Breast Cancer Prediction Using Machine Learning With Decision Tree Algorithm

A project report submitted in partial fulfilment of the requirement for the Award of the Degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

by

Tauseef Ali Khan (160719733003)

Santosh Naga Manideep .B (160719733002)

CH Upendra (1607197330019)

Under the Guidance of

Mr. D. Raja Shekar, Assistant Professor, Dept. of CSE

Department of Computer Science and Engineering

Department of Computer Science and Engineering

DECLARATION BY THE CANDIDATES

We, **Tauseef Ali Khan(160719733003), Santosh Naga Manideep (160719733002) and CH Upendra(160719733019**) students of Methodist College of Engineering and Technology, pursuing Bachelor's degree in Computer Science and Engineering, hereby declare that this project report entitled "**Breast Cancer Prediction Using Machine Learning With Decision Tree Algorithm",** carried out under the guidance of **Mr. D. Raja Shekar** submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science. This is a record work carried out by us and the results embodied in this project have not been reproduced/copied from any source.

Tauseef Ali Khan (160719733003)

Santosh Naga Manideep (160719733002)

CH Upendra (160719733019)

Department of Computer Science and Engineering

CERTIFICATE BY THE SUPERVISOR

This is to certify that this project report entitled "**Breast Cancer Prediction Using Machine Learning With Decision Tree Algorithm**" being submitted *by* Tauseef Ali Khan (160719733003), Santosh Naga Manideep.B(160719733002) and CH Upendra (160719733019), submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering, during the academic year 2018, is a bonfide record of work carried out by them.

Mr. D. Raja Shekar
Assistant Professor,

Date:

Department of Computer Science and Engineering

CERTIFICATE BY THE HEAD OF THE DEPARTMENT

This is to certify that this project report entitled "Breast Cancer Prediction Using Machine Learning With Decision Tree Algorithm" BY **Tauseef Ali Khan** (160719733003), **Santosh Naga Manideep.B**(160719733002), **CH Upendra**(160719733019), submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering of the Osmania University, Hyderabad, during the academic year 2017-2018, is a bonafide record of work carried out by them.

Mrs. P. Lavanya

H.O.D

DATE:

ACKNOWLEDGEMNTS

We would like to express our sincere gratitude to my project guide **Mr. D. Raja Shekar Assistant Professor,** for giving us the opportunity to work on this topic. It would never be possible for us to take this project to this level without his innovative ideas and his relentless support and encouragement.

We would like to thank our project coordinator **Mrs. Sowjanya, Assistant Professor,** who helped us by being an example of high vision and pushing towards greater limits of achievement.

Our sincere thanks to **Mrs. P. Lavanya, Associate Professor** and **Head of the Department of Computer Science and Engineering,** for her valuable guidance and encouragement which has played a major role in the completion of the project and for helping us by being an example of high vision and pushing towards greater limits of achievement.

We would like to express a deep sense of gratitude towards the **Dr. G. Ravinder Reddy, Principal, Methodist College of Engineering and Technology,** for always being an inspiration and for always encouraging us in every possible way.

We would like to express a deep sense of gratitude towards the **Dr. Lakshmipathi Rao, Director, Methodist College of Engineering and Technology,** for always being an inspiration and for always encouraging us in every possible way.

We are indebted to the Department of Computer Science & Engineering and Methodist College of Engineering and Technology for providing us with all the required facility to carry our work in a congenial environment. We extend our gratitude to the CSE Department staff for providing us to the needful time to time whenever requested.

We would like to thank our parents for allowing us to realize our potential, all the support they have provided us over the years was the greatest gift anyone has ever given us and also for teaching us the value of hard work and education. Our parents has offered us with tremendous support and encouragement, thanks to our parents for all the moral support and the amazing opportunities they have given us over the years.

ABSTRACT

According to the world health organization (WHO) Breast cancer is the most frequent cancer among women, impacting 2.1 million women each year, and also causes the greatest number of cancer related deaths among women. In 2018, it is estimated that 627,000 women died from breast cancer – that is approximately 15% of all cancer deaths among women. While breast cancer rates are higher among women in more developed regions, rates are increasing in nearly every region globally. In order to improve breast cancer outcomes and survival, early detection is critical. There are two early detection strategies for breast cancer: early diagnosis and screening. Limited resource settings with weak health systems where the majority of women are diagnosed in late stages should prioritize early diagnosis programs based on awareness of early signs and symptoms and prompt referral to diagnosis and treatment. Early diagnosis strategies focus on providing timely access to cancer treatment by reducing barriers to care and/or improving access to effective diagnosis services. The goal is to increase the proportion of breast cancers identified at an early stage, allowing for more effective treatment to be used and reducing the risks of death from breast cancer. Since early detection of cancer is key to effective treatment of breast cancer , we use various machine learning algorithms to predict if a tumor is benign or malignant, based on the features provided by the data.

Table of contents

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Breast cancer (BC) is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society. The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of benign tumors can prevent patients undergoing unnecessary treatments. Thus, the correct diagnosis of BC and classification of patients into malignant or benign groups is the subject of much research. Because of its unique advantages in critical features detection from complex BC datasets, machine learning (ML) is widely recognized as the methodology of choice in BC pattern classification and forecast modeling.

Breast cancer is a malignant cell growth in the breast. If left untreated, the cancer spreads to other areas of the body. Excluding skin cancer, breast cancer is the most common type of cancer in women in the United States, accounting for one of every three cancer diagnoses.

An estimated 211,240 new invasive cases of breast cancer were expected to occur among women in the United States during 2005. About 1,690 new male cases of breast cancer were expected in 2005.

The incidence of breast cancer rises after age 40. The highest incidence (approximately 80% of invasive cases) occurs in women over age 50.

In addition to invasive breast cancer, 58,590 new cases of in situ breast cancer are expected to occur among women during 2005. Of these, approximately 88% will be classified as ductal carcinoma in situ (DCIS). The detection of DCIS cases is a direct result of the increased use of mammography screening. This screening method is also responsible for detection of invasive cancers at a less advanced stage than might have occurred otherwise.

An estimated 40,870 deaths (40,410 women, 460 men) were anticipated from breast cancer in 2005. Breast cancer ranks second among cancer deaths in women. According to the most recent data, mortality rates declined significantly during 1992-1998, with the largest decreases in younger women, both white and black.

1.1 Some Risk Factors for Breast Cancer

The following are some of the known risk factors for breast cancer. However, most cases of breast cancer cannot be linked to a specific cause. Talk to your doctor about your specific risk.

- Age. The chance of getting breast cancer increases as women age. Nearly 80 percent of breast cancers are found in women over the age of 50.
- Personal history of breast cancer. A woman who has had breast cancer in one breast is at an increased risk of developing cancer in her other breast.
- Family history of breast cancer. A woman has a higher risk of breast cancer if her mother, sister or daughter had breast cancer, especially at a young age (before 40). Having other relatives with breast cancer may also raise the risk.
- Genetic factors. Women with certain genetic mutations, including changes to the BRCA1 and BRCA2 genes, are at higher risk of developing breast cancer during their lifetime. Other gene changes may raise breast cancer risk as well.
- Childbearing and menstrual history. The older a woman is when she has her first child, the greater her risk of breast cancer. Also at higher risk are:

1. Women who menstruate for the first time at an early age (before 12)

2. Women who go through menopause late (after age 55)

3. Women who've never had children.

1.2 Role of Machine Learning In Detection of Breast Cancer

A mammogram is an x-ray picture of the breast. It can be used to check for breast cancer in women who have no signs or symptoms of the disease. It can also be used if you have a lump or other sign of breast cancer. Screening mammography is the type of mammogram that checks you when you have no symptoms. It can help reduce the number of deaths from breast cancer among women ages 40 to 70. But it can also have drawbacks. Mammograms can sometimes find something that looks abnormal but isn't cancer. This leads to further testing and can cause you anxiety. Sometimes mammograms can miss cancer when it is there. It also exposes you to radiation. You should talk to your doctor about the benefits and drawbacks of mammograms. Together, you can decide when to start and how often to have a mammogram.

Now while its difficult to figure out for physicians by seeing only images of x-ray that weather the tumor is toxic or not training a machine learning model according to the identification of tumour can be of great help.

2. LITERATURE SURVEY

Twenty-four recent research articles have been reviewed to explore the computational methods to predict breast cancer. The summaries of them are presented below. Chaurasia et al. developed prediction models of benign and malignant breast cancer. Wisconsin breast cancer data set was used. The dataset contained 699 instances, two classes (malignant and benign), and nine integer valued clinical attributes such as uniformity of cell size. The researchers removed the 16 instances with missing values from the data set to become the data set of 683 instances. The benign were 458 (65.5%) and malignant were 241 (34.5%). The experiment was analysed by the Waikato Environment for Knowledge Analysis (WEKA). Naive Bayes, RBF Network, and J48 are the three most popular data mining algorithms were used to develop the prediction models.

The researchers used 10- fold cross-validation methods to measure the unbiased estimate of the three prediction models for performance comparison purposes. The models' performance evaluation was presented based on the methods‟ effectiveness and accuracy. Experimental results showed that the Naive Bayes had gained the best performance with a classification accuracy of 97.36%; followed by RBF Network with a classification accuracy of 96.77% and the J48 was the third with a classification accuracy of 93.41%. In addition, the researchers conducted sensitivity analysis and specificity analysis of the three algorithms to gain insight into the relative contribution of the independent variables to predict survival. The sensitivity results indicated that the prognosis factor „„Class‟‟ was by far the most important predictor.

2.1 DATA MINING AND MACHINE LEARNING

The term "data mining" is a misnomer, because the goal is the extraction of patterns and knowledge from large amounts of data, not the extraction (mining) of data itself. It also is a buzzword and is frequently applied to any form of large-scale data or information processing (collection, extraction, warehousing, analysis, and statistics) as well as any application of computer decision support system, including artificial intelligence (e.g., machine learning) and business intelligence. The book Data mining: Practical machine learning tools and techniques with Java[8] (which covers mostly machine learning material) was originally to be named just Practical machine learning, and the term data mining was only added for marketing reasons. Often the more general terms (large scale) data analysis and analytics – or, when referring to

actual methods, artificial intelligence and machine learning – are more appropriate. In this project we use the following machine learning algorithms:

Decision tree algorithms: Decision tree algorithms are successful machine learning classification techniques. They are the supervised learning methods which use information gained and pruned to improve results. Moreover, decision tree algorithms are commonly used for classification in many research, for example, in the medicine area and health issues. There are many kinds of decision tree algorithms such as ID3 and C4.5. However J48 is the most popular decision tree algorithm. J48 is the implementation of an improved version of C4.5 and isan extension of ID3.

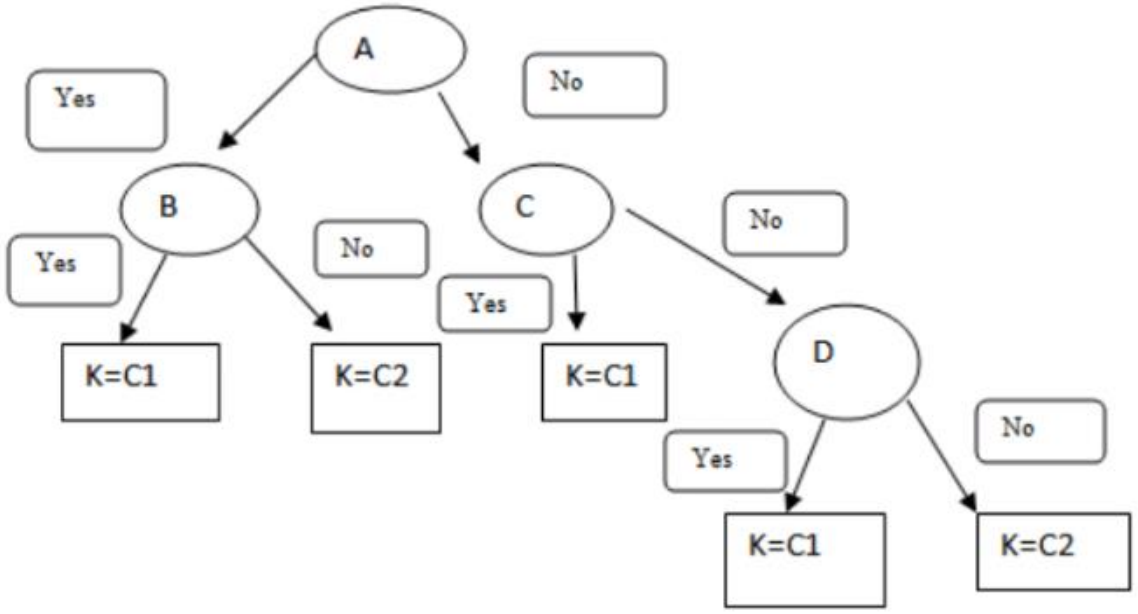


Figure 2.1 decision tree classifier

Aruna et al. used naïve Bayes, support vector machine, and decision trees to classify a Wisconsin breast cancer dataset and got the best result by using support vector machine (SVM) with an accuracy score of 96.99%. Chaurasia et al. compared the performance of supervised learning classifiers by using a Wisconsin breast cancer dataset and naïve Bayes, SVM, neural networks, decision tree methods applied. According to the study results, SVM gave the most accurate result with a score of 96.84%. Asri et al. [24] also used the same data and made a performance comparison among machine learning algorithms: SVM, decision tree (c4.5), naïve Bayes, and k-nearest neighbours. The study aimed to classify data in terms of efficiency and effectiveness by comparing the accuracy, precision, sensitivity, and specificity of each algorithm. The experimental result showed that SVM had the best score with an accuracy of 97.13%. Delen et al. [25] studied the prediction of breast cancer data with 202,932 patient records. The dataset was divided into two different groups as survived (93,273) and not survived (109,659), then naïve Bayes, neural network, and c4.5 decision tree algorithms were applied. The achieved results showed that the c4.5 decision tree had better performance than the other techniques.

3.SYSTEM DESIGN ANALYSIS

3.1 System architecture:

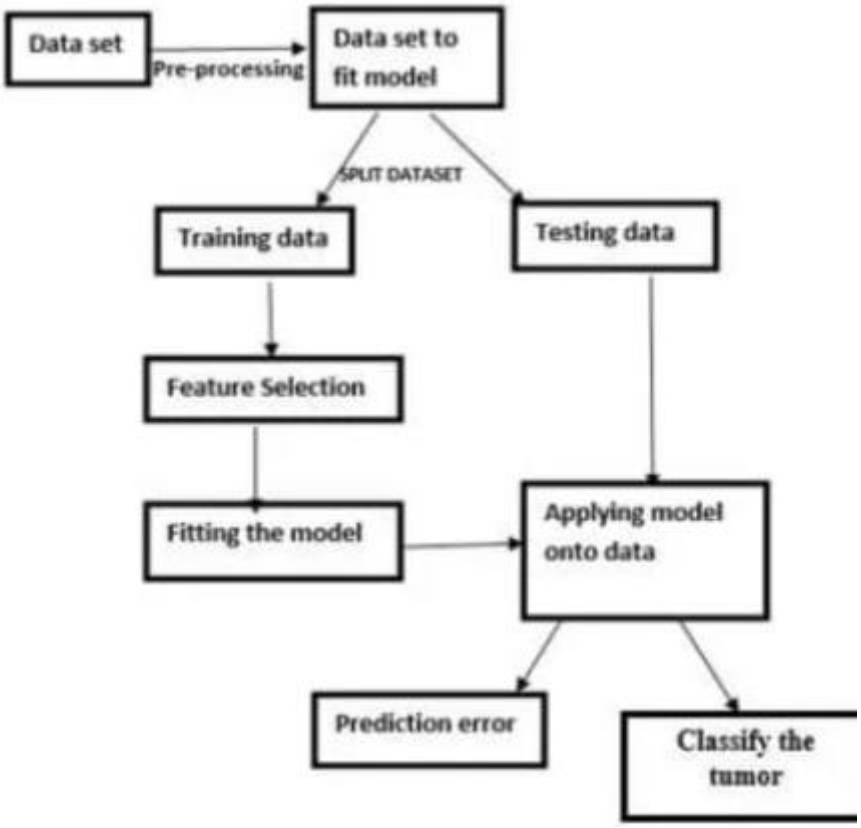


Figure 3.1: system architecture

3.2 UML diagrams

UML stands for unified modeling language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standardized is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of objectoriented computer software. UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or association with UML. It is important to distinguish between the UML model and the set of diagrams of a system. A diagram is a partial graphic representation of a system's model. The set of diagrams need not completely cover the model and deleting a diagram does not change the model. The model may also contain documentation that drives the model elements and diagrams. In 1997 UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005 UML was also published by the International Organization for Standardization (ISO) as an approved ISO standard. Since then it has been periodically revised to cover the latest revision of UML.

UML is not a development method by itself; however, it was designed to be compatible with the leading object-oriented software development methods of its time, for example OMT, Booch method, Objectory and especially RUP that it was originally intended to be used with when work began at Rational Software. UML (Unified Modeling Language) is a standard notation for the modeling of real-world objects as a first step in developing an object-oriented design methodology. The major perspectives of a UML are Design, Implementation, Process and Deployment. The center is the Use Case view which connects all these four.

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Activity Diagram

3.3 Use Case diagram

A Use case represents the functionality of the system.

• Use Case diagrams in the UML is a type of behaviour diagrams defined by and created from a Use-case analysis.

• It is used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).

• The main purpose of a Use case diagram is to show what system functions are performed for which actor.

• Use case diagrams are drawn to capture the functional requirements of a system.

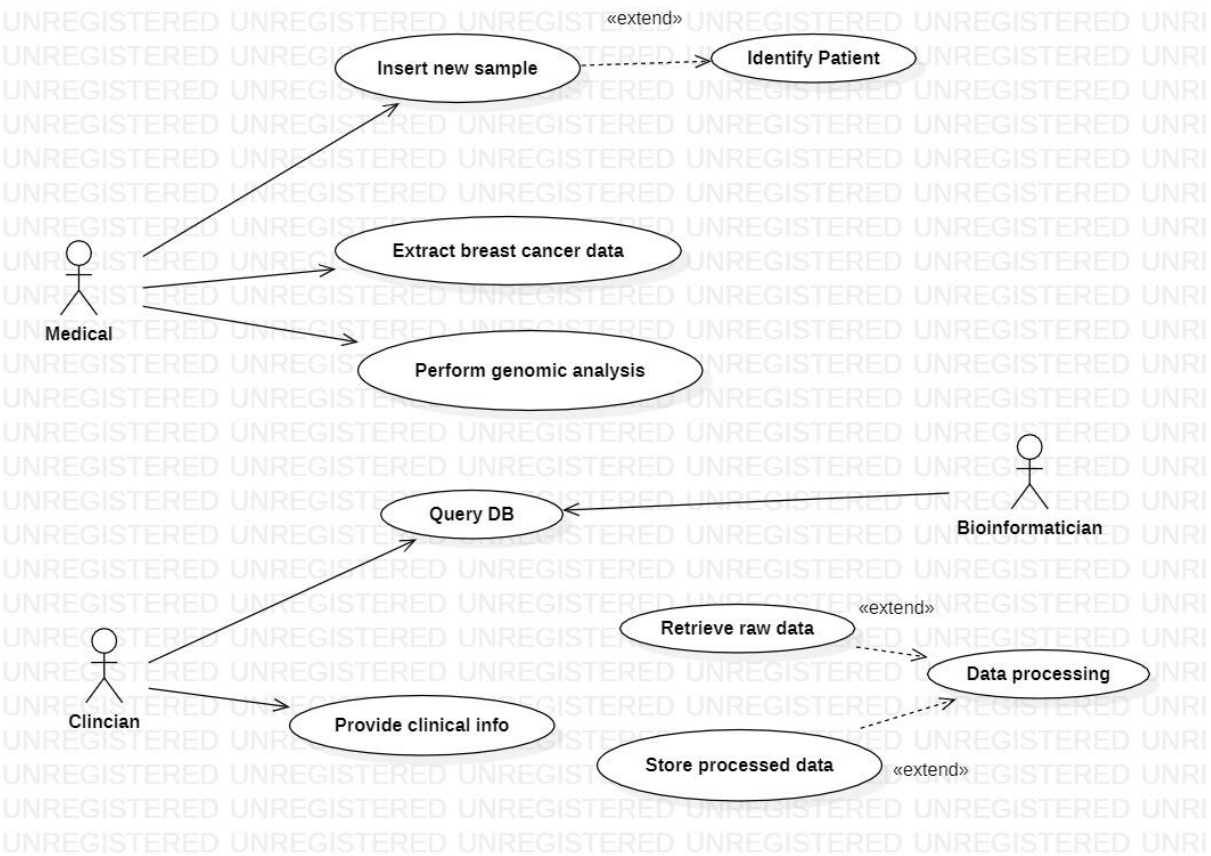


Figure 3.3 Use Case diagram

3.4 Class diagram

Class Diagram is a type of static structure diagram that describes the structure of a system.

• It shows the system's classes, their attributes, operations (or methods), and the relationships among objects.

• It explains which class contains information.

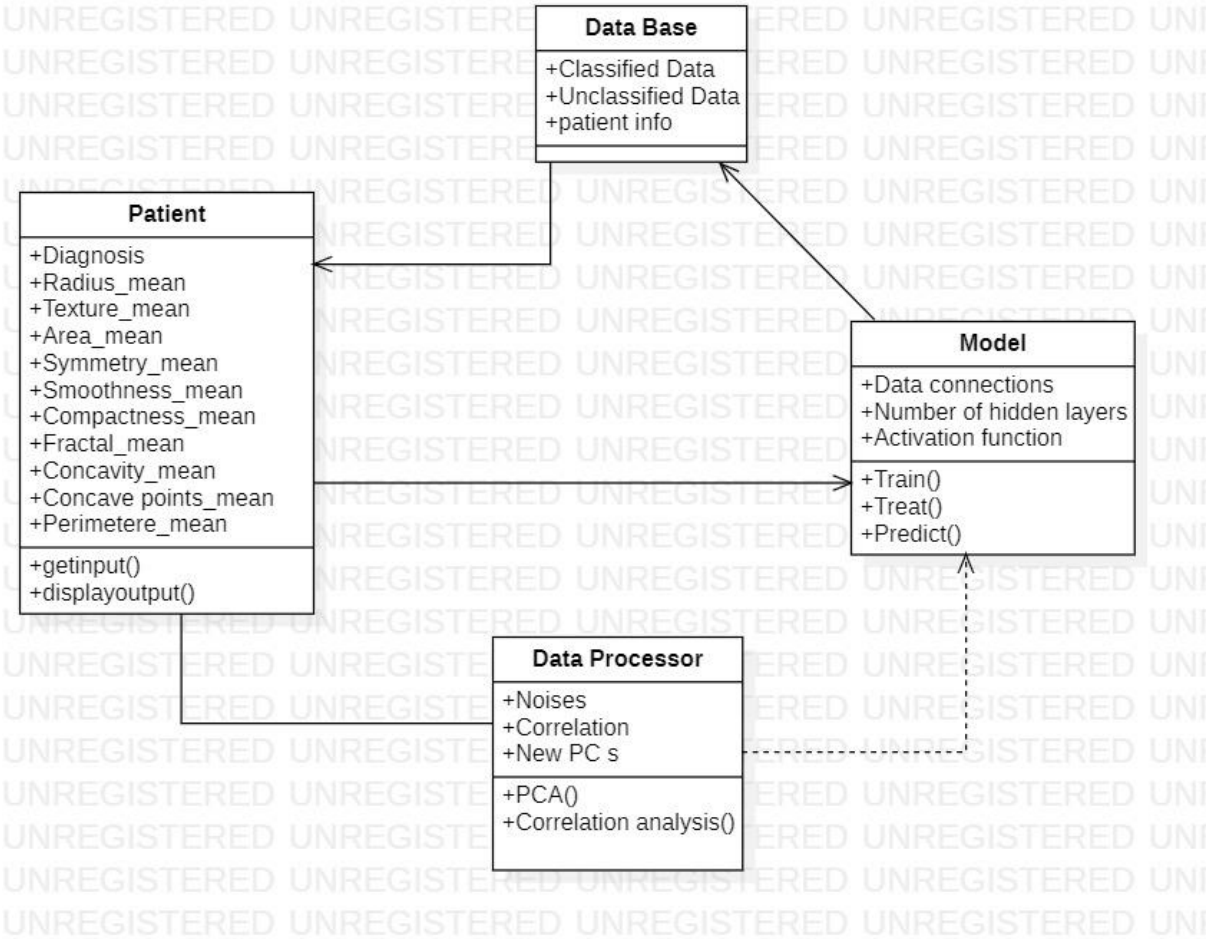


figure 3.4: Class diagram

3.5 Sequence diagram

• A sequence diagram shows object interactions arranged in time sequence.

• It depicts objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

• Sometimes called as event diagrams, event scenarios, timing diagram.

• A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order

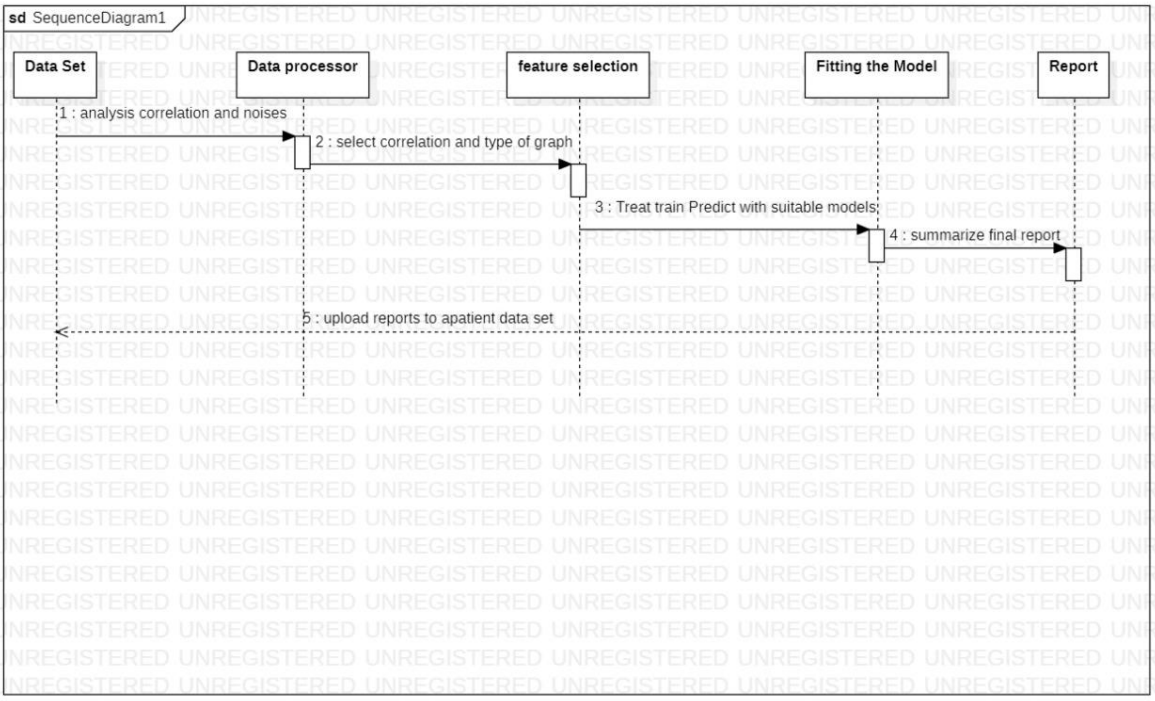


figure 3.5: Sequence diagram

3.6 Activity diagram

Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. It can be used to describe the business and operational step-by-step workflows of components in the system.

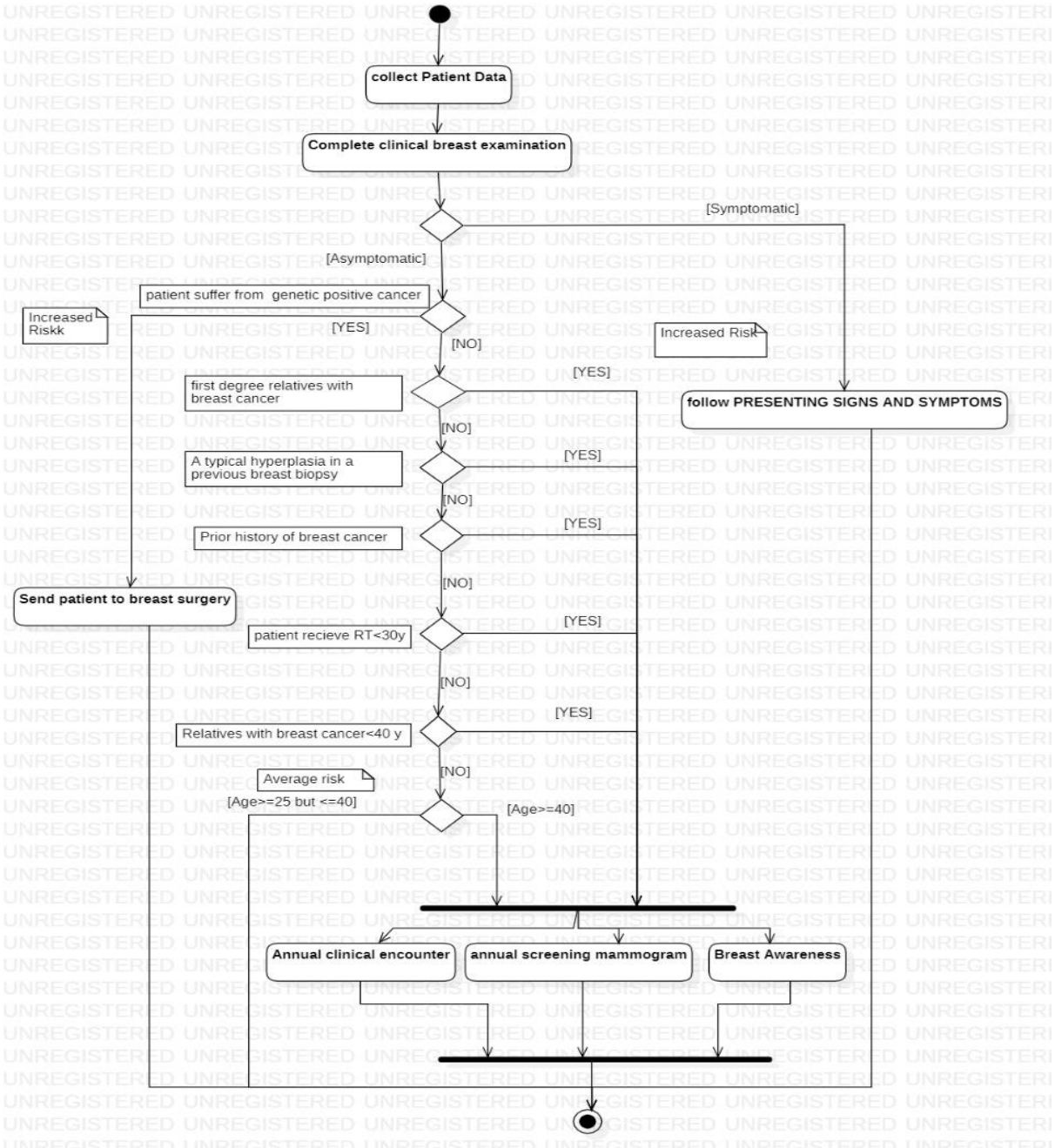


figure 3.6 activity diagram

4.IMPLEMENTATION

4.1 MODULES

This project has the following modules:

- Data exploration.
- Training and Testing.
- Prediction Model.

4.2 MODULES DESCRIPTION

4.2.1 Data Exploration

Exploring data sets and developing deep understanding about the data is one of the most important skills every data scientist should possess. People estimate that the time spent on these activities can go as high as 80% of the project time in some cases.

Python has been gaining a lot of ground as preferred tool for data scientists lately, and for the right reasons. Ease of learning, powerful libraries with integration of C/C++, production readiness and integration with web stack are some of the main reasons for this move lately.

Input data sets can be in various formats (.XLS, .TXT, .CSV, JSON). In Python, it is easy to load data from any source, due to its simple syntax and availability of predefined libraries, such as Pandas. Here I will make use of Pandas itself.

Pandas features a number of functions for reading tabular data as a Pandas DataFrame object. Below are the common functions that can be used to read data

We will first go with importing the necessary libraries and import our dataset to colab.research.google.com. We can examine the data set using the pandas'

Then we refine the data by removing unwanted noise and redundancy. The required data set for data prediction is acquired.

Next the graphical visualisation of data set is acquired by python libraries such as matplot and seaborn. The graphical representation of malignant and benign tumors are compared with a bar graph respectively.

Followed by this the correlation between all the attributes is visualised in a graphical format and a heat map accordingly.

We can examine the data set using the pandas' head() method

Sample code:

```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#Load the data
from google.colab import files
# Use to load data on Google Colab #
uploaded = files.upload()
# Use to load data on Google Colab
df = pd.read_csv('data.csv')
df.head(20)
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving data.csv to data.csv

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883
5	843786	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.15780	0.08089	0.2087	0.07613
6	844359	M	18.25	19.98	119.60	1040.0	0.09463	0.10900	0.11270	0.07400	0.1794	0.05742

We can find the dimensions of the data set using the panda dataset 'shape' attribute.

```
#Count the number of rows and columns in the data set
df.shape
```

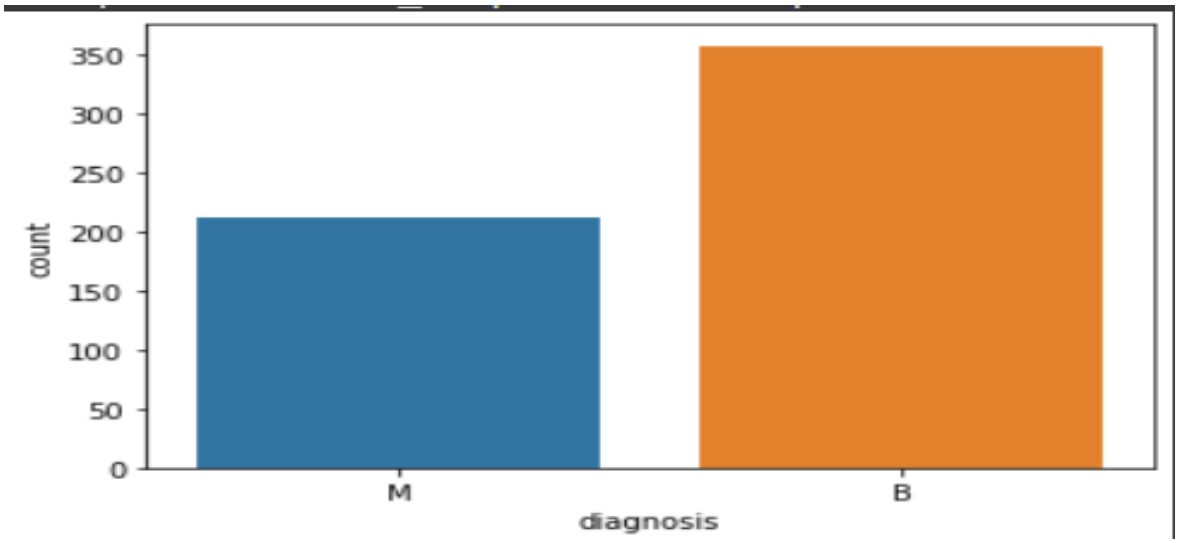
```
(569, 33)
```

We can observe that the data set contain 569 rows and 33 columns. ‘Diagnosis’ is the column which we are going to predict , which says if the cancer is M = malignant or B = benign. 1 means the cancer is malignant and 0 means benign. We can identify that out of the 569 persons, 357 are labeled as B (benign) and 212 as M (malignant). Each row represents a patient and 33 features on the 569 patients. The last column Unnamed 32 has NaN values so we need to remove that column with empty values. So we count the number of empty values.

```
#Count the empty (NaN, NAN, na) values in each column
df.isna().sum()
```

```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
concavity_worst            0
concave points_worst       0
symmetry_worst             0
fractal_dimension_worst    0
Unnamed: 32              569
dtype: int64
```

So column Unnamed: 32 has 569 missing values so we drop it. So the new shape of the data is (569, 32) which means 569 rows and 32 columns. Now we can see the number of Malignant (M) (harmful) or Benign (B) cells (not harmful) cells and plot it in a graph.



Now we will see what the data types of the columns are:

```
#Look at the data types
df.dtypes
```

```
id                         int64
diagnosis                 object
radius_mean              float64
texture_mean             float64
perimeter_mean           float64
area_mean                float64
smoothness_mean          float64
compactness_mean         float64
concavity_mean           float64
concave points_mean      float64
symmetry_mean            float64
fractal_dimension_mean   float64
radius_se                float64
texture_se               float64
perimeter_se             float64
area_se                  float64
smoothness_se            float64
compactness_se           float64
concavity_se             float64
concave points_se        float64
symmetry_se              float64
fractal_dimension_se     float64
radius_worst             float64
texture_worst            float64
perimeter_worst          float64
area_worst               float64
smoothness_worst         float64
compactness_worst        float64
concavity_worst          float64
concave points_worst     float64
symmetry_worst           float64
fractal_dimension_worst  float64
dtype: object
```

We can see that id column acts as the identifier of the patient and it is of integer type and it cannot be used as a feature to predict the tumor. Next we encode categorical data values (Transforming categorical data/ Strings to integers)

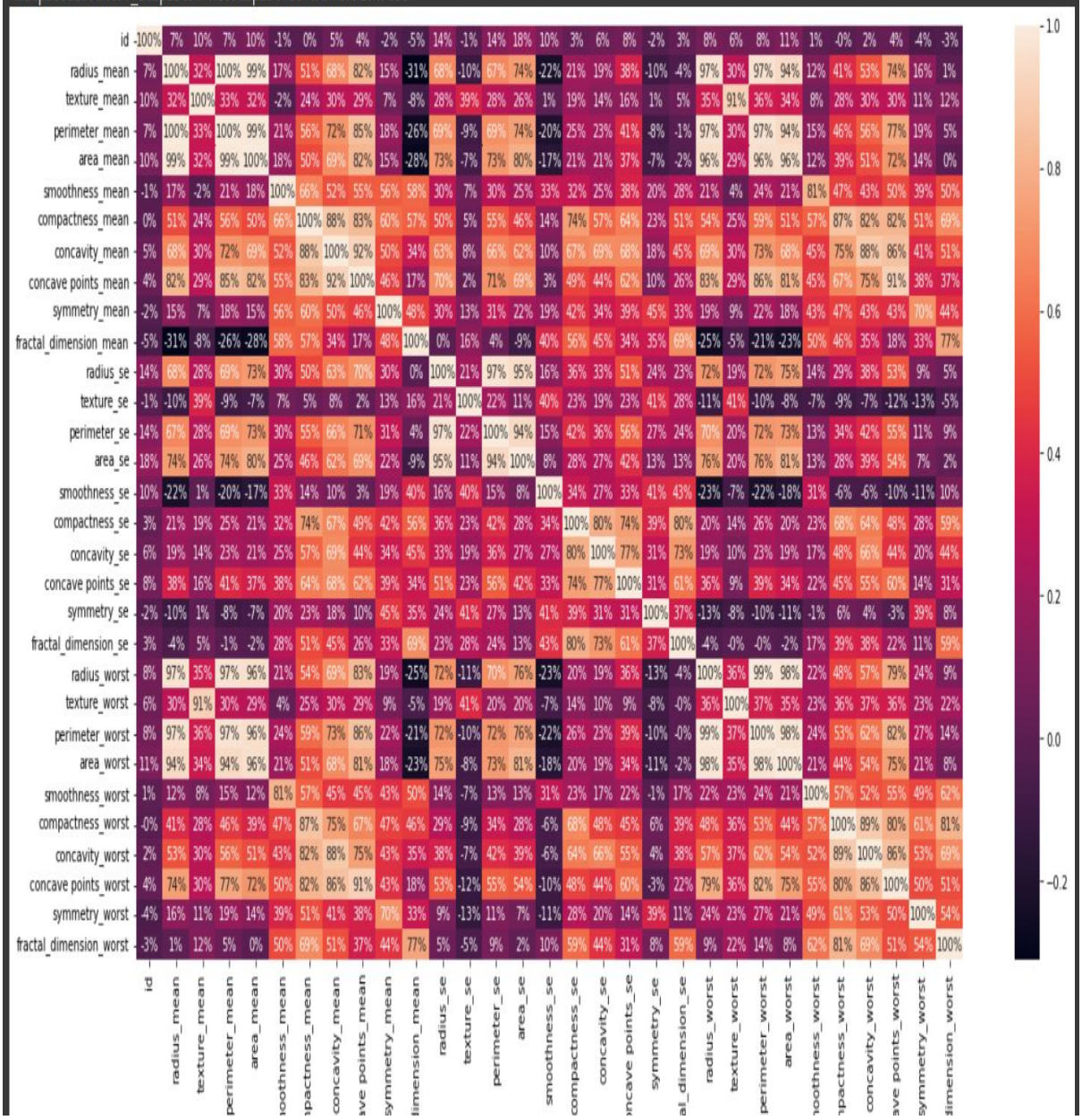
```
#Encoding categorical data values (
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
df.iloc[:,1]= labelencoder_Y.fit_transform(df.iloc[:,1].values)
print(labelencoder_Y.fit_transform(df.iloc[:,1].values))
```

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

Here the value 1 represents Malignant (M) (harmful) and value 0 represents Benign (B) cells (not harmful) cells. Now we visualize a correlation between the different attributes.

```
plt.figure(figsize=(20,10))
sns.heatmap(df.corr(), annot=True, fmt='.0%')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdf9a047c50>
```



4.2.2 Training and testing

Next we split the datasets into independent (X) and dependent (Y) datasets. The dependent data set(Y) has the diagnosis whether the patient has cancer and the independent dataset(X) has the features that are used to predict the outcome. Now we split the dataset into 75% training and 25%testing and use decision tree classifier to the training set. Now we print the accuracy on the training data.

We will now predict the test set results and check the accuracy with each of our model: To check the accuracy we need to import confusion_matrix method of metrics class. The confusion matrix is a way of tabulating the number of mis-classifications, i.e., the number of predicted classes which ended up in a wrong classification in based on the true classes.

```
X = df.iloc[:, 2:31].values
Y = df.iloc[:, 1].values


from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25, random_state = 0)


#Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
[ ]  def models(X_train,Y_train):


       #Using DecisionTreeClassifier
       from sklearn.tree import DecisionTreeClassifier
       tree = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
       tree.fit(X_train, Y_train)

       #print model accuracy on the training data.

       print('Decision Tree Classifier Training Accuracy:', tree.score(X_train, Y_train))


       return tree
```

```
[ ]  model = models(X_train,Y_train)
```

```
Decision Tree Classifier Training Accuracy: 1.0
```

we can see that the Decision Tree Classifier has the best accuracy, ie.100%

```
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

print('Decision Tree Classifier')
  #Check precision, recall, f1-score
print( classification_report(Y_test, model.predict(X_test)) )
  #Another way to get the models accuracy on the test data
print( accuracy_score(Y_test, model.predict(X_test)))
print()#Print a new line
```

```
Decision Tree Classifier
              precision    recall  f1-score   support

           B       0.99      0.93      0.96        90
           M       0.90      0.98      0.94        53

    accuracy                           0.95       143
   macro avg       0.94      0.96      0.95       143
weighted avg       0.95      0.95      0.95       143

0.951048951048951
```

4.2.3 Prediction model

So here we printed the predictions. The first data shows the actual result of which patient had cancer and the second data is the one predicted by the model. The accuracy of the model is 96.5% so we can see a few wrong predictions but mostly this model is successful in predicting a tumor Malignant (M) (harmful) or Benign (B) (not harmful) based upon the features provided in the data and the training given.

Sample code:

```
#Print Prediction of Decision tree Classifier model
pred = model.predict(X_test)
print(pred)

#Print a space
print()

#Print the actual values
print(Y_test)
```

```
['M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'M' 'M' 'B' 'M'
 'M' 'M' 'M' 'M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B' 'M' 'B' 'M' 'B' 'M'
 'B' 'M' 'B' 'M' 'M' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'M' 'M' 'M' 'B'
 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'B' 'M' 'B' 'M' 'M' 'M' 'B' 'B' 'M'
 'B' 'M' 'M' 'B' 'B' 'M' 'B' 'B' 'M' 'M' 'M' 'B' 'M' 'B' 'B' 'B' 'M' 'M'
 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B'
 'M' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B'
 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'M']

['M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'M'
 'M' 'M' 'M' 'M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B' 'M' 'B' 'M' 'B' 'M'
 'B' 'M' 'B' 'M' 'M' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'M' 'M' 'M' 'B'
 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'B' 'M' 'B' 'M' 'M' 'M' 'B' 'B' 'M'
 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'M' 'B' 'B' 'B' 'M' 'M'
 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B'
 'M' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B'
 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'M']
```

5. GRAPHICAL USER INTERFACE

5.1 Input Design

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.

This system uses google colab as its interface and its is perfect for data science concepts and machine learning algorithms.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent parts after completing all the entries in the current part.

5.2 OUTPUT DESIGN

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients.

The output design of the project is simple and easy to understand. The google colab takes the code input and give the required output. The output includes the data set representation followed by data set refining, it also includes the graphical representation of data and prediction of the data using ML algorithms respectively.

6. TESTING

6.1 System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 Testing Methodologies:

The following are the Testing Methodologies:

• Unit Testing.

• Integration Testing.

• User Acceptance Testing.

• Output Testing.

• Validation Testing.

6.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

6.2.2 Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds

a program structure that has been dictated by design. The following are the types of Integration Testing:

1. Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

• The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.

• A driver (i.e.) the control program for testing is written to coordinate test case input and output.

• The cluster is tested.

6.2.3 User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

6.3 TEST CASES

6.3.1 Test case for data set consistency

Test id	Step number	case	Expected result	Actual result	Pass/fail
TC_01	Step 1	Uploading data set to google colab			
	Step 2	View the data in the data set	Data set info is viewed and displayed	Data set info is viewed and displayed	pass
	Step 3	Analyze the data set	Data set can be edited and modified	Data set can be edited and modified	pass

Table 6.3.1 test case for data consistency

6.3.2 Test case for data prediction

Test id	Step number	case	Expected result	Actual result	Pass/fail
TC_02	Step 1	Using the required algorithm for prediction			
	Step 2	Implementing the algorithm	The required prediction is given is output	The required prediction is given is output	pass
	Step 3	Check for errors	No errors identified	No errors identified	pass

Table 6.3.2 test case for data prediction

7.TECHNOLOGIES USED

Python:

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle-detecting garbage collection system (in addition to reference counting). Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020.

Python consistently ranks as one of the most popular programming languages.

Python was conceived in the late 1980s] by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.] Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member "Steering Council" to lead the project.

Python 2.0 was released on 16 October 2000, with many major new features, including a cycle-detecting garbage collector (in addition to reference counting) for memory management and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible.] Many of its major features were backported to Python 2.6.x] and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported.

Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues leading to possible remote code execution] and web cache poisoning.

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals,[115] manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites. However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of September 2021, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 329,000 packages with a wide range of functionality, including:

- Automation
- Data analytics
- Databases
- Documentation
- Graphical user interfaces
- Image processing
- Machine learning

- Mobile apps
- Multimedia
- Computer networking
- Scientific computing
- System administration
- Test frameworks
- Text processing
- Web frameworks
- Web scraping

Python can serve as a scripting language for web applications, e.g., via mod_wsgi for the Apache web server.[185] With Web Server Gateway Interface,Web frameworks like Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications. SQLAlchemy can be used as a data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.

Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing, with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is a computer algebra system with a notebook interface programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus. OpenCV has Python bindings with a rich set of features for computer vision and image processing.

Python is commonly used in artificial intelligence projects and machine learning projects with the help of libraries like TensorFlow, Keras, Pytorch and Scikit-learn.[190][191][192][193] As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.

Python can also be used to create games, with libraries such as Pygame, which can make 2D games.

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro,[196] and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go.

Many operating systems include Python as a standard component. It ships with most Linux distributions,] AmigaOS 4 (using Python 2.7), FreeBSD (as a package), NetBSD, OpenBSD (as a package) and macOS and can be used from the command line (terminal). Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora Linux use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.

Python is used extensively in the information security industry, including in exploit development.

Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. The Raspberry Pi single-board computer project has adopted Python as its main user-programming language.

Google colab:

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

Google Colab was developed by Google to provide free access to GPU’s and TPU’s to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook.

Google Colab provides tons of exciting features that any modern IDE offers, and much more. Some of the most exciting features are listed below.

- Interactive tutorials to learn machine learning and neural networks.
- Write and execute Python 3 code without having a local setup.
- Execute terminal commands from the Notebook.
- Import datasets from external sources such as Kaggle.
- Save your Notebooks to Google Drive.
- Import Notebooks from Google Drive.
- Free cloud service, GPUs and TPUs.
- Integrate with PyTorch, Tensor Flow, Open CV.
- Import or publish directly from/to GitHub.

Why Should you Use Google Colab?

1, Pre-installed libraries:

Google Colab comes pre-installed with the most popular machine learning libraries. Colab comes pre-installed with Keras, PyTorch, TensorFlow, which saves you the time and hassle of setting up a local environment.

2.Saved on the cloud :

Every Notebook you create in the Google Google Colab is saved on the cloud. This lets you access and work with those Notebooks from any machine. All you need is a browser and a reliable network connection, and you can work from anywhere and anytime.

3. Collaboration :

Collaboration is another amazing reason to choose Google Google Colab when you are working on a project with a team of developers. You can share your Notebook with your teammates and assign them roles so that they can only perform operations that fit their roles. The various options available for each role is shown below:

- Editors can change permissions and share – Viewers and commenters can see the option to download, print, and copy

4. Free GPU and TPU use :

Google Colab provides free access to GPUs and TPUs developed by Google Research. So you can work on your personal projects with powerful GPUs irrespective of your local machine.

Machine learning:

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. Some implementations of machine learning use data and neural networks in a way that mimics the working of a biological brain. In its application across business problems, machine learning is also referred to as predictive analytics.

Learning algorithms work on the basis that strategies, algorithms, and inferences that worked well in the past are likely to continue working well in the future. These inferences can be obvious, such as "since the sun rose every morning for the last 10,000 days, it will probably rise tomorrow morning as well". They can be nuanced, such as "X% of families have geographically separate species with color variants, so there is a Y% chance that undiscovered black swans exist".

Machine learning programs can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the

machine develop its own algorithm, rather than having human programmers specify every needed step.

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset of handwritten digits has often been used.

Decision trees:

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

In decision analysis, a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values (or expected utility) of competing alternatives are calculated.

A decision tree consists of three types of nodes:

Decision nodes – typically represented by squares

Chance nodes – typically represented by circles

End nodes – typically represented by triangles

Decision trees are commonly used in operations research and operations management. If, in practice, decisions have to be taken online with no recall under incomplete knowledge, a decision tree should be paralleled by a probability model as a best choice model or online

selection model algorithm.[citation needed] Another use of decision trees is as a descriptive means for calculating conditional probabilities.

Decision trees, influence diagrams, utility functions, and other decision analysis tools and methods are taught to undergraduate students in schools of business, health economics, and public health, and are examples of operations research or management science methods.

Data science:

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from noisy, structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains. Data science is related to data mining, machine learning and big data.

Data science is a "concept to unify statistics, data analysis, informatics, and their related methods" in order to "understand and analyze actual phenomena" with data. It uses techniques and theories drawn from many fields within the context of mathematics, statistics, computer science, information science, and domain knowledge. However, data science is different from computer science and information science. Turing Award winner Jim Gray imagined data science as a "fourth paradigm" of science (empirical, theoretical, computational, and now data-driven) and asserted that "everything about science is changing because of the impact of information technology" and the data deluge.

A data scientist is someone who creates programming code, and combines it with statistical knowledge to create insights from data.

Data science is an interdisciplinary field focused on extracting knowledge from data sets, which are typically large (see big data), and applying the knowledge and actionable insights from data to solve problems in a wide range of application domains. The field encompasses preparing data for analysis, formulating data science problems, analyzing data, developing data-driven solutions, and presenting findings to inform high-level decisions in a broad range of application domains. also links data science to human-computer interaction: users should be able to intuitively control and explore data. In 2015, the American Statistical Association identified database management, statistics and machine learning, and distributed and parallel systems as the three emerging foundational professional communities.

There is a variety of different technologies and techniques that are used for data science which depend on the application.

1. Linear regression
2. Logistic regression
3. Decision trees are used as prediction models for classification and data fitting. The decision tree structure can be used to generate rules able to classify or predict target/class/label variable based on the observation attributes.
4. Support-vector machine (SVM)
5. Cluster analysis is a technique used to group data together.
6. Dimensionality reduction is used to reduce the complexity of data computation so that it can be performed more quickly.
7. Machine learning is a technique used to perform tasks by inferencing patterns from data
8. Naive Bayes classifiers are used to classify by applying the Bayes' theorem. They are mainly used in datasets with large amounts of data, and can aptly generate accurate results.

CONCLUSION AND FUTURE SCOPE

In this project in python, we learned to build a breast cancer tumour predictor on the wisconsin dataset and created graphs and results for the same. It has been observed that a good dataset provides better accuracy. Selection of appropriate algorithms with good home dataset will lead to the development of prediction systems. These systems can assist in proper treatment methods for a patient diagnosed with breast cancer. There are many treatments for a patient based on breast cancer stage; data mining and machine learning can be a very good help in deciding the line of treatment to be followed by extracting knowledge from such suitable databases.

REFERENCES

1. https://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/

2.https://towardsdatascience.com/buildinga-simple-machine-learning-model-onbreast-cancer-data-eca4b3b99fa3

3. Original data Set:

http://archive.ics.uci.edu/ml/datasets/breas t+cancer+wisconsin+%28diagnostic% 29

4.Confusion Matrix:

https://tatwan.github.io/How-To-Plot-AConfusion-Matrix- In-Python/

5.https://seaborn.pydata.org/tutorial/axis_grids.html

6.https://seaborn.pydata.org/generated/seaborn.pairplot.html

7. https://seaborn.pydata.org/generated/seabo rn.heatmap.html