CMR UNIVERSITY

Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

Capstone Project Report

On

“CARCINO”

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

Bachelor of Technology

in

Information Technology

of CMR University, Bangalore

Submitted by:

ANKITHA CHOWDARY – 18BBTCS011

Under the Guidance of:

Prof. Mouna M Naravani

Assistant Professor

Dept. of CSE, SOET

Department of Information Technology

Off Hennur - Bagalur Main Road,

Near Kempegowda International Airport, Chagalahatti,

Bangalore, Karnataka-562149

2021-2022

SCHOOL OF ENGINEERING AND TECHNOLOGY

Department of Information Technology

CERTIFICATE

Certified that the ML project work, entitled **“CARCINO”**, submitted to the CMR University, Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology is a record of work done by **Ms. ANKITHA CHOWDARY** bearing university register number **18BBTCS011** respectively during the academic year 2021- 22 at School of Engineering and Technology, CMR University, Bangalore under my supervision and guidance. The Contents of this project work, in full or in parts, have not been submitted to any other Institute or University for the award of a degree or diploma.

Signature of the Guide	Signature of the HOD	Signature of the Dean
……………………	……………………	……………………
Dept. of CSE, SoET, CMRU, Bangalore	Dept. of CSE, SoET, CMRU, Bangalore	SoET, CMRU

<u>External Viva</u>

Name of the Examiners:	**Signature with Date:**
1. …………………….	…………………….
2. ……………………	…………………….

DECLARATION

I, **Ankitha Chowdary (18BBTCS011)** student of 7th semester B.Tech. Computer Science and Technology, School of Engineering and Technology, Bangalore, hereby declare that the ML Project work entitle **“Carcino”** has been carried out by me under the guidance of **Prof. Mouna M Naravani,** Assistant Professor, Department of Computer Science and Engineering, School of Engineering and Technology.

I further declare that the work reported in this capstone work has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this University
or any other institute or University.

Place:
Bangalore

Date:

ANKITHA CHOWDARY (18BBTCS011)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain.

I consider myself privileged to express gratitude and respect towards all those who guided me through the completion of the project. I express my heartfelt sincere gratitude to **Dr. C Prabhakar Reddy**, **Dean,** School of Engineering and Technology, CMR University for his support. I would like to express my thanks to **Dr. Rubini P, Professor and Head, Department of Computer Science and Engineering & Information Technology,** School of Engineering and Technology, CMR University, Bangalore, for her encouragement that motivated me for the successful completion of Project work. I express my thanks to the Project Guide **Prof. Mouna M Naravani, Assistant Professor,** Department of Computer Science and Engineering, School of Engineering and Technology, CMR University for his constant support.

I would like to thank all the professors and staff of Computer Science and Engineering Department for their co-operation and timely guidance.

ANKITHA CHOWDARY

ABSTRACT

Breast cancer is the uncontrolled growth of breast cells. It represents the second cause of cancer death in women worldwide. It is important for patients to understand their disease and know what to expecting the future so that they can make decisions about treatment, rehabilitation, financial aid decisions and personal matters.

Our project presents an approach for diagnosing breast cancer based on a set of input variables that describe some characteristics of tumor. The proposed approach builds a binary logistic model with an addition of few other algorithms namely Random Forest and Support vector machine that classifies between malignant and benign cases. The approach is applied to the Wisconsin Diagnostic Breast Cancer (WDBC) dataset. Experimental results show that the regression model that is statistically significant includes only the area, texture, concavity and symmetry features of a tumor. In addition to the simplicity of the used model, the reduced set of features gives performance measures that outperform similar approaches.

Our project gives the idea of predicting the breast cancer cells either they are cancerous or non-cancerous (Malignant or Benign) respectively based on the data collected. Also, the presented approach can be used for feature selection and reduction of the breast cancer data.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background

Breast cancer classification divides breast cancer into categories according to different schemes criteria and serving a different purpose. The major categories are the histopathological type, the grade of the tumor, the stage of the tumor, and the expression of proteins and genes as knowledge of cancer cell biology develops these classifications are updated. The purpose of classification is to select the best treatment. The effectiveness of a specific treatment is demonstrated for a specific breast cancer (usually by randomized, controlled trials). That treatment may not be effective in a different breast cancer. Some breast cancers are aggressive and life-threatening, and must be treated with aggressive treatments that have major adverse effects. Other breast cancers are less aggressive and can be treated with less. Treatment algorithms rely on breast cancer classification to define specific subgroups that are each treated according to the best evidence available. Classification aspects must be carefully tested and validated, such that confounding effects are minimized, making them either true prognostic factors, which estimate disease outcomes such as disease-free or overall survival in the absence of therapy, or true predictive factors, which estimate the likelihood of response or lack of response to a treatment. Classification of breast cancer is usually, but not always, primarily based on the histological appearance of tissue in the tumor. A variant from this approach, defined on the basis of physical exam findings, is that inflammatory breast cancer (IBC), a form of ductal carcinoma or malignant cancer in the ducts, is distinguished from other carcinomas by the inflamed appearance of the affected breast, which correlates with increased cancer aggressivity.

1.2 Problem Statement

Breast Cancer is one of the leading cancers developed in many countries including India. Through the endurance rate is high – with early diagnosis 97% women can survive for more than 5 years. Statistically, the death toll due to this disease has increased drastically in last few decades. The main issue pertaining to its cure is early recognition. This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection. The goal is to classify whether the breast cancer is benign or malignant.

1.3 Objective

The main objective is to access the Correctness in classifying data with respect to efficiency and effectiveness in terms of accuracy, precision, sensitivity and specificity. The objective of this is used to compare and identify an accurate model to predict the incidence of breast cancer based on various patients clinical records.

Chapter 2

LITERATURE SURVEY

[1] "Predicting Breast Cancer using Logistic Regression and Multiclass Classifiers", J. sultan November 2018.

This phase provides review of the present-day studies being accomplished on breast cancer and the usage of various data mining techniques to predict and diagnose the breast cancer. Presently, most of the physicians opt to make surgical biopsy in order to figure out different kinds of cancers for benign breast tumors from malignant. As biopsy could be very crucial challenge maximum of them believed that it should be stopped as much as possible. Thus, to recognize the kind of cancer and keep away from needless surgical biopsy, a smart system was presented which can be beneficial for both patients and physicians. To predict breast cancer comparative analysis was done by using neural network, decision tree, genetic algorithm and logistic regression by Wei-pin Chang et al. Their experimental outcomes found out that, amongst those applied techniques for predicting breast cancers lowest prediction accuracy was obtained by decision tree model and better accuracy rate was obtained by logistic regression model. In addition to this, genetic algorithm achieved maximum accuracy by generating standard classification rules inside the class of breast cancers. By making use of specific classification techniques for diagnosis of breast cancers, Shweta Kharia observed from a comprehensive survey and claimed that, decision tree yielded excessive accuracy rate and is the first-class predictor among the involved techniques and the Bayesian network, a well-known technique that used in medical world. Their experimental outcomes stated that, support Vector machine obtained high classification accuracy compared to other classifiers. Using C4.5 algorithm patients were categorized into either "Carcinoma in situ" or "Malignant potential" group and confirmed that to diagnose breast cancers, C4.5 obtained high accuracy by Rajesh Etal. The most prominent methods used in mining of data are Classification and Prediction. Supervised learning is performed for classification tasks. Some portion of data called is taken as training set comprising of instances. Further each instance comprises collection of features and further these features describes one single entity known as a class. The foremost objective of classification technique is to generate a model with proficiency of forecasting the class label as accurate as feasible in earlier hidden records. Furthermore, to predict the correctness or accuracy of the created model, a test set is used. Classifying tumor cells, studying the effectiveness of remedies are some applications of classification in medical diagnosis.

[2]. "Breast cancer analysis using Logistic Regression", H. Yusuff:

There are many different types of breast cancer, with different stages or spread, aggressiveness, and genetic makeup. Survival rates for breast cancer may be increased when the disease is detected in its earlier stage through mammograms. The implementation of mass screening would result in increased caseloads for radiologists. This will increase chances of improper diagnosis. The prediction using logistic regression would aid the radiologist to detect the breast cancer. The patient's history is used to predict and detect whether the patient had breast cancer or not. The patient’s history include information about their age, menopause condition, age of menopause, whether the patient had any first degree relative with history of breast cancer or family member having cancer other than breast cancer, and if the patient had breast trauma. These variables may be the cause of breast cancer. The patient’s history can help the doctor to decide on the next mode of detection procedure. The next cause of action is to conduct clinical examination. The clinical examination is physical examination of both breasts by the doctors using the hands (Goodson III, 2010). The clinical examination includes inspection and palpation of the entire breast area including the lymph node areas above and below the collarbone under each arm. The doctor will gently palpate each breast. Special attention will be given to the shape and texture of the breasts, location of any lumps, and whether such lumps are attached to the skin or to deeper tissues. The next procedure in breast examination is to undergo mammographic screening which is to aid in the diagnosis of breast disease in women. Mammography is a common screening method since it is relatively fast and is widely available in developed countries Logistic regression is one of the variety of popular multivariate tools used in biomedical informatics. It is one of the most common models for prediction and has been applied to cancer prediction (Samatha, 2009; Zhou, 2004). From previous studies, logistic regression is widely used in medical literature especially for correlating the dichotomous outcomes with the predictor variables that include different physiological data. In logistic regression, the predicted odd ratio of positive outcome is expressed as a sum of product. Product is formed by multiplying the values of independent variable and its coefficients. The probability of positive outcome is obtained from the odd ratio through a simple transformation (Samatha, 2009). The problems are formulated first from the logistic regression. Then, the coefficient obtained from the logistic regression is used to calculate the predictor variables

[3]. “Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression”- Jitendra Kumar Jaiswal

Feature subset selection becomes quite important and predominant in the case of data sets those are contained with higher number of variables. It discards insignificant variables and produces efficient and improved prediction performance on the class variables that is more cost effective and more reliable understanding of the data. Random forest has been emerged as a quite efficient and robust algorithm that can handle feature selection problem even with the higher number of variables. It is also very much efficient while dealing with Missing data imputation, classification, and regression problems. It can also handle outliers and noisy data very well. In this paper we applied the concept of random forest algorithm on the feature subset selection and classification and regression to perform the comparative study of the random forest algorithm in different perspectives.

[4] "Logistic Regression ": Wright, Raymond E.

Although logistic regression is used primarily with dichotomous dependent variables, the technique can be extended to situations involving outcome variables with 3 or more categories (polynomial, or multinomial, dependent variables) / give an overview of the logistic regression model / discuss the main similarities and differences between logistic regression and linear regression and the basic assumptions of logistic regression / use data from a hypothetical study to show how to interpret a logistic regression analysis / in particular, how to interpret model coefficients, test hypotheses, and interpret classification results / use data from actual research studies to show how to interpret logistic regression analyses that involve more than 1 predictor variable / describe model-building procedures for studies that have many potential predictor variables.

[5] "Data Classification Using Support Vector Machine ": Durgesh K. Srivastava & Lekha Bhambhu

Classification is one of the most important tasks for different application such as text categorization, tone recognition, image classification, micro-array gene expression, proteins structure predictions, data Classification etc. Most of the existing supervised classification methods are based on traditional statistics, which can provide ideal results when sample size is tending to infinity. However, only finite samples can be acquired in practice. In this paper, a novel learning method, Support Vector Machine (SVM), is applied on different data (Diabetes data, Heart Data, Satellite Data and Shuttle data) which have two or multi class. SVM, a powerful machine method developed from statistical learning and has made significant achievement in some field. Introduced in the early 90’s, they led to an explosion of interest in machine learning.

SUMMARY:

In this project we are using machine learning techniques to assess tumor behavior for breast cancer patients. One hassle is that there is a class imbalance in the training data, since the probability of not having this disease is higher than the one of having it. This project introduces a simple way to detect at what level the cancer is using classifier, logistic regression and random forest algorithms with respect to accuracy in detection of breast cancer. Our goal is to find the cancer stage by proposing a suitable method that can predict if it is preliminary level or last level.

Chapter 3

SOFTWARE AND HARDWARE REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS:

3.1.1 Operating system:

- Windows 7, Windows 8 or Windows 10
- Mac OSX 10.8, 10.9, 10.10 or 10.11
- Linux - Debian, Fedora, Ubuntu • Language: Python (Python 2.7, or Python 3.5 or newer)
- IDE: Jupiter Notebook/Google Collab

3.1.2 Browsers:

- Mozilla Firefox
- Internet Explorer
- Google Chrome
- Opera

3.2 HARDWARE REQUIREMENTS:

- 64-bit versions of Microsoft Windows 10, 8, 7
- 4 GB RAM minimum, 8 GB RAM recommended
- 1.5 GB hard disk space + at least 1 GB for caches
- 1024 x 768 minimum screen resolution

3.3 SOFTWARE DESCRIPTION:

3.3.1 Anaconda Navigator:

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage Anaconda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- Jupyter Lab
- Jupyter Notebook

- Qt Console
- Spyder
- Glueviz
- Orange
- RStudio
- Visual Studio Code

3.3.2 Jupyter Notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning and much more. Notebook documents: It is the document produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc.) Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc...) as well as executable documents which can be run to perform data analysis.

Notebook Dashboard: It is the component which is shown first when you launch Jupyter Notebook App. The Notebook Dashboard is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown). The Notebook Dashboard has other features similar to a file manager, namely navigating folders and renaming/deleting files.
Notebook kernel: It is a “computational engine” that executes the code contained in a Notebook document. The I python kernel, referenced in this guide, executes python code. Kernels for many other languages exist (official kernels).

When you open a Notebook document, the associated kernel is automatically launched. When the notebook is executed (either cell-by-cell or with menu Cell -> Run All), the kernel performs the computation and produces the results. Depending on the type of computations, the kernel may consume significant CPU and RAM. Note that the RAM is not released until the kernel is shut-down.

3.4 PACKAGES REQUIRED

3.4.1 NumPy

NumPy open-source software and a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high level mathematical functions to operate on these arrays. Incorporating features of the competing Num array into Numeric, with extensive modifications. NumPy targets the C Python reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops using NumPy. Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. NumPy is intrinsically integrated with Python, a more modern and complete programming language. Complementary Python packages are available. SciPy is a library that adds more MATLAB-like functionality and Matplotlib is a plotting package that provides MATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient linear algebra computations. Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three dimensional arrays, indexing, slicing or masking with other arrays are very efficient ways to access specific pixels of an image.

3.4.2 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. Data Frame object for data manipulation with integrated indexing Tools for reading and writing data between in-memory data structures and different file formats. Data alignment and integrated handling of missing data. Reshaping and pivoting of data sets. Label based slicing, fancy indexing, and sub setting of large data sets. Data structure column insertion and deletion. Group by engine allowing split-apply- combine operations on data sets. Data set merging and joining.
Hierarchical axis indexing to work with high-dimensional data in a lower- dimensional data structure. Time series functionality date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging. Provides data filtration. The library is highly optimized for performance, with critical code paths written in Python or C.

3.4.3 Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Scikit-learn is largely written in Python, and uses NumPy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in python to improve performance. Support vector machines are implemented by a python wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible.

Chapter 4

SYSTEM DEVELOPMENT PROCESSS

4.1 SYSTEM DESIGN

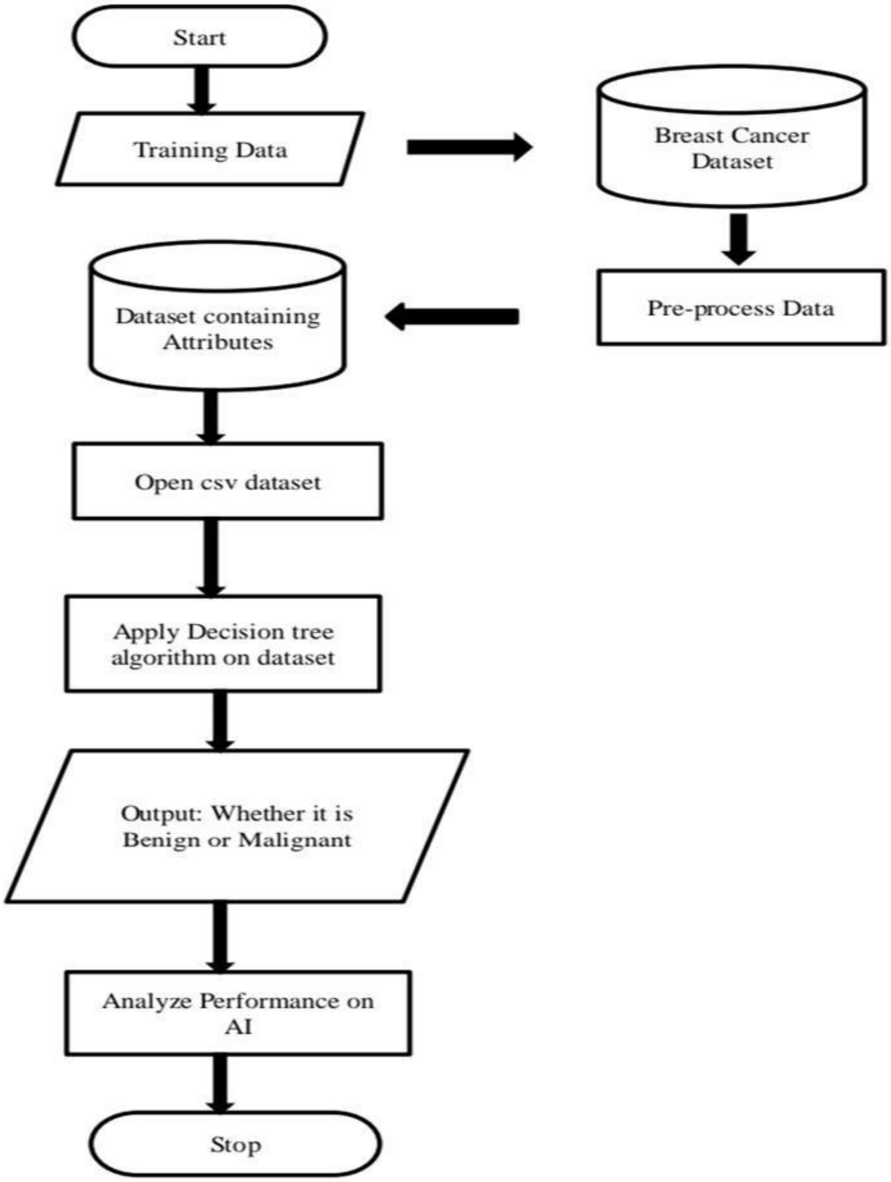

Start
Training Data
Breast Cancer Dataset
Pre-process Data
Dataset containing Attributes
Open csv dataset
Apply Decision tree algorithm on dataset
Output: Whether it is Benign or Malignant
Analyze Performance on AI
Stop

Fig 4.1: Flowchart

The proposed design system is simple as shown in the flow chart above. After loading the dataset and understanding the insights for the dataset, the training phase extracts the features from the dataset and testing phase is used to determine how the appropriate model behaves for prediction.

Then the dataset is divided into 2 sections. These are the Training and Testing Phase.
Once the pre-processing is done, we get dataset containing attributes without null values. After applying the algorithm on then we get the output as Benign or Malignant.

This system also presents a comparison of machine learning (ML) algorithms: Support machine vector (SVM), Random Forest (RT), Logistic Regression (LR). This enables the accuracy in predicting the accuracy, With the patient’s pervious data the algorithm detects the cancer stage.

Chapter 5

METHODOLOGY

- **Pre-processing data:**

  The first phase in we do is to collect the data that we are interested in collecting for pre-processing and to apply classification and Regression methods. Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real world data is often incomplete, inconsistent, and lacking certain to contain many errors. Data pre-processing is a proven method of resolving such issues. Data pre-processing prepares raw data for further processing.

- **Data Preparation:**

  Data Preparation is where we load our data into a suitable place and prepare it for use in our machine learning training. We will first put all our data together, and then randomize the ordering.

- **Feature Selection:**

  Feature selection, also known as variable selection, attribute selection, is the process of selection a subset of relevant features for use in model construction. Data Set from Kaggle repository and out of 31 parameters we have selected about 8-9 parameters.
  Our target parameter is breast cancer diagnosis - malignant or benign.

- **Feature Projection:**

  Feature projection is transformation of high-dimensional space data to a lower dimensional space (with few attributes). Both linear and nonlinear reduction techniques can be used in accordance with the type of relationships among the features in the dataset.

- **Model Selection**

  Supervised learning is the method in which the machine is trained on the data which the input and output are well labelled. The model can learn on the training data and can process the future data to predict outcome. They are grouped to Regression and Classification techniques

- **Prediction:**

  Prediction, or inference, is the step where we get to answer some questions. This is the point of all this work, where the value of machine learning is real.

Chapter 6

IMPLEMENTATION

6.1 Import Libraries

```
import numpy as np
import sklearn.datasets
```

6.2 Load the dataset

```
breast_cancer = sklearn.datasets.load_breast_cancer()
```

6.3 Print the data present in the breast_cancer(dataset)

```
print(breast_cancer)
```

6.4 Extracting test data and print

```
X = breast_cancer.data
Y = breast_cancer.target
print(X)
print(Y)
```

6.5 Checking out data instances

```
print(X.shape, Y.shape)
```

6.6 Importing data to the pandas DataFrame

```
import pandas as pd
data=pd.DataFrame(breast_cancer.data,columns=breast_cancer.feature_names)
```

6.7 Adding target to this class

```
data['class'] = breast_cancer.target
data.head()
```

6.8 Displaying statistical values

```
data.describe()
```

6.9 Count the data for Malignant and Benign

```
print(data['class'].value_counts())
print(breast_cancer.target_names)
```

6.10 Finding the mean for malignant and benign by grouping the data

```
data.groupby('class').mean()
```

6.11 Train and Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y)

print(Y.shape, Y_train.shape, Y_test.shape)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1)
print(Y.shape, Y_train.shape, Y_test.shape)
print(Y.mean(), Y_train.mean(), Y_test.mean())
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1, stratify=Y)
print(Y.mean(), Y_train.mean(), Y_test.mean())
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1, stratify=Y,
random_state=1)
print(X_train.mean(), X_test.mean(), X.mean())
print(X_train)
```

6.12 Logistic Regression

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier.fit?
```

6.13 Random Forest

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier()
```

6.14 Support Vector Machine

```
from sklearn.svm import SVC
classifier = SVC()
```

6.15 Evaluation of the model

```
from sklearn.metrics import accuracy_score
prediction_on_training_data = classifier.predict(X_train)
accuracy_on_training_data=accuracy_score(Y_train, prediction_on_training_data)
print('Accuracy on training data : ', accuracy_on_training_data)
classifier.fit(X_train, Y_train)

 prediction_on_test_data = classifier.predict(X_test)
accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)
print('Accuracy on test data : ', accuracy_on_test_data)
```

6.16 Detecting whether the Patient has breast cancer in benign or Malignant stage

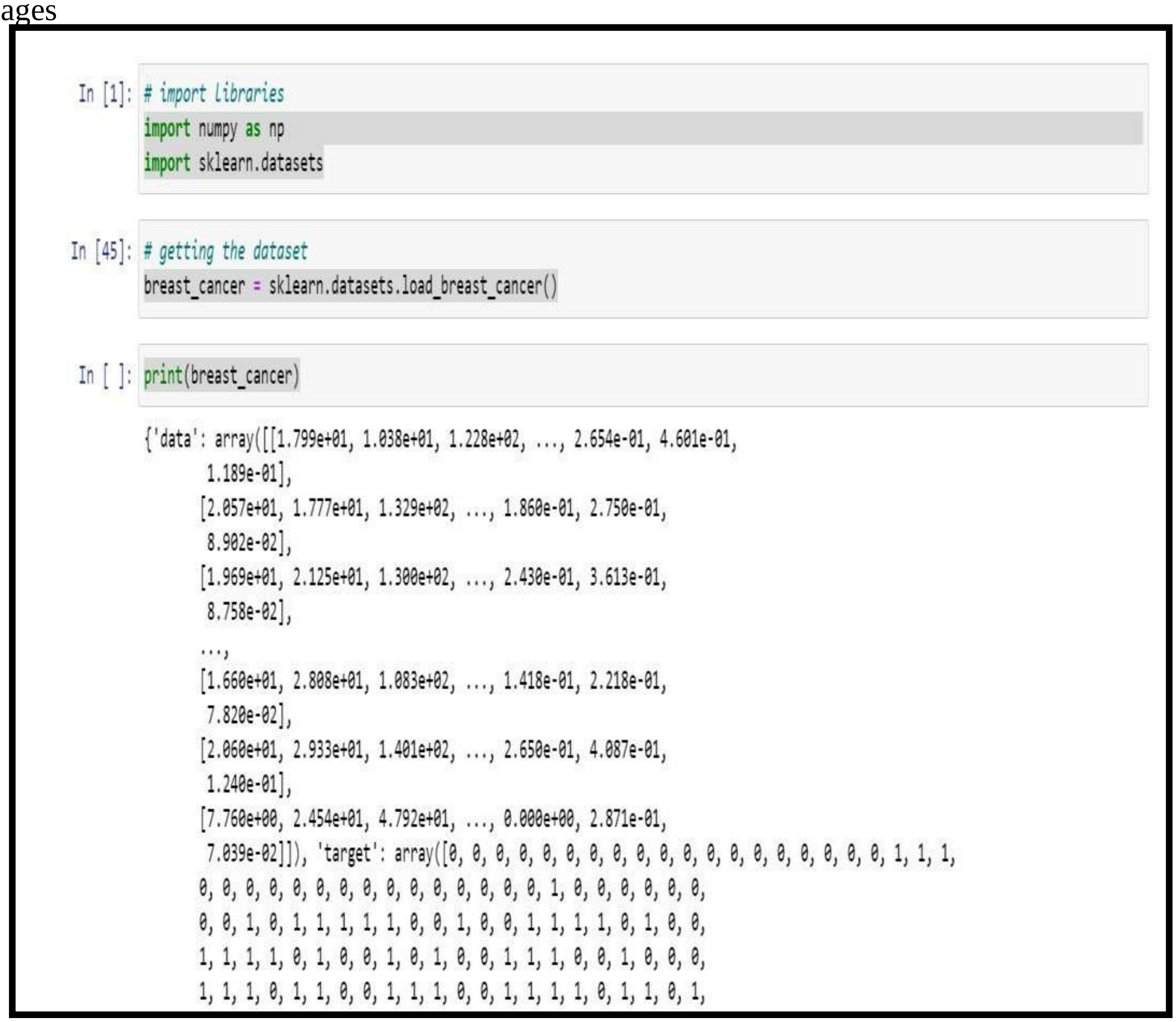
```
 input_data=(17.99,10.38,122.8,1001,0.1184,0.2776,0.3001,0.1471,0.2419,0.07871,1.09
  5,0.9053,8.589,153.4,0.006399,0.04904,0.05373,0.01587,0.03003,0.006193,25.38,17.33
,184.6,2019,0.1622,0.6656,0.7119,0.2654,0.4601,0.1189)

# change the input_data to numpy_array to make prediction
input_data_as_numpy_array = np.asarray(input_data)
print(input_data)

 # reshape the array as we are predicting the output for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

 #prediction
 prediction = classifier.predict(input_data_reshaped)
print(prediction) # returns a list with element [0] if Malignant; returns a listwith
element[1], if benign.

if (prediction[0]==0):
print('The breast Cancer is Malignant')
else:
print('The breast cancer is Benign')
```

Chapter 7

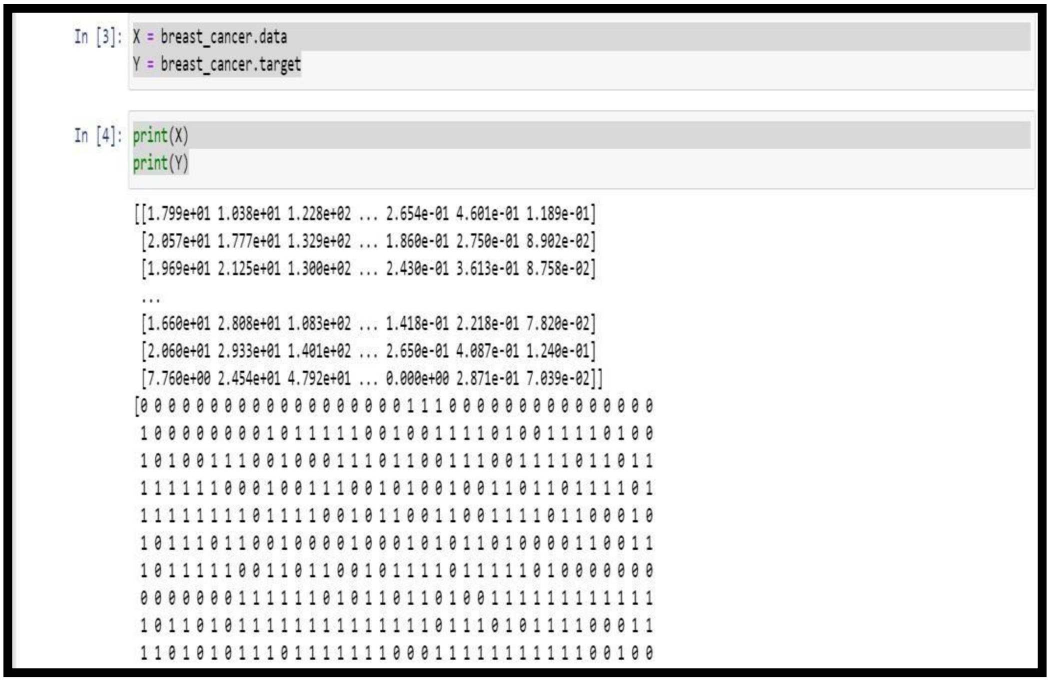
RESULTS AND DISCUSSION

Breast cancer detection may be accomplished with the assist of modern machine learning algorithms. In this project, we focus on a way to cope with imbalanced facts which have lacking values the usage of resampling strategies to decorate the type accuracy of detecting breast cancer. In our work, classifier carried out on specific breast cancer datasets. Results display that the usage of the resample filter in the preprocessing section complements the classifier's performance. In the future, the identical experiments will apply to specific classifiers and specific datasets. Data is grouped based on malignant and begnin tumors. 0 indicates malignant and 1 indicates benign. If result is 1 the person is having cancer at first stage else 0 it is malignant which means later stage of cancer. Converting data into numpy and using input data from the features given in dataset, predicting the malignant and begnin stage.

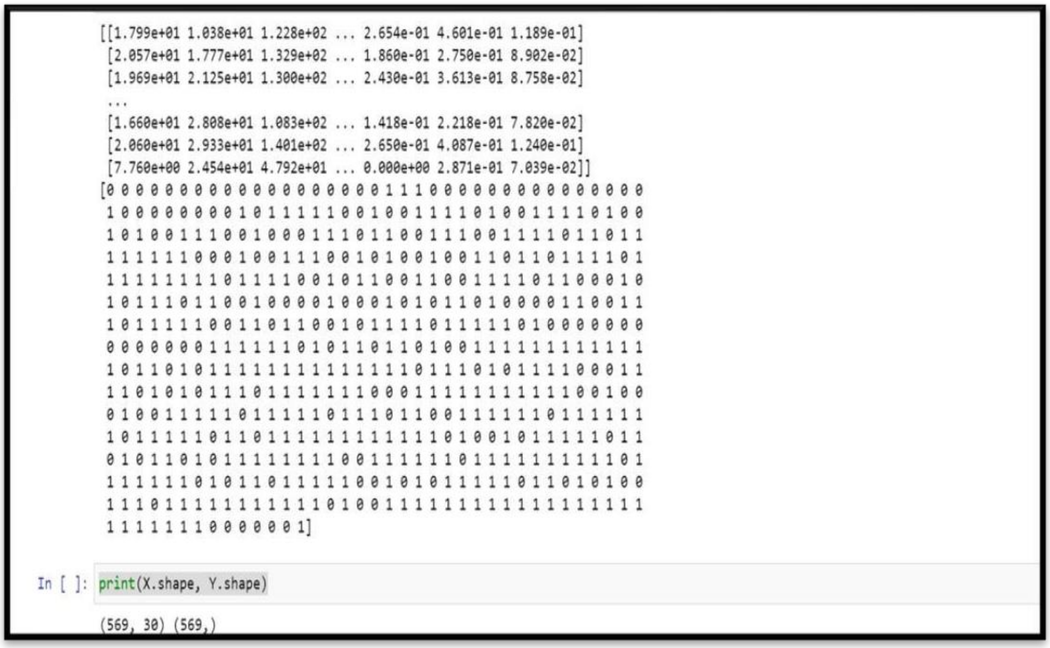
7.1 Images



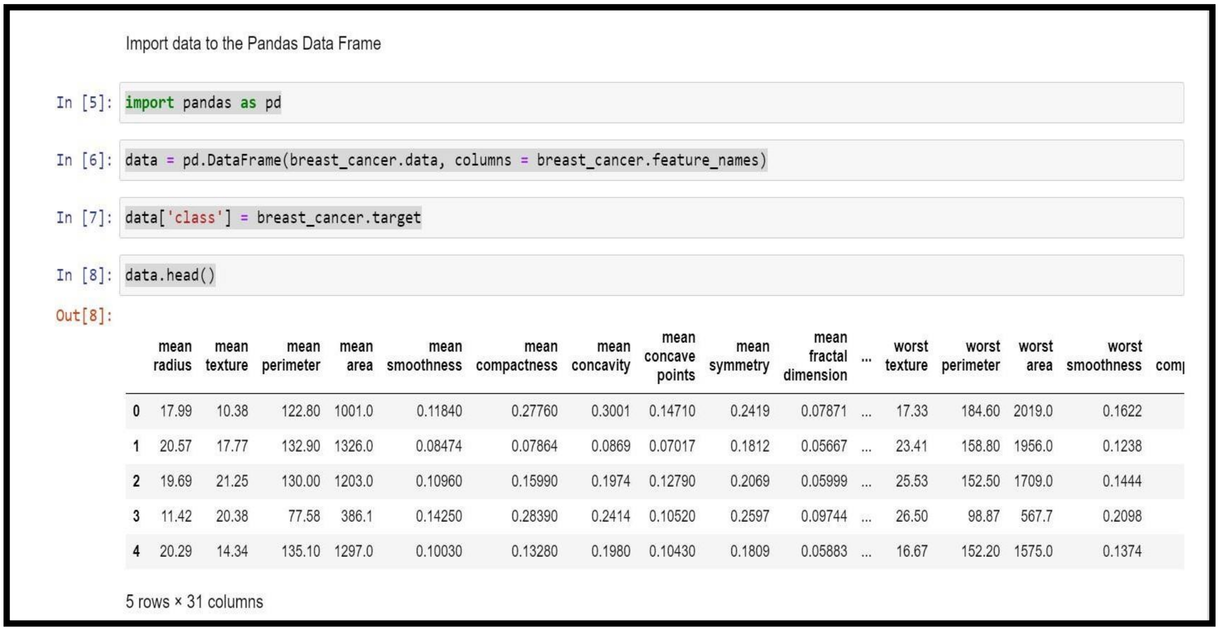
```
In [1]: # import Libraries
        import numpy as np
        import sklearn.datasets

In [45]: # getting the dataset
         breast_cancer = sklearn.datasets.load_breast_cancer()

In [ ]: print(breast_cancer)

        {'data': array([[1.799e+01, 1.038e+01, 1.228e+02, ..., 2.654e-01, 4.601e-01,
                 1.189e-01],
                [2.057e+01, 1.777e+01, 1.329e+02, ..., 1.860e-01, 2.750e-01,
                 8.902e-02],
                [1.969e+01, 2.125e+01, 1.300e+02, ..., 2.430e-01, 3.613e-01,
                 8.758e-02],
                ...,
                [1.660e+01, 2.808e+01, 1.083e+02, ..., 1.418e-01, 2.218e-01,
                 7.820e-02],
                [2.060e+01, 2.933e+01, 1.401e+02, ..., 2.650e-01, 4.087e-01,
                 1.240e-01],
                [7.760e+00, 2.454e+01, 4.792e+01, ..., 0.000e+00, 2.871e-01,
                 7.039e-02]]), 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
                0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
                1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
                1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
```



```
In [3]: X = breast_cancer.data
        Y = breast_cancer.target

In [4]: print(X)
        print(Y)

[[1.799e+01 1.038e+01 1.228e+02 ... 2.654e-01 4.601e-01 1.189e-01]
 [2.057e+01 1.777e+01 1.329e+02 ... 1.860e-01 2.750e-01 8.902e-02]
 [1.969e+01 2.125e+01 1.300e+02 ... 2.430e-01 3.613e-01 8.758e-02]
 ...
 [1.660e+01 2.808e+01 1.083e+02 ... 1.418e-01 2.218e-01 7.820e-02]
 [2.060e+01 2.933e+01 1.401e+02 ... 2.650e-01 4.087e-01 1.240e-01]
 [7.760e+00 2.454e+01 4.792e+01 ... 0.000e+00 2.871e-01 7.039e-02]]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 0
 1 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 1 1 0 1 1 0 1 1
 1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0 1
 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 1 1 0 0 0 1 0
 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1
 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 0 1 1
 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0
```



```
[[1.799e+01 1.038e+01 1.228e+02 ... 2.654e-01 4.601e-01 1.189e-01]
 [2.057e+01 1.777e+01 1.329e+02 ... 1.860e-01 2.750e-01 8.902e-02]
 [1.969e+01 2.125e+01 1.300e+02 ... 2.430e-01 3.613e-01 8.758e-02]
 ...
 [1.660e+01 2.808e+01 1.083e+02 ... 1.418e-01 2.218e-01 7.820e-02]
 [2.060e+01 2.933e+01 1.401e+02 ... 2.650e-01 4.087e-01 1.240e-01]
 [7.760e+00 2.454e+01 4.792e+01 ... 0.000e+00 2.871e-01 7.039e-02]]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 0
 1 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 1 1 0 1 1 0 1 1
 1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0 1
 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 1 1 0 0 0 1 0
 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1
 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 0 1 1
 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0
 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1
 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 0 1 1
 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1
 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0 1 0 1 0 0
 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 1]

In [ ]: print(X.shape, Y.shape)

(569, 30) (569,)
```



Import data to the Pandas Data Frame

```
In [5]: import pandas as pd
```

```
In [6]: data = pd.DataFrame(breast_cancer.data, columns = breast_cancer.feature_names)
```

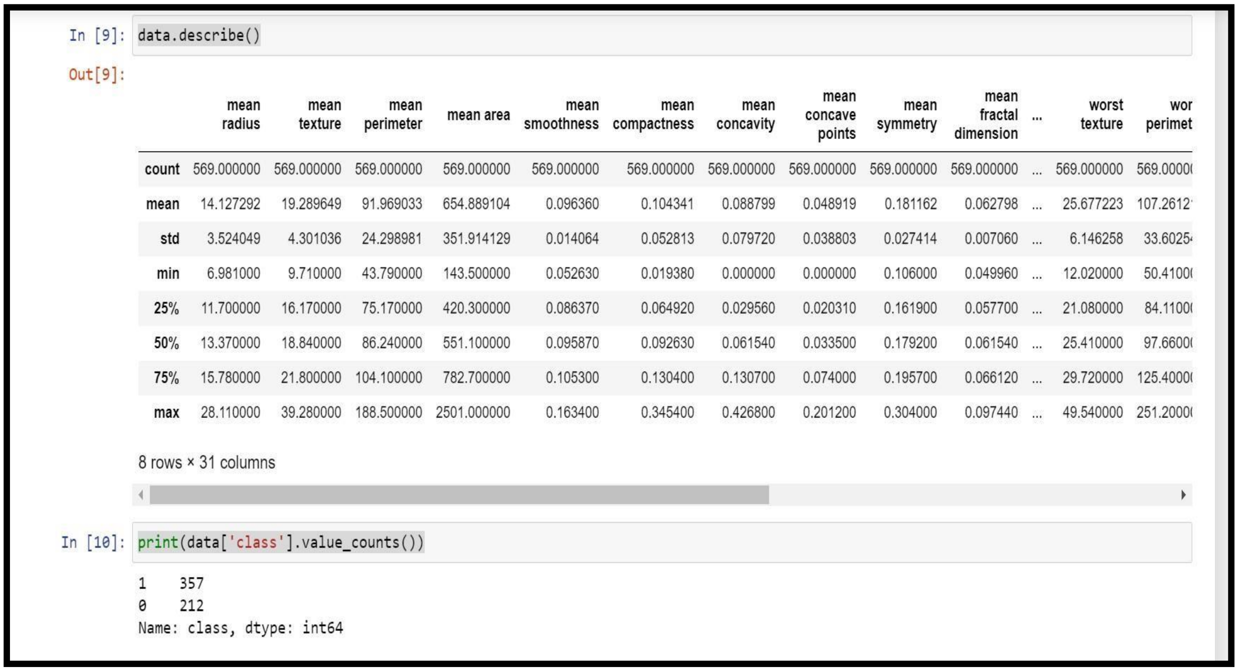
```
In [7]: data['class'] = breast_cancer.target
```

```
In [8]: data.head()
```

Out[8]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	comp
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	17.33	184.60	2019.0	0.1622	
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	23.41	158.80	1956.0	0.1238	
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	25.53	152.50	1709.0	0.1444	
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	26.50	98.87	567.7	0.2098	
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	16.67	152.20	1575.0	0.1374	

5 rows × 31 columns



```
In [9]: data.describe()
```

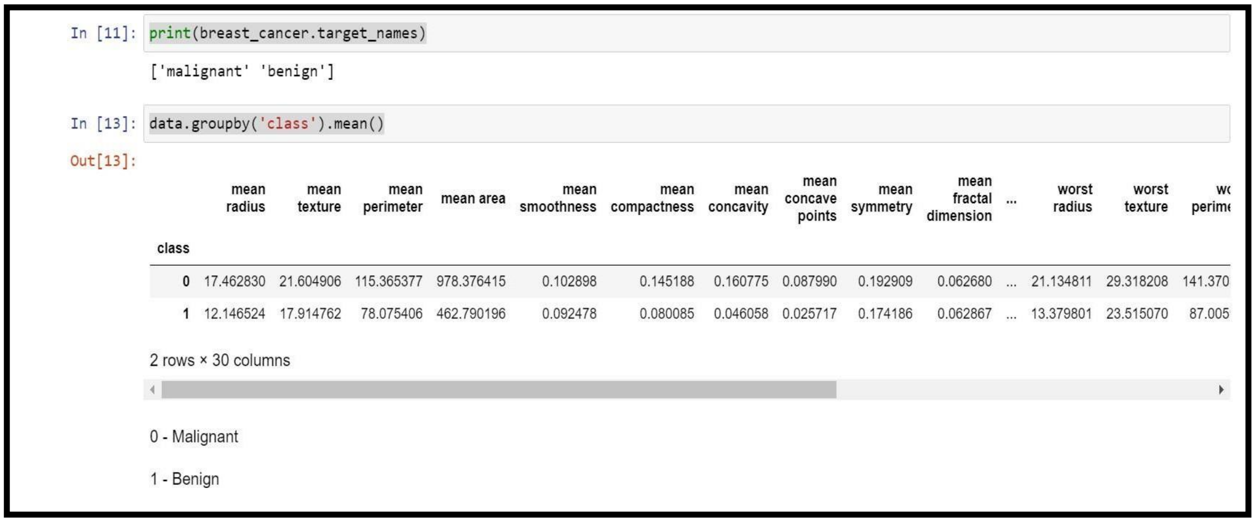
Out[9]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	wor perimet
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	...	569.000000	569.0000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	0.062798	...	25.677223	107.2612
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	0.007060	...	6.146258	33.6025
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	0.049960	...	12.020000	50.4100
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	0.057700	...	21.080000	84.1100
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	0.061540	...	25.410000	97.6600
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	0.066120	...	29.720000	125.4000
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	0.097440	...	49.540000	251.2000

8 rows × 31 columns

```
In [10]: print(data['class'].value_counts())
```

```
1    357
0    212
Name: class, dtype: int64
```



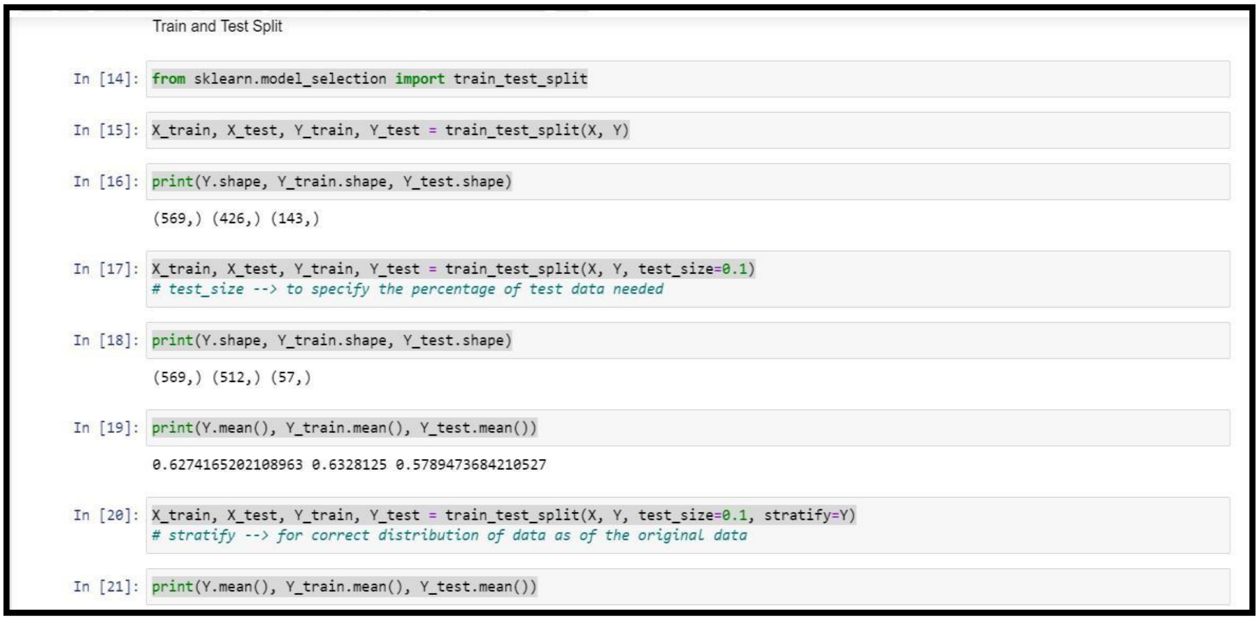
```
In [11]: print(breast_cancer.target_names)

         ['malignant' 'benign']

In [13]: data.groupby('class').mean()
Out[13]:
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	wo perime
class														
0	17.462830	21.604906	115.365377	978.376415	0.102898	0.145188	0.160775	0.087990	0.192909	0.062680	...	21.134811	29.318208	141.370
1	12.146524	17.914762	78.075406	462.790196	0.092478	0.080085	0.046058	0.025717	0.174186	0.062867	...	13.379801	23.515070	87.005

2 rows × 30 columns

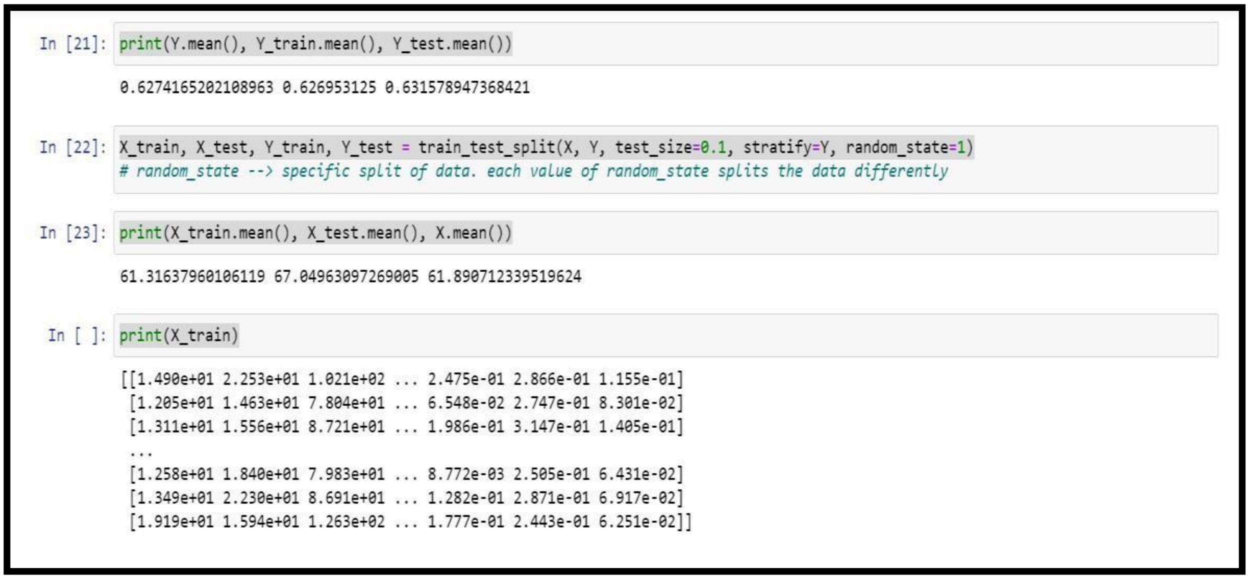
0 - Malignant

1 - Benign



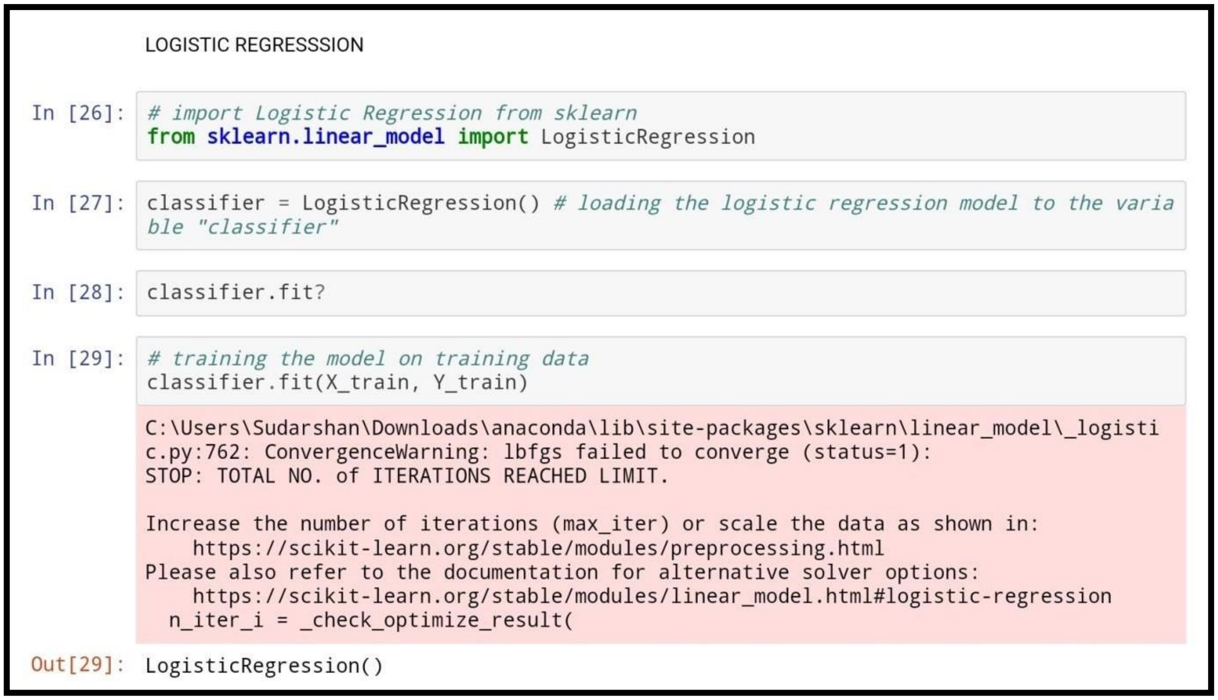
Train and Test Split

```
In [14]: from sklearn.model_selection import train_test_split

In [15]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y)

In [16]: print(Y.shape, Y_train.shape, Y_test.shape)

         (569,) (426,) (143,)

In [17]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1)
         # test_size --> to specify the percentage of test data needed

In [18]: print(Y.shape, Y_train.shape, Y_test.shape)

         (569,) (512,) (57,)

In [19]: print(Y.mean(), Y_train.mean(), Y_test.mean())

         0.6274165202108963 0.6328125 0.5789473684210527

In [20]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1, stratify=Y)
         # stratify --> for correct distribution of data as of the original data

In [21]: print(Y.mean(), Y_train.mean(), Y_test.mean())
```



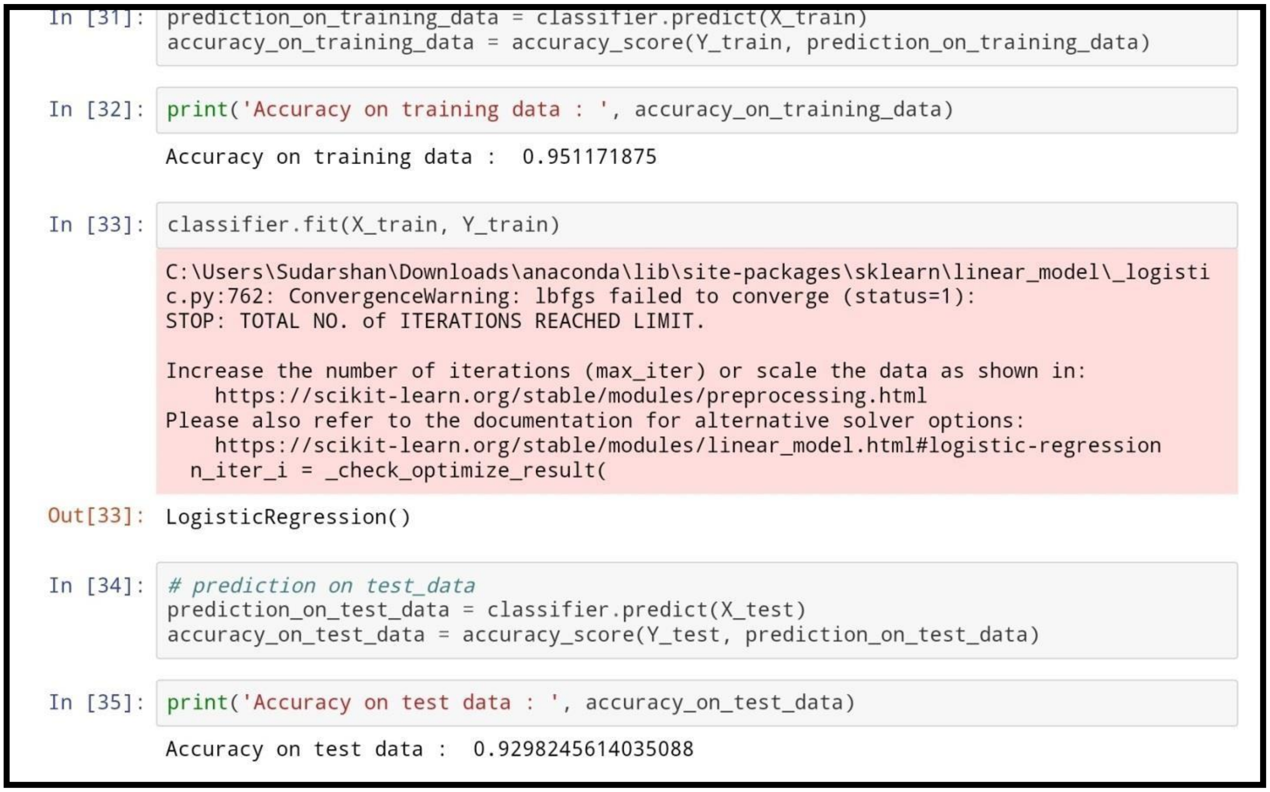
```
In [21]: print(Y.mean(), Y_train.mean(), Y_test.mean())

         0.6274165202108963 0.626953125 0.631578947368421

In [22]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1, stratify=Y, random_state=1)
         # random_state --> specific split of data. each value of random_state splits the data differently

In [23]: print(X_train.mean(), X_test.mean(), X.mean())

         61.31637960106119 67.04963097269005 61.890712339519624

In [ ]: print(X_train)

         [[1.490e+01 2.253e+01 1.021e+02 ... 2.475e-01 2.866e-01 1.155e-01]
          [1.205e+01 1.463e+01 7.804e+01 ... 6.548e-02 2.747e-01 8.301e-02]
          [1.311e+01 1.556e+01 8.721e+01 ... 1.986e-01 3.147e-01 1.405e-01]
          ...
          [1.258e+01 1.840e+01 7.983e+01 ... 8.772e-03 2.505e-01 6.431e-02]
          [1.349e+01 2.230e+01 8.691e+01 ... 1.282e-01 2.871e-01 6.917e-02]
          [1.919e+01 1.594e+01 1.263e+02 ... 1.777e-01 2.443e-01 6.251e-02]]
```

7.1.1 Train and Test Split



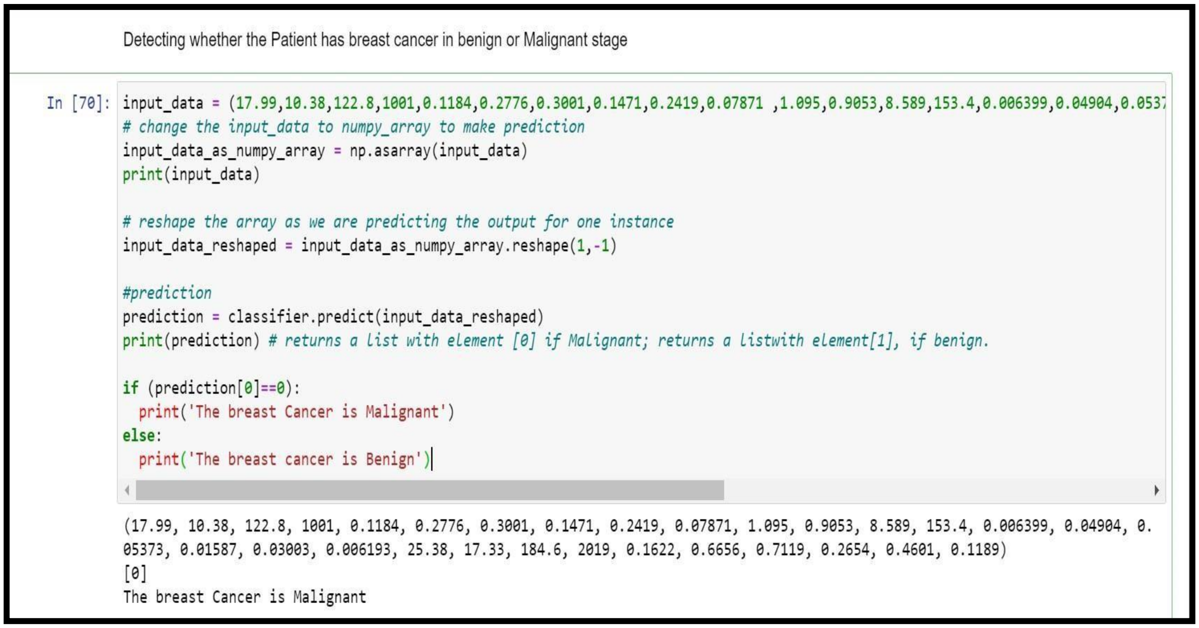
LOGISTIC REGRESSSION

```
In [26]: # import Logistic Regression from sklearn
         from sklearn.linear_model import LogisticRegression

In [27]: classifier = LogisticRegression() # loading the logistic regression model to the varia
         ble "classifier"

In [28]: classifier.fit?

In [29]: # training the model on training data
         classifier.fit(X_train, Y_train)

         C:\Users\Sudarshan\Downloads\anaconda\lib\site-packages\sklearn\linear_model\_logisti
         c.py:762: ConvergenceWarning: lbfgs failed to converge (status=1):
         STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

         Increase the number of iterations (max_iter) or scale the data as shown in:
             https://scikit-learn.org/stable/modules/preprocessing.html
         Please also refer to the documentation for alternative solver options:
             https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
           n_iter_i = _check_optimize_result(

Out[29]: LogisticRegression()
```



```
In [31]: prediction_on_training_data = classifier.predict(X_train)
         accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)

In [32]: print('Accuracy on training data : ', accuracy_on_training_data)

         Accuracy on training data :  0.951171875

In [33]: classifier.fit(X_train, Y_train)

         C:\Users\Sudarshan\Downloads\anaconda\lib\site-packages\sklearn\linear_model\_logisti
         c.py:762: ConvergenceWarning: lbfgs failed to converge (status=1):
         STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

         Increase the number of iterations (max_iter) or scale the data as shown in:
             https://scikit-learn.org/stable/modules/preprocessing.html
         Please also refer to the documentation for alternative solver options:
             https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
           n_iter_i = _check_optimize_result(

Out[33]: LogisticRegression()

In [34]: # prediction on test_data
         prediction_on_test_data = classifier.predict(X_test)
         accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)

In [35]: print('Accuracy on test data : ', accuracy_on_test_data)

         Accuracy on test data :  0.9298245614035088
```

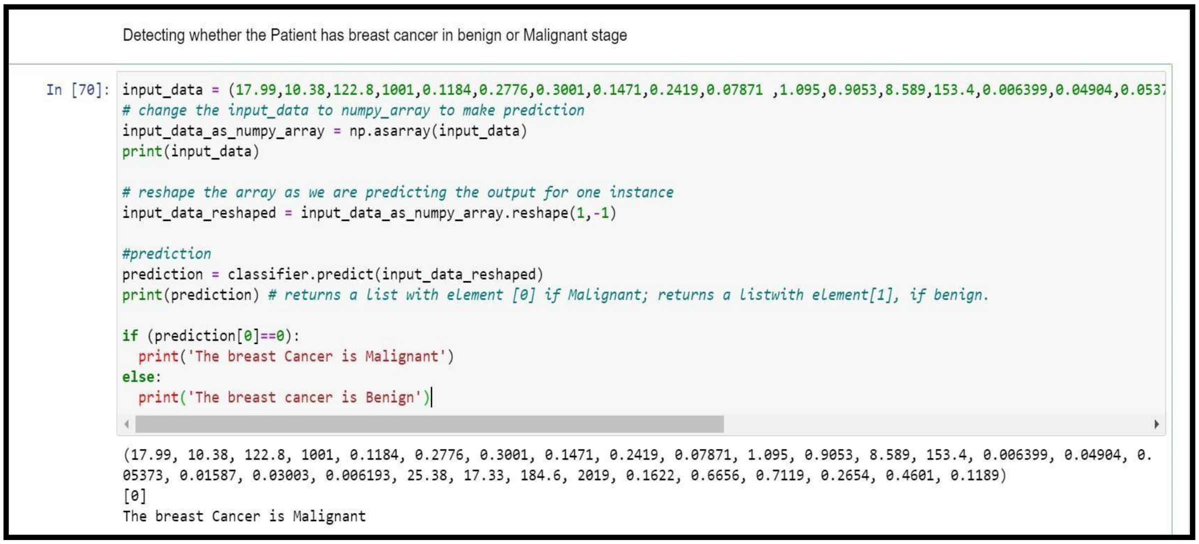
7.1.2 Logistic Regression and evaluation model



Detecting whether the Patient has breast cancer in benign or Malignant stage

```
In [70]: input_data = (17.99,10.38,122.8,1001,0.1184,0.2776,0.3001,0.1471,0.2419,0.07871 ,1.095,0.9053,8.589,153.4,0.006399,0.04904,0.0537
         # change the input_data to numpy_array to make prediction
         input_data_as_numpy_array = np.asarray(input_data)
         print(input_data)

         # reshape the array as we are predicting the output for one instance
         input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

         #prediction
         prediction = classifier.predict(input_data_reshaped)
         print(prediction) # returns a list with element [0] if Malignant; returns a listwith element[1], if benign.

         if (prediction[0]==0):
           print('The breast Cancer is Malignant')
         else:
           print('The breast cancer is Benign')

         (17.99, 10.38, 122.8, 1001, 0.1184, 0.2776, 0.3001, 0.1471, 0.2419, 0.07871, 1.095, 0.9053, 8.589, 153.4, 0.006399, 0.04904, 0.
         05373, 0.01587, 0.03003, 0.006193, 25.38, 17.33, 184.6, 2019, 0.1622, 0.6656, 0.7119, 0.2654, 0.4601, 0.1189)
         [0]
         The breast Cancer is Malignant
```

7.1.3 Random Forest and evaluation model

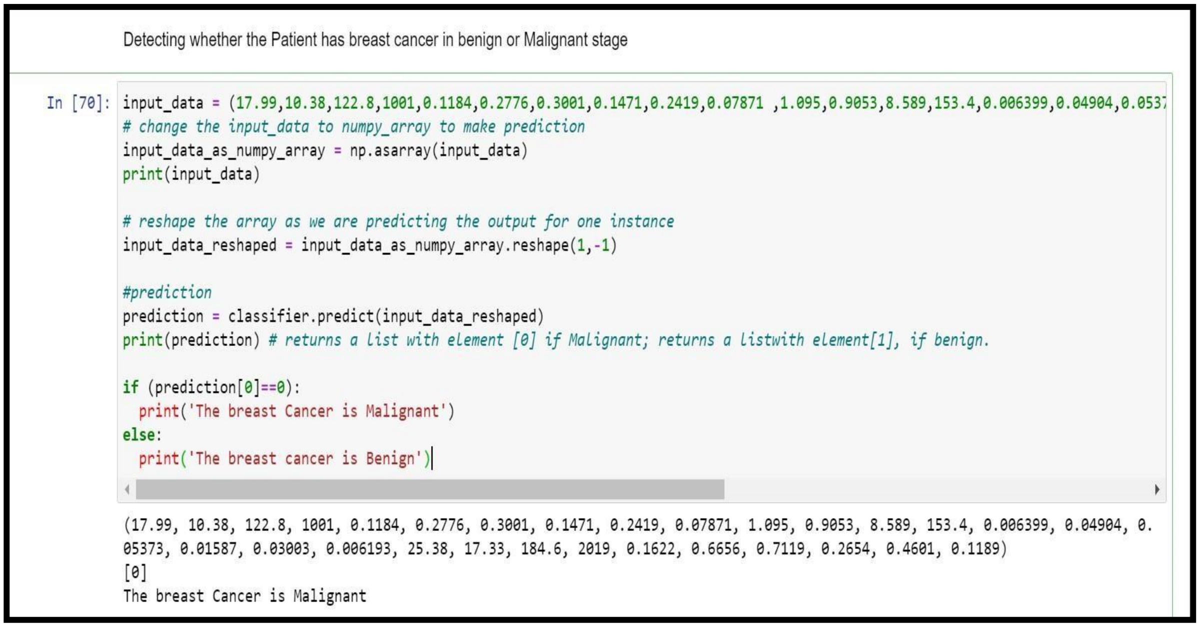


Detecting whether the Patient has breast cancer in benign or Malignant stage

```
In [70]: input_data = (17.99,10.38,122.8,1001,0.1184,0.2776,0.3001,0.1471,0.2419,0.07871 ,1.095,0.9053,8.589,153.4,0.006399,0.04904,0.0537
         # change the input_data to numpy_array to make prediction
         input_data_as_numpy_array = np.asarray(input_data)
         print(input_data)

         # reshape the array as we are predicting the output for one instance
         input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

         #prediction
         prediction = classifier.predict(input_data_reshaped)
         print(prediction) # returns a list with element [0] if Malignant; returns a listwith element[1], if benign.

         if (prediction[0]==0):
           print('The breast Cancer is Malignant')
         else:
           print('The breast cancer is Benign')
```

```
(17.99, 10.38, 122.8, 1001, 0.1184, 0.2776, 0.3001, 0.1471, 0.2419, 0.07871, 1.095, 0.9053, 8.589, 153.4, 0.006399, 0.04904, 0.
05373, 0.01587, 0.03003, 0.006193, 25.38, 17.33, 184.6, 2019, 0.1622, 0.6656, 0.7119, 0.2654, 0.4601, 0.1189)
[0]
The breast Cancer is Malignant
```

7.1.4 Support Vector and evaluation model



Detecting whether the Patient has breast cancer in benign or Malignant stage

```
In [70]: input_data = (17.99,10.38,122.8,1001,0.1184,0.2776,0.3001,0.1471,0.2419,0.07871 ,1.095,0.9053,8.589,153.4,0.006399,0.04904,0.0537
         # change the input_data to numpy_array to make prediction
         input_data_as_numpy_array = np.asarray(input_data)
         print(input_data)

         # reshape the array as we are predicting the output for one instance
         input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

         #prediction
         prediction = classifier.predict(input_data_reshaped)
         print(prediction) # returns a list with element [0] if Malignant; returns a listwith element[1], if benign.

         if (prediction[0]==0):
           print('The breast Cancer is Malignant')
         else:
           print('The breast cancer is Benign')
```

```
(17.99, 10.38, 122.8, 1001, 0.1184, 0.2776, 0.3001, 0.1471, 0.2419, 0.07871, 1.095, 0.9053, 8.589, 153.4, 0.006399, 0.04904, 0.
05373, 0.01587, 0.03003, 0.006193, 25.38, 17.33, 184.6, 2019, 0.1622, 0.6656, 0.7119, 0.2654, 0.4601, 0.1189)
[0]
The breast Cancer is Malignant
```

CONCLUSION

Breast cancer detection maybe accomplished with the assist of modern machine learning algorithms. Currently we have used three machine learning models named logistic regression model, Random Forest model and support vector machine and tried to predict the accuracy by comparing all the other algorithms used in this project.

We have focused on way to cope with facts which have a lacking on the values of the usage of resampling strategies to predict the accuracy of detecting the breast cancer.

The results display the predictions on different algorithm during the preprocessing if the classifier performances.

In the future, we would like to work on deep learning techniques such as Auto Encoder, Convolutional Neural Network and Recurrent Neural Network. Also in addition, application of sensors can be used for the prediction purpose. The following are the future scope for the project.

REFERENCES

[1] “Predicting Breast Cancer using Logistic Regression and Multiclass Classifiers”, J.sultan November 2018.

[2]. “Breast cancer analysis using Logistic Regression”, H. Yusuff.:

[3]. “Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression”- Jitendra Kumar Jaiswal

[4] "Logistic Regression ": Wright, Raymond E.

[5] “Data Classification Using Support Vector Machine ": Durgesh K. Srivastava & Lekha Bhambhu

[6] “Diagnostic Accuracy of Different Machine Learning Algorithms for Breast Cancer Risk Calculation”: Ricvan Dana Nindrea , Teguh Aryandono, Lutfan Lazuardi, and Iwan Dwiprahasto

[7] “Different Machine Learning Algorithms For Breast Cancer Diagnosis”: Adel Aloraini

[8] “Support vector machine based diagnostic system for breast cancer”: Hui-Ling Chen, Su-Jing Wang, Jie Liu, Da-You Liu

[9] “Breast Cancer Detection using Machine Learning Algorithm”: M. Udaya Bhanu, Dr.P. Kalyanasundaram

[10] “Automated Breast Cancer Diagnosis Based on Machine Learning Algorithms”: Habib Dhahri, Eslam Al Maghayreh, Mohammed Faisal Nagi