

A Project Report

on

**Breast Cancer Detection Using
Machine Learning**

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

K. Sumiya (164G1A05A6)

S. Nahida Anjum (164G1A0562)

C.S Thayyaba Sultana (164G1A05B3)

G. Tejaswini (164G1A05B2)

Under the Guidance of

G. Hemanth Kumar Yadav, M.Tech(Ph.D)
Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(B.Tech Program Accredited by NBA)

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPURAMU
(Accredited by NAAC with 'A' Grade, Affiliated to JNTUA, Approved by AICTE, New Delhi)

2019-2020



Certificate

This is to certify that the project report entitled **Breast Cancer Detection Using Machine Learning** is the bonafide work carried out by **K.Sumiya** bearing Roll Number **164G1A05A6**, **S. Nahida Anjum** bearing Roll Number **164G1A0562**, **G. Tejaswini** bearing Roll Number **164G1A05B2** and **C.S Thayyaba Sultana** bearing Roll Number **164G1A05B3** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2019-2020.

Guide

Mr.G. Hemanth Kumar Yadav,M.Tech(Ph.D)
Assistant Professor

Head of the Department

Dr. G.K.V. Narasimha Reddy,Ph.D
Professor & HOD

Date:

EXTERNAL EXAMINER

Ananthapuramu

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Mr.G.Hemanth Kumar Yadav_{M.Tech(Ph.D)} Computer Science & Engineering**, who has guided me a lot and encouraged me in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep-felt gratitude to **Mr.R.SandeepKumar_(Ph.D), Assistant Professor**, project coordinator valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We are very much thankful to **Dr. G.K.V.Narasimha Reddy, Ph.D, Professor & Head of the Department, Computer Science & Engineering**, for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr.T.Hitendra Sarma_{Ph.D}, Principal of Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities.

Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

Declaration

We, Ms K. Sumiya with reg no: 164G1A05A6, Ms C.S.Thayyaba Sultana with reg no: 164G1A05B3, Ms S.Nahida Anjum with reg no: 164G1A0562, Ms G.Tejaswini with reg no: 164G1A05B2 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that the dissertation entitled “BREAST CANCER DETECTION USING MACHINE LEARNING” embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of Mr Y.Hemanth Kumar Yadav M.Tech, Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, and this work has been submitted for the partial fulfilment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

K.SUMIYA

Reg no: 164G1A05A6

C.S THAYYABA SULTANA

Reg no: 164G1A05B3

G.TEJASWINI

Reg no: 164G1A05B2

S.NAHIDA ANJUM

Reg no: 164G1A0562

Contents

List of Figures	vii
List of Screens	viii
List of Abbreviations	ix
Abstract	x
Chapter 1: Introduction	1
1.1 Objective	1
1.2 MachineLearning	2
1.3 FeatureSelection	3
1.3.1 Feature SelectionTechniques	3
1.4 Classification	4
1.4.1 NaïveBayesClassifier	6
1.4.2 SupportVectorMachines	7
1.4.3 DecisionTreeClassifier	8
1.4.4 RandomForestClassifier	9
Chapter 2: Literature Survey	12
2.1 ExistingSystem	12
2.2 ProposedSystem	13
Chapter 3: Analysis	15
3.1 Introduction	15
3.2 SoftwareRequirementsSpecification	15
3.3 HardwareRequirements	16
3.4 SoftwareRequirements	16
3.4.1 GoogleCollaboratory	16
3.4.2 LanguagesUsed	19
Chapter 4: Design	21
4.1 UMLIntroduction	21
4.1.2 Usage of UMLinProject	21
4.2 Data FlowDiagram	22
4.3 Steps involvedinDesign	23
Chapter 5: Implementation	24
5.1 LibrariesUsed	24
5.2 Implementation	26
5.2.1 Feature Selection usingSFS	28
5.2.2 Feature Selection usingSBE	30
5.2.3 ModelConstructionusingRFC	32

5.2.4	PredictionsandMetrics	38
5.2.4.1	Predictions and Metrics forSFS	41
5.2.4.2	Predictions and Metrics forSBE	42
Chapter 6: Testing		43
6.1	Black-BoxTesting	43
6.2	White-BoxTesting	44
6.3	PerformanceEvaluation	44
Conclusion		45
Bibliography		46

List of Figures

Fig.No	Description	Page No.
1.1	Binomial Classification	5
1.2	Multi-class classification	5
1.3	SVM	7
1.4	Decision Trees	9
1.5	Tress in Random Forest	10
3.4	Uploading file into Google Colab	18
4.1	Data flow for breast cancer Prediction	22
5.1	Importing Libraries	27
5.7	Process of Randomly selected Features	33
5.9	Data Splitting	35

List of Screens

Screen No.	List of screen	Page No.
3.1	Pop for creating new tab	17
3.2	New notebook	17
3.3	Running Environment	18
5.2	Input dataset of 30 features	28
5.3	Sequential forward selection function	29
5.4	Selected features using SFS	30
5.5	Sequential backward selection function	31
5.6	Output of sequential backward elimination	32
5.8	Random forest function	34
5.10	Selected features using SFS	35
5.11	Training the dataset	36
5.12	Training and testing dataset	36
5.13	Selected features using SBE	37
5.14	Splitting the data and constructing model	38
5.15	Training and testing dataset	38
5.16	Metrics	40
5.17	Prediction of SFS	41
5.18	Metrics of SFS	41
5.19	Prediction of SBE	42
5.20	Metrics of SBE	42

List of Abbreviations

CSV	Comma-separated values
SFS	Sequential feature selection
SBE	Sequential Backward Elimination
SVM	Support Vector Machine
KNN	K-Nearest Neighbour
SRS	Software Requirement Specification
UML	Unified modelling language
Numpy	Numerical Python
ML	Machine Learning

ABSTRACT

Health care industry plays a vital role in saving people's life. Developing a software which can be used in such a field increases the scope of engineering in larger context. Cancer has been portrayed as a heterogeneous disease comprising of a wide range of subtypes. The early diagnosis of a cancer type is very important to determine the course of medical treatment required by the patient. The significance of classifying cancerous cells into benign or malignant has driven many research studies. In the past years researchers have been encouraged to use different machine learning (ML) techniques for cancer detection, as well as prediction of survivability and recurrence.

Many ML projects have been developed on Breast Cancer Detection but Most of the models were build with utmost 30 features and the drawbacks due to the presence of large number of dimensions are difficulty of models to interpret by researchers/users, Larger training times, curse of dimensionality, problem of overfitting.

In our project, we have used machine learning algorithm like Random Forest Algorithm to classify cancer patients and detect the type of cancer. Given a few parameters, our algorithms can predict whether the patient has malignant cancer or benign cancer and we have applied two different feature selection techniques namely sequential forward selection and sequential backward elimination with Random Forest algorithm to cancer dataset and sequential forward selection with the selected algorithm was found to provide the most effective results. Proper subset of features was found which was crucial in detecting malignancy.

CHAPTER 1

INTRODUCTION

Cancer has been portrayed as a heterogeneous disease comprising of a wide range of subtypes. Breast cancer is cancer that forms in the cells of the breasts. After skin cancer, breast cancer is the most common cancer diagnosed in women all over the world. Breast cancer can occur in both men and women, but it's far more common in women. The early diagnosis of a cancer is very important to determine the course of medical treatment required by the patient. Classifying cancerous cells into benign or malignant is a significant task. It has been an attempt over last many years that how to detect and cure cancer. There are several levels of cancer from 1 to 6. However, on the good side if cancer is detected when it is in level 1 or 2 or at the very initial stage, there is a significant probability that it will get cured within a period of time.

The data science and machine learning algorithms are used for classification. For classification using machine learning algorithm all of the features present in the dataset might not be useful in building a machine learning model to make the necessary prediction. Using too many features might even make the predictions worse.

1.1 Objective:

The main goal of this project is build a model to obtain best possible accuracy with minimal set of features to classify whether the cancerous cells are benign or malignant. Feature selection plays a huge role in building a machine learning model. Feature selection can make use of less number of data features to give best possible accuracy. Especially when dealing with a large number of variables there is a need for dimensionality reduction. Feature Selection can significantly improve a learning algorithm's performance. In this project Wisconsin Breast Cancer Diagnosis(WBCD) dataset is employed and the dataset is available in .csv format.

1.2 Machine Learning:

Machine Learning is a study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. It combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights.

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

1.Supervised Learning:

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training_data, and consists of a set of training examples.

Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range.

2.Unsupervised Learning:

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms therefore learn from test data that has not been labeled, classified or categorized. Cluster

analysis is the assignment of a set of observations into subsets (called *clusters*) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Other methods are based on estimated density and graph connectivity.

1.3 Feature Selection:

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for several reasons:

- simplification of models to make them easier to interpret by researchers/users
- shorter training times,
- to avoid the curse of dimensionality,
- enhanced generalization by reducing overfitting.

The central premise when using a feature selection technique is that the data contains some features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information.

1.3.1 Feature selection techniques:

There are three general classes of feature selection algorithms: filter methods, wrapper methods and embedded methods. In our project we have used Wrapper Methods.

Filter Methods

Filter feature selection methods apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either selected to be kept or removed from the dataset.

Some examples of some filter methods include the Chi squared test, information gain and correlation coefficient scores.

Embedded methods

This method complete the feature selection process within the construction of the machine learning algorithm itself. In other words, they perform feature selection during the model training, which is why we call them embedded methods.

Wrapper Methods

Wrapper methods are based on greedy search algorithms as they evaluate all possible combinations of the features and select the combination that produces the best result for a specific machine learning algorithm.

Categories:

- a) Sequential feature selection(SFS),
- b) Sequential Backward Elimination(SBE) and
- c) Exhaustive feature selection.

In this project, SFS and SBE are used for selecting feature.

1.4 Classification:

Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). Classification belongs to the category of supervised learning where the targets also provided with the input data. There are many applications in classification in many domains such as in credit approval, medical diagnosis, target marketing etc.

There are 2 types of Classification:

- Binomial
- Multi-Class

Binomial:

Binary or binomial classification is the task of classifying the elements of a given set into two groups (predicting which group each one belongs to) on the basis of a classification rule. Our project is a Binomial Classification that is Benign(B) and Malignant(M).

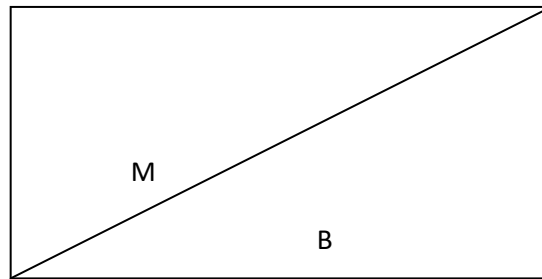


Fig 1.1: Binomial Classification

Multi-class:

This classification is the task of classifying the elements of a given set into more than two groups (predicting which group each one belongs to) on the basis of a classification rule.

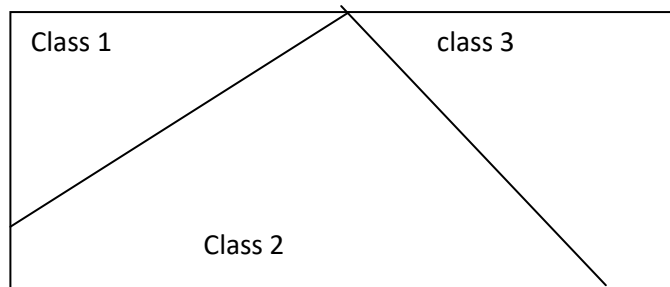


Fig 1.2: Multi-class classification

Some of the Classification Algorithms are:

- Random Forest
- Decision Tree
- Support Vector Machine
- Naïve Bayes Classifier
- Nearest Neighbour

1.4.1 Naïve Bayes Classifier:

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve:** It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes:** It is called Bayes because it depends on the principle of Bayes' Theorem.

1.4.2 Support Vector Machines:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

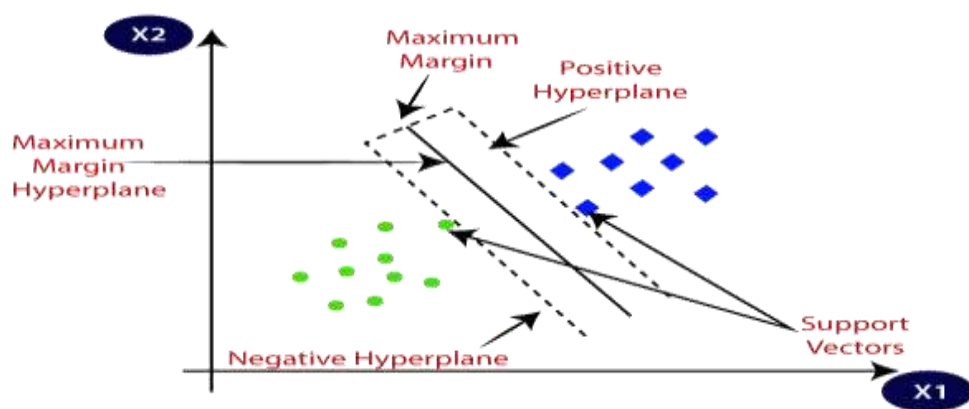


Fig 1.3: SVM.

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

1.4.3 Decision Tree Classifier:

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. An example is shown in the diagram at right. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

A decision tree is a simple representation for classifying examples. For this section, assume that all of the input features have finite discrete domains, and there is a single target feature called the "classification". Each element of the domain of the classification is called a class. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with an input feature are labeled with each of the possible values of the target or output feature or the arc leads to a subordinate decision node on a different input feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. See the examples illustrated in the figure for spaces that have and have not been partitioned using recursive partitioning, or recursive binary splitting. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This process

of top-down induction of decision trees is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data.

In data mining, decision trees can be described also as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data. Data comes in records of the form:

$$(X, Y) = (x_1, x_2, x_3, \dots, x_k, Y)$$

The dependent variable, Y , is the target variable that we are trying to understand, classify or generalize. The vector \mathbf{X} is composed of the features, x_1, x_2, x_3 etc., that are used for that task.

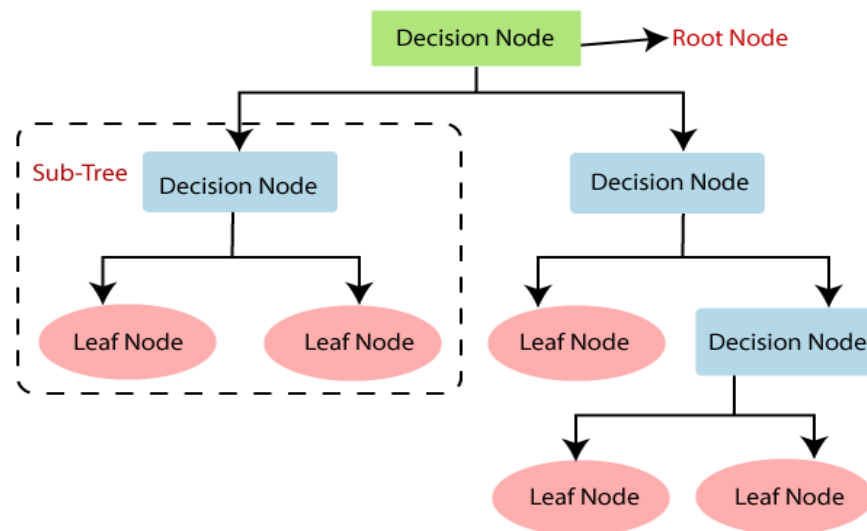


Fig 1.4: Decision Tress

In this project the Random Forest classification algorithm is used as its performance is quite effective.

1.4.4 Random Forest Classifier:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process

of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output.

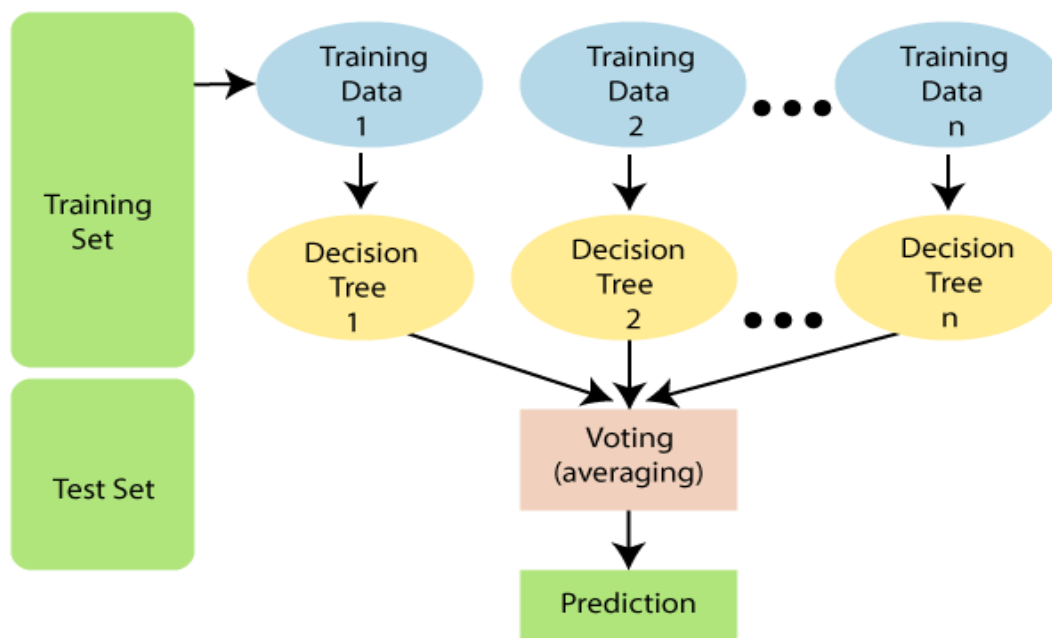


Fig 1.5: Trees in Random Forest

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. Two types of randomness are built into the trees.

1. Each tree is built on a random sample from the original data.
2. At each tree node, a subset of features are randomly selected to generate the best split.

Random sampling of training observations:

When training, each tree in a random forest learns from a random sample of the data points. The samples are drawn with replacement, known as *bootstrapping*, which means that some samples will be used multiple times in a single tree. The idea is that by training each tree on different samples, although each tree might have high variance with respect to a particular set of the training data, overall, the entire forest will have lower variance but not at the cost of increasing the bias.

At test time, predictions are made by averaging the predictions of each decision tree. This procedure of training each individual learner on different bootstrapped subsets of the data and then averaging the predictions is known as *bagging*, short for *bootstrap aggregating*.

Random Subsets of features for splitting nodes:

The other main concept in the random forest is that only a subset of all the features are considered for splitting each node in each decision tree. Generally this is set to \sqrt{n} (features) for classification meaning that if there are 16 features, at each node in each tree, only 4 random features will be considered for splitting the node. (The random forest can also be trained considering all the features at every node as is common in regression. These options can be controlled in the Scikit-Learn Random Forest implementation).

Advantages

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing System:

The most frequently occurring cancer among Indian women is breast cancer. There is a chance of fifty percent for fatality in a case as one of two women diagnosed with breast cancer die in the cases of Indian women. The existing projects of Breast Cancer Detection using machine learning algorithms aims to present comparison of the largely popular machine learning algorithms and techniques commonly used for breast cancer prediction, namely Random Forest, KNN (k-Nearest-Neighbour) and Naïve Bayes. The Wisconsin Diagnosis Breast Cancer data set was used as a training set to compare the performance of the various machine learning techniques in terms of key parameters such as accuracy, and precision. The results obtained are very competitive and can be used for detection and treatment.

Most of the models were build with utmost 30 features and the problems arise due to the large number of dimensions are

- Difficulty of models to interpret by researchers/users
- Larger training times,
- curse of dimensionality,
- problem of overfitting.

Sreyam Dasgupta, Ronit Chaudhuri & Swarnalatha Purushotham published a paper in International Journal of Innovative Technology and Exploring Engineering (IJITEE) on July 2019[1].

HabibDhahri Eslam Al Maghayreh, Awais Mahmood,Wail Elkilani & Mohammed Faisal Nagi in their journal and research article Automated Breast Cancer Diagnosis Based on Machine Learning Algorithms discussed about the different feature selection techniques and methods such as wrapper methods, filter methods & embedded methods to select the best features and perfect parameter values[2].

Bin Dai ,Rung-Ching ,Chen Shun-Zhi Zhu &Wei-Wei Zhang being the authors IEEE published a paper in 2018 International Symposium on Computer, Consumer and Control (IS3C) [3] .

Shubham Sharma ,Archit Aggarwal & Tanupriya Choudhury in the paper published by IEEE in 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)[4].

Hiba Asria, Hajar Mousannif, Hassan Al Moatassime & Thomas Noel published a paper Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis in The 6th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS 2016) [5].

2.2 Proposed System:

Our project aims to resolve a key challenge in cancer detection, which is “how to classify tumours into Malignant or Benign”. In this project we will try to make every effort to classify tumours into Malignant or benign tumours using features of pain from several cell images using machine learning techniques which helps to improve the accuracy of diagnosis dramatically.

Here we have used machine learning algorithm like Random Forest Algorithm to classify cancer patients and detect the type of cancer. Given a few parameters, our algorithms can predict whether the patient has malignant cancer or benign cancer.

Now, apply feature selection techniques to the breast cancer dataset which has 32 features in it. Then a subset of features which are most important in the prediction of

breast cancer to be malignant or benign are obtained. choose the classifier that gives the best possible accuracy with the subset of features obtained after feature selection. Applying the classifiers to the dataset actually mean that needs to train the model with the classifiers and test the data so that the model will be fit.

CHAPTER 3

ANALYSIS

3.1 Introduction

The Analysis Phase is where the project life cycle begins. This is the phase where you break down the deliverables in the high-level Project Charter into the more detailed business requirements. Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal process, which will be developed in more detail, consists of four basic steps.

1. **Elicitation** – I ask questions, you talk, I listen
2. **Validation** – I analyze, I ask follow-up questions
3. **Specification** – I document, I ask follow-up questions
4. **Verification** – We all agree

Most of the work in the Analysis Phase is performed by the role of analyst.

3.2 Software Requirement Specification

SRS is a document created by system analyst after the requirements are collected. SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.

The requirements received from client are written in natural language. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.

3.3 Hardware Requirements

Any Contemporary PC.

3.4 Software Requirements

- Operating System : Windows 10
- Tools : Google Collaboratory
- Dataset : Excel Sheet
- Languages Used : Python

3.4.1 Google Collaboratory

Google Colab is the platform and a free Jupyter notebook environment provided by Google where we can build a Machine Learning Models using Python programming language.

Steps:

1. Upload your data into Google drive. Here we have uploaded .CSV file which contains Breast cancer dataset of many patients. Upload your data into Google drive. Here we have uploaded .CSV file which contains Breast cancer dataset of many patients. CSV is a simple file format used to store tabular data, such as a spreadsheet or database. Files in the CSV format can be imported to and exported from programs that store data in tables, such as Microsoft Excel or OpenOffice Calc. Its data fields are most often separated, or delimited, by a comma. A CSV is a comma-separated values file, which allows data to be saved in a tabular format. CSVs look like a garden-variety spreadsheet but with a .csv extension. CSV

files can be used with most any spreadsheet program, such as Microsoft Excel or Google Spreadsheets.

2. To start working with Colab you first need to log in to your google account, then go to this link <https://colab.research.google.com>.
3. **Opening your Jupiter Notebook:** On opening the website you will see a pop-up containing following tab.

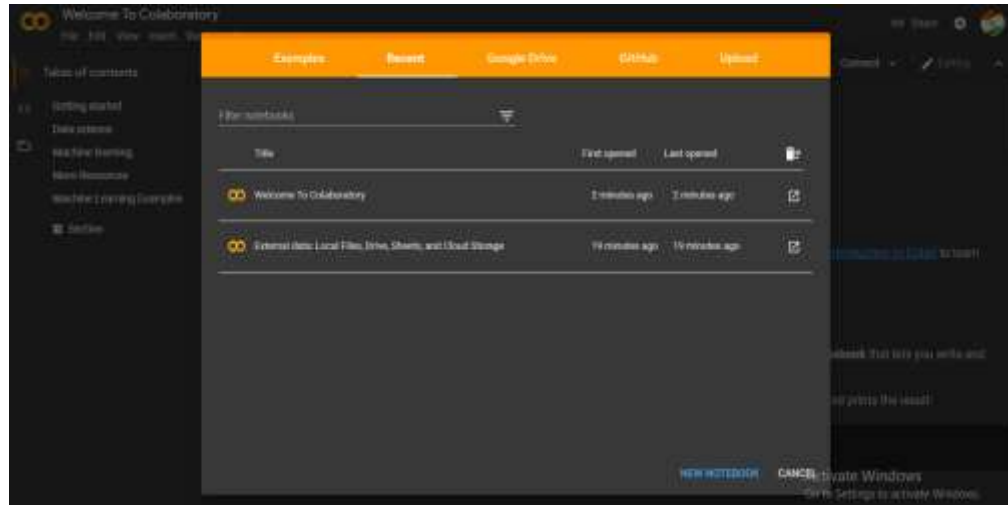


Fig 3.1: Pop up tab for creating new notebook.

4. Click on New Notebook at the bottom right corner to create new Notebook.

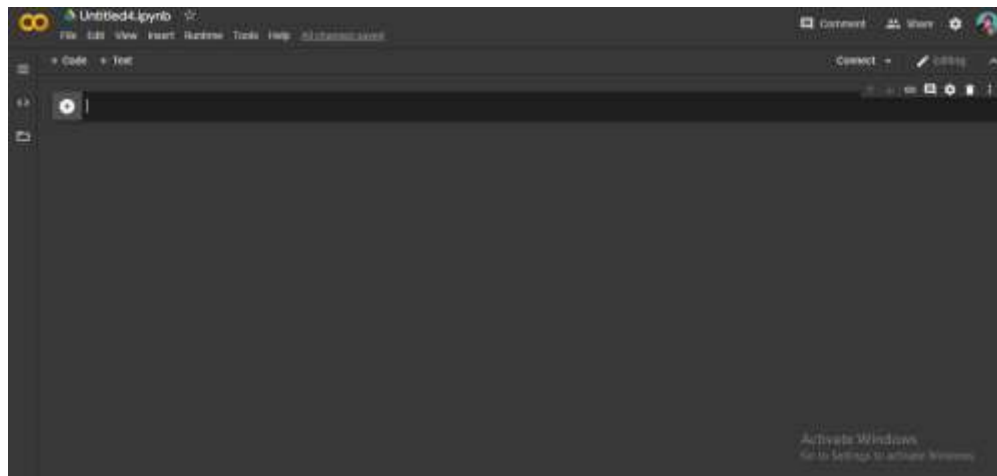


Fig 3.2: New notebook.

On creating a new notebook, it will create a Jupyter notebook with Untitled0.ipynb and save it to your google drive in a folder named **Colab Notebooks**. Now as it is essentially a Jupyter notebook, all commands of Jupyter notebooks will work here. You can change the file name by file ->rename and save it.

5. **Runtime Environment:** Click the “Runtime” dropdown menu. Select “Change runtime type”. Select python2 or 3 from “Runtime type” dropdown menu.

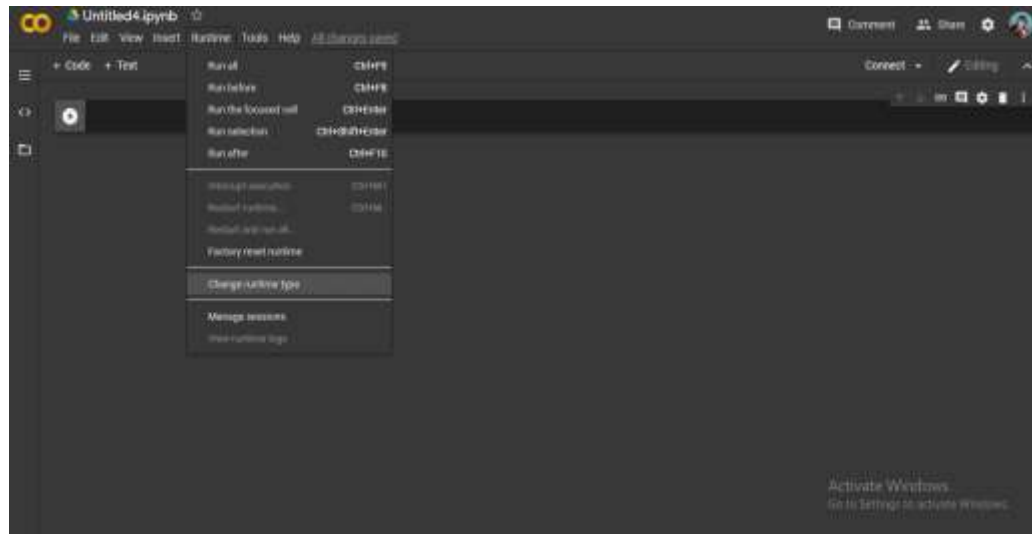


Fig 3.3: Running Environment

6. Upload file into google colab from google drive using following code

```
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

auth.authenticate_user()

gauth = GoogleAuth()

gauth.credentials = GoogleCredentials.get_application_default()

drive = GoogleDrive(gauth)
```

Fig: 3.4: Uploading file into google colab

3.4.2 Languages Used

The programming language that was used in our Breast Cancer Detection project is Python. The implementation of source code was done through python. Python is an interpreted, interactive, object-oriented programming language which is suitable for implementing machine learning algorithms in easier way.

Features of Python

Python provides lots of features that are listed below:

Easy to Learn and Use:

Python is easy to learn and use. It is developer-friendly and high-level programming language.

Expressive Language:

Python language is more expressive means that it is more understandable and readable.

Interpreted Language:

Python is an interpreted language i.e. interpreter executes the code line by line at a line. This makes debugging easy and thus suitable for beginners.

Cross-platform Language:

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

Free and Open Source:

Python language is freely available at official web address. The source-code is also available. Therefore, it is open source.

Object-Oriented Language:

Python supports object-oriented language and concepts of classes and objects come into existence.

Extensible:

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

Large Standard Library:

Python has a large and broad library and provides rich set of module and functions for rapid application development.

GUI Programming Support:

Graphical user interfaces can be developed using Python.

Integrated:

It can be easily integrated with languages like C, C++ and JAVA etc.

CHAPTER 4

DESIGN

4.1 UML Introduction:

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective.

UML is specifically constructed through two different domains, they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.
- UML Design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

4.1.2 Usage of UML in Project

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

4.2 Data Flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system (usually an information system). This also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

The data-flow diagram is part of the structured-analysis modeling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram.

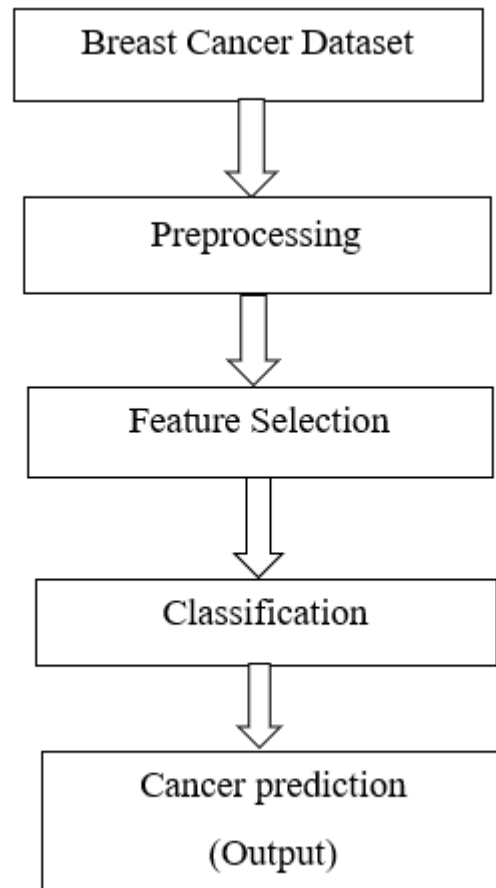


Fig 4.1: Data flow diagram for breast cancer prediction

The above data flow diagram describes how breast cancer prediction can be done. Initially obtain the datasets from any websites or generate own datasets. After obtaining the datasets, perform data transformation to it in such a way that there shouldn't be any integration problem or any redundancy issue.

Now, apply feature selection techniques to the breast cancer dataset which has 32 features in it. Then a subset of features which are most important in the prediction of breast cancer to be malignant or benign are obtained. choose the classifier that gives the best possible accuracy with the subset of features obtained after feature selection. Applying the classifiers to the dataset actually mean that needs to train the model with the classifiers and test the data so that the model will be fit.

4.3 Steps involved in Design:

- Data Collection
- Data Preprocessing
- Feature Selection
- Applying ML algorithms(Random forest classifier)

Each step has its own specific reason and plays prominent role in building up a model of the project. Each step has been explained in detail in implementation part.

CHAPTER 5

IMPLEMENTATION

Here, in our project, feature selection is made from 30 features by using different wrapper feature selection methods of Machine learning. Models are constructed with the selected features that are generated from different selection methods by using Random Forest Classifier Algorithm. Accuracy is calculated for each model and the best model with best features is selected as result based on high accuracy.

5.1 Libraries Used

Python is increasingly being used as a scientific language. Matrix and vector manipulation are extremely important for scientific computations. Both NumPy and Pandas have emerged to be essential libraries for any scientific computation, including machine learning, in python due to their intuitive syntax and high-performance matrix computation capabilities.

NumPy:

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since, arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem.

NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation. NumPy can be imported into the notebook using

```
import numpy as np.
```

Pandas:

Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Pandas provides in-memory 2d table object called Data frame. It is like a spreadsheet with column names and row labels.

Hence, with 2d tables, pandas are capable of providing many additional functionalities like creating pivot tables, computing columns based on other columns and plotting graphs. Pandas can be imported into Python using:

```
import pandas as pd.
```

pip:

The pip command is a tool for installing and managing Python packages, such as those found in the Python Package Index. It's a replacement for easy install. The easiest way to install the nfl* python modules and keep them up-to-date is with a Python-based package manager called Pip.

Sklearn:

Skikit-learn is a free software machine library for Python programming language. It features various classification, regression and clustering algorithms including support vector machine, random forest, k-means and gradient boosting. In our project we have used different features.

- **from sklearn.ensemble import RandomForestClassifier:**

Used for Random Forest Classifier algorithm.

- **from sklearn.model_selection import train_test_split:**

Used for Splitting the dataset into Training and Testing.

- **from sklearn.metrics import accuracy_score as acc**

Used for calculating the Accuracy.

MLxtend:

MLxtend is a library that implements a variety of core algorithms and utilities for machine learning and data mining. It implements a large variety of functions, highlights include sequential feature selection algorithms, implementations of stacked generalization for classification and regression, and algorithms for frequent pattern mining. The sequential feature selection algorithms cover forward, backward, forward floating, and backward floating selection and leverage scikit-learn's cross-validation to ensure satisfactory generalization performance upon constructing and selecting feature subsets. It is imported as:

```
from mlxtend.feature_selection import SequentialFeatureSelector as sfs
```

5.2 Implementation:

After importing the .CSV dataset file into the Google collaborator as we mentioned in Chapter 3.

Implementing Libraries:

```
import pandas as pd
```

```
import numpy as np
```

```
from sklearn.ensemble import RandomForestClassifier  
  
from sklearn.model_selection import train_test_split  
  
from sklearn.metrics import accuracy_score as acc  
  
from sklearn.preprocessing import StandardScaler  
  
from mlxtend.feature_selection import SequentialFeatureSelector as sfs  
  
from sklearn.metrics import confusion_matrix
```

Fig 5.1 : Libraries.

Input:

```
data = pd.read_csv('CancerDataset.csv')
```

Reading the dataset file using pandas library.

Comma-separated values(CSV):

The dataset used in this project is a .CSV file.

In computing, a comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format.

CSV is a simple file format used to store tabular data, such as a spreadsheet or database. Files in the CSV format can be imported to and exported from programs that store data in tables, such as Microsoft Excel or OpenOffice Calc. Its data fields are most often separated, or delimited, by a comma. A CSV is a comma-separated values file,

which allows data to be saved in a tabular format. CSVs look like a garden-variety spreadsheet but with a .csv extension. CSV files can be used with most any spreadsheet program, such as Microsoft Excel or Google Spreadsheets.

The difference between CSV and XLS file formats is that CSV format is a plain text format in which values are separated by commas (Comma Separated Values), while XLS file format is an Excel Sheets binary file format which holds information about all the worksheets in a file, including both content and formatting.

The screenshot shows a text editor window titled 'Cancerdataset.csv'. The file is open, displaying a large table of data. The table has 30 columns, each representing a different feature used for breast cancer detection. The columns are labeled with names like 'mean radius', 'mean texture', 'mean perimeter', etc. The data is organized into rows, with each row representing a single data point or patient record. The values are separated by commas, which is the standard format for CSV files. The editor interface includes a menu bar at the top with options like 'File', 'Edit', 'Format', and 'Tools'. There are also icons for saving, opening, and printing the file.

Fig 5.2 : input dataset of 30 features.

5.2.1 Feature Selection using Sequential feature selection(SFS):

Sequential feature selection starts with the evaluation of each individual feature, and selects that which results in the best performing selected algorithm model. Best is

depends entirely on the defined evaluation criteria (AUC, prediction accuracy, RMSE, etc. It the simplest greedy search algorithm. Starting from the empty set, sequentially add the feature x_+ that results in the highest objective function $J(Y_k + x_+)$ when combined with the features Y_k that have already been selected .

Algorithm:

1. Start with the empty set $Y_0 = \{\phi\}$
2. Select the next best feature $X_+ = \text{argmax}[J(Y_k + X)]; x \notin Y_k$
3. Update $Y_{k+1} = Y_k + X_+ ; k = k + 1$
4. Goto 2
5. Process continues until the specified number of features are selected.

Code:

```
clf=RandomForestClassifier(n_estimators=100,n_jobs=-1)
sfs1=sfs(clf,k_features=9,forward=True,floating=False,scoring='accuracy',cv=
5,verbose=2)
sfs1=sfs1.fit(X_train,Y_train)
```

Fig 5.3: Sequential forward selection function.

Verbose: Displays the information of building tree.

Forward: When it is true then it is a SFS , else it is SBE.

Cv: Cross Validation.

Output:

ForwardSelection_RandomForest.csv

Open with ▾

	A	B	C	D	E	F	G	H	I	J	K
id	diagnosis	radius_mean	compactness_mean	area_worst	texture_worst	radius_worst	compactness_worst	asymmetry_mean	asymmetry_worst	concavity_worst	
842302	M	17.33	0.2776	2019	17.33	25.38	0.8856	0.2418	0.4601	0.7119	
842517	M	25.57	0.37864	1958	23.41	24.35	0.1885	0.1912	0.275	0.2418	
84300903	M	18.69	0.1559	1703	25.53	23.57	0.4245	0.2059	0.3613	0.4594	
84348301	M	11.42	0.2839	587.7	25.5	14.91	0.8863	0.2587	0.6638	0.9859	
84356402	M	20.29	0.1328	1575	16.67	22.54	0.205	0.1858	0.2364	0.4	
843788	M	52.45	0.17	741.6	23.75	18.47	0.5249	0.2087	0.3965	0.5355	
844359	M	18.25	0.109	1698	27.06	22.88	0.2576	0.1734	0.7063	0.3784	
84458202	M	12.71	0.1645	897	28.14	17.05	0.7682	0.2195	0.3195	0.2878	
844981	M	13	0.1932	738.3	30.73	15.48	0.5401	0.235	0.4378	0.5339	
84501001	M	12.48	0.2386	711.4	40.88	15.08	1.058	0.293	0.4365	1.185	
845636	M	16.02	0.08859	1150	33.88	19.19	0.1551	0.1528	0.2948	0.1458	
84610002	M	15.78	0.1292	1239	27.28	26.42	0.5609	0.1842	0.3792	0.3985	
846226	M	18.17	0.2488	1332	29.94	28.98	0.3803	0.2387	0.2178	0.3638	
846381	M	15.85	0.1062	878.5	27.66	16.84	0.1824	0.1847	0.2809	0.2322	
84867401	M	12.73	0.2283	897.7	32.01	15.03	0.7725	0.2088	0.3586	0.5943	
84799002	M	14.54	0.1595	843.2	37.13	17.48	0.6577	0.2393	0.4218	0.7028	
846408	M	14.89	0.872	1138	30.88	18.37	0.1871	0.1588	0.3029	0.2814	
84862001	M	16.13	0.2822	1315	31.48	28.98	0.4233	0.2164	0.3706	0.4784	
848014	M	19.81	0.1027	2398	30.86	27.32	0.315	0.1982	0.2768	0.5372	
8510426	B	12.54	0.08129	711.2	19.28	15.11	0.1773	0.1885	0.2977	0.239	
8510853	B	13.08	0.127	836.5	20.48	14.5	0.2778	0.1987	0.3184	0.188	
8510824	B	9.504	0.08492	314.3	15.88	10.23	0.1148	0.1818	0.245	0.08887	
8511133	M	15.34	0.2135	980.3	18.08	18.37	0.5864	0.2521	0.4867	0.6385	
851509	M	21.18	0.1822	2615	35.59	28.17	0.26	0.1768	0.2822	0.8155	
852552	M	16.05	0.1457	2215	31.56	26.48	0.3578	0.1995	0.3613	0.4835	
852831	M	17.14	0.2278	1481	21.4	22.25	0.3949	0.394	0.4888	0.5853	

Fig 5.4: Selected features using SFS.

5.2.2 Features selection using Sequential Backward Elimination(SBE):

Sequential Backward Elimination works in the opposite direction of SFS. Also referred to as SBS (Sequential Backward Selection). Starting from the full set, sequentially remove the feature x_i that results in the smallest decrease in the value of the objective function $J(Y-x_i)$. Notice that removal of a feature may actually lead to an increase in the objective function $J(Y_k-x_i) > J(Y_k)$. Such functions are said to be non-monotonic.

Algorithm:

1. Start with the full set $Y_0=X$

2. Remove the worst feature $X_- = \text{argmax}[J(Y_k - X)]; x_{Y_k}$
3. Update $Y_{k+1} = Y_k - X_-$; $k = k + 1$
4. Goto 2
5. Process continues until the specified number of features are selected.

SBS works best when the optimal feature subset has a large number of features, since SBS spends most of its time visiting large subsets.

Code:

```
clf=RandomForestClassifier(n_estimators=100,n_jobs=-1)

sfs1=sfs(clf,k_features=9,forward=False,floating=False,scoring='accuracy',cv
=5,verbose=2)

sfs1=sfs1.fit(X_train,Y_train)
```

Fig 5.5: Sequential backward selection function.

When forward is equal to “False” it is a “Sequential backward elimination.”

Output

A	B	C	D	E	F	G	H	I	J	K
842302	M	0.1164	0.7119	0.4881	25.36	3019	17.33	0.1622	0.000390	0.03993
842517	M	0.08474	0.2416	0.275	24.69	1956	23.41	0.1236	0.005225	0.01389
8430983	M	0.1086	0.4904	0.3913	23.57	1709	25.53	0.1444	0.00019	0.0225
8434531	M	0.1425	0.9889	0.6238	14.81	587.7	26.5	0.2056	0.00911	0.05993
8439402	M	0.1005	0.4	0.2394	22.54	1575	16.37	0.1274	0.01149	0.01759
843798	M	0.1278	0.5355	0.3885	15.47	741.8	23.75	0.1791	0.00751	0.02185
844353	M	0.08463	0.3794	0.3983	22.88	1800	27.88	0.1442	0.004314	0.01383
84458202	M	0.1169	0.2676	0.3126	17.06	837	28.54	0.1854	0.008005	0.01498
844381	M	0.1273	0.533	0.4278	15.49	738.3	30.73	0.1703	0.005731	0.02143
8450101	M	0.1186	1.105	0.4365	15.09	711.4	40.58	0.1863	0.007148	0.01759
845636	M	0.06206	0.1489	0.2848	19.19	1150	33.88	0.1181	0.004029	0.01468
84510002	M	0.0871	0.3985	0.3782	20.42	1299	27.29	0.1388	0.005771	0.02999
846220	M	0.0874	0.3939	0.3179	20.98	1332	29.94	0.1057	0.003139	0.04484
845581	M	0.08401	0.2322	0.2858	18.54	576.5	27.88	0.1131	0.009789	0.02881
84987401	M	0.1151	0.6943	0.3996	15.03	857.7	32.91	0.1851	0.004429	0.01381
84789002	M	0.1139	0.7929	0.4218	17.46	843.2	37.13	0.1878	0.005007	0.01857
848408	M	0.09667	0.2214	0.3029	18.07	1138	30.88	0.1464	0.003718	0.01411
84822001	M	0.117	0.4784	0.3796	20.96	1315	31.48	0.1768	0.007026	0.01689
849014	M	0.09031	0.5372	0.2788	27.38	2394	30.88	0.1819	0.006494	0.01256
8510420	B	0.09779	0.239	0.2877	19.11	711.2	19.28	0.144	0.004402	0.0188
8510883	B	0.1076	0.189	0.3184	14.5	836.5	20.49	0.1212	0.004097	0.01679
8510524	B	0.1024	0.8997	0.245	19.25	814.9	19.88	0.1324	0.009906	0.02027
8511123	M	0.1073	0.5395	0.4887	18.07	980.9	19.58	0.138	0.005789	0.02672
851509	M	0.08426	0.3155	0.2822	29.17	3515	35.59	0.1401	0.004728	0.01582
852032	M	0.1121	0.4995	0.3813	29.48	2215	31.58	0.1866	0.008048	0.01488
852831	M	0.1186	0.3953	0.4888	22.25	1481	21.4	0.1545	0.008025	0.02298
857501	M	0.1194	0.4515	0.4384	17.80	858.9	19.51	0.1806	0.004402	0.0188

Fig 5.6: Output of Sequential backward elimination.

5.2.3 Model construction using Random Forest Classifier:

Random Forest Classifier:

The Random forest combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree. This model uses two key concepts that gives it the name *random*:

Algorithm:

1. Randomly select “K” features from total “m” features where $k \ll m$
2. Among the “K” features, calculate the node “d” using the best split point

3. Split the node into daughter nodes using the best split
4. Repeat the 1 to 3 steps until number of nodes has been reached
5. Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees



Fig 5.7: process of randomly selected features.

In the next stage, with the random forest classifier created, we will make the prediction. The random forest prediction pseudocode is shown below:

1. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target).
2. Calculate the votes for each predicted target.
3. Consider the high voted predicted target as the final prediction from the random forest algorithm.

```
Function:
clf = RandomForestClassifier(n_estimators=100, random_state=1, max_depth=4, ver
bose=2)
```

Fig 5.8: Random forest function.

Where,

X_train: Training input data set.

X_test: Testing input dataset.

Y_train: Training output label.

Y_test: Testing output label.

Test_size: Testing dataset size.

N_estimators: Number of trees in random forest.

Random_state: Selects the particular data rows for each execution.

Max_depth: Maximum depth of a tree.

Model: Model is a system that answers the question of a problem statement and this model is created via a process called “training”. The goal of training is to create an accurate model that answers our questions correctly most of the time.

Splitting of data: Splitting of data is dividing your data set into two subsets:

- **training set**—a subset to train a model which is used for fit and tune the model.

- **testing set**—a subset to test the trained model which is used to evaluate the model of unseen data.



Fig 5.9: Data splitting.

5.2.3(a) Model construction with SFS features:

Input:

```
data = pd.read_csv('ForwardSelection_RandomForest.csv')
```

The screenshot shows a CSV file named 'ForwardSelection_RandomForest.csv' with columns A through K. The data includes patient IDs, diagnoses, and various breast cancer features. The 'diagnosis' column has values 'M' (Malignant) and 'B' (Benign). The features include 'radius_mean', 'compactness_mean', 'area_worst', 'texture_worst', 'radius_worst', 'compactness_worst', 'asymmetry_mean', 'asymmetry_worst', and 'concavity_worst'.

	A	B	C	D	E	F	G	H	I	J	K
	id	diagnosis	radius_mean	compactness_mean	area_worst	texture_worst	radius_worst	compactness_worst	asymmetry_mean	asymmetry_worst	concavity_worst
	842302	M	17.33	0.2776	2019	17.33	25.38	0.8856	0.2419	0.4601	0.7119
	842517	M	20.57	0.37864	1956	23.41	24.95	0.1865	0.1812	0.275	0.2418
	84300603	M	18.59	0.1556	1709	25.53	23.57	0.4245	0.2059	0.2613	0.4504
	84345301	M	11.42	0.2839	587.7	25.5	14.91	0.8963	0.2597	0.6608	0.8859
	84356402	M	20.29	0.1328	1575	16.57	22.54	0.205	0.1939	0.2364	0.4
	843798	M	12.45	0.17	741.6	23.75	15.47	0.5249	0.2097	0.3965	0.5355
	844359	M	18.25	0.169	1698	27.66	22.88	0.2576	0.1794	0.3063	0.3784
	84456202	M	12.71	0.1645	897	25.14	17.05	0.3652	0.2195	0.3195	0.2978
	844981	M	13	0.1932	739.3	30.73	15.42	0.5401	0.235	0.4378	0.539
	84501001	M	12.48	0.2396	711.4	40.88	15.08	1.056	0.203	0.4368	1.185
	845636	M	18.02	0.39659	1150	33.86	19.19	0.1551	0.1528	0.2948	0.1459
	84810002	M	15.78	0.1292	1299	27.26	26.42	0.5609	0.1842	0.3792	0.3965
	846225	M	19.17	0.3458	1332	29.94	25.96	0.3903	0.2397	0.2178	0.3639
	846381	M	15.95	0.1002	876.5	27.66	16.84	0.1624	0.1847	0.2609	0.2322
	84967401	M	12.73	0.2293	987.7	32.51	15.02	0.7725	0.2089	0.3566	0.8943
	84799002	M	14.54	0.1595	943.2	37.13	17.48	0.6577	0.2303	0.4218	0.7929
	846406	M	14.89	0.072	1138	30.88	19.07	0.1671	0.1586	0.3029	0.2914
	84952001	M	16.13	0.2922	1215	31.48	20.98	0.4233	0.2154	0.3706	0.4784
	849014	M	19.61	0.1027	2398	30.86	27.32	0.315	0.1592	0.2768	0.5372
	8510426	B	12.54	0.08129	711.2	19.26	15.11	0.1773	0.1885	0.2977	0.239
	8510653	B	13.08	0.127	830.5	20.49	14.5	0.2776	0.1987	0.3184	0.189
	8510624	B	9.594	0.09492	214.9	15.86	10.23	0.1148	0.1815	0.245	0.09897
	8511133	M	15.34	0.2135	380.9	19.08	16.07	0.5964	0.2521	0.4667	0.6395
	851509	M	21.18	0.1622	2615	35.59	29.17	0.26	0.1769	0.2822	0.8355
	852552	M	16.65	0.1457	2215	31.56	26.48	0.3578	0.1995	0.2613	0.4935
	852831	M	17.14	0.2278	1481	21.4	22.25	0.3949	0.394	0.4068	0.3953

Fig 5.10: Selected features using SFS.

Code:

```

X=data.iloc[:,2:]

Y=data.iloc[:,1].values

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

sc=StandardScaler()

X_train=sc.fit_transform(X_train)

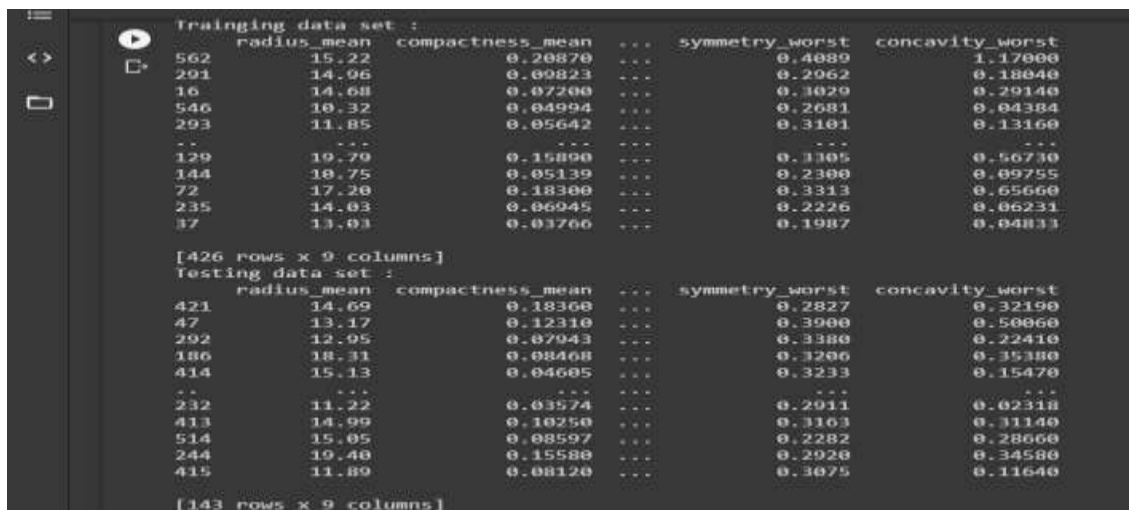
X_test=sc.fit_transform(X_test)

clf = RandomForestClassifier(n_estimators=100, random_state=1, max_depth=4,verbose
e=2)

clf.fit(X_train, Y_train)

```

Fig 5.11: Training the data.

Displaying Training and Testing dataset:


Training data set :				
	radius_mean	compactness_mean	symmetry_worst	concavity_worst
562	15.22	0.20870	0.4089	1.17000
291	14.96	0.09823	0.2962	0.18040
16	14.68	0.07200	0.3020	0.29140
546	10.32	0.04994	0.2681	0.04384
293	11.85	0.05642	0.3101	0.13160
..
129	10.79	0.15890	0.3305	0.56730
144	10.75	0.05139	0.2300	0.09755
72	17.20	0.18300	0.3313	0.65660
235	14.03	0.06945	0.2226	0.06231
37	13.03	0.03766	0.1987	0.04833

[426 rows x 9 columns]				
Testing data set :				
	radius_mean	compactness_mean	symmetry_worst	concavity_worst
421	14.69	0.18360	0.2827	0.32190
47	13.17	0.12310	0.3900	0.50060
292	12.95	0.07943	0.3380	0.22410
186	18.31	0.08468	0.3206	0.35380
414	15.13	0.04605	0.3233	0.15470
..
232	11.22	0.03574	0.2911	0.02318
413	14.99	0.10250	0.3163	0.31140
514	15.05	0.08597	0.2282	0.28060
244	19.40	0.15580	0.2920	0.34580
415	11.89	0.08120	0.3075	0.11640

[143 rows x 9 columns]				
------------------------	--	--	--	--

Fig 5.12: Training and Testing dataset.

5.2.3(b) Model construction with SBE features:

Input:

Data=pd.read(“forwardselection.csv”)

A	B	C	D	E	F	G	H	I	J	K
id	diagnosis	mean_freq_men	concavity_worst	symmetry_worst	radius_worst	area_worst	texture_worst	mean_freq_worst	mean_freq_se	symmetry_se
842302	M	0.1184	0.7119	0.4601	25.39	2019	17.33	0.1622	0.006399	0.03003
842517	M	0.08474	0.2416	0.279	24.99	1998	23.41	0.1238	0.006225	0.01389
8428805	M	0.1096	0.4504	0.3613	23.57	1708	25.53	0.1444	0.00615	0.0225
8434301	M	0.1425	0.6655	0.8838	14.91	587.7	25.5	0.2095	0.00811	0.05963
8435402	M	0.1003	0.4	0.2364	22.54	1573	18.67	0.1374	0.01149	0.01798
843786	M	0.1278	0.5355	0.3895	19.47	741.8	23.79	0.1791	0.00791	0.02185
844358	M	0.08465	0.3784	0.3065	22.88	1696	27.88	0.1442	0.004314	0.01398
8448802	M	0.1189	0.2675	0.3190	17.06	697	28.14	0.1654	0.008905	0.01485
844961	M	0.1273	0.539	0.4378	19.49	739.3	39.73	0.1703	0.005731	0.02143
84581001	M	0.1188	1.100	0.4368	19.09	711.4	48.65	0.1693	0.007149	0.01789
845836	M	0.08206	0.1459	0.2946	18.19	1159	31.88	0.1181	0.004029	0.0148
84610002	M	0.0871	0.3905	0.3792	20.42	1299	27.28	0.1395	0.005771	0.02098
846226	M	0.0874	0.3639	0.3178	20.96	1332	29.94	0.1037	0.003139	0.04494
846361	M	0.08401	0.2322	0.2809	18.54	878.5	27.60	0.1131	0.009768	0.02981
84687401	M	0.1131	0.6943	0.3596	15.03	587.7	32.01	0.1851	0.006428	0.01981
84789002	M	0.1130	0.7026	0.4218	17.46	943.2	37.13	0.1679	0.008607	0.01857
848406	M	0.08667	0.2914	0.3029	19.07	1138	30.88	0.1404	0.005718	0.0141
84882001	M	0.117	0.4764	0.3706	20.96	1315	31.48	0.1789	0.007026	0.01889
849014	M	0.08631	0.5372	0.2768	27.32	2398	35.88	0.1512	0.006494	0.01358
8510425	B	0.09779	0.239	0.2977	15.11	711.2	18.28	0.144	0.006462	0.0198
8510003	B	0.1075	0.199	0.3184	14.2	339.5	20.49	0.1312	0.004027	0.01678
8510624	B	0.1024	0.08667	0.245	18.23	314.9	15.00	0.1324	0.006606	0.02027
8511135	M	0.1073	0.6205	0.4667	18.07	888.9	19.88	0.139	0.006788	0.03672
851300	M	0.08425	0.3155	0.2822	28.17	2615	35.09	0.1401	0.004728	0.01983
852552	M	0.1121	0.4695	0.3613	26.46	2215	31.56	0.1802	0.006048	0.01484
852631	M	0.1188	0.3803	0.4068	22.25	1461	21.4	0.1545	0.008029	0.02398
852761	M	0.1094	0.4068	0.4068	27.69	888.9	31.91	0.1504	0.006473	0.01484

Fig 5.13: selected features using SBE.

Code:

```
X=data.iloc[:,2:]

Y=data.iloc[:,1].values

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

sc=StandardScaler()

X_train=sc.fit_transform(X_train)
```

```
X_test=sc.fit_transform(X_test)

clf = RandomForestClassifier(n_estimators=100, random_state=1, max_depth=4,verbose
=2)
```

Fig 5.14: Splitting the data and Constructing model.

Displaying Training and Testing dataset:

Training dataset					
	smoothness_mean	concavity_worst	...	smoothness_se	symmetry_se
562	0.10480	1.17000	...	0.004625	0.02137
291	0.08992	0.18040	...	0.005332	0.01522
16	0.09867	0.29140	...	0.005718	0.01410
546	0.09434	0.04384	...	0.007086	0.01560
293	0.08372	0.13160	...	0.007595	0.01792
..
129	0.10150	0.56730	...	0.005033	0.01578
144	0.07793	0.09755	...	0.006547	0.01671
72	0.10710	0.65660	...	0.005820	0.01527
235	0.09070	0.06231	...	0.007389	0.01263
37	0.08983	0.04833	...	0.004352	0.02671
[426 rows x 9 columns]					
Testing dataset					
	smoothness_mean	concavity_worst	...	smoothness_se	symmetry_se
421	0.10310	0.32190	...	0.009976	0.02653
47	0.11580	0.50060	...	0.006532	0.01743
292	0.10050	0.22410	...	0.008725	0.02625
186	0.08588	0.35380	...	0.002866	0.01069
414	0.08320	0.15470	...	0.006831	0.03151
..
232	0.07780	0.02318	...	0.004359	0.01916
413	0.08515	0.31140	...	0.004449	0.01906
514	0.09215	0.28660	...	0.004952	0.01152
244	0.10270	0.34580	...	0.010610	0.02186
415	0.09773	0.11640	...	0.009895	0.02258
[143 rows x 9 columns]					

Fig 5.15 : Training and Testing dataset.

5.2.4 Predictions and Metrics:

Prediction refers to the output of an algorithm after it has been trained on a dataset and applied to new data when forecasting the likelihood of a particular outcome, such as a patient has a cancer or not.

Function used for prediction:

Predict());

```
y_train_pred = clf.predict(X_train)
```


Metrics:**Accuracy:**

Accuracy is one metric for evaluating classification models. It is the Number of correct predictions.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}.$$

Confusion Matrix: A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

Precision :

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{TP}{TP+FP}.$$

Recall (Sensitivity) :

Recall is the ratio of correctly predicted positive observations to the all observations in actual class

$$\text{Recall} = \frac{TP}{TP+FN}.$$

F1 score :

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false

negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

F1 Score = $2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$.

```
cm=confusion_matrix(Y_test,clf.predict(X_test))

tn=cm[0][0]

tp=cm[1][1]

fn=cm[1][0]

fp=cm[0][1]

print("Confusion matrix: ")

cm=tp+tn/tp+tn+fn+fp.

print("Testing Accuracy"cm)

print("Precision of testing data=",tp/(tp+fp))

print("Recall of testing data=",tp/(tp+fn))

P=tp/(tp+fp)

R=tp/(tp+fn)

print("F1 score of testing data=",2*((P*R)/(P+R)))
```

Fig 5.16: Metrics.

5.2.4.1 Predictions and metrics for SFS :

```

y_train_pred = clf.predict(X_train)
print('Training accuracy of Forward feature selection features: %.3f' % acc(Y_train, y_train_pred))
y_test_pred = clf.predict(X_test)
print('Testing accuracy of Forward feature selection features: %.3f' % acc(Y_test, y_test_pred))
print("Expected Output:")
print(Y_test)
print("Actual Output:")
print(y_test_pred)

Training accuracy of Forward feature selection features: 0.988
Testing accuracy of Forward feature selection features: 0.958
Expected Output:
['B' 'M' 'B' 'M' 'M' 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'M' 'M' 'M' 'B' 'M' 'M' 'B' 'B'
 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'B'
 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B'
 'M' 'M' 'B' 'M' 'M' 'M' 'B' 'M' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B'
 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B'
 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'M'
 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B']
Actual Output:
['B' 'M' 'B' 'M' 'M' 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'M' 'M' 'M' 'B' 'M' 'M' 'B' 'B'
 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'B'
 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B'
 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B'
 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'M'
 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'M']

```

Fig 5.17 : Prediction of SFS

```

Confusion matrix:
[[86  2]
 [ 4 51]]
Precision of testing data= 0.9622641509433962
Recall of testing data= 0.9272727272727272
F1 score of testing data= 0.9444444444444444

```

Fig 5.18 :Metric of SFS.

5.2.4.2 Prediction and metrics for SBS

```

y_train_pred = clf.predict(X_train)
print('Training accuracy: %.3f' % acc(Y_train, y_train_pred))
y_test_pred = clf.predict(X_test)
print('Testing accuracy: %.3f' % acc(Y_test, y_test_pred))
print("Expected Output:")
print(Y_test)
print("Actual Output:")
print(y_test_pred)

```

```

Training accuracy: 0.993
Testing accuracy: 0.951
Expected Output:
['B' 'M' 'B' 'M' 'M' 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'M' 'M' 'B' 'M' 'M' 'M' 'B' 'B'
 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'B'
 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B'
 'M' 'M' 'B' 'M' 'M' 'M' 'B' 'M' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B'
 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B'
 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'M'
 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' ]
Actual Output:
['B' 'M' 'B' 'M' 'M' 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'M' 'M' 'M' 'B' 'M' 'M' 'B' 'B'
 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'B'
 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B'
 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'M'
 'M' 'M' 'M' 'M' 'B' 'M' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'M' 'M' 'M' 'B' ]

```

Fig 5.19 : Prediction of SBE.

```

confusion matrix
[[85  3]
 [ 4 51]]
Precision of testing data= 0.9444444444444444
Recall of testing data= 0.9272727272727272
F1 score of testing data= 0.9357798165137615

```

Fig 5.20: Metrics of SBE.

CHAPTER 6

TESTING

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

Testing is a process of executing a program with the intent of finding an error.

- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if it exists.

The first approach is what known as Black box testing and the second approach is White box testing. We apply white box testing techniques to ascertain the functionalities top-down and then we use black box testing techniques to demonstrate that everything runs as expected.

Black-Box Testing:

This technique of testing is done without any knowledge of the interior workings of the application. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining the outputs without knowing how and where the inputs are worked upon.

- Well suited and efficient for large code segments
- Code access is not required
- Clearly separates user's perspectives from the developer's perspective through visibly defined roles

White-Box Testing:

White-box testing is the detailed investigation of internal logic and structure of the code. It is also called “glass testing” or “open-box testing”. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code.

The tester needs to look inside the source code and find out which part of the code is working inappropriately.

In this, the test cases are generated on the logic of each module. It has been used to generate the test cases in the following cases:

- Guarantee that all independent modules have been executed.
- Execute all logical decisions and loops.
- Execute through proper plots and curves.

Performance Evaluation

This project has been successfully executed its source code. Initially there were some errors in the code. By resolving them, the code is fully free from errors and bugs.

After performing feature selection models are built using different classifiers. The classifier that gives best possible accuracy has been considered in this project. The classifier used to build a model in this project is Random forest classifier which gave the effective results. Hence, for the considered dataset, the Random forest classifier performed well with the accuracy of 95.8.

CONCLUSION

Cancer has been portrayed as a heterogeneous disease comprising of a wide range of subtypes. The early diagnosis of a cancer type is very important to determine the course of medical treatment required by the patient. The significance of classifying cancerous cells into benign or malignant has driven many research studies. In the past years researchers have been encouraged to use different machine learning (ML) techniques for cancer detection, as well as prediction of survivability and recurrence. In this project we successfully developed a model that can classify the tumours into malignant and benign.

Here we have used machine learning algorithm like Random Forest Algorithm to classify cancer patients and detect the type of cancer. Given a few parameters, our algorithms can predict whether the patient has malignant cancer or benign cancer and we have applied two different feature selection techniques namely sequential forward selection and sequential backward elimination with Random Forest algorithm to cancer dataset and sequential forward selection with the selected algorithm was found to provide the most effective results. Proper subset of features was found which was crucial in detecting malignancy.

Integration of multi dimensional features can give more effective tools for detection of cancer. Other machine learning models like support vector machine, other models of neural networks (CNN or ANN) could be implemented. Other learning algorithms can be applied with using our chosen set of features. A dataset with more number of examples can be used.

REFERENCES

JOURNALS

- [1] Sreyam Dasgupta, Ronit Chaudhuri, Swarnalatha Purushotham “Feature Selection for Breast Cancer Detection using Machine Learning Algorithms”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-9, July 2019.
- [2] HabibDhahri Eslam Al Maghayreh, Awais Mahmood,Wail Elkilani & Mohammed Faisal Nagi “Automated Breast Cancer Diagnosis Based on Machine Learning Algorithms” Journal of Health Care Industry Volume 2019,Article ID 4253641,11 pages.

CONFERENCE PAPERS

- [3] Bin Dai ; Rung-Ching ; Chen Shun-Zhi Zhu ; Wei-Wei Zhang “Using Random Forest Algorithm for Breast Cancer Diagnosis”, 2018 IEEE International Symposium on Computer, Consumer and Control (IS3C).
- [4] Shubham Sharma ; Archit Aggarwal ; Tanupriya Choudhury “Breast Cancer Detection Using Machine Learning Algorithms”, 2018 IEEE International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS).
- [5] Hiba Asria ; Hajar Mousannif ; Hassan Al Moatassime ; Thomas Noel “ Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis”, Procedia Computer Science Volume 83, 2016, Pages 1064-1069.