

**A PROJECT REPORT  
ON**

**A REVIEW ON BREAST CANCER PREDICTION FROM PAST DATA USING  
SUPERVISED MACHINE LEARNING**

*Submitted in partial fulfillment of the requirements for the award of the degree*

Bachelor of Technology  
in  
Computer Science & Engineering

Submitted by

<b>B KAMALESH</b>	<b>17G01A0507</b>
<b>C PRUDVI</b>	<b>17G01A0518</b>
<b>JANGAM RUCHITHA</b>	<b>17G01A0535</b>
<b>K S NIRMALA</b>	<b>17G01A0539</b>

Under the Guidance of

**Ms. K.Pujitha**, M. Tech.,  
ASSISTANT PROFESSOR



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**SRI VENKATESA PERUMAL COLLEGE OF ENGINEERING & TECHNOLOGY**  
**(AUTONOMOUS)**

**RVS Nagar, KN Road, Puttur, Chittoor (Dist.) – 517 583**

**[www.svpcet.org](http://www.svpcet.org)**

**(2017-2021)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**SRI VENKATESA PERUMAL COLLEGE OF ENGINEERING & TECHNOLOGY**  
**(AUTONOMOUS)**

**RVS Nagar, KN Road, Puttur, Chittoor (Dist) – 517 583**

**[www.svpcet.org](http://www.svpcet.org)**

**(2017-2021)**

## **CERTIFICATE**

\*\*\*\*\*

This is to certify that the project report entitled “*A REVIEW ON BREAST CANCER PREDICTION FROM PAST DATA USING SUPERVISED MACHINE LEARNING*” is being submitted by the members of batch no: **CS17A5**

<b>B KAMALESH</b>	<b>17G01A0507</b>
<b>C PRUDVI</b>	<b>17G01A0518</b>
<b>JANGAM RUCHITHA</b>	<b>17G01A0535</b>
<b>K S NIRMALA</b>	<b>17G01A0539</b>

in partial fulfillment of the requirements for the award of the degree Bachelor of Technology in *Computer Science & Engineering* from *Sri Venkatesa Perumal College of Engineering & Technology, Puttur*, affiliated to Jawaharlal Nehru Technological University Anantapur, Anantapuram. This is the bona fide work carried out by them under my guidance and supervision during the academic year 2020-21.

**PROJECT GUIDE**  
**Ms. K.Pujitha, M. Tech.**

**HEAD OF THE DEPARTMENT**  
**Mr.N.Munisankar, M.tech., (Ph.D)**

*Submitted for the viva-voce examination held on .....*

*Internal Examiner*

*External Examiner*

## **DECLARATION BY PROJECT GUIDE**

I hereby declare that the project report entitled “*A REVIEW ON BREAST CANCER PREDICTION FROM PAST DATA USING SUPERVISED MACHINE LEARNING*” is the bonafide work carried out by the members of batch no.**CS17A5** of *Sri Venkatesa Perumal College of Engineering & Technology, Puttur* for the award of degree **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2017-21 is original work and the project has not formed the basis for the award of any degree, diploma, associate fellowship or any other similar title submitted previously.

### **PROJECT GUIDE**

**Ms. K.Pujitha, M. Tech.**

Assistant Professor

## **DECLARATION BY PROJECT MEMBERS**

We hereby combinedly declare that the project entitled “*A REVIEW ON BREAST CANCER PREDICTION FROM PAST DATA USING SUPERVISED MACHINE LEARNING*” submitted by

Batch no. **CS17A5** for the award of our degree in **B. Tech Computer Science & Engineering** is our original work and the project has not formed the basis for the award of any degree, diploma, associate fellowship or any other similar title submitted previously.

B KAMALESH  
(17G01A0507)

C PRUDVI  
(17G01A0518)

JANGAM RUCHITHA  
(17G01A0535)

K S NIRMALA  
(17G01A0539)

**Place:**

**Date:**

## **ACKNOWLEDGEMENT**

The satisfaction and euphoria accompany the successful completion of task and would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deepest sense of gratitude and pay our sincere thanks to our project guide **Ms. K.Pujitha, M. Tech.** Assistant Professor, Department of CSE, who evinced keen interest in our efforts and provided his valuable guidance throughout our project work.

We also express our sincere gratitude to **Mr.N.Munisankar, M. Tech., (Ph.D)** Head of the Department of CSE for his great encouragement and valuable support throughout our study.

We owe our gratitude to our Principal **Dr. T. Sunil kumar Reddy., M. Tech., (Ph.D)** for his kind attention and valuable guidance given to us throughout this course.

We sincerely and whole heartedly thank to our beloved **Sri. Ravuri V Balaji**, Vice-Chairman for giving art of infrastructure facilities to us throughout our course study and leading to successful completion of our project.

We are very much thankful to our beloved **Dr. R. Venkataswamy** Chairman of Sri Venkatesa Perumal College of Engineering & Technology, Puttur for his kind attention and valuable guidance to us throughout the course.

We also thankful to all staff members of CSE Department for helping me to complete this project work by giving valuable suggestions.

We would like to thankful the members of our family who assisted in the preparation of this report financially.

The last but not least we express our sincere thanks to all our friends who have supported us in the accomplishment of this project.

## **ABSTRACT**

*Breast Cancer is considered as one of the deadliest and chronic diseases which cause an increase in blood sugar. Many complications occur if Breast Cancer remains untreated and unidentified. The tedious identifying process results in visiting of a patient to a diagnostic centre and consulting doctor. But the rise in machine learning approaches solves this critical problem. The motive of this study is to design a model which can prognosticate the likelihood of Breast Cancer in patients with maximum accuracy. Therefore three machine learning classification algorithms namely SVM and Logistic Regression are used in this experiment to detect Breast Cancer at an early stage.*

## LIST OF FIGURES

<b>S. No.</b>	<b>Name of the Figure</b>	<b>Page No.</b>
1.	Fig: 3.1. SOFTWARE ARCHITECTURE	8
2.	Fig: 4.2.1. Use Case Diagram	14
3.	Fig: 4.2. Class Diagram	15
4.	Fig: 4.3. Sequence Diagram	16
5.	Fig: 4.4. Collaboration Diagram	17
6.	Fig: 4.6. Deployment Diagram	18
7.	Fig: 4.5. Activity Diagram	19

# LIST OF CONTENTS

<b>CHAPTER</b>	<b>PAGE NO</b>
Title page	i
Certificate	ii
Declaration by the Project Guide	iii
Declaration by Project Members	iv
Acknowledgement	V
Abstract	vi
List of figures	vii
<b>1. INTRODUCTION</b>	
<b>1.1 Existing System</b>	<b>3</b>
<b>1.2 Proposed System</b>	<b>3</b>
<b>2. LITERATURE SURVEY</b>	<b>5</b>
<b>3. SYSTEM ANALYSIS</b>	
<b>3.1 Input Output Design</b>	<b>9</b>
<b>3.1.1 Input Design</b>	<b>9</b>
<b>3.1.2 Output Design</b>	<b>10</b>
<b>3.2 Modules</b>	<b>11</b>
<b>3.3 Feasibility Study</b>	<b>11</b>
<b>3.3.1 Economical Feasibleness</b>	<b>12</b>
<b>3.3.2 Technical Feasibleness</b>	<b>12</b>
<b>3.3.3 Social feasibility</b>	<b>13</b>
<b>3.4 System configuration</b>	<b>13</b>
<b>4. SYSTEM DESIGN</b>	
<b>4.1 UML</b>	<b>14</b>
<b>4.2 UML Diagrams</b>	<b>15</b>
<b>4.2.1 Use Case Diagram</b>	<b>15</b>
<b>4.2.2 Class Diagram</b>	<b>16</b>
<b>4.2.3 Sequence Diagram</b>	<b>17</b>
<b>4.2.4 Collaboration Diagram</b>	<b>18</b>
<b>4.2.5 Deployment Diagram</b>	<b>19</b>

---



4.2.6 Activity diagram	20
4.2.7 Component Diagram	21
4.3 Algorithms	21
4.3.1 Logistic regression	21
4.3.2 KNN Algorithm	22
4.3.3 SVM Algorithm	23
5. SYSTEM IMPLEMENTATION	
5.1 Description of Technologies Used	24
5.2 Sample Code	30
6. SYSTEM TESTING	
6.1 Testing Activities	36
6.1.1 Unit testing	36
6.1.2 Integration testing	37
6.1.3 Functional testing	37
6.1.4 System Testing	37
6.1.5 Acceptance testing	38
6.2 Types of Testing	
6.2.1 White Box Testing	38
6.2.2 Black Box Testing	38
7. CONCLUSION & FUTURE ENHANCEMENTS	
7.1 Conclusion	39
7.2 Future Enhancements	39
8. REFERENCES	40
APPENDIX	42

# Chapter I

---

## Introduction

---

## 1. INTRODUCTION

The second major cause of women's death is breast cancer (after lung cancer). 246,660 of women's new cases of invasive breast cancer are expected to be diagnosed in the US during 2016 and 40,450 of women's death is estimated. Breast cancer is a type of cancer that starts in the breast. Cancer starts when cells begin to grow out of control. Breast cancer cells usually form a tumour that can often be seen on an x-ray or felt as a lump. Breast cancer can spread when the cancer cells get into the blood or lymph system and are carried to other parts of the body. The cause of Breast Cancer includes changes and mutations in DNA. There are many different types of breast cancer and common ones include ductal carcinoma in situ (DCIS) and invasive carcinoma. Others, like phyllodes tumours and angiosarcoma are less common. There are many algorithms for classification of breast cancer outcomes.

The side effects of Breast Cancer are – Fatigue, Headaches, Pain and numbness (peripheral neuropathy), Bone loss and osteoporosis. There are many algorithms for classification and prediction of breast cancer outcomes. The present paper gives a comparison between the performance of four classifiers: SVM , Logistic Regression , Random Forest and kNN which are among the most influential data mining algorithms. It can be medically detected early during a screening examination through mammography or by portable cancer diagnostic tool. Cancerous breast tissues change with the progression of the disease, which can be directly linked to cancer staging. The stage of breast cancer (I–IV) describes how far a patient's cancer has proliferated. Statistical indicators such as tumour size, lymph node metastasis, and distant metastasis and so on are used to determine stages. To prevent cancer from spreading, patients have to undergo breast cancer surgery, chemotherapy, radiotherapy and endocrine. The goal of the research is to identify and classify Malignant and Benign patients and intending how to parametrize our classification techniques hence to achieve high accuracy. We are looking into many datasets and how further Machine Learning algorithms can be used to characterize Breast Cancer. We want to reduce the error rates with maximum accuracy. 10-fold cross validation test which is a Machine Learning Technique is used in JUPYTER to evaluate the data and analyse data in terms of effectiveness and efficiency.

Breast cancer remains to be the outmost identified cancer in the whole universe and is the prime source of cancer demise amid women. Earlier detection of breast cancer can save many lives in a best effective manner. Without reaching for surgical biopsy, prompt diagnosis entails accurate and steadfast diagnosis processthat permits medical practitioner to differentiate benign breast

tumors from malignant tumors. Every single minute, anywhere all over the globe context of breast cancer is identified amongst females and every one minute, everywhere all over the globe and somebody expires from breast cancer. Breast tumors can be identified and further classified into three different categories known as benign breast cancers, in situ cancers, and invasive cancers. Benign breast tumors fall into the Chief category of tumors often detected by undergoing mammography. They cannot extent to external organs as by nature they are non-cancerous. By means of mammography in rare cases, this one is hard to discriminate specific bulk of benign from malignant lesions. In situ or noninvasive cancer, is fully confined in the ducts. . Also, the basal membrane should be free from malignant cells. Coming to invasive, if the cancer spreads above the basal membrane and overreaches into the neighboring tissue. Consequently, early detection of breast cancer is crucial. To classify patients into either non-cancerous group called into "benign" or cancerous group into "malignant" is the prime purpose of these forecast Machine learning permits processors to learn from prior instances, to intimate complex patterns from huge, noisy or compound data sets. It is a separation of artificial intelligence which employs a range of optimized, statistical and probabilistic techniques. In general, this expertise is fine suitable to health applications, definitely those which relies on difficult genomic and proteomic measurements. Currently machine learning methodologies are being applied to detect and classify tumors in broad range of medical solicitations. To diagnose and assist cancer, machine learning has been used initially and foremost. It permits interpretations or conclusions to be made that could not rather be made using standard statistical procedures. . Therefore, it is intensely more influential because of this reason . Remarkably, supervised learning is applied in more or less all algorithms of machine learning used in diagnosing prognosis and tumor prediction.

According to World health organization, Breast cancer is the most frequent cancer among women and it is the second dangerous cancer after lung cancer. In 2018, from the research it is estimated that total 627,000 women lost their life due to breast cancer that is 15% of all cancer deaths among women In case of any symptom, people visit to oncologist. Doctors can easily identify breast cancer by using Breast ultrasound, Diagnostic mammogram, Magnetic resonance imaging (MRI), Biopsy. . Based on these test results, doctor may recommend further tests or therapy. Early detection is very crucial in breast cancer. If chances of cancer are predicted at early stage then survivability chances of patient may increase.

**1.1 Existing system:**

This paper is to address this important problem and design a cloud-assisted privacy. It preserving health records to protect the privacy of the involved parties and their data. Thresholding method, K means clustering, manual analysis. The outsourcing decryption technique and a newly proposed key private proxy re encryption are adapted.

**Disadvantages:**

- No security for user's data. No authentication or security provided
- High resource costs needed for the implementation.
- Medical Resonance images contain a noise caused by operator performance which can lead to serious inaccuracies classification.

**1.2 Proposed System:**

Classification is one of the most important decision making techniques in many real world problem. In this work, the main objective is to classify the data as diabetic and improve the classification accuracy. For many classification problem, the higher number of samples chosen but it doesn't leads to higher classification accuracy. In many cases, the performance of algorithm is high in the context of speed but the accuracy of data classification is low. Classification accuracy can be increase if we use much of the data set for training and few data sets for testing. This survey has analysed various classification techniques for classification of diabetic and non-diabetic data. Thus, it is observed that techniques like Support Vector Machine, Logistic Regression, and are most suitable for implementing the Breast Cancer prediction system.

Machine learning (ML) algorithms: Support machine vector (SVM), Decision Tree (DT), Random Forest (RT), Artificial Neural Net- works (ANN), Naive Bayes (NB), Nearest Neighbour (NN) search. The data-set used is obtained from the Wisconsin datasets. For the implementation of the ML algorithms, the dataset was partitioned into the training set and testing set. A comparison between all the six algorithms will be made. The algorithm that gives the best results will be supplied as a model to the website.

---

The website will be made from a python framework, called flask. And it will host the database on Xampp or Firebase or inbuilt Python and flask libraries. This data set is available on the UCI Machine Learning Repository. It consists of 32 real world attributes which are multivariate. The total number of instances is 569 and there are no missing values in this data set. The process of the proposed system is as follows,

- The patient books an appointment through our website.
- The patient will then meet the doctor offline for the respective appointment.
- The doctor will first check the patient manually, then perform a breast mammogram or an ultrasound. That ultra sound will show an image of the breast consisting the lumps or not.
- If the lumps are detected, a biopsy will be performed. The digitised image of the Fine Needle Aspirate (FNA) is what forms the features of the dataset.
- Those numbers will be provided to the system by the doctor and the model will detect if its a benign or a malignant cancer.
- The report will be then forwarded to the patient on their respective account.

This survey has analyzed various classification techniques for classification of diabetic and non-diabetic data. Thus, it is observed that techniques like Support Vector Machine, Logistic Regression, and are most suitable for implementing the Breast Cancer prediction system.

**Advantages:**

- High accuracy, fastest prediction, and consistency of results.
- It can segment the Breast Cancer regions from the data accurately.
  - It is useful to classify the lung Tumor from trained data set for accurate detection.
- Women who receive high estimated risks could be motivated to seek out a doctor or take other preventative actions.
- Our models could easily be incorporated into phone application or website breast cancer risk prediction tools.

## Chapter II

---

# Literature survey

---

---

## 2.LITERATURE SURVEY

Data mining is been applied on medical data of the past and current research papers. Thorough study is done on various base reports. Jacob et al. have compared various classifier algorithms on Wisconsin Breast Cancer diagnosis dataset. They came across that Random Tree and SVM classification algorithm produce best result i.e. 100% accuracy. However they mainly worked on 'Time' feature along with other parameters to predict the outcome of non-recurrence or recurrence of breast cancer among patients. In this paper, "Time" feature has not been relied upon for prediction of recurrence of the disease. Here, prediction is based on "Diagnosis" feature of WBCD dataset.

Chih-Lin Chi et al. used the ANN model for Breast Cancer Prognosis on two dataset. They predicted recurrence and non-recurrence based on probability of breast cancer and grouped patients with bad (5 years) prognoses. Delen et al. used the SEER dataset of breast cancer to predict the survivability of a patient using 10-fold cross validation method. The result indicated that the decision tree is the best predictor with 93.6% accuracy on the dataset as compared to ANN and logistic regression model.

Predicting the early detection of chronic Cancer disease also known as chronic renal disease for Cancer patients with the help of machine learning methods and finally suggests a decision tree to arrive at concrete results with desirable accuracy by measuring its performance to its specification and sensitiveness. In order to increase the accuracy of the prediction result, we have utilized algorithms such as neural network and clustering data which greatly helped in our mission and also gave scope for future work.

This Paper predicts Breast Cancer for using Classification Techniques. The detailed information about Cancer diseases such as its Facts, Common Types, and Risk Factors has been explained in this paper. The Data Mining tool used is WEKA (Waikato Environment for Knowledge Analysis), a good Data Mining Tool for Bioinformatics Fields. The all three available Interface in WEKA is used here. Naive Bayes, Artificial Neural Networks and Decision Tree (J48) are Main Data Mining Techniques and through this techniques Breast Cancer is predicted in this System.



**Author: S. Gokhale**

### **Ultrasound characterisation of breast masses**

written by proposed a system where they found that doctors have known and experienced that breast cancer occurs when some breast cells begin to grow abnormally. These cells divide more briskly and disperse faster than healthy cells do and continue to accumulate, forming a lump or mass that may start causing pain. Cells may spread rapidly through your breast to your lymph nodes or to other parts of your body. Some women can be at a higher risk for breast cancer

because of their family history, lifestyle, obesity, radiation, and reproductive factors. In the case of cancer, if the diagnosis occurs quickly, the patient can be saved as there have been advances in cancer treatment. In this study we use four machine learning classifiers which are Naive Bayesian Classifier, k-Nearest Neighbour, Support Vector Machine, Artificial Neural Network and random forest.

**Author: Pragya Chauhan and Amit Swami**

### **Breast Cancer Prediction Using Genetic Algorithm Based Ensemble Approach**

proposed a system where they found that Breast cancer prediction is an open area of research. In this paper different machine learning algorithms are used for detection of Breast Cancer Prediction. Decision tree, random forest, support vector machine, neural network, linear model, adaboost, naive bayes methods are used for prediction.

An ensemble method is used to increase the prediction accuracy of breast cancer. New technique is implemented which is GA based weighted average ensemble method of classification dataset which overcame the limitations of the classical weighted average method. Genetic algorithm based weighted average method is used for the prediction of multiple models. The comparison between Particle swarm optimization (PSO), Differential evolution (DE) and Genetic algorithm (GA) and it is concluded that the genetic algorithm outperforms for weighted average methods. One more comparison between classical ensemble method and GA based weighted average method and it is concluded that GA based weighted average method outperforms.

**Author: Abien Fred M. Agarap**

**On Breast Cancer Detection: An Application of Machine Learning Algorithms on the Wisconsin Diagnostic Dataset**

In this paper, six machine learning algorithms are used for detection of cancer. GRU-SVM model is used for the diagnosis of breast cancer GRU- SVM, Linear Regression, Multilayer Perceptron (MLP), Nearest Neighbour (NN) search, Softmax Regression, and Support Vector Machine (SVM) on the Wisconsin Diagnostic Breast Cancer (WDBC) dataset by measuring their classification test accuracy, and their sensitivity and specificity values. The said dataset consists of features which were computed from digitised images of FNA tests on a breast mass. For the implementation of the ML algorithms, the dataset was partitioned in the following fashion 70 percent for training phase, and 30 percent for the testing phase. Their results were that all presented ML algorithms exhibited high performance on the binary classification of carcinoma, i.e. determining whether benign tumour or malignant tumour. Therefore, the statistical measures on the classification problem were also satisfactory.

**Author: Priyanka Gandhi and Prof. Shalini**

**Analysis of Machine Learning Techniques for Breast Cancer Prediction**

In this paper, ML techniques are explored in order to boost the accuracy of diagnosis. Methods such as CART, Random Forest, K-Nearest Neighbours are compared. The dataset used is acquired from UC Irvine Machine Learning Repository. It is found that KNN algorithm has much better performance than the other techniques used in comparison. The most accurate model was K-Nearest Neighbour. The classification model such as Random Forest and Boosted Trees showed the similar accuracy.

**Author: Muhammet Fatih Aslan, Yunus Celik, Kadir Sabanci, and Akif Durdu**

**Breast Cancer Diagnosis by Dierent Machine Learning Methods Using Blood Analysis Data**

During this paper, four dierent machine learning algorithms are used for the early detection of carcinoma. The aim of this project is to process the results of routine blood analysis with dierent ML methods. Methods used are Artificial Neural Network (ANN),

Extreme Learning Machine (ELM), Support Vector Machine (SVM) and Nearest Neighbor (k-NN). Dataset is taken from the UCI library. In this dataset age, BMI, glucose, insulin, homeostasis model assessment (HOMA), leptin, adiponectin, resistin, and chemokine monocyte chemoattractant protein (MCP1) attributes were used. Parameters that have the best accuracy values were found by using four different Machine Learning techniques. This dataset includes age, BMI, glucose, insulin, HOMA, leptin, adiponectin, resistin and MCP1 features that can be acquired in routine blood analysis. The significance of these data in breast cancer detection was investigated by ML methods.

The analysis was performed with four different ML methods. k-NN and SVM methods are determined using Hyperparameter optimization technique. The highest accuracy and lowest training time were given by ELM which was 80%. and 0.42 seconds.

**Author: Yixuan Li and Zixuan Chen**

### **Study for breast cancer datasets**

The study firstly collects the data of the BCCD dataset which contains 116 volunteers with 9 attributes and data of WBCD dataset which contains 699 volunteers and 11 attributes. Then we preprocesses the raw data of WBCD dataset and obtained the info that contains 683 volunteers with nine attributes and therefore the index indicating whether the volunteer has the malignant tumour. After comparing the accuracy, F- measure metric and ROC curve of 5 classification models, the result has shown that RF is chosen as the primary classification model during this study. Therefore, the results of this study provide a reference for experts to distinguish the character of carcinoma. In this study, there are still some limitations that ought to be solved in further work.

For instance, though there also exist some indices people haven't found yet, this study only collects the info of 10 attributes during this experiment. The limited data has an impact on the accuracy of results. additionally, the RF can

also be combined with other data mining technologies to get more accurate and efficient results in the longer term work.

# Chapter III

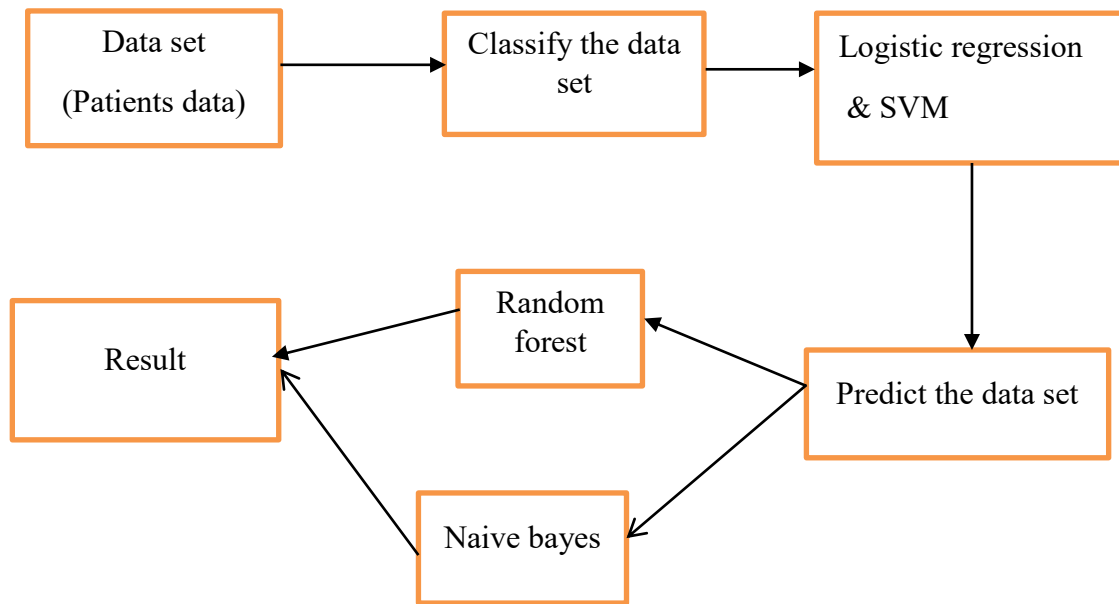
---

## SYSTEM ANALYSIS

---

### 3.SYSTEM ANALYSIS

#### 3.1 SYSTEM ARCHITECTURE:



##### 3.1.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

---

**OBJECTIVES:**

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

1. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
2. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**3.1.2 OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the future
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.

- 
- ❖ Confirm an action.

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

## 3.2 MODULES:

### **Data Collection:**

Here we collect the three type disease data's from the internet. Then we download the dataset from Google dataset search engine. In this dataset contain three types' disease symptoms like heart and diabetics and cancer.

### **Data – Pre-processing:**

In this module we remove all the unnecessary things from the dataset. If you provide the incomplete knowledge to the machine it won't to be accuracy. That's why we remove Nan value. That means Non Attribute number.

### **Data splitting:**

Here we split the dataset into two types one is trained data and another one is test data. Here how we split means using from sklearn.model\_selection import train\_test\_split this package. In our project we split the dataset into trained data is 80% and test data 20%.

### **Classification:**

In this module we apply the algorithm for Logistic Regression and Svm algorithms. It should provide the high accuracy compare to existing algorithm.

## 3.3FEASIBILITY STUDY

Feasibility study is a compressed capsule version of scope and an objective is confirmed and corrected any constraints imposed on the system are identified. There is a need to know the likelihood the system will be useful to the organization that can be obtained through efficient and effective feasibility study The important tests feasibility studies are:

- Technical feasibility
- Operational feasibility
- Economical feasibility

### **3.3.1 Economic Feasibility:**

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would add from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits and account of automation. These could include increased customer satisfaction, improvement in product quality better decision of Agriculture crop price analysis making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale. A procedure that identifies describes and evaluates the proposed system, selects the best system for the job is called Feasibility study.

### **3.3.2 Technical Feasibility:**

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while during a technical analysis.

- Understand the different technologies involved in the proposed system Before commencing the project, we have to be very clear about what are the technologies those are required for the development of new system.
- Find out whether the organization currently possesses the required technologies Is the required technology available with the organization? If so is the capacity sufficient? For instance – “Will the current printer be able to handle the new reports and forms required for the new system?”



### 3.3.3 Social Feasibility:

Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to implementation? Here are questions that will help test the operational feasibility of a project: Is there sufficient support for the project from management from users?

- If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
- Are the current business methods acceptable to the user?
- If they are not, users may welcome a change that will bring about a more operational and useful systems.

## 3.4 SYSTEM CONFIGURATION

### 3.4.1 SOFTWARE SPECIFICATION:

The following are the minimum software requirements to run this application are:

- Operating System : Windows 7 and above
- Coding Language : Python
- IDE : Jupyter Notebook
- Packages : Matplotlib, Pandas, Scikit-Learn, Seaborn, Numpy

### 3.4.2 HARDWARE SPECIFICATION:

The following are the minimum hardware requirements to run this application are:

- Processor : Dual Core
- Hard Disk : 100GB and Above

## Chapter IV

---

# SYSTEM DESIGN

---

## 4.SYSTEM DESIGN

### 4.1 UML

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

- The goal is for UML to become a common language for creating models of object oriented computer software. In its current form 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 associated with, UML.
- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.
- The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

#### GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

## 4.2 UML DIAGRAM

### 4.2.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

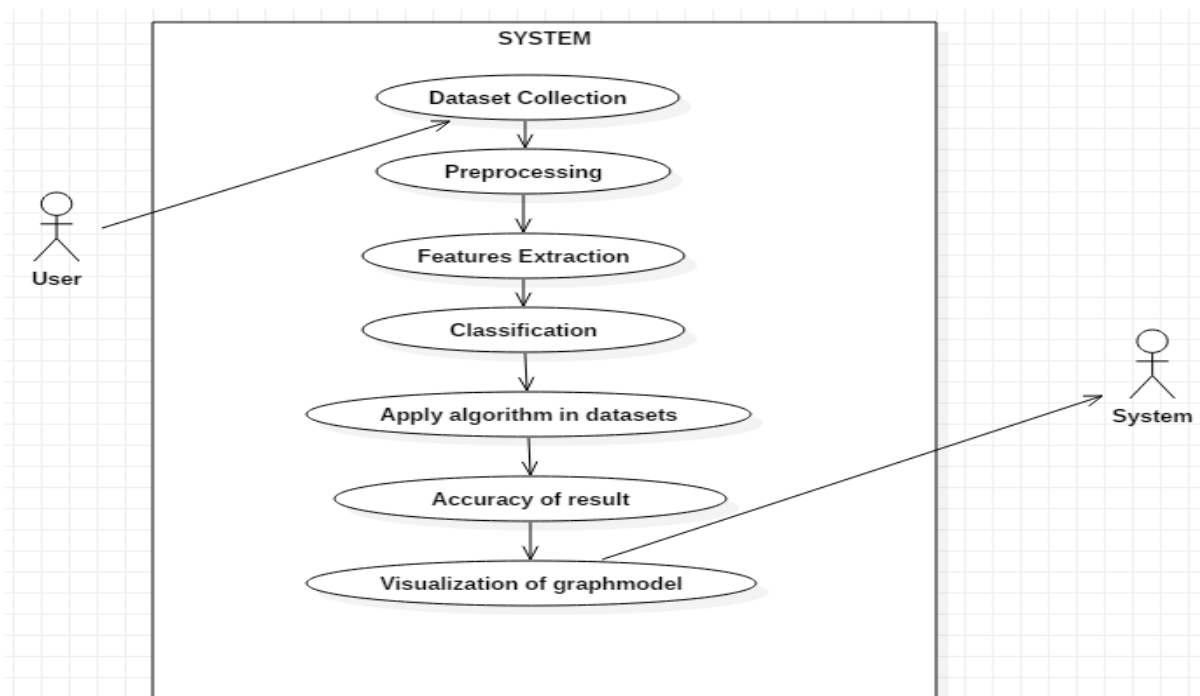


Fig: Use case diagram

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases.

We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications.

Use case diagrams are drawn to capture the functional requirements of a system. understand the dynamics of a system, we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors.

### 4.2.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

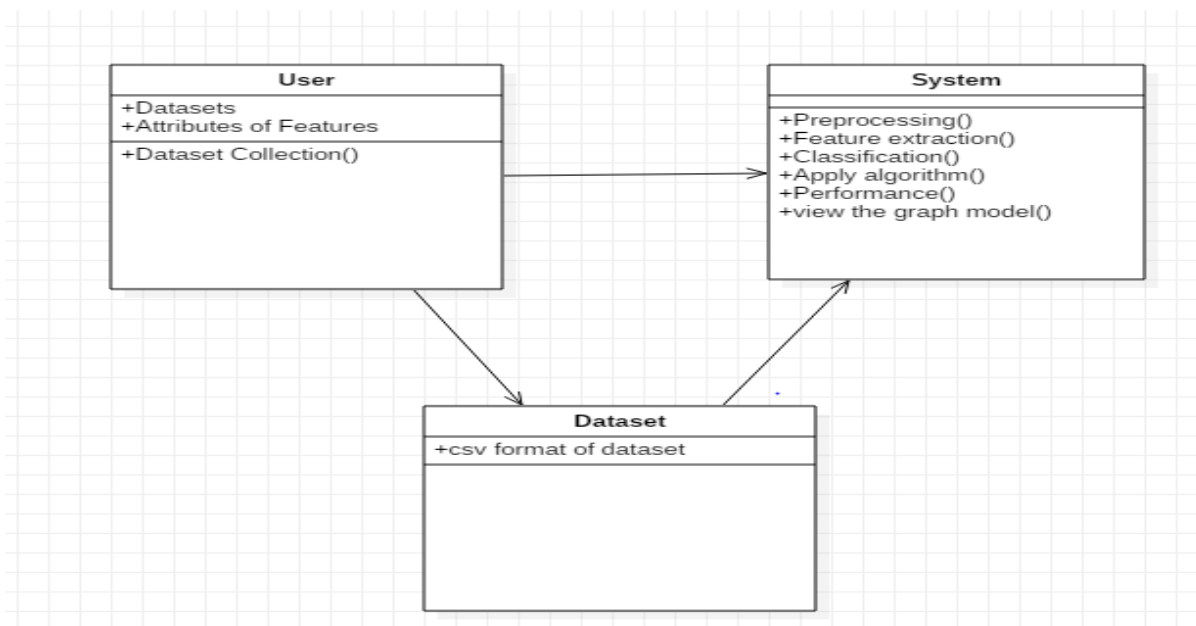


Fig: Class diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. In detailed modeling, the classes of the conceptual design are often split into subclasses.

### 4.2.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams

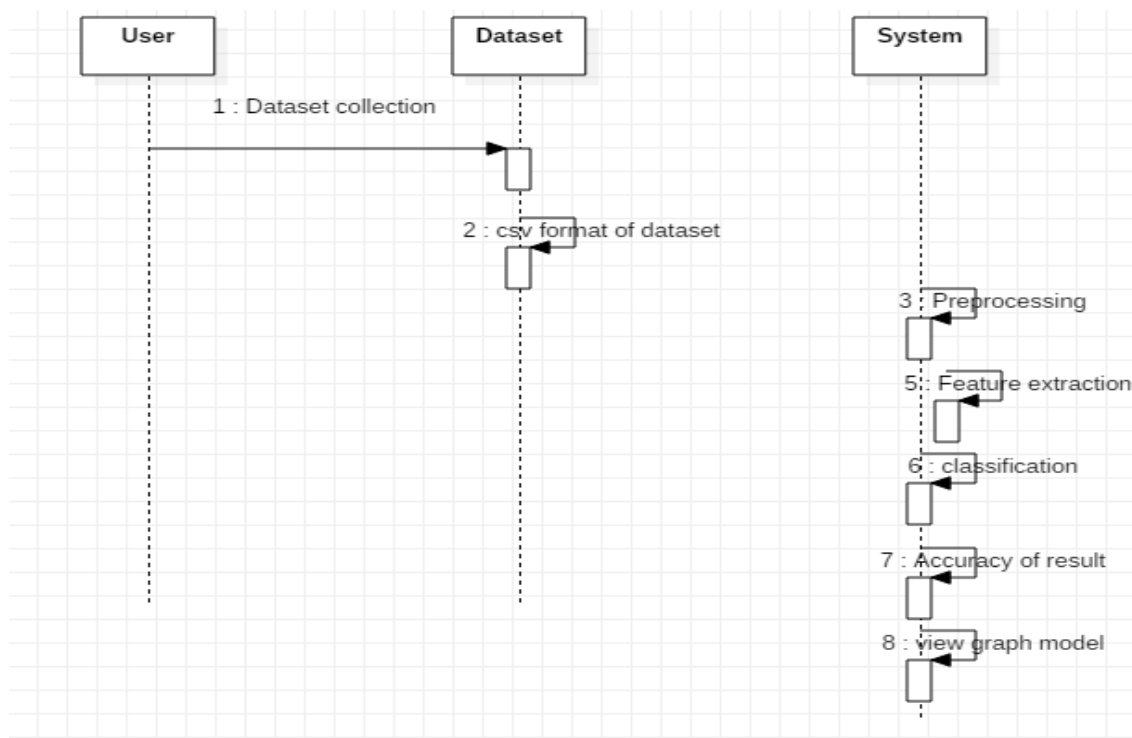


Fig: Sequence diagram

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent what messages are sent.

#### 4.2.4 COLLABORATION DIAGRAM

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

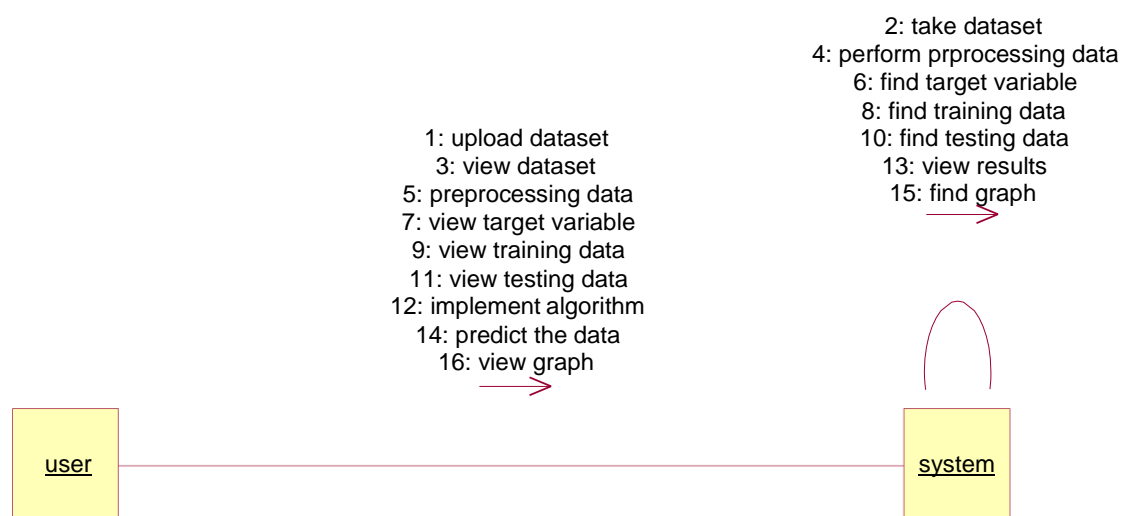


Fig:Collaboration diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

### 4.2.5 DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application

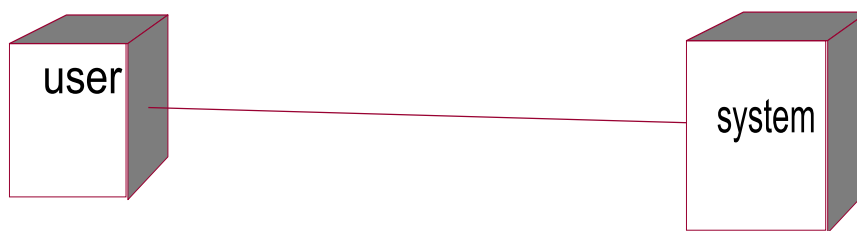


Fig: Deployment diagram

deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware).

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes.[1] To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

There are two types of Nodes:

- Device Node
- Execution Environment Node

Device nodes are physical computing resources with processing memory and services to execute software, such as typical computers or mobile phones. An execution environment node (EEN) is a software computing resource that runs within an outer node and which itself provides a service to host and execute other executable software elements.



#### 4.2.6ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

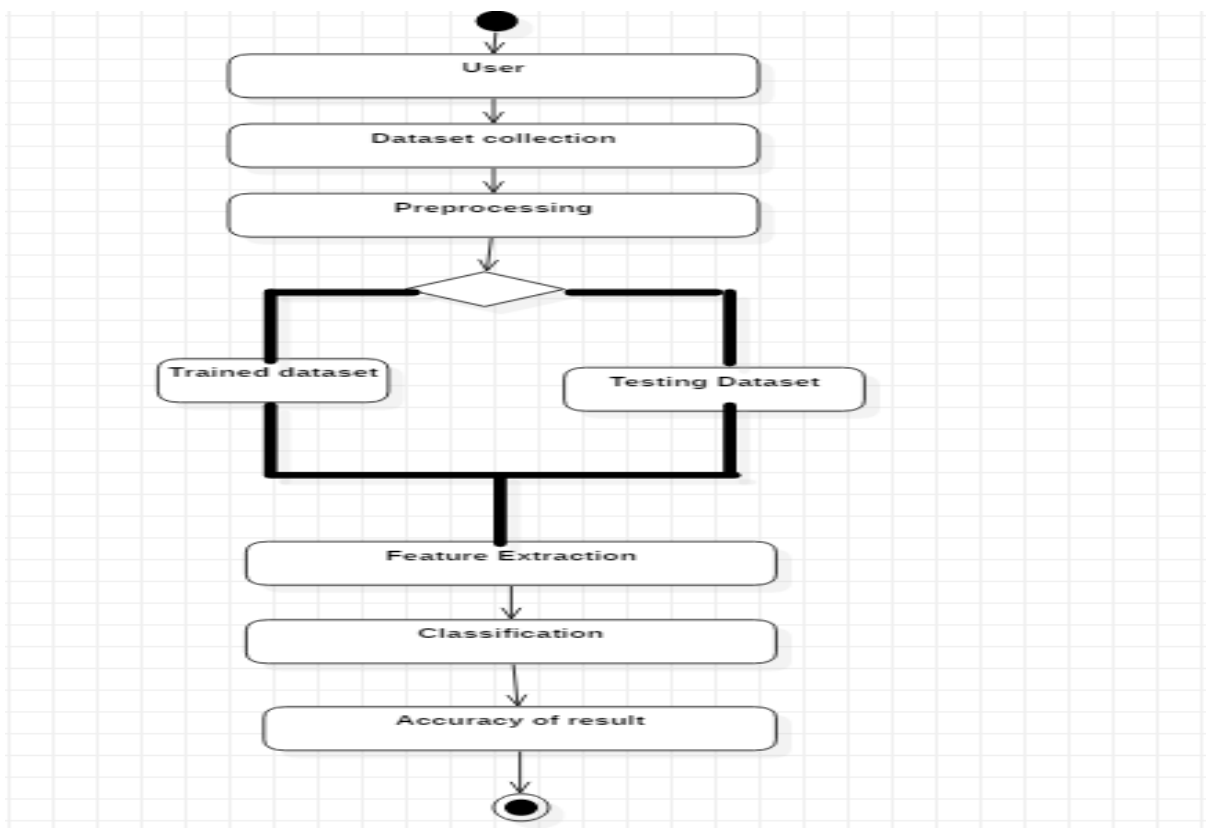


Fig: Activity diagram

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to represent business workflows

### 4.2.7 COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.



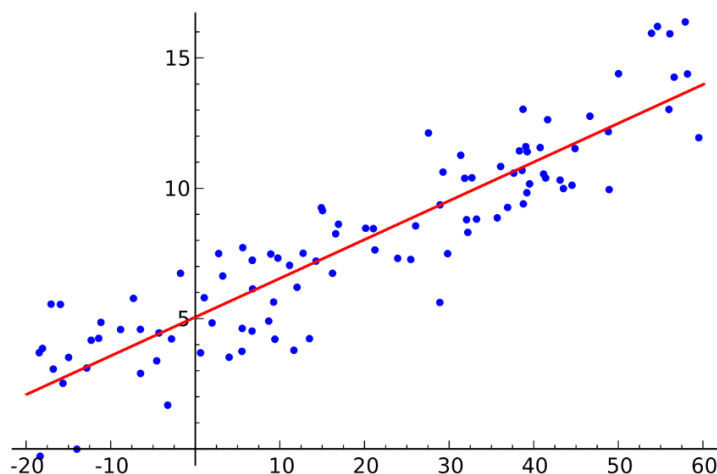
Fig: Component diagram

## 4.3ALGORITHMS

### 4.3.1LINEAR REGRESSION:

Regression is a method of modelling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables.

Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.



Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable. The red line in the above graph is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best. The line can be modelled based on the linear equation shown below.

The motive of the linear regression algorithm is to find the best values for  $a_0$  and  $a_1$ . Before moving on to the algorithm, let's have a look at two important concepts you must know to better understand linear regression.

The cost function helps us to figure out the best possible values for  $a_0$  and  $a_1$  which would provide the best fit line for the data points. Since we want the best values for  $a_0$  and  $a_1$ , we convert this search problem into a minimization problem where we would like to minimize the error between the predicted value and the actual value.

## DISADVANTAGES

### Cost Function:

- Linear Regression Only Looks at the Mean of the Dependent Variable. Linear regression looks at relationship between the mean of the dependent variable and the independent variables. ...
- Linear Regression Is Sensitive to Outliers. ...
- Data Must Be Independent.

### 4.3.2KNN (K-NEAREST NEIGHBOUR):

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.

#### STEPS:

- Input the dataset and split it into a training and testing set.
- Pick an instance from the testing sets and calculate its distance with the training set.
- List distances in ascending order.

- The class of the instance is the most common class of the 3 first trainings instances (k=3).

### 4.3.3 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.

If a set of training data is given to the machine, each data item will be assigned to one or the other categorical variables, a SVM training algorithm builds a model that plots new data item to one or the other category. In an SVM model, each data item is represented as points in an ndimensional space where n is the number of features where each feature is represented as the value of a particular coordinate in the n-dimensional space. Classification is carried out by finding a hyper-plane that divides the two classes proficiently. Later, new data item is mapped into the same space and its category is predicted based on the side of the hyper-plane they turn up.

An SVM classifier performs binary classification, i.e., it separates a set of training vectors for two different classes  $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ , where  $x_i \in R^d$  denotes vectors in a  $d$ -dimensional feature space and  $y_i \in \{-1, +1\}$  is a class label. The SVM model is generated by mapping the input vectors onto a new higher dimensional feature space denoted as  $\Phi: R^d \rightarrow H^f$  where  $d < f$ . Then, an optimal separating hyperplane in the new feature space is constructed by a kernel function  $K(x_i, x_j)$ , which is the product of input vectors  $x_i$  and  $x_j$  and where  $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ . support vectors is usually small compared to the size of the training set and they determine the margin of the hyperplane, and thus the decision surface.

# Chapter V

---

## SYSTEM IMPLEMENTAION

---

## 5.SYSTEM IMPLEMENTATION

The Implementation phase begins with leaf's sample being captured using regular digital camera with black background with the help of a stand. The image is loaded into matlab for processing. The features such as texture and color features are extracted fir identifying and classifying such as healthy or diseased are extracted for classifying the sample image.

### 5.1DESCRIPTION OF TECHNOLOGY USED:

#### PYTHON3

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

**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.

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

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

**Python is a Beginner's 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.

#### History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many otherlanguages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell, and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).Python is now maintained by a core

development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

## Python Features

Python's features include:

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

**Easy-to-read:** Python code is more clearly defined and visible to the eyes.

**Easy-to-maintain:** Python's source code is fairly easy-to-maintain.

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

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

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

**Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

**Databases:** Python provides interfaces to all major commercial databases.

**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.

**Scalable:** Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

1. IT supports functional and structured programming methods as well as OOP.
2. It can be used as a scripting language or can be compiled to byte-code for building large applications.
3. It provides very high-level dynamic data types and supports dynamic type checking.
4. IT supports automatic garbage collection.
5. It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

---

## Python built-in modules

1. Numpy
2. Pandas
3. Matplotlib
4. Sklearn
5. seaborn

### NUMPY

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the array object. This encapsulates dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). hanging the size of an array will create a new array and delete the original. • The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.

- NumPy arrays facilitate advanced mathematical and other types of operations on large numbers

of data. Typically, such operations are executed more efficiently and with less code than is possible using

Python's built-in sequences.

- A growing plethora of scientific and mathematical Python-based packages are

using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know how to use NumPy arrays. The points about sequence size and speed are



---

particularly important in scientific computing.

As a simple example, consider the case of multiplying each element in a 1-D sequence with the corresponding element in another sequence of the same length. If the data are stored in two Python lists, *a* and *b*, we could iterate over each element:

The Numeric Python extensions (NumPy henceforth) is a set of extensions to the Python programming language which allows Python programmers to efficiently manipulate large sets of objects organized in grid like fashion. These sets of objects are called arrays, and they can have any number of dimensions: one dimensional arrays are similar to standard Python sequences, two-dimensional arrays are similar to matrices from linear algebra. Note that one-dimensional arrays are also different from any other Python sequence, and that two-dimensional matrices are also different from the matrices of linear algebra, in ways which we will mention later in this text. Why are these extensions needed? The core reason is a very prosaic one, and that is that manipulating a set of a million numbers in Python with the standard data structures such as lists, tuples or classes is much too slow and uses too much space.

Anything which we can do in NumPy we can do in standard Python – we just may not be alive to see the program finish. A more subtle reason for these extensions however is that the kinds of operations that programmers typically want to do on arrays, while sometimes very complex, can often be decomposed into a set of fairly standard operations. This decomposition has been developed similarly in many array languages. In some ways, NumPy is simply the application of this experience to the Python language – thus many of the operations described in NumPy work the way they do because experience has shown that way to be a good one, in a variety of contexts. The languages which were used to guide the development of NumPy include the infamous APL family of languages, Basis, MATLAB, FORTRAN, S and S+, and others. This heritage will be obvious to users of NumPy who already have experience with these other languages. This tutorial, however, does not assume any such background, and all that is expected of the reader is a reasonable working knowledge of the standard Python language. This document is the “official” documentation for NumPy. It is both a tutorial and the most authoritative source of information about NumPy with the exception of the source code. The tutorial material will walk you through a set of manipulations of simple, small, arrays of numbers, as well as image files. This choice was made because:

- A concrete data set makes explaining the behavior of some functions much easier to motivate than simply talking about abstract operations on abstract data sets;

- Every reader will at least an intuition as to the meaning of the data and organization of image files
- The result of various manipulations can be displayed simply since the data set has a natural graphical representation. All users of NumPy, whether interested in image processing or not, are encouraged to follow the tutorial with a working NumPy installation at their side, testing the examples, and, more importantly, transferring the understanding gained by working on images to their specific domain. The best way to learn is by doing the aim of this tutorial is to guide you along this “doing.”

## PANDAS

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal.

pandas is well suited for many different kinds of data:

1. Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
2. Ordered and unordered (not necessarily fixed-frequency) time series data.
3. Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
4. Any other form of observational / statistical data sets. The data actually need not be labeled at all to be placed into a pandas data structure
5. The two primary data structures of pandas, Series (1-dimensional) and DataFrame (2-dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering. For R users, DataFrame provides everything that R's provides and much more. pandas is built on top of **NumPy** and is intended to integrate well within a scientific computing environment with many other 3rd party libraries.

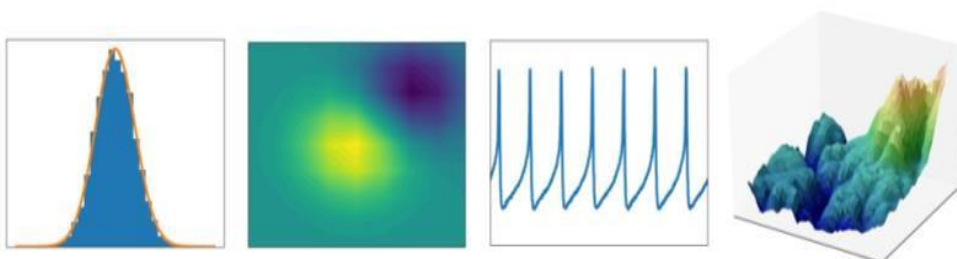
Here are just a few of the things that pandas does well:

1. Easy handling of **missing data** (represented as NaN) in floating point as well as non-floating point data
2. Size mutability: columns can be **inserted and deleted** from DataFrame and higher dimensional objects
3. Automatic and explicit **data alignment**: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series, DataFrame, etc. automatically align the data for you in computations
4. Powerful, flexible **group by** functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
5. Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structure into DataFrame objects
6. Intelligent label-based slicing, fancy indexing, and subsetting of large data sets
7. Intuitive merging and joining data sets
8. Flexible reshaping and pivoting of data sets
9. Hierarchical labeling of axes (possible to have multiple labels per tick)
10. Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving loading data from the ultrafast HDF5 format .
11. Time series-specific functionality: date range generation and frequency conversion, moving window statistics, date shifting and lagging.

## MATPLOTLIB

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

visualizations in Python.



Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits.

## SEABORN

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

1. A dataset-oriented API for examining relationships between multiple variables
2. Specialized support for using categorical variables to show observations or aggregate statistics
3. Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
4. Automatic estimation and plotting of linear regression models for different kinds dependent variables
5. Convenient views onto the overall structure of complex datasets
6. High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
7. Concise control over matplotlib figure styling with several built-in themes
8. Tools for choosing color palettes that faithfully reveal patterns in your data
9. Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on dataframes and arrays containing whole
10. datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

## 5.2SOURCE CODE:

Importing Libraries:

```
import warnings
warnings.filterwarnings('ignore')
import os
import numpy as np
import pandas as pd
```

---

```
import seaborn as sns
import datetime as dt
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
data = pd.read_csv('data.csv')
data
data.drop('id',axis=1,inplace=True)
data.drop('Unnamed: 32',axis=1,inplace=True)
Binarizing the target variable:
data['diagnosis'] = data['diagnosis'].map({'M':1,'B':0})
datas = pd.DataFrame(preprocessing.scale(data.iloc[:,1:32]))
datas.columns = list(data.iloc[:,1:32].columns)
datas['diagnosis'] = data['diagnosis']
#Looking at the number of patients with Malignant and Benign Tumors:
datas.diagnosis.value_counts().plot(kind='bar', alpha = 0.5, facecolor = 'b', figsize=(12,6))
plt.title("Diagnosis (M=1 , B=0)", fontsize = '18')
plt.ylabel("Total Number of Patients")
plt.grid(b=True)#Looking at the number of patients with Malignant and Benign Tumors:
datas.diagnosis.value_counts().plot(kind='bar', alpha = 0.5, facecolor = 'b', figsize=(12,6))
plt.title("Diagnosis (M=1 , B=0)", fontsize = '18')
plt.ylabel("Total Number of Patients")
plt.grid(b=True)
data_mean
=data[['diagnosis','radius_mean','texture_mean','perimeter_mean','area_mean','smoothness_mean',
'compactness_mean',      'concavity_mean','concave      points_mean',      'symmetry_mean',
'fractal_dimension_mean']]
plt.figure(figsize=(14,14))
foo = sns.heatmap(data_mean.corr(), vmax=1, square=True, annot=True)
_ = sns.swarmplot(y='perimeter_mean',x='diagnosis', data=data_mean)
plt.show()
from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
```

---

```
from sklearn import metrics
predictors = data_mean.columns[2:11]
target = "diagnosis"
X = data_mean.loc[:,predictors]
y = np.ravel(data.loc[:,[target]])

# Split the dataset in train and test:
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
print ('Shape of training set : %i || Shape of test set : %i' % (X_train.shape[0],X_test.shape[0]) )
print ('The dataset is very small so simple cross-validation approach should work here')
print ('There are very few data points so 10-fold cross validation should give us a better estimate')
```

```
from sklearn.linear_model import LogisticRegression
# Initiating the model:
lr = LogisticRegression()
scores = cross_val_score(lr, X_train, y_train, scoring='accuracy',cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))
```

```
from sklearn import svm
# Initiating the model:
svm = svm.SVC()
scores = cross_val_score(svm, X_train, y_train, scoring='accuracy',cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))
```

```
from sklearn.neighbors import KNeighborsClassifier
# Initiating the model:
knn = KNeighborsClassifier()

scores = cross_val_score(knn, X_train, y_train, scoring='accuracy',cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))
```

```
from sklearn.linear_model import Perceptron
# Initiating the model:
```

---

---

```
pct = Perceptron()
scores = cross_val_score(pct, X_train, y_train, scoring='accuracy', cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))
```

```
from sklearn.ensemble import RandomForestClassifier
# Initiating the model:
rf = RandomForestClassifier()
scores = cross_val_score(rf, X_train, y_train, scoring='accuracy', cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))
```

```
from sklearn.naive_bayes import GaussianNB
# Initiating the model:
nb = GaussianNB()
scores = cross_val_score(rf, X_train, y_train, scoring='accuracy', cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))
for i in range(1, 21):
    knn = KNeighborsClassifier(n_neighbors = i)
    score = cross_val_score(knn, X_train, y_train, scoring='accuracy', cv=10).mean()
    print("N = " + str(i) + " :: Score = " + str(round(score,2)))
N = 1 :: Score = 0.87
N = 2 :: Score = 0.87
N = 3 :: Score = 0.88
N = 4 :: Score = 0.89
N = 5 :: Score = 0.88
N = 6 :: Score = 0.89
N = 7 :: Score = 0.89
N = 8 :: Score = 0.88
N = 9 :: Score = 0.87
N = 10 :: Score = 0.88
N = 11 :: Score = 0.87
N = 12 :: Score = 0.88
N = 13 :: Score = 0.88
N = 14 :: Score = 0.87
N = 15 :: Score = 0.88
```

---

---

N = 16 :: Score = 0.88

N = 17 :: Score = 0.88

N = 18 :: Score = 0.88

N = 19 :: Score = 0.88

N = 20 :: Score = 0.88

for i in range(1, 21):

    rf = RandomForestClassifier(n\_estimators = i)

    score = cross\_val\_score(rf, X\_train, y\_train, scoring='accuracy', cv=10).mean()

    print("N = " + str(i) + " :: Score = " + str(round(score,2)))

N = 1 :: Score = 0.91

N = 2 :: Score = 0.91

N = 3 :: Score = 0.92

N = 4 :: Score = 0.92

N = 5 :: Score = 0.93

N = 6 :: Score = 0.91

N = 7 :: Score = 0.93

N = 8 :: Score = 0.95

N = 9 :: Score = 0.93

N = 10 :: Score = 0.94

N = 11 :: Score = 0.94

N = 12 :: Score = 0.94

N = 13 :: Score = 0.93

N = 14 :: Score = 0.93

N = 15 :: Score = 0.93

N = 16 :: Score = 0.93

N = 17 :: Score = 0.94

N = 18 :: Score = 0.93

N = 19 :: Score = 0.93

N = 20 :: Score = 0.94

from sklearn.ensemble import RandomForestClassifier

# Initiating the model:

rf = RandomForestClassifier(n\_estimators=18)



```
rf = rf.fit(X_train, y_train)
predicted = rf.predict(X_test)
acc_test = metrics.accuracy_score(y_test, predicted)
print ('The accuracy on test data is %s' % (round(acc_test,2)))
```

```
from sklearn.naive_bayes import GaussianNB
# Initiating the model:
nb = GaussianNB()
nb = nb.fit(X_train, y_train)
predicted = nb.predict(X_test)
acc_test = metrics.accuracy_score(y_test, predicted)
print ('The accuracy on test data is %s' % (acc_test))
```

# Chapter VI

---

## SYSTEM TESTING

---

## 6. 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, sub assemblies, 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 Each test type addresses a specific testing requirement.

### 6.1 TESTING ACTIVITIES

#### 6.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

#### Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

#### Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

### 6.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### 6.1.3 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined

### 6.1.4System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### 6.1.5 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## 6.2 TYPES OF TESTING

### 6.2.1 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Simpler models such as linear regression and decision trees on the other hand provide less predictive capacity and are not always capable of modelling the inherent complexity of the dataset (i.e. feature interactions). They are however significantly easier to explain and interpret.

White-box models are the type of models which one can clearly explain how they behave, how they produce predictions .

### 6.2.2 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Black-box models such as neural networks, gradient boosting models or complicated ensembles often provide great accuracy. The inner workings of these models are harder to understand and they don't provide an estimate of the importance of each feature on the model predictions. black-box models, users can only observe the input-output relationship. For example, input the customer profile then output customer churn propensity score. But the underlying reasons or processes to produce the output are not available Black-box models often result in 1pc to 3pc better accuracy than white-box models, but you sacrifice transparency and accountability.

## Chapter VII

---

# CONCLUSION & FUTURE ENHANCEMENT

---

## **7.CONCLUSION & FUTURE ENHANCEMENTS:**

### **7.1. Conclusion :**

Breast cancer if found at an early stage will help save lives of thousands of women or even men. These projects help the real world patients and doctors to gather as much information as they can. The research on nine papers has helped us gather the data for the project proposed by us. By using machine learning algorithms we will be able to classify and predict the cancer into being or malignant. Machine learning algorithms can be used for medical oriented research, it advances the system, reduces human errors and lowers manual mistakes.

### **7.2. Future Enhancement:**

The analysis of the results signifies that the integration of multidimensional data along with different classification, feature selection and dimensionality reduction techniques can provide auspicious tools for inference in this domain. Further research in this field should be carried out for the better performance of the classification techniques so that it can predict on more variables. We are intending how to parametrize our classification techniques hence to achieve high accuracy. We are looking into many datasets and how further Machine Learning algorithms can be used to characterize Breast Cancer. We want to reduce the error rates with maximum accuracy.

## Chapter VIII

---

# REFERENCES

---



---

## 8.REFERENCES

- [1] Shomona G. Jacob, R. Geetha Ramani, "Efficient Classifier for Classification of Prognosis Breast cancer Data through Data Mining Techniques", *Proceedings of the World Congress on Engineering and Computer Science 2012*, vol. I, October 2012. Wang, D. Zhang and Y. H. Huang "Breast Cancer Prediction Using Machine Learning" (2018), Vol. 66, NO. 7.
- [2] B. Akbugday, "Classification of Breast Cancer Data Using Machine Learning Algorithms," 2019 Medical Technologies Congress (TIPTEKNO), Izmir, Turkey, 2019, pp. 1-4.
- [3] Keles, M. Kaya, "Breast Cancer Prediction and Detection Using Data Mining Classification Algorithms: A Comparative Study." *Tehnicki Vjesnik - Technical Gazette*, vol. 26, no. 1, 2019, p. 149+.
- [4] V. Chaurasia and S. Pal, "Data Mining Techniques: To Predict and Resolve Breast Cancer Survivability", *IJCSMC*, Vol. 3, Issue. 1, January 2014, pg.10 – 22.
- [5] Delen, D.; Walker, G.; Kadam, A. Predicting breast cancer survivability: A comparison of three data mining methods. *Artif. Intell. Med.* 2005, 34, 113–127.
- [6] R. K. Kavitha<sup>1</sup>, D. D. Rangasamy, "Breast Cancer Survivability Using Adaptive Voting Ensemble Machine Learning Algorithm Adaboost and CART Algorithm" Volume 3, Special Issue 1, February 2014.
- [7] P. Sinthia, R. Devi, S. Gayathri and R. Sivasankari, "Breast Cancer detection using PCPCET and ADEWNN", *CIEEE' 17*, p.63-65.
- [8] Vikas Chaurasia and S.Pal, "Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis" (*FAMS 2016*) 83 ( 2016 ) 1064 – 1069.
- [9] N. Khuriwal, N. Mishra. "A Review on Breast Cancer Diagnosis in Mammography Images Using Deep Learning Techniques", (2018), Vol. 1, No. 1.
- [10] Y. Khourdifi and M. Bahaj, "Feature Selection with Fast Correlation-Based Filter for Breast Cancer Prediction and Classification Using Machine Learning Algorithms," 2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT), Rabat, Morocco, 2018, pp. 1-6.
- [11] R. M. Mohana, R. Delshi Howsalya Devi, Anita Bai, "Lung Cancer Detection using Nearest Neighbour Classifier", *International Journal of Recent Technology and Engineering (IJRTE)*, Volume-8, Issue-2S11, September 2019.
- [12] Ch. Shravya, K. Pravalika, Shaik Subhani, "Prediction of Breast Cancer Using

Supervised Machine Learning Techniques”, International Journal of Innovative Technology and Exploring Engineering (IJITEE)Volume-8 Issue-6, April 2019.

[13] Haifeng Wang and Sang Won Yoon, “Breast Cancer Prediction Using Data Mining Method”, Proceedings of the 2015 Industrial and Systems Engineering Research Conference,

[14] Abdelghani Bellaachia, Erhan Guven, “Predicting Breast Cancer Survivability Using Data Mining Techniques”.

[15] D. Delen, G. Walker, A. Kadam, "Predicting breast cancer survivability: a comparison three data mining methods", *Artificial Intelligence in Medicine*, vol. 34, no. 2,pp.113127, 2004.

[16] <https://www.ijitee.org/wpcontent/uploads/papers/v8i6/F3384048619.pdf>

**APPENDIX**

---

## APPENDIX

### **Installation:**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

### **Why use Navigator?**

In order to run, many scientific packages depend on specific versions of other many packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

### **What Applications Can I Access Using Navigator**

The following applications are available by default in Navigator:

- 1.JupyterLab
2. Jupyter Notebook
3. QT Console
- 4.Spyder
- 5.VSCode
- 6.Glueviz
- 7.Orange 3 App
- 8.Rodeo
- 9.RStudi

## How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

Step1: Open the following URL in your browser -<https://colab.research.google.com>  
Your browser would display the following screen (assuming that you are logged into your Google Drive) –

Step 2: Click on the NEW PYTHON 3 NOTEBOOK link at the bottom of the screen. A new notebook would open up as shown in the screen below.

### Step 3: **Entering Code**

You will now enter a trivial Python code in the code window and execute it.

The screenshot shows a Jupyter Notebook titled "Breast cancer (2)" with a last checkpoint from yesterday at 17:35. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook contains two code blocks:

```
In [1]: # Importing Libraries:
import warnings
warnings.filterwarnings('ignore')
import os
import numpy as np
import pandas as pd
import seaborn as sns
import datetime as dt
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
```

```
In [2]: data = pd.read_csv('data.csv')
data
```

The output of the second code block is a DataFrame with the following columns: id, diagnosis, radius\_mean, texture\_mean, perimeter\_mean, area\_mean, smoothness\_mean, compactness\_mean, concavity\_mean, and concave points\_mean. The output shows a preview of the data with rows 0 through 567.

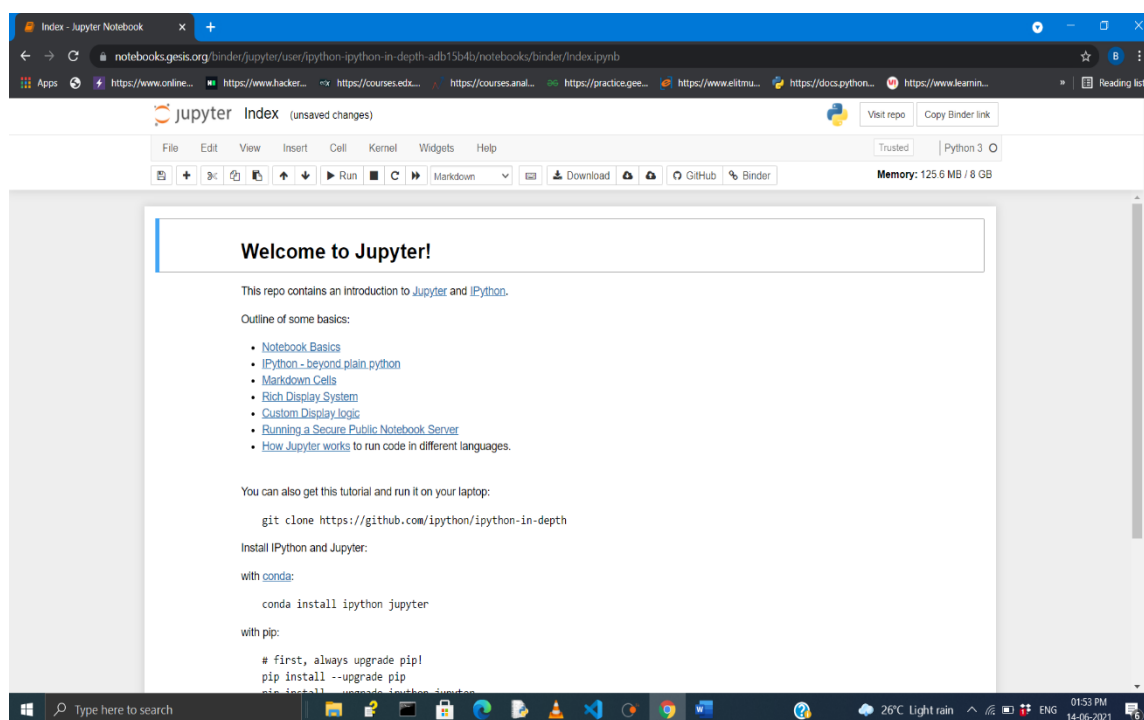
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430
...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302
567	927241	M	20.60	28.33	140.10	1265.0	0.11780	0.27700	0.35140	0.14500

After a while, you will see the output underneath the code block by block you will get the respected result.

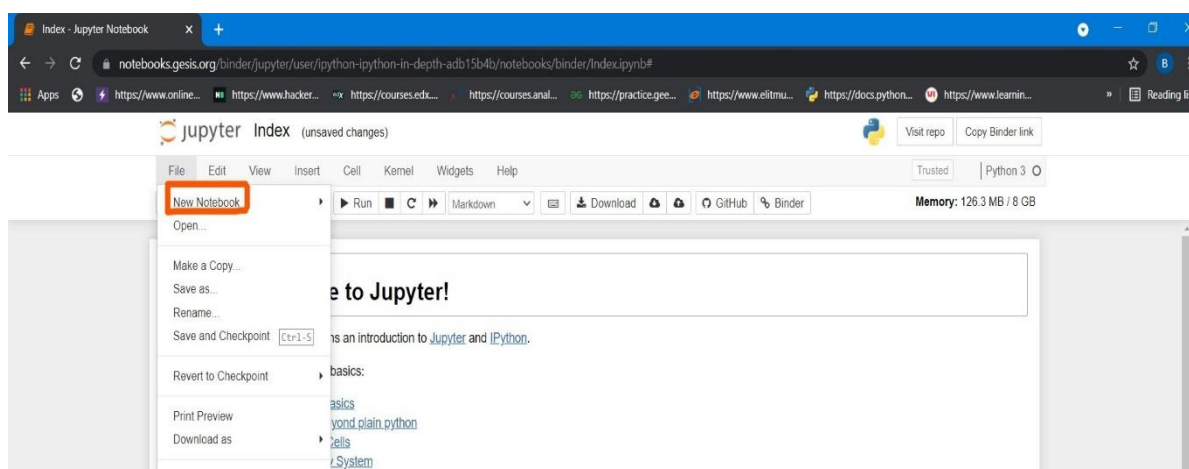
Like following above steps and keep on executing code block by block you will get the respected result.

## EXECUTION STEPS:

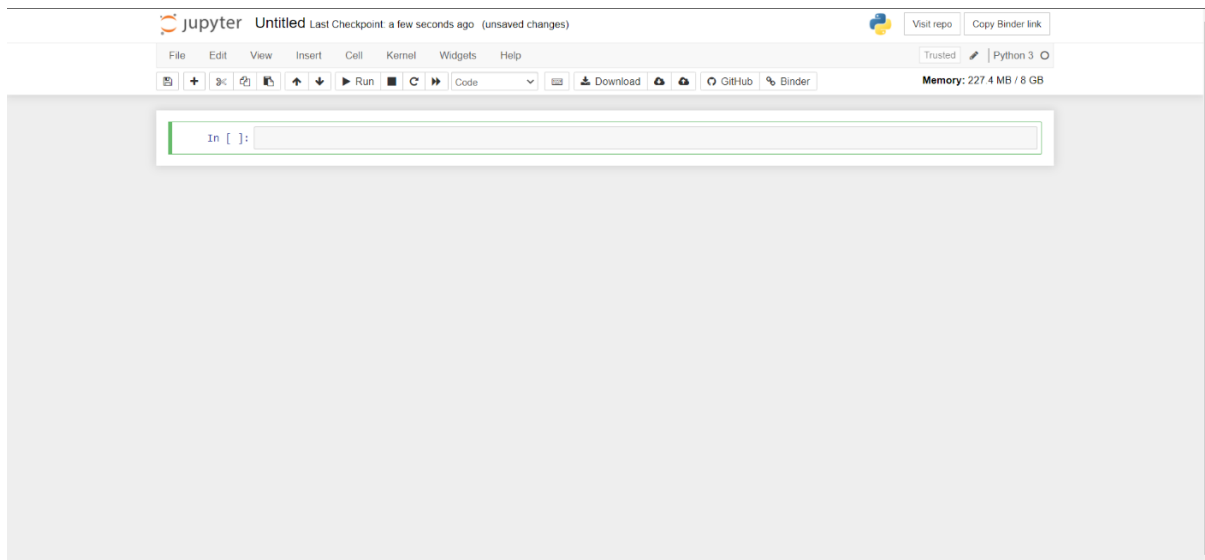
Step1: Open the following URL in your browser –<https://notebooks.gesis.org/binder/jupyter/> Your browser would display the following screen.



Step 2: Click on the **NEW PYTHON 3 NOTEBOOK** link at the bottom of the screen. A new notebook would open up as shown in the screen below.

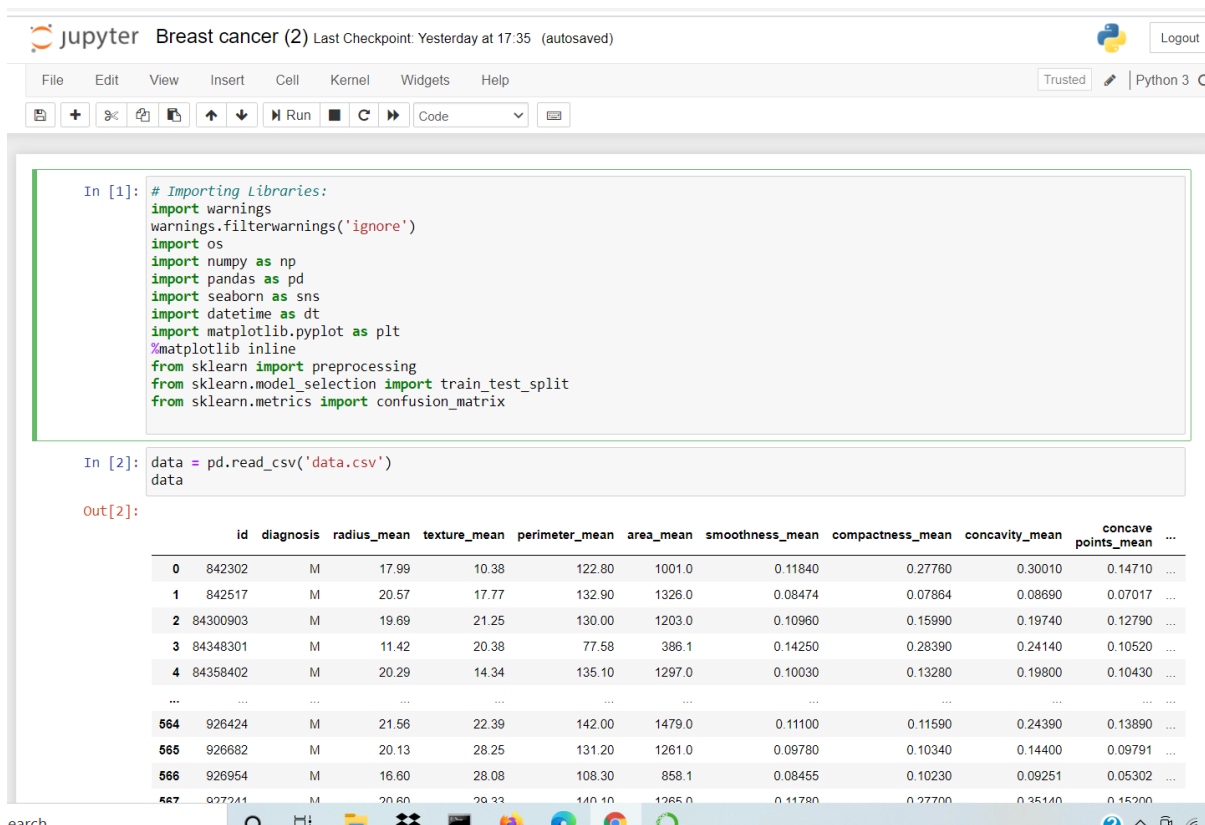


Step3: After You click on New Notebook, The following window will open



#### Step 4: Entering Code

After open python shell, you need to upload your code file, data file By using **open** option in file, You will now enter a trivial Python code in the code window and execute it.



After a while, you will see the output underneath the code window.



The first screenshot shows the Jupyter Notebook interface with the following code and output:

```

%matplotlib inline
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix

In [2]: data = pd.read_csv('data.csv')
data
Out[2]:

```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28380	0.24140	0.10520
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430
...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000

569 rows x 33 columns

```

In [3]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):

```

The second screenshot shows the output of the `data.info()` command:

```

#      column      Non-Null Count  Dtype
---  -
0      id          569 non-null      int64
1      diagnosis   569 non-null      object
2      radius_mean 569 non-null      float64
3      texture_mean 569 non-null      float64
4      perimeter_mean 569 non-null     float64
5      area_mean    569 non-null      float64
6      smoothness_mean 569 non-null     float64
7      compactness_mean 569 non-null     float64
8      concavity_mean 569 non-null     float64
9      concave_points_mean 569 non-null     float64
10     symmetry_mean 569 non-null     float64
11     fractal_dimension_mean 569 non-null     float64
12     radius_se     569 non-null     float64
13     texture_se     569 non-null     float64
14     perimeter_se   569 non-null     float64
15     area_se        569 non-null     float64
16     smoothness_se  569 non-null     float64
17     compactness_se 569 non-null     float64
18     concavity_se   569 non-null     float64
19     concave_points_se 569 non-null     float64
20     symmetry_se     569 non-null     float64
21     fractal_dimension_se 569 non-null     float64
22     radius_worst   569 non-null     float64
23     texture_worst  569 non-null     float64
24     perimeter_worst 569 non-null     float64
25     area_worst     569 non-null     float64
26     smoothness_worst 569 non-null     float64
27     compactness_worst 569 non-null     float64
28     concavity_worst 569 non-null     float64
29     concave_points_worst 569 non-null     float64
30     symmetry_worst  569 non-null     float64

```

**Figure 1: Jupyter Notebook - Breast cancer (1)**

```

23 texture_worst      569 non-null float64
24 perimeter_worst    569 non-null float64
25 area_worst         569 non-null float64
26 smoothness_worst   569 non-null float64
27 compactness_worst  569 non-null float64
28 concavity_worst     569 non-null float64
29 concave_points_worst 569 non-null float64
30 symmetry_worst     569 non-null float64
31 fractal_dimension_worst 569 non-null float64
32 Unnamed: 32        0 non-null float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB

```

In [4]: data.head(2)

```

Out[4]:
   id  diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  smoothness_mean  compactness_mean  concavity_mean  concave points_mean  ...  textu
0  842302      M         17.99         10.38           122.8       1001.0         0.11840         0.27760         0.3001         0.14710  ...
1  842517      M         20.57         17.77           132.9       1326.0         0.08474         0.07864         0.0869         0.07017  ...
2 rows x 33 columns

```

In [ ]: data.diagnosis.unique()

In [ ]: data.describe()

In [ ]: data.drop('id',axis=1,inplace=True)  
data.drop('Unnamed: 32',axis=1,inplace=True)

In [ ]: # Binarizing the target variable:  
data['diagnosis'] = data['diagnosis'].map({'M':1,'B':0})

---

**Figure 2: Jupyter Notebook - Breast cancer (2)**

In [4]: data.head(2)

```

Out[4]:
   id  diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  smoothness_mean  compactness_mean  concavity_mean  concave points_mean  ...  textu
0  842302      M         17.99         10.38           122.8       1001.0         0.11840         0.27760         0.3001         0.14710  ...
1  842517      M         20.57         17.77           132.9       1326.0         0.08474         0.07864         0.0869         0.07017  ...
2 rows x 33 columns

```

In [5]: data.diagnosis.unique()

```

Out[5]: array(['M', 'B'], dtype=object)

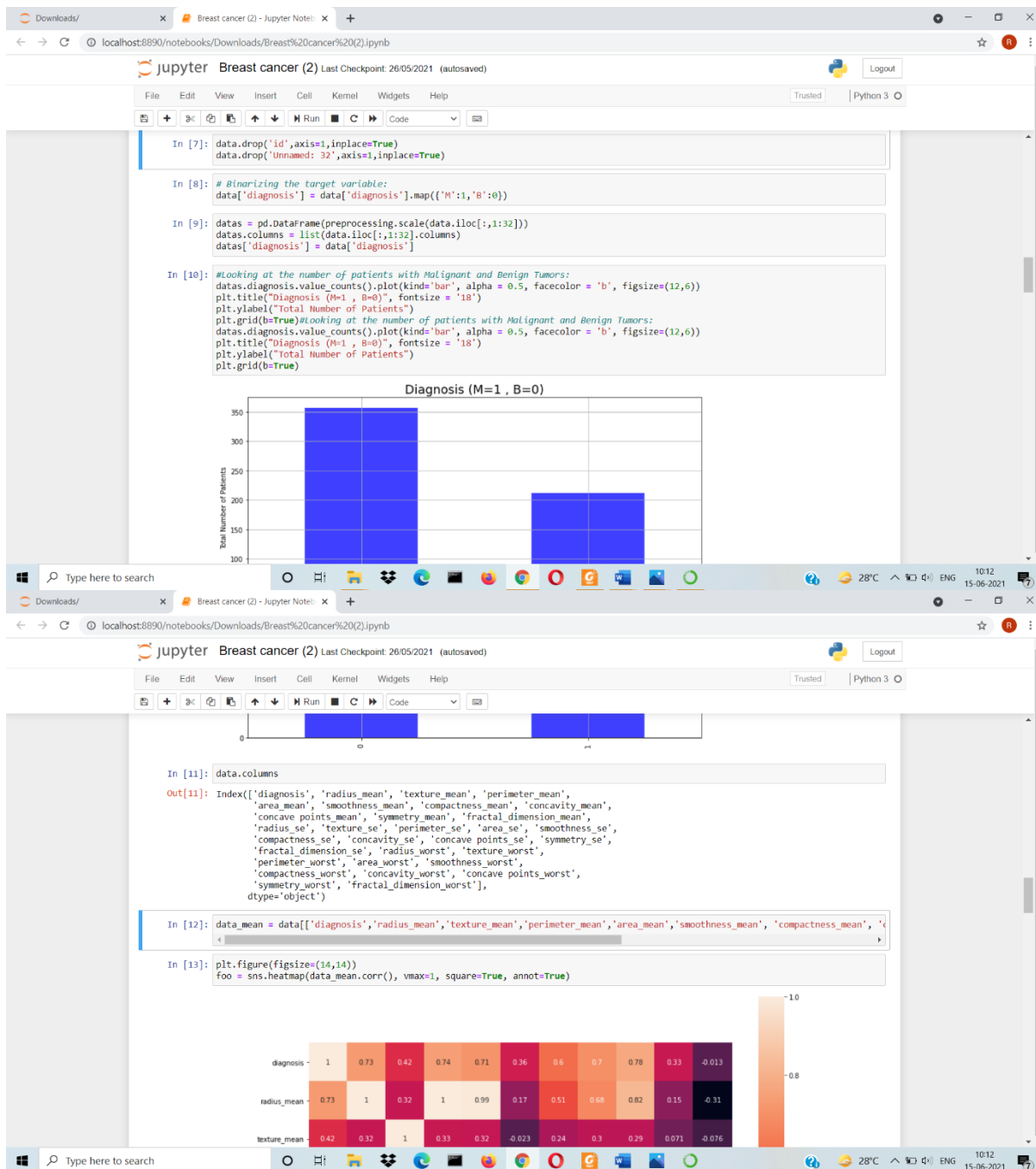
```

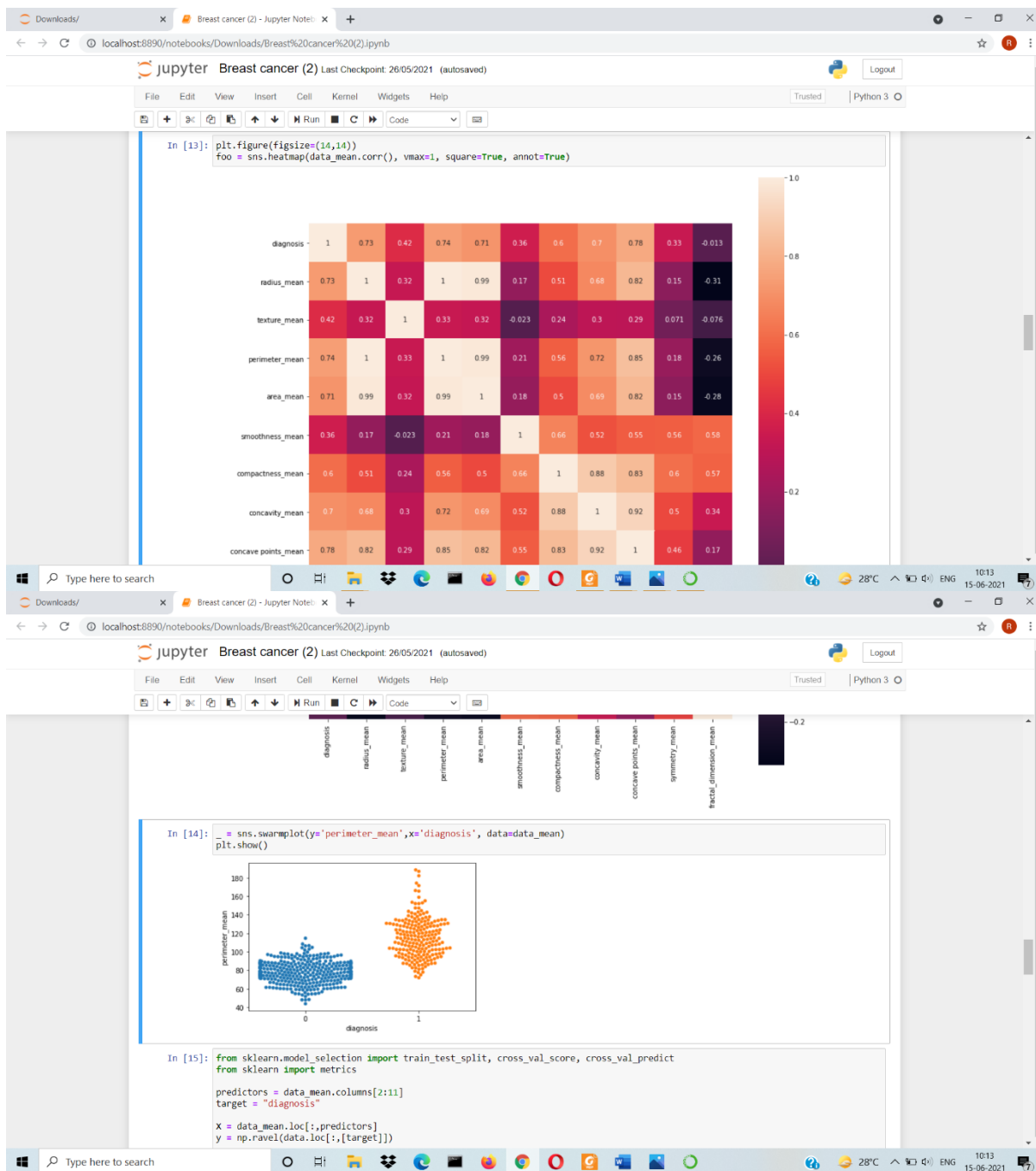
In [6]: data.describe()

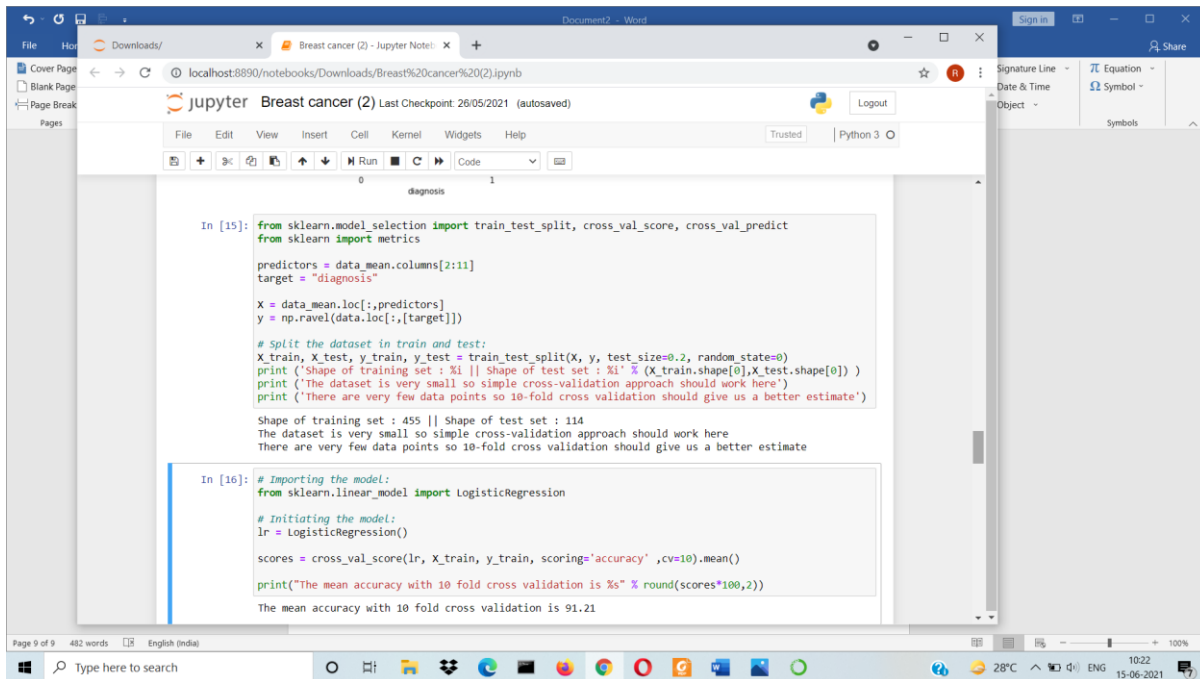
```

Out[6]:
   id  radius_mean  texture_mean  perimeter_mean  area_mean  smoothness_mean  compactness_mean  concavity_mean  concave points_mean  symmetr
count  5.690000e+02  569.000000  569.000000  569.000000  569.000000  569.000000  569.000000  569.000000  569.000000  569.000000
mean   3.037183e+07  14.127292  19.289649  91.969033  654.889104  0.096360  0.104341  0.088799  0.048919  0.048919
std    1.250206e+08  3.524049  4.301036  24.298981  351.914129  0.014064  0.052813  0.079720  0.038803  0.038803
min    8.670000e+03  6.981000  9.710000  43.790000  143.500000  0.052630  0.019380  0.000000  0.000000  0.000000
25%   8.692180e+05  11.700000  16.170000  75.170000  420.300000  0.086370  0.064920  0.029560  0.020310  0.020310
50%   9.060240e+05  13.370000  18.840000  86.240000  551.100000  0.095870  0.092630  0.061540  0.033500  0.033500
75%   8.813120e+06  15.780000  21.800000  104.100000  782.700000  0.105300  0.130400  0.130700  0.074000  0.074000
max   9.113205e+08  28.110000  39.280000  188.500000  2501.000000  0.163400  0.345400  0.426800  0.201200  0.201200
8 rows x 32 columns

```







The screenshot shows a Jupyter Notebook titled "Breast cancer (2)" with the following code in cell [15]:

```
from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
from sklearn import metrics

predictors = data_mean.columns[2:11]
target = "diagnosis"

X = data_mean.loc[:,predictors]
y = np.ravel(data.loc[:,target])

# Split the dataset in train and test:
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
print('Shape of training set : %i || Shape of test set : %i' % (X_train.shape[0], X_test.shape[0]))
print('The dataset is very small so simple cross-validation approach should work here')
print('There are very few data points so 10-fold cross validation should give us a better estimate')

Shape of training set : 455 || Shape of test set : 114
The dataset is very small so simple cross-validation approach should work here
There are very few data points so 10-fold cross validation should give us a better estimate

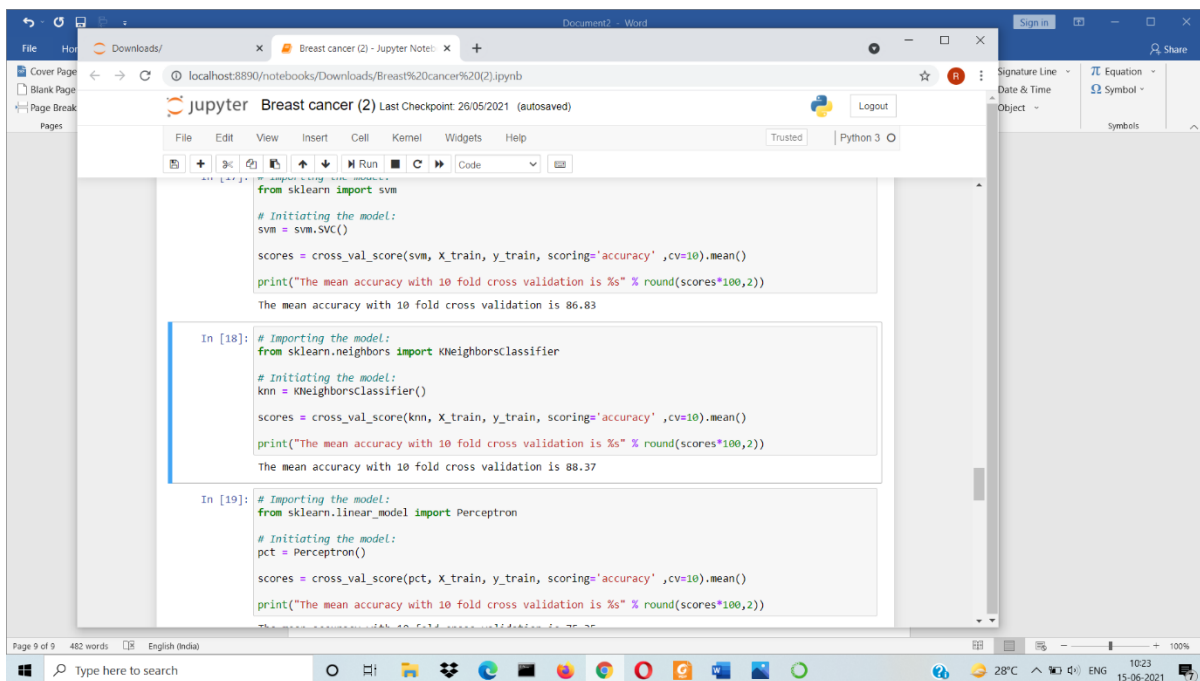
In [16]: # Importing the model:
from sklearn.linear_model import LogisticRegression

# Initiating the model:
lr = LogisticRegression()

scores = cross_val_score(lr, X_train, y_train, scoring='accuracy', cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))

The mean accuracy with 10 fold cross validation is 91.21
```

The notebook interface shows the file explorer on the left, the code editor in the center, and the output area on the right. The status bar at the bottom indicates the page number (9 of 9), word count (482 words), and language (English (India)).



The screenshot shows a Jupyter Notebook titled "Breast cancer (2)" with the following code in cells [17], [18], and [19]:

```
from sklearn import svm

# Initiating the model:
svm = svm.SVC()

scores = cross_val_score(svm, X_train, y_train, scoring='accuracy', cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))

The mean accuracy with 10 fold cross validation is 86.83

In [18]: # Importing the model:
from sklearn.neighbors import KNeighborsClassifier

# Initiating the model:
knn = KNeighborsClassifier()

scores = cross_val_score(knn, X_train, y_train, scoring='accuracy', cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))

The mean accuracy with 10 fold cross validation is 88.37

In [19]: # Importing the model:
from sklearn.linear_model import Perceptron

# Initiating the model:
pct = Perceptron()

scores = cross_val_score(pct, X_train, y_train, scoring='accuracy', cv=10).mean()
print("The mean accuracy with 10 fold cross validation is %s" % round(scores*100,2))

The mean accuracy with 10 fold cross validation is 75.35
```

The notebook interface shows the file explorer on the left, the code editor in the center, and the output area on the right. The status bar at the bottom indicates the page number (9 of 9), word count (482 words), and language (English (India)).