BREAST CANCER DETECTION USING MACHINE LEARNING

Project Report Submitted in partial fulfillment of the requirements for the degree of

Master of Computer Applications
Computer Science and Engineering

Submitted by

Saraswati Tiwari (Roll No. 2021PGCACA100)

Under the esteemed Supervision of
Dr. Chandrashekhar Azad
Assistant Professor

Department of Computer Science and Engineering
National Institute of Technology Jamshedpur

May 2024

TABLE OF CONTENTS

Dept of Computer Science & Engineering
National Institute of Technology Jamshedpur
(An Institution of National importance under MHRD, Government of India)

Ref.no: ……. Date: …….

TO WHOM IT MAY CONCERN

Certificate of Project Work

This is to certify that the project work titled “BREAST CANCER DETECTION USING MACHINE LEARNING” is a research work carried out by Saraswati Tiwari bearing institute registration no. 2021PGCACA100, a student of 6th semester, Master of Computer Applications (MCA), under the department of Computer Science & Engineering, National Institute of Technology Jamshedpur. This work has been carried out under my supervision from January to May. This project report is submitted in the partial fulfillment of the requirement for the award of the degree of Master of Computer Application (MCA) and has been carried out under my joint supervision.

I wish all the best in her career and future endeavors.

Dr. Chandrashekhar Azad
Assistant Professor
Department of Computer Science & Engineering

NIT Jamshedpur

DECLARATION

I certify that the work contained in this report is original and has been done by me under the guidance of my supervisor(s). The work has not been submitted to any other Institute for any degree. I have followed the guidelines provided by the institute in preparing the report. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Saraswati Tiwari

Date:……

2021PGCACA100

MCA 6th Semester

Department of Computer Science & Engineering

NIT Jamshedpur

ACKNOWLEDGMENT

Here I gladly present my project report on “Breast cancer detection using Machine learning” as a part of the 6th semester MCA, Masters of Computer Applications. I thank almighty god for the endless blessings. An honorable mention goes to my guide, **Dr. C. Azad**.

Sincere thanks to all my colleagues for their support throughout the project, creating a wonderful environment where I received assistance at every step.

My heartfelt gratitude also extends to the **H.O.D. C.S.E., Prof. D.A Khan** of the National Institute of Technology, Jamshedpur, for allowing me to use the excellent facilities and infrastructure.

I equally appreciate **Dr. Alekha Kumar Mishra** and the entire MCA faculty for their guidance and support during the project's development. Acknowledging my family and friends' contributions, I must say their love and blessings keep me motivated.

I apologize for any errors or omissions in this work, which are solely my responsibility.

Lastly, I am grateful for this wonderful opportunity.

Place : ……

Date:……

Saraswati Tiwari

2021PGCACA100

MCA 6th Semester

Department of Computer Science & Engineering

NIT Jamshedpur

ABSTRACT

Breast cancer remains one of the most prevalent and lethal forms of cancer among women worldwide. Early detection is pivotal for effective treatment and improved survival rates.

In recent years, the integration of machine learning (ML) techniques has shown promising results in enhancing the accuracy and efficiency of breast cancer detection. Considering the available information, this report provides a comprehensive review of the current state-of-the-art ML methodologies employed in breast cancer detection.

This research review begins by outlining the importance of early detection in improving survival rates and reducing the burden of breast cancer. I have then discussed various ML algorithms, including logistic regression, support vector machines, decision trees, naive Bayes, k-nearest neighbors, as well as deep learning architectures such as neural networks and convolutional neural networks, highlighting their respective strengths and applications in breast cancer detection.

Chapter 1. INTRODUCTION

Breast cancer is a global health challenge impacting people of all ages and backgrounds, with it ranking among the top kinds of cancer affecting women. Detecting and predicting it early is critical for effective treatment.

Breast cancer accounts for 1 in 3 of new female cancers annually. In 2023, an estimated 297,790 women in the United States were diagnosed with invasive breast cancer, and 55,720 women were diagnosed with non-invasive(in situ) breast cancer.

1.1 About

"In situ" in the context of breast cancer refers to a stage where abnormal cells are confined within the breast ducts or lobules and have not invaded surrounding breast tissue or spread to other parts of the body.

There are two main types of breast cancer in situ:

Ductal Carcinoma In Situ (DCIS): This is a non-invasive or pre-invasive form of breast cancer that originates in the milk ducts of the breast. In DCIS, abnormal cells are confined to the ductal system and have not spread beyond the duct walls into surrounding tissue. While DCIS is not considered invasive cancer, if left untreated, it can progress to invasive breast cancer over time.

Lobular Carcinoma In Situ (LCIS): LCIS is a condition where abnormal cells are found in the lobules or milk-producing glands of the breast. Similar to DCIS, LCIS is considered a risk factor for developing invasive breast cancer rather than a true cancer itself. Women with LCIS have an increased risk of developing invasive breast cancer in either breast over time.

Breast cancer is a global health challenge impacting people of all ages and backgrounds, with it ranking among the top kinds of cancer affecting women. Detecting and predicting it early is critical for effective treatment.

1.2 Key facts

- Breast cancer caused 670,000 deaths globally in 2022.
- Roughly half of all breast cancers occur in women with no specific risk factors other than sex and age.
- Breast cancer was the most common cancer in women in 157 countries out of 185 in 2022.
- Breast cancer occurs in every country in the world.
- Approximately 0.5–1% of breast cancers occur in men. [1]

1.3 Scope of the problem

In 2022, there were 2.3 million women diagnosed with breast cancer and 670 000 deaths globally. Breast cancer occurs in every country of the world in women at any age after puberty but with increasing rates in later life.

Global estimates reveal striking inequities in the breast cancer burden according to human development. For instance, in countries with a very high Human Development Index (HDI), 1 in 12 women will be diagnosed with breast cancer in their lifetime and 1 in 71 women die of it.

In contrast, in countries with a low HDI; while only 1 in 27 women is diagnosed with breast cancer in their lifetime, 1 in 48 women will die from it. [1]

1.4 Signs and symptoms

Various factors that contribute to breast cancer risk include age, family history, genetic mutations, and hormone replacement therapy.

Most people will not experience any symptoms when the cancer is still early hence the importance of early detection.

Breast cancer can have combinations of symptoms, especially when it is more advanced. Symptoms of breast cancer can include:

- a breast lump or thickening, often without pain
- change in size, shape or appearance of the breast
- dimpling, redness, pitting or other changes in the skin
- change in nipple appearance or the skin surrounding the nipple (areola)
- abnormal or bloody fluid from the nipple.

People with an abnormal breast lump should seek medical care, even if the lump does not hurt.

Most breast lumps are not cancerous. Breast lumps that are cancerous are more likely to be successfully treated when they are small and have not spread to nearby lymph nodes.

Breast cancers may spread to other areas of the body and trigger other symptoms. Often, the most common first detectable site of spread is to the lymph nodes under the arm although it is possible to have cancer-bearing lymph nodes that cannot be felt.

Over time, cancerous cells may spread to other organs including the lungs, liver, brain and bones. Once they reach these sites, new cancer-related symptoms such as bone pain or headaches may appear. [1]

1.5 Treatment [1]

Treatment for breast cancer depends on the subtype of cancer and how much it has spread outside of the breast to lymph nodes (stages II or III) or to other parts of the body (stage IV).

Doctors combine treatments to minimize the chances of the cancer coming back (recurrence). These include:

- surgery to remove the breast tumor
- radiation therapy to reduce recurrence risk in the breast and surrounding tissues
- medications to kill cancer cells and prevent spread, including hormonal therapies, chemotherapy or targeted biological therapies.

Treatments for breast cancer are more effective and are better tolerated when started early and taken to completion.

Surgery may remove just the cancerous tissue (called a lumpectomy) or the whole breast (mastectomy). Surgery may also remove lymph nodes to assess the cancer's ability to spread.

Radiation therapy treats residual microscopic cancers left behind in the breast tissue and/or lymph nodes and minimizes the chances of cancer recurring on the chest wall.

Advanced cancers can erode through the skin to cause open sores (ulceration) but are not necessarily painful. Women with breast wounds that do not heal should seek medical care to have a biopsy performed.

Medicines to treat breast cancers are selected based on the biological properties of the cancer as determined by special tests (tumor marker determination). The great majority of drugs used for breast cancer are already on the WHO Essential Medicines List (EML).

Lymph nodes are removed at the time of cancer surgery for invasive cancers. Complete removal of the lymph node bed under the arm (complete axillary dissection) in the past was thought to be necessary to prevent the spread of cancer. A smaller lymph node procedure called "sentinel node biopsy" is now preferred as it has fewer complications.

Medical treatments for breast cancers, which may be given before ("neoadjuvant") or after ("adjuvant") surgery, is based on the biological subtyping of the cancers. Certain subtypes of breast cancer are more aggressive than others such as triple negative (those that do not express estrogen receptor (ER), progesterone receptor (PR) or HER-2 receptor). Cancer that express the estrogen receptor (ER) and/or progesterone receptor (PR) are likely to respond to endocrine (hormone) therapies such as tamoxifen or aromatase inhibitors. These medicines are taken orally for 5–10 years and reduce the chance of recurrence of these "hormone-positive" cancers by nearly half. Endocrine therapies can cause symptoms of menopause but are generally well tolerated.

Cancers that do not express ER or PR are "hormone receptor negative" and need to be treated with chemotherapy unless the cancer is very small. The chemotherapy regimens available today

are very effective in reducing the chances of cancer spread or recurrence and are generally given as outpatient therapy. Chemotherapy for breast cancer generally does not require hospital admission in the absence of complications.

Breast cancers that independently overexpress a molecule called the HER-2/neu oncogene (HER-2 positive) are amenable to treatment with targeted biological agents such as trastuzumab. When targeted biological therapies are given, they are combined with chemotherapy to make them effective at killing cancer cells.

Radiotherapy plays a very important role in treating breast cancer. With early-stage breast cancers, radiation can prevent a woman having to undergo a mastectomy. With later stage cancers, radiotherapy can reduce cancer recurrence risk even when a mastectomy has been performed. For advanced stages of breast cancer, in some circumstances, radiation therapy may reduce the likelihood of dying of the disease.

The effectiveness of breast cancer therapies depends on the full course of treatment. Partial treatment is less likely to lead to a positive outcome. [1]

Chapter 2. Literature Review

Recent diagnostic and treatment advancements have improved prognosis, particularly for ductal carcinoma in situ (DCIS), which remains localized without spreading to other organs. However, screening for early detection faces challenges, including limitations of mammography, such as sensitivity issues in dense breast tissue and risks of overdiagnosis and overtreatment. Magnetic resonance imaging (MRI) supplements mammography but poses interpretation challenges, mitigated by multi-parametric MRI methods incorporating T2-weighted sequences.
Emerging technologies like Machine Learning(ML), Neural Networks(NN), Convolutional Neural Networks(CNNs) and Artificial Intelligence(AI) show promise in enhancing image processing and analysis, aiding in early detection efforts.

Understanding breast cancer subtypes, diagnostic challenges, and technological advancements is crucial for improving screening accuracy and treatment outcomes. By addressing these complexities, research in machine learning and imaging technologies can significantly impact early detection efforts, ultimately improving patient care and outcomes.

Mammography, conducted by radiologists, is the primary method for detecting breast cancer.

- The breast consists of lobules, connective tissue, and ducts, with cancer typically originating in the ducts or lobules.
- Signs of breast cancer include lumps, changes in breast size or shape, skin dimpling, redness, nipple changes, and abnormal discharge.
- Breast cancer tumors are categorized as benign or malignant, with malignant tumors having the potential to invade surrounding tissues.
- ML algorithms analyze various features to predict breast cancer diagnoses, utilizing the performance of each classifier to optimize outcomes.

2.1 OBJECTIVES:

Through this research paper, I aim to bring together current research findings and methodologies to shed light on how machine learning can be applied to enhance breast cancer detection. By synthesizing this information, I hope to contribute to our collective understanding of this critical area, helping both researchers and clinicians stay up-to-date with the latest advancements in the field.

Improving Diagnosis and Treatment: The insights presented in my paper have the potential to make a significant impact on the accuracy and efficiency of breast cancer diagnosis. By leveraging machine learning techniques, we can potentially streamline the diagnostic process, leading to earlier detection and initiation of treatment. Ultimately, this could translate into better outcomes for patients.

Reducing Mortality Rates: Early detection is paramount in reducing mortality rates associated with breast cancer. By highlighting the role of machine learning in facilitating early detection, my research paper aims to raise awareness and promote the adoption of advanced diagnostic methods. By doing so, we can improve survival rates and save lives.

Informing Clinical Practice: My paper provides valuable insights into the practical application of machine learning techniques in clinical settings. By addressing the strengths, limitations, and challenges associated with these methods, I hope to empower clinicians to make informed decisions about incorporating machine learning tools into their practice. This knowledge can enhance patient care and improve treatment outcomes.

2.2 Existing diagnostic methods, advantages, and limitations [2]:

Method	Advantages	Limitations
Mammography	Well-established	Limited sensitivity in dense breast tissue
	Widely accessible	False positives/negatives
	Detects structural changes and calcifications	False positives/negatives
Ultrasound	No radiation	Limited specificity
	Useful for dense breasts	Operator-dependent
	Differentiates cysts from solid masses	Limited detection in deep tissues
MRI (Magnetic Resonance Imaging)	High sensitivity	High cost
	No radiation	Longer exam duration
	Detailed soft tissue visualization	Requires specialized expertise to detect benign lesions
Biopsy (Fine Needle Aspiration or Core Needle Biopsy)	Provides tissue samples for definitive diagnosis	Invasive and uncomfortable
	High diagnostic accuracy	Small risk of complications
		Requires skilled medical staff
		Sample may not be representative
Clinical Breast Examination (CBE)	No radiation	Limited sensitivity
	Low cost	Dependent on examiner's expertise
	Can detect palpable masses	May miss non-palpable masses
Genetic Testing (BRCA1/BRCA2 Testing)	Identifies genetic mutations linked to increased risk	Applicable to specific subsets of patients

	Enables targeted prevention and treatment strategies	Limited to hereditary breast cancer cases

Fig 2.1 (A table depicting various methods, advantages and their limitations in detecting breast cancer cells)

Mammography is yet the best available technique for breast cancer screening, recognized and recommended in many countries as a cornerstone of early detection efforts. It involves the use of X-rays to generate detailed images of the breast tissue, allowing radiologists to identify abnormalities that may indicate the presence of cancer. Mammography's effectiveness in detecting breast cancer, especially in its early stages, has been extensively studied and validated.

In cases where mammography may not provide sufficient clarity, such as in women with dense breast tissue, **ultrasound** serves as a valuable complementary tool. Ultrasound uses sound waves to create images of the breast, offering additional information that can aid in the detection and characterization of abnormalities. It is particularly useful for evaluating lumps or masses that may be obscured on mammograms due to breast density.

While **thermography**, which detects heat patterns in the breast, has been explored as a potential screening tool, it is not recommended as a primary method for breast cancer screening. Concerns regarding its sensitivity and specificity, as well as the lack of robust evidence supporting its effectiveness, have led to its exclusion from standard screening protocols.

The choice of screening modality is highly individualized and depends on various factors, including the patient's age, risk factors, breast density, and clinical circumstances. Healthcare providers, in consultation with patients, determine the most appropriate screening approach tailored to each individual's needs and preferences. This collaborative decision-making process ensures that screening strategies are optimized to maximize detection while minimizing potential harms and risks.

Chapter 3 Methodology

3.1 Data Collection

The database I have used is the WDBC dataset.

The "WDBC" dataset stands for the "Wisconsin Diagnostic Breast Cancer" dataset. This dataset is widely used in machine learning and medical research for studying breast cancer detection and diagnosis.

The dataset contains features computed from digitized images of fine needle aspirates (FNA) of breast masses. These features include characteristics such as texture, radius, perimeter, area, smoothness, compactness, concavity, symmetry, and fractal dimension.

Each sample in the dataset is labeled as either benign (non-cancerous) or malignant (cancerous), making it suitable for binary classification tasks. Researchers and data scientists use the WDBC dataset to develop and evaluate machine learning models for distinguishing between benign and malignant breast tumors based on their features extracted from FNA images.

The dataset was originally created at the University of Wisconsin by Dr. William H. Wolberg, W. Nick Street, and Olvi L. Mangasarian, and it has since become a standard benchmark dataset in the field of breast cancer research.

3.2 Why Wisconsin Dataset of Breast Cancer?

The WDBC dataset is a benchmark dataset for breast cancer classification and detection. It has allowed many researchers to apply machine learning techniques in their research process.

Here are some key characteristics of the Wisconsin Breast Cancer dataset:

1. Origin: The dataset originates from research conducted at the University of Wisconsin, Madison, and the Wisconsin Comprehensive Cancer Center.
2. Size: The dataset consists of 569 instances, each with 30 features, resulting in a total of 17,070 data points.
3. Features: The features included in the dataset are primarily numerical and describe properties of cell nuclei observed in the FNA images. These features include attributes such as radius, texture, perimeter, area, smoothness, compactness, concavity, symmetry, and fractal dimension.
4. Classes: The dataset contains two classes: benign and malignant. Each instance is labeled with one of these classes based on the diagnosis of the corresponding breast mass.
5. Outliers: Within the malignant class, 21 data points are considered outliers. These outliers may have unique characteristics that distinguish them from the rest of the malignant samples and are often of interest in data analysis.

Researchers use the Wisconsin Breast Cancer dataset for various purposes, including developing and evaluating machine learning algorithms for breast cancer diagnosis and prognosis. Its availability and well-defined nature make it a valuable resource for researchers interested in applying machine learning techniques to medical datasets, particularly in the context of breast cancer detection and classification.

Chapter 4 Analyzing the dataset

4.1 Data Visualization

```
print(breast_cancer["diagnosis"].value_counts())

# Visualize the counts

sns.countplot(breast_cancer["diagnosis"])

plt.show()
```

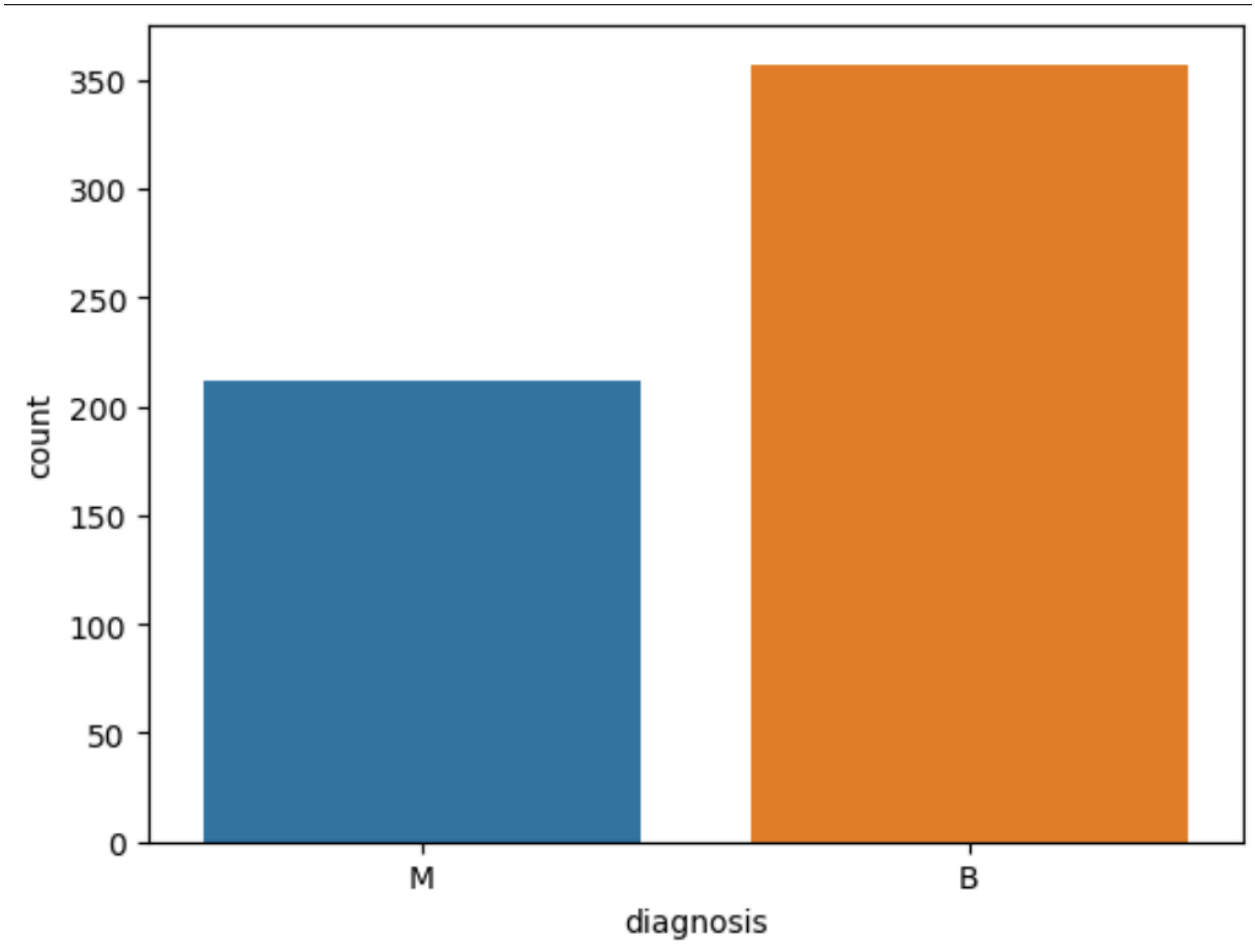


Fig 4.1 (**M = Malignant tumors B = Benign Tumors)**

Visualizing some 2-D features to see patterns:

```
def make_scatterplot(x,y):

    sns.scatterplot(x,y,data=breast_cancer,hue='diagnosis')

    plt.title(y + " vs " + x)

    plt.show()

make_scatterplot('radius_mean', 'texture_mean')

make_scatterplot('perimeter_mean', 'area_mean')

make_scatterplot('smoothness_mean', 'smoothness_se')

make_scatterplot('concavity_mean', 'compactness_mean')

make_scatterplot('fractal_dimension_mean', 'perimeter_se')

make_scatterplot('symmetry_worst', 'concave points_worst')
```

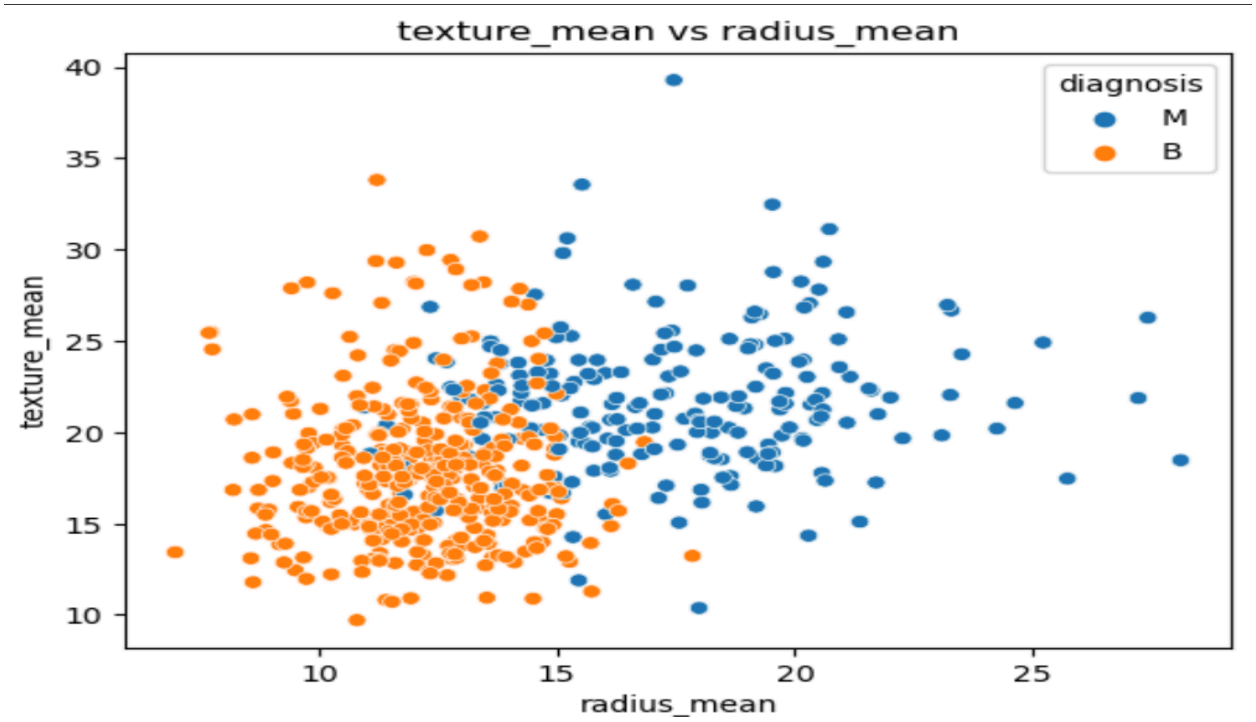


Fig 4.2 (Visualizing 2-D features to see patterns)

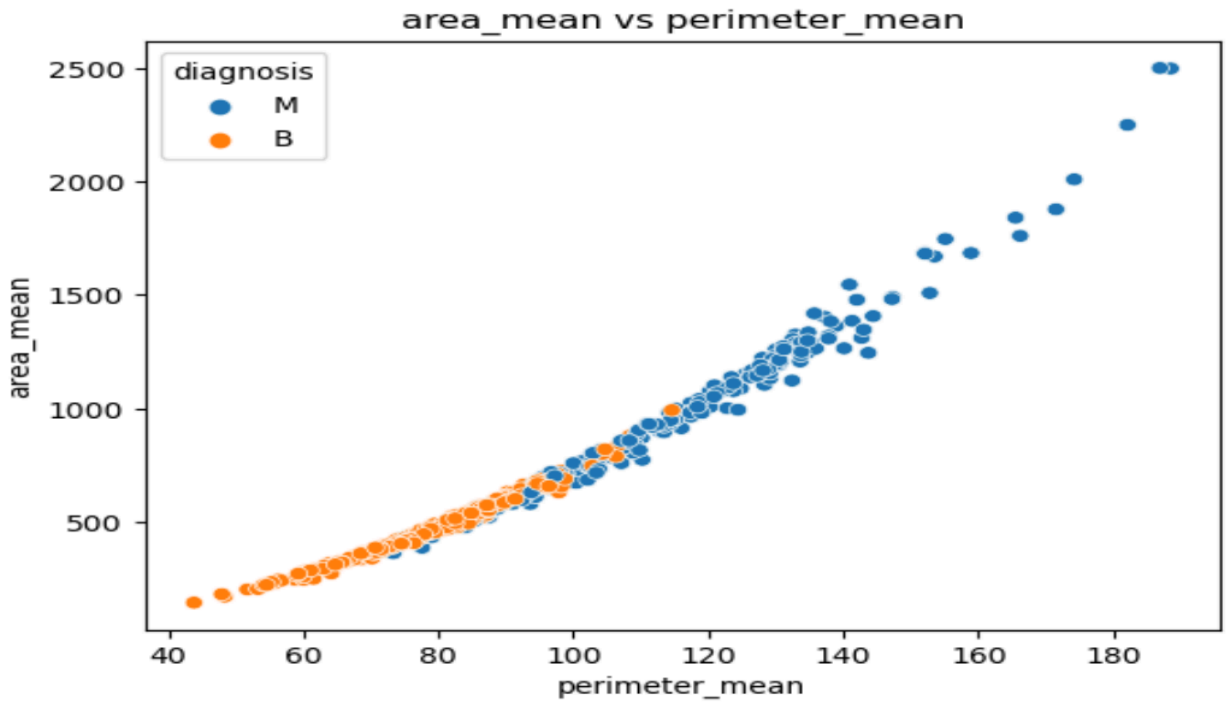


Fig 4.3 **(Area Mean vs. Perimeter mean of the breast cancer cells)**

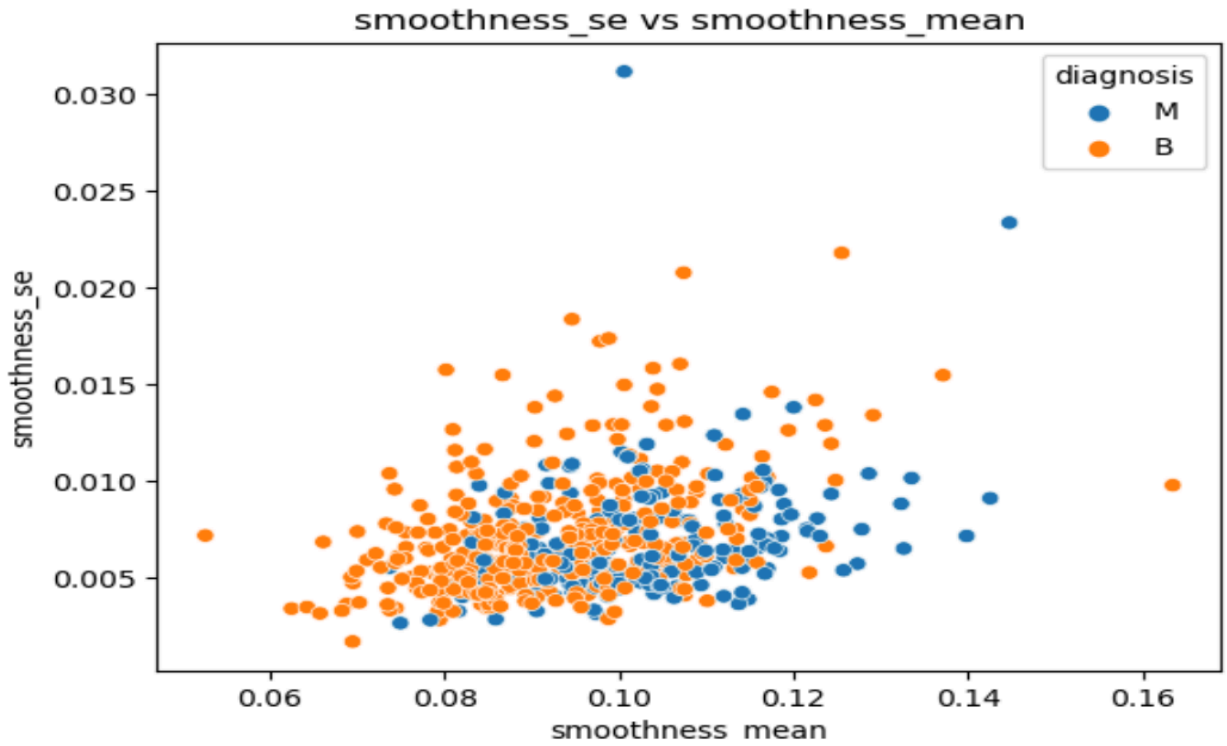


Fig 4.4 **(smoothness_se vs. smoothness mean of the breast cancer cells)**

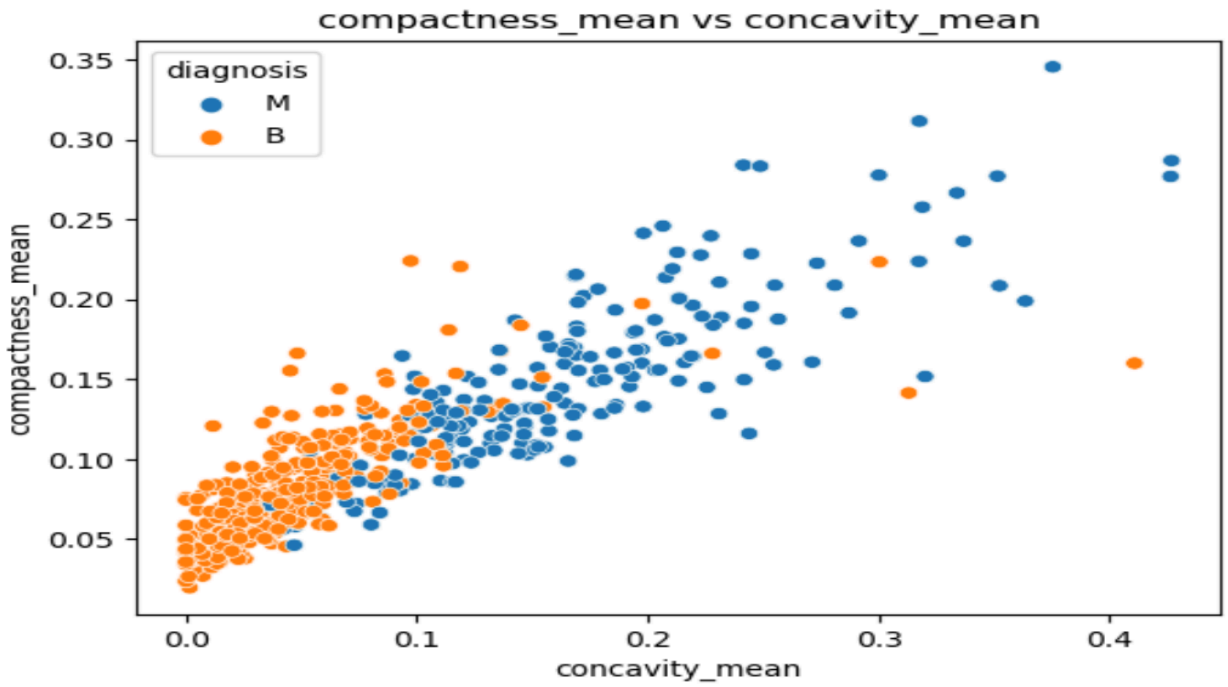


Fig 5.5 **(compactness_mean vs. concavity_mean of the breast cancer cells)**

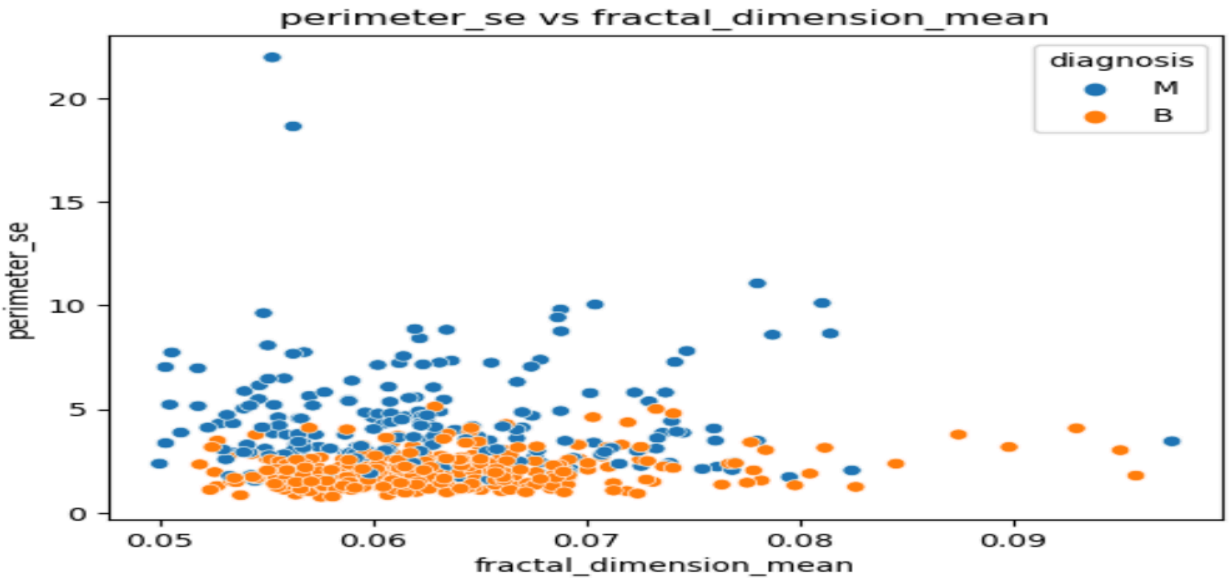


Fig 4.6 **(perimeter_se vs. fractal_dimension_mean of the breast cancer cells)**

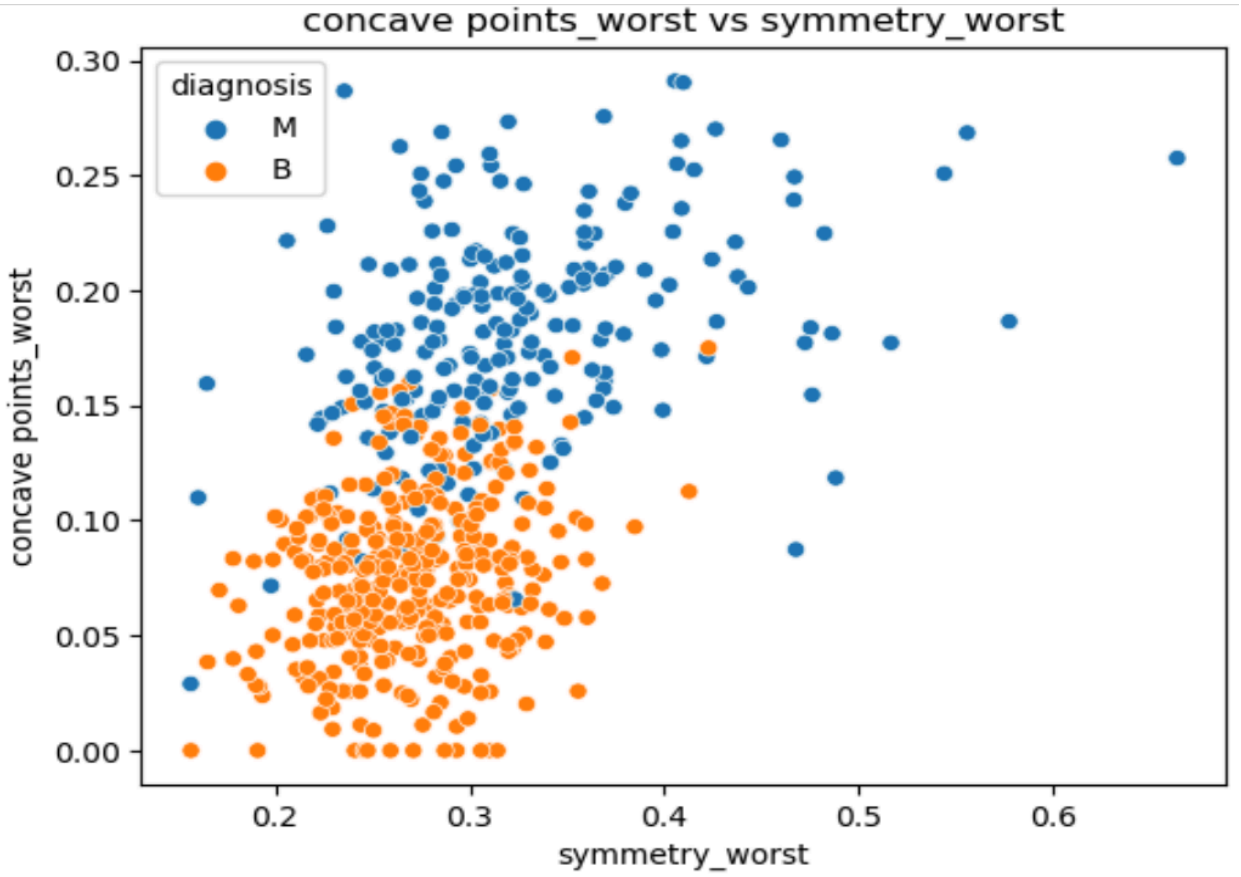


Fig 4.7 **(concave points_worst vs. symmetry_worst of the breast cancer cells)**

Correlation matrix:

```
print(breast_cancer.corr())
```

```
…                          radius_mean  texture_mean  perimeter_mean  area_mean  \
     radius_mean              1.000000      0.323782        0.997855   0.987357
     texture_mean             0.323782      1.000000        0.329533   0.321086
     perimeter_mean           0.997855      0.329533        1.000000   0.986507
     area_mean                0.987357      0.321086        0.986507   1.000000
     smoothness_mean          0.170581     -0.023389        0.207278   0.177028
     compactness_mean         0.506124      0.236702        0.556936   0.498502
     concavity_mean           0.676764      0.302418        0.716136   0.685983
     concave points_mean      0.822529      0.293464        0.850977   0.823269
     symmetry_mean            0.147741      0.071401        0.183027   0.151293
     fractal_dimension_mean  -0.311631     -0.076437       -0.261477  -0.283110
     radius_se                0.679090      0.275869        0.691765   0.732562
     texture_se              -0.097317      0.386358       -0.086761  -0.066280
     perimeter_se             0.674172      0.281673        0.693135   0.726628
     area_se                  0.735864      0.259845        0.744983   0.800086
```

```
area_se                  0.735864      0.259845        0.744983   0.800086
smoothness_se           -0.222600      0.006614       -0.202694  -0.166777
compactness_se           0.206000      0.191975        0.250744   0.212583
concavity_se             0.194204      0.143293        0.228082   0.207660
concave points_se        0.376169      0.163851        0.407217   0.372320
symmetry_se             -0.104321      0.009127       -0.081629  -0.072497
fractal_dimension_se    -0.042641      0.054458       -0.005523  -0.019887
radius_worst             0.969539      0.352573        0.969476   0.962746
texture_worst            0.297008      0.912045        0.303038   0.287489
perimeter_worst          0.965137      0.358040        0.970387   0.959120
area_worst               0.941082      0.343546        0.941550   0.959213
...
symmetry_worst                        0.537848
fractal_dimension_worst               1.000000

[30 rows x 30 columns]
```

Visualizing with a heatmap

```
figure, ax = plt.subplots(figsize=(20,20))
mask = np.triu(np.ones_like(breast_cancer.corr(), dtype=np.bool))
sns.heatmap(breast_cancer.corr(), mask=mask, annot=True)
```

```
plt.show()
```

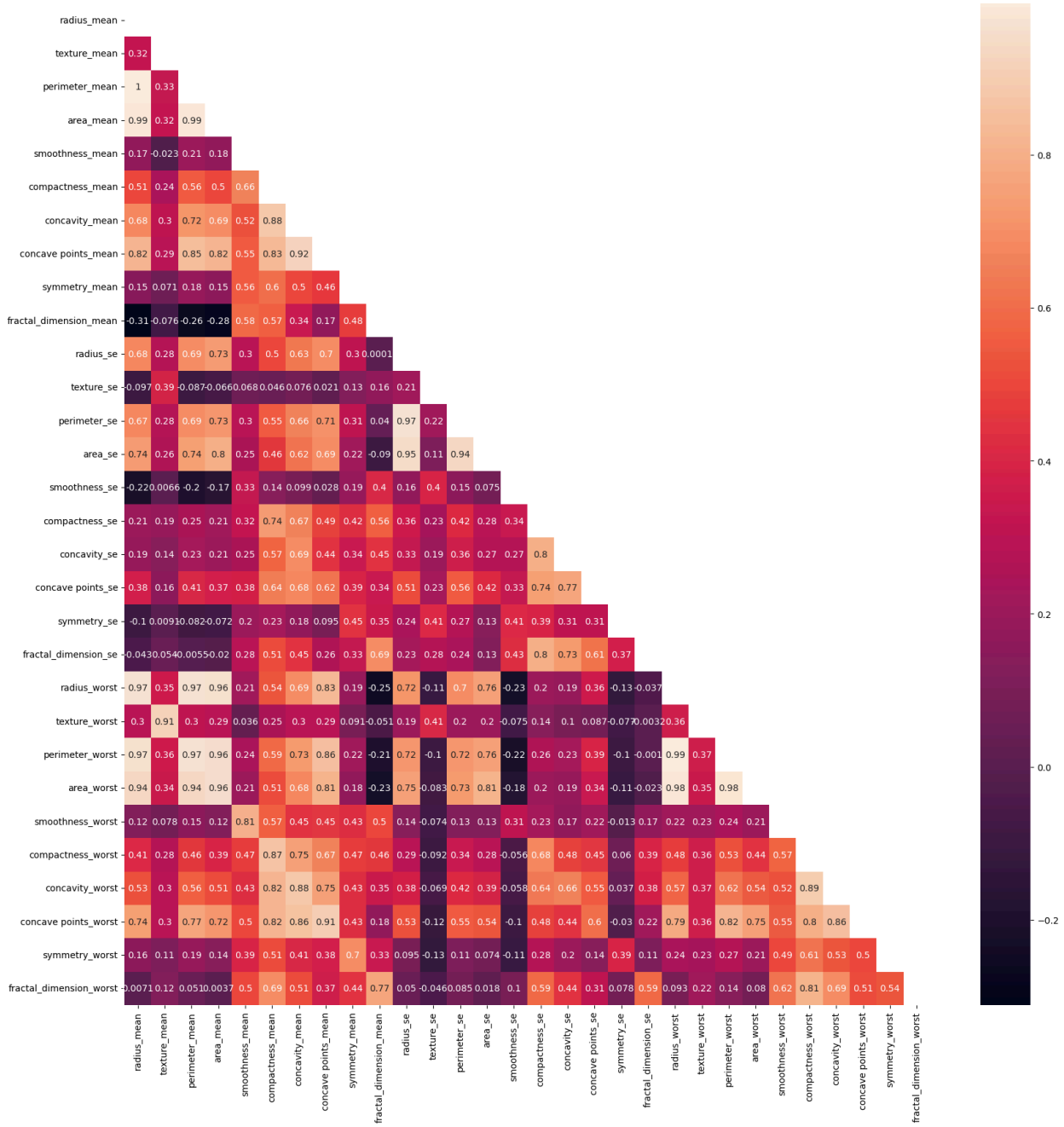


Fig 4.8 (Visualization of all the parameters using a heat map)

Plotting the histograms

```
def plot_histogram(column):
    sns.distplot(breast_cancer[column])
    plt.title(column)
```

```
    plt.show()

plot_histogram("radius_mean")
plot_histogram("texture_mean")
plot_histogram("perimeter_mean")
plot_histogram("area_mean")
plot_histogram("smoothness_mean")
plot_histogram("compactness_mean")
plot_histogram("concavity_mean")
plot_histogram("concave points_mean")
plot_histogram("symmetry_mean")
plot_histogram("fractal_dimension_mean")
```

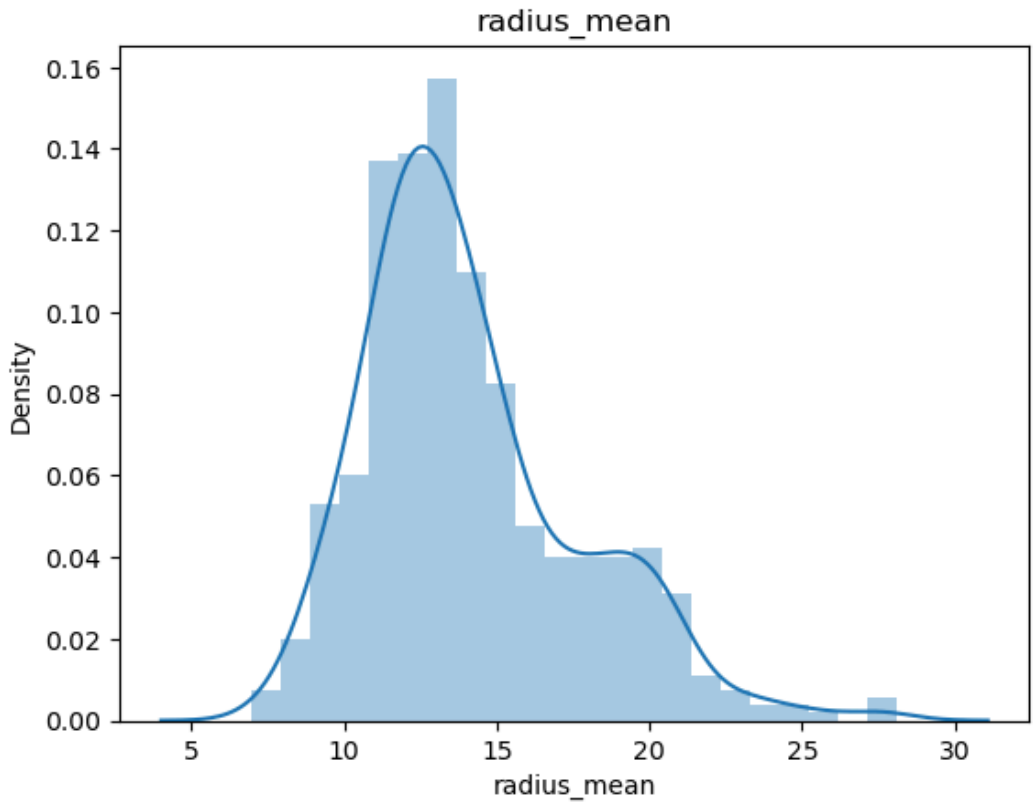


Fig 4.9 (Density vs. radius_mean of cells)

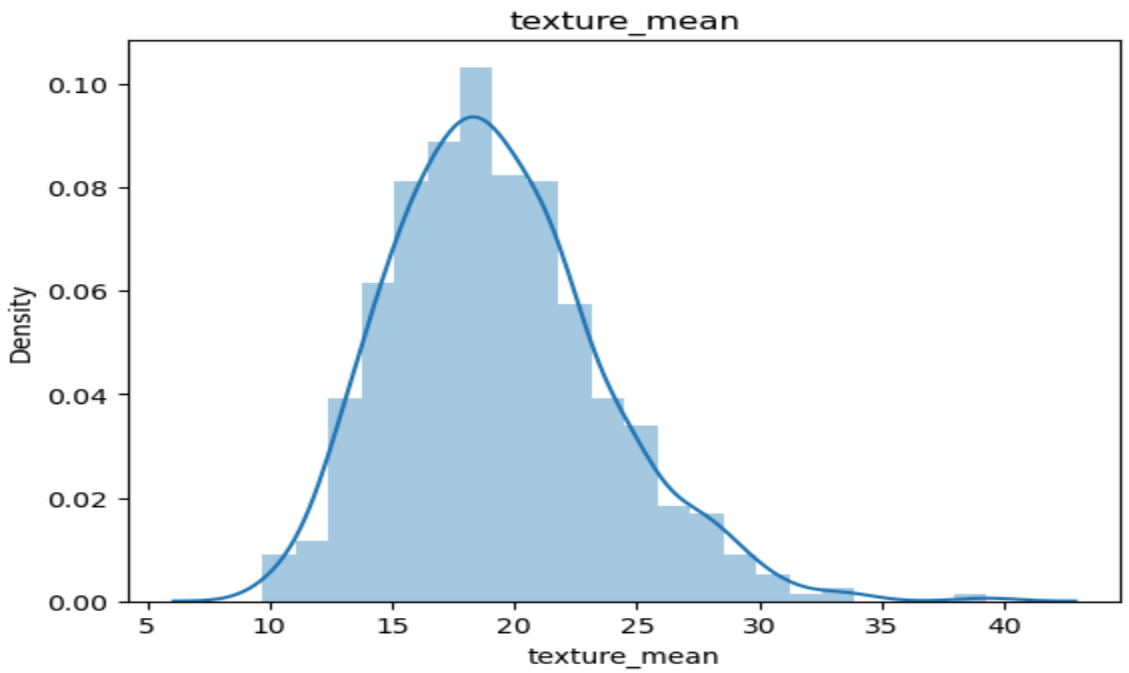


Fig 4.10 (Density vs. texture_mean of cells)

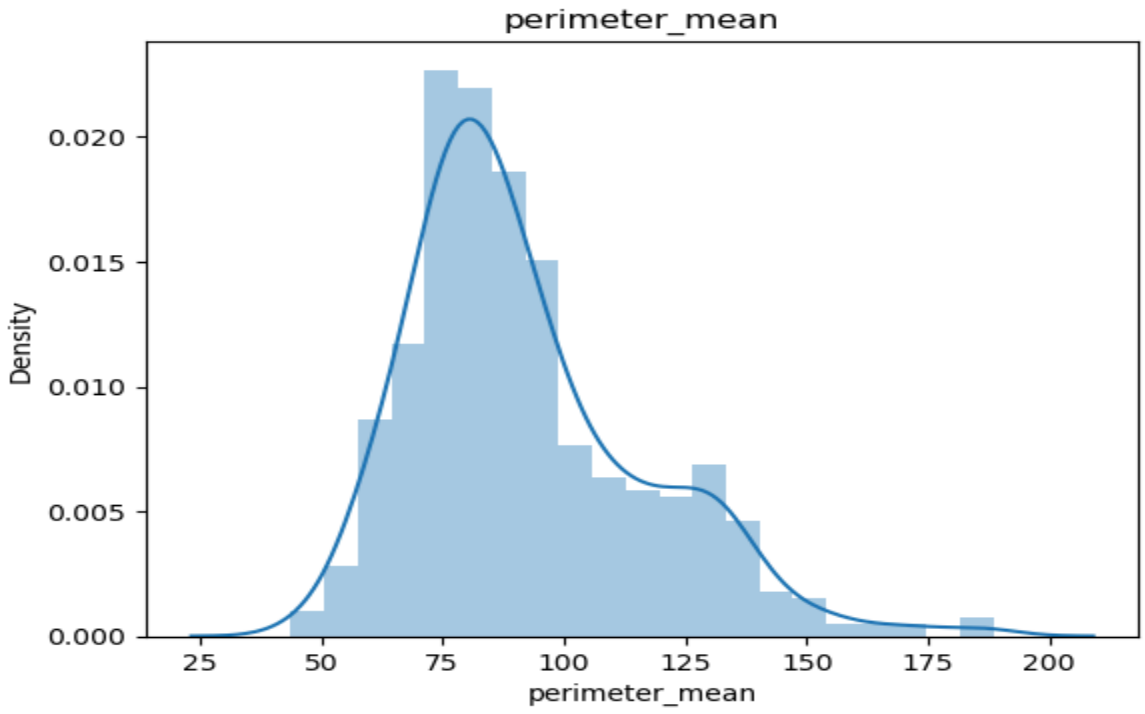


Fig 4.11(Density vs. perimeter_mean of cells)

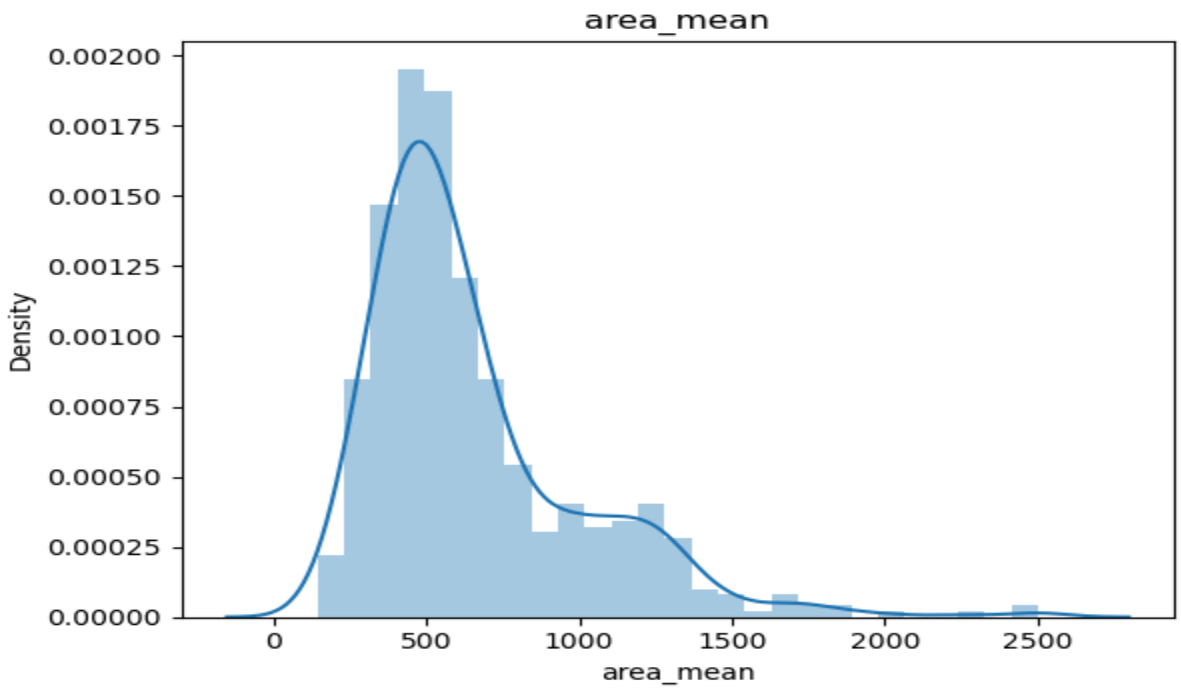


Fig 4.12 (Density vs. area_mean of cells)

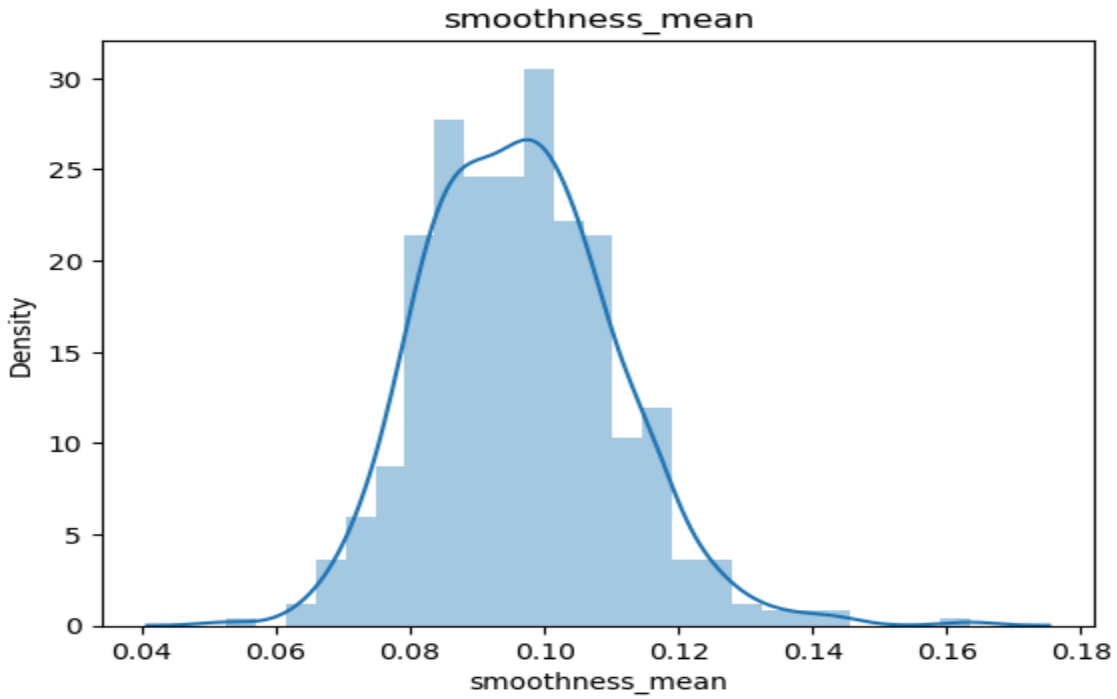


Fig 4.13 (Density vs. smoothness_mean of cells)

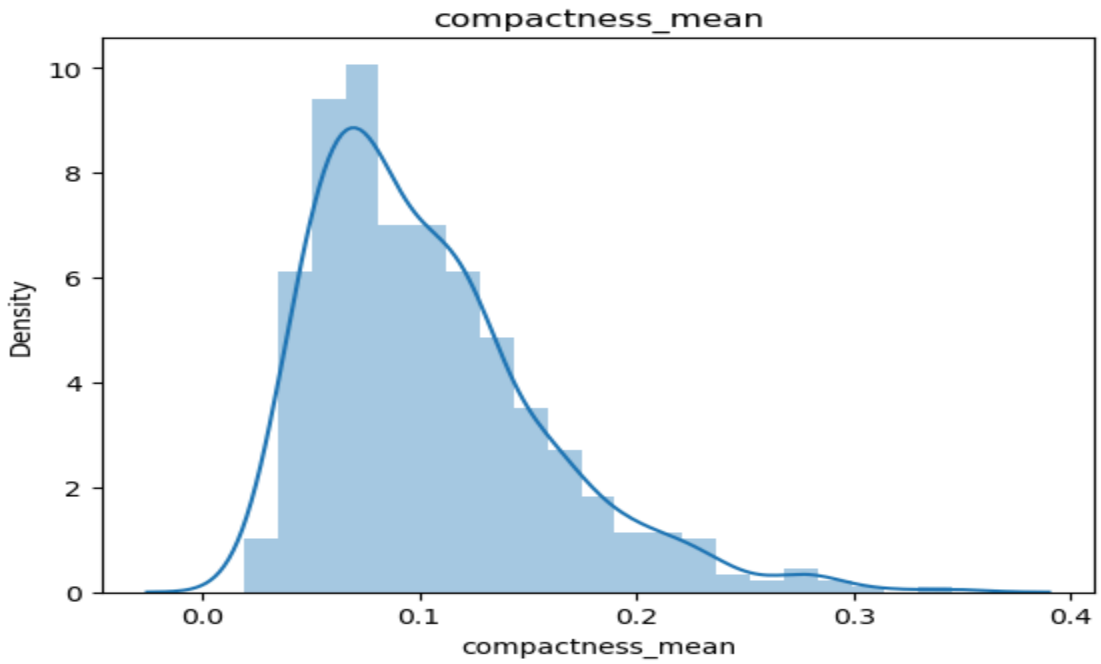


Fig 4.14 (Density vs. compactness_mean of cells)

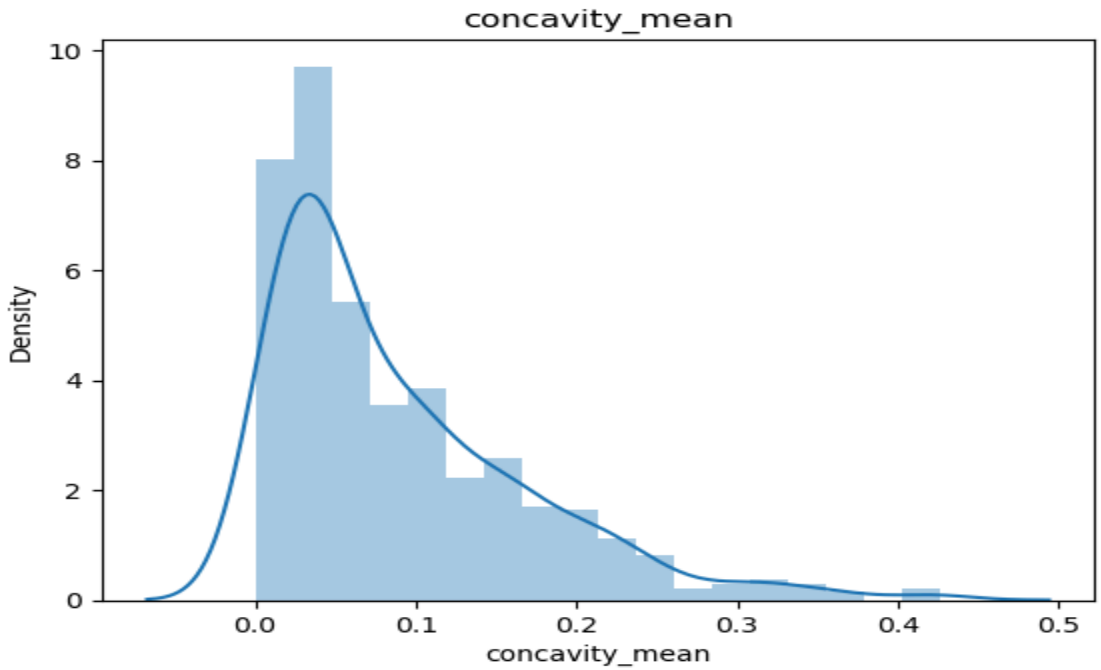


Fig 4.15 (Density vs. concavity_mean of cells)

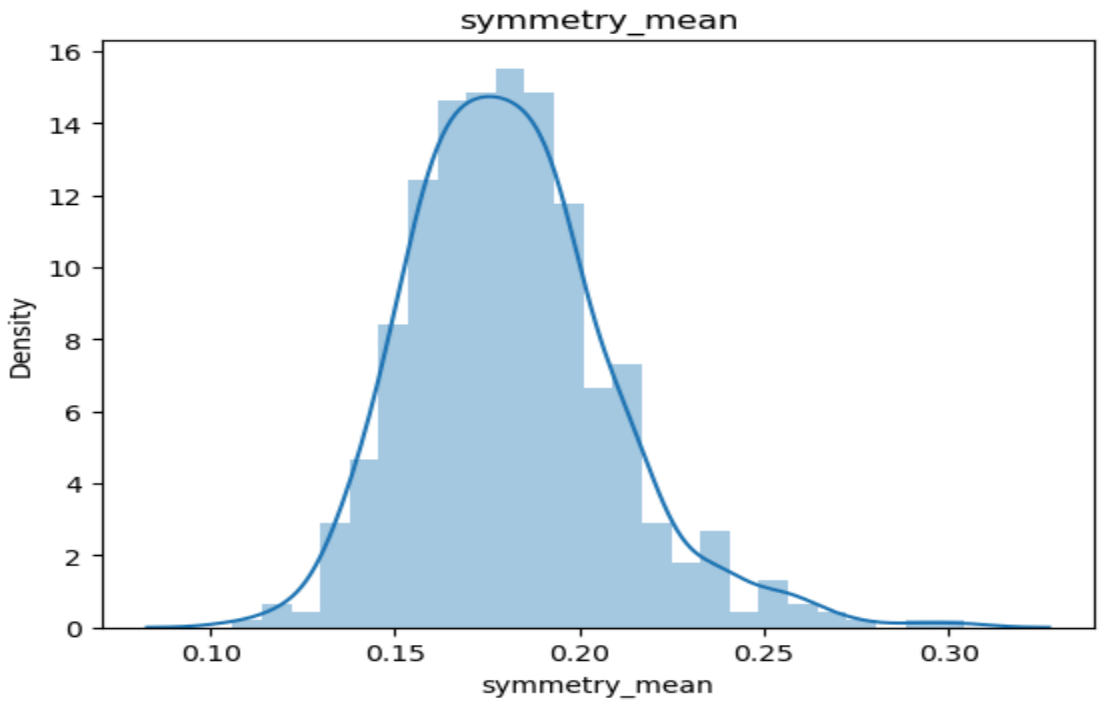


Fig 4.15 (Density vs. symmetry_mean of cells)

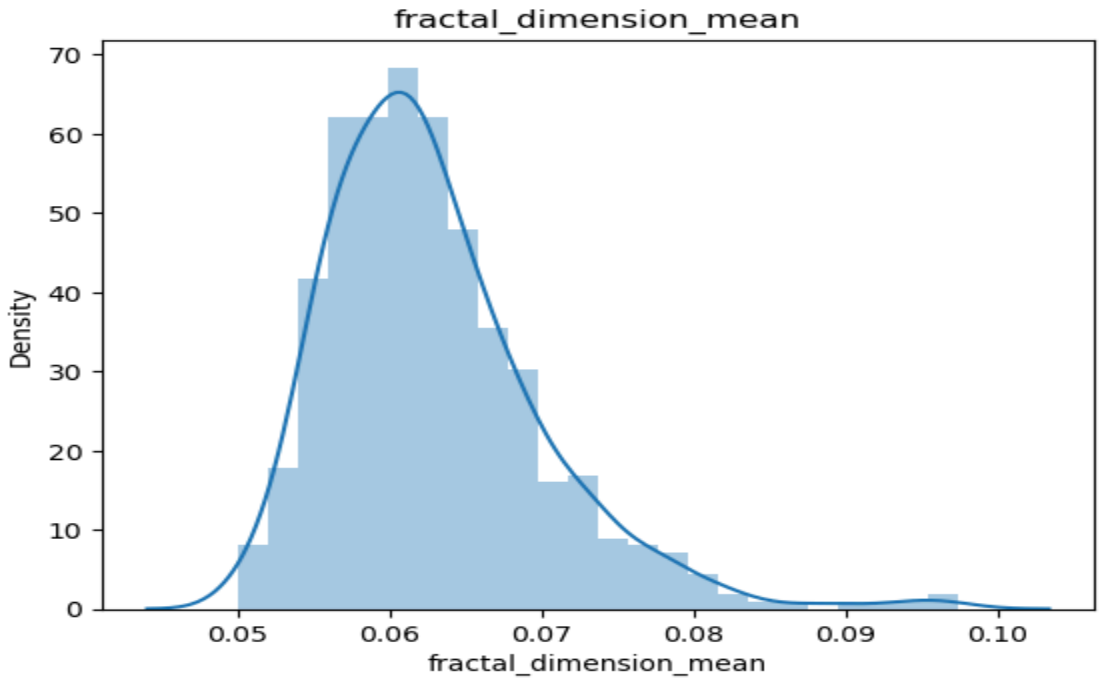
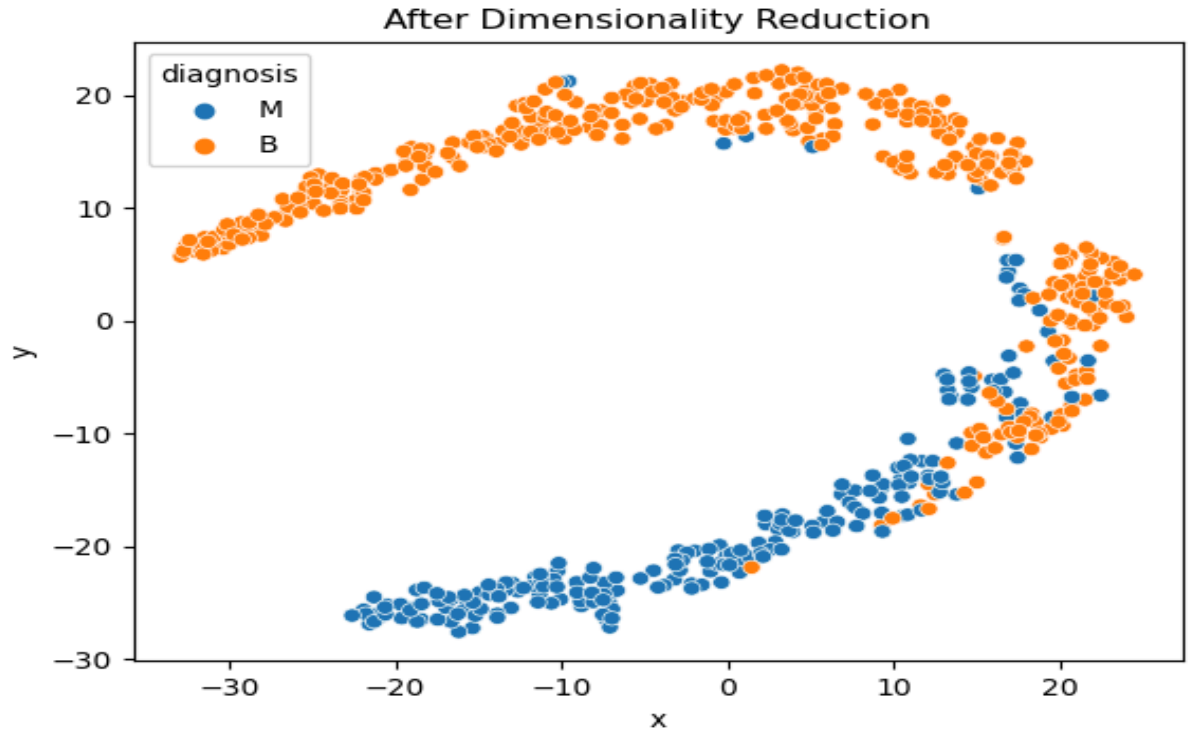


Fig 4.9 (Density vs. fractal_dimensional_mean of cells)

4.2 DIMENSIONALITY REDUCTION

```
# import TSNE
from sklearn.manifold import TSNE

# fit and transform the TSNE model
tsne = TSNE(learning_rate =  50)
tsne_f = tsne.fit_transform(breast_cancer.drop("diagnosis", axis=1))

# Create a new DataFrame to store reduced features
df = pd.DataFrame({'x':tsne_f[:,0],'y':tsne_f[:,1]})

print("Before:",breast_cancer.shape)
print("After",df.shape)

display(df.head())
sns.scatterplot(x='x', y='y', hue=breast_cancer['diagnosis'],data=df)
plt.title("After Dimensionality Reduction")
plt.show()
```



4.2.1 (Malign vs. Benign cancer after dimensionality reduction)

Getting features and the target-

```
X = breast_cancer.drop("diagnosis", axis=1)
y = breast_cancer["diagnosis"]
```

Splitting the data as 20% test and 80% training sets-

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=34)
```

Chapter 5 ALGORITHMS USED

Many machine learning algorithms, each with its own strengths and limitations exist. However, it's challenging to certify the superiority of one algorithm over others since the efficacy of these techniques generally tilts towards the characteristics of the dataset being analyzed. Previous researchers commonly have employed the following machine learning classifiers which have given promising results. I have used the following algorithms:

1. **Logistic Regression**
2. **Support Vector Machines (SVM)**
3. **Decision Trees**
4. **Naive Bayes**
5. **K-Nearest Neighbors (KNN)**
6. **Artificial Neural Networks (ANN)**
7. **Convolutional Neural Networks (CNN)**

Each classifier offers unique advantages and is suited to different types of data and tasks. The selection of the most appropriate classifier depends on factors such as the nature of the data, the complexity of the problem, and the desired outcome. Researchers often experiment with multiple classifiers and evaluate their performance to determine the most effective approach for a particular dataset and problem domain.

1. **LOGISTIC REGRESSION**

   Logistic regression is a statistical method used for binary classification tasks, where the outcome variable has two possible outcomes, such as "yes" or "no," "positive" or "negative." Despite its name, logistic regression is a classification algorithm rather than a regression algorithm. It's widely used in various fields, including healthcare, finance, and marketing, due to its simplicity and interpretability.

   How it Works:

Logistic regression models the relationship between one or more independent variables (features) and a binary dependent variable (outcome) using the logistic function. The logistic function, also known as the sigmoid function, maps any real-valued input to a value between 0 and 1. This output represents the probability that the given input belongs to a particular class.Mathematically, logistic regression computes the probability

`P(y=1|x) of the positive class (e.g., presence of a disease) given the input features x. It then uses this probability to make predictions by applying a threshold. If the predicted probability is above the threshold, the sample is classified as the positive class; otherwise, it's classified as the negative class.`

How it Works in Breast Cancer Detection:

In breast cancer detection, logistic regression can be applied to classify mammography images or patient data as either benign or malignant. Input features may include characteristics extracted from mammograms, such as texture, shape, and density of breast tissue, as well as patient demographics and clinical data.Logistic regression models the relationship between these features and the likelihood of breast cancer. By analyzing a dataset with known outcomes (e.g., biopsy results), the logistic regression model learns to predict the probability of malignancy for new cases based on their features. Healthcare providers can then use these predictions to assist in diagnosing breast cancer and planning appropriate treatment strategies.

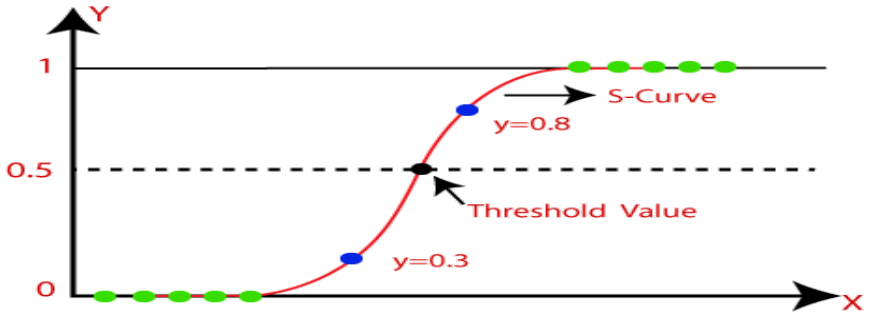


Fig 5.1 (Diagram depicting Logistic regression) [4]

2. **Support Vector Classifier (SVC)**, also known as Support Vector Machine (SVM), is a powerful supervised learning algorithm used for classification and regression tasks. In the context of breast cancer detection, SVC can be applied to classify mammography images or patient data into two classes: benign and malignant.

   How SVC Works:

   SVC works by finding the optimal hyperplane that separates data points of different classes in a high-dimensional space. The hyperplane is defined as the decision boundary that maximizes the margin, i.e., the distance between the hyperplane and the nearest data points of each class, known as support vectors. SVC aims to find the hyperplane that not only separates the classes but also generalizes well to unseen data. To achieve this, SVC employs a kernel trick, which maps the original input features into a higher-dimensional space where the classes are linearly separable. Commonly used kernels include linear, polynomial, and radial basis function (RBF) kernels.

   In breast cancer detection, SVC can be applied to classify mammography images or patient data as either benign or malignant based on relevant features extracted from the images or clinical data. These features may include characteristics such as texture, shape, and density of breast tissue, as well as patient demographics and medical history. SVC learns to classify new cases by analyzing a training dataset containing mammograms or patient data with known outcomes (e.g., biopsy results). By finding the optimal

hyperplane that separates benign and malignant cases, SVC can accurately predict the likelihood of breast cancer for new cases. Healthcare providers can use SVC predictions to assist in diagnosing breast cancer and making informed decisions about patient management and treatment strategies. By leveraging the power of SVC, clinicians can enhance the accuracy and efficiency of breast cancer detection, ultimately improving patient outcomes.

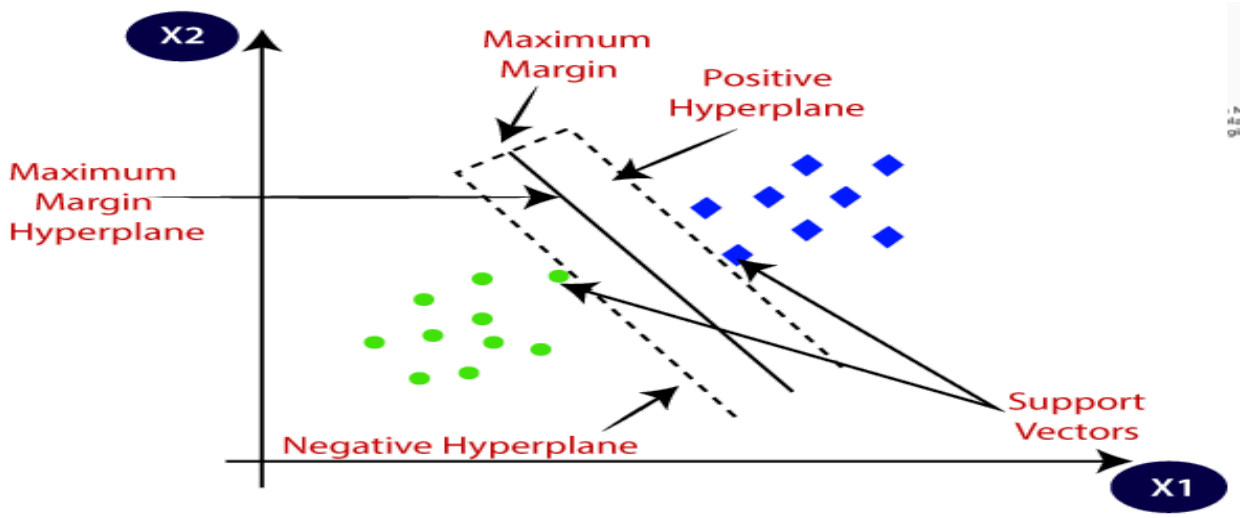


Fig 5.2 (Diagram depicting Support Vector Classifier) [5]

3. **Decision Tree**

DT is used in classification as well as regression. DT uses the tree structure & contains two types of nodes, namely the decision node and the leaf node. The decision node is a test, and the classification happens in the leaf node. The tree representation of the decision tree is given in the figure below:

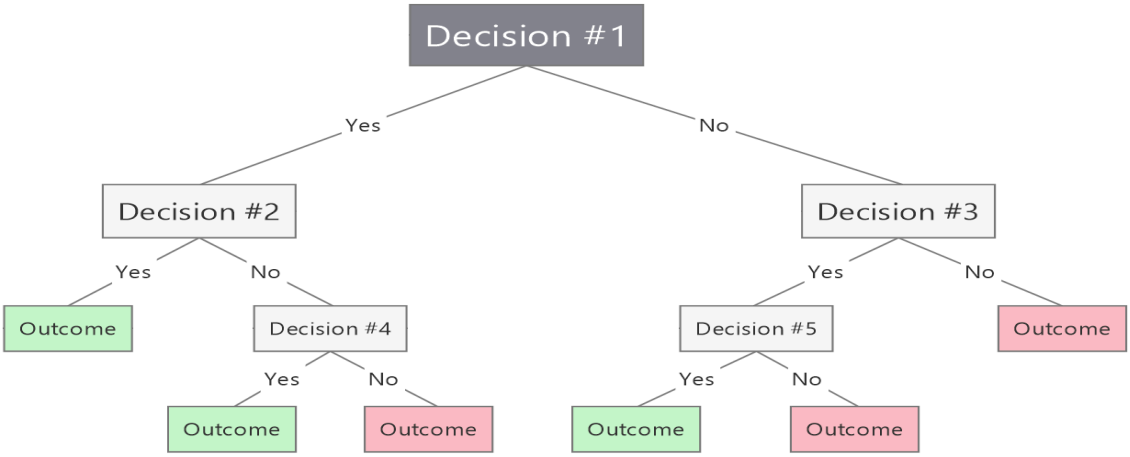


Fig 5.3 (Diagram depicting Decision Tree) [6]

4. **Naive Bayes (NB)** is a probabilistic classifier based on Bayes' theorem, assuming independence among features.

   In breast cancer detection, NB analyzes patient mammography images to classify cases as benign or malignant. By considering features- texture, shape, and density of breast tissue, NB calculates the probability of a given case belonging to each class.

   NB's simplicity, computational efficiency, and ability to handle high-dimensional data make it suitable for breast cancer detection tasks.

   Healthcare providers can use NB predictions to assist in diagnosing breast cancer and making informed decisions about patient care and treatment strategies.

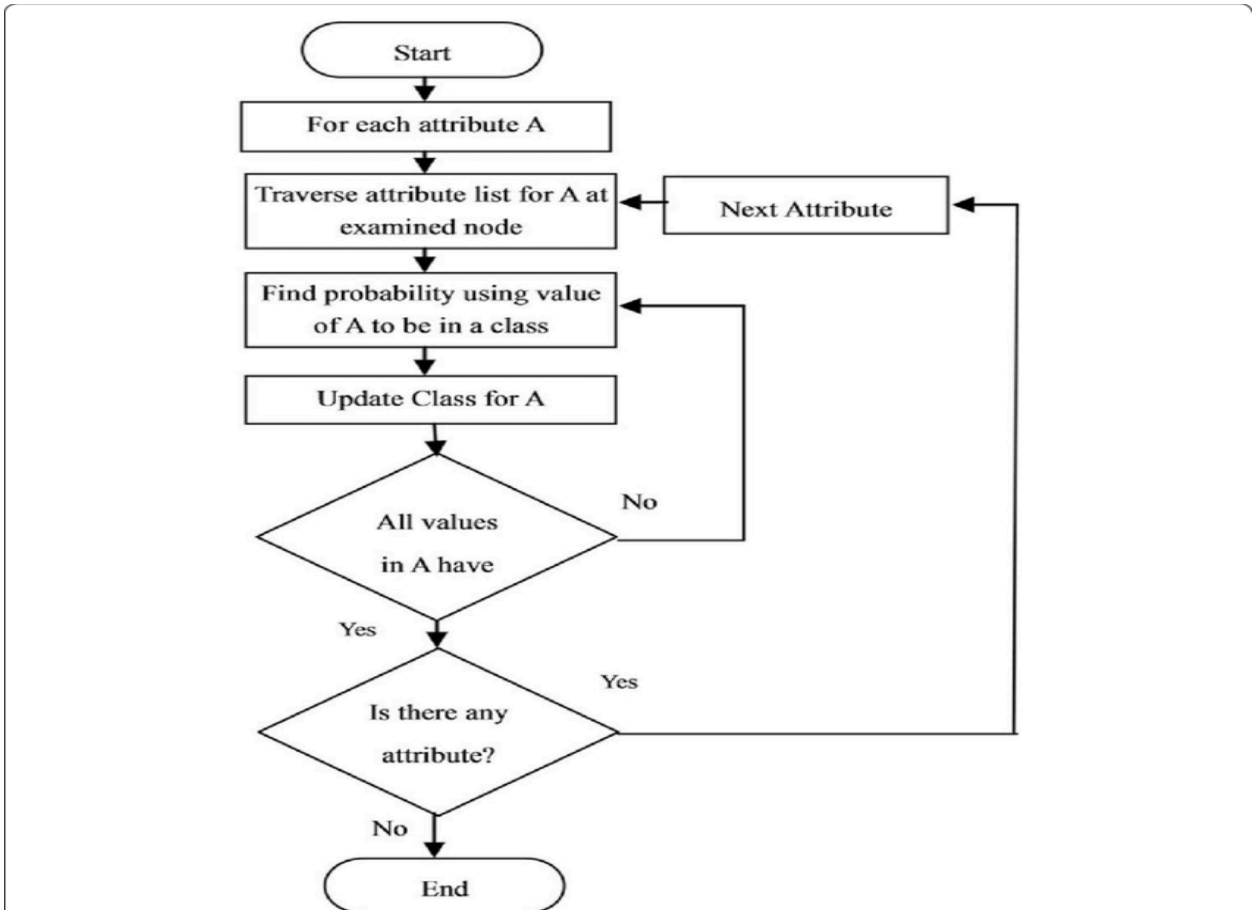


Fig 5.4 (Diagram depicting Naive Bayes) [7]

5. **KNN, or K-Nearest Neighbors** is a type of supervised learning algorithm used for making predictions based on the proximity of data points. It's versatile, capable of handling both regression and classification tasks, but it's predominantly employed for classification.

   The core idea behind KNN is that similar data points tend to cluster together.

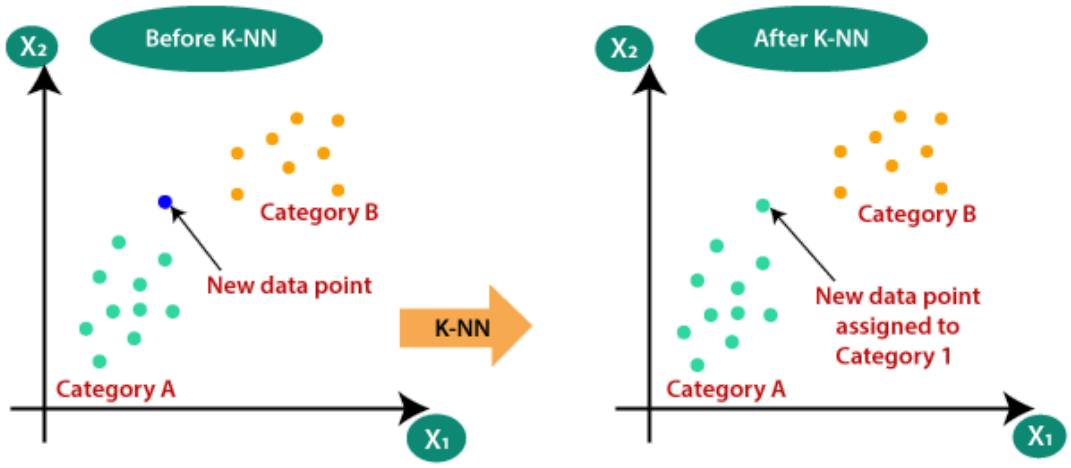


Fig 5.5 (Diagram depicting **KNN**) [8]

6. **Artificial Neural Networks (ANN):**

Neural networks, often abbreviated as NNs, are a type of machine learning model inspired by the structure and function of the human brain. They consist of interconnected layers of nodes (neurons) that process and transform input data to make predictions or classifications.

In breast cancer detection, neural networks are utilized to analyze mammography images or patient data to classify cases as either benign or malignant. By extracting features such as texture, shape, and density of breast tissue, neural networks learn complex patterns indicative of cancerous cells. Their ability to handle large amounts of data and capture intricate relationships makes them valuable tools in improving the accuracy and efficiency of breast cancer diagnosis.

Deep Neural Networks (DNN) were employed to tackle overfitting issues and achieved an impressive F1 score of 98. Another approach, discussed in another study, combined an unsupervised Artificial Neural Network (ANN), known as self-organizing map (SMO), with a supervised classifier called stochastic gradient descent (SGD), resulting in an accuracy of 99.68% on the WDBC(Wisconsin Diagnostic Breast Cancer) dataset.

Similarly, a study utilized a computer-aided system (CAD) featuring joint variable system and constructive Deep Neural Network (DNN), obtaining accuracies of 99.1% and 89.3% on the WDBC and SEER 2017 datasets, respectively. In comparison, it was concluded that ANN techniques outperformed other machine learning methods, achieving the highest accuracy of 99.73%.

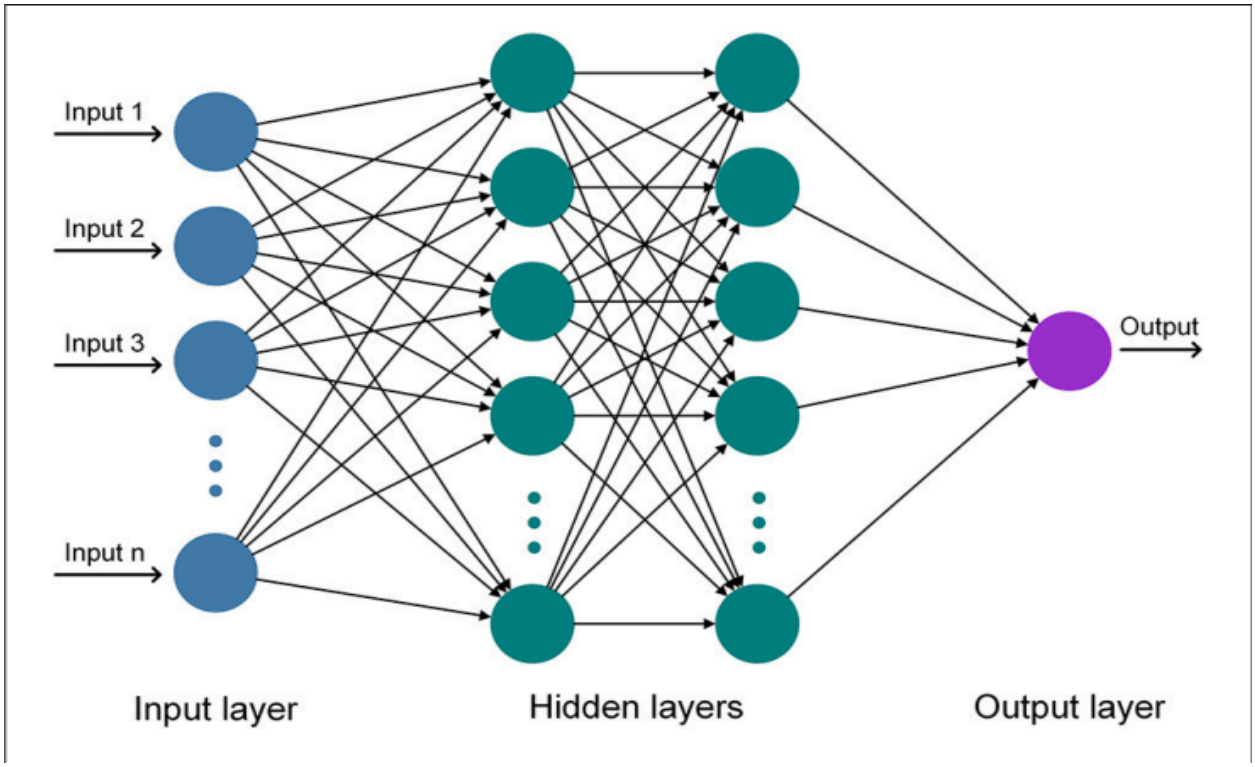


Fig 5.6 (Diagram depicting **ANN**) [9]

7. **CNN (Convolutional Neural Network):**

Convolutional Neural Networks (NNs) function similarly to traditional Artificial Neural Networks (ANNs), comprising interconnected neurons that adjust themselves to optimize performance. Each neuron receives input and performs operations, culminating in a perceptual scoring function from raw input picture vectors to the final class score output. The network's last layer is associated with loss functions linked to different classes,

following standard ANN principles and strategies. Unlike many other machine learning classifiers, Convolutional Neural Networks (CNNs) demand less preprocessing.

Studies have reported a remarkable 99.67% accuracy & achieved an F1 score of 98, and attained 98% accuracy using CNNs on the WDBC dataset.

Moreover, we have applied a CNN model to a 2D matrix for breast cancer diagnosis using the WDBC dataset, achieving 98.6% accuracy.

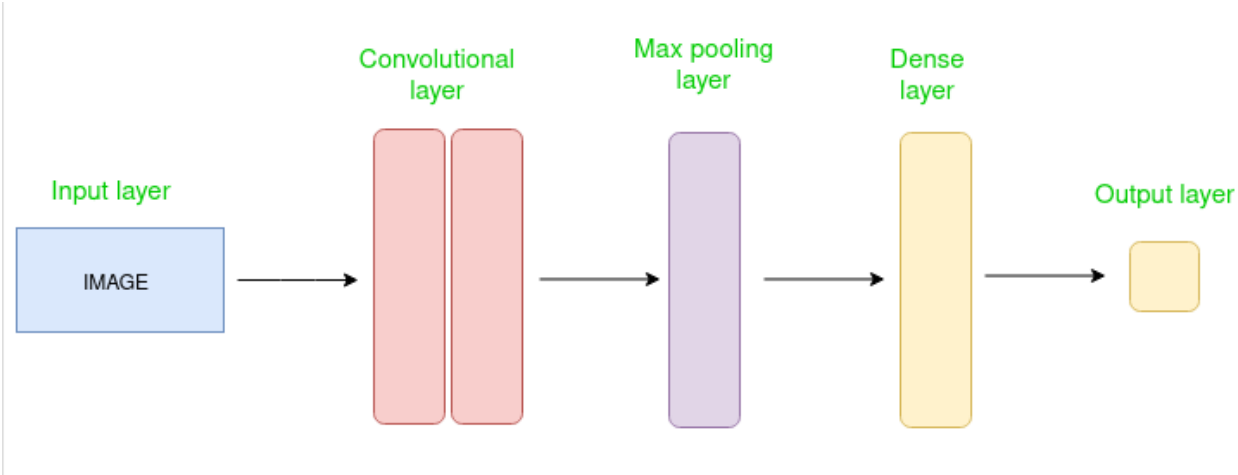


Fig 5.7 (Diagram depicting **CNN**) [10]

Chapter 6 Performance Metrics

6.1 Accuracy

Accuracy is a common metric used to evaluate the performance of machine learning models, particularly in classification tasks. It measures the proportion of correctly classified instances out of the total number of instances in the dataset.

Accuracy = (Number of correct predictions/Total Number of predictions) * 100 %

MODEL	ACCURACY
Logistic Regression	98.9%
SVC	98.2%
Decision Tree Classifier	91.2%
Naive Bayes (NB)	90.3%
K-Nearest Neighbours (KNN)	95.61%
Artificial Neural Networks (ANN)	95.614%
Convolutional Neural Networks (CNN)	98.3%

(Table depicting the different accuracies of different algorithms when applied on WDBC)

6.1.1

MODEL	ACCURACY	Problems
LR, SVM, KNN, NV	98.24%	No literature study available to describe
ANN, DL	98%	Less number of references used
Hybrid Model	99.48%	Model with a high level of complexity
CNN	99.67%	A small number of features are employed
PCA, MLP, CNN	99.10%	Low value in confusion matrix

ML technique with genetic programming	98.24%	No literature study available to describe

(Fig 6.1 Compared results of other researchers' work in the field of Diagnosis of Breast Cancer) [11]

6.2 F1 Score

F1 Score = 2 *((precision * recall) / (precision + recall))

where precision is the proportion of true positive predictions out of all positive predictions made by the model, and recall is the proportion of true positive predictions out of all actual positive instances in the dataset (WDBC used here).

6.3 Specificity

Specificity, also known as true negative rate, measures the proportion of actual negative instances that are correctly identified as negative by the model. It is calculated using the formula:

Specificity = True Negatives / (True Negative + False Positives)

Specificity is particularly useful in scenarios where correctly identifying negative instances is critical, such as in medical diagnostics where false positives can lead to unnecessary treatments or interventions.

6.4 Precision

Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It is calculated using the formula:

Precision = True Positives / (True Positives + False Positives)

Precision is a useful metric when the cost of false positives is high, as it indicates the accuracy of the positive predictions made by the model.

6.5 Recall or sensitivity:

Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive instances that are correctly identified as positive by the model. It is calculated using the formula:

Recall = True Positives / (True Positives + False Negatives)

Recall is particularly useful when it is important to identify all positive instances, as it indicates the model's ability to capture all relevant cases of the positive class.

Chapter 7 Code Implementation & Results

jupyter ML_Algorithms Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

```
[1]: import pandas as pd

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

[53]: # df.head(10)
      # df.info()
      # df.isna().sum()
      # df.describe()

[4]: df = df.dropna(axis=1)

[54]: df.head(5)
```

[54]:

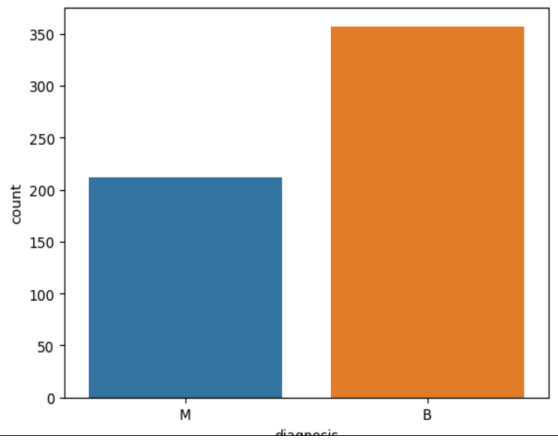
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst
0	842302	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	25.38
1	842517	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	24.99
2	84300903	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	23.57
3	84348301	1	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	14.91
4	84358402	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	22.54

5 rows × 32 columns

7.1 Importing and exploring the dataset

```
[9]: import seaborn as sns

[10]: # sns.countplot(df['diagnosis'], label='count')
      sns.countplot(x='diagnosis', data=df, label='count')
```

[10]: <Axes: xlabel='diagnosis', ylabel='count'>



7.2 Visualizing the dataset

```
[11]: from sklearn.preprocessing import LabelEncoder
      lb=LabelEncoder()

[12]: df.iloc[:, 1]=lb.fit_transform(df.iloc[:, 1].values)

[13]: # 1 = Malignant, 0 = Benign

[55]: df.head(5)
```

[55]:

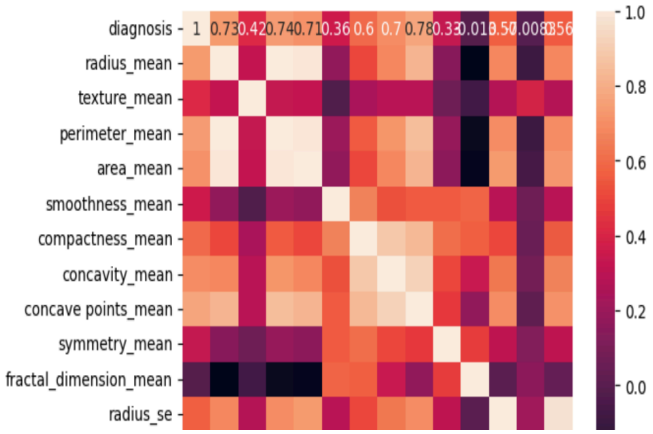
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst
0	842302	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	25.38
1	842517	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	24.99
2	84300903	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	23.57
3	84348301	1	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	14.91
4	84358402	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	22.54

5 rows × 32 columns

```
[15]: df['diagnosis'].value_counts()

[15]: diagnosis
      0    357
      1    212
```

7.3 use of labelEncoder to encode the values of diagnosis column

```
[17]: import matplotlib.pyplot as plt

[18]: plt.figure(figsize=(10,10))

[18]: <Figure size 1000x1000 with 0 Axes>
      <Figure size 1000x1000 with 0 Axes>

[19]: sns.heatmap(df.iloc[:, 1:15].corr(), annot=True)

[19]: <Axes: >
```



7.4 data visualization using a heatmap

1. Applying LogisticRegression:

```
[33]: from sklearn.model_selection import train_test_split

[35]: from sklearn.preprocessing import StandardScaler

[38]: X_train.shape

[38]: (455, 30)

[39]: y_train.shape

[39]: (455,)

[40]: from sklearn.linear_model import LogisticRegression

[41]: import numpy as np

      unique_classes = np.unique(y_train)
      print(unique_classes)

      [0 1]

[42]: # Apply Label encoding to the target variable
      lb = LabelEncoder()
      df['diagnosis'] = lb.fit_transform(df['diagnosis'].values)

      # Check the distribution of the encoded target variable
      print(df['diagnosis'].value_counts())

      # Split the data into features and labels
```

7.1.1 Importing the required libraries

```
[42]: # Apply Label encoding to the target variable
      lb = LabelEncoder()
      df['diagnosis'] = lb.fit_transform(df['diagnosis'].values)

      # Check the distribution of the encoded target variable
      print(df['diagnosis'].value_counts())

      # Split the data into features and labels
      X = df.iloc[:, 2:32].values
      y = df.iloc[:, 1].values

      # Split the data into training and testing sets
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

      # Standardize the feature values
      scaler = StandardScaler()
      X_train = scaler.fit_transform(X_train)
      X_test = scaler.transform(X_test)

      # Initialize and fit the Logistic Regression model
      log_reg = LogisticRegression()
      log_reg.fit(X_train, y_train)

      diagnosis
      0    357
      1    212
      Name: count, dtype: int64

[42]: ▾ LogisticRegression
```

7.1.2 Applying label Encoding, Splitting the datasets into (test & train), standardizing the feature values

```
[43]: # Initialize and fit the Logistic Regression model
log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)

# Verify if the model has converged and attributes are available
if hasattr(log_reg, 'intercept_') and hasattr(log_reg, 'coef_'):
    # Model has converged, calculate training accuracy
    training_accuracy = log_reg.score(X_train, y_train)
    print("Training Accuracy:", training_accuracy)
else:
    # Model hasn't converged or training failed
    print("Model training failed or hasn't converged.")

# Check coefficients if the model has converged
if hasattr(log_reg, 'coef_'):
    # Access coefficients and intercept if available
    coefficients = log_reg.coef_
    intercept = log_reg.intercept_
    print("Coefficients:", coefficients)
    print("Intercept:", intercept)
else:
    print("Model coefficients are not available.")
```

```
Training Accuracy: 0.989010989010989
Coefficients: [[ 0.33876214  0.48939456  0.330574    0.40274765  0.19380529 -0.44575519
   0.67211017  0.84612924  0.33758483 -0.21274935  1.39050559 -0.0394851
   0.84507703  0.97876119 -0.25537755 -0.6623816  -0.12210568  0.2273378
  -0.12585617 -0.86832724  0.93114793  1.04054114  0.76764961  0.9005161
   0.53303391  0.0284274   0.86650877  0.97095476  0.51867962  0.60055403]]
```

7.1.3 Applying the algorithm and checking for it's performance

2. Applying SVC Algorithm:

```
[45]: from sklearn.svm import SVC
# Initialize the SVC classifier
svc = SVC()

# Train the SVC classifier
svc.fit(X_train, y_train)

# Predict the labels of the test set
y_pred = svc.predict(X_test)

# Evaluate the performance of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Generate a classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.9824561403508771
Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99        67
           1       1.00      0.96      0.98        47

    accuracy                           0.98       114
   macro avg       0.99      0.98      0.98       114
weighted avg       0.98      0.98      0.98       114
```

7.2.1 Application of SVC and it's result

3. Decision_Tree:

```
[46]: from sklearn.tree import DecisionTreeClassifier

[47]: # Initialize the Decision Tree Classifier
dt_classifier = DecisionTreeClassifier()

# Train the Decision Tree Classifier
dt_classifier.fit(X_train, y_train)

# Predict the labels of the test set
y_pred = dt_classifier.predict(X_test)

# Evaluate the performance of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Generate a classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.9122807017543859
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.90      0.92        67
           1       0.86      0.94      0.90        47

    accuracy                           0.91       114
   macro avg       0.91      0.92      0.91       114
weighted avg       0.92      0.91      0.91       114
```

7.3.1 Applying Decision tree algorithm and printing its results

4. Naive Bayes:

```
[48]: from sklearn.naive_bayes import GaussianNB

[49]: # Initialize the Gaussian Naive Bayes classifier
nb_classifier = GaussianNB()

# Train the Gaussian Naive Bayes classifier
nb_classifier.fit(X_train, y_train)

# Predict the labels of the test set
y_pred = nb_classifier.predict(X_test)

# Evaluate the performance of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Generate a classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.9035087719298246
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.91      0.92        67
           1       0.88      0.89      0.88        47

    accuracy                           0.90       114
   macro avg       0.90      0.90      0.90       114
weighted avg       0.90      0.90      0.90       114
```

7.4.1 Applying Naive Bayes algorithm & checking it's performance

5. KNN:

```
[50]: from sklearn.neighbors import KNeighborsClassifier

[51]: # Initialize the K-Nearest Neighbors classifier
knn_classifier = KNeighborsClassifier()

# Train the K-Nearest Neighbors classifier
knn_classifier.fit(X_train, y_train)

# Predict the labels of the test set
y_pred = knn_classifier.predict(X_test)

# Evaluate the performance of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Generate a classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

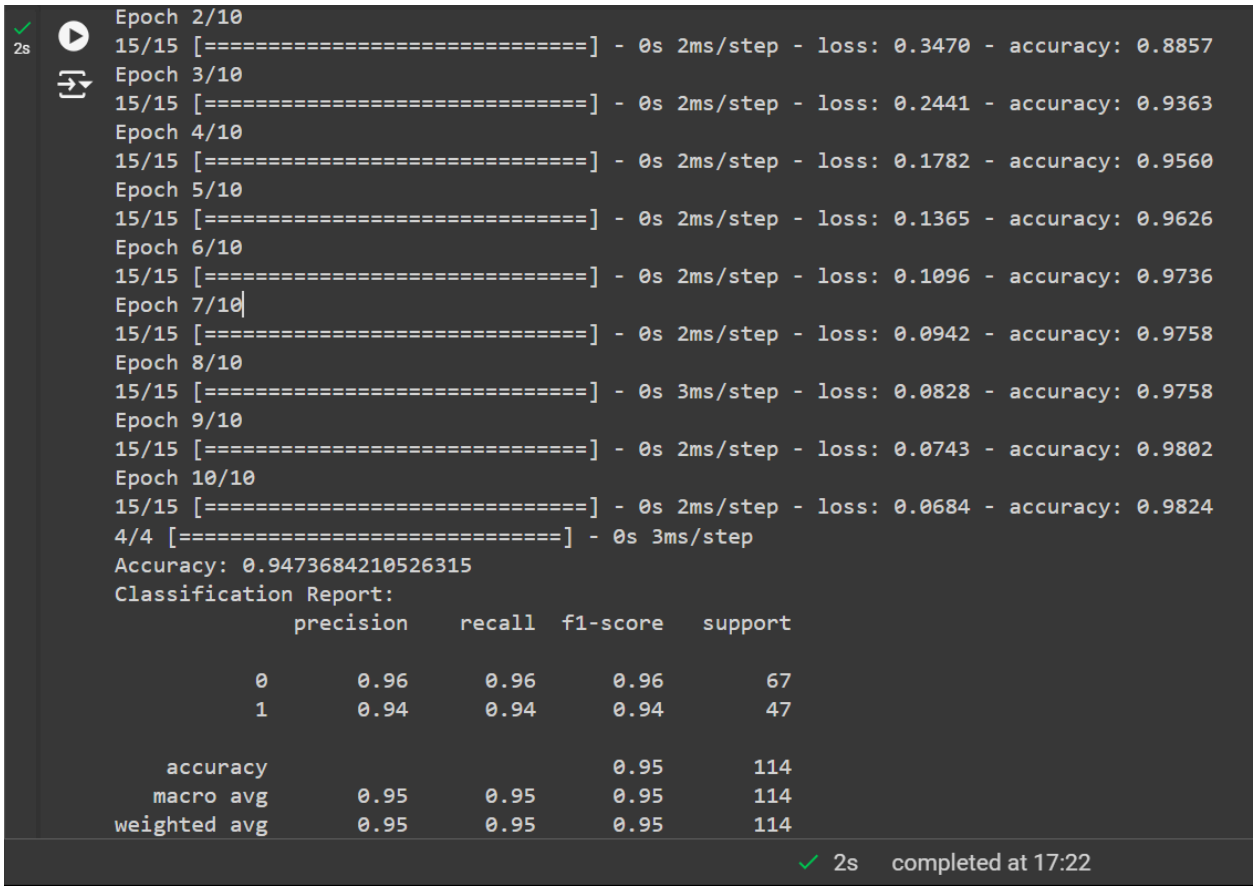
```
Accuracy: 0.956140350877193
Classification Report:
              precision    recall  f1-score   support

           0       0.93      1.00      0.96        67
           1       1.00      0.89      0.94        47

    accuracy                           0.96       114
   macro avg       0.97      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114
```

7.5.1 Applying KNN algorithm

6. ANN:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense


# Build the ANN model
model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32, verbose=1)

# Evaluate the performance of the model
y_pred = model.predict(X_test)
y_pred_binary = (y_pred > 0.5).astype(int)
accuracy = accuracy_score(y_test, y_pred_binary)
print("Accuracy:", accuracy)

# Generate a classification report
print("Classification Report:")
print(classification_report(y_test, y_pred_binary))
```

7.6.1 Applying ANN algorithm, building an ANN model



7.6.1 Results of ANN model

7. CNN:

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_breast_cancer
from sklearn.preprocessing import StandardScaler
import tensorflow as tf
from tensorflow.keras import layers, models

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Reshape data for CNN input (add channel dimension)
X_train = X_train.reshape(-1, X.shape[1], 1)
X_test = X_test.reshape(-1, X.shape[1], 1)

# Build the CNN model
model = models.Sequential([
    layers.Conv1D(32, kernel_size=3, activation='relu', input_shape=(X_train.shape[1], 1)),
    layers.MaxPooling1D(pool_size=2),
    layers.Conv1D(64, kernel_size=3, activation='relu'),
    layers.MaxPooling1D(pool_size=2),
    layers.Flatten(),
```

7.7.1 CNN algorithm applied

```
# Building the CNN model
model = models.Sequential([
    layers.Conv1D(32, kernel_size=3, activation='relu', input_shape=(X_train.shape[1], 1)),
    layers.MaxPooling1D(pool_size=2),
    layers.Conv1D(64, kernel_size=3, activation='relu'),
    layers.MaxPooling1D(pool_size=2),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])

# Compiling the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Training the model
history = model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_test, y_test), verbose=2)

# Evaluating the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Loss: {loss}, Test Accuracy: {accuracy}')
```

7.7.2 Building, compiling & Evaluating the CNN model

```
# Evaluating the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Loss: {loss}, Test Accuracy: {accuracy}')
```

```
Epoch 1/20
15/15 - 2s - loss: 0.3883 - accuracy: 0.9055 - val_loss: 0.1833 - val_accuracy: 0.9211 - 2s/epoch - 122ms/step
Epoch 2/20
15/15 - 0s - loss: 0.1970 - accuracy: 0.9209 - val_loss: 0.1074 - val_accuracy: 0.9561 - 101ms/epoch - 7ms/step
Epoch 3/20
15/15 - 0s - loss: 0.1284 - accuracy: 0.9560 - val_loss: 0.1039 - val_accuracy: 0.9561 - 107ms/epoch - 7ms/step
Epoch 4/20
15/15 - 0s - loss: 0.1024 - accuracy: 0.9648 - val_loss: 0.0735 - val_accuracy: 0.9649 - 101ms/epoch - 7ms/step
Epoch 5/20
15/15 - 0s - loss: 0.0944 - accuracy: 0.9692 - val_loss: 0.0850 - val_accuracy: 0.9561 - 144ms/epoch - 10ms/step
Epoch 6/20
15/15 - 0s - loss: 0.0812 - accuracy: 0.9780 - val_loss: 0.0656 - val_accuracy: 0.9649 - 170ms/epoch - 11ms/step
Epoch 7/20
15/15 - 0s - loss: 0.0725 - accuracy: 0.9780 - val_loss: 0.0675 - val_accuracy: 0.9737 - 168ms/epoch - 11ms/step
Epoch 8/20
15/15 - 0s - loss: 0.0691 - accuracy: 0.9736 - val_loss: 0.0681 - val_accuracy: 0.9737 - 151ms/epoch - 10ms/step
Epoch 9/20
15/15 - 0s - loss: 0.0611 - accuracy: 0.9824 - val_loss: 0.0648 - val_accuracy: 0.9737 - 150ms/epoch - 10ms/step
Epoch 10/20
15/15 - 0s - loss: 0.0562 - accuracy: 0.9802 - val_loss: 0.0668 - val_accuracy: 0.9737 - 150ms/epoch - 10ms/step
Epoch 11/20
15/15 - 0s - loss: 0.0539 - accuracy: 0.9868 - val_loss: 0.0755 - val_accuracy: 0.9561 - 158ms/epoch - 11ms/step
Epoch 12/20
```

✓ 6s completed at 13:25

7.7.3

+ Code + Text

```
Epoch 8/20
15/15 - 0s - loss: 0.0742 - accuracy: 0.9780 - val_loss: 0.0599 - val_accuracy: 0.9649 - 111ms/epoch - 7ms/step
Epoch 9/20
15/15 - 0s - loss: 0.0679 - accuracy: 0.9758 - val_loss: 0.0755 - val_accuracy: 0.9561 - 103ms/epoch - 7ms/step
Epoch 10/20
15/15 - 0s - loss: 0.0646 - accuracy: 0.9802 - val_loss: 0.0636 - val_accuracy: 0.9561 - 97ms/epoch - 6ms/step
Epoch 11/20
15/15 - 0s - loss: 0.0578 - accuracy: 0.9802 - val_loss: 0.0589 - val_accuracy: 0.9737 - 111ms/epoch - 7ms/step
Epoch 12/20
15/15 - 0s - loss: 0.0636 - accuracy: 0.9758 - val_loss: 0.0786 - val_accuracy: 0.9649 - 113ms/epoch - 8ms/step
Epoch 13/20
15/15 - 0s - loss: 0.0486 - accuracy: 0.9846 - val_loss: 0.0569 - val_accuracy: 0.9649 - 111ms/epoch - 7ms/step
Epoch 14/20
15/15 - 0s - loss: 0.0569 - accuracy: 0.9802 - val_loss: 0.0582 - val_accuracy: 0.9561 - 117ms/epoch - 8ms/step
Epoch 15/20
15/15 - 0s - loss: 0.0496 - accuracy: 0.9824 - val_loss: 0.0600 - val_accuracy: 0.9649 - 114ms/epoch - 8ms/step
Epoch 16/20
15/15 - 0s - loss: 0.0406 - accuracy: 0.9868 - val_loss: 0.0547 - val_accuracy: 0.9825 - 118ms/epoch - 8ms/step
Epoch 17/20
15/15 - 0s - loss: 0.0444 - accuracy: 0.9824 - val_loss: 0.0576 - val_accuracy: 0.9561 - 121ms/epoch - 8ms/step
Epoch 18/20
15/15 - 0s - loss: 0.0405 - accuracy: 0.9890 - val_loss: 0.0643 - val_accuracy: 0.9561 - 133ms/epoch - 9ms/step
Epoch 19/20
15/15 - 0s - loss: 0.0374 - accuracy: 0.9890 - val_loss: 0.0546 - val_accuracy: 0.9649 - 121ms/epoch - 8ms/step
Epoch 20/20
15/15 - 0s - loss: 0.0328 - accuracy: 0.9890 - val_loss: 0.0555 - val_accuracy: 0.9825 - 154ms/epoch - 10ms/step
4/4 [==============================] - 0s 9ms/step - loss: 0.0555 - accuracy: 0.9825
Test Loss: 0.05546577274799347, Test Accuracy: 0.9824561476707458
```

✓ 6s completed at 14:40

7.7.3 & 7.7.4 Results of CNN model

CONCLUSION

In summary, this research paper explores the application of machine learning and deep learning techniques for breast cancer detection. Utilizing datasets like the Wisconsin Diagnostic Breast Cancer dataset, various algorithms from logistic regression to convolutional neural networks are evaluated for their effectiveness in distinguishing between benign and malignant tumors. Through systematic analysis, the study emphasizes the importance of early detection in combating breast cancer. By providing insights into the performance of different methods, this research aims to contribute to the improvement of diagnostic practices, ultimately leading to better patient care and outcomes.

FUTURE SCOPE: In the future, I will continue my research in this domain on more algorithms and their efficacies on different databases. Also, there is a need for research focusing on addressing the challenges associated with class imbalance and small sample sizes in medical datasets, as well as ensuring the ethical and responsible deployment of machine learning models in clinical settings.

Furthermore, I also aim to delve into further research regarding the efficacy of various drugs utilized in breast cancer management and treatment. Specifically, I am interested in investigating how these drugs interact with the female body, examining both their effectiveness in combating breast cancer and their potential side effects. Furthermore, I intend to explore any gender-specific differences in the response to these drugs, considering how they may impact males and females differently. By delving into these areas, I hope to contribute valuable insights that could inform more personalized and effective treatment strategies for breast cancer patients, ultimately leading to improved outcomes and quality of life for individuals affected by this disease.

RESOURCES USED AND REFERENCES:

1. World Health Organization (WHO) (13 March 2024)
2. National Institute of Health (NIH) Int J Sci Acad Res. 2021 Jan; 2(1): 3081–3086.
3. Breast Cancer Wisconsin (Diagnostic)
   https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic
4. Tutorials Point |
   https://www.tutorialspoint.com/machine_learning/machine_learning_logistic_regression.htm#:~:text=Logistic%20regression%20is%20a%20type,value%20between%200%20and%201.
5. Javatpoint | https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
6. MindManager | https://blog.mindmanager.com/decision-tree-diagrams/
7. Research Gate |
   https://www.researchgate.net/figure/Flow-chart-for-Naive-Bayesian-classification_fig2_330922872
8. JavaTPoint | https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
9. Research Gate |
   https://www.researchgate.net/figure/Schematic-diagram-of-an-Artificial-Neural-Network-ANN_fig4_352111703
10. Gfg | https://www.geeksforgeeks.org/introduction-convolution-neural-network/
11. Kaggle, Research Gate, National Institute of Health (NIH)
12. Colab, Anaconda (Jupyter notebook)
13. Analytics Vidhya
14. National Library of Medicine
15. Healthline
16. Scikit-learn
17. Taylor and Francis Online
18. MDPI
19. sciencedirectassets.com