Detection of Breast Cancer with Python

Neha Singh

Indian Institute of Health Management Research (*IIHMR*), Jaipur

Abstract

Global cancer data confirms more than 2 million women diagnosed with breast cancer each year reflecting majority of new cancer cases and related deaths, making it significant public health concern.But fortunately, it is also the curable cancer in its early stage. Early diagnosis of breast cancer with timely and effective treatment services improves the prognosis and survival of patients. During classifying tumors, there are significant chances of error and false diagnosis which is needed to be worked upon. Accurate classification can prevent patients from unnecessary treatments. Thus, it is important to accurately classify patients into malignant and benign groups with right diagnosis. This study is based on machine learning (ML) algorithms, aiming to review python technique and its application in breast cancer diagnosis and prognosis by building simple machine learning model. Machine learning has unique advantage as it detects critical features from complex breast cancer datasets. The methodology is widely used for classification of pattern and forecast modelling. Theprimary data for this study is extracted from Wisconsin breast cancer database (WBCD). It is the benchmark database which compares result via different algorithms.

Introduction

Cancer is named after the body part it is originating; thus breast cancer refers to abrupt growth of cells in breast tissue. This forms a lump or mass of extra tissue, also known as tumor. Tumors are of two types viz. cancerous (malignant)or non-cancerous (benign). The term breast cancer refers to malignant tumor. Women of age range 40-55 faces the highest risk of death due to breast cancer and is ranked second highest cause of death among women [[1]]. With increased emphasis on diagnostic techniques and effective treatment, the mortality rate has decreased significantly [[2]]. The different signs and symptoms that may occur in breast cancer are; a lump or thickening compared to surrounding breast tissue, change in breast size, shape or appearance, changes in skin such as dimpling, appearance of inverted nipple, redness on skin over breast, or peeling, scaling, crusting of the pigmented area around areola or breast skin [[3]]. Taking symptoms into consideration, medical diagnosis is gradually increasing the use of classifier system. Undoubtedly, evaluating patients dataset by expert decision on it is an important factor but however, systems and artificial intelligence techniques improves diagnosis on a higher level. It will not only minimize the possible error, but also examines the medical report precisely in short period of time.

Down the line of decades, machine learning techniques have been widely used in healthcare systems. With the arrival of new technologies, it is easier to obtain and store big data, for example that of electronic patient records [[4]]. Without the aid of technology it is impossible to handle and analyse complex data sets, especially in complex interrogation of data. The healthcare system driven by technologies is an important asset. It assists professionals to diagnose patients accurately and provide more meaningful benchmark. Machine learning now handles some complex manual work in health industry like text and voice industry, which identify patients' emotion corresponding to health professional responses [[5]-[6]].

Several data mining and machine learning techniques have been developed and worked upon in last few decades for breast cancer detection and classification [[7]-[9]]. It can be divided into three stages viz. preprocessing, feature extraction and, classification. Preprocessing of mammography films helps to improve visibility and intensity distribution of peripheral region. There are several methods to assist preprocessing. Feature extraction is next stage in detection of breast cancer as it helps in differentiating benign tumor from malignant. After this, image properties of breast viz. smoothness, depth, regularity and coarseness are extracted by segmentation [[10]]. The stage of classification is complex optimization problem and many machine learning techniques have

been used by researchers while solving classification problem as the veracity of machine learning technology is promising.

Nowadays, the dependency on machine learning is growing until it will be adapted in services. But, machine learning is yet a field unexplored in many ways with barriers and often required expert knowledge. In the following sections, literature review and a comprehensive explanation to methodology applied to breast cancer detection using python will be given. While in search of best and accurate algorithm classification result, variable quality result of data affects the classification result. So, to test the machine learning technique in dataset, a benchmark dataset of Wisconsin breast cancer diagnosis (WBCD) is usedin application [[11]-[12]]. Also there are other breast cancer benchmark datasets[[13]], for instance in this paper Wisconsin Breast Cancer (Diagnostics) (WBCD)[[14]] is taken in use.

Literature review

Many works have been submitted which attempted to diagnose breast cancer using machine learning algorithms. For instance, Sun at al. in year 2005 [[15]], proposed comparing feature selection methods for a unified detection of breast cancers in mammograms. Another approach, introduced by Malek at al. in year 2009 [[16]], proposed a method using wavelet and proposed a design of automated detection,segmentation, and classification of breast cancer nuclei using a fuzzy logic for feature exaction and classification respectively. Zheg at al. in year 2014 [[17]] combined support vector machine (SVM) and K-means algorithm for breast cancer diagnosis. Aličković and Subasi in year 2017 [[18]] applied a genetic algorithm for feature extraction and rotation for classification. Another approach is conducted by Bannaie in year 2018 [[19]]based on the dynamic contrast-enhanced magnetic resonance imaging (DCE-MRI) technique to attain output of interest.There are several other works performed based on clustering and classification [[20]].AlirezaOsarech, BitaShadgar achieved 98.80% and 96.63% accuracies upon using SVM classification technique on two different benchmark datasets for breast cancer [[21]]. MandeepRana, PoojaChandorkar, AlishibaDsouza applied KNN, SVM, Gaussian Naïve Bayes, and Logistic Regression techiniques programmed in MATLAB to diagnose and predict recurrence of breast cancer. The classification techniques were applied on two dataset from UCI depository. One dataset was used for identification of diseases (WBCD), and other is used for prediction of recurrence [[22]]. VikasChaurasia, BB Tiwari and Saurabh Pal build predictive models on breast cancer and compared their accuracies using famous algorithms viz. J48, Naïve Bayes,and RBF. The results indicated that Naïve Bayes predicted well among them with 97.36% accuracy[[23]]. Haifeng Wang and Sang Won Yoon developed a powerful model for breast cancer prediction by using and comparing Naive Bayes Classifier, Support Vector Machine (SVM), AdaBoost tree and Artificial Neural Networks (ANN). They implemented PCA for dimensionality reduction [[24]]. S.Kharya proposed Artificial Neural Networks (ANN) while working on breast cancer prediction. The paper highlighted advantages of using machine learning methods like SVM, Naive Bayes, Neural network and Decision trees[25]. NareshKhuriwal and Nidhi Mishra used Wisconsin Breast Cancer database to work on breast cancer diagnosis. Based on their experiments they concluded that, ANN and Logistic Algorithm worked better and achieved an accuracy of 98.50% [[26]].

Methodology

The methodology aims to analyse the most helpful feature in prediction of malignant and benign tumor. This may help to visualize general trend in selecting appropriate model. The objective is to classify benign and malignant tumors of breast cancer with the help of python. The focus is on using Logistic Regression, K-Neighbours Classifier, Support Vector Classifier linear (SVC linear), Gaussian Naive Bayes (GaussianNB), and Decision Tree Classifier (DT).

1. **Dataset**

In this paper, Wisconsin Breast Cancer Diagnostics (WBCD) dataset is used which is obtained from UCI Machine Learni0ng Repository **[[**14**]].** It was created by Dr. William H. Wolberg, physician at the University of Wisconsin Hospital at Madison, Wisconsin, USA.The data consists of 569 patients and 32 characteristics. These characteristics formed 32 columns in the dataset. Ten highlights of these characteristics are as per following:

- Radius
- Texture
- Perimeter
- Area
- Smoothness
- Compactness
- Concavity
- Concave points
- Symmetry
- Fractal dimension- the mean, standard error, and worst (mean of the three largest values) of all the features or characteristics.

2. Data exploration and cleaning

In this paper the software used to work on dataset of interest is Jupyter Notebook. To begin with, data exploration and cleaning is done via following steps in Jupyter Notebook:

2.1. Import librabries

Necessary libraries viz**. numpy, pandas**, **matplotlib.pyplot** and **seaborn**are imported.

```
In [1]: #import libraries
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
```

Figure 1 Import libraries

2.2. Load data

Download the data.csv file of breast cancer from https://www.kaggle.com/uciml/breast-cancer-wisconsin-data and read this file via pandas**read_csv** command. Print the first 5 rows of resultant DataFrame. Each row of data in output represents a patient that may or may not have cancer.

```
In [2]: #Load the data
        df = pd.read_csv('data.csv')
        df.head()
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	te
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	

5 rows × 33 columns

Figure 2 First five rows of loaded data

2.3. Count number of rows and columns

Explore and examine the shape of the data by counting the number of rows and columns.

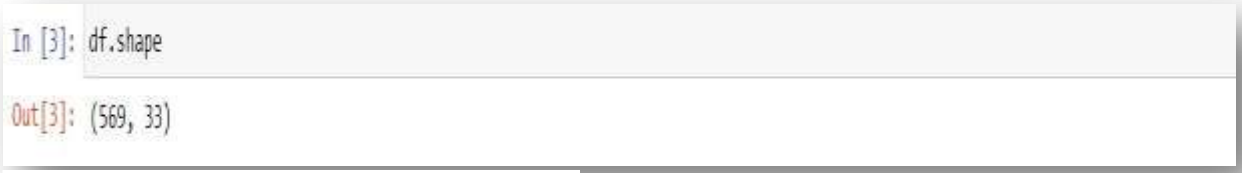

Figure 3 Number of rows and columns

There are 569 rows and 33 columns which represent 569 patients with 33 data points or features for individual patient in this dataset.

2.4. Count of columns containing empty values

Data sets are not perfect and DataFrame from dataset might contain some missing values in any column or row which can mislead the interpretion.

For every missing value,**pandas** add NaN at its place. So, fetch the count of all the column and row from the dataset that contain empty values viz. **NaN, NAN** or **na**.NaN occurs in case of any missing value.

```
In [4]: #Count the empty (NaN, NAN, na) values in each column
        df.isna().sum()

Out[4]: id                        0
        diagnosis                 0
        radius_mean               0
        texture_mean              0
        perimeter_mean            0
        area_mean                 0
        smoothness_mean           0
        compactness_mean          0
        concavity_mean            0
        concave points_mean       0
        symmetry_mean             0
        fractal_dimension_mean    0
        radius_se                 0
        texture_se                0
        perimeter_se              0
        area_se                   0
        smoothness_se             0
        compactness_se            0
        concavity_se              0
        concave points_se         0
        symmetry_se               0
        fractal_dimension_se      0
        radius_worst              0
        texture_worst             0
        perimeter_worst           0
        area_worst                0
        smoothness_worst          0
        compactness_worst         0
        concavity_worst           0
        concave points_worst      0
        symmetry_worst            0
        fractal_dimension_worst   0
        Unnamed: 32             569
        dtype: int64
```

Figure 4 Count of empty values in each column

From the output it can be seen that only the column named 'Unnamed: 32', contains 569 empty values. Thus, the column 'Unnamed: 32' is of no use.

2.5. Dropping column 'Unnamed: 32' and creating new count of dataset

Since the column 'Unnamed: 32' is empty and contains no value, it is removed or dropped from the original dataset by using pandas **dropna** method function.

```
In [5]: df = df.dropna(axis =1 )
```

Figure 5 Function to drop column of empty value

Here **axis = 1** means dropping column containing any missing value.

Number of rows and columns of new DataFrame**df**will be counted by using pandas **shape**.

```
In [6]: df.shape

Out[6]: (569, 32)
```

Figure 6 Number of rows and columns of new DataFrame

The new count of DataFrame contains **569 rows** and **32 columns**.

2.6. Count number of patients with malignant and benign tumor

The number of patients diagnosed with either of malignant **(M)** or benign **(B)** tumor is counted by using pandas**value_counts**function.

```
In [7]: #Get a count of the number of 'M' & 'B' cells
        df['diagnosis'].value_counts()

Out[7]: B    357
        M    212
        Name: diagnosis, dtype: int64
```

Figure 7 Count of 'M' and 'B'

So, the number of patients diagnosed with benign tumor is **357** and those with malignant tumor are **212**.

2.7. Listing columns and their data type

This is done to check the type of data present in the dataset. In case of any categorical data, the data is changed in dummy numeric. This is done because categorical data will mislead the interpretation of data.

The data type of columns is listed by using pandas**dtypes**.

```
In [9]: #Look at the data types
        df.dtypes

Out[9]: id                       int64
        diagnosis               object
        radius_mean            float64
        texture_mean           float64
        perimeter_mean         float64
        area_mean              float64
        smoothness_mean        float64
        compactness_mean       float64
        concavity_mean         float64
        concave points_mean    float64
        symmetry_mean          float64
        fractal_dimension_mean float64
        radius_se              float64
        texture_se             float64
        perimeter_se           float64
        area_se                float64
        smoothness_se          float64
        compactness_se         float64
        concavity_se           float64
        concave points_se      float64
        symmetry_se            float64
        fractal_dimension_se   float64
        radius_worst           float64
        texture_worst          float64
        perimeter_worst        float64
        area_worst             float64
        smoothness_worst       float64
        compactness_worst      float64
        concavity_worst        float64
        concave points_worst   float64
        symmetry_worst         float64
        fractal_dimension_worst float64
        dtype: object
```

Figure 8 Data type of columns

It can be seen in data types that all the features/columns are numerical except 'diagnosis' which is a categorical data and represented as '**object'** in python.

2.8. Encoding categorical data

To make sure that the learning algorithm interprets the malignant and benign tumor correctly, the categorical string values is converted into integers.

The categorical data present in column 'diagnosis' is encoded/ transformed from M and B to **1** and **0** by using **LabelEncoder** from **sklearn.preprocessing.**

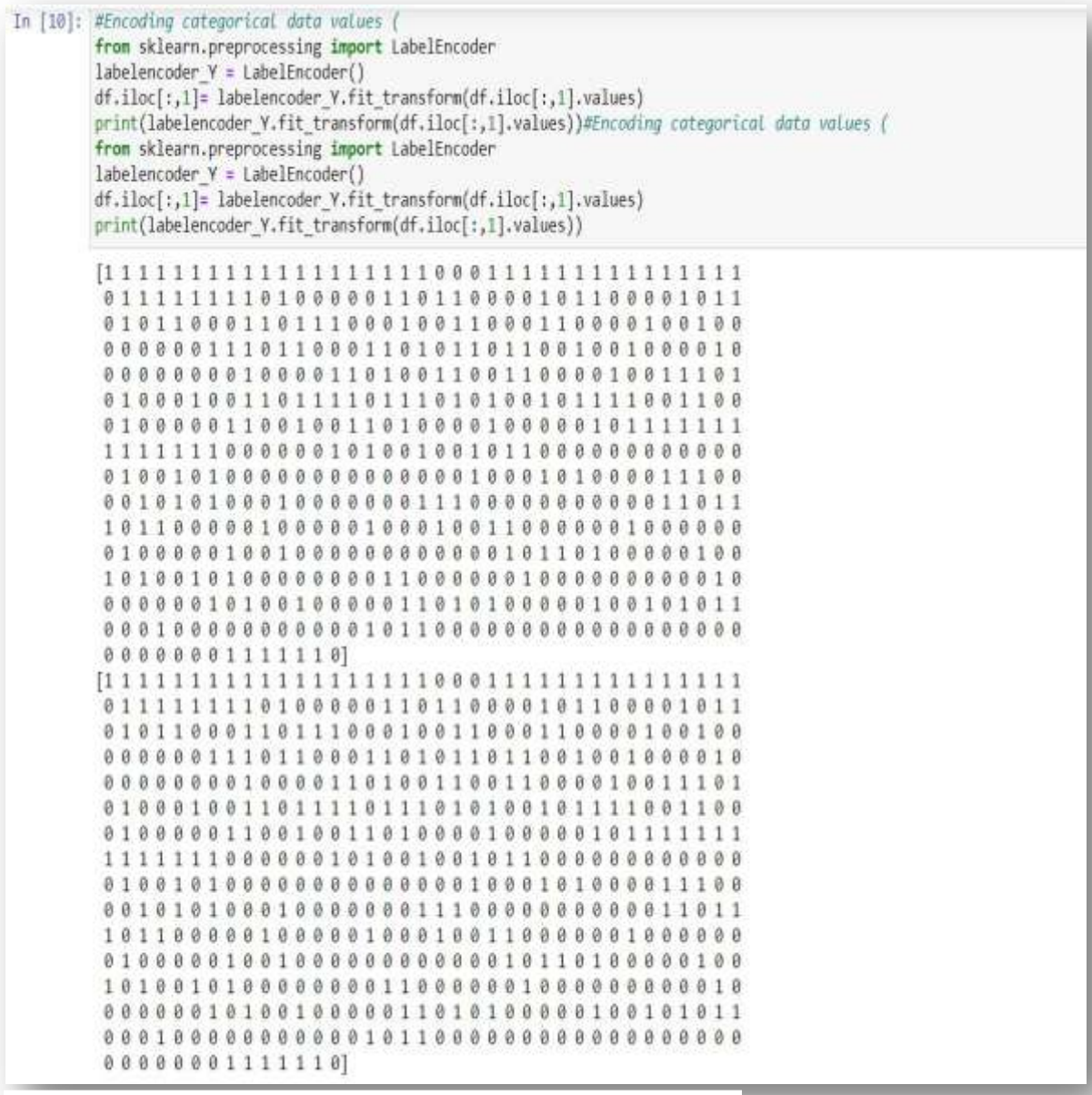

Figure 9 Encoding categorical data to numerical data

3. Data visualization

Data visualization is the discipline to understand data by placing it into visual form in order to interactively and efficiently convey insights so that the patterns, trends and correlations of the data that might not otherwise be detected can be visualized in large data sets. It removes the noise from the data and highlights the useful information.

As visualization makes it easier to detect patterns, trends and outliers, and providesclear, better and reliable result, it is implemented in this paper by creating count plot, pair plot and heat map.

In this work, data visualization is done with the help of **seaborn** library.

3.1. Count Plot

The plot will represent the class distribution of diagnosed malignant and benign patients. Here we have 212 malignant diagnosed patients i.e. around 38% of the data and, 357 i.e. 62% of patients diagnosed with benign tumor.
Count plot visualization of above data in python is done by using **seabornountplot** function.

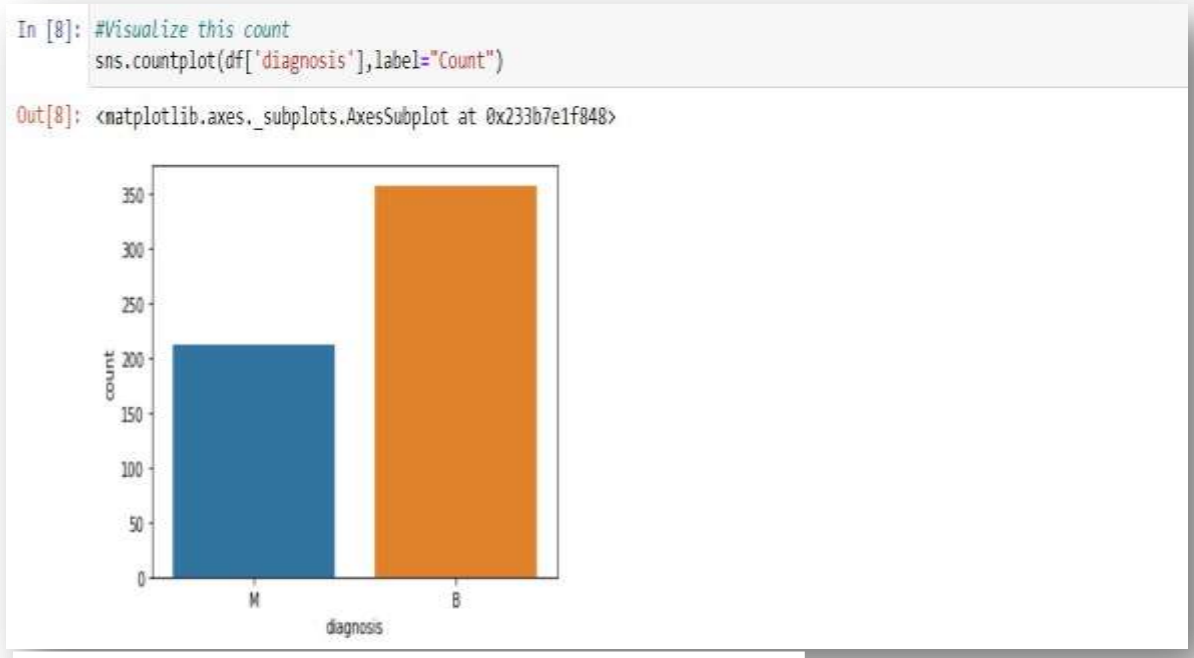

Figure 10 Count Plot for Malignant & Benignpatients

3.2. Pair Plot

As the dataset contain many variables, and relationship between each and every variable is to be analysed,a pair plot is used to visualise the data further. It shows the data as a collection of points. The position of one variable in the same data row is matched with another variable's value. Each value is a position on either the vertical or horizontal dimension indicates its correlation. It allows both, distribution of single variables and relationships between two variables. It is an effective method to identify trends for analysis.
To implement pair plots in python **seaborn** is used and is made by using seaborn**pairplot** function.

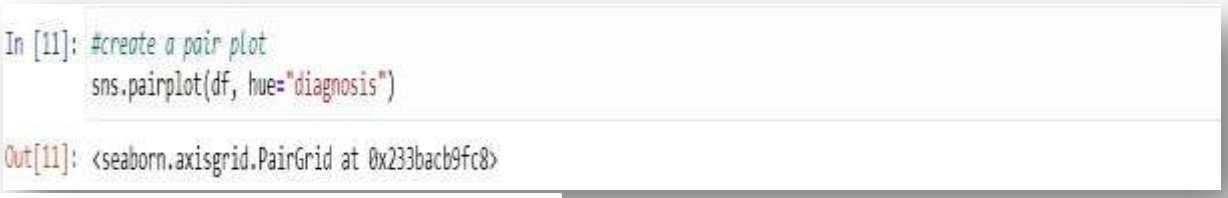

Figure 11.1 Function to create pair plot

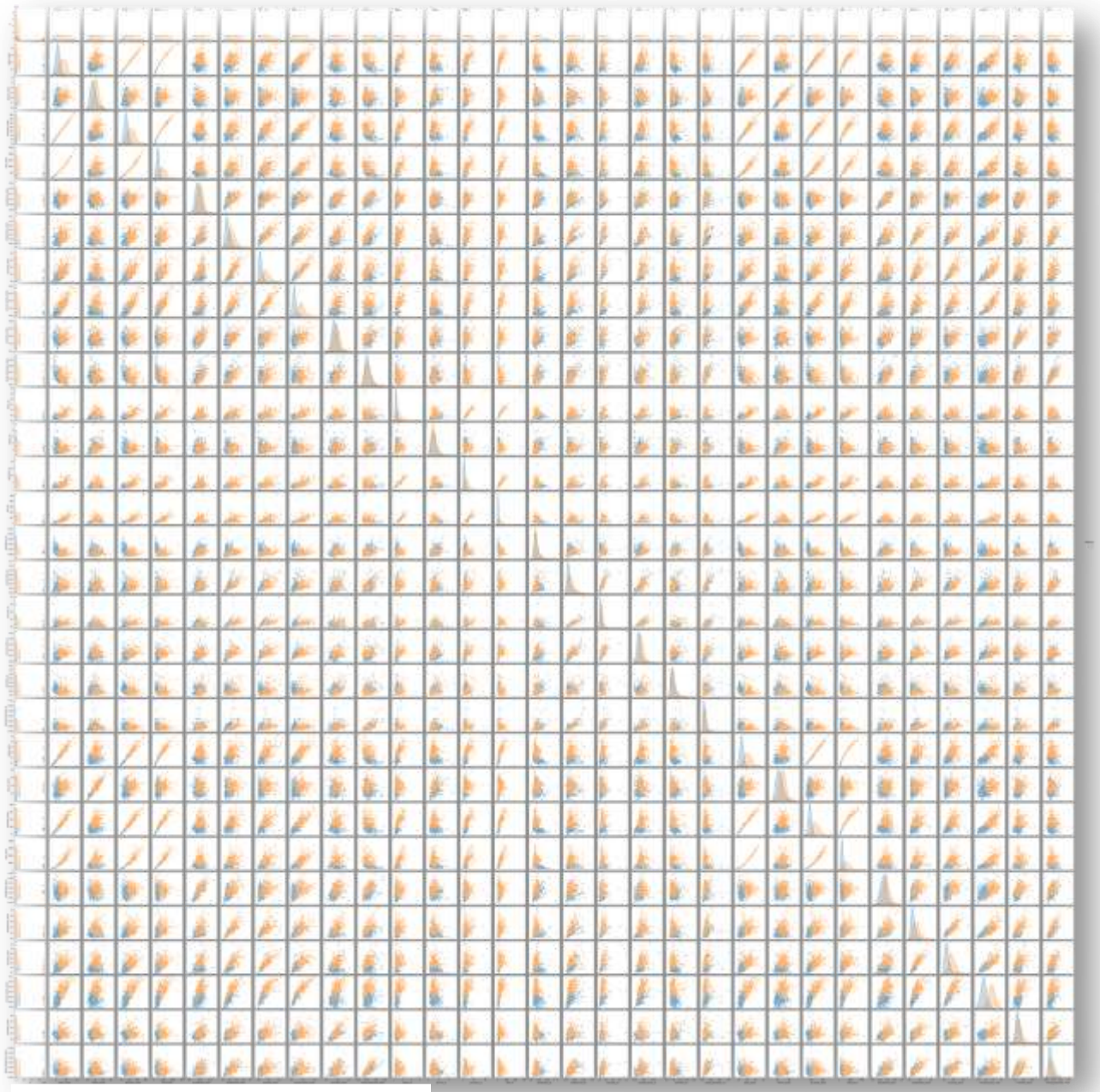

Figure 11.2 Pair Plot

In Figure 11.2, pair plot of all the columns highlighting the diagnosis points is formed. The orange points is for 1 and blue is for 0. Basically, the pair is used to show the numeric distribution in the scatter plot.

3.3. Heat Map

To visualize the data further, print the first five rows of dataset. In python this is done by using pandas **head** function.

In [12]: df.head(5)

Out[12]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
0	842302	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001
1	842517	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869
2	84300903	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974
3	84348301	1	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414
4	84358402	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980

5 rows × 32 columns

Figure 12 First five rows of DataFrame

Following this correlation of columns is performed by using pandas**corr** function.

In [13]:
```
#Get the correlation of the columns
df.corr().head()
```

Out[13]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
id	1.000000	0.039769	0.074626	0.099770	0.073159	0.096893	-0.012968	0.000096	0.050080
diagnosis	0.039769	1.000000	0.730029	0.415185	0.742636	0.708984	0.358560	0.596534	0.696360
radius_mean	0.074626	0.730029	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764
texture_mean	0.099770	0.415185	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418
perimeter_mean	0.073159	0.742636	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136

5 rows × 32 columns

Figure 13 Correlation of columns

From figure 13, correlation analysis of columns in reference to pair plot is displayed which is helpful to analyse the relationship among the features in individual patients.

The visualization of correlation is more reliable and easier via heatmap. A heatmap is a two dimentional illustration of data by colors. The use of colors makes the visualization better as it might be tough to grasp a data if presented numerically. Heatmap provides a direct visual outline of data. So, to find the correlation between each feature and diagnosis heatmap is vizualized using correlation matrix.

In python heatmap is formed by using seaborn**heatmap** function.

```
In [14]: plt.figure(figsize=(20,20))
         sns.heatmap(df.corr(), annot=True, fmt='.0%')

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1fd76e7ec88>
```

Figure 14.1 Function to make Heat Map

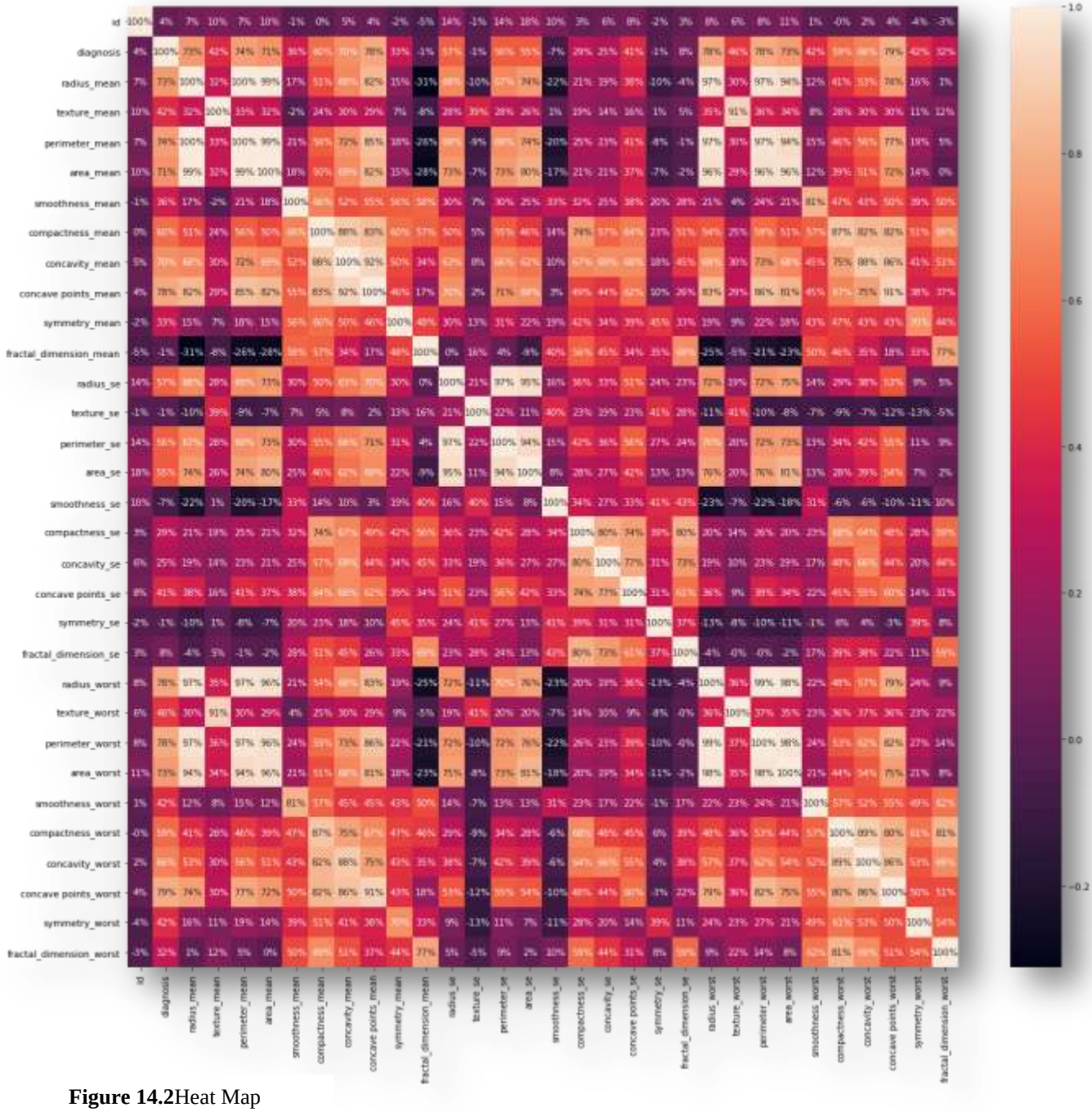

Figure 14.2 Heat Map

Looking at the heatmap, the focus is on the light and the dark areas. It shows the strengthof correlation. The lightarea representshigh correlationwhile dark area represents weak correlation among the attributes/characteristics. The added annotation in heatmap, which are the actual correlation values, enables to easily form a conclusion.

4. Model selection

Model selection is the process of choosing different machine learning algorithms. More than one kind of machine learning techniques can be used. The algorithms are classified in two major groups:

1. **Supervised learning**: Method in which machine is trained on data in which the input and output are labelled. The model can learn the training data and process the future data to predict outcome. It is again divided into two groups viz.
   Regression: It is used when the result is continuous or real.
   Classification: It is used when the result is a category.

2. **Unsupervised learning**: Method in which the machine is trained from the unlabelled or unclassified data making the algorithm work without providing any directions.

In our dataset of breast cancer, the dependent variable has two set of values, either malign (M), or Benign (B). Thus, the classification algorithm of supervised learning is used. The different types of classification algorithm used in this model are:

- Logistic Regression
- K Neighbour classifier
- SVC linear
- Gaussian Naïve Bayes
- Decision Tree classifier

The data is set up for model as per following:

4.1. Split data set

The first split is splitting data into a feature data set, also known as independent data set (X), and a target data set also known as the dependent data set (Y).

In python it is done by using **slice** operation. In X variable, values of all rows and column from 3 -31is assigned. And in Y variable, all rows but only second column is selected and assigned.

Figure 15 Function to split independent and dependent variable

Now, split the data again but this time into 75% training and 25% testing data sets. In machine learning, training data is used to fit the model and testing data to test it. The models generated are to predict unknown results i.e. test set. Thus the dataset is divided into train and test set in order to check accuracies and precisions by training and testing it on it.

This has done in python by using **train_test_split** from **sklearn.model_selection**.

```
In [16]: from sklearn.model_selection import train_test_split
         X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25, random_state = 0)
```

Figure 16 Splitting of 75% training set and 25% test set from DataFrame

4.2. Feature scaling

At times, features in dataset vary in magnitude, units and range. But as most of the machine learning algorithms use eucledian distance between two data points, it is important to bring all features to same level of magnitude with the help of scaling. This will result the feature/ independent data within a specific range. For example, 0-100 or 0-1.

In python this is done by using **StandardScaler** function from **sklearn.preprocessing**.

```
In [17]: #Feature Scaling
         from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         X_train = sc.fit_transform(X_train)
         X_test = sc.transform(X_test)
```

Figure 17 Feature scaling

4.3. Create function to hold different models

Function is created to hold all the models used in dataset to make classification. In python the algorithm is applied by **sklearn** library to import all the methods of classification algorithms.

After importing library, and then Logistic Regression Algorithm is used to the Training Set,KNeighbors classifier Method of neighbors class to use Nearest Neighbor algorithm, SVC linear method of svm class to use Support Vector Machine Algorithm,GaussianNB method of naïve_bayes class to use Naïve Bayes Algorithm, and DecisionTree classifier of tree class to use Decision Tree Algorithm.Within this function, accuracy of each model on the training data is also printed.

```
In [27]: def models(X_train,Y_train):
           #Using Logistic Regression
           from sklearn.linear_model import LogisticRegression
           log = LogisticRegression(random_state = 0)
           log.fit(X_train, Y_train)
           #Using KNeighborsClassifier
           from sklearn.neighbors import KNeighborsClassifier
           knn = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
           knn.fit(X_train, Y_train)
           #Using SVC linear
           from sklearn.svm import SVC
           svc_lin = SVC(kernel = 'linear', random_state = 0)
           svc_lin.fit(X_train, Y_train)
           #Using GaussianNB
           from sklearn.naive_bayes import GaussianNB
           gauss = GaussianNB()
           gauss.fit(X_train, Y_train)
           #Using DecisionTreeClassifier
           from sklearn.tree import DecisionTreeClassifier
           tree = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
           tree.fit(X_train, Y_train)
           #Using RandomForestClassifier method of ensemble class to use Random Forest Classification algorithm
           from sklearn.ensemble import RandomForestClassifier
           #print model accuracy on the training data.
           print('[0]Logistic Regression Training Accuracy:', log.score(X_train, Y_train))
           print('[1]K Nearest Neighbor Training Accuracy:', knn.score(X_train, Y_train))
           print('[2]Support Vector Machine (Linear Classifier) Training Accuracy:', svc_lin.score(X_train, Y_train))
           print('[3]Gaussian Naive Bayes Training Accuracy:', gauss.score(X_train, Y_train))
           print('[4]Decision Tree Classifier Training Accuracy:', tree.score(X_train, Y_train))
           return log, knn, svc_lin, gauss, tree
```

Figure 18 Function to build models and its accuracies

4.4. Create model

Create model that contains all the models and observe the accuracy score on the training data for each model. This is done to classify and detect if the patient has cancer or not.

```
In [28]: model = models(X_train,Y_train)

[0]Logistic Regression Training Accuracy: 0.9906103286384976
[1]K Nearest Neighbor Training Accuracy: 0.9765258215962441
[2]Support Vector Machine (Linear Classifier) Training Accuracy: 0.9882629107981221
[3]Gaussian Naive Bayes Training Accuracy: 0.9507042253521126
[4]Decision Tree Classifier Training Accuracy: 1.0
```

Figure 19 Accuracies of training set model

To test the model, accuracy of testing data is used. To check the accuracy, import **confusion_matrix** method of metric class. It summarises the performance of a classification algorithm. Confusion matrix calculates and

provides what classification model is getting right and what types of errors it is making. The numbers of correct and incorrect predictions are summarized with count values and are broken down by each class. It provides insights to error being made by classifier and the types of error that are being made.

```
In [29]: from sklearn.metrics import confusion_matrix
         for i in range(len(model)):
           cm = confusion_matrix(Y_test, model[i].predict(X_test))

           TN = cm[0][0]
           TP = cm[1][1]
           FN = cm[1][0]
           FP = cm[0][1]

           print(cm)
           print('Model[{}] Testing Accuracy = "{}!"'.format(i,  (TP + TN) / (TP + TN + FN + FP)))
           print()# Print a new line

[[86  4]
 [ 4 49]]
Model[0] Testing Accuracy = "0.9440559440559441!"

[[89  1]
 [ 5 48]]
Model[1] Testing Accuracy = "0.958041958041958!"

[[87  3]
 [ 2 51]]
Model[2] Testing Accuracy = "0.965034965034965!"

[[85  5]
 [ 6 47]]
Model[3] Testing Accuracy = "0.9230769230769231!"

[[84  6]
 [ 1 52]]
Model[4] Testing Accuracy = "0.951048951048951!"
```

Figure 20 Confusion matrix of test set

Confusion matrix accuracy of each matrix is displayed.Also, here in case of breast cancer, **type II** error is more dangerous error and would cause greater consequence. Thus, considering both type I and II errorsin each model with respect to its accuracies, model 2 i.e. SVM linear is comparatively more accurate than other mentioned model for breast cancer detection.

4.5. Cross validation

Since our model does not have huge data set, testing set is left with few observations to lead any real conclusion. So, cross validation is performed on the data set to make predictions on all data. Also, to check if the machine learning model is over fitted, generalized or under fitted, cross validation of model is performed. If the model is under fit, cross validation will have high training and testing error and if the model is an overfit, cross validation will have extremely low training error but a high testing error. Over fitting and under fitting is the common error that occurs in selection of a model. Thus, to overcome this, cross validation of 10 fold is performed to validate the result.

```
In [30]: from sklearn.model_selection import cross_val_score
         cross_validation0 = cross_val_score(model[0],  X= X_train, y = Y_train, cv= 10)
         cross_validation1 = cross_val_score(model[1],  X= X_train, y = Y_train, cv= 10)
         cross_validation2 = cross_val_score(model[2],  X= X_train, y = Y_train, cv= 10)
         cross_validation3 = cross_val_score(model[3],  X= X_train, y = Y_train, cv= 10)
         cross_validation4 = cross_val_score(model[4],  X= X_train, y = Y_train, cv= 10)

         cross_validation0.mean(), cross_validation1.mean(), cross_validation2.mean(), cross_validation3.mean(),cross_validation4.mean()

Out[30]: (0.9860465116279069,
          0.9648394241417497,
          0.9719269102990034,
          0.9437984496124031,
          0.9270764119601329)
```

Figure 21 Cross validation

Result and Discussion

The result of cross validation of each model is compared against training and testing set. Taking confusion matrix into consideration and analysing the accuracies, it is observed that although,SVM linear model is slightly imbalanced with cross validation 97.19%, training set 98.83% and testing 96.50%, but this can be generalized by changing and modifying the training and testing set. The number of observations in training and testing set also has significant effect on the accuracy of data. Thus, with performance metric of **97.19%**, SVM linear is comparatively more accurate than Logistic Regression, KNN, GaussianNB, and Decision Tree in detecting breast cancer.

Accuracy check of SVC linear model with actual values

As SVC linear is comparatively more accurate for detection of breast cancer, the accuracy of model is analysed with the actual values of breast cancer diagnosis. This will help in highlighting the result of detection with percentage of error more clearly.

```
In [22]: #Print Prediction of SVC linear model
         pred = model[2].predict(X_test)
         print(pred)

         #Print a space
         print()

         #Print the actual values
         print(Y_test)

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

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

Figure 22 Accuracy check

From this accuracy check it can be analysed that, the model SVC linear with performance metric of approximately 97.19% can predict and diagnose if a patient has cancer or not.

Conclusion

This study attempts to analyse various supervised machine learning algorithms and select the most accurate model in detection of breast cancer. The work focused in advancement of predictive models with the help of python to achieve better accuracy in predicting correct outcomes. The analysis of result signifies that, integration of data, feature scaling along with different classification method and analysis provide markedly successful tool in prediction. It has also observed that the model misdiagnosed few patients with cancer when they were not having cancer and vice versa. Although, the model is accurate but when dealing with lives of people, further research in building the most accurate and precise model must be carried out for better performance of classification techniques and get the accuracy as close to 100% as possible. Thus, the tuning of each of the models is necessary with the building of more reliable model.

References

[1] http:// www.imaginis.com/breasthealth/breast_cancer.asp, Last Accessed August 2007.

[2] West, D., Mangiameli, P., Rampal, R., & West, V. (2005). Ensemble strategies for a medical diagnosis decision support system: A breast cancer diagnosis application. European Journal of Operational Research(162), 532–551

[3] https://www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470

[4] Pavlopoulos, S.A.; Delopoulos, A.N. Designing and implementing the transition to a fully digital hospital. IEEE Trans. Inf. Technol. Biomed. 1999, 3, 6–19.

[5] Barracliffe, L.; Arandjelović, O.; Humphris, G. A pilot study of breast cancer patients: Can machine learning predict healthcare professionals' responses to patient emotions? In Proceedings of the International Conference on Bioinformatics and Computational Biology, Honolulu, HI, USA, 20–22 March 2017; pp. 101–106.

[6] Birkett, C.; Arandjelović, O.; Humphris, G. Towards objective and reproducible study of patient-doctor interaction: Automatic text analysis based VR-CoDES annotation of consultation transcripts. In Proceedings of the IEEE Engineering in Medicine and Biology Society Conference, Jeju Island, Korea, 11–15 July 2017; pp. 2638–2641.

[7] A. J. Cruz and D. S. Wishart, "Applications of machine learning in cancer prediction and prognosis," Cancer Informatics, vol. 2, pp. 59–77, 2006.

[8] G. Valvano, G. Santini, N. Martini et al., "Convolutional neural networks for the segmentation of microcalcification in mammography imaging," Journal of Healthcare Engineering, vol. 2019, Article ID 9360941, 9 pages, 2019.

[9] M. F. Akay, "Support vector machines combined with feature selection for breast cancer diagnosis," Expert Systems with Applications, vol. 36, no. 2, pp. 3240–3247, 2009.

[10] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," IEEE Transactions on Image Processing, vol. 9, no. 4, pp. 561–576, 2000.

[11] Mangasarian, O.L.; Setiono, R.; Wolberg, W.H. Pattern recognition via linear programming: Theory and application to medical diagnosis. In Large-Scale Numerical Optimization; SIAM: Philadelphia, PA, USA, 1990; pp. 22–31.

[12] Wolberg, W.H.; Mangasarian, O.L. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc. Natl. Acad. Sci. USA 1990, 87, 9193–9196.

[13] Sharma, A.; Kulshrestha, S.; Daniel, S. Machine learning approaches for breast cancer diagnosis and prognosis. In Proceedings of the International Conference on Soft Computing and Its Engineering Applications, Changa, India, 1–2 December 2017.

[14] https://www.kaggle.com/uciml/breast-cancer-wisconsin-data

[15] Y. Sun, C. F. Babbs, and E. J. Delp, "A comparison of feature selection methods for the etection of breast cancers in mammograms: adaptive sequential floating search vs. genetic algorithm," in *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pp. 6532–6535, Shanghai, China, September 2005.

[16] J. Malek, A. Sebri, S. Mabrouk, K. Torki, and R. Tourki, "Automated breast cancer diagnosis based on GVF-snake segmentation, wavelet features extraction and fuzzy classification," *Journal of Signal Processing Systems*, vol. 55, no. 1–3, pp. 49–66, 2009.

[17] B. Zheng, S. W. Yoon, and S. S. Lam, "Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1476–1482, 2014.

[18] E. Aličković and A. Subasi, "Breast cancer diagnosis using GA feature selection and Rotation Forest," *Neural Computing and Applications*, vol. 28, no. 4, pp. 753–763, 2017.

[19] M. Banaie, H. Soltanian-Zadeh, H.-R. Saligheh-Rad, and M. Gity, "Spatiotemporal features of DCE-MRI for breast cancer diagnosis," *Computer Methods and Programs in Biomedicine*, vol. 155, pp. 153–164, 2018.

[20] M. F. Akay, "Support vector machines combined with feature selection for breast cancer diagnosis," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3240–3247, 2009.

[21] AlirezaOsarech, BitaShadgar, "A Computer Aided Diagnosis System for Breast Cancer", International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.

[22] MandeepRana, PoojaChandorkar and AlishibaDsouza, "Breast cancer diagnosis and recurrence prediction using machine learning techniques", International Journal of Research in Engineering and Technology Volume 04, Issue 04, April 2015.

[23] VikasChaurasia, BB Tiwari and Saurabh Pal, "Prediction of benign and malignant breast cancer using data mining techniques", Journal of Algorithms and Computational Technology.

[24] Haifeng Wang and Sang Won Yoon, Breast Cancer Prediction using Data Mining Method, IEEE Conference paper.

[25] D.Dubey, S.Kharya and S.Soni, "Predictive Machine Learning techniques for Breast Cancer Detection", International Journal of Computer Science and Information Technologies, Vol.4 (6), 2013, 1023-1028.

[26] Nidhi Mishra, NareshKhuriwal, "Breast cancer diagnosis using adaptive voting ensemble machine learning algorithm", 2018 IEEMA Engineer Infinite Conference (eTechNxT), 2018.

Annexure

[1] The benchmark data of interest for detection of breast cancer can be referred fromhttps://www.kaggle.com/uciml/breast-cancer-wisconsin-data

[2] For more reference, the program of this paper is uploaded in GitHub linkhttps://github.com/neha-2568/Detection-of-Breast-Cancer

[3] The more clear vision of Pair Plot and Heat Map is available in GitHub link https://github.com/neha-2568/Detection-of-Breast-Cancer