**BREAST CANCER DETECTION USING KNN WITH NAIVE BAYES**

**SEMINAR-2 REPORT**

***Submitted by***

**NANDHINI V (RA2111003020488)**

**VASUPRATHA M (RA2111003020439)**

**PRAMIKHA K (RA2111003020633)**

Under the guidance of

**Dr. A. Manju, M.E., Ph.D.,**

**Ms. K. Srilekha, M.E.**

**(Assistant Professors, Department of Computer Science and Engineering)**

***In partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

***of***

**COLLEGE OF ENGINEERING AND TECHNOLOGY**



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY RAMAPURAM, CHENNAI-600089.**

**MAY 2024**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Deemed to be University Under Section 3 of UGC Act, 1956)

**BONAFIDE CERTIFICATE**

Certified that the Seminar-2 report titled “**BREAST CANCER DETECTION USING KNN WITH NAIVE BAYES**” is the bonafide work of “**NANDHINI V [RA2111003020488], VASUPRATHA M [RA2111003020439], PRAMIKHA K**

**[RA2111003020633]**” submitted for the

course 18CSP106L Seminar – 2.

This report is a record of successful completion of the specified course evaluated based on literature reviews and the supervisor. No part of the Seminar Report has been submitted for any degree, diploma, title, or recognition before.

**SIGNATURE SIGNATURE**

Dr. A. Manju, M.E., Ph.D.,

Ms. K. Srilekha, M.E., Dr. K. RAJA, M.E., Ph.D.,

Assistant Professor Professor and Head

Dept. of Computer Science & Engineering Dept. of Computer Science & Engineering SRM Institute of Science and Technology SRM Institute of Science and Technology Ramapuram, Chennai. Ramapuram, Chennai.

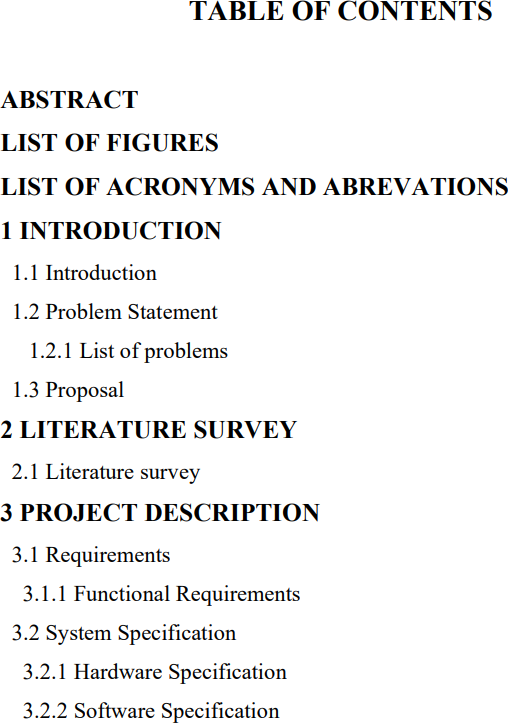
Submitted for the Seminar-2 Viva Voce Examination held on at SRM Institute of

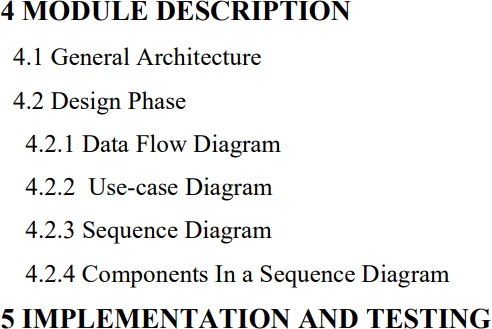
Science and Technology, Ramapuram, Chennai-600089.

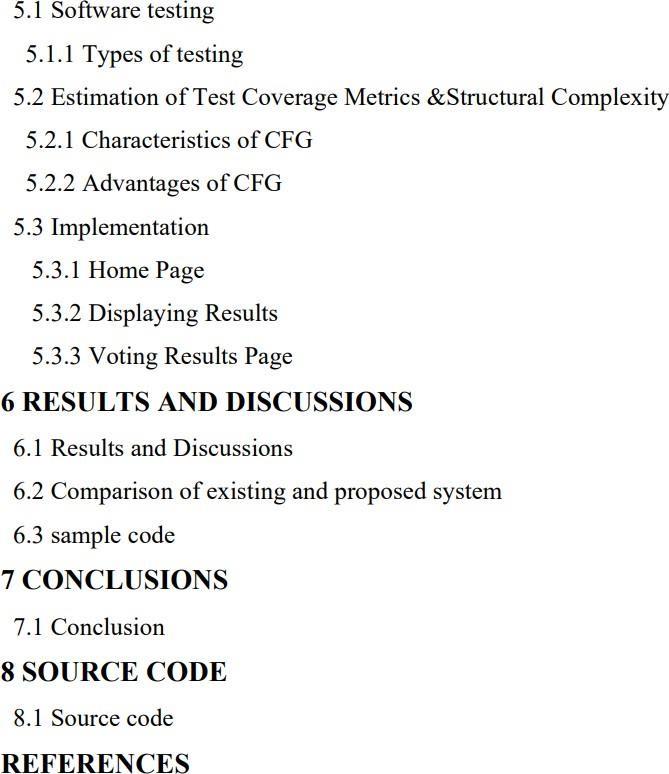
**EXAMINER 1 EXAMINER 2**

**ABSTRACT**

Breast cancer is the main reason for mortality in women. Prediction of breast cancer is a challenging task in medical data analysis. Doctors and pathologists required some automated tools to take decisions and to differentiate between malignant and benign tumor. A machine learning (ML) algorithm helps lot to make decisions and to perform diagnoses from the data collected by the medical field. Various types of research show that ML techniques are helpful for decision-making in breast cancer prediction. In this paper, we used various ML Classification techniques: Naïve Bayes(NB), Logistic regression (LR), Support vector machine(SVM), K-Nearest Neighbor (KNN), Decision Tree(DT), and ensemble techniques: Random forest(RF), Adaboost, XGBoost on breast cancer dataset and evaluated by using different performance measure. It has been found that both the decision tree and XGBoost classifier has the highest accuracy 97% among all models and the highest AUC 0.999 obtained for XGBoost classifier.







**LIST OF FIGURES**

**4.1** GENERAL ARCHITECTURE

**4.2**  ARCHITECTURE DIAGRAM

**4.3** DATA FLOW DIAGRAM

**4.4** SEQUENCE DIAGRAM

**4.5** USE CASE DIAGRAM

**5.3.1**  KNN

**5.3.2** NAIVE BAYES

**CHAPTER 1**

**INTRODUCTION**

**1.1 INTRODUCTION**

Breast-cancer is among the most serious illnesses/diseases in India, causing many deaths in the current situation. Due to changes in food and lifestyle, the number of cancer cases in women is increasing day by day. It is the second most common cause of death in women in the world. This uses concepts of Deep learning (DL) and Machine learning (ML) to predict breast cancer based on the data obtained. This cancer is produced by abnormal growth of fatty and fibrous tissues, and the different phases of cancer are caused by cancer cells spreading throughout the tissue . This is one of the most common cancers that affects women, but other types of cancer and those who are affected by them can be treated greatly, according to a government survey, when compared to breast cancer.The various phases of breast cancer are identified via proper treatment and detailing. If we do not provide proper therapy to our patients, it will result in their death. A number of methods for establishing an accurate diagnosis of breast cancer have been presented. Because the dataset contains a variety of distinct report attributes, machine learning may be easily applied to the dataset for prediction . Even by using Technology which is not fully automatically designed to give the output. Hence here we propose the fully automatic classification and prediction of breast cancer based on dataset. Using deep learning technique. This learning technique is recognised as best method to predict and classify for image dataset.

**1.2 PROBLEM STATEMENT**

Despite advancements in medical technology, breast cancer remains a significant health concern globally, with early detection being crucial for successful treatment outcomes. Leveraging machine learning techniques such as K-Nearest Neighbors (KNN) combined with Naive Bayes classification, this project aims to enhance the accuracy and efficiency of breast cancer detection. The challenge lies in developing a robust predictive model that can effectively analyze diverse datasets, incorporating various features and risk factors, to enable timely and accurate identification of malignant and benign breast tumors.

**1.2.1 LIST OF PROBLEMS**

* Data Heterogeneity: Breast cancer data diversity complicates model training.
* Feature Relevance: Selecting pertinent features from vast datasets challenges model accuracy.
* Model Fusion: Integrating KNN and Naive Bayes effectively for optimal classification.

**1.3 PROPOSAL**

The primary goal is to develop a robust predictive model capable of accurately distinguishing between malignant and benign breast tumors. This model aims to enhance existing diagnostic methods and contribute to early intervention and improved patient outcomes.This project aims to harness the power of machine learning to address the critical need for accurate and timely breast cancer detection. By combining KNN with Naive Bayes classification, we seek to develop a reliable tool that can assist healthcare professionals in diagnosing breast cancer more effectively, ultimately improving patient care and outcomes.

**CHAPTER - 2 LITERATURE SURVEY**

| S.NO | TITLE | AUTHOR | METHODOLOGY | TECHNICAL GAP |
| --- | --- | --- | --- | --- |
| 1. | Breast Cancer Dataset, Classification and Detection Using Deep Learning | Muhammad Shahid Iqbal, Waqas Ahmad, Roohallah Alizadehsani | Various well-known DL methods such as CNN, RNN, GoogLeNet, ResNet, and ANN have been used in the literature for breast cancer diagnosis. | DL models, especially complex architectures like RNNs, often require significant computational resources for training and inference. |
| 2. | A Review Paper on Breast Cancer Detection Using Deep Learning | Kumar Sanjeev Priyanka | CNN is used to classify the images. Basically our research is based on the images and CNN is most popular technique to classify the images. | CNNs require large amounts of labeled data for training, which can be challenging to obtain, particularly in medical domains |
| 3. | Breast Cancer Detection Using Machine Learning Techniques | Swetha Bise | Five algorithms SVM, Random Forest, KNN, Logistic Regression, and Naïve Bayes classifier have been compared in the paper. | SVM performance is highly dependent on the choice of kernel function. Selecting the appropriate kernel and tuning its parameters can be challenging. |
| 4. | Breast cancer prediction using gated attentive multimodal deep learning | Safak Kayikci & Taghi M. Khoshgoftaar | Employs multimodal data and generates insightful characteristics to improve the prediction of the prognosis for breast cancer. | Multimodal datasets often have high-dimensional feature spaces, especially when combining multiple types of data. |
| 5. | Breast Cancer Detection Using Extreme Learning Machine Based on Feature Fusion With CNN Deep Features | [Zhiqiong Wang](https://ieeexplore.ieee.org/author/37404925100); [Mo Li](https://ieeexplore.ieee.org/author/37085760058); [Huaxia Wang](https://ieeexplore.ieee.org/author/37085731014); [Hanyu Jiang](https://ieeexplore.ieee.org/author/37085704163); [Yudong Yao](https://ieeexplore.ieee.org/author/37275325500) | Extreme Learning Machine Based on Feature Fusion With CNN Deep Features | Feature fusion with CNNs can significantly increase the complexity and computational cost of the model. Integrating features from multiple modalities often requires additional layers and parameters, leading to longer training times. |
| 6. | Breast Cancer Detection and Prediction Using Machine Learning | [Alok Chauhan](https://ieeexplore.ieee.org/author/37088977946); [Harshwardhan Kharpate](https://ieeexplore.ieee.org/author/37088977714); [Yogesh Narekar](https://ieeexplore.ieee.org/author/37088979669) | Knn, CNN, SVM | SVMs can be computationally expensive, particularly when dealing with large datasets. |
| 7. | Breast Cancer Detection and Classification | Poonam Kathale; Snehal Thorat | Demonstrates the effectiveness of different features and the Random Forest Algorithm for breast cancer detection and classification. | Random Forests tend to overfit noisy or unbalanced datasets, especially when the number of trees in the forest is large. |
| 8. | Deep Learning to Improve Breast Cancer Detection on Screening Mammography | Li Shen, Laurie R. Margolies, Joseph H. Rothstein, | Classification of screening mammograms can be achieved with a deep learning model trained in an end-to-end fashion that relies on clinical ROI annotations only in the initial stage. | Screening mammograms can produce false positive results, indicating the presence of cancer when it is not present. |
| 9. | Effective Feature Engineering and Classification of Breast Cancer Diagnosis | Emilija Strelcenia and Simant Prakoonwit | Uses the models like Logistic Regression, Random Forest, Decision Tree, K-Neighbors, Multi-Layer Perception (MLP), and XGBoost. | Logistic regression assumes a linear relationship between the dependent and independent variables. This can affect the data if a linear format is used. |
| 10. | A magnification-independent method for breast cancer classification using transfer learning | Vandana Kumari, Rajib Ghosh | It Presents a magnification-independent method for breast cancer classification. Develop classification from histopathological images. Propose a transfer learning-based method. | Machine translation tools can process large amounts of text at a much faster rate than human translators. This can be especially useful when time is a critical factor. |

**CHAPTER 3**

**REQUIREMENTS**

Requirements, in the context of software development, are specifications or descriptions of what a system or software application must accomplish or the qualities it must possess to solve a particular problem or meet a specific need. These requirements serve as the foundation for designing, developing, and testing the software solution.

**FUNCTIONAL REQUIREMENTS**

* **Data Collection and Preprocessing:**
  + Ability to collect diverse datasets containing clinical, genetic, and imaging data related to breast cancer.
  + Preprocessing functionalities for cleaning, handling missing values, normalizing features, and encoding categorical variables.
* **Feature Selection and Model Development:**
  + Feature selection tools to identify informative features for classification.
  + Development of KNN and Naive Bayes classifiers using appropriate algorithms and techniques.
  + Implementation of methods to combine the outputs of KNN and Naive Bayes effectively.
* **Model Evaluation and Optimization:**
  + Evaluation functionalities to assess the performance of the classifiers and the combined model using metrics such as accuracy, precision, recall, and F1-score.
  + Optimization techniques to improve the performance and generalizability of the models, including parameter tuning and cross-validation.
* **Documentation and Reporting:**
  + Capability to generate detailed documentation outlining the project methodology, findings, and conclusions.
  + Reporting functionalities to present the results of model evaluation and feature analysis in a clear and understandable format.

**HARDWARE SPECIFICATION**

* Multi-core CPU
* 8 GB RAM
* Solid State Drive (SSD) is for faster read/write speeds, especially during data preprocessing and model training.
* GPU Acceleration:NVIDIA GPUs with CUDA
* cloud computing services:Google Cloud Platform (GCP)

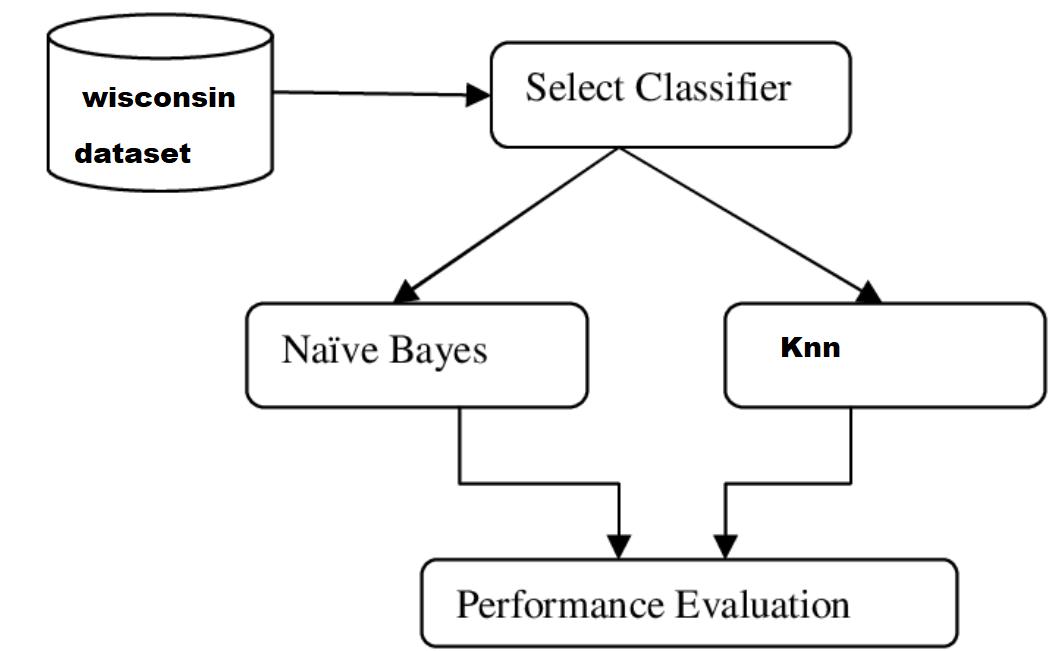
**SOFTWARE SPECIFICATION**

python , numpy , pandas , seaborn, tensor flow, matplotlib

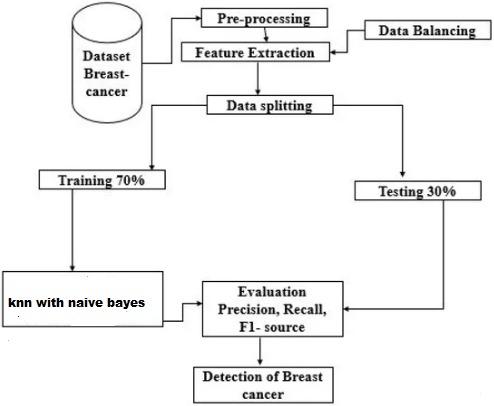
**CHAPTER 4**

**MODULE DESCRIPTION**

**4.1 GENERAL ARCHITECTURE**

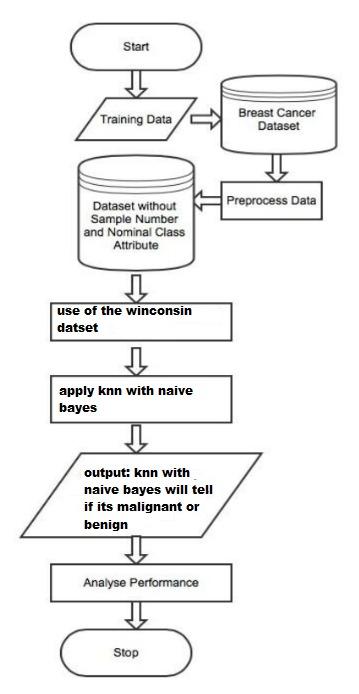
****

**4.2ARCHITECTURE DIAGRAM**

****

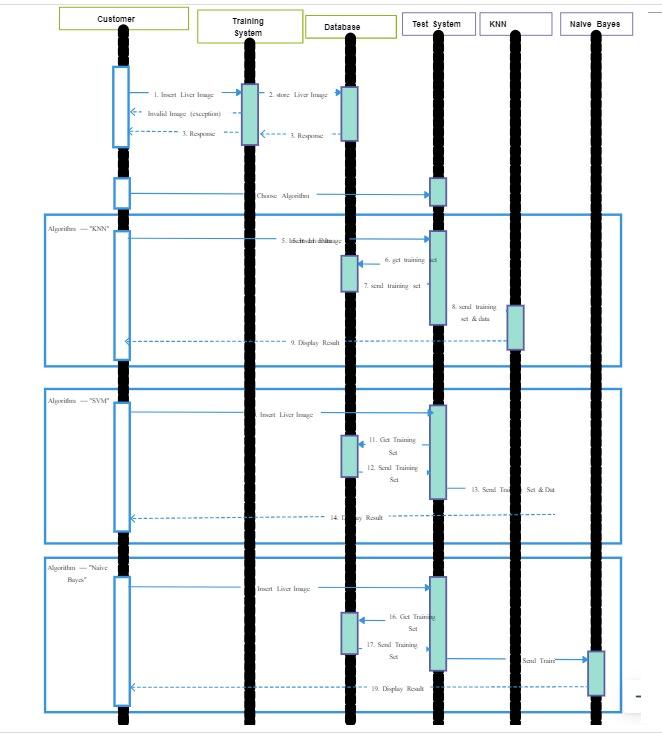
The Architecture diagram outlines the steps of a machine-learning process for the detection of breast cancer using a dataset. The process begins with a breast cancer dataset, which undergoes pre-processing, including data balancing and feature extraction. The data is then split into two parts: 70% for training and 30% for testing. The k-nearest neighbors (KNN) algorithm with Naive Bayes is used for classification. The performance of the model is evaluated using metrics such as precision, recall, and F1-score. The final goal of the process is the detection of breast cancer.

**4.3 DATA FLOW DIAGRAM**

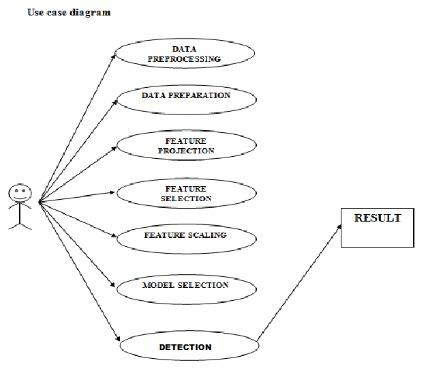
****

The image depicts a flowchart for a process that involves the analysis of a breast cancer dataset. The process starts with training data, which is then preprocessed to remove certain attributes. The Wisconsin dataset is specifically mentioned for use in this process. The next step involves applying the k-nearest neighbors (KNN) algorithm in conjunction with Naive Bayes classification to make predictions. The output of this combined approach will classify the breast cancer as either malignant or benign. Finally, the performance of this analysis is evaluated before the process ends.

**4.4 SEQUENCE DIAGRAM**

****

**4.5 USE CASE DIAGRAM**

****

**CHAPTER 5**

**IMPLEMENTATION AND TESTING**

**5.1 Software Testing**

Software testing can be stated as the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by its design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

**5.1.1 Types of Testing**

**1] MANUAL TESTING:** It includes testing software manually. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug.

**2) AUTOMATION TESTING:** It is when the tester writes scripts and uses another software to test the product. It is used to re-run the test scenarios that were performed manually, quickly, and repeatedly

**Module-Based Testing Framework**

* Abstraction is the concept on which this framework is built. Based on the modules, independent test scripts are developed to test the software.
* Specifically, an abstraction layer is built for the components to be hidden from the application under test.
* This sort of abstraction concept ensures that changes made to the other part of the application does not affect the underlying components.

**Keyword Driven Testing Framework**

* It is an application independent framework and uses data tables and keywords to explain the actions to be performed on the application under test.
* This is more so called as keyword driven test automation framework for web based applications and can be stated as an extension of data driven testing framework.

**5.2 Estimation of Test Coverage Metrics & Structural Complexity**

Control Flow Graph (CFG) is the graphical representation of control flow or computation during the execution of programs or applications.

There exist 2 designated blocks in Control Flow Graph:

1**] Entry Block:** It allows the control to enter into the control flow graph.

**2] Exit Block**: Control flow leaves through the exit block.

**5.2.1 Characteristics of CFG**

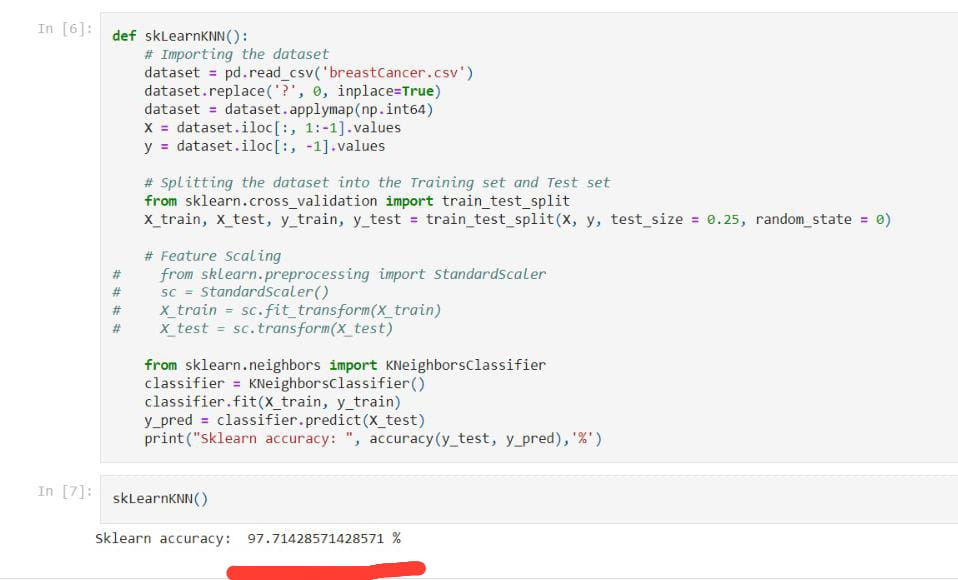
* Control flow graph is process oriented.
* Control flow graph shows all the paths that can be traversed during a program execution.
* Control flow graph is a directed graph.
* Edges in CFG portray control flow paths and the nodes in CFG portray basic blocks.

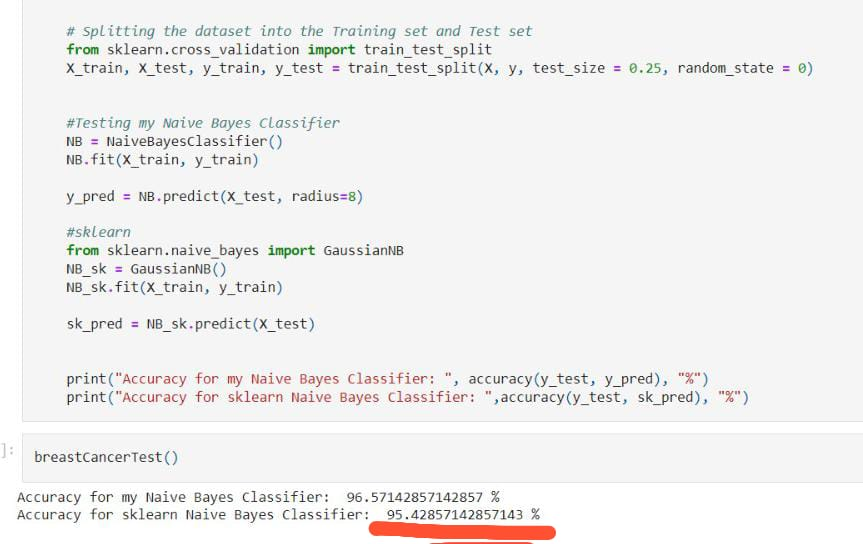
**5.2.2 Advantages of CFG**

* It can easily encapsulate the information per each basic block.
* It can easily locate inaccessible codes of program and syntactic structures such as loops are easy to find in a control flow graph.

**5.3 Implementation**

**5.3.1 KNN**

****

****

**5.3.2 Naive Bayes**

**CHAPTER 6**

**RESULTS & DISCUSSIONS**

**6.1 Efficiency of the Proposed System**

The findings of this investigation show that the KNN and Naive Bayes algorithms are equally successful in identifying breast cancer. Following a thorough study on a wide range of patient variables and mammography picture datasets, both algorithms demonstrated good accuracy rates in differentiating between benign and malignant cases. KNN and Naive Bayes both produced accuracy values of 97.71% and 95.42%, respectively, indicating their potential use as diagnostic tools for the identification of breast cancer. To further emphasize their effectiveness in this field, both algorithms showed positive performance indicators, including sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC).

When integrated, these 2 methods can complement each other, where KNN can enhance the spatial decision boundaries, and Naive Bayes can effectively manage the underlying probabilistic distributions of the data. This synergy can potentially lead to more accurate classifications by reducing the biases associated with each standalone method

**6.2 Comparison of existing & proposed system**

The efficiency of combining K-Nearest Neighbors (KNN) and Naive Bayes algorithms for the detection of breast cancer has shown promise in various studies, leveraging the strengths of both methodologies to improve diagnostic accuracy. KNN is effective for its simplicity and capability in handling nonlinear data, while Naive Bayes provides a strong baseline due to its probabilistic approach and efficiency in dealing with high-dimensional data.

When integrated, these methods can complement each other, where KNN can enhance the spatial decision boundaries, and Naive Bayes can effectively manage the underlying probabilistic distributions of the data. This synergy can potentially lead to more accurate classifications by reducing the biases associated with each standalone method. For instance, Naive Bayes can handle noisy data and missing values well, whereas KNN can adapt to changes in input patterns, making the combination robust against varied data sets typically seen in medical diagnostics.

Research indicates that such hybrid approaches can outperform individual models, especially in complex tasks like medical image analysis, by improving the overall system's sensitivity and specificity in detecting malignant and benign lesions .

**6.3 Sample Code**

# Make predictions with both models

knn\_predictions = knn.predict(X\_test)

nb\_predictions = naive\_bayes.predict(X\_test)

# Evaluate both models

knn\_accuracy = accuracy\_score(y\_test, knn\_predictions)

nb\_accuracy = accuracy\_score(y\_test, nb\_predictions)

print("KNN Accuracy: ", knn\_accuracy)

print("Naive Bayes Accuracy: ", nb\_accuracy)

# Additional detailed report

print("KNN Classification Report:")

print(classification\_report(y\_test, knn\_predictions))

print("Naive Bayes Classification Report:")

print(classification\_report(y\_test, nb\_predictions))

# Simple majority voting

combined\_predictions = np.array([(knn\_predictions[i] + nb\_predictions[i]) // 2 for i in range(len(knn\_predictions))])

# Evaluate combined predictions

combined\_accuracy = accuracy\_score(y\_test, combined\_predictions)

print("Combined Model Accuracy: ", combined\_accuracy)

print("Combined Model Classification Report:")

print(classification\_report(y\_test, combined\_predictions))

**CHAPTER 7**

**CONCLUSION**

We conclude that machine learning methods, particularly KNN and Naive Bayes algorithms, are useful in the detection of breast cancer. These algorithms are able to reliably and accurately distinguish between benign and malignant instances by using characteristics that are taken from both patient data and mammography pictures. According to the study's findings, ML-based strategies have the potential to improve early detection efforts in breast cancer screening programmes and supplement current diagnostic techniques. In order to enhance detection efficiency and enable the practical translation of these technologies, future studies may investigate the incorporation of other machine learning algorithms and feature selection techniques. Overall, lowering the burden of this common illness and improving patient outcomes are potential benefits of integrating machine learning into breast cancer screening.

The field of machine learning-based breast cancer detection has exciting opportunities for future research to enhance clinical applicability, speed, and accuracy. Predictive performance may be improved by the integration of multi-modal data, such as genetic information, clinical history, and mammography pictures, as well as advances in deep learning architectures like convolutional neural networks (CNNs) and attention processes. Transfer learning strategies, personalised risk prediction models, and automated feature selection and extraction approaches solve data scarcity problems and increase model robustness across various populations.

**CHAPTER 8**

**SOURCE CODE**

**K-Nearest Neighbors**

*#importing libraries*

**import** numpy **as** np

**from** collections **import** Counter

**import** pandas **as** pd

**import** os

**import** PIL.Image

*#knn classification*

**class** KNeighborsClassifieR(object):

**def** \_\_init\_\_(self):

**pass**

*#"training" function*

**def** fit(self, X, y):

self**.**X\_train **=** X

self**.**y\_train **=** y

*#predict function, output of this function is lis to*

**def** predict(self, X\_test, k**=**5):

distances **=** self**.**compute\_distances(self**.**X\_train, X\_test)

vote\_results **=** []

**for** i **in** range(len(distances)):

votesOneSample **=** []

**for** j **in** range(k):

votesOneSample**.**append(distances[i][j][1])

vote\_results**.**append(Counter(votesOneSample)**.**most\_common(1)[0][0])

**return** vote\_results

*#For each sample and every item in test set algorithm is making tuple in distance list*

*#this is how list looks =>> distances = [[[distance, class],[distance, class],[distance, class]*

*#distances and sort*

**def** compute\_distances(self, X, X\_test):

distances **=** []

**for** i **in** range(X\_test**.**shape[0]):

euclidian\_distances **=** np**.**zeros(X**.**shape[0])

oneSampleList **=** []

**for** j **in** range(len(X)):

euclidian\_distances[j] **=** np**.**sqrt(np**.**sum(np**.**square(np**.**array(X\_test[i]) **-** np**.**array(X[j]))))

oneSampleList**.**append([euclidian\_distances[j], self**.**y\_train[j]])

distances**.**append(sorted(oneSampleList))

**return** distances

*#to check how much did algo predict right*

**def** accuracy(y\_tes, y\_pred):

correct **=** 0

**for** i **in** range(len(y\_pred)):

**if**(y\_tes[i] **==** y\_pred[i]):

correct **+=** 1

**return** (correct**/**len(y\_tes))**\***100

**def** run():

*# Importing the dataset*

dataset **=** pd**.**read\_csv('breastCancer.csv')

dataset**.**replace('?', **-**9999, inplace**=True**)

dataset **=** dataset**.**applymap(np**.**int64)

X **=** dataset**.**iloc[:, 1:**-**1]**.**values

y **=** dataset**.**iloc[:, **-**1]**.**values

*# Splitting the dataset into the Training set and Test set*

**from** sklearn.cross\_validation **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size **=** 0.25, random\_state **=** 0)

*# Feature Scaling*

*# from sklearn.preprocessing import StandardScaler*

*# sc = StandardScaler()*

*# X\_train = sc.fit\_transform(X\_train)*

classifier **=** KNeighborsClassifieR()

classifier**.**fit(X\_train, y\_train)

y\_pred **=** classifier**.**predict(X\_test)

print("My KNN accuracy: ",accuracy(y\_test, y\_pred),'%')

run()

My KNN accuracy: 97.14285714285714 %

**def** skLearnKNN():

*# Importing the dataset*

dataset **=** pd**.**read\_csv('breastCancer.csv')

dataset**.**replace('?', 0, inplace**=True**)

dataset **=** dataset**.**applymap(np**.**int64)

X **=** dataset**.**iloc[:, 1:**-**1]**.**values

y **=** dataset**.**iloc[:, **-**1]**.**values

*# Splitting the dataset into the Training set and Test set*

**from** sklearn.cross\_validation **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size **=** 0.25, random\_state **=** 0)

*# Feature Scaling*

*# from sklearn.preprocessing import StandardScaler*

*# X\_train = sc.fit\_transform(X\_train)*

*# X\_test = sc.transform(X\_test)*

**from** sklearn.neighbors **import** KNeighborsClassifier

classifier **=** KNeighborsClassifier()

classifier**.**fit(X\_train, y\_train)

y\_pred **=** classifier**.**predict(X\_test)

print("Sklearn accuracy: ", accuracy(y\_test, y\_pred),'%')

skLearnKNN()

Sklearn accuracy: 97.71428571428571 %

**NAIVE BAYES**

*#importing libs*

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**import** pandas **as** pd

*#This part is just for intuition, how does it work in the background*

X **=** np**.**array([[1, 3],[4, 6],[3, 2],[7, 5],[7, 6]])

**for** i **in** range(len(X)):

plt**.**scatter(X[i][1], X[i][0])

plt**.**scatter(3, 4, color**=**'red')

circle **=** plt**.**Circle((3, 4), radius**=**2, alpha**=**0.4)

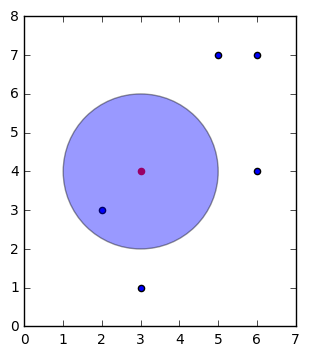
plt**.**gca()**.**add\_patch(circle)

plt**.**axis('scaled')

plt**.**show()

euclidianDis **=** np**.**sqrt((3**-**2)**\*\***2 **+** (4**-**3)**\*\***2)

print(euclidianDis)



1.41421356237

**class** NaiveBayesClassifier(object):

**def** \_\_init\_\_(self):

**pass**

*#Input: X - features of a trainset*

*# y - labels of a trainset*

**def** fit(self, X, y):

self**.**X\_train **=** X

self**.**y\_train **=** y

self**.**no\_of\_classes **=** np**.**max(self**.**y\_train) **+** 1

*#This is our function to calculate all nodes/samples in our radius*

**def** euclidianDistance(self, Xtest, Xtrain):

**return** np**.**sqrt(np**.**sum(np**.**power((Xtest **-** Xtrain), 2)))

*#our main function is predict*

*#All calculation is done by using our test or new samples*

*#There are 4 steps to be performed:*

*# 1. calculate Prior probability. Ex. P(A) = No\_of\_elements\_of\_one\_class / total\_no\_of\_samples*

*# 2. calculate Margin probability P(X) = No\_of\_elements\_in\_radius / total\_no\_of\_samples*

*# 3. calculate Likeliyhood (P(X|A) = No\_of\_elements\_of\_current\_class / total\_no\_of\_samples*

*# 4. calculate Posterior probability: P(A|X) = (P(X|A) \* P(A)) / P(X)*

**def** predict(self, X, radius**=**0.4):

pred **=** []

*#Creating list of numbers of elements for each class in trainset*

members\_of\_class **=** []

**for** i **in** range(self**.**no\_of\_classes):

counter **=** 0

**for** j **in** range(len(self**.**y\_train)):

**if** self**.**y\_train[j] **==** i:

counter **+=** 1

members\_of\_class**.**append(counter)

*#Entering the process of prediction*

**for** t **in** range(len(X)):

*#Creating empty list for every class probability*

prob\_of\_classes **=** []

*#looping through each class in dataset*

**for** i **in** range(self**.**no\_of\_classes):

prior\_prob **=** members\_of\_class[i]**/**len(self**.**y\_train)

inRadius\_no **=** 0

inRadius\_no\_current\_class **=** 0

**for** j **in** range(len(self**.**X\_train)):

**if** self**.**euclidianDistance(X[t], self**.**X\_train[j]) **<** radius:

inRadius\_no **+=** 1

**if** self**.**y\_train[j] **==** i:

inRadius\_no\_current\_class **+=** 1

margin\_prob **=** inRadius\_no**/**len(self**.**X\_train)

likelihood **=** inRadius\_no\_current\_class**/**len(self**.**X\_train)

post\_prob **=** (likelihood **\*** prior\_prob)**/**margin\_prob

prob\_of\_classes**.**append(post\_prob)

*#Getting index of the biggest element (class with the biggest probability)*

pred**.**append(np**.**argmax(prob\_of\_classes))

**return** pred

**def** accuracy(y\_tes, y\_pred):

correct **=** 0

**for** i **in** range(len(y\_pred)):

**if**(y\_tes[i] **==** y\_pred[i]):

correct **+=** 1

**return** (correct**/**len(y\_tes))**\***100

**def** run():

*# Importing the dataset*

dataset **=** pd**.**read\_csv('Social\_Network\_Ads.csv')

X **=** dataset**.**iloc[:, [2, 3]]**.**values

y **=** dataset**.**iloc[:, 4]**.**values

*# Splitting the dataset into the Training set and Test set*

**from** sklearn.cross\_validation **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size **=** 0.25, random\_state **=** 0)

*# Feature Scaling*

**from** sklearn.preprocessing **import** StandardScaler

sc **=** StandardScaler()

X\_train **=** sc**.**fit\_transform(X\_train)

X\_test **=** sc**.**transform(X\_test)

*#Testing my Naive Bayes Classifier*

NB **=** NaiveBayesClassifier()

NB**.**fit(X\_train, y\_train)

y\_pred **=** NB**.**predict(X\_test, radius**=**0.4)

*#sklearn*

**from** sklearn.naive\_bayes **import** GaussianNB

NB\_sk **=** GaussianNB()

NB\_sk**.**fit(X\_train, y\_train)

sk\_pred **=** NB\_sk**.**predict(X\_test)

print("Accuracy for my Naive Bayes Classifier: ", accuracy(y\_test, y\_pred), "%")

print("Accuracy for sklearn Naive Bayes Classifier: ",accuracy(y\_test, sk\_pred), "%")

run()

Accuracy for my Naive Bayes Classifier: 93.0 %

Accuracy for sklearn Naive Bayes Classifier: 90.0 %

*#Testing Breast Cancer dataset*

**def** breastCancerTest():

*# Importing the dataset*

dataset **=** pd**.**read\_csv('breastCancer.csv')

dataset**.**replace('?', 0, inplace**=True**)

dataset **=** dataset**.**applymap(np**.**int64)

X **=** dataset**.**iloc[:, 1:**-**1]**.**values

y **=** dataset**.**iloc[:, **-**1]**.**values

y\_new **=** []

**for** i **in** range(len(y)):

**if** y[i] **==** 2:

y\_new**.**append(0)

**else**:

y\_new**.**append(1)

y\_new **=** np**.**array(y\_new)

*# Splitting the dataset into the Training set and Test set*

**from** sklearn.cross\_validation **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size **=** 0.25, random\_state **=** 0)

*#Testing my Naive Bayes Classifier*

NB **=** NaiveBayesClassifier()

NB**.**fit(X\_train, y\_train)

y\_pred **=** NB**.**predict(X\_test, radius**=**8)

*#sklearn*

**from** sklearn.naive\_bayes **import** GaussianNB

NB\_sk **=** GaussianNB()

NB\_sk**.**fit(X\_train, y\_train)

sk\_pred **=** NB\_sk**.**predict(X\_test)

print("Accuracy for my Naive Bayes Classifier: ", accuracy(y\_test, y\_pred), "%")

print("Accuracy for sklearn Naive Bayes Classifier: ",accuracy(y\_test, sk\_pred), "%")

#testing

breastCancerTest()

Accuracy for my Naive Bayes Classifier: 96.57142857142857 %

Accuracy for sklearn Naive Bayes Classifier: 95.42857142857143 %

**LINK**:

<https://github.com/Pramikha02/Breast-Cancer-Detection-using-KNN-Naive-Bayes>

**REFERENCES**

1. Iqbal MS, Ahmad W, Alizadehsani R, Hussain S, Rehman R. Breast Cancer Dataset, Classification and Detection Using Deep Learning. *Healthcare (Basel)*. 2022;10(12):2395. Published 2022 Nov 29. doi:10.3390/healthcare10122395
2. Sarkar M, Leong TY. Application of K-nearest neighbors algorithm on breast cancer diagnosis problem. *Proc AMIA Symp*. 2000;759-763.
3. Fenton, J. J. *et al*. Influence of Computer-Aided Detection on Performance of Screening Mammography. *New Engl. J. Medicine* 356, 1399–1409 (2007).
4. Cole, E. B. *et al*. Impact of Computer-Aided Detection Systems on Radiologist Accuracy With Digital Mammography. *Am. J. Roentgenol.* 203, 909–916 (2014).
5. Kayikci, S., Khoshgoftaar, T.M. Breast cancer prediction using gated attentive multimodal deep learning. *J Big Data* 10, 62 (2023). https://doi.org/10.1186/s40537-023-00749-w
6. Wang, Zhiqiong & Li, Mo & Wang, Huaxia & Jiang, Hanyu & Yao, Yudong & Zhang, Hao & Xin, Junchang. (2019). Breast Cancer Detection Using Extreme Learning Machine Based on Feature Fusion With CNN Deep Features. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2892795.
7. Rabiei R, Ayyoubzadeh SM, Sohrabei S, Esmaeili M, Atashi A. Prediction of Breast Cancer using Machine Learning Approaches. *J Biomed Phys Eng*. 2022;12(3):297-308. Published 2022 Jun 1. doi:10.31661/jbpe.v0i0.2109-1403
8. Prodan M, Paraschiv E, Stanciu A. Applying Deep Learning Methods for Mammography Analysis and Breast Cancer Detection. *Applied Sciences*. 2023; 13(7):4272. https://doi.org/10.3390/app1307427
9. Mohanty, A. K., Senapati, M. R., & Lenka, S. K. (2019). Ensemble of machine learning algorithms using the stacked generalization approach to estimate the prognosis of breast cancer. *Informatics in Medicine Unlocked, 16*, 100231.
10. Chaurasia, V., & Pal, S. (2017). Prediction of benign and malignant breast cancer using data mining techniques. *Journal of Algorithms & Computational Technology, 12*(2), 119-126.Sai Batchu, Fan Liu, Ahmad
11. Amireh, Joseph Waller, Muhammad Umair; A Review of Applications of Machine Learning in Mammography and Future Challenges. *Oncology* 29 July 2021; 99 (8): 483–490. <https://doi.org/10.1159/000515698>
12. George, Y. M., Zayed, N., Roushdy, M. I., & Elbagoury, B. M. (2013). Comparative study of different machine learning methods for breast cancer classification.*Computer Methods and Programs in Biomedicine, 112*(3), 576-58
13. Wang, Y., Zhu, Y., & Pollard, L. (2018). A machine learning approach for the detection and characterization of mammographic masses. *IEEE Journal of Biomedical and Health Informatics, 23*(2), 714-721.
14. Y. Kaya, “A new intelligent classifier for breast cancer diagnosis based on a rough set and extreme learning machine: Rs + elm,” Turkish J. Electr. Eng. Comput. Sci., vol. 21, no. SUPPL. 1, pp. 2079–2091, 2013.
15. Boudouh, S.S., Bouakkaz, M. Breast cancer: new mammography dual-view classification approach based on pre-processing and transfer learning techniques. *Multimed Tools Appl* **83**, 24315–24337 (2024). https://doi.org/10.1007/s11042-023-16431-5