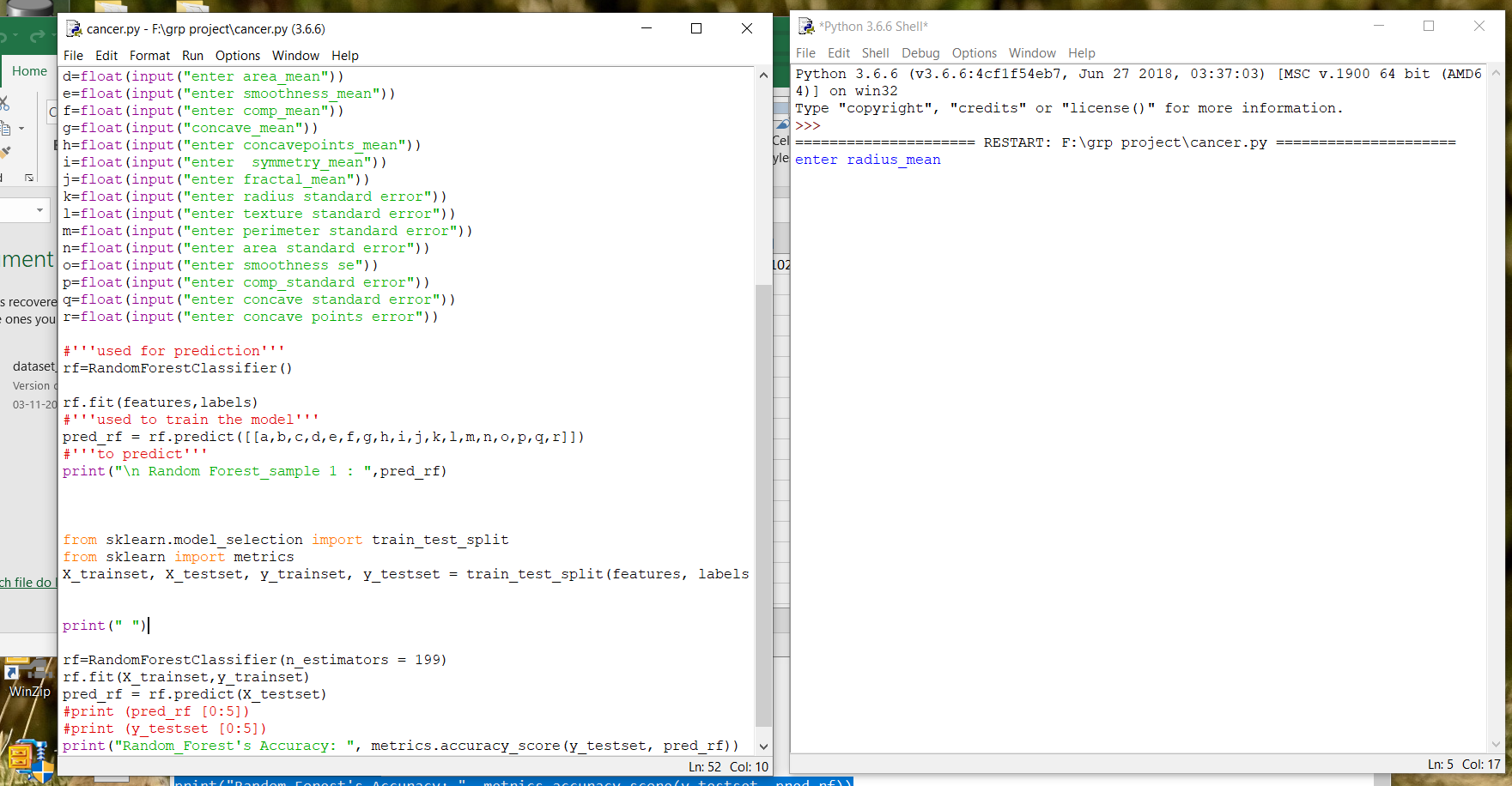


Fig 5.1 Screenshot of where the values have to be entered.

The values of the features of the tumor have to be provided by the user. There are total 18 features that have to be provided by the user in order to predict if the tumor is benign or malignant. After the values are provided we use Random Forest Classifier to predict the type of tumor and the accuracy.

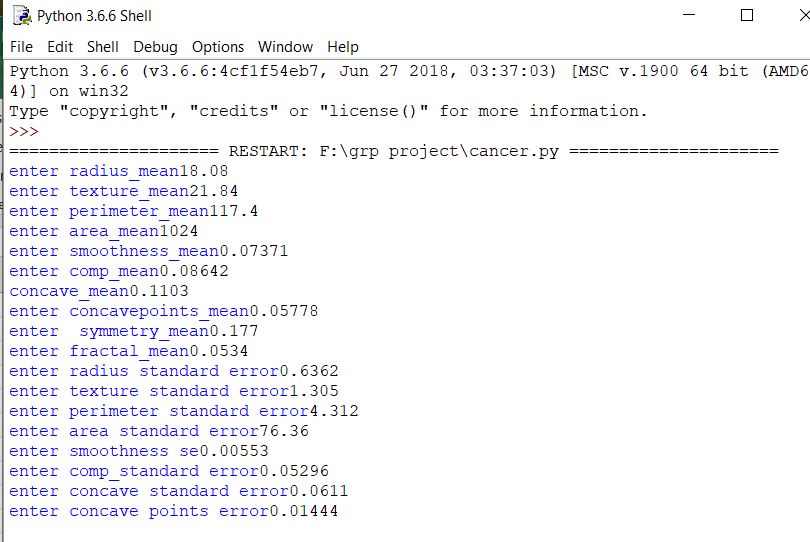


Fig 5.2: Screenshot of provided features

C:\Users\admin\Desktop\grp project\images\Capture2.PNG

C:\Users\admin\Desktop\grp project\images\Capture3.PNG

Fig 5.3 Output for Prediction and Generating Accuracy

The output image in Fig 5.3 is the prediction of the type of tumor. ‘M’ refers to malignant i.e. the tissue which spreads the cancer cells. The accuracy of the model is 92.5%. On various other cases it has generate accuracy as high as 94%.

**6. CONCLUSION AND FUTURE SCOPE**

**Conclusion**

Breast cancer is one of the leading causes of death in women. The proposed system helps in early detection of the cancer. Breast cancer detection done by less invasive technique gives digitized result which makes prediction easier. The proposed system uses Random Forest Classifier in order to find their accuracy. The classification model using Random Forest gave high accuracy. This method is fast and overcomes the problems caused by the traditional methods of prediction Therefore, the most accurate classifier can be used to detect the tumor so that the cure can be found in early stage.

**Future Scope**

* It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used.
* We can combine classification and prediction for more accuracy.
* If we consider more categories, then the output will be more accurate.
* We can even ease the approach , so that every individual can understand it easily.

**BIBLIOGRAPHY**

[1] Grady Booch, James Rumbaugh, Ivar Jacobson. *The Unified Modeling Language User*

*Guide*. Addison-Wesley, Reading, Mass., 1999.

[2] [www.towardsdatascience.com/understanding-random-forest](http://www.towardsdatascience.com/understanding-random-forest)

[3] *www.coderanch.com*

[4] www.w3schools.com

[5][*www.wikipedia.org*](http://www.wikipedia.org)

**APPENDIX**

**Python**

According to the latest [TIOBE Programming Community Index](https://www.tiobe.com/tiobe-index/), Python is one of the top 10 popular programming languages of 2017. Python is a general purpose and high level programming language. You can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks. The simple syntax rules of the programming language further makes it easier for you to keep the code base readable and application maintainable. There are also a number of reasons why you should prefer Python to other programming languages.

**7 Reasons Why You Must Consider Writing Software Applications in Python**

**1) Readable and Maintainable Code**

While writing a software application, you must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

**2) Multiple Programming Paradigms**

Like other modern programming languages, Python also supports several programming paradigm. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.

**3) Compatible with Major Platforms and Systems**

At present, Python is supports many operating systems. You can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation. Hence, you are not required to recompile the code after making any alteration. You can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to make changes to the code without increasing development time.

**4) Robust Standard Library**

Its large and robust standard library makes Python score over other programming languages. The standard library allows you to choose from a wide range of modules according to your precise needs. Each module further enables you to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, you can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. You can even gather information about various modules by browsing through the Python Standard Library documentation.

**5) Many Open Source Frameworks and Tools**

As an open source programming language, Python helps you to curtail software development cost significantly. You can even use several open source Python frameworks, libraries and development tools to curtail development time without increasing development cost. You even have option to choose from a wide range of open source Python frameworks and development tools according to your precise needs. For instance, you can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy. Likewise, you can accelerate desktop GUI application development using [**Python GUI frameworks**](http://www.allaboutweb.biz/python-gui-frameworks-usage/)and toolkits like PyQT, PyJs, PyGUI, Kivy, PyGTK and WxPython.

**6) Simplify Complex Software Development**

Python is a general purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. You can take advantage of the data analysis features of Python to create custom big data solutions without putting extra time and effort. At the same time, the data visualization libraries and APIs provided by Python help you to visualize and present data in a more appealing and effective way. Many [**Python developers**](http://www.mindfiresolutions.com/python-development.htm) even use Python to accomplish artificial intelligence (AI) and natural language processing tasks

**7) Adopt Test Driven Development**

You can use Python to create prototype of the software application rapidly. Also, you can build the software application directly from the prototype simply by refactoring the Python code. Python even makes it easier for you to perform coding and testing simultaneously by adopting test driven development (TDD) approach. You can easily write the required tests before writing code and use the tests to assess the application code continuously. The tests can also be used for checking if the application meets predefined requirements based on its source code.

However, Python, like other programming languages, has its own shortcomings. It lacks some of the built-in features provided by other modern programming language. Hence, you have to use Python libraries, modules, and frameworks to accelerate custom software development. Also, several studies have shown that Python is slower than several widely used programming languages including Java and C++. You have to speed up the Python application by making changes to the application code or using custom runtime. But you can always use Python to speed up software development and simplify software maintenance.