A Major-Project Report
on

BREAST CANCER PROGNOSIS USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements

for the award of degree of

BACHELOR OF TECHNOLOGY

in

Information Technology

by

R.Parimala (19WH1A1268)
P.Aditi Kiran (19WH1A1277)
Ch.Lakshmi Durga (19WH1A1295)
Y.Raveena (19WH1A1296)

Under the esteemed guidance of
Mr.B. Srinivasulu
Assistant Professor

VISHNU
UNIVERSAL LEARNING

Department of Information Technology

BVRIT HYDERABAD College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

(NAAC ‘A’ Grade & NBA Accredited- ECE, EEE, CSE, IT)

June, 2023

DECLARATION

We hereby declare that the work presented in this project entitled “Breast Cancer Prognosis using Machine Learning” submitted towards completion of the Major Project of the Project in IV year II sem of B.Tech IT at “BVRIT HYDERABAD College of Engineering for Women”, Hyderabad is an authentic record of our original work carried out under the esteemed guidance of Mr.B. Srinivasulu, Assistant Professor, Department of Information Technology.

R.Parimala (19WH1A1268)

P.Aditi Kiran (19WH1A1277)

Ch.Lakshmi Durga (19WH1A1295)

Y.Raveena (19WH1A1296)

VISHNU
UNIVERSAL LEARNING

BVRIT HYDERABAD

College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)
(NAAC ‘A’ Grade & NBA Accredited- ECE, EEE, CSE IT)

CERTIFICATE

This is to certify that the major-project report on **“Breast Cancer Prognosis using Machine Learning ”** is a bonafide work carried out by **R.Parimala (19WH1A1268), P.Aditi Kiran (19WH1A1277), Ch.Lakshmi Durga (19WH1A1295)** and **Y.Raveena (19WH1A1296)** in the partial fulfillment for the award of B.Tech degree in **Information Technology, BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad** affiliated to Jawaharlal Nehru Technological University, Hyderabad under my guidance and supervision.

The results embodied in the major project work have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide	**Head of the Department**
Mr. B. Srinivasulu	**Dr. Aruna Rao S L**
Assistant Professor	**Professor & HoD**
Department of IT	**Department of IT**

External Examiner

ACKNOWLEDGEMENT

We would like to express our profound gratitude and thanks to **Dr. K. V. N. Sunitha, Principal, BVRIT HYDERABAD** for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. Aruna Rao S L, Professor & Head, Department of IT, BVRIT HYDERABAD** for all the timely support, constant guidance and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Mr. B. Srinivasulu, Assistant Professor, Department of IT, BVRIT HYDERABAD** for his constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinators **Dr. P. Kayal, Associate Professor** and **Ms. K. S. Niraja, Assistant Professor,** , all the faculty and staff of Department of IT who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

Ramisetty Parimala (19WH1A1268)

Porika Aditi Kiran (19WH1A1277)

Chitikina Lakshmi Durga (19WH1A1295)

Yadlapalli Raveena (19WH1A1296)

ABSTRACT

Today, Breast Cancer has become the most frequent type among all other cancers. The major cause of death in women worldwide is identified as breast cancer. Each year the number of deaths is increasing rapidly due to breast cancer. Early prediction and diagnosis is very important for a healthy life. Women are getting worse effected due to breast cancer. The lack of models to predict the disease in its early stages made it difficult for doctors to prepare a medication plan for the patient which may prolong the patient's survival time. Machine learning techniques can be helpful in the process of early prediction and diagnosis of breast cancer. Hence, there is a requirement to develop the technique which gives minimum error to increase accuracy. In this project, we applied five machine learning algorithms: Support Vector Machine (SVM), Random Forest, Logistic Regression, Decision tree, and K-Nearest Neighbors (KNN) on the Breast Cancer Wisconsin Diagnostic data set. After acquiring the results, a performance assessment and comparison is carried out between these 5 different classifiers. The main objective of our project is to predict and perform diagnosis on breast cancer. Using machine-learning algorithms, we find out the most effective results with respect to confusion matrix, accuracy, and precision. The proposed work can be used to predict the outcome of different techniques and suitable techniques can be used depending upon the requirement. This research is carried out to predict the accuracy and its mostly based on input data set. It is observed that the Support Vector Machine (SVM) performed well than all other classifiers and achieved the highest accuracy. But with respect some type of input sets other algorithms may give best results. Hence, we have used different algorithms to choose the best result after prediction and analysis. We are also predicting the re occurrence of the tumors/breast cancer based on accuracy given by different algorithms like the Naive bays algorithm predicts the probability of re occurrence using the prior data set probability. All the work is done in the google colab environment based on python programming language and Scikit-learn library.

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1.1	Causes and risk factors of breast cancer	02
1.2	Machine learning is a subset of artificial intelligence	03
1.3	Count of diagnosis	06
1.5	Architecture	10
1.6	Benign and malignant Tumors	12
4.2	Feature Extraction	16
4.3.1	Visualization of Data	19
4.3.5	Bayes Theorem	23
4.4	Experimenting Environment	24
5.1	Scatter plot between attributes	40
5.2	Correlation matrix between attributes	41
5.3	Correlation matrix for 12 Attributes	42
5.4	Correlation Graph	43
5.5	Confusion matrix for Logistic Regression	44
5.6	Confusion matrix for K Nearest Neighbor	44
5.7	Confusion matrix for SVM	45
5.8	Confusion matrix for Naïve Bayes	45
5.9	Confusion Matrix for Decision tree	46
5.10	Confusion matrix for random forest	46
5.11	Score and Acuracy of Algorithms	47
5.12	Order of Score and Acuracy of Algorithms	48
5.13	Prediction of Breast Cancer	49
5.14	Prediction of Reoccurance	50

LIST OF ACRONYMS

ACRONYMS	ABBREVIATION
ML	Machine Learning
KNN	K – Nearest Neighbor
SVM	Support Vector Machine
DT	Decision Tree
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
OOP	Object Oriented Program
GUI	Graphic User Interface
BCRAT	Breast Cancer Risk Assessment tool
BOADICEA	Breast and Ovarian Analysis of Disease Incidence and Carrier Estimation Algorithm
MRI	Magnetic Resonance Imaging
ROC	Receiver Operation Characteristics

CONTENTS

TOPIC PAGE NO.

Abstract V

List of Figures VI

List of Acronyms VII

1. Introduction 1

1.1 Objective 1

1.2 Machine Learning 2

1.3 Data Preprocessing 4

1.4 Python 7

1.5 Existing Work 9

1.6 Present Work 11

2. Literature Survey 13

3. Hardware/Software Tools 15

3.1 Hardware Requirements 15

3.2 Software Requirements 15

4. Project Implementation 16

4.1 Problem Statement 16

4.2 Methodology 16

4.3 Dataset 17

4.3.1 Importing Dataset 18

4.3.2 Data Preprocessing 19

4.3.3 Training and Testing 19

4.3.4 Prediction 20

4.3.5 Algorithms 21

4.4 Experimental Environment 24

4.5 Code 25

5. Results & Analysis 40

6. Conclusion and Future Scope 51

References 52

1. Introduction

1.1 Objective

In the developing world, cancer death is one of the major problems for humankind. Even though there are many ways to prevent it before happening, some cancer types still do not have any treatment. One of the most common cancer types is breast cancer, and early diagnosis is the most important thing in its treatment. Accurate diagnosis is one of the most important processes in breast cancer treatment. Women are seriously threatened by breast cancer with high morbidity and mortality. The lack of robust prognosis models results in difficulty for doctors to prepare a treatment plan that may prolong patient survival time. Hence, the requirement of time is to develop the technique which gives minimum error to increase accuracy. In this, we have used different algorithms to compare the accuracy of different algorithm of Machine Learning about cancer depending on the given data sets. The algorithms used are Logistic Regression, Decision Tree, Naïve Bayes, K Nearest Neighbors (KNN),Support Vector Machine (SVM) to evaluate and compare the classification of accuracy.

According to the statistics discharged by the International Agency for analysis on Cancer in Gregorian calendar month 2020 carcinoma has overtaken carcinoma because the most diagnosed cancer in girls worldwide. within the past twenty years, the amount of individuals diagnosed for cancer has increased double i.e., from associate degree estimate of ten million in year 2000 to 19.3 million within the year 2020. Prognosis counsel that the amount of individuals being diagnosed with cancer can increase additional within the coming back years and it'll be nearly 50 % higher in year 2040 than within the year 2020. This fortifies the need to take a position in each the fight against the cancer and cancer hindrance. The prosperous introduction of knowledge and communication technologies in medical field is a crucial stake within the melioration of the aid system and additional squarely in cancer care. data processing algorithms enforced within the aid business play an interesting role because of their high performance in prediction and designation of diseases, that successively results in reducing prices of drugs, and taking period selections to save lots of peoples lives. the foremost common data processing modelling objectives are classification and prediction that uses many algorithms for the prediction of carcinoma. This project primarily provides a comparison between the performance of 5 classifiers. Support Vector Machine (SVM), Random Forest, Logistic Regression, decision tree, and K-Nearest Neighbor (KNN). per the analysis community are these algorithms are among the foremost influential data processing algorithms and additionally among the highest ten data processing algorithms. we are also predicting the reoccurrence of breast cancer using SVM. In the dataset the best predictors of breast cancer are tumor size, tumor margin, number of lymph nodes, number of tumors, lympho-vascular invasion. We applied SVM algorithm on the data set and based on the best accuracy we found the reoccurrence of tumors/best cancer.

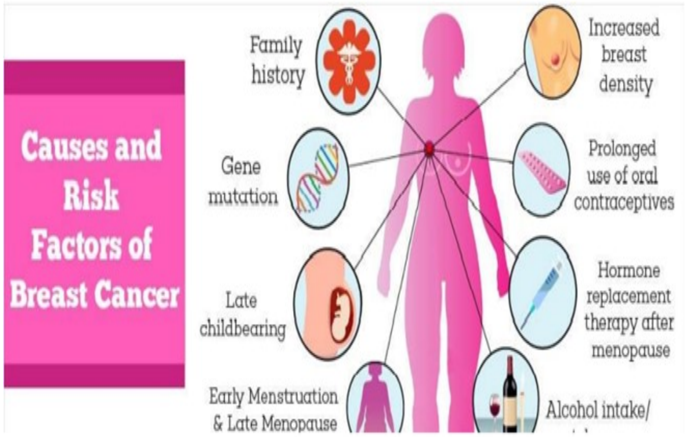

Causes and Risk Factors of Breast Cancer
Family history
Gene mutation
Late childbearing
Early Menstruation & Late Menopause
Increased breast density
Prolonged use of oral contraceptives
Hormone replacement therapy after menopause
Alcohol intake/

Figure 1.1: Causes and risk factors of breast cancer

1.2 Machine Learing

Machine learning (ML) is the study of computer algorithms that 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, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed task.

In addition to an informed, working definition of machine learning(ML), we detail the challenges and limitations of getting machines to 'think,' some of the issues being tackled today in deep learning (the frontier of machine learning), and key takeaways for developing machine learning applications for business use-cases. Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. All of these things mean it's possible to quickly and automatically produce models that can analyze bigger, more complex data and deliver faster, more accurate results – even on a very large scale. And by building precise models, an organization has a better chance of identifying profitable opportunities – or avoiding unknown risks. Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.

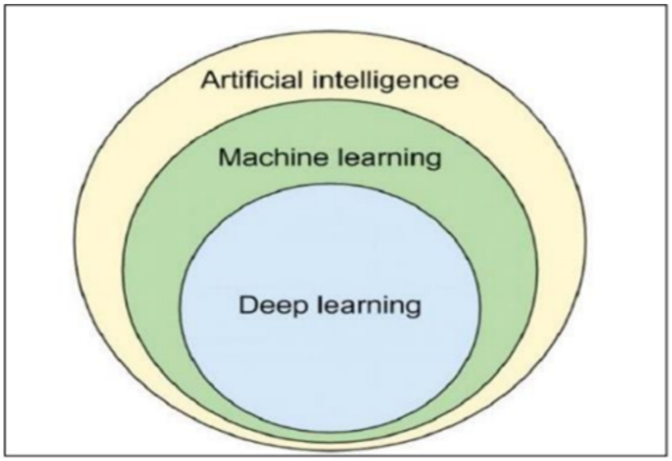


Figure 1.2: Machine learning is a subset of artificial intelligence

Supervised learning: In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.

Unsupervised learning: This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined. Examples of Unsupervised Learning: Apriori algorithm, K-means.

Semi-supervised learning: This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

Reinforcement learning: Data scientists typically use reinforcement learning to teach a machine to complete a multistep process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way. Example of Reinforcement Learning: Markov Decision Process.

1.3 Data Preprocessing

There are seven significant steps in data pre-processing in Machine Learning:

1.Acquire the dataset:

To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprised of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases.

2.Import all crucial libraries:

Python is the most extensively used and also the most preferred library by Data Scientists around the world. The predefined Python libraries can perform specific data preprocessing jobs. The three core Python libraries used for this data preprocessing in Machine Learning are:

i) NumPy:

NumPy is the fundamental package for scientific calculation in Python. Hence, it is used for inserting any type of mathematical operation in the code. Using NumPy, you can also add large multidimensional arrays and matrices in your code.

ii) Pandas:

Pandas is an excellent open-source Python library for data manipulation and analysis. It is extensively used for importing and managing the datasets. It packs in high performance, easy-to- use data structures and data analysis tools for Python.

iii) Matplotlib:

Matplotlib is a Python 2D plotting library that is used to plot any type of charts in Python. It can deliver publication-quality figures in numerous hard copy formats and interactive environments across platforms (IPython shells, Jupyter notebook, web application servers, e.t.c).

3.Import the dataset:

In this step, import the dataset's that you have gathered for the ML project at hand. However, before importing the dataset's, you must set the current directory as the working directory. Once you've set the working directory containing the relevant dataset, you can import the dataset using the "read-csv()" function of the Pandas library. This function can read a CSV file (either locally or through a URL) and also perform various operations on it. The read-csv() is written as: data-set= pd.read-csv('Dataset.csv')

4.Identifying and handling the missing values:

In data preprocessing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data. Needless to say, this will hamper your ML project. Basically, there are two ways to handle missing data:

i) Deleting a particular row – In this method, you remove a specific row that has a null value for a feature or a particular column where more than 75 % of the values are missing. However, this method is not 100 % efficient, and it is recommended that you use it only when the dataset has adequate samples. You must ensure that after deleting the data, there remains no addition of bias.

ii) Calculating the mean – This method is useful for features having numeric data like age, salary, year, etc. Here, you can calculate the mean, median, or mode of a particular feature or column or row that contains a missing value and replace the result for the missing value. This method can add variance to the dataset, and any loss of data can be efficiently negated. Hence, it yields better results compared to the first method (omission of rows/columns). Another way of approximation is through the deviation of neighboring values. However, this works best for linear data.

5.splitting the data set:

Every dataset for Machine Learning model must be split into two separate sets –training set and test set. Training set denotes the subset of a dataset that is used for training the machine learning model. Here, you are already aware of the output output. A test set, on the other hand, is the subset of the dataset that is used for testing the machine learning model. The ML model uses the test set to predict outcomes.

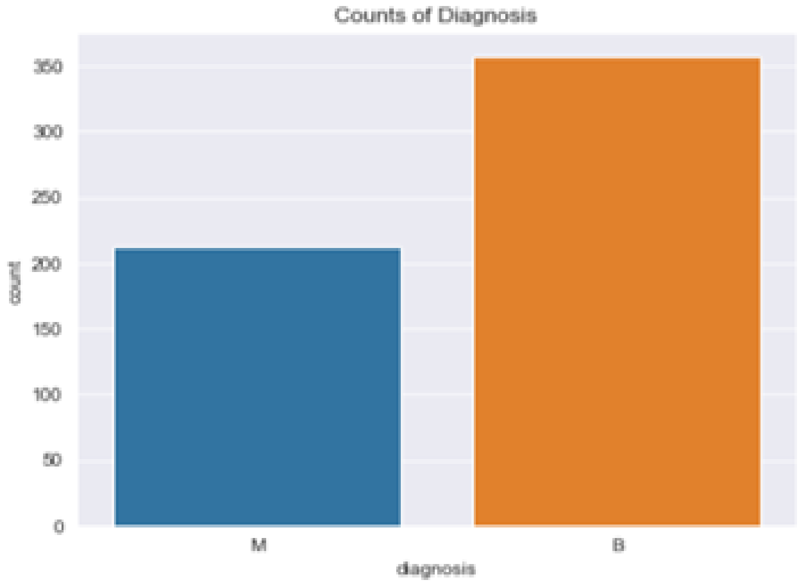


Figure 1.3: Count of diagnosis

1.4 Python

Python is a popular object-oriented programming language having the capabilities of high- level programming language. Its easy to learn syntax and portability capability makes it popular these days. The following facts given us the introduction to Python Python was developed by Guido van Rossum at Stitching Mathematics' Centrum in the Netherlands. It was written as the successor of programming language named 'ABC'. Its first version is released in 1991. The name Python was picked by Guido van Rossum from a TV show named Monty Python's Flying Circus. It is an open-source programming language which means that we can freely download it and use it to develop programs. It can be downloaded from "www.python.org.." Python programming language is having the features of Java and C both. It is having the elegant 'C' code and on the other hand, it is having classes and objects like java for object oriented programming. It is an interpreted language, which means the source code of Python program would be first converted into bytecode and then executed by Python virtual machine.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other language. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python.

Python is Interpreted Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

i) Python is Interactive You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

ii) Python is Object-Oriented Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

iii) Python is a Beginners Language Python is a great language for the beginner level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Characteristics of Python:

Following are important characteristics of Python Programming

i) It supports functional and structured programming methods as well as OOP.

ii) It can be used as a scripting language or can be compiled to byte-code for building large applications.

iii) It provides very high-level dynamic data types, supports dynamic type checking.

iv) It supports automatic garbage collection.

v) It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Applications of Python:

As mentioned before, Python is one of the most widely used language over the web. List few of them here:

i) Easy-to-learn Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

ii) Easy-to-read Python code is more clearly defined and visible to the eyes.

iii) Easy-to-maintain - Python's source code is fairly easy-to-maintain.

iv) A broad standard library Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh.

v) Interactive Mode Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

vi) Portable Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

vii) GUI Programming Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix .

viii) Scalable Python provides a better structure and support for large programs than shell scripting.

1.5 Existing Work

Comprehensive breast cancer risk prediction models enable identifying and targeting women at high-risk, while reducing interventions in those at low risk. Breast cancer risk prediction models used in clinical practice have low discriminatory accuracy (0.53–0.64).Since 2009, the U.S. Preventive Services Task Force recommends breast cancer screening with biannual mammograms for women age 50 to 74years old. In 2013, Switzerland also adopted a national strategy, recommending biannual breast cancer screening for women over 50. Age over 50 years is the sole risk factor considered for entering a population screening program. However, about 25 % of breast cancer patients are diagnosed in women under 50years old. Mammograms are less effective as a breast cancer screening tool for younger women, who are more likely to have dense breast tissue, compromising the utility of routine mammograms in this age group. This contributes to diagnostic delays and increased morbidity and mortality. Risk-based screening could be more effective, less morbid, and more cost-effective. Comprehensive breast cancer risk prediction models, able to classify women into clinically meaningful risk groups, will enable identifying and targeting women at high-risk, while reducing interventions in those at low risk.

The Breast Cancer Risk Assessment Tool (BCRAT), also known as the Gail model, and the Breast and Ovarian Analysis of Disease Incidence and Carrier Estimation Algorithm (BOADICEA) model were developed to identify high-risk women based on known risk factors and have been integrated into clinical guidelines to help guide decision making about breast cancer risk management. BCRAT was developed and validated with data from the US Surveillance, Epidemiology, and End Results registry. The model uses eight risk factors, i.e., age, age of menarche, age of first live birth, number of previous biopsies, benign disease, BRCA mutations, race, and number of first-degree relatives affected with breast cancer, to calculate 5-year and lifetime risk for women older than 35years old. The National Comprehensive Cancer Network suggests using BCRAT to identify women with a 5-year risk greater than 1.66 % and women with remaining lifetime risk greater than 20 %, who could consider risk-reducing chemo preven-

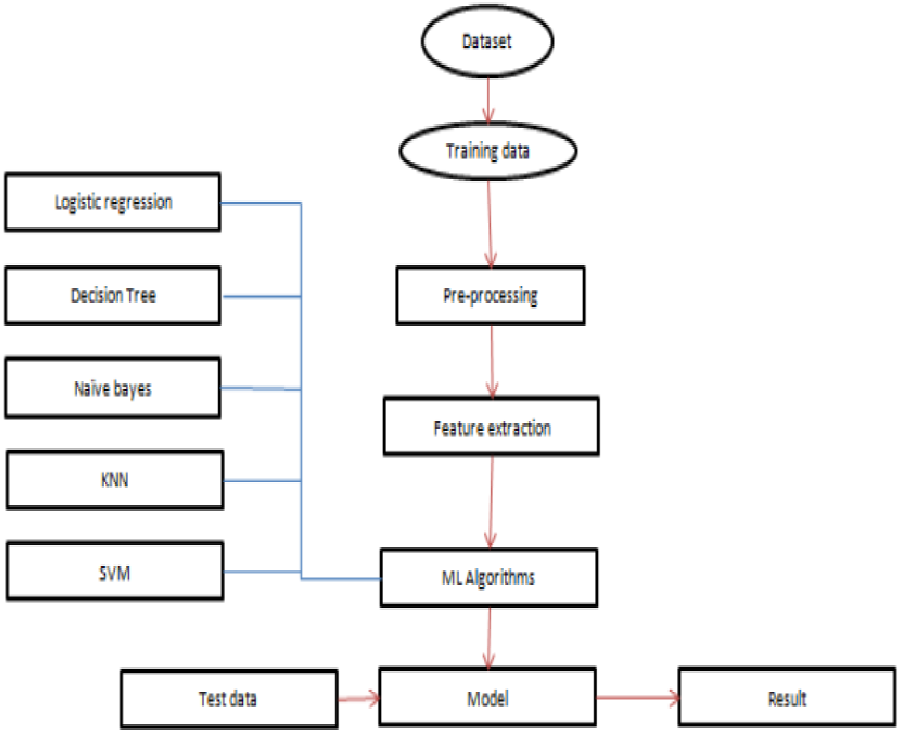


Figure 1.5: Architecture

tion and annual screening with mammograms and MRI's (magnetic resonance imaging) starting at 30years old. The BOADICEA model was the first polygenic breast cancer risk prediction model, based on data from 2785 UK families. BOADICEA uses information from personal and family history of breast cancer, including information from breast cancer pathology, ethnicity, and BRCA mutations. Clinical guidelines in several European countries and Switzerland recommend using BOADICEA for breast cancer risk prediction.

However, both models have considerable limitations. BCRAT can only be used for women above 35years old, and only takes into account history of breast cancer in first-degree relatives (mother, sisters, or daughters), without including age at diagnosis of these relatives. It does not consider family history of ovarian cancer, which may be of crucial importance for women with hereditary breast and ovarian cancer (HBOC). The BOADICEA model does not account for risk factors associated with reproductive history and hormonal exposure and has limited utility in cases with small family history. Although both models have been validated with large cohort data, their discriminatory ability, area under the ROC (receiver operating characteristics) curve, is between 0.53 and 0.64. There is 36 to 47% chance that the BCRAT and BOADICEA model will not identify high-risk women, while some low-risk women may receive unnecessary preventive treatments. Both models make implicit assumptions that risk factors relate to cancer development in a linear way and are mostly independent of other risk factors. Thus, both models likely oversimplify complex relationships and non-linear interactions in numerous risk factors. Machine learning (ML) offers an alternative approach to standard prediction modeling that may address current limitations and improve the accuracy of those tools.

1.6 Present Work

Data mining algorithms enforced in the healthcare industry play a remarkable role due to their high performance in prediction and diagnosis of diseases, which in turn leads to reducing costs of medicine, and taking real-time decisions to save peoples lives. The most common Data mining modelling objectives are classification and prediction which uses several algorithms for the prediction of breast cancer. This project mainly gives a comparison between the performance of five classifiers. Support Vector Machine (SVM), Random Forest, Logistic Regression, Decision tree, and K-Nearest Neighbors (KNN).

Our objective is to predict and diagnose breast cancer, using machine learning algorithms, and find out the most effective result based on the performance of each classifier/algorithm in terms of confusion matrix, precision, accuracy, and sensitivity. And we are also predicting the reoccurrence of breast cancer by applying SVM algorithm. Basically, there are two types of breast cancer.

They are malignant and benign.

i) Malignant - Malignant tumors have cells that grow unlimitedly and spread in to distant sites. Malignant tumors are cancerous i.e., they occupy other sites. This type of tumor will generally have the potential to be dangerous.

ii)Benign - Benign tumors are those that stay in their primary location without infecting other sites of the body. They do not spread to the local structures or to the distant parts of the body. This type of tumor will generally not be dangerous.

The main objective of this project is to build a model for predicting cancer and its reoccurrence using various machine learning techniques.

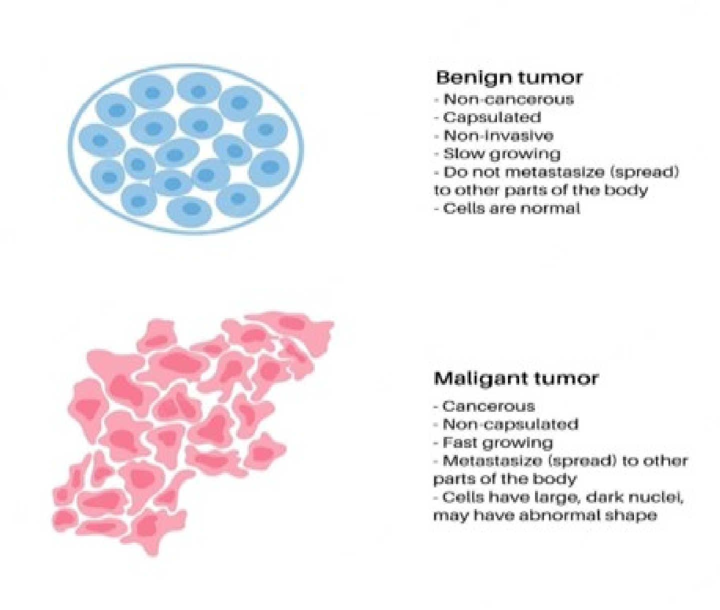

Benign tumor
- Non-cancerous
- Capsulated
- Non-invasive
- Slow growing
- Do not metastasize (spread) to other parts of the body
- Cells are normal
Maligant tumor
- Cancerous
- Non-capsulated
- Fast growing
- Metastasize (spread) to other parts of the body
- Cells have large, dark nuclei, may have abnormal shape

Figure 1.6: Benign and malignant Tumors

2. Literature Survey

A lot of researchers have realized research in breast cancer by using several datasets such as using SEER dataset, Mammogram images as a dataset, Wisconsin Dataset, and datasets from various hospitals. By exploiting these datasets authors extract and select variously features and complete their research. These are some significant researches.

1) The author Sudarshan Nayak demonstrates the use of various supervised machine learning algorithms in the classification of breast cancer by using 3D images and find out that SVM is the best based on his overall performance.

2) The authors Wang, D. Zhang and Y. H. Huang worked on breast cancer prediction using the Logistic Regression algorithm and found it was working well with an accuracy of 91% but it was a bit slow process when compared to other algorithms.

3) The author B. Akbugday proposed breast cancer prediction using kNN algorithm. He observed that it required high memory and was expensive too. The accuracy obtained was about 89%. 4) On the other side, we find that B.M.Gayathri, work on a comparative study of Relevance vector machine which provides Low computational cost while comparing with other machine learning techniques which are used for breast cancer detection and explaining how SVM is better than other machine learning algorithms for diagnosing breast cancer even the variables are reduced and achieved good accuracy. 5) The authors Keles and M. Kaya worked on the same problem statement using the algorithm Random Forest and achieved an accuracy of about 92%. But the demerit was that this algorithm required more time for training when compared with other algorithms. 6) V. Chaurasia and S. Pal both worked on breast cancer prediction using Naive Bayes algorithm and achieved an accuracy of nearly 87%. The only demerit of this algorithm is a lousy estimator so we should not take the probability output too seriously.

7) The authors R. K. Kavitha and D. D. Rangasamy worked on same problem using a different approach. The algorithm used was Neural Networks and achieved an accuracy of about 90% but the problem was that in this algorithm the statistical distribution of input keeps changing as training proceeds. 8) Similarly, the authors P. Sinthia, R. Devi, S. Gayathri and R. Sivasankari

performed the same using the Back Propagation technique which was actually sensitive to the complex data but the accuracy was good with 91%.

In recent works, we find that Youness khoudfi and Mohamed Bahaj, similarly proposed a comparison between Machine learning algorithms and they found the SVM is the best classifier compared with K-NN, RF, and NB, they are based on Multilayer perception with 5 layers and 10 times cross-validation using MLP.

3. Hardware/Software Tools

3.1 Hardware Requirements

i) System

ii) Hard disk

iii) Ram – 4 GB

iv) Processor – intel core i5 7th generation

v) Hard drive – 64 GB

3.2 Software Requirements

i) Operating System - Windows 10

ii) Jupyter Notebook

iii) Google collab

iv) Google Chrome

v) MS – Word

4. Project Implementation

4.1 Problem Statement

To identify breast cancer, I early stage to save people's life. Early prediction and diagnosis are very important to make a proper medication plan to increase one's survival of life. And also to identify the reoccurrence of breast cancer

4.2 Methodology

The main objective of our project is to identify an effective algorithm for the detection of breast cancer. So we applied machine learning classifiers like Support Vector Machine (SVM), Logistic Regression, Random Forests-Nearest Neighbours (KNN), Decision tree on the Kaggle Breast Cancer dataset and evaluate the results acquired to define which model provides higher/best accuracy.

In the implementation phase we have followed these steps:

i) Importing data

ii) Data Pre-processing

iii) Training and testing the data

iv) Prediction

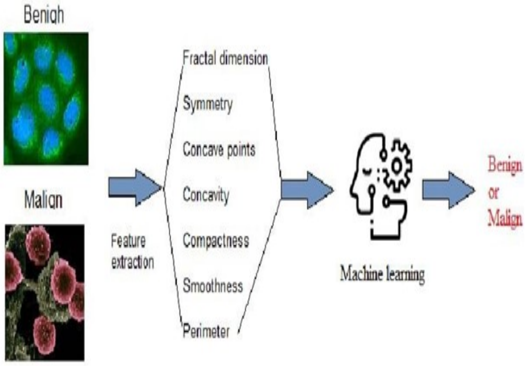


Figure 4.2: Feature extraction

4.3 Dataset

In this project we have used the dataset which is available from online website i.e., from Kaggle website.

The dataset is already pre-processed by using datamining concepts like removing null values etc... and already the dataset is split into training dataset and testing dataset. The training dataset contains 70 % of data and testing dataset contains 30 % of data which is used for training and testing of the data.

There are 32 attributes for the project. The following datatypes for attributes are int64, float 64,float64.

S.No.	Attribute	Data Type
1	Id	Int64
2	Diagnosis	Object
3	Radius-mean	Float64
4	Texture-mean	Float64
5	Perimeter-mean	Float64
6	Area-mean	Float64
7	Smoothness-mean	Float64
8	Compactness-mean	Float64
9	Concavity-mean	Float64
10	Concave points-mean	Float64
11	Symmetry-mean	Float64
12	Fractal-dimension-mean	Float64
13	Radius-se	Float64
14	Texture-se	Float64
15	Perimeter-se	Float64
16	Area-se	Float64

S.No.	Attribute	Data Type
17	Smoothness-se	Float64
18	Compactness-se	Float64
19	Concavity-se	Float64
20	Concave points-se	Float64
21	Symmetry-se	Float64
22	Fractal-dimension-se	Float64
23	Radius-worst	Float64
24	Texture-worst	Float64
25	Perimeter-worst	Float64
26	Area-worst	Float64
27	Smoothness-worst	Float64
28	Compactness-worst	Float64
29	Concavity-worst	Float64
30	Concave points-worst	Float64
31	Symmetry-worst	Float64
32	Fractal-dimension-worst	Float64

4.3.1 Importing Dataset

The features of the dataset are computed from a digitized image of a breast cancer sample obtained from a fine-needle aspirate (FNA). The characteristics of the cell nuclei present in the image are determined by these features. The dataset has 20,000 instances, 2 classes (62.74 % benign and 37.26 % malignant), and 11 integer-valued attributes (-Id -Diagnosis -Radius - Texture -Area - Perimeter -Smoothness -Compactness -Concavity -Concave points -Symmetry -Fractal dimension).

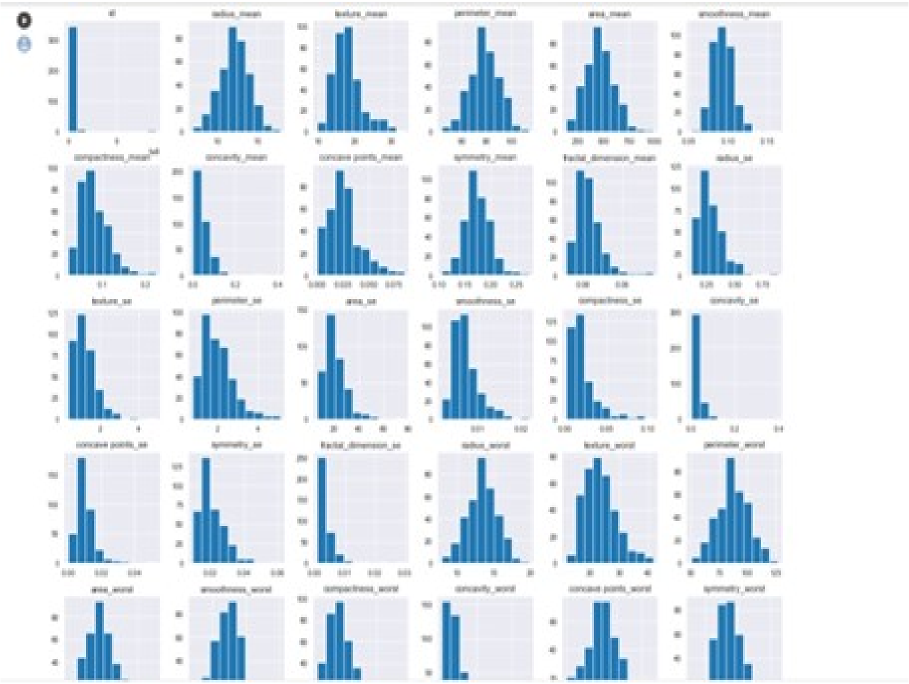


Figure 4.3.1: Visualization of Data

4.3.2 Data Preprocessing

Data preprocessing is used to complement missing values, identify or remove outliers, and solve self-contradiction. The prepared data is used to build machine learning algorithms that can predict breast cancer for a new set of measurements.

4.3.3 Training and Testing

To evaluate the performances of the algorithms, we show the model new data for which we have labels. This is usually done by splitting the labelled data we have collected into two parts with ”Train-test-split”method. 75 % of the data is used to build our machine learning model and is called the training data or training set. 25 % of the data will be used to access how well the model works and is called test data, test set.

4.3.4 Prediction

Prediction of Breast Cancer:

After testing the models, we compare the obtained results to select the algorithm that provides high accuracy and identify the most predictive algorithm for the detection of breast cancer. With the help of best algorithm selected predicting the cancer on given values.

The data collection consists of the parameters regarding the ratio,smoothness, radius,from the lab results. The data is pre-processed and cleaned using Dimensional reduction techniques. The relevant data sets are then used in further steps like classification or regression.The pre-processed data undergoes K-fold cross validation, where the accuracy of various machine learning algorithms on the data is tested. From, the results of the cross validation, the data is trained using SVM algorithm for the analysis of the tumor as cancerous(malignant) or non- cancerous(benign). The data is split into 80is then tested for its accuracy and miss rate in the performance evaluation layer. The tested data is then used to identify the breast cancer and classify it based on the SVC algorithm.

Prediction of Reccurence:

The chance of recurrence of breast cancer and the risk factors associated with it can vary depending on several factors, including the stage of the cancer at the time of diagnosis, the treatment received, and individual characteristics. Here are some general factors that can influence the risk of breast cancer recurrence:

Stage of cancer: The stage of breast cancer at the time of diagnosis plays a significant role in the risk of recurrence. Generally, the higher the stage, the greater the risk of recurrence.

Tumor characteristics: Certain characteristics of the tumor, such as size, grade, and hormone receptor status (estrogen and progesterone receptors, HER2 status), can impact the likelihood of recurrence. Higher-grade tumors and those that are hormone receptor-negative or HER2-positive may have a higher risk of recurrence.

Lymph node involvement: If cancer has spread to the lymph nodes, it increases the risk of recurrence. The number of involved lymph nodes and the extent of involvement can further affect the risk.

4.3.5 Algorithms

LogisticRegression:

Logistic regression was introduced by statistician DR Cox in 1958 and so predates the field of machine learning. It is a supervised machine learning technique, employed in classification jobs (for predictions based on training data). Logistic Regression uses an equation like Linear Regression, but the outcome of logistic regression is a categorical variable whereas it is a value for other regression models. Binary outcomes can be predicted from the independent variables. The general workflow is: First we need to extract the dataset. Then train the classifier with the dataset. Make prediction from the dataset using classifier.

k-Nearest Neighbour (kNN):

K-Nearest Neighbour is a supervised machine learning algorithm as the data given to it is labelled. It is a nonparametric method as the classification of test data point relies upon the nearest training data points rather than considering the dimensions (parameters) of the dataset. Input the dataset and split it into training and testing set. Now pick an instance from the testing sets and calculate its distance with the training set. Make a list of distances obtained in ascending order. The class of the given instance is the most common class of first three training instances(k=3).

Support Vector machine: Support Vector Machine is a supervised machine learning algorithm which is doing well in pattern recognition problems, and it is used as a training algorithm for studying classification and regression rules from data. SVM is most precisely used when the number of features and number of instances are high. A binary classifier is built by the SVM algorithm. In an SVM model, each data item is represented as points in an n-dimensional space where n is the number of features where each feature is represented as the value of a coordinate in the n-dimensional space. In this algorithm, first it finds and boundaries and lines that correctly classify the data set into coordinates. Then from these lines and boundaries the classifier picks the one that has maximum distance from the closest data points.

Random Forest:

Random forests or random decision forests are an ensemble method for classification, regression, and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees & habit of overfitting to their training set. In this algorithm n number of random records are considered from the data set having k number of records. For each sample individual decision trees are constructed and each decision tree will generate a output. The final output is considered based on majority averaging or voting for classification and regression respectively.

Decision Tree:

Decision Tree is a predictive modelling tool that can be applied across many areas. It can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. It is started with the original set S as the root node. On each iteration of the algorithm, it iterates through the very unused data/attribute of given set S and then calculates the Entropy and Information Gain of attribute. Then it selects the attribute which has the largest information gain or smallest entropy. Then the set S is split by the selected attribute to produce a subset of data. The algorithm continues to recur on each subset, considering only attributes that never selected before.

Naïve Bayes:

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. NaïveBayes 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. Baye’s; theorem is also known as

Baye's; Rule or Baye's; law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability. The formula for Baye's; theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 4.3.5: Bayes Theorem.

4.4 Experimental Environments

All experiments on the machine learning algorithms described during this project were conducted using the Python programming language and Scikit-learn library. Scikit-learn library also known as sklearn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, logistic regression, k- means, and Decision tree, and is designed to interoperate with the Python numerical and scientific libraries NumPy and Pandas.

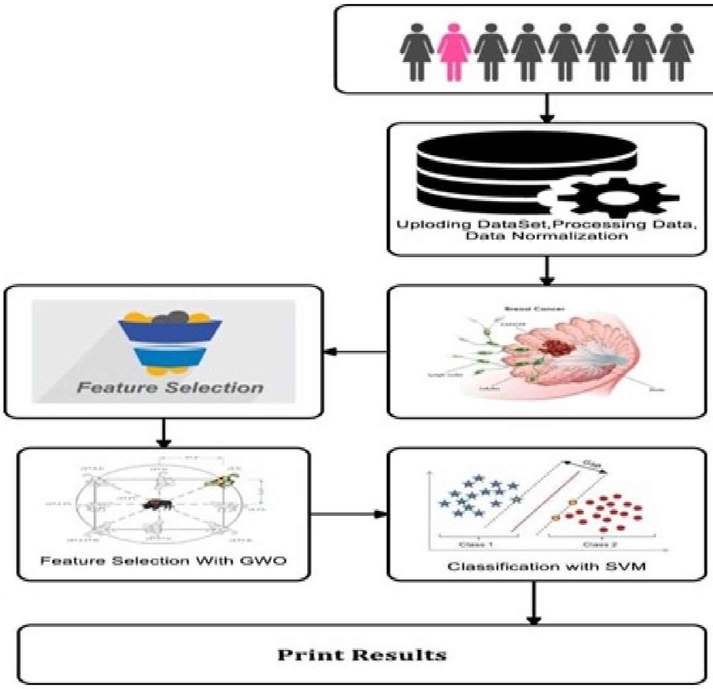


Figure 4.4: Experimenting environment

4.5 Code

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
%matplotlib inline
dataset = pd.read_csv(’data.csv’)
X = dataset.iloc[:, 1:31].values
Y = dataset.iloc[:, 31].values
dataset.head()
print("Cancer data set dimensions : {}".format(dataset.shape))
diagnosis_unique = dataset.diagnosis.unique()
print(diagnosis_unique)
benign,malignant=dataset["diagnosis";].value_counts()
print(’Number of cells labeled Benign: ’ , benign)
print(’Number of cells labeled Malignant : ’, malignant)
print(’% of cells labeled Benign&#39;, round(benign / len(dataset) * 100, 2), ’%’)
print(’% of cells labeled Malignant&#39;, round(malignant / len(dataset) * 100,
2), ’%’)
sns.set_style(’darkgrid’)
plt.figure(figsize=(15, 5))
plt.xlabel("Diagnosis")
plt.subplot(1, 2, 2)
plt.title("Counts of Diagnosis")
sns.countplot('diagnosis', data=dataset);
dataset.isnull()
dataset.isna().sum()
df_pred = pd.DataFrame(df_prediction, columns=df_prediction_cols)
print(len(confusion_matrixs))
df_pred
df_pred.sort_values("score", ascending=False)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,prediction)
print(cm)
model = SVC(kernel = 'rbf', random_state = 2)
model.fit(X_train,y_train)
prediction = model.predict(X_test)
import random
a = random.random()
meanR=round(random.uniform(0,30),3)
meanT=round(random.uniform(0,50),3)
meanS=round(random.uniform(0,1),5)
meanC=round(random.uniform(0,1),5)
meanSy=round(random.uniform(0,1),4)
meanF=round(random.uniform(0,1),5)
seR=round(random.uniform(0,2),4)
seT=round(random.uniform(0,3),4)
seS=round(random.uniform(0,1),6)
seC=round(random.uniform(0,1),6)
seSy=round(random.uniform(0,1),6)
```

```
seF=round(random.uniform(0,1),6)
input_1=[]
lst =  map(lambda x : x[1], filter(lambda x : x[0].startswith('mean'),
globals().items()))
for i in lst:
    input_1.append(i)
lst1 =  map(lambda x : x[1], filter(lambda x : x[0].startswith('se'),
globals().items()))
for i in lst1:
    input_1.append(i)
input_array = np.asarray(input_1)
input_reshaped = input_array.reshape(1,-1)
predict = model.predict(input_reshaped)
if (predict[0] &lt; 0.5):
  print("breast cancer is malignant")
else:
   print('breast cancer is benign")

#predicting individual accuracy based on cm
print("accuracy with breast cancer = ",cm[0,0]/(cm[0,0]+cm[1,0]))
print("accuracy without breast cancer = ",cm[1,1]/(cm[0,1]+cm[1,0]))

#Visualization of data
dataset.groupby('diagnosis').hist(figsize=(18, 18))
cols = ['diagnosis',
        'radius_mean',
        'texture_mean',
        'perimeter_mean',
        'area_mean',
        'smoothness_mean',
        'compactness_mean',
        'concavity_mean',
        'concave points_mean',
        'symmetry_mean,
        'fractal_dimension_mean]
sns.pairplot(dataset[cols], hue='diagnosis')
plt.show()
df_pred = pd.DataFrame(df_prediction, columns=df_prediction_cols)
print(len(confusion_matrixs))
df_pred
df_pred.sort_values("score", ascending=False)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,prediction)
print(cm)
model = SVC(kernel = 'rbf', random_state = 2)
model.fit(X_train,y_train)
prediction = model.predict(X_test)
import random
a = random.random()
meanR=round(random.uniform(0,30),3)
meanT=round(random.uniform(0,50),3)
meanS=round(random.uniform(0,1),5)
meanC=round(random.uniform(0,1),5)
meanSy=round(random.uniform(0,1),4)
```

```
meanF=round(random.uniform(0,1),5)
seR=round(random.uniform(0,2),4)
seT=round(random.uniform(0,3),4)
seS=round(random.uniform(0,1),6)
seC=round(random.uniform(0,1),6)
seSy=round(random.uniform(0,1),6)
seF=round(random.uniform(0,1),6)
input_1=[]
lst =  map(lambda x : x[1], filter(lambda x : x[0].startswith(’mean’),
globals().items()))
for i in lst:
    input_1.append(i)
lst1 =  map(lambda x : x[1], filter(lambda x : x[0].startswith(’se’),
globals().items()))
for i in lst1:
    input_1.append(i)
input_array = np.asarray(input_1)
input_reshaped = input_array.reshape(1,-1)
predict = model.predict(input_reshaped)
if (predict[0] &lt; 0.5):
  print("breast cancer is malignant")
else:
   print(’breast cancer is benign")

#predicting individual accuracy based on cm
print("accuracy with breast cancer = ",cm[0,0]/(cm[0,0]+cm[1,0]))
print("accuracy without breast cancer = ",cm[1,1]/(cm[0,1]+cm[1,0]))
```

```
# to generate and visualize the correlation matrix
corr = dataset.corr().round(2)
# Mask for the upper triangle
mask = np.zeros_like(corr, dtype=bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(30, 20))
cmap = sns.diverging_palette(10, 200, as_cmap=True)
# Draw the heatmap
sns.heatmap(corr, mask=mask, cmap=cmap, vmin=-1, vmax=1, center=0,
            square=True, linewidths=.5, cbar_kws={&quot;shrink&quot;: .5}, annot=Tr
plt.tight_layout()
cols = ['radius_worst',
        'texture_worst',
        'perimeter_worst',
        'area_worst',
        'smoothness_worst',
        'compactness_worst',
        'concavity_worst',
        'concave points_worst',
        'symmetry_worst',
        'fractal_dimension_worst']
dataset = dataset.drop(cols, axis=1)
cols = ['perimeter_mean',
        'perimeter_se',
        'area_mean',
        'area_se']
```

```
dataset = dataset.drop(cols, axis=1)
cols = [’concavity_mean’,
        ’concavity_se’,
        ’concave points_mean’,
        ’concave points_se’]
dataset = dataset.drop(cols, axis=1)
#new correlation matrix
corr = dataset.corr().round(2)
mask = np.zeros_like(corr, dtype=bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(20, 20))
sns.heatmap(corr, mask=mask, cmap=cmap, vmin=-1, vmax=1, center=0,
            square=True, linewidths=.5, cbar_kws={&quot;shrink&quot;: .5}, annot=Tr
plt.tight_layout()
cols = [’diagnosis’,
        ’radius_mean’,
        ’texture_mean’,
        ’smoothness_mean’,
        ’compactness_mean’,
        ’symmetry_mean&quot;,
        ’fractal_dimension_mean’,
        ’radius_se’,
        ’texture_se’,
        ’smoothness_se’,
        ’compactness_se’,
        ’fractal_dimension_se’]
sns.pairplot(dataset[cols], hue=’diagnosis’)
```

```
plt.show()
dataframe = pd.DataFrame(Y)
#Encoding categorical data values
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
dataset.diagnosis = labelencoder_Y.fit_transform(dataset.diagnosis)
dataset.diagnosis
print(dataset.diagnosis.value_counts())
print(’\n’, dataset.diagnosis.value_counts().sum())
#finding the correlations for mean features
cols = [’diagnosis’,
        ’radius_mean’,
        ’texture_mean’,
        ’smoothness_mean’,
        ’compactness_mean’,
        ’symmetry_mean’,
        ’fractal_dimension_mean&’,
        ’radius_se’,
        ’texture_se,
        ’smoothness_se’,
        ’compactness_se,
        ’symmetry_se’,
        ’fractal_dimension_se’]
print(len(cols))
dataset[cols].corr()
plt.figure(figsize=(12, 9))
plt.title(&quot;Correlation Graph&quot;)
```

```
cmap = sns.diverging_palette( 1000, 120, as_cmap=True)
sns.heatmap(dataset[cols].corr(), annot=True, fmt=&#39;.1%&#39;,
linewidths=.05,
cmap=cmap);
#using plotly
plt.figure(figsize=(15, 10))
fig = px.imshow(dataset[cols].corr());
fig.show()
#train test splitting
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, f1_score
from sklearn.metrics import classification_report
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_validate, cross_val_score
from sklearn.svm import SVC
from sklearn import metrics
#train test splitting
prediction_feature = ['radius_mean',
                      'texture_mean',
                      'smoothness_mean',
                      'compactness_mean',
                      'symmetry_mean',
                        'fractal_dimension_mean&',
                        'radius_se',
                        'texture_se,
                        'smoothness_se',
                        'compactness_se,
                        'symmetry_se',
                        'fractal_dimension_se']
targeted_feature = 'diagnosis';
len(prediction_feature)
#spliting ths data into training and testing sets
X = dataset[prediction_feature]
X
y = dataset.diagnosis
y
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=40)
#Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
def model_building(model, X_train, X_test, y_train, y_test):
    """


    Model Fitting, Prediction And Other stuff
    return (&#39;score&#39;, &#39;accuracy_score&#39;, &#39;predictions&#39; )
    """

    model.fit(X_train, y_train)
    score = model.score(X_train, y_train)
    predictions = model.predict(X_test)
    accuracy = accuracy_score(predictions, y_test)
    return (score, accuracy, predictions)
models_list = {
    "LogisticRegression"; :  LogisticRegression(random_state=0),
    "K-NearestNeighbor"; :  KNeighborsClassifier(n_neighbors = 5, metric =
’minkowski’, p = 2),
    "SVC" :  SVC(kernel = ’rbf’, random_state = 2,C = 2),
    "SVM"; :  SVC(kernel = ’linear’, random_state = 0),
    "NaiveBayes" :  GaussianNB(),
    "DecisionTreeClassifier" :  DecisionTreeClassifier(criterion=’entropy’,
random_state=0),
    "RandomForestClassifier" :  RandomForestClassifier(n_estimators=10,
criterion=’entropy’,
print(list(models_list.keys()))
print()
print(list(models_list.values()))
def cm_metrix_graph(cm):
    sns.heatmap(cm,annot=True,fmt=&quot;d&quot;)
    plt.show()
df_prediction = []
confusion_matrixs = []
df_prediction_cols = [ ’model_name’, ’score’, ’accuracy_score’ ,
'accuracy_percentage']
for name, model in zip(list(models_list.keys()), list(models_list.values())):

    (score, accuracy, predictions) = model_building(model, X_train, X_test,
y_train, y_test )

    print("\n\nClassification Report of " "+ str(name), " '\n')

    print(classification_report(y_test, predictions))
    df_prediction.append([name, score, accuracy, "{0:.2%}".format(accuracy)])

    # For Showing Metrics
    confusion_matrixs.append(confusion_matrix(y_test, predictions))


df_pred = pd.DataFrame(df_prediction, columns=df_prediction_cols)
print(len(confusion_matrixs))
df_pred
df_pred.sort_values("score", ascending=False)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,prediction)
print(cm)
model = SVC(kernel = 'rbf', random_state = 2)
model.fit(X_train,y_train)
prediction = model.predict(X_test)
import random
a = random.random()
```

```
meanR=round(random.uniform(0,30),3)
meanT=round(random.uniform(0,50),3)
meanS=round(random.uniform(0,1),5)
meanC=round(random.uniform(0,1),5)
meanSy=round(random.uniform(0,1),4)
meanF=round(random.uniform(0,1),5)
seR=round(random.uniform(0,2),4)
seT=round(random.uniform(0,3),4)
seS=round(random.uniform(0,1),6)
seC=round(random.uniform(0,1),6)
seSy=round(random.uniform(0,1),6)
seF=round(random.uniform(0,1),6)
input_1=[]
lst =  map(lambda x : x[1], filter(lambda x : x[0].startswith(’mean’),
globals().items()))
for i in lst:
    input_1.append(i)
lst1 =  map(lambda x : x[1], filter(lambda x : x[0].startswith(’se’),
globals().items()))
for i in lst1:
    input_1.append(i)
input_array = np.asarray(input_1)
input_reshaped = input_array.reshape(1,-1)
predict = model.predict(input_reshaped)
if (predict[0] &lt; 0.5):
  print("breast cancer is malignant")
else:
   print('breast cancer is benign")

#predicting individual accuracy based on cm
print("accuracy with breast cancer = ",cm[0,0]/(cm[0,0]+cm[1,0]))
print("accuracy without breast cancer = ",cm[1,1]/(cm[0,1]+cm[1,0]))

import random
a = random.random()
meanR=round(random.uniform(0,30),3)
meanT=round(random.uniform(0,50),3)
meanS=round(random.uniform(0,1),5)
meanC=round(random.uniform(0,1),5)
meanSy=round(random.uniform(0,1),4)
meanF=round(random.uniform(0,1),5)
seR=round(random.uniform(0,2),4)
seT=round(random.uniform(0,3),4)
seS=round(random.uniform(0,1),6)
seC=round(random.uniform(0,1),6)
seSy=round(random.uniform(0,1),6)
seF=round(random.uniform(0,1),6)
input_2=[]
lst =  map(lambda x : x[1], filter(lambda x : x[0].startswith('mean'),
globals().items()))
for i in lst:
    input_2.append(i)
lst1 =  map(lambda x : x[1], filter(lambda x : x[0].startswith('se'),
\globals().items()))
```

```
for i in lst1:
    input_2.append(i)
input_array = np.asarray(input_2)
input_reshaped = input_array.reshape(1,-1)
predict = model.predict(input_reshaped)
if (predict[0] == 1):
  print("breast cancer is malignant and more chances to occur")
else:
   print("breast cancer is benign and more chances to occur")
```

5. Results and Analysis

As there are many columns in given dataset we try to reduce the columns. To reduce the columns choose the similar columns and diagnosis to check the relationship. To know the relationship we generate a scatter plot matrix.

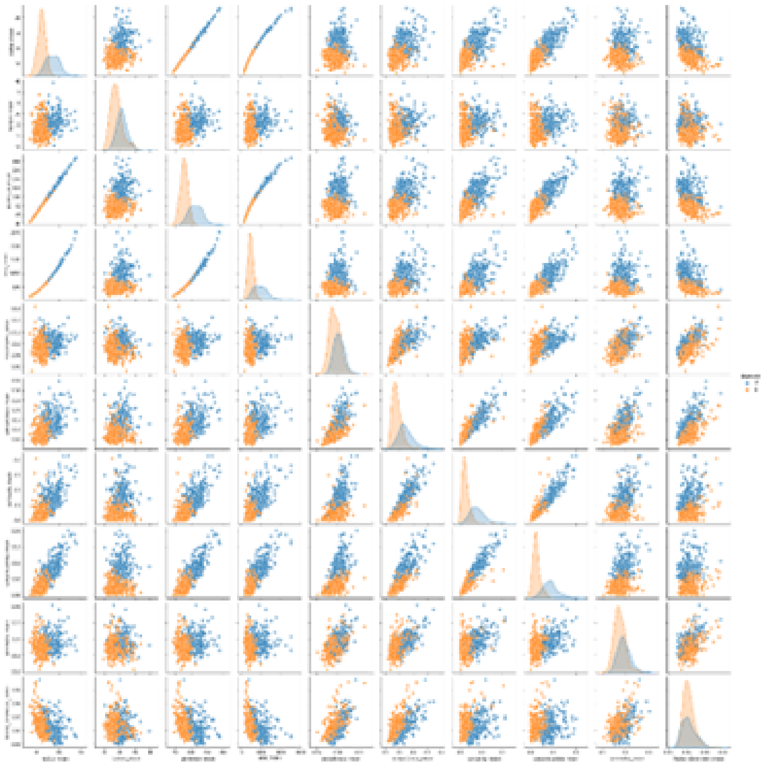


Figure 5.1: Scatter plot between attributes

We have observed that radius, perimeter, area attributes are hinting the presence of multicollinearity between these variables. Another set is concavity , concave points and compactness. Now we will generate a matrix similar to the above but instead of scatter plot we will display the correlations between the variables. Correlation matrix with all variables mean, standard errors ,worst.

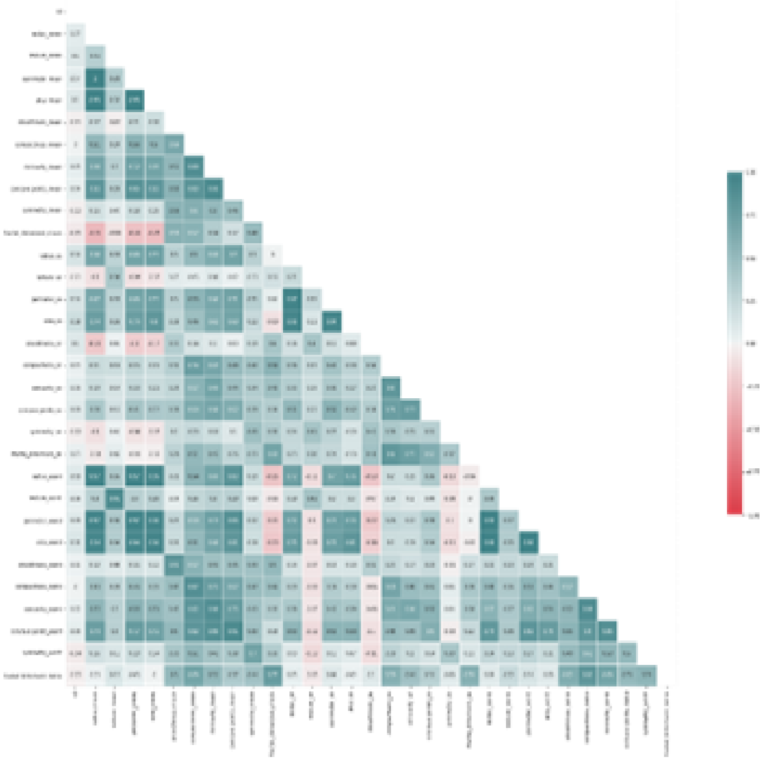


Figure 5.2: Correlation matrix between attributes

The above graph we can verify that the presence of multicollinearity between some variables. Radius has correlation of 1 and 0.99 with perimeter and area. This is because, they almost has same info of size. Therefore we can pick anyone of it. we can also see that there is multicollinearity between the mean and worst column radius-worst, radius-mean-0.97. As worst columns are subset of mean column we can drop worst col from our analysis and only focus on the mean columns. The attributes to be chosen would be radius because it is basic building block of its size. similarly we have multicollinearity between compactness, concavity, concave points from three we choose compactness because it gives info of shape. Finally we choose only 12 attributes to predict they are: radius mean, texture mean, smoothness mean, compactness mean, symmetry mean, fractal dimension mean, radius se, texture se, smoothness se, compactness se, symmetry se, fractal dimension se. The correlation matrix for 12 attributes is:

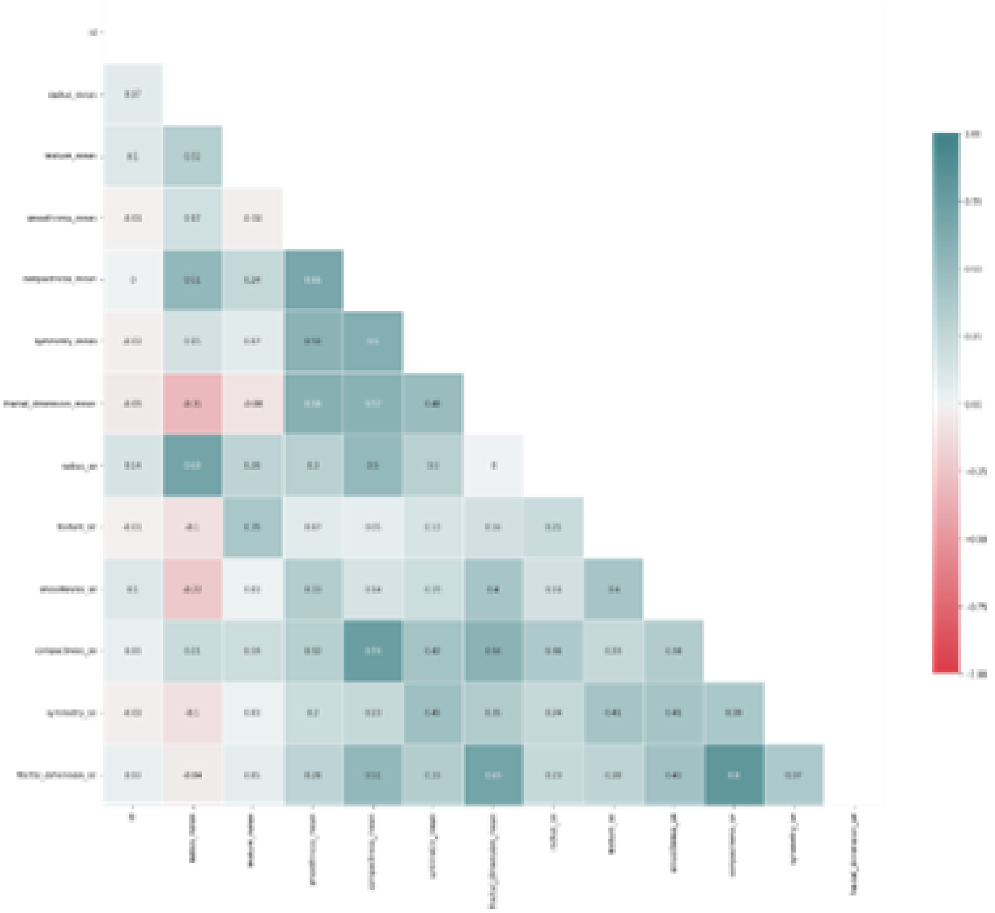


Figure 5.3: Correlation matrix for 12 attributes

After applying Machine Learning Algorithms to the Breast Cancer dataset, we used Confusion Matrix, Accuracy, Precision, Sensitivity, F1 Score, and AUC as performance metrics to evaluate and compare the models and identify the best algorithm for breast cancer Prediction. Confusion Matrix is

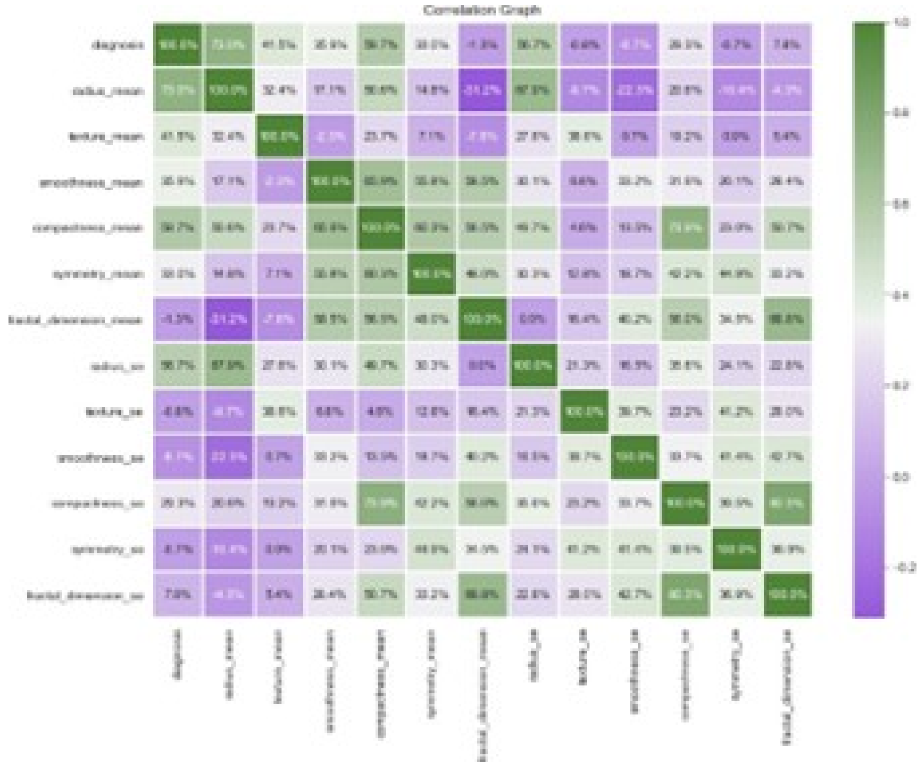
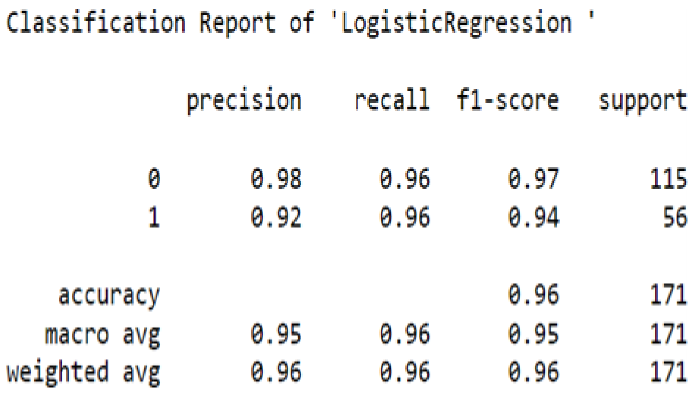


Figure 5.4: Correlation Graph

the way to measure the performance of a classification problem where the output can be of two or more types of classes. Accuracy is the most common performance metric for classification algorithms. It is defined as the number of correct predictions made as a ratio of all predictions made. Accuracy is the ratio of correctly classified samples to total samples Accuracy = (TP +TN) / (TP +TN +FP +FN)

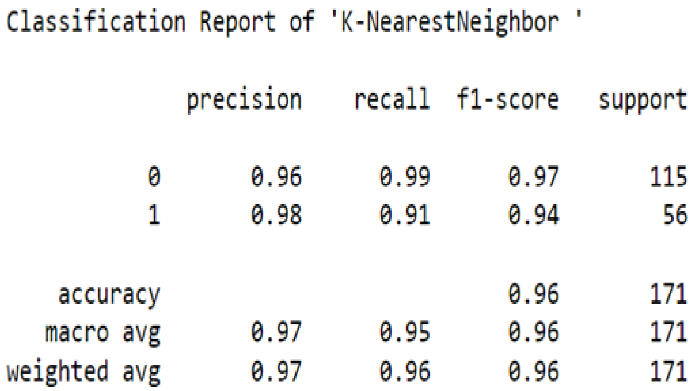
Confusion matrix for logistic regression:



```
Classification Report of 'LogisticRegression '

              precision    recall  f1-score   support

           0       0.98      0.96      0.97       115
           1       0.92      0.96      0.94        56

    accuracy                           0.96       171
   macro avg       0.95      0.96      0.95       171
weighted avg       0.96      0.96      0.96       171
```

Figure 5.5: Confusion matrix of logistic regression

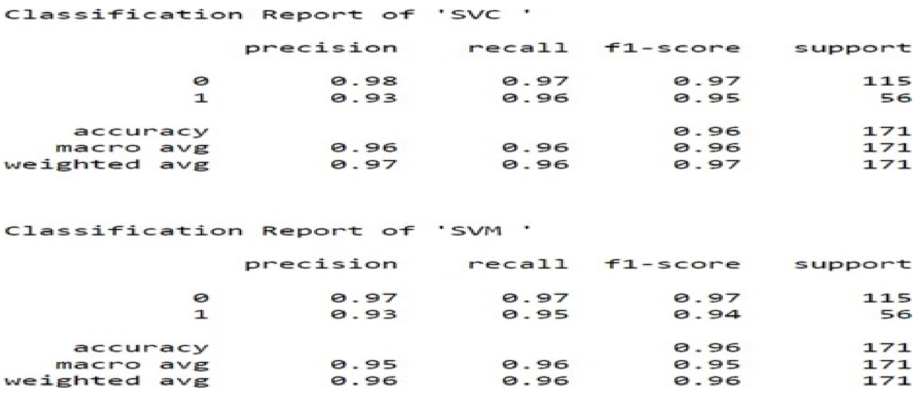
Confusion matrix for K-nearest neighbor:



```
Classification Report of 'K-NearestNeighbor '

              precision    recall  f1-score   support

           0       0.96      0.99      0.97       115
           1       0.98      0.91      0.94        56

    accuracy                           0.96       171
   macro avg       0.97      0.95      0.96       171
weighted avg       0.97      0.96      0.96       171
```

Figure 5.6: Confusion matrix for KNN

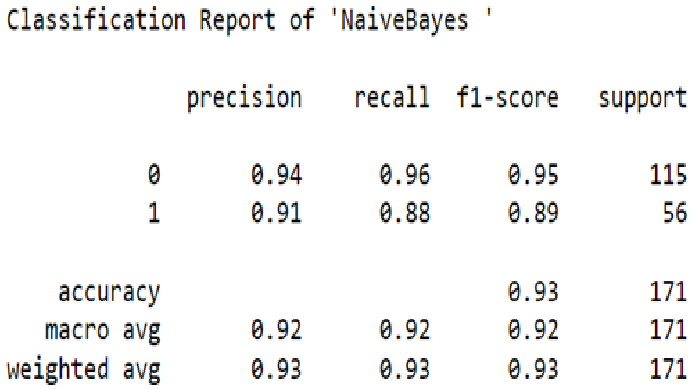
Confusion matrix for Support Vector Machine:



```
Classification Report of 'SVC '
              precision    recall  f1-score   support

           0       0.98      0.97      0.97       115
           1       0.93      0.96      0.95        56

    accuracy                           0.96       171
   macro avg       0.96      0.96      0.96       171
weighted avg       0.97      0.96      0.97       171

Classification Report of 'SVM '
              precision    recall  f1-score   support

           0       0.97      0.97      0.97       115
           1       0.93      0.95      0.94        56

    accuracy                           0.96       171
   macro avg       0.95      0.96      0.95       171
weighted avg       0.96      0.96      0.96       171
```

Figure 5.7: Confusion matrix for SVM

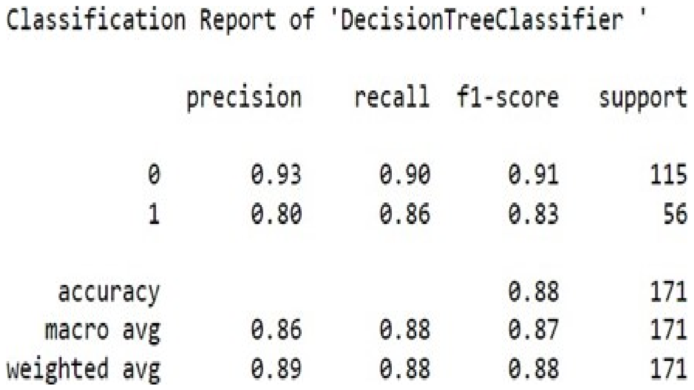
Confusion matrix for naïve bayes:



```
Classification Report of 'NaiveBayes '

              precision    recall  f1-score   support

           0       0.94      0.96      0.95       115
           1       0.91      0.88      0.89        56

    accuracy                           0.93       171
   macro avg       0.92      0.92      0.92       171
weighted avg       0.93      0.93      0.93       171
```

Figure 5.8: Confusion matrix for Naïve bayes

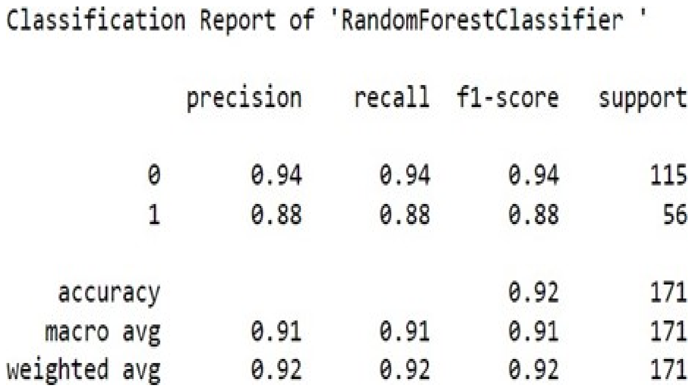
Confusion matric for decision Tree:



```
Classification Report of 'DecisionTreeClassifier '

              precision    recall  f1-score   support

           0       0.93      0.90      0.91       115
           1       0.80      0.86      0.83        56

    accuracy                           0.88       171
   macro avg       0.86      0.88      0.87       171
weighted avg       0.89      0.88      0.88       171
```

Figure 5.9: Confusion matrix for decision tree

Confusion matrix for random forest:

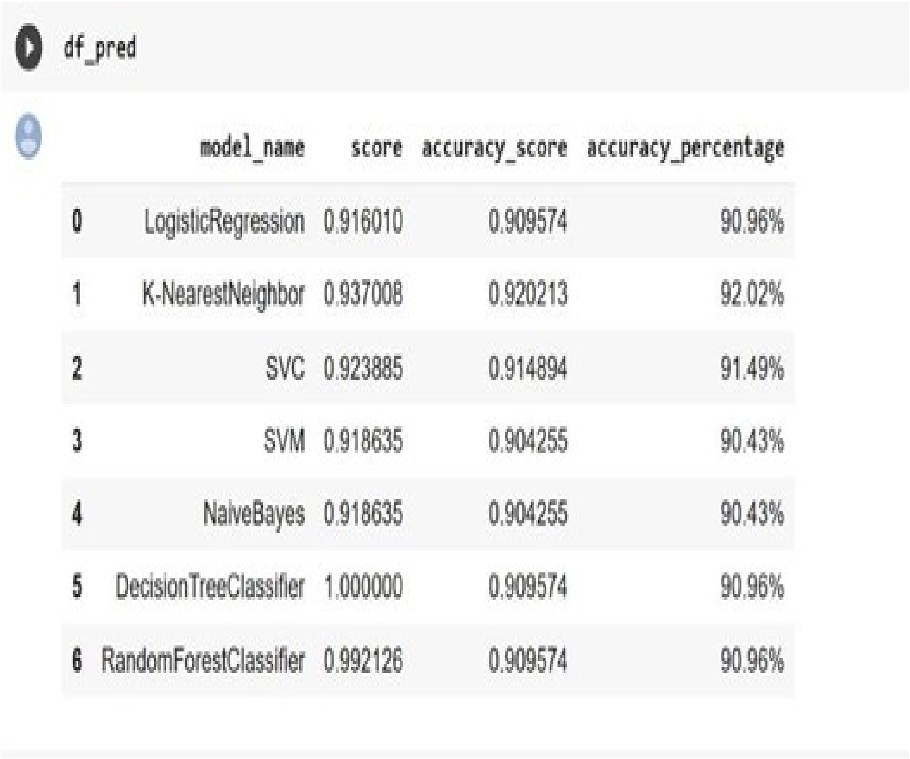


```
Classification Report of 'RandomForestClassifier '

              precision    recall  f1-score   support

           0       0.94      0.94      0.94       115
           1       0.88      0.88      0.88        56

    accuracy                           0.92       171
   macro avg       0.91      0.91      0.91       171
weighted avg       0.92      0.92      0.92       171
```

Figure 5.10: Confusion matrix for random classifier

Precision, used in document retrievals, may be defined as the number of correct documents returned by our ML model. Sensitivity may be defined as the number of positives returned by your ML model. F1 score gives us the harmonic mean of precision and Sensitivity. Mathematically, the F1 score is the weighted average of precision and Sensitivity.

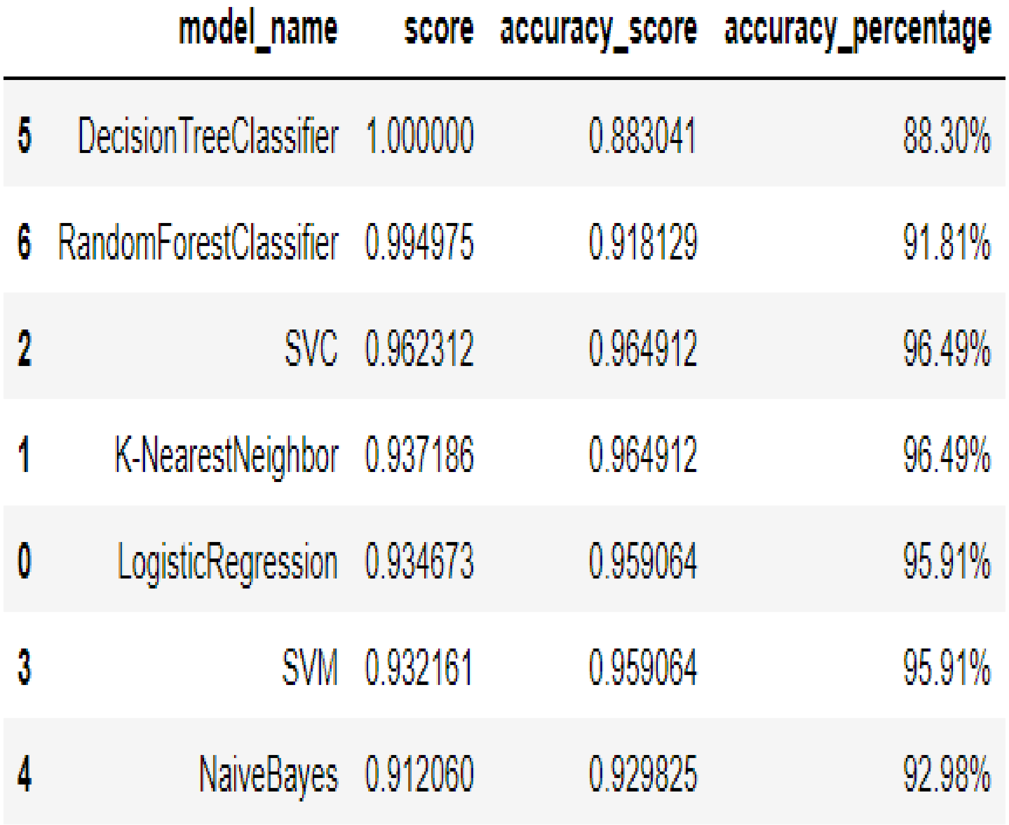
The confusion matrix shows that the Support Vector Machine predicts correctly 18849 cases out of 20,000 cases constituted of 65 % of malignant cases that are actually malignant and 30 % benign cases that are actually benign, and 5 % cases incorrectly predicted. That is why the accuracy of the Support Vector Machine is better than other classification techniques.



```
df_pred
```

	model_name	score	accuracy_score	accuracy_percentage
0	LogisticRegression	0.916010	0.909574	90.96%
1	K-NearestNeighbor	0.937008	0.920213	92.02%
2	SVC	0.923885	0.914894	91.49%
3	SVM	0.918635	0.904255	90.43%
4	NaiveBayes	0.918635	0.904255	90.43%
5	DecisionTreeClassifier	1.000000	0.909574	90.96%
6	RandomForestClassifier	0.992126	0.909574	90.96%

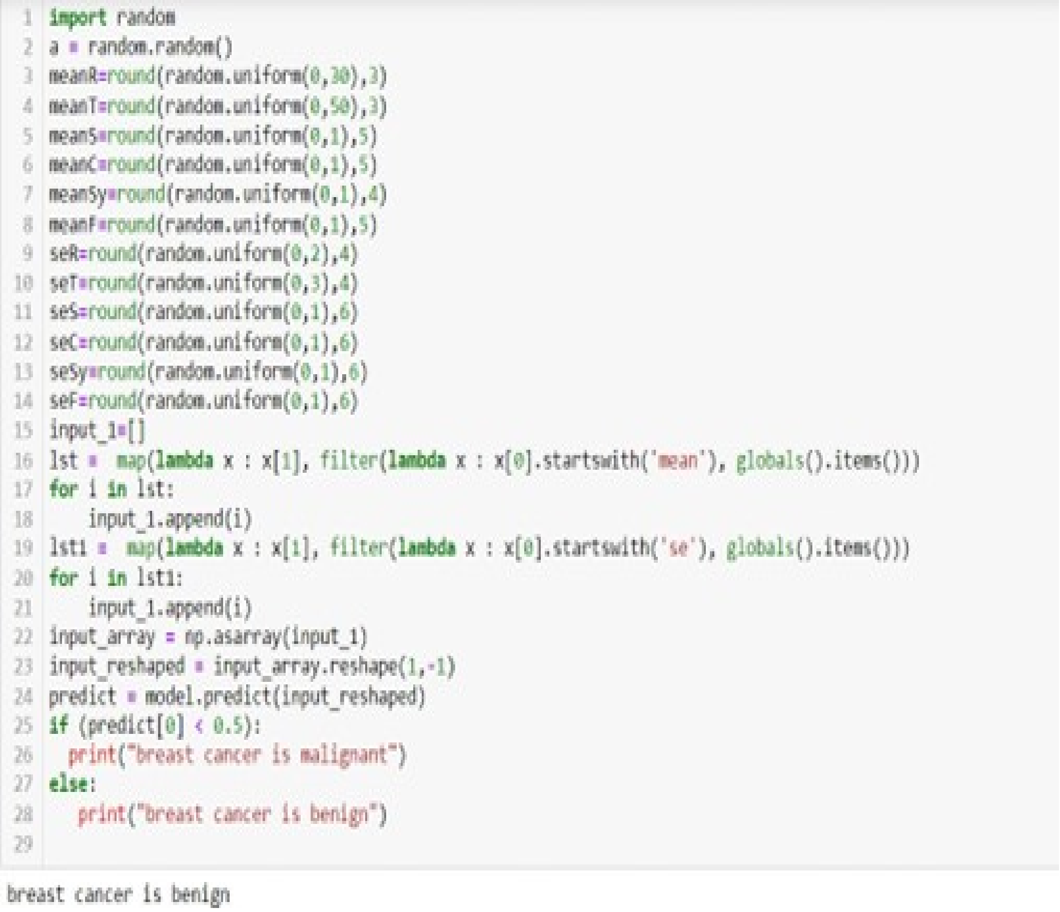
Figure 5.11: Score and Accuracy of Algorithms



	model_name	score	accuracy_score	accuracy_percentage
5	DecisionTreeClassifier	1.000000	0.883041	88.30%
6	RandomForestClassifier	0.994975	0.918129	91.81%
2	SVC	0.962312	0.964912	96.49%
1	K-NearestNeighbor	0.937186	0.964912	96.49%
0	LogisticRegression	0.934673	0.959064	95.91%
3	SVM	0.932161	0.959064	95.91%
4	NaiveBayes	0.912060	0.929825	92.98%

Figure 5.12: Order of Score and Accuracy of Algorithms

Prediction of breast Cancer:

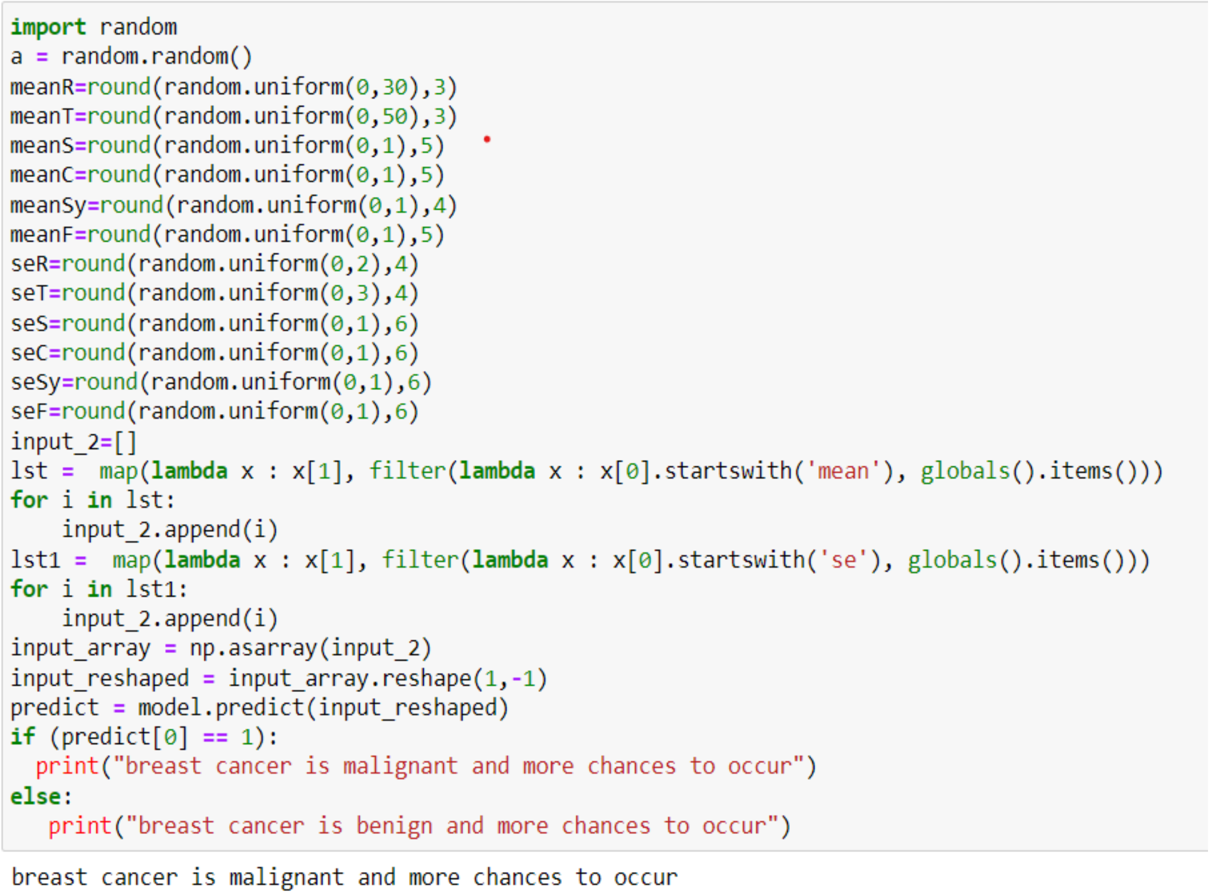


```
import random
a = random.random()
meanR=round(random.uniform(0,30),3)
meanT=round(random.uniform(0,50),3)
meanS=round(random.uniform(0,1),5)
meanC=round(random.uniform(0,1),5)
meanSy=round(random.uniform(0,1),4)
meanF=round(random.uniform(0,1),5)
seR=round(random.uniform(0,2),4)
seT=round(random.uniform(0,3),4)
seS=round(random.uniform(0,1),6)
seC=round(random.uniform(0,1),6)
seSy=round(random.uniform(0,1),6)
seF=round(random.uniform(0,1),6)
input_1=[]
lst =  map(lambda x : x[1], filter(lambda x : x[0].startswith('mean'), globals().items()))
for i in lst:
    input_1.append(i)
lst1 =  map(lambda x : x[1], filter(lambda x : x[0].startswith('se'), globals().items()))
for i in lst1:
    input_1.append(i)
input_array = np.asarray(input_1)
input_reshaped = input_array.reshape(1,-1)
predict = model.predict(input_reshaped)
if (predict[0] < 0.5):
  print("breast cancer is malignant")
else:
   print("breast cancer is benign")

```

```
breast cancer is benign
```

Figure 5.13: Prediction of breast cancer

Prediction of reoccurance:



```
import random
a = random.random()
meanR=round(random.uniform(0,30),3)
meanT=round(random.uniform(0,50),3)
meanS=round(random.uniform(0,1),5)
meanC=round(random.uniform(0,1),5)
meanSy=round(random.uniform(0,1),4)
meanF=round(random.uniform(0,1),5)
seR=round(random.uniform(0,2),4)
seT=round(random.uniform(0,3),4)
seS=round(random.uniform(0,1),6)
seC=round(random.uniform(0,1),6)
seSy=round(random.uniform(0,1),6)
seF=round(random.uniform(0,1),6)
input_2=[]
lst =  map(lambda x : x[1], filter(lambda x : x[0].startswith('mean'), globals().items()))
for i in lst:
    input_2.append(i)
lst1 =  map(lambda x : x[1], filter(lambda x : x[0].startswith('se'), globals().items()))
for i in lst1:
    input_2.append(i)
input_array = np.asarray(input_2)
input_reshaped = input_array.reshape(1,-1)
predict = model.predict(input_reshaped)
if (predict[0] == 1):
  print("breast cancer is malignant and more chances to occur")
else:
   print("breast cancer is benign and more chances to occur")
```

```
breast cancer is malignant and more chances to occur
```

Figure 5.14: Prediction of reoccurance

6. Conclusion and Future scope

We discovered that SVC and KNN provide the most accurate results for predicting breast cancer. The accuracy of this work can be improved in the future by altering the currently used machine learning approaches or by creating new algorithms. The future of breast cancer prediction is machine learning, and machine learning models are improving daily thanks to the tremendous research being done in this area by researchers. Artificial intelligence and machine learning will revolutionise the medical sector in the next decades. The prediction of breast cancer will perform better than conventional pathology testing with the addition of cutting-edge technology like convolutional neural networks.

REFERENCES

[1] R. M. Mohana, R. Delshi Howsalya Devi, Anita Bai, “Lung Cancer Detection using Nearest Neighbour Classifier”, -2019

[2] B. Akbugday, Çlassification of Breast Cancer Data Using Machine Learning Algorithms, 2019.

[3] Keles, M. Kaya, ”Breast Cancer Prediction and Detection Using Data Mining Classification Algorithms: A Comparative Study. 2019.

[4] Wang, D. Zhang and Y. H. Huang “Breast Cancer Prediction Using Machine Learning” Vol. 66, NO. 7, -2018

[5] Y. Khourdifi and M. Bahaj, ”Feature Selection with Fast Correlation-Based Filter for Breast Cancer Prediction and Classification Using Machine Learning Algorithms,2018.

[6] V. Chaurasia and S. Pal, “Data Mining Techniques: To Predict and Resolve Breast Cancer Survivability”-2014.

[7] R. K. Kavitha1, D. D. Rangasamy, “Breast Cancer Survivability Using Adaptive Voting Ensemble Machine Learning Algorithm Adaboost and CART Algorithm” -2014.