**VISTULA UNIVERSITY**

**The Faculty of Computer Engineering, Graphic Design and Architecture**

**Program of study Computer Science**

**Arnold Mukucha**

Student’s number : 55322

**HEALTHCARE PREDICTIVE MODELING WITH MACHINE LEARNING**

The bachelor’s thesis

written under the supervision of

Dr. Igor Protasowicki

Warsaw, 2024

# List of Symbols and Abbreviations

|  |  |
| --- | --- |
| ML | Machine Learning |
| *HTML* | Hypertext Transfer Protocol |
| *CSS* | Cascading Style Sheets |
| AI | Artificial Intelligence |
|  |  |

# Table Of Contents

[List of Symbols and Abbreviations](#_gjdgxs)

[Table Of Contents](#_17dp8vu)

Chapter 1: Introduction

1.1 Introduction

1.2 Background of the study

1.3 Problem definition

1.4 Aim

1.5 Objectives

1.6 Instruments and methods

1.7 Delimitations and Limitations of the research

1.8 Feasibility Analysis

1.9 Justification and rationale

1.10 Conclusion

Chapter 2: Analysis Phase

2.1 Introduction

2.2. Information gathering

2.3 Weaknesses of the existing systems

2.4 Process analysis

2.5 Data Analysis

2.6 Weaknesses of current system

2.9 Conclusion

Chapter 3: Design Phase

3.1 Introduction

3.2.1 Context Diagram

3.3 Architectural design

3.4 Physical design

3.7 Interface design

Comments Page

3.9 Security design

3.9.1 Physical security

3.9.2 Network security

3.9.3 Operational security

3.10 Conclusion

Chapter 4: Implementation Phase

4.1 Introduction

4.5 Conclusion

Appendices

# Chapter 1: Introduction

## 1.1 Introduction

Breast cancer has become a growing concern worldwide, with a significant increase in cases observed in recent years. Despite extensive research, finding a definitive solution to this rising incidence remains elusive. The mortality rate associated with breast cancer is alarmingly high compared to other types of cancer. In Turkey, the incidence rate is 7.32 cases per 100,000 individuals, and breast cancer-related deaths account for 24.1% of reported carcinoma cases in women. The United States has a relatively higher incidence rate at 91.6 cases per 100,000 individuals.(Zaim Gökbay, İnci 2007)

Early detection plays a crucial role in saving lives from breast cancer, and various guidelines and methods are available for detecting the disease in asymptomatic women. These include breast self-exams, clinical breast exams, breast awareness, imaging tests such as mammograms, MRI, breast ultrasound, PET scan, CT scan, and biopsies. Among these methods, screening mammography is considered the most reliable for early detection. However, radiologists often face challenges in identifying certain lesions on mammograms due to the complex and varying densities of breast tissue structures. Image processing techniques are continuously being improved to enhance detection.(Zaim Gökbay, İnci 2007)

This study proposes a system specifically designed to detect malignant masses on mammograms. The research focuses on investigating the behavior of Median at different scales and utilizes a median filter to achieve this. Suspicious regions are then segmented through adaptive thresholding.

Breast cancer ranks as the primary cancer among women in India and second in the United States. While breast cancer can also affect men, its prevalence is relatively lower in males. The majority of breast cancer cases originate from epithelial cells within breast tissues, with ductal carcinoma in situ (DCIS) and invasive ductal carcinoma (IDC) being considered malignant cancers requiring intensive treatment and care. The chapter primarily discusses breast cancer and the utilization of machine learning (ML) applications, covering various issues related to prognosis, early detection, and diagnostic techniques employing diverse ML algorithms.(Aswathy M. A., Jagannath Mohan 2020)

Breast cancer is one of the most prevalent diseases causing fatalities in women worldwide. Suspected cases typically undergo periodic examinations utilizing various diagnostic tools and tests, including digital mammograms, infrared thermography, MRI, ultrasound, histological images, and microwave images. Recent advancements in hardware and software have facilitated the application of different techniques, particularly machine learning, to achieve high-quality results in breast cancer detection and diagnosis. The study presents a comprehensive survey of accurate techniques used in breast cancer detection and diagnosis, discussing both commercial and non-commercial hardware and software, along with their advantages and disadvantages in detecting and diagnosing breast lesions. While several techniques have emerged to assist in breast cancer detection and diagnosis, no single modality can achieve perfect accuracy. The development of a complete system capable of handling different modalities and delivering 100% accuracy remains a challenge due to the diverse structures observed in breast cancer.(R M. Al-Tam , S M. Narangale 2021**)**

## 1.2 Background of the study

In the study "Breast Cancer Detection and Diagnosis Using Machine Learning" conducted by Riyadh M. Al-Tam and Sachin M. Narangale in 2021, it was reaffirmed that breast cancer remains a prevalent disease causing significant mortality among women worldwide. The researchers emphasized the diverse structure of the disease, which complicates the detection and diagnosis process. To aid in addressing this challenge, numerous hardware and software solutions have been developed to assist in the analysis and classification of breast cancer.

The study primarily focused on providing a comprehensive review of the most recent techniques in breast cancer detection and diagnosis based on machine learning. The researchers discussed a group of medical modalities, including mammography, MRI, and ultrasound, highlighting their respective strengths and limitations. Additionally, they summarized a set of commercial and non-commercial Computer-Aided Diagnosis (CAD) systems, outlining their advantages and disadvantages. The CAD system stages for analyzing and classifying breast cancer were also examined in detail, which encompassed pre-processing, segmentation, feature extraction and selection, and classification. The authors presented the most commonly used methods in each stage.

Furthermore, the study discussed a collection of breast cancer datasets that contained medical images or extracted features, providing valuable resources for researchers in this field. Through a systematic comparison of recent machine learning approaches in medical images, the researchers demonstrated the promising results achieved by advances in machine learning methods. These advancements have the potential to aid radiologists and physicians in the detection and diagnosis of breast cancer.( Riyadh M. Al-Tam and Sachin M. Narangale 2021)

However, the researchers concluded that while many techniques have been developed to assist in breast cancer detection and diagnosis, no single modality can achieve perfect accuracy on its own. They also highlighted the ongoing challenge of developing a complete system capable of handling different modalities and achieving 100% accuracy. The diverse structure of breast cancer and the utilization of different modalities contribute to this challenge. As a result, the researchers emphasized the necessity for further improvements to effectively combat the rising risk and protect patients from this deadly disease.

Another study "Machine Learning Techniques in Breast Cancer Detection" conducted by Zaim Gökbay and İnci in 2007, it was highlighted that breast cancer is a major cause of cancer-related mortality among women. Early detection and accurate diagnosis are crucial factors in improving long-term survival rates for patients. The study aimed to develop a program for detecting malignant masses in digitized mammograms using machine learning techniques.

The researchers focused on the utilization of the median filter to enhance rounded convex objects in mammograms. They investigated the behavior of the median filter at different scales and determined the optimum size for detecting masses. Additionally, gray level information was incorporated into the output of the median filter to improve the enhancement of mass regions. Suspicious regions were then segmented using an adaptive threshold level in the enhanced image.( Zaim Gökbay , İnci 2007)

The program developed in the study was evaluated using two independent datasets. Performance evaluation was conducted using the free response observer characteristic methodology. The proposed method was tested on a dataset consisting of 100 images from 10 malignant cases and 20 normal cases. The results suggested that the program could serve as a valuable tool for radiologists as a second reader in mammographic screening.

The researchers also discussed future directions for their work. They acknowledged that their study had limitations, particularly the limited case set used. They proposed expanding the clinical case set to develop a more comprehensive computer-aided detection (CAD) system. This could involve incorporating feature extraction methods, a neural network system, and a reporting algorithm. The feature extraction method would involve segmenting suspect regions and characterizing them using various types of features. A neural network system could be trained using this dataset for improved decision-making. Additionally, patient-specific statistical risk factor features, such as age, race, medical history, and lifestyle factors, could be incorporated to generate detailed reports on a patient's risk of developing breast cancer if no detection is performed using image processing methods.

In conclusion, the study demonstrated the effectiveness of machine learning techniques, particularly the utilization of the median filter, in detecting malignant masses in mammograms. The proposed program showed promising results and could serve as a valuable tool for radiologists in mammographic screening. Further research and development are needed to expand the study's case set, incorporate advanced feature extraction methods and neural networks, and consider patient-specific risk factors to enhance the accuracy and capabilities of breast cancer detection systems.( Zaim Gökbay , İnci 2007)

## 1.3 Problem definition

Breast cancer is a significant global health concern and a leading cause of death among women. Early detection plays a crucial role in improving patient outcomes and reducing mortality rates. While various imaging modalities and diagnostic techniques are available, the accurate and timely detection of breast cancer remains a challenge. Traditional diagnostic methods often require extensive manual analysis by radiologists, leading to subjective interpretations and potential human errors. There is a need for an automated and reliable system that can effectively detect breast cancer using machine learning classifiers.

Machine learning techniques have shown promise in various medical applications, including breast cancer detection. However, the development of an accurate and robust machine learning classifier specifically tailored for breast cancer detection poses several challenges. One key challenge is the diversity and complexity of breast cancer patterns and manifestations, making it difficult to identify consistent and reliable features for classification. Additionally, the availability of large and diverse datasets containing labeled breast cancer images for training and validation purposes is limited.

Furthermore, the performance and generalizability of machine learning classifiers heavily depend on the selection and extraction of relevant features from medical images. Identifying the most discriminative features that can effectively differentiate between benign and malignant breast lesions is crucial for developing a reliable classifier. Additionally, the classifier should be able to handle various types of imaging modalities, such as mammography, MRI, and ultrasound, and provide accurate predictions across different patient populations.

Addressing these challenges and developing an efficient and accurate machine learning classifier for breast cancer detection is of utmost importance. Such a classifier can significantly aid radiologists in their decision-making process, improve early detection rates, reduce false-positive and false-negative results, and ultimately contribute to saving lives.

## 1.4 Aim

1. Develop an accurate and reliable machine learning classifier specifically designed for breast cancer detection.

- The primary aim is to create a classifier that can effectively differentiate between benign and malignant breast lesions with high accuracy.

- The classifier should be able to handle different imaging modalities, such as mammography, MRI, and ultrasound, and provide consistent predictions.

2. Improve early detection rates and reduce false-positive and false-negative results.

- By leveraging machine learning techniques, the aim is to enhance the sensitivity and specificity of breast cancer detection.

- The classifier should minimize the occurrence of false-positive results, which can lead to unnecessary interventions and anxiety for patients.

- Additionally, the classifier should reduce false-negative results, ensuring that malignant breast lesions are not missed during the detection process.

3. Enhance the efficiency and speed of breast cancer diagnosis.

- The proposed classifier aims to automate the detection process, reducing the dependency on manual analysis and interpretation by radiologists.

- By leveraging machine learning algorithms, the classifier should provide quick and accurate assessments, allowing for timely diagnoses and treatment planning.

4. Ensure the generalizability and adaptability of the classifier.

- The aim is to develop a classifier that can perform reliably across diverse patient populations and healthcare settings.

- The classifier should be trained and validated using large and diverse datasets, encompassing various breast cancer subtypes, age groups, and ethnicities.

5. Facilitate seamless integration into clinical workflows.

- The proposed classifier aims to be user-friendly and compatible with existing healthcare systems.

- It should provide clear and interpretable results that can be readily incorporated into the decision-making process of radiologists and healthcare professionals.

6. Contribute to improving patient outcomes and reducing breast cancer mortality rates.

- Ultimately, the goal of this proposed topic is to enhance the early detection and accurate diagnosis of breast cancer, leading to improved patient outcomes and reduced mortality rates.

- By providing reliable and efficient breast cancer detection using a machine learning classifier, the aim is to contribute to saving lives and improving the overall prognosis of breast cancer patients.

## 1.5 Objectives

1. Design and develop a user-friendly web-based application for breast cancer detection.

- The application should have an intuitive interface that allows users to input various parameters and interact with the system easily.

- The app should be accessible via standard web browsers, ensuring compatibility across different devices and platforms.

2. Implement an input module that enables users to input relevant parameters for breast cancer detection.

- The module should provide options for users to input demographic information, medical history, genetic risk factors, and any other relevant data that can contribute to accurate detection.

- It should also allow users to upload or input medical images, such as mammograms or MRI scans, for analysis.

3. Develop a backend system that integrates machine learning algorithms for breast cancer detection.

- Implement machine learning models that can process the input parameters and medical images to predict the likelihood of breast cancer.

- Explore and evaluate different machine learning techniques, such as ensemble methods or deep learning architectures, to determine the most accurate and efficient approach.

4. Incorporate feature extraction and selection methods to identify relevant features from the input parameters and medical images.

- Explore techniques to extract key features from demographic data, medical history, and genetic risk factors that can contribute to breast cancer detection.

- Develop methods to extract meaningful features from medical images, such as texture analysis or shape descriptors, to enhance the accuracy of the detection process.

5. Train the machine learning models using a labeled dataset that encompasses various breast cancer subtypes and diverse patient populations.

- Collect and curate a well-annotated dataset that includes input parameters and corresponding medical images.

- Utilize the dataset to train the machine learning models, optimizing their performance through cross-validation and hyperparameter tuning.

6. Evaluate the performance of the web-based app and machine learning models using appropriate metrics.

- Conduct rigorous testing and validation to assess the accuracy, sensitivity, specificity, and overall performance of the app.

- Compare the results obtained from the web-based app with established diagnostic methods to evaluate its efficacy and reliability.

7. Incorporate security measures to protect sensitive patient data and ensure compliance with privacy regulations.

- Implement robust security protocols to safeguard user inputs, including encryption techniques and access controls.

- Adhere to relevant privacy regulations, such as HIPAA or GDPR, to ensure the protection of patient information.

8. Continuously improve the web-based app based on user feedback and advancements in machine learning.

- Gather feedback from users, including healthcare professionals and patients, to enhance the app's usability and functionality.

- Stay up to date with the latest developments in machine learning algorithms and techniques, incorporating relevant advancements to improve the accuracy and efficiency of the app.

## 1.6 Instruments and methods

**1. Programming Languages:**

- Python: Python is widely used for machine learning and web development due to its extensive libraries and frameworks, such as Flask or Django.

- HTML/CSS: These languages are essential for creating the user interface and designing the web pages.

- JavaScript: JavaScript can be used for client-side interactivity and enhancing the user experience.

**2. Web Frameworks:**

- Streamlite: Streamlit is a Python library designed to help data scientists and machine learning engineers quickly create web apps from their Python code.

- Streamlit is ideal for creating data dashboards, reports, interactive machine learning models, and other data science applications that you want to share with a wider audience.

**3. Machine Learning Libraries:**

- scikit-learn: scikit-learn provides a comprehensive set of tools for machine learning, including classification algorithms, feature extraction methods, and model evaluation metrics.

- TensorFlow : These deep learning frameworks offer advanced capabilities for training and deploying deep neural networks for image analysis and classification tasks.

- OpenCV: OpenCV is a widely used computer vision library that can be utilized for image preprocessing, feature extraction, and manipulation.

**4. Cloud Services and Hosting:**

- **Streamlit Cloud (Free for simple apps):** Streamlit offers a free tier for deploying simple apps. It's a quick and easy option for getting started.

**5. Version Control:**

- Git: Git is a widely used version control system for tracking changes in the source code, enabling collaboration, and managing the project's development history.

**6. Integrated Development Environment (IDE):**

- Visual Studio Code: Visual Studio Code is a lightweight and versatile IDE that supports various programming languages and offers extensions for web development and machine learning.

**Algorithms/Techniques:**

1. Support Vector Classifier (SVC): SVC is a powerful algorithm for binary classification that finds an optimal hyperplane to separate different classes by maximizing the margin between them.

2. Logistic Regression: Logistic Regression is a widely used algorithm that models the probability of an instance belonging to a particular class using a logistic function, making it suitable for binary classification tasks.

3. K-Nearest Neighbor Classifier (KNN): KNN is a non-parametric algorithm that classifies instances based on their proximity to the k nearest neighbors in the training dataset. It can handle both binary and multi-class classification problems.

4. Naive Bayes Classifier: Naive Bayes is a probabilistic algorithm that applies Bayes' theorem with the assumption of independence among features. It works well with high-dimensional data and is computationally efficient.

5. Decision Tree Classifier: Decision Tree is a versatile algorithm that builds a tree-like model to make decisions based on feature values. It can handle both classification and regression tasks and provides interpretable results.

6. Random Forest Classifier: Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It reduces overfitting and improves generalization by aggregating the results of individual trees.

7. Adaboost Classifier: Adaboost is an ensemble learning technique that sequentially trains weak classifiers on different weighted versions of the training dataset. It combines their predictions to create a strong classifier.

8. XGBoost Classifier: XGBoost (Extreme Gradient Boosting) is a gradient boosting algorithm known for its high performance and accuracy. It combines multiple weak learners to create a strong classifier, using gradient-based optimization techniques.

**Information Gathering Tools:**

1. Medical Research Papers and Journals:

- PubMed: PubMed is a widely used database for accessing biomedical literature, including research papers and studies related to breast cancer detection, machine learning, and medical imaging.

- IEEE Xplore or ACM Digital Library: These platforms provide access to conference proceedings and scholarly articles in the field of computer science, machine learning, and medical imaging.

2. Clinical Guidelines and Protocols:

- National Comprehensive Cancer Network (NCCN): NCCN provides evidence-based guidelines for breast cancer diagnosis, treatment, and follow-up, which can inform the development of the web-based app.

- American Cancer Society (ACS): ACS offers clinical guidelines and recommendations for breast cancer screening, detection, and management.

3. Datasets and Repositories:

- The Cancer Imaging Archive (TCIA): TCIA is a valuable resource for accessing publicly available medical imaging datasets, including mammograms, MRI scans, and ultrasound images related to breast cancer.

- UCI Machine Learning Repository: UCI offers various datasets that can be relevant for feature extraction, model training, and validation in the context of breast cancer detection.

4. Clinical Experts and Radiologists:

- Collaborating with healthcare professionals, radiologists, and oncologists specializing in breast cancer can provide valuable insights into the current diagnostic practices, challenges, and desired features for the web-based app.

- Conducting interviews or surveys with experts in the field can help gather domain-specific knowledge, identify key parameters, and validate the relevance of the proposed approach.

5. Online Communities and Forums:

- Engaging with online communities and forums dedicated to breast cancer, machine learning, and medical imaging can provide access to real-world experiences, discussions, and insights from patients, caregivers, researchers, and healthcare professionals.

6. Collaboration with Research Institutions and Medical Centers:

- Collaborating with research institutions, medical centers, or hospitals working on breast cancer detection can provide access to proprietary datasets, expert knowledge, and potential partnerships for validating and refining the web-based app.

7. Regulatory and Ethical Guidelines:

- Familiarizing oneself with regulatory and ethical guidelines, such as HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation), ensures compliance with patient privacy and data protection regulations.

## 1.7 Delimitations and Limitations of the research

1. Availability and Quality of Data:

- The research may be limited by the availability and quality of publicly accessible datasets for training and validation. Limited access to large and diverse datasets may affect the generalizability and performance of the developed models.

2. External Validity:

- The findings and performance of the web-based app may be specific to the dataset(s) used during the research and may not fully represent the variability and complexity of real-world clinical scenarios and patient populations.

3. User Input Variability:

- The accuracy and reliability of the web-based app heavily rely on the quality and consistency of user-provided input parameters. Variability and errors in user input may impact the performance and predictions of the app.

4. Interpretability of Machine Learning Models:

- Some machine learning algorithms, such as deep learning models, are known for their black-box nature, making it challenging to interpret and explain the reasoning behind their predictions. Lack of interpretability may limit the trust and acceptance of the app by healthcare professionals.

5. Inherent Biases in Data:

- The presence of inherent biases in the training data, such as demographic or institutional biases, may influence the performance and fairness of the developed models, leading to potential disparities in breast cancer detection across different populations.

6. Medical Expertise and Clinical Decision-making:

- The web-based app should be seen as a support tool rather than a substitute for medical expertise. The limitations of the app include its inability to replace the clinical judgment and decision-making skills of healthcare professionals.

7. Ethical and Legal Considerations:

- The research must adhere to ethical guidelines and regulations regarding patient privacy, data protection, and informed consent. Compliance with regulations, such as HIPAA and GDPR, may impose limitations on data collection, storage, and sharing.

8. Technological Constraints:

- The performance and efficiency of the web-based app may be influenced by technological limitations, such as limited computational resources, network connectivity issues, or the scalability of the system to handle a high volume of user requests.

9. Time Constraints:

- The timeframe of the research may impose limitations on the scope, depth, and refinement of the developed web-based app. More extensive research and longer-term evaluation may be required to fully assess the impact and effectiveness of the app in clinical settings.

10. User Acceptance and Adoption:

- The successful deployment and adoption of the web-based app in clinical practice may depend on factors such as user acceptance, training requirements, integration with existing healthcare systems, and institutional policies.

## 1.8 Feasibility Analysis

1. Technical Feasibility:

- Availability of Required Skills: Assess the availability of skilled developers and data scientists proficient in web development, machine learning, and medical imaging analysis.

- Technology Stack: Evaluate the compatibility and feasibility of the chosen programming languages, frameworks, and tools for implementing the web-based app.

- Infrastructure and Resources: Ensure access to adequate computational resources, storage capacity, and network infrastructure to support the development, deployment, and maintenance of the app.

2. Data Feasibility:

- Data Availability: Determine the availability of relevant and diverse datasets for training machine learning models and validating the app's performance.

- Data Quality and Preprocessing: Assess the quality of available data and evaluate the feasibility of preprocessing steps required for data cleaning, normalization, and feature extraction.

- Privacy and Security Compliance: Ensure compliance with privacy regulations and ethical considerations when handling sensitive patient data.

3. Financial Feasibility:

- Cost Analysis: Conduct a cost analysis to estimate the financial resources required for development, hosting, maintenance, and potential scalability of the web-based app.

- Funding Opportunities: Explore potential funding sources, grants, or collaborations to support the development and sustainability of the app.

4. Legal and Ethical Feasibility:

- Regulatory Compliance: Evaluate the legal and ethical implications of handling patient data, ensuring compliance with regulations such as HIPAA and GDPR.

- Informed Consent: Establish procedures to obtain informed consent from users for data collection, usage, and sharing, while adhering to ethical guidelines.

5. User Acceptance and Adoption Feasibility:

- User Needs and Requirements: Conduct user surveys, interviews, or focus groups to understand the needs and expectations of healthcare professionals and end-users for the web-based app.

- User Interface and Experience: Ensure the app's user interface is intuitive, user-friendly, and accessible to both technical and non-technical users.

- Training and Support: Assess the feasibility of providing training and support to healthcare professionals for effectively utilizing the app in their clinical practice.

6. Scalability and Performance Feasibility:

- Scalability: Evaluate the ability of the web-based app to handle a growing number of users, increased data volume, and concurrent requests.

- Performance Optimization: Identify potential performance bottlenecks and develop strategies to optimize the app's speed, responsiveness, and computational efficiency.

7. Stakeholder Engagement:

- Collaboration and Partnerships: Identify potential collaborators, research institutions, or healthcare providers for knowledge sharing, validation studies, and real-world implementation of the app.

- Stakeholder Support: Assess the level of support and engagement from key stakeholders, such as healthcare professionals, radiologists, patients, and regulatory bodies, to ensure the feasibility and acceptance of the app.

## 1.9 Justification and rationale

1. Increasing Breast Cancer Incidence:

- Breast cancer is one of the most common types of cancer worldwide, with a significant impact on public health. The development of a web-based app for breast cancer detection is justified by the need to improve early detection rates and enhance patient outcomes.

2. Potential to Improve Diagnostic Accuracy:

- Machine learning algorithms have shown promise in improving the accuracy of breast cancer detection and reducing false negatives and false positives. The web-based app can leverage these algorithms to provide more accurate and reliable predictions, aiding in early diagnosis and timely interventions.

3. Accessibility and Convenience:

- A web-based app offers the advantage of accessibility and convenience, allowing users to access the breast cancer detection tool from various devices and locations. This accessibility can help reach a wider user base, including patients, healthcare professionals, and researchers.

4. Complementary Tool for Healthcare Professionals:

- The web-based app can serve as a complementary tool for healthcare professionals, providing them with additional support and insights during the diagnostic process. It can assist in decision-making, offer second opinions, and enhance the overall efficiency of the diagnostic workflow.

5. Potential for Timely Interventions:

- Early detection of breast cancer is crucial for successful treatment outcomes. By providing a reliable and efficient detection tool, the web-based app can aid in identifying potential cases at an early stage, enabling prompt interventions and improving survival rates.

6. Reducing Healthcare Costs:

- Timely detection and intervention can potentially reduce the overall healthcare costs associated with breast cancer. By aiding in early detection, the web-based app can contribute to cost savings by reducing the need for extensive treatment, hospitalization, and advanced interventions.

7. Research and Data Collection:

- The web-based app can facilitate data collection and contribute to research efforts in breast cancer detection. With appropriate consent and privacy measures, the app's usage can generate valuable anonymized data that can be used for research, improving algorithms, and advancing knowledge in the field.

8. Empowering Patients:

- The web-based app can empower patients by providing them with access to a user-friendly tool for self-assessment and risk evaluation. It can increase patient awareness, encourage proactive engagement in their healthcare, and promote informed discussions with healthcare professionals.

9. Potential for Continuous Improvement:

- The web-based app can be designed to collect user feedback, performance metrics, and outcomes data, allowing for continuous improvement and refinement of the algorithms and features. This iterative approach can enhance the app's accuracy, usability, and effectiveness over time.

## 1.10 Conclusion

The development of a web-based app for breast cancer detection is justified by the increasing incidence of breast cancer, the potential to improve diagnostic accuracy, and the need for accessible and convenient tools in healthcare. By leveraging machine learning algorithms, the app has the potential to enhance early detection rates, aid in timely interventions, and improve patient outcomes.

The web-based app serves as a complementary tool for healthcare professionals, providing them with additional support and insights during the diagnostic process. It can assist in decision-making, offer second opinions, and improve the efficiency of the diagnostic workflow. Additionally, the app empowers patients by providing them with a user-friendly tool for self-assessment and risk evaluation, enhancing awareness and encouraging proactive engagement in their healthcare.

The app's accessibility and convenience, being accessible from various devices and locations, make it a valuable resource for a wider user base, including patients, healthcare professionals, and researchers. Furthermore, the app's potential to reduce healthcare costs through early detection and interventions can have a significant impact on the healthcare system.

The web-based app also contributes to research efforts by facilitating data collection and generating valuable anonymized data for further analysis and algorithm improvement. Continuous feedback and data collection allow for iterative improvements to enhance the app's accuracy, usability, and effectiveness over time.

In conclusion, the breast cancer detection web-based app addresses a critical need in healthcare by leveraging machine learning algorithms, enhancing early detection rates, empowering patients, and supporting healthcare professionals. By embracing technology and accessibility, the app has the potential to contribute to improved patient outcomes, reduced healthcare costs, and advancements in breast cancer research.

# 

# Chapter 2: Analysis Phase

## 2.1 Introduction

The analysis phase lays the foundation for the development of the breast cancer detection web-based app by understanding stakeholder requirements, assessing available data, selecting appropriate algorithms, designing the user interface, and addressing privacy and security considerations. Thorough analysis ensures the development process aligns with the project goals, user expectations, and regulatory requirements, setting the stage for successful implementation in the subsequent phases.

1. Identify Stakeholders:

- Identify key stakeholders involved in the development and use of the web-based app, such as healthcare professionals, patients, researchers, regulatory bodies, and IT personnel.

2. Gather Requirements:

- Conduct interviews, surveys, and workshops with stakeholders to understand their needs, expectations, and desired functionalities of the web-based app.

- Define the specific goals and objectives of the app, including the accuracy threshold, usability requirements, and integration with existing healthcare systems.

3. Define Scope:

- Determine the scope of the web-based app, considering factors such as target user group, geographical coverage, supported diagnostic modalities (e.g., mammograms, MRI, ultrasound), and specific features to be included.

4. Data Collection and Analysis:

- Identify and evaluate available datasets for training and validation purposes. Assess the quality, diversity, and representativeness of the data to ensure the reliability and generalizability of the developed models.

- Analyze the data to understand patterns, trends, and potential biases that may impact the performance and fairness of the app.

5. Model Selection and Evaluation:

- Research and evaluate different machine learning algorithms suitable for breast cancer detection, considering factors such as accuracy, interpretability, computational efficiency, and scalability.

- Implement and train multiple models using the selected algorithms, and evaluate their performance using appropriate evaluation metrics, cross-validation techniques, and statistical analysis.

6. User Interface Design:

- Collaborate with UX/UI designers to create an intuitive and user-friendly interface for the web-based app. Consider the needs of different user groups, accessibility requirements, and best practices for visualizing and presenting diagnostic information.

7. System Architecture Design:

- Define the overall system architecture, including the backend infrastructure, database management, and integration with external systems or APIs for data retrieval and storage.

- Determine the technology stack, frameworks, and tools required for developing and deploying the web-based app.

8. Privacy and Security Considerations:

- Address privacy and security concerns by implementing appropriate measures to ensure the protection of patient data, adherence to regulatory guidelines (e.g., HIPAA, GDPR), and secure data transmission and storage.

9. Performance and Scalability Analysis:

- Conduct performance testing to assess the app's response time, scalability, and resource requirements. Optimize the app's performance to ensure smooth operation and handle a growing user base and data volume.

10. Risk Assessment:

- Identify potential risks and challenges associated with the development and deployment of the app, such as data privacy breaches, algorithmic biases, regulatory compliance issues, and technical constraints. Develop mitigation strategies and contingency plans.

## 2.2. Information gathering

By leveraging these information gathering strategies, you can acquire the necessary knowledge, datasets, expert insights, and guidelines to inform the design, development, and validation of the breast cancer detection web-based app. It ensures that the app aligns with best practices, industry standards, and the needs of healthcare professionals and patients.

1. Identify Information Needs:

- Determine the specific information required for the development of the web-based app, such as clinical guidelines, research papers, datasets, industry best practices, and regulatory guidelines.

2. Medical Research Papers and Journals:

- Conduct a literature review to gather relevant research papers and journal articles related to breast cancer detection, machine learning algorithms, medical imaging, and web-based healthcare applications.

- Explore databases like PubMed, IEEE Xplore, and ACM Digital Library to access scholarly publications in the field.

3. Clinical Guidelines and Protocols:

- Refer to clinical guidelines and protocols provided by reputable organizations such as the National Comprehensive Cancer Network (NCCN) and the American Cancer Society (ACS). These guidelines inform the development of the web-based app by outlining best practices for breast cancer detection, diagnosis, and treatment.

4. Datasets and Repositories:

- Identify publicly available datasets related to breast cancer detection, such as mammograms, MRI scans, and ultrasound images. The Cancer Imaging Archive (TCIA) and UCI Machine Learning Repository are valuable resources for accessing relevant datasets for model training and validation.

5. Clinical Experts and Radiologists:

- Collaborate with healthcare professionals, radiologists, and experts specializing in breast cancer detection. Conduct interviews, surveys, or focus groups to gather insights into current diagnostic practices, challenges, and desired features for the web-based app.

6. Online Communities and Forums:

- Engage with online communities, forums, and social media platforms dedicated to breast cancer, machine learning, and medical imaging. Participate in discussions, ask questions, and gather real-world experiences and insights from patients, caregivers, researchers, and healthcare professionals.

7. Industry Conferences and Workshops:

- Attend conferences, workshops, and seminars focused on breast cancer, machine learning, and medical imaging. These events provide opportunities to learn about the latest advancements, research findings, and industry best practices.

8. Collaboration with Research Institutions and Medical Centers:

- Collaborate with research institutions, medical centers, or hospitals actively involved in breast cancer research and detection. Establish partnerships to access proprietary datasets, gain expert knowledge, and validate the web-based app in clinical settings.

9. Regulatory and Ethical Guidelines:

- Familiarize yourself with regulatory guidelines, such as HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation), to ensure compliance with patient privacy and data protection regulations.

## 2.3 Weaknesses of the existing systems

Weaknesses of Existing Systems for Breast Cancer Detection:

1. Limited Accessibility:

- Traditional breast cancer detection methods, such as mammography or physical examinations, may not be easily accessible to all individuals, especially those in remote or underserved areas. Limited access to screening facilities can result in delayed or missed diagnoses.

2. Subjectivity and Variability:

- Interpretation of mammograms and other imaging tests for breast cancer detection relies on the expertise and subjective judgment of radiologists. Variability in interpretation can lead to inconsistencies and potential misdiagnosis, impacting patient outcomes.

3. False Positives and False Negatives:

- Existing systems may exhibit high rates of false positives and false negatives. False positives can lead to unnecessary invasive procedures and increased patient anxiety, while false negatives can result in missed diagnoses and delayed treatment.

4. Radiation Exposure:

- Mammography, the most common breast cancer screening tool, involves the use of ionizing radiation. Prolonged and repeated exposure to radiation during screening can increase the risk of radiation-related health issues, such as radiation-induced cancer.

5. Discomfort and Pain:

- Some screening methods, such as mammography and breast biopsies, can cause discomfort or pain for patients, leading to reduced compliance with regular screenings and reluctance to undergo necessary diagnostic procedures.

6. Lack of Personalized Risk Assessment:

- Existing systems may not provide comprehensive personalized risk assessment for individuals. Factors such as family history, genetic predisposition, and lifestyle choices are not always taken into account, potentially impacting the accuracy of risk estimation.

7. Time and Resource Constraints:

- High demand for mammography and limited availability of screening facilities can result in long waiting times for appointments and delays in obtaining results. This can hinder timely detection and treatment initiation.

8. Cost and Affordability:

- Traditional breast cancer detection methods, such as mammography and MRI scans, can be expensive, making them less accessible to individuals with limited financial resources. The cost of screening and subsequent diagnostic procedures can pose a barrier to early detection and treatment.

9. Lack of Integration and Data Sharing:

- Existing systems may lack seamless integration and interoperability, hindering the sharing of patient data and collaboration among healthcare professionals. This can impede the comprehensive assessment and continuity of care for individuals undergoing breast cancer detection.

10. Limited Support for Decision-making:

- Current systems may not provide sufficient decision support tools for healthcare professionals, such as risk prediction models or clinical decision support systems. This can result in subjective decision-making and potential variations in treatment approaches.

## 2.4 Process analysis

Process analysis ensures a systematic and structured approach to the development of the breast cancer detection web-based app. It involves gathering requirements, collecting and preprocessing data, developing and evaluating machine learning algorithms, designing the user interface, implementing the system, testing, deployment, and providing ongoing support. By following this process, the app can be developed efficiently, ensuring quality, accuracy, and usability while addressing the specific needs of stakeholders and end-users.

1. Requirement Gathering:

- Identify and document the specific requirements of the web-based app through stakeholder interviews, surveys, and workshops. Define the functional and non-functional requirements, including desired features, usability criteria, and integration with existing systems.

2. Data Collection and Preprocessing:

- Gather relevant datasets for breast cancer detection, including mammograms, MRI scans, and ultrasound images. Preprocess the data by cleaning, normalizing, and augmenting it as necessary to ensure data quality and consistency.

3. Algorithm Selection and Development:

- Research and evaluate various machine learning algorithms suitable for breast cancer detection, support vector machines (SVMs). Develop and train the selected algorithms using the preprocessed data.

4. Model Evaluation and Validation:

- Evaluate the performance of the developed models using appropriate evaluation metrics, cross-validation techniques, and statistical analysis. Validate the models using separate datasets or through collaboration with healthcare professionals and radiologists.

5. User Interface Design:

- Design visually appealing screens, interactive features, and user-friendly workflows to ensure a positive user experience.

6. System Architecture Design:

- Design the overall system architecture, including the selecting suitable technologies, frameworks, and tools for developing and deploying the web-based app.

7. App Development:

- Implement the web-based app according to the defined requirements and design specifications. Develop the frontend components, backend functionalities, and necessary integration points. Implement security measures and data privacy protocols.

8. Testing and Quality Assurance:

- Conduct comprehensive testing to identify and fix any bugs, errors, or usability issues. Perform unit testing, integration testing, and system testing to ensure the app functions as intended, meets the requirements, and delivers accurate results.

9. Deployment and Integration:

- Deploy the web-based app on appropriate servers or cloud platforms.

10. User Training and Support:

- Provide training materials and resources to educate healthcare professionals and end-users on how to effectively use the web-based app. Offer technical support and assistance to address any user queries or issues that may arise during app usage.

11. Continuous Improvement and Updates:

- Monitor the app's performance, user feedback, and emerging research findings to identify areas for improvement. Incorporate updates, bug fixes, and algorithm refinements based on user needs, changing requirements, and advancements in the field of breast cancer detection.

## 2.5 Data Analysis

1. Data Collection:

- Identify and gather relevant datasets for breast cancer detection

2. Data Preprocessing:

- Clean the collected data by removing any duplicate or erroneous entries. Handle missing data by imputation techniques or excluding records with significant missing values. Normalize the data to ensure consistency and comparability.

3. Exploratory Data Analysis (EDA):

- Perform exploratory data analysis to gain insights into the data. Visualize the data distribution, identify outliers, and investigate correlations between variables. EDA helps understand the characteristics of the data and potential patterns related to breast cancer.

4. Feature Selection and Engineering:

- Select relevant features from the dataset that contribute to breast cancer detection. Use domain knowledge, statistical techniques, and feature importance analysis to identify informative features. Additionally, engineer new features that may enhance the predictive power of the models.

5. Splitting the Dataset:

- Split the dataset into training, validation, and testing sets. The training set is used to train the models, the validation set helps tune hyperparameters and assess model performance, while the testing set is used to evaluate the final model's performance.

6. Model Training and Evaluation:

- Select appropriate machine learning algorithms, such as support vector machines (SVMs), for breast cancer detection. Train the models using the training dataset and evaluate their performance using appropriate evaluation metrics, such as accuracy, precision, recall, and F1-score.

7. Cross-Validation:

- Perform cross-validation techniques, such as k-fold cross-validation, to assess the models' generalization performance. This technique helps mitigate overfitting and provides a more robust estimate of the models' performance on unseen data.

8. Model Optimization:

- Fine-tune the hyperparameters of the models using techniques like grid search or random search. Optimize the models to achieve the best possible performance while avoiding overfitting or underfitting. Regularization techniques, such as L1 or L2 regularization, can help improve the models' generalization ability.

9. Model Comparison and Selection:

- Compare the performance of different models and select the one that demonstrates the highest accuracy, sensitivity, specificity, and overall performance on the validation and testing datasets. Consider factors such as computational efficiency, interpretability, and scalability.

10. Ethical Considerations:

- Evaluate the models for potential biases and fairness issues. Investigate and address any biases that may arise due to imbalanced datasets or algorithmic limitations. Ensure the models do not disproportionately impact any specific demographic or population group.

11. Validation with Independent Datasets:

- Validate the selected model with independent datasets or collaborate with healthcare professionals and radiologists to assess its performance in real-world clinical settings. Validate the accuracy, reliability, and generalizability of the model's predictions and ensure it aligns with clinical expertise.

**2.7 Evaluate Alternatives**

Other Machine Learning Algorithms:

Consider alternative machine learning algorithms, such as random forests, gradient boosting, or deep learning architectures, for breast cancer detection. Evaluate their performance, computational efficiency, interpretability, and scalability compared to the selected algorithms.

Hybrid Approaches:

Explore hybrid approaches that combine multiple screening methods or algorithms to enhance the accuracy and reliability of breast cancer detection. Evaluate the potential synergy and performance improvements achieved through the integration of different techniques.

**2.8 Requirements Analysis**

To address the weaknesses identified in section 3.6 and ensure the effectiveness of the new web-based student grievance management and resolution system, it's essential to conduct a thorough requirements analysis. This analysis will encompass both functional and non-functional requirements.

**2.8.1 Functional Requirements**

1. Data Input and Management:

- Enable users to input and manage patient data, including demographic information, medical history, and relevant risk factors associated with breast cancer. Ensure secure storage and retrieval of data.

2. Automated Classification:

- Utilize machine learning algorithms to automatically classify breast cancer cases based on input parameters. Provide accurate predictions for the presence or absence of breast cancer using the trained model.

3. Interactive User Interface:

- Design a user-friendly and visually appealing interface using Streamlit and Python that allows users to interact with the app seamlessly. Ensure a responsive and intuitive user experience.

4. Error Handling and Validation:

- Implement error handling mechanisms and input validation to ensure data integrity and prevent erroneous inputs. Display clear and informative error messages to guide users in correcting any issues.

**2.8.2 Non-functional Requirements**

Non-functional requirements specify constraints and quality attributes of the system that are not directly related to its functionalities. Here are some non-functional requirements:

1. Security:

- Ensure the app follows best practices for data security, including encryption of sensitive data, secure storage and transmission, and protection against unauthorized access. Implement user authentication and access control mechanisms to safeguard patient information.

2. Performance:

- Design the app to handle a large number of concurrent users and process data efficiently. Optimize the app's performance to provide fast response times, minimize loading delays, and handle peak usage periods without performance degradation.

3. Scalability:

- Build the app to be scalable, allowing it to handle an increasing number of users, data inputs, and computation requirements. Ensure the app's architecture and infrastructure can accommodate future growth and demand without compromising performance.

4. User Experience (UX):

- Focus on creating an intuitive and user-friendly interface that is easy to navigate and understand. Ensure consistency in design elements, use appropriate visual cues, and employ responsive design principles to provide a seamless and pleasant user experience across different devices and screen sizes.

5. Reliability:

- Ensure the app is reliable and available for users at all times. Implement mechanisms to handle errors, crashes, or server failures gracefully, and provide appropriate error messages to guide users. Implement regular backups and data recovery procedures to minimize data loss and ensure system reliability.

6. Maintainability:

- Design the app with modularity and maintainability in mind, allowing for easy updates, bug fixes, and future enhancements. Follow coding best practices, use proper documentation, and establish version control processes to facilitate efficient maintenance and support.

7. Compatibility:

- Ensure the app is compatible with a wide range of browsers, operating systems, and devices. Test and validate the app's functionality across different platforms to ensure consistent performance and user experience.

## 2.9 Conclusion

In conclusion, the development of a breast cancer detection web-based app presents an opportunity to enhance the accuracy, efficiency, and accessibility of breast cancer screening and diagnosis. By evaluating alternatives and defining functional and non-functional requirements, we can ensure the app's effectiveness and usability.

The functional requirements encompass crucial aspects such as user registration and authentication, data input and management, automated classification, interactive user interface, and error handling. These requirements enable healthcare professionals and users to input and manage patient data, leverage machine learning algorithms for automated classification, interact with the app seamlessly through an intuitive user interface, and receive accurate predictions for breast cancer presence or absence.

Additionally, non-functional requirements address critical aspects such as security, performance, scalability, user experience, reliability, maintainability, accessibility. These requirements ensure the app's security and protection of sensitive patient information, its ability to handle increasing user demands and data inputs, responsiveness and usability across different devices, reliability in terms of availability and error handling, ease of maintenance and future enhancements, adherence to accessibility and regulatory standards, compatibility with various platforms, and provision of comprehensive documentation.

By meeting these requirements, the breast cancer detection web-based app can provide healthcare professionals with valuable decision support, aid in risk assessment and stratification, facilitate collaboration and communication, generate comprehensive reports, and improve patient outcomes. The app's accurate classification capabilities, user-friendly interface, and adherence to security and privacy standards contribute to its effectiveness and reliability.

# Chapter 3: Design Phase

## 3.1 Introduction

During the design phase of the software development lifecycle, the focus is on transforming the requirements into a well-structured and detailed design that will guide the implementation process.

**3.2 System Design**

The System Design phase represents a crucial stage in the development lifecycle of an SGRS, where conceptual ideas are transformed into concrete structures and functionalities. It involves the systematic analysis, specification, and visualization of the system's architecture, components, workflows, and interfaces. Additionally, this phase necessitates careful consideration of various factors, including stakeholder requirements, technological capabilities, legal frameworks, and best practices in grievance resolution.

### 3.2.1 Context Diagram

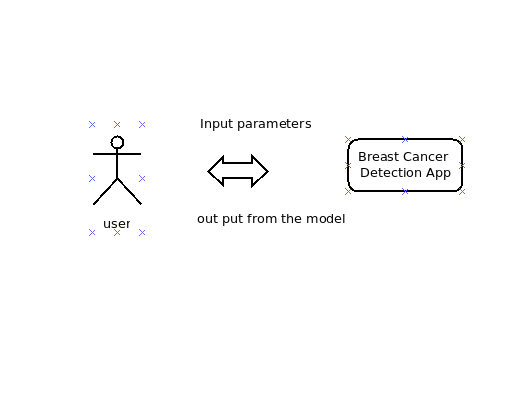


Fig 1.0

Above is the Context Diagram for a Breast Cancer Dectection app that provides an overview of the system's interactions with external entities.

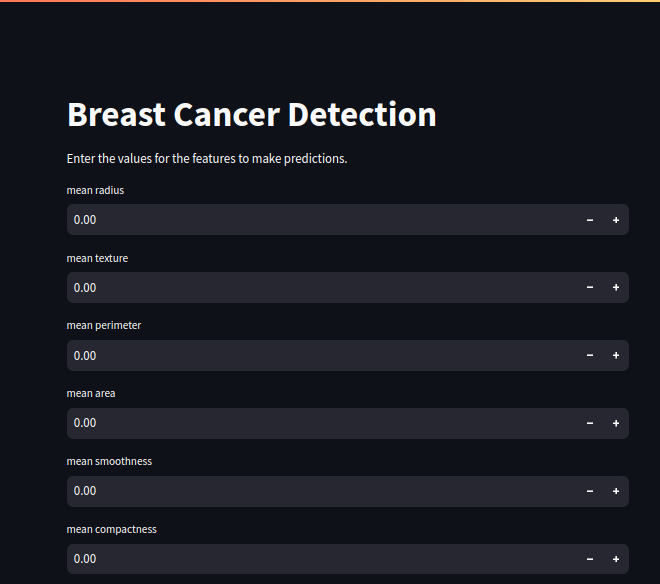
## 3.2 Architectural design

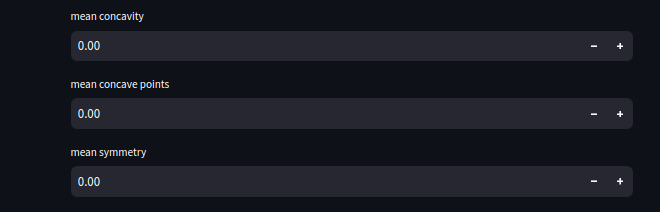
Single Page Architecture: Since it's a one-page Streamlit app, the architecture will typically follow a single page architecture pattern. The app will consist of a single web page that contains all the necessary components and functionality.

## 3.3 Interface design

**3.3.1 Input design**

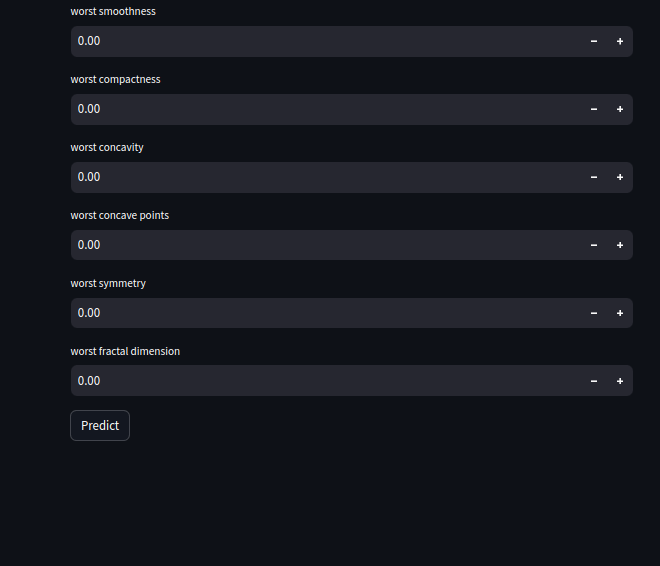
**Input Parameter Pages**

****

****

****

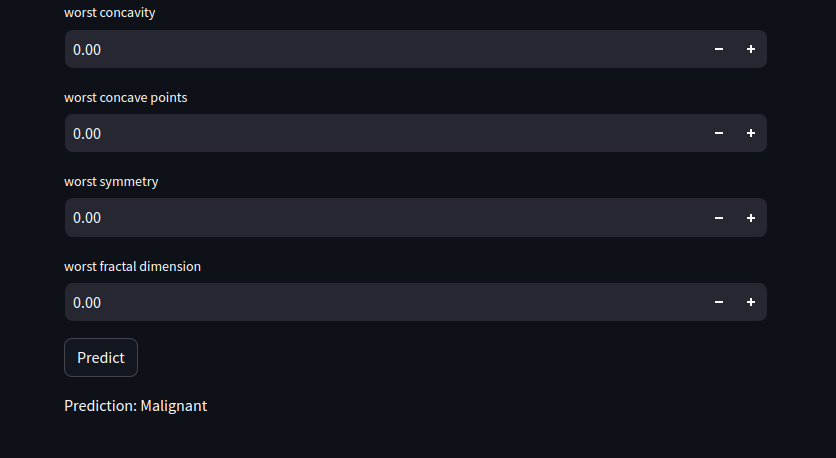
****

****

**Fig 1.2**

The above images are of a form taking input from user then feed it into the model for prediction

**3.2 Output design**

****

The above image is an output display if user tries to predict without inputing values

## 3.3 Unified Modeling Language

**System Sequence Diagram**

The Unified Modeling Language (UML) is a standardized way to visually

represent a software system's design. It's like a blueprint for developers, using a

collection of diagrams to show different aspects of the system :

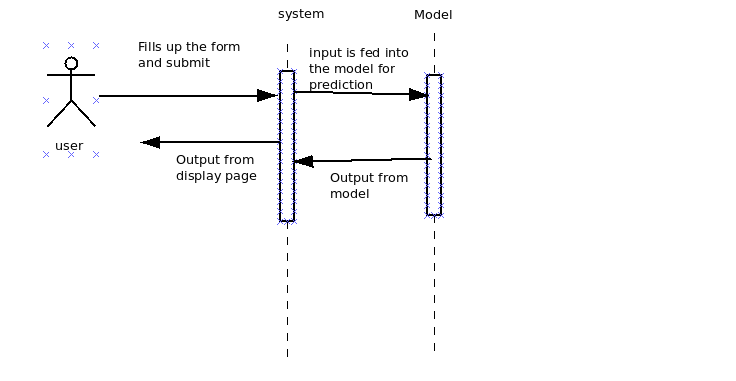


Fig 1.3

The above **Fig 1.3** is a sequence diagram showing the sequence of user input , prediction and the displayed output

## 3.4 Security design

1. Secure Communication:

- Implement secure communication protocols (such as HTTPS) to encrypt data transmitted between the app and clients, ensuring confidentiality and integrity of data during transmission.

2. Input Validation:

- Apply robust input validation techniques to prevent common security vulnerabilities, such as cross-site scripting (XSS) . Validate and sanitize user inputs to mitigate the risk of malicious input exploitation.

3. Data Protection:

- If the app stores or processes sensitive data, employ appropriate encryption methods to protect the data at rest. This can include encrypting data in databases or using encrypted file storage.

4. Error Handling:

- Implement proper error handling mechanisms to avoid exposing sensitive information in error messages. Display generic error messages to users while logging detailed errors for system administrators to investigate.

5. Secure Coding Practices:

- Follow secure coding practices to mitigate potential security risks. This includes avoiding hardcoded sensitive information, utilizing secure libraries and frameworks, and regularly updating dependencies to address known vulnerabilities.

6. Rate Limiting:

- Implement rate limiting mechanisms to prevent abuse or excessive usage of the app's resources. This can help protect against denial-of-service (DoS) attacks or brute-force attacks.

7. Security Testing:

- Conduct regular security testing, such as vulnerability scanning or penetration testing, to identify and address any security weaknesses. This can help ensure the app remains resilient against potential threats.

8. Logging and Monitoring:

- Implement logging and monitoring mechanisms to track app behavior and detect any suspicious activities. Regularly review logs to identify potential security incidents or anomalous behavior.

9. Security Updates:

- Stay updated with security patches for the underlying software, frameworks, and libraries used in the app. Regularly apply updates to address known vulnerabilities and protect against emerging threats.

# Chapter 4: Implementation Phase

## 4.1 Introduction

The Implementation Phase marks a crucial stage in the development and deployment of this system. This phase signifies the transition from planning and design to the actual execution of the system . With a comprehensive plan in place, the implementation phase aims to bring the envisioned system to life, ensuring its functionality, accessibility, and effectiveness in addressing student grievances efficiently.

**4.2 Coding**

This phase shifts from design to actual programming and development of the model and the web app . This phase involves writing code to implement the functionalities and features outlined in the design specifications. Below are some code snippets from the code base .

**Data Loading from dataset**

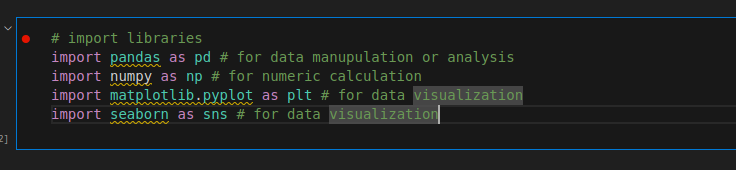


fig 1.4

Fig 1.4 is a screenshot of libraries being imported ,that are going to be used in training the model

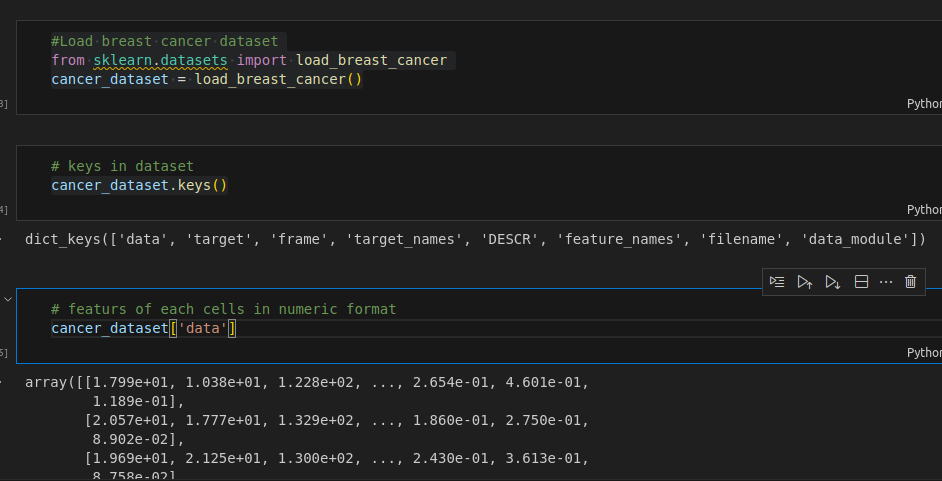


fig 1.5

Fig 1.5 The provided code snippet loads the breast cancer dataset using scikit-learn's load\_breast\_cancer function.The cancer\_dataset['data'] contains the features of each cell in numeric format. The array represents a collection of cells, where each row represents a cell and each column represents a specific feature of that cell.

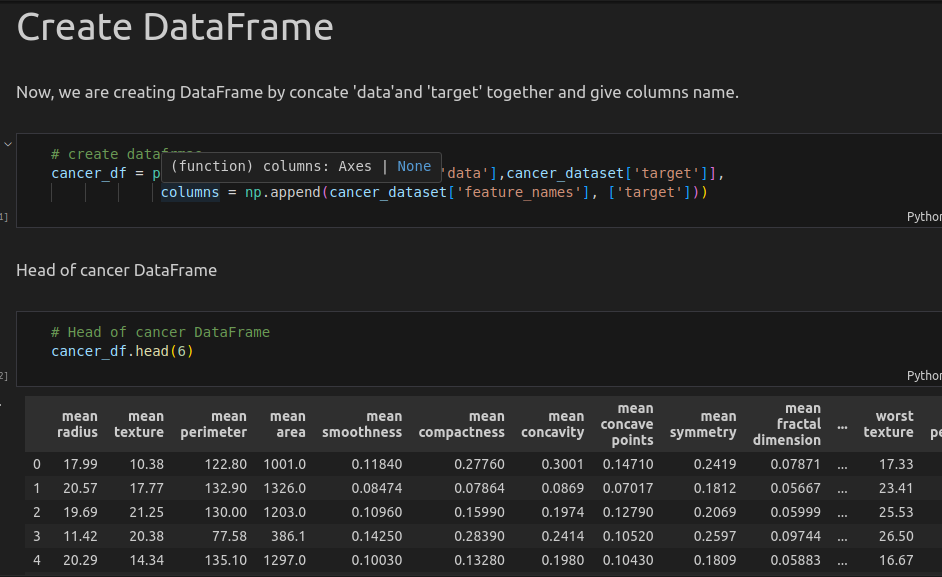


Fig 1.6

In Fig 1.6 The code creates a DataFrame named cancer\_df using the np.c\_ function to concatenate the cancer\_dataset['data'] (features) and cancer\_dataset['target'] (target) arrays. The column names are set as the feature names from cancer\_dataset['feature\_names'], with an additional column named 'target' appended.

The print(cancer\_df.head(6)) statement displays the first six rows of the DataFrame, showing the features and target values for each cell in the breast cancer dataset.

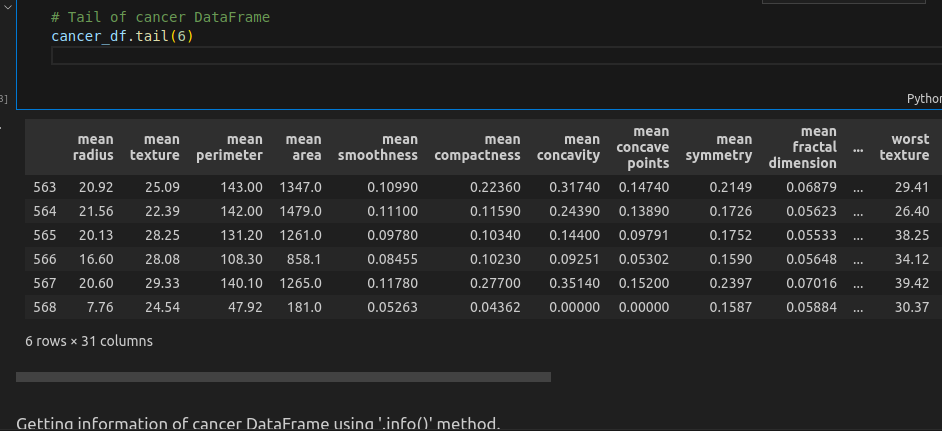


Fig 1.7

In Fig 1.7 The tail(6) function displays the last six rows of the cancer\_df DataFrame, providing a view of the remaining data. Each row represents a cell, and the columns represent the features and target values.

**Data Visualiozation**

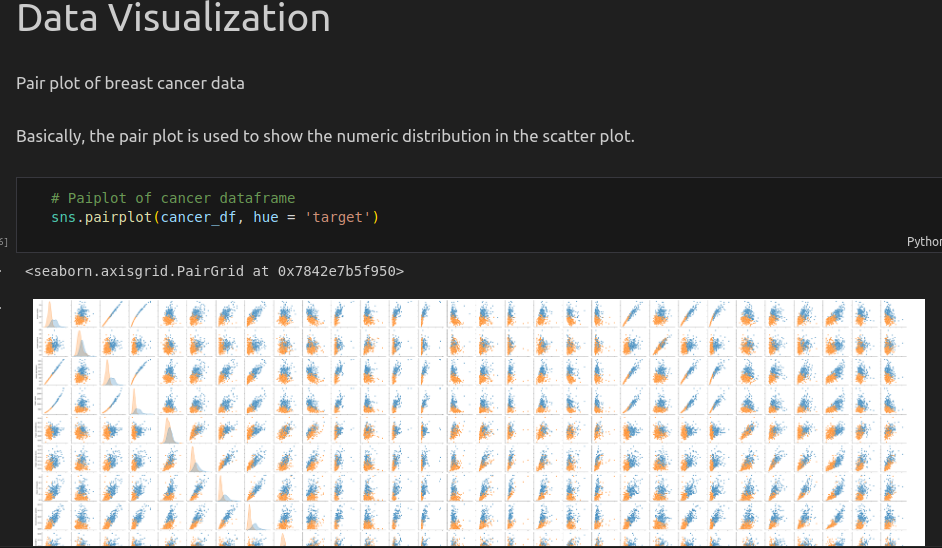


Fig 1.8

fig 1.8 This code will generate a grid of scatter plots, where each scatter plot represents the relationship between two variables (features) from the cancer DataFrame. The hue='target' parameter is used to color the data points based on the target variable, providing visual differentiation between the two classes (0 and 1) in the dataset.

The resulting pairplot will display scatter plots for each combination of features, showing the data distribution and potential patterns or trends. The points will be colored based on the target variable, allowing you to observe any distinguishing characteristics between the two classes.



Fig 1.9

This code will generate a grid of scatter plots, focusing on the specified features: 'mean radius', 'mean texture', 'mean perimeter', 'mean area', and 'mean smoothness'. The hue='target' parameter colors the data points based on the target variable, allowing you to differentiate between the two classes (0 and 1) in the dataset.

The resulting pair plot will display scatter plots for the selected features, showing the pairwise relationships between them. Each scatter plot will have data points colored based on the target variable, providing insights into potential patterns or differences between the two classes for these specific features.

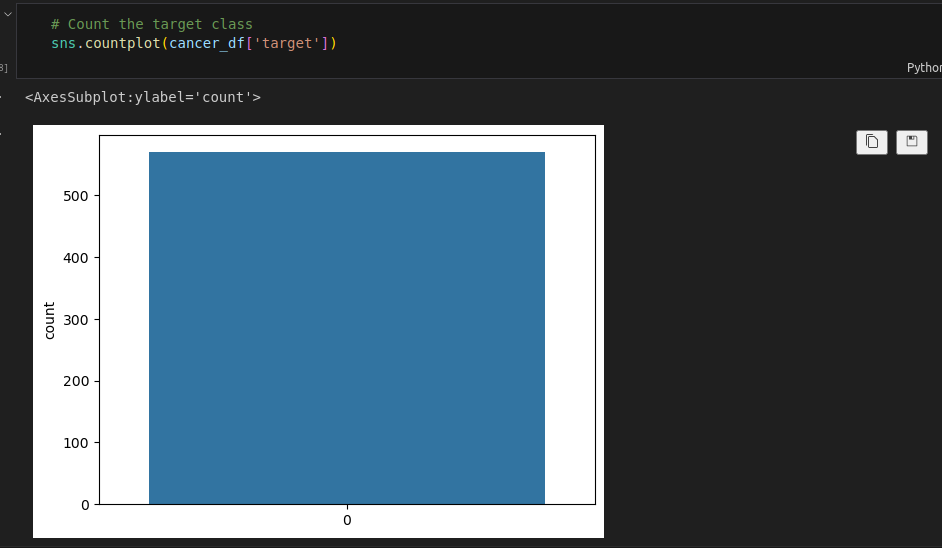


Fig 1.10

This code will generate a bar plot that displays the count of each target class in the 'target' column of the cancer DataFrame.

The resulting count plot will have the x-axis representing the target classes (0 and 1), and the y-axis representing the count of occurrences for each class. This allows you to visually compare the distribution of the target classes and see if there is any class imbalance in the dataset.

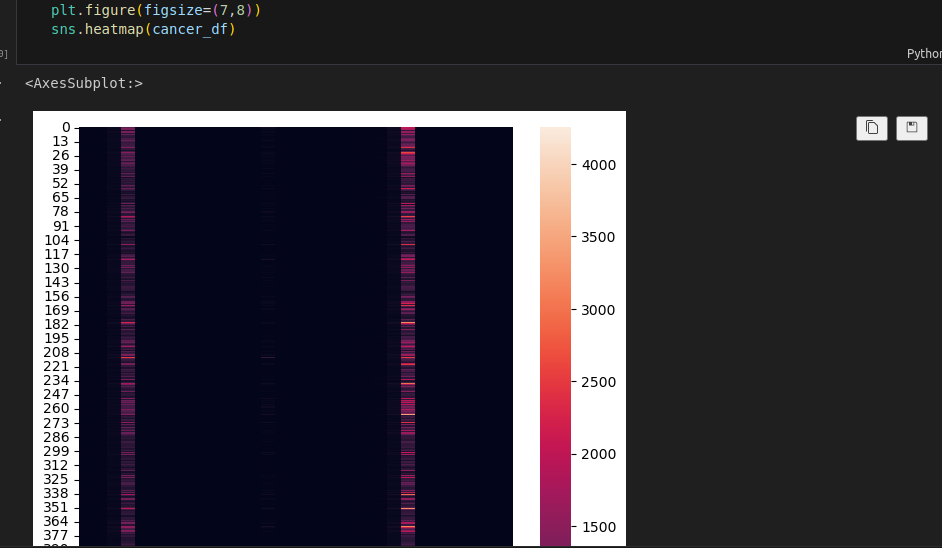


Fig 1.11

In the fig1.11 heatmap we can see the variety of different feature’s value. The value of feature 'mean area' and 'worst area' are greater than other and 'mean perimeter', 'area error', and 'worst perimeter' value slightly less but greater than remaining features.



Fig 1.12

Fig 1.12 finds a correlation between each feature and target we visualize heatmap using the correlation matrix.

In fig 1.12 correlation barplot only feature 'smoothness error' is strongly positively correlated with the target than others. The features 'mean factor dimension', 'texture error', and 'symmetry error' are very less positive correlated and others remaining are strongly negatively correlated.

**Data Preprocessing**

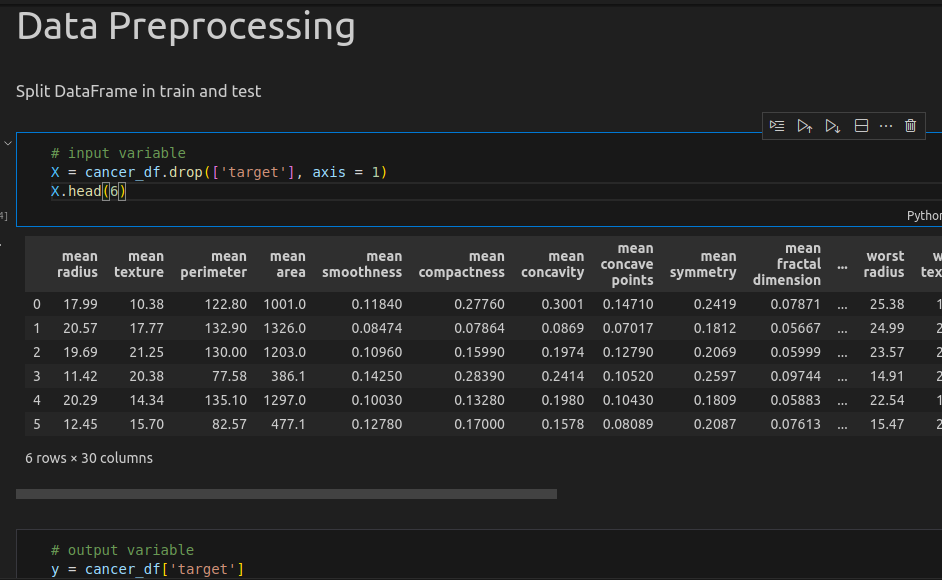
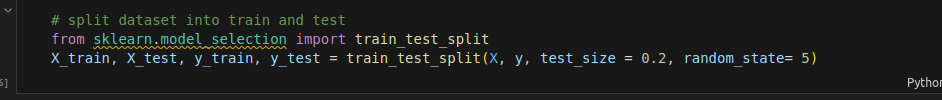
fig 1.13

fig 1.13 The code drops the 'target' column from the cancer\_df DataFrame using the drop() function with axis=1 (indicating column-wise operation). The resulting DataFrame X contains all the remaining columns (features) from the original cancer\_df DataFrame.

The print(X.head(6)) statement displays the first six rows of the X DataFrame, showing the features of each cell in the breast cancer dataset, excluding the 'target' column.

 Fig 1.14

In this code, the train\_test\_split() function is called with the following parameters:

X: The feature dataset (X) that you want to split.

y: The target variable (y) that corresponds to the feature dataset.

test\_size: The proportion of the dataset to include in the testing split. Here, it is set to 0.2, indicating a 20% testing split and an 80% training split.

random\_state: The seed value used by the random number generator for reproducibility. Setting a specific random\_state ensures that the same split is obtained each time the code is run.

After executing the code, the feature dataset and target variable are divided into training and testing sets:

X\_train: The training set of features.

X\_test: The testing set of features.

y\_train: The training set of target values.

y\_test: The testing set of target values.

**Feature Scaling**

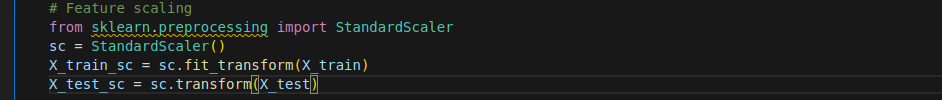


fig 1.15

In this code, the StandardScaler() object sc is created. The fit\_transform() method is then called on X\_train, which computes the mean and standard deviation of each feature in the training set and scales the features accordingly. The resulting scaled features are stored in X\_train\_sc.

Next, the transform() method is called on X\_test to scale the testing set using the mean and standard deviation values computed from the training set. The scaled testing set is stored in X\_test\_sc.

By applying feature scaling, the data is standardized, ensuring that each feature has a mean of 0 and a standard deviation of 1. This process is beneficial for many machine learning algorithms as it can improve their performance and convergence.

The data is described as clean and suitable for building an ML model. Since the output is in categorical format, supervised classification machine learning algorithms are deemed appropriate.

To identify the best ML model, it is necessary to train and test the dataset using multiple algorithms. The paragraph suggests approaching this by importing the required packages to facilitate the implementation of various algorithms and subsequent evaluation.

Some of the mentioned packages include numpy and pandas for data manipulation and analysis, train\_test\_split for dataset splitting, StandardScaler for feature scaling, and several classification algorithms such as LogisticRegression, DecisionTreeClassifier, RandomForestClassifier, SVC, and KNeighborsClassifier. Additionally, the accuracy\_score and classification\_report metrics are mentioned for evaluating the performance of the models.

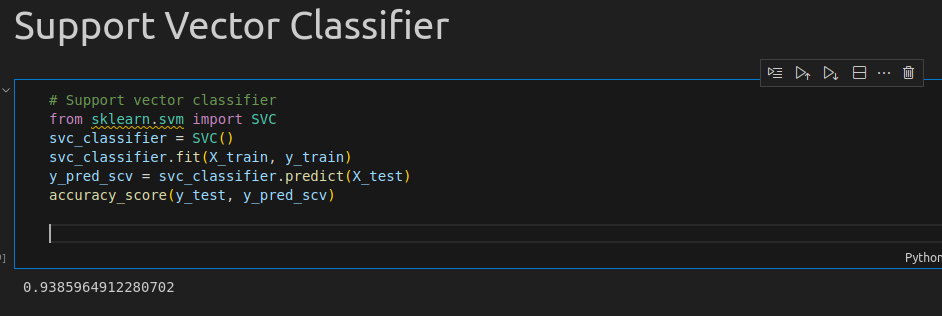
****

Fig 1.16

The accuracy\_score() function from scikit-learn's metrics module is used to calculate the accuracy of the SVC model's predictions. It compares the predicted values y\_pred\_scv with the true values y\_test.

The calculated accuracy score provides an evaluation of the SVC model's performance on the test data.Which is 93,8 % accurate

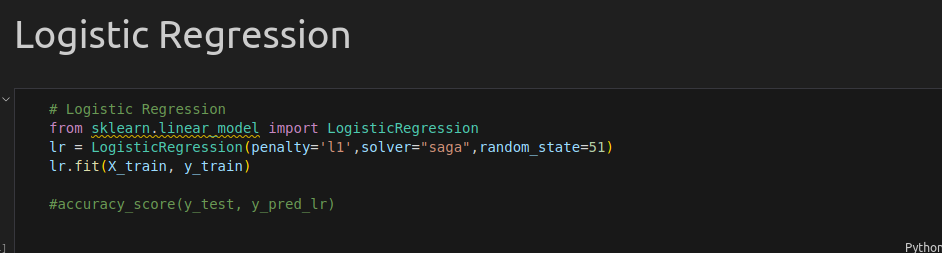
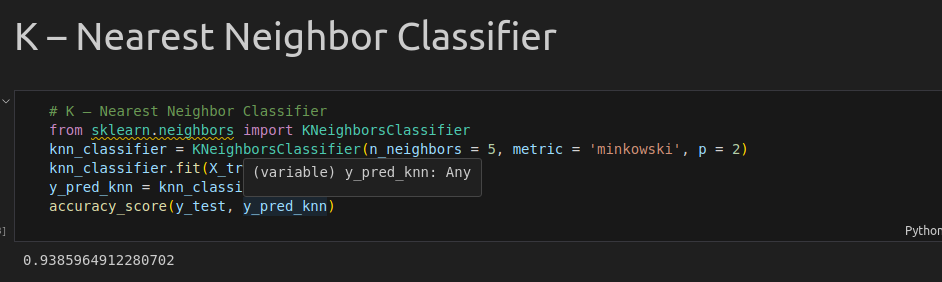


fig 1.17logistic regression

****

**fig 1.18**

The calculated accuracy score provides an evaluation of the K-Nearest Neighbor Classifier model's performance on the test data.Which is 93,8 % accurate

****

**fig 1.19**

Naive Bayes classifier with an accuracy of 94.7



fig 1.20

Decission Tree with an accuracy of 94.7 %

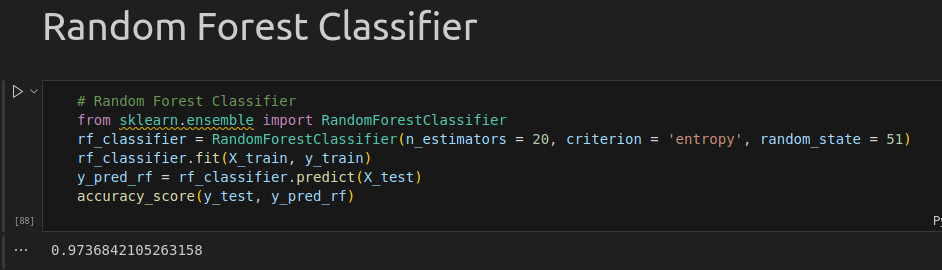
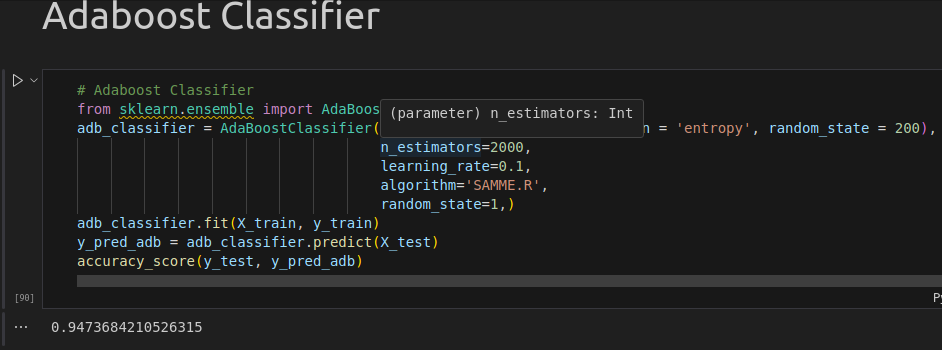


fig 1.21

Random Forest Classifier with an accuracy of 97.4%



**fig 1.22**

Adaboost with an accuracy of 94.7%



**fig 1.23**

XGBoost with an accuracy of 98.2%

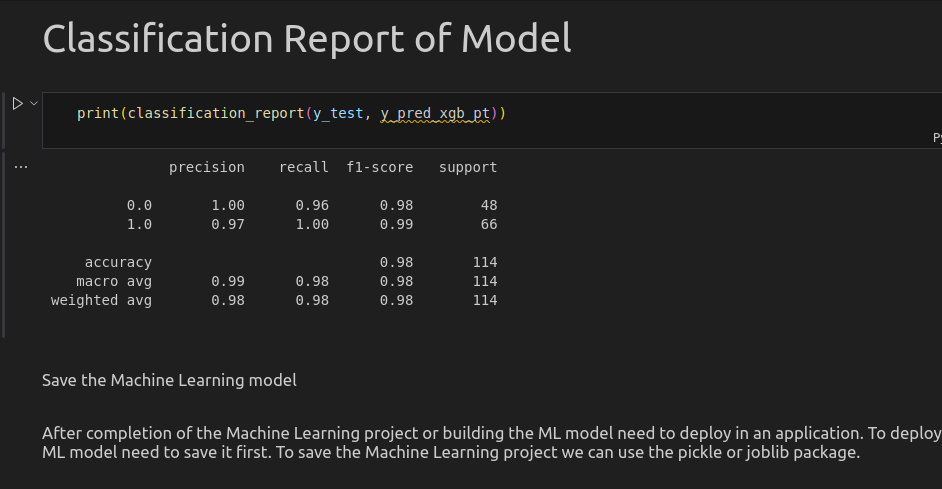


fig 1.24

Once the machine learning project or model is completed, it needs to be deployed in an application for practical use. Before deployment, it is necessary to save the machine learning model. Two commonly used packages for saving machine learning models are mentioned: pickle and joblib.

To save the machine learning project, the pickle or joblib package can be employed. These packages provide functionalities to serialize and save the trained model to a file. This allows the model to be stored and retrieved later for inference or prediction tasks.

Saving the machine learning model is an important step in the deployment process as it ensures that the trained model can be reused without having to retrain it every time the application is run.

**Conclusion**

1. Training multiple algorithms: To enhance accuracy, all supervised classification algorithms were trained. While the statement encourages trying out different algorithms, it specifies that Logistic Regression, Random Forest, and XGBoost classifiers achieved higher accuracy compared to others.

2. Selection of XGBoost: Among the trained algorithms, XGBoost classifier was chosen due to its superior accuracy. XGBoost is known for its effectiveness and is a popular choice in machine learning tasks.(XGBoost Documentation ,2024)

3. Retraining for sustained accuracy: the deployed model is regularly retrained to uphold accuracy. This highlights the commitment to continuously improving and updating the model over time.

4. Impact on breast cancer patients: the hope that the efforts put into developing and refining the machine learning model will have a positive impact on saving the lives of breast cancer patients. It underscores the significance of accurate prediction models in the healthcare domain.

In summary, the findings emphasizes the training and selection of high-performing algorithms, the importance of regular model retraining, and the potential life-saving implications of accurate machine learning models for breast cancer patients.

**4.3 Testing**

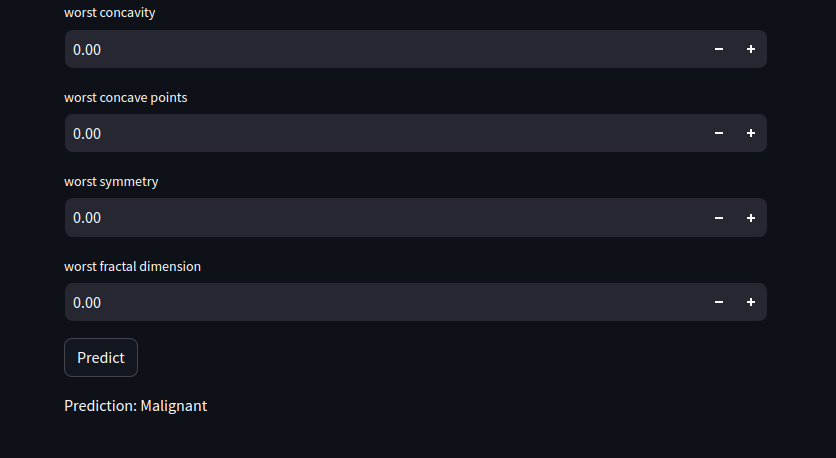
****

fig 1.25

The above image is an output display if user tries to predict without inputing values

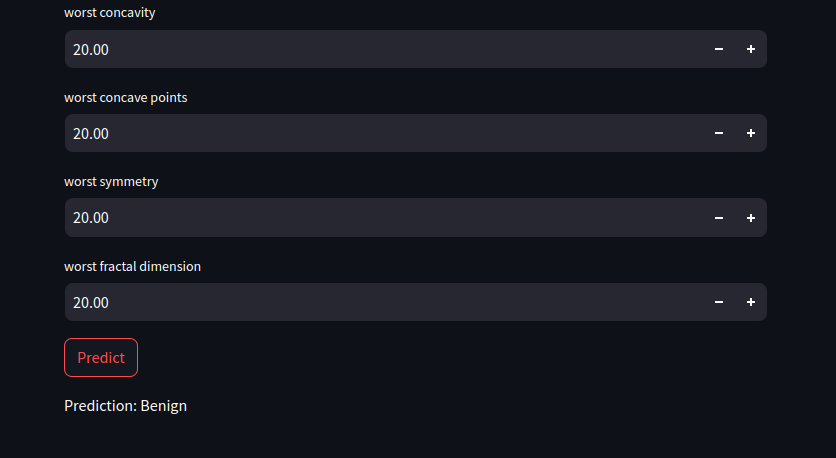


fig 1.26

**Reference**

MACHINE LEARNING TECHNIQUES IN BREAST CANCER DETECTION Zaim Gökbay, İnci 2007

Analysis of Machine Learning Algorithms for Breast Cancer Detection

Aswathy M. A., Jagannath Mohan 2020

Al-Tam, R. M., & Narangale, S. M. (2021). "Breast Cancer Detection and Diagnosis Using Machine Learning." Breast\_Cancer\_Detection\_and\_Diagnosis\_Us.pdf

available at:https://d1wqtxts1xzle7.cloudfront.net/71893326/32-libre.pdf?1634007386=&response-content-disposition=inline%3B+filename%3DBreast\_Cancer\_Detection\_and\_Diagnosis\_Us.pdf&Expires=1716707899&Signature=J3F57CKbxaiFLmHiZUpxuvgAPScSeTa3uVTDM6YOFg3ZxUtuHyxi21OA50-aPIHlVx8EoyXoaDi58G4mpp2Typ6p058kX8ULzVIxiMQZKspR17gvolP9s4F523JRIxn~wPriwZcrZlTbGvHL7~N9zxIsMzPDdNc7M6M8WwLZjLbr~LvCwdNJUOJPHuVQ4XsFsJYlgrapH1FbxlXgjZyPydLnElIeLBBC8TOPAmFkFMeoLeH29xf2FfOuZh5Ao8ixaiUV7g1zMuCY9auLMIrf5vbw4Opgl6QS3lzA31Dt3PDXHX540FOt6VeR-ypE2t0DFdSlrggGpbfYy4IFzXJjOg\_\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

accessed on 26 May 2024

(ACS), A. C. S. (2020). Breast Density and Mammogram

Reports | Dense Breast Tissue. ,available at :

https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/mammograms/breast-density-and-your-mammogram-report.html

accessed on :26 May 2024

XGBoost Documentation ,2024 [online] Available at: <https://xgboost.readthedocs.io/en/stable/> Accessed 26 May 2024.

Adem, K. (2020). Diagnosis of breast cancer with Stacked autoencoder and Subspace kNN. Physica A: Statistical Mechanics and Its Applications, 551, 124591.

Al-Antari, M. A., Han, S.-M., & Kim, T.-S. (2020). Evaluation of deep learning detection and classification towards computer-aided diagnosis of breast lesions in digital X-ray mammograms. Computer Methods and Programs in Biomedicine, 196, 105584.

Al-Tam, R. M. (2015). Diversifying medical imaging of breast lesions (Doctoral dissertation) [University ofAlgarve,sapientia.ualg.pt].

http://hdl.handle.net/10400.1/8120

Aly, G. H., Marey, M., El-Sayed, S. A., & Tolba, M. F. (2021).YOLO Based Breast Masses Detection and Classification in Full-Field Digital Mammograms. Computer Methods and Programs in Biomedicine, 200, 105823.

American Cancer Society. (2020). Breast Cancer Risk Factors and Prevention Methods. Retrieved July 09, 2020, from https://www.cancer.org/cancer/breast-cancer/risk-and-prevention.html

American College of Radiology. (2019). ACR Atlas, BI-RADS. Retrieved July 06, 2020, from <https://www.acr.org/-> /media/ACR/Files/%0ARADS/BI-RADS/Mammography-Reporting.pdf

Arar, Ö. F., & Ayan, K. (2017). A feature dependent Naive Bayes approach and its application to the software defect prediction problem. Applied Soft Computing, 59, 197–209. Araújo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy,

C., Polónia, A., & Campilho, A. (2017).

Classification of breast cancer histology images using convolutional neural networks. PloS One, 12(6), e0177544. Aroef, C., Rivan, Y., & Rustam, Z. (2020). Comparing random forest and support vector machines for breast cancer classification. Telkomnika, 18(2), 815–821. Bagchi, S., Huong, A., & Tay, K. G. (2017). Investigation of different spatial filters performance toward mammogram de-noising. Int. J. Integr. Eng, 9(3), 49–53. Bagchi, S., Tay, K. G., Huong, A., & Debnath, S. K. (2020).

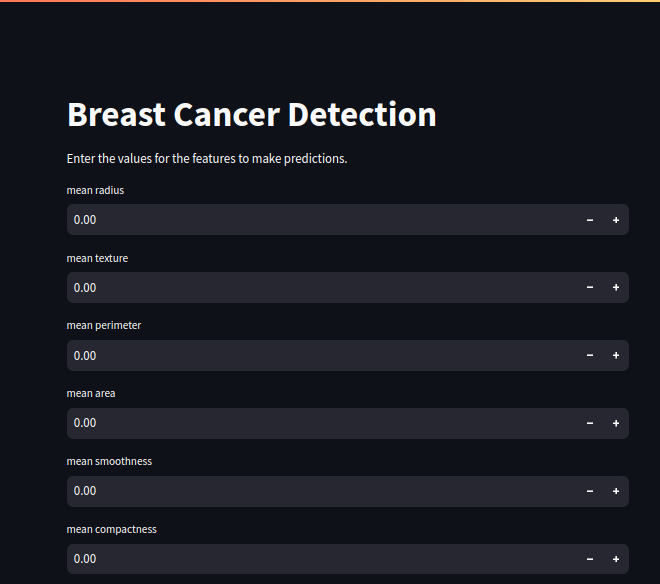
Image processing and machine learning techniques used in computer-aided detection system for mammogram screening-A review. International Journal of Electrical and

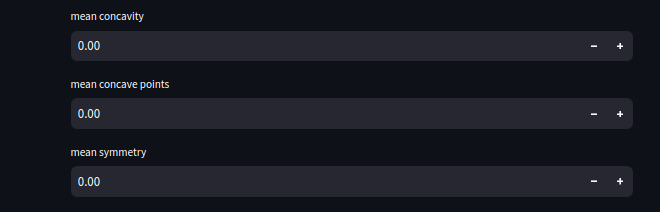
Computer Engineering, 10(3), 2336. Bakker, M. F., de Lange, S. V, Pijnappel, R. M., Mann, R. M., Peeters, P. H. M., Monninkhof, E. M., Emaus, M. J., Loo, C. E., Bisschops, R. H. C., Lobbes, M. B. I., & others. 2019). Supplemental MRI screening for women with extremely dense breast tissue. New England Journal of

Medicine, 381(22), 2091–2102.

# Appendix

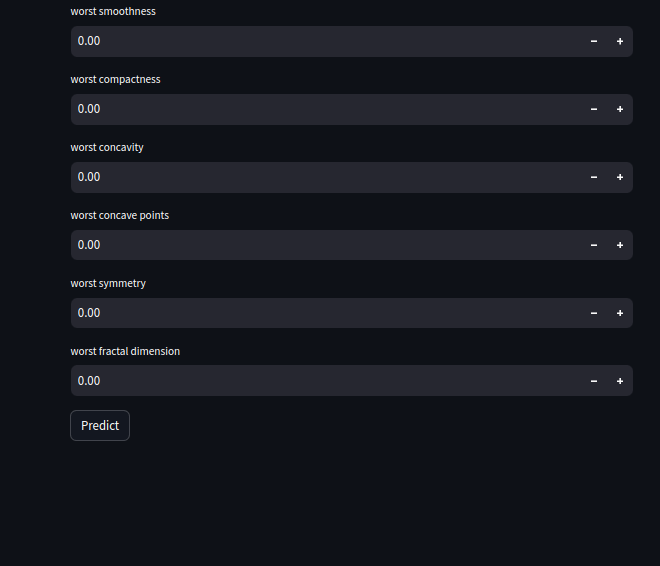
**Appendix** : User manual

****

****

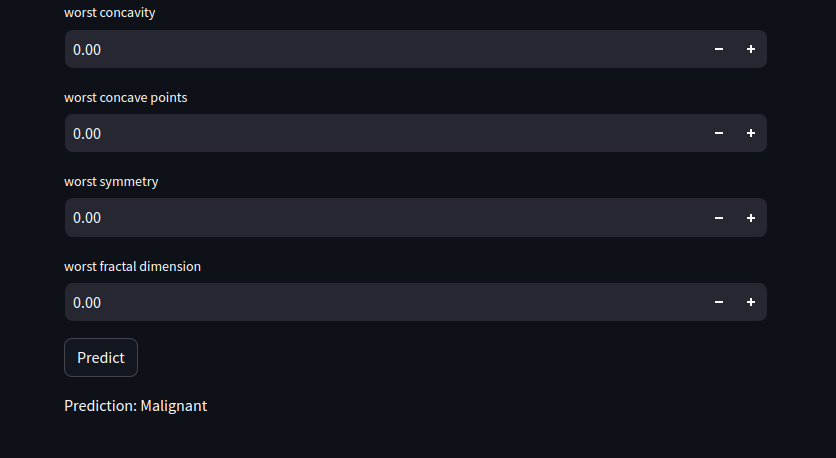
****

****



**The above images are the input forms**

**The below images are displaying output which one is supposed to expect**

****

The above image is an output display if user tries to predict without inputing values

