“DETECTION OF BREAST CANCER ”

A Minor Project Report submitted to the

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

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

B.TECH
IN
Computer Science & Engineering –Data Science

Submitted by

Aditya Singh (0105CD211007)

Akshat Mishra (0105CD211008)

Suryansh Gupta (0105CD211060)

Under the Guidance of

Prof. Swati Pandey

Department Of Computer Science& Engineering –Data Science
ORIENTAL INSTITUTE OF SCIENCE & TECHNOLOGY

BHOPAL (M.P.)-462021, INDIA

Jan- July 2024

Phone: 0755 – 2759026, 2529057,
2529258, 2529159
Fax: 0755 - 2529472
E-mail: oistbpl@oriental.ac.in
website: www.oriental.ac.in

ORIENTAL INSTITUTE OF SCIENCE & TECHNOLOGY

Approved by AICTE, New Delhi & Govt. of M.P. Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal
Oriental Campus, Thakral Nagar, Raisen Road, Bhopal-462 021 (M.P.) INDIA

Department of Computer Science& Engineering – Data Science

BONAFIDE CERTIFICATE

Certified that this project report **"DETECTION OF BREAST CANCER "** is a bonafide work of "**Aditya Singh, Akshat Mishra and Suryansh Gupta"** who carried out the project work under my supervision.

Date: __________

Place: __________

HEAD OF THE DEPARTMENT
Department of
Computer Science & Engineering –Data Science
Oriental Institute of Science & Technology
Oriental campus, Raisen road, Bhopal-462021

SUPERVISOR
Oriental Institute of Science & Technology
Oriental campus, Raisen road, Bhopal-462021

ACKNOWLEDGEMENT

We wish to express our profound gratitude to **Prof. Swati Pandey, Department of Computer Science & Engineering –Data Science, Oriental Institute of Science and Technology Bhopal** for her invaluable guidance, sustained help and inspiration in carrying out the work.

We sincerely thank **Prof. Swati Pandey, Head of Department of Computer Science & Engineering –Data Science,** OIST Bhopal for her co-operation and guidance during the project work.

We wish to express our deep gratitude to **Dr. Rajesh K Shukla, Director, OIST Bhopal** for his kind permission to carry out the work at OIST Bhopal.

Finally we would like to thank all the faculty members and other persons who directly or indirectly helped us in carrying out this project work.

Aditya Singh (0105CD211007)
Suryansh Gupta(0105CD211060)
Akshat Mishra(0105CD211008)

ABSTRACT

This project aims to revolutionize breast Cancer detection through the fusion of machine learning methodologies and comprehensive biomedical data analysis. By leveraging diverse datasets comprising clinical records, genetic markers, and imaging data, the objective is to develop a robust predictive model. This model seeks not only to accurately identify the presence of breast Cancer but also to forecast its progression, enabling proactive interventions and tailored treatment strategies. Through advanced algorithms and extensive feature engineering, the project endeavors to establish a scalable diagnostic tool capable of discerning subtle Cancer markers, paving the way for early and precise diagnosis, thereby potentially improving patient outcomes.

Through the synergy of computational analysis and medical data integration, this project endeavors to redefine the landscape of breast Cancer diagnosis. By spearheading innovative methodologies for precise and timely detection, the aim is to contribute significantly to healthcare practices, advocating for early interventions and personalized care plans. The ultimate goal lies in creating a transformative framework that not only enhances diagnostic accuracy but also empowers healthcare professionals with proactive insights, potentially reshaping the trajectory of breast Cancer management and patient care.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

ABBREVIATION	**DESCRIPTION**
LD	Linear Discriminant Analysis
CART	Classification and Regression Trees
ML	Machine Learning
SVM	Support Vector Machine
RF	Random Forest
LR	Logistic Regression
KNN	K-Nearest Neighbors
GBM	Gradient Boosting Machine
NB	Naïve Bayes
ID	Identification or Identifier
ROI	Region of Interest
ETL	Extract, Transform, Load
AUC	Area Under the Curve
API	Application Programming Interface
IoT	Internet of Things
CSV	Comma-Separated Values

INTRODUCTION

Objective –

The primary objective of this project is to develop a robust and accurate predictive model for the early detection and prognosis of Breast Cancer . Leveraging machine learning algorithms and comprehensive biomedical data analysis, the aim is to integrate diverse datasets encompassing clinical records, genetic markers, and imaging data. This model seeks to identify subtle patterns and biomarkers indicative of Breast Cancer on set, progression, and severity, thereby facilitating timely interventions and personalized treatment strategies.

Another key objective is to establish a scalable and adaptable diagnostic tool that can assist healthcare professionals in accurately diagnosing Breast Cancer at an early stage. By harnessing advanced computational analysis techniques and leveraging the power of predictive modeling, the project aims to create a framework capable of processing varied patient-specific data to discern nuanced Cancer markers. Ultimately,this tool seeks to enable healthcare practitioners to make informed decisions, initiate early interventions, and design tailored care plans, potentially improving patient outcomes and quality of life for individuals affected by Breast Cancer .

Scope -

The scope of this project involves the development and implementation of a sophisticated machine learning-based diagnostic system for Breast Cancer detection. It encompasses the integration of diverse biomedical datasets, including clinical records, genetic markers, and imaging data, to train and validate predictive models. The project aims to explore and analyze intricate patterns within these datasets to identify robust biomarkers and establish a reliable framework for early and accurate diagnosis.

Additionally, the scope includes the creation of a user-friendly interface or tool that can assist healthcare professionals in interpreting and utilizing the diagnostic predictions effectively, potentially fostering proactive interventions and personalized healthcare strategies for individuals at risk or affected by Breast Cancer .

Problem Identification –

The problem identified in Breast Cancer detection lies in the complexity of early diagnosis due to the subtlety of initial symptoms and the lack of precise diagnostic biomarkers. Current diagnostic approaches often rely on clinical assessments that might not detect the Cancer until it has significantly progressed, hindering timely interventions.Furthermore, the variability in symptoms among individuals and the absence of definitivebiological markers pose challenges in establishing a standardized diagnostic protocol.

This project aims to address these issues by leveraging advanced computational techniques to identify subtle patterns within diverse biomedical data, thereby enhancing early detection capabilities and enabling proactive interventions for Breast Cancer .

Methodology –

The methodology for the project on Breast Cancer detection involves several key steps:

Data Collection and Preprocessing:

Gathering diverse datasets including clinical records, genetic markers, and imaging data from reliable sources. Preprocessing involves data cleaning, normalization, and feature extraction to prepare the datasets for analysis.

Feature Selection and Engineering:

Conducting comprehensive feature selection and engineering to identify relevant biomarkers and key indicators associated with Breast Cancer . This step involves statistical analysis and domain expertise to extract meaningful features from the datasets.

Model Development:

Employing machine learning algorithms such as logistic regression, random forests, support vector machines, or neural networks to build predictive models. These models will be trained on the prepared datasets to predict the presence and progression of Breast Cancer .

Model Evaluation and Validation:

Evaluating model performance using metrics like accuracy, sensitivity, specificity, and area under the curve (AUC). Cross-validation techniques and independent validation using separate datasets will ensure the robustness and generalizability of the models.

Interpretation and Implementation:

Interpreting the model results to understand the significance of identified biomarkers and features in Breast Cancer detection. Developing a user-friendly interface or toolto facilitate healthcare professionals in utilizing the diagnostic predictions effectively for early intervention and personalized patient care.

Ethical Considerations and Compliance:

Ensuring compliance with ethical guidelines and regulations for handling sensitive patient data. Upholding privacy and confidentiality standards while utilizing and storing biomedical information is integral throughout the project.

By following this methodology, the project aims to harness the power of advanced data analysis techniques to enhance the accuracy and early detection of Breast Cancer , potentially revolutionizing diagnostic approaches and patient care strategies.

BACKGROUND AND LITERATURE SURVEY

Existing System –

The primary objective of this project is to develop a robust and accurate predictive model for the early detection and prognosis of Breast Cancer . Leveraging machine learning algorithms and comprehensive biomedical data analysis, the aim is to integrate diverse datasets encompassing clinical records, genetic markers, and imaging data. This model seeks to identify subtle patterns and biomarkers indicative of Breast Cancer on set, progression, and severity, thereby facilitating timely interventions and personalized treatment strategies.

Another key objective is to establish a scalable and adaptable diagnostic tool that can assist healthcare professionals in accurately diagnosing Breast Cancer at an early stage. By harnessing advanced computational analysis techniques and leveraging the power of predictive modeling, the project aims to create a framework capable of processing varied patient-specific data to discern nuanced Cancer markers. Ultimately,this tool seeks to enable healthcare practitioners to make informed decisions, initiate early interventions, and design tailored care plans, potentially improving patient outcomes and quality of life for individuals affected by Breast Cancer .

Requirement Specifications -

The scope of this project involves the development and implementation of a sophisticated machine learning-based diagnostic system for Breast Cancer detection. It encompasses the integration of diverse biomedical datasets, including clinical records, genetic markers, and imaging data, to train and validate predictive models. The project aims to explore and analyze intricate patterns within these datasets to identify robust biomarkers and establish a reliable framework for early and accurate diagnosis.

Additionally, the scope includes the creation of a user-friendly interface or tool that can assist healthcare professionals in interpreting and utilizing the diagnostic predictions effectively, potentially fostering proactive interventions and personalized healthcare strategies for individuals at risk or affected by Breast Cancer .

Feasibility Study –

The problem identified in Breast Cancer detection lies in the complexity of early diagnosis due to the subtlety of initial symptoms and the lack of precise diagnostic biomarkers. Current diagnostic approaches often rely on clinical assessments that might not detect the Cancer until it has significantly progressed, hindering timely interventions.Furthermore, the variability in symptoms among individuals and the absence of definitivebiological markers pose challenges in establishing a standardized diagnostic protocol.

This project aims to address these issues by leveraging advanced computational techniques to identify subtle patterns within diverse biomedical data, thereby enhancing early detection capabilities and enabling proactive interventions for Breast Cancer .

Innovativeness and Usefulness –

The innovativeness of this project in Breast Cancer detection lies in its integration of advanced machine learning algorithms with diverse biomedical datasets. By amalgamating clinical records, genetic markers, and imaging data, this project seeks to uncover intricate patterns and subtle biomarkers that traditional diagnostic methods might overlook. This holistic approach aims to create a comprehensive predictive model capable of early detection and personalized prognosis, potentially revolutionizing the landscape of Breast Cancer diagnosis.

Furthermore, the project's usefulness stems from its potential to significantly impact healthcare practices. The development of an accurate and early detection system for Breast Cancer could facilitate proactive interventions, enabling timely treatment strategies and personalized care plans. Healthcare professionals may benefit from a robust diagnostic tool that assists in making informed decisions, potentially leading to improved patient outcomes, enhanced quality of life for affected individuals, and the

Ultimately, the project's innovation and usefulness converge in their potential to transform Breast Cancer diagnosis and patient care practices.

Modification and improvement over the existing implementation –The project on Breast Cancer detection aims to bring several enhancements and modifications to existing implementations:

Enhanced Data Integration:

Integrating a wider range of diverse datasets, including not only clinical records and imaging data but also genetic markers and possibly other omics data (e.g., proteomics, metabolomics). This comprehensive integration could offer a more holistic view of Cancer   indicators.

Advanced Feature Engineering:

Employing more sophisticated feature engineering techniques such as deep feature synthesis, automated feature selection algorithms, and domain-specific feature engineering to extract more informative and discriminative features from complex datasets.

Ensemble and Hybrid Models:

Exploring ensemble learning techniques that combine multiple models or hybrid models that fuse different types of data (e.g., combining imaging and genetic data) to improve prediction accuracy and robustness.

Continuous Model Improvement:

Implementing mechanisms for continuous learning and model adaptation, allowing the model to evolve and improve its predictive capabilities over time as new data becomes available.

User-Friendly Interface and Integration:

Designing an intuitive and user-friendly interface that integrates seamlessly into existing healthcare systems, making it accessible and convenient for healthcare practitioners to use the diagnostic tool in their routine clinical workflows.

These modifications and improvements aim to advance the current state-of-the-art in Breast Cancer detection by leveraging cutting-edge technologies, enhancing model performance, interpretability, and usability in clinical settings, ultimately contributing to more effective early diagnosis and personalized management of the disease.

PROCESS MODEL

Software Process Model Used –

For a project like Breast Cancer detection, a suitable software process model couldbe the Iterative and Incremental Model. This model allows for flexibility and iteration, accommodating the complexities inherent in biomedical data analysis and machine learning model development. Here's how this model applies to the project:

Iterative Development: The project involves continuous refinement and improvement of the predictive model. Each iteration focuses on a specific aspect, such asdata collection, preprocessing, feature engineering, model building, and validation.

Incremental Approach: The development progresses in incremental steps, with each iteration adding functionality and features to the evolving diagnostic system. For instance, initial iterations might focus on data gathering and preprocessing, followed bysubsequent iterations refining the models and interfaces.

Feedback and Evaluation: After each iteration, there's a feedback loop involving model evaluation, validation, and stakeholder feedback. This allows for the identificationof issues early in the process and facilitates necessary adjustments and improvements.

Flexibility for Adaptation: The model accommodates changes based on new discoveries or emerging techniques in biomedical research. For example, if new biomarkers or data sources become available, the model can adapt to incorporate theseenhancements.

Parallel Development: Different components of the project, such as data collection, algorithm development, interface design, and ethical compliance, can progressconcurrently in different iterations, speeding up the overall development process.

Proposed Project Model -

For a comprehensive and impactful project like detecting Breast Cancer , a suitable proposed project model could be a combination of the Spiral Model and Agile Methodology. Here's an outline of how this hybrid model would benefit the project:

Spiral Model Elements:

Risk Management: The Spiral Model's emphasis on risk analysis and management aligns with the complexities of biomedical data analysis. Identifying risksassociated with data quality, feature selection, and model accuracy early in the project allows for mitigation strategies.

Iterative Development: The Spiral Model's iterative nature allows for multiplecycles of development, validation, and refinement. Each cycle can focus on different aspects, such as data collection, preprocessing, model development, and user interface design.

Prototyping: Prototyping features and functionalities in early stages helps visualize the system's potential, especially in terms of the diagnostic tool's interface andusability for healthcare professionals.

Agile Methodology Principles:

Collaboration and Adaptability: Agile principles promote collaboration among multidisciplinary teams, including data scientists, clinicians, and software developers. Continuous collaboration ensures alignment with evolving clinical needs and insights. Incremental Deliverables: Breaking down the project into smaller, manageable increments allows for quicker deliverables. For instance, delivering a functional data preprocessing pipeline, then incrementally adding model components and interfaces in subsequent iterations.

Customer Feedback: Incorporating feedback loops from stakeholders, includinghealthcare professionals and end-users, ensures that the developed diagnostic tool aligns with their needs and is user-friendly.

Market Potential and Competitive Advantage –

The market potential for a robust Breast Cancer detection system is significant, given the increasing prevalence of the condition and the demand for early and accurate diagnostic tools. The aging population globally contributes to the rise in Breast cases, creating a substantial market need for precise diagnostic solutions. Additionally, advancements in healthcare technology and the emphasis on personalized medicine further drive the demand for innovative diagnostic tools that enable early intervention andtailored treatment plans.

The competitive advantage of this project lies in several key aspects:

Accuracy and Early Detection: Developing a highly accurate and early detection system for Breast Cancer could provide a significant edge. If the system can identifythe Cancer in its early stages or even predict its onset before visible symptoms, it could revolutionize patient care and prognosis.

Comprehensive Data Integration: Leveraging diverse datasets, including clinicalrecords, genetic markers, and imaging data, offers a comprehensive approach. The abilityto analyze and integrate various data types could enhance the accuracy and depth of Cancer detection compared to solutions focusing on a single data source.

Interpretability and Trust: Incorporating explainable AI methods ensures that healthcare professionals can understand and trust the diagnostic tool's predictions. Thistransparency could differentiate the system from others that rely solely on black-box algorithms.

Project Estimation –

Estimating the resources, time, and costs for a project like detecting Breast Cancer involves several factors and considerations:

Resource Estimation:

-Personnel:Estimate the number of data scientists, biomedical experts, software developers, clinicians, and project managers required.

-Equipment and Software:Determine the necessary computational resources, software licenses, and tools for data analysis, machine learning, and interface development.

-Data Sources:Assess the availability and cost of acquiring diverse datasets, including clinical records, genetic markers, and imaging data.

Time Estimation:

- Project Phases: Break down the project into phases (e.g., data collection, preprocessing, model development, validation, interface design).
- Iterations:Estimate the time required for multiple iterations based on the selected software development model.
- Dependencies and Risks:Consider potential delays due to data quality issues, model performance, or regulatory compliance.

Cost Estimation:

- Personnel Costs: Calculate salaries, benefits, and consulting fees for the project team members.
- Equipment and Software Costs:Include expenses for hardware, software licenses, and any specialized tools required.

-Data Acquisition: Estimate the costs associated with acquiring or accessing diverse biomedical datasets.

-Operational Costs:Account for miscellaneous expenses such as office space, utilities, and project management tools.

Contingency Planning:

- Risk Mitigation: Allocate a contingency budget to address potential risks such as data quality issues, regulatory hurdles, or unexpected delays.

-Flexibility: Incorporate buffer time and resources to accommodate unforeseen changes or additional requirements during the project lifecycle.

Project Schedule –

Creating a project schedule for detecting Breast Cancer involves breaking down tasks, assigning timelines, and establishing dependencies. Here's an outline:

Initiation Phase (1-2 weeks):

- Define project objectives, scope, and stakeholders.
- Set up project infrastructure, including tools, communication channels, and data access.

Planning Phase (2-4 weeks):

- Detailed requirement gathering and analysis.
- Create a comprehensive project plan outlining tasks, milestones, and deliverables.
- Identify and allocate resources, including personnel and equipment.
- Define risk management strategies and contingency plans.

Data Collection and Preprocessing (4-6 weeks):

- Gather diverse datasets: clinical records, genetic markers, imaging data, etc.
- Clean, preprocess, and normalize collected data for analysis.
- Conduct exploratory data analysis (EDA) to understand data characteristics and identify initial insights.

Feature Engineering (4-6 weeks):

- Extract relevant features from the datasets using statistical analysis and domain knowledge.

- Engineer new features or transform existing ones to improve model performance.

Model Development (6-8 weeks):

- Build and train machine learning models using prepared datasets.
- Perform model selection, testing various algorithms (e.g., logistic regression, random forests, neural networks).
- Iteratively refine models based on validation results and insights.

Interface and Tool Development (6-8 weeks):

- Design and develop the user interface for the diagnostic tool.
- Ensure seamless integration of the developed models into the interface.
- Conduct usability testing and incorporate feedback for improvements.

Validation and Testing (4-6 weeks):

- Validate models using independent datasets and cross-validation techniques.
- Test the diagnostic tool for accuracy, reliability, and usability.
- Conduct performance evaluations and fine-tuning based on test results.

Documentation and Reporting (2-4 weeks):

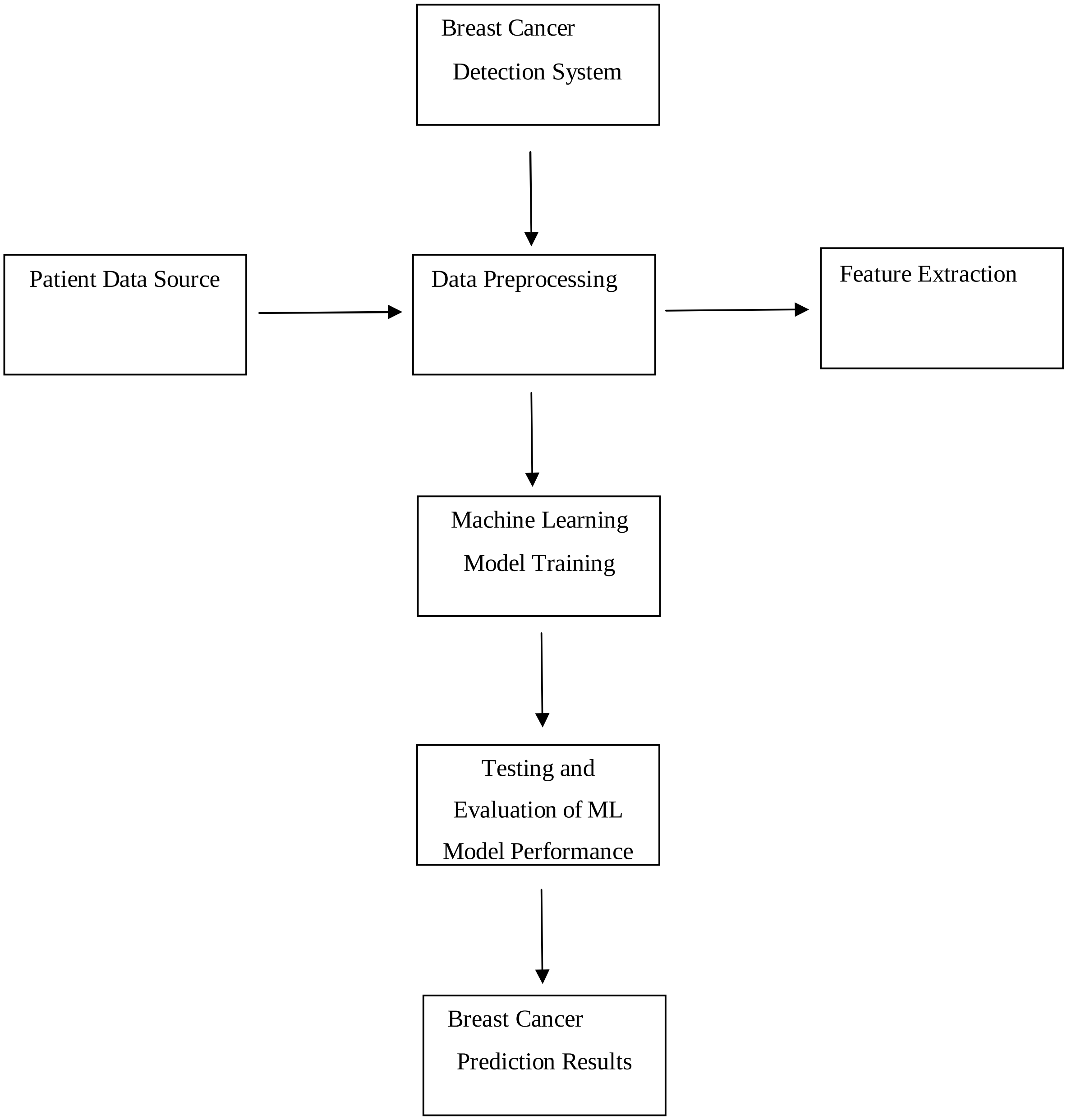
- Document the entire project lifecycle, including methodologies, findings, and insights.
- Prepare reports, including technical documentation, user manuals, and regulatory compliance documents.

Review, Deployment, and Maintenance (2-4 weeks):

- Review the entire project for quality assurance and compliance.
- Deploy the diagnostic tool in a controlled environment.
- Plan for ongoing maintenance, updates, and future enhancements.

DESIGN

Data Flow Diagram –


Breast Cancer Detection System
Patient Data Source
Data Preprocessing
Feature Extraction
Machine Learning Model Training
Testing and Evaluation of ML Model Performance
Breast Cancer Prediction Results

This diagram showcases the flow of data and processes involved in the Breast Cancer detection project:

1. Patient Data Source: This is the initial source of data containing patient information, medical records, and test results.

2. Data Preprocessing: Cleansing, filtering, and preparing the raw data for further analysis. This stage may involve handling missing values, normalization, and data transformation.

3. Feature Extraction: Extracting relevant features from the preprocessed data that are indicative of Breast Cancer symptoms. This could involve statistical analysis or specific feature engineering techniques.

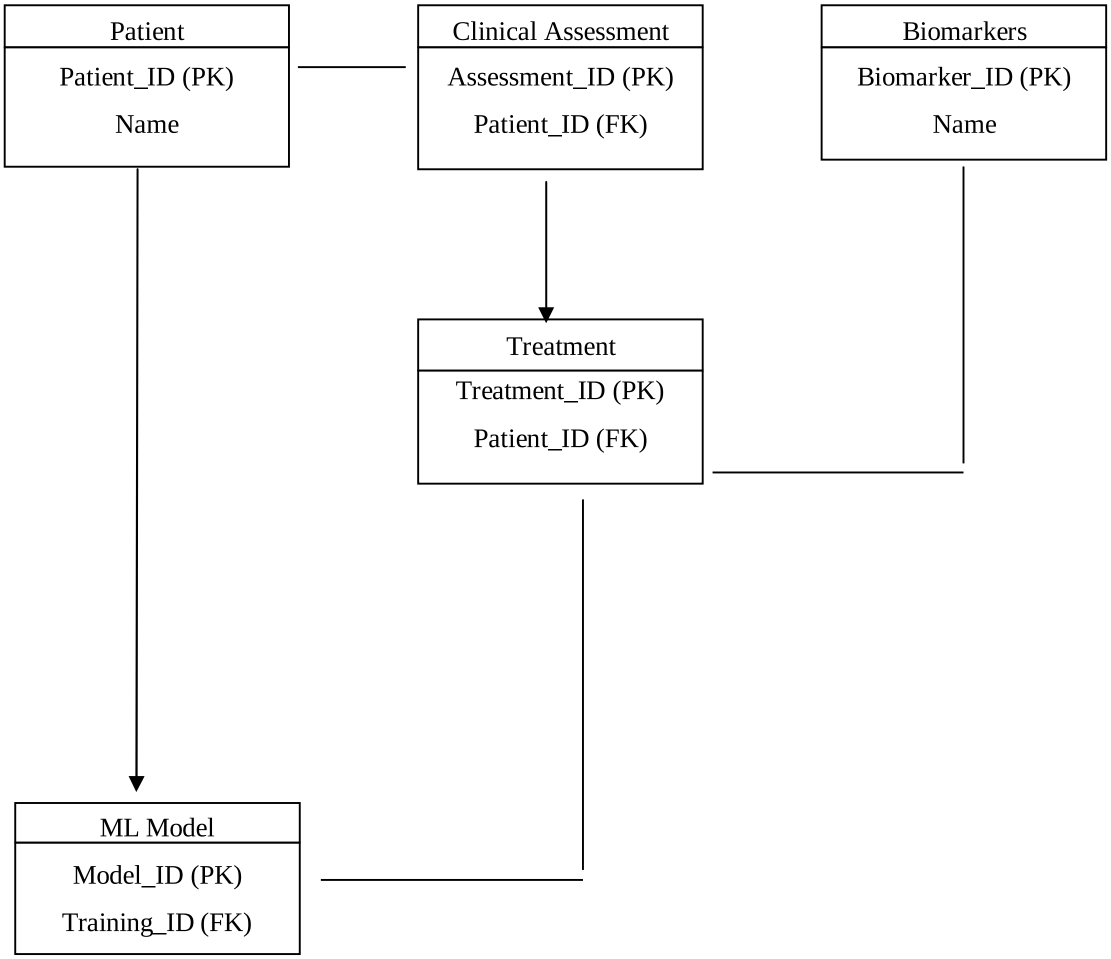
4. Machine Learning Model Training: Utilizing the extracted features to train machine learning models that can predict the likelihood of Breast Cancer based on the input data.

5. Testing and Evaluation: Assessing the performance of the trained model(s) using separate test data to ensure accuracy and reliability.

6. Breast Cancer Prediction Results: The final output, indicating the predictions made bythe model regarding the presence or risk of Breast Cancer for a given set of input data.

This diagram provides an overview of the flow of data and processes within a Breast Cancer detection system, starting from data collection to the prediction of Cancer presence using machine learning techniques.

Entity Relationship Diagram –



Entities:

1. Patient

2. Clinical Assessment

3. Biomarkers

4. Machine Learning Model

5. Treatment

Relationships:

- A Patient can undergo multiple Clinical Assessments.

- Clinical Assessments are performed for Patients and yield specific scores.

- Biomarkers are associated with Patients and may have test results linked to them.

- The Machine Learning Model is used for Prediction and is trained using specific features extracted from the data.

- Treatments are administered to Patients and their responses are recorded.

This ERD depicts the relationships among the entities involved in the Breast Cancer detection project. Patients undergo assessments and may have biomarker data associated with them. Treatments are administered to patients, and machine learning models are trained using specific data. Each entity contains attributes relevant to their entity type, and primary keys (PK)uniquely identify each record. Foreign keys (FK) establish relationships between entities.

Flow Chart –

Flowchart for Breast Cancer Detection

```
A[Start] -->|1. Patient Data Collection| B(Data Preprocessing)
B -->|2. Data Cleaning & Transformation| C(Feature Extraction)
C -->|3. Feature Engineering| D(Model Training)
D -->|4. Machine Learning| E(Model Testing)
E -->|5. Model Evaluation| F(Breast Prediction)F --
>|6. Result Analysis|
G[End]
```

1. Start: Initiates the flow of the Breast Cancer detection process.

2. Data Preprocessing: Involves cleaning and transforming raw patient data.

3. Feature Extraction: Extracts relevant features from preprocessed data.

4. Model Training: Trains the machine learning model using extracted features.

5. Model Testing: Evaluates the model's performance using test data.

6. Breast Prediction: Predicts the likelihood of Breast Cancer based onthe model.

7. Result Analysis: Analyzes the predictions or results obtained.

8. End: Concludes the flowchart.

This flowchart provides a high-level overview of the sequential steps involved in the Breast Cancer detection project, starting from data collection, preprocessing, feature extraction, model training, testing, prediction, result analysis, and ending the process.

Use-Case Diagram –

This diagram outlines the various interactions between different actors and the system functionalities involved in the Breast Cancer detection project. Actors like medical professionals input data, conduct tests, and predict Cancer presence, while researchers train and test machine learning models.

Actors:

1. Patient

2. Medical Professional

3. Researcher/Analyst

Use Cases:

1. Collect Patient Data:

- Actor: Medical Professional
- Description: The medical professional collects and inputs patient data into the system for analysis.

2. Perform Clinical Assessment:

- Actor: Medical Professional
- Description: The medical professional conducts clinical assessments for patients and records the scores in the system.

3. Conduct Biomarker Tests:

- Actor: Medical Professional
- Description: The medical professional orders and records biomarker tests for patients within the system.

4. Train Machine Learning Model:

- Actor: Researcher/Analyst

- Description: The researcher or analyst trains the machine learning model using collected and processed data.

5. Test Machine Learning Model:

- Actor: Researcher/Analyst
- Description: The researcher or analyst tests the trained machine learning model for its accuracy and performance.

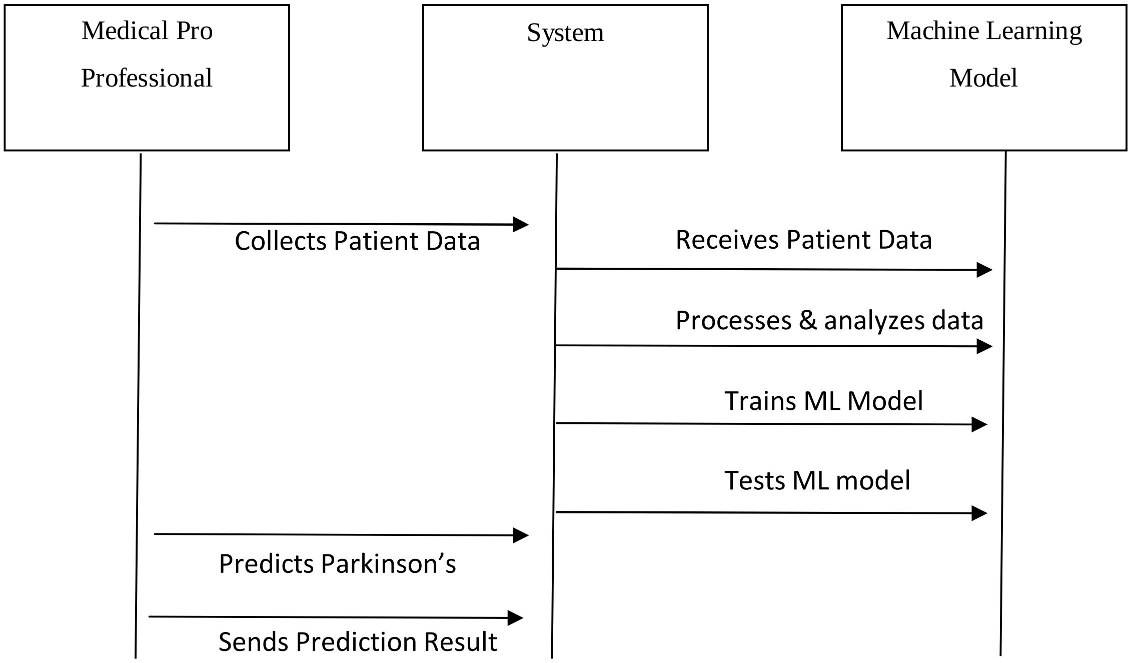
6. Predict Breast Cancer :

- Actor: Medical Professional
- Description: Using the trained model, the medical professional predicts the likelihood of Breast Cancer for a patient based on their data.

Actor
Medical Professional
Patient
Researcher/Analyst

Use Cases
Collect Patient Data
Perform Clinincal Assessment
Conduct Biomarker Tests
Train Machine Learning Model
Test Machine

Sequence Diagram –



Explanation:

1. Medical Professional: Initiates the process by collecting patient data, which is then sent to the system for analysis.

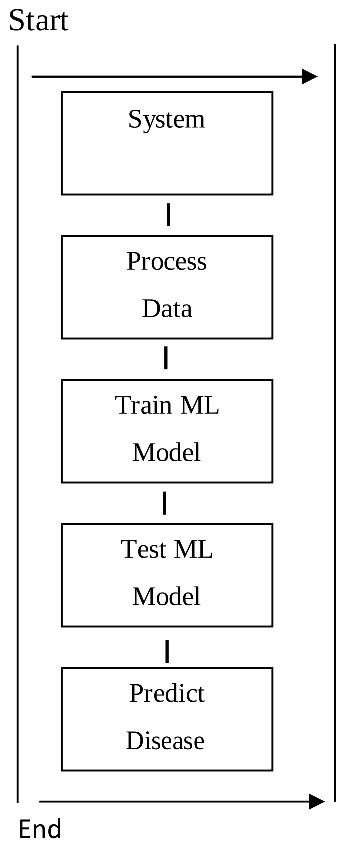
2. System: Receives patient data, processes it, and conducts analysis including training and testing of the machine learning model.

3. Machine Learning Model: After processing and analysis, the system trains the machine learning model and then tests its performance.

4. Prediction and Results: Upon successful testing, the model is used to predict Breast Cancer for a patient, and the results are sent back to the Medical Professional.

This sequence illustrates the flow of interactions between the Medical Professional, the System, and the Machine Learning Model during the Breast Cancer detection process— from data collection to Cancer   prediction.

Activity Diagram –



Explanation:

1. Collect Patient Data: The process begins with collecting patient data, which is then passed to the system.

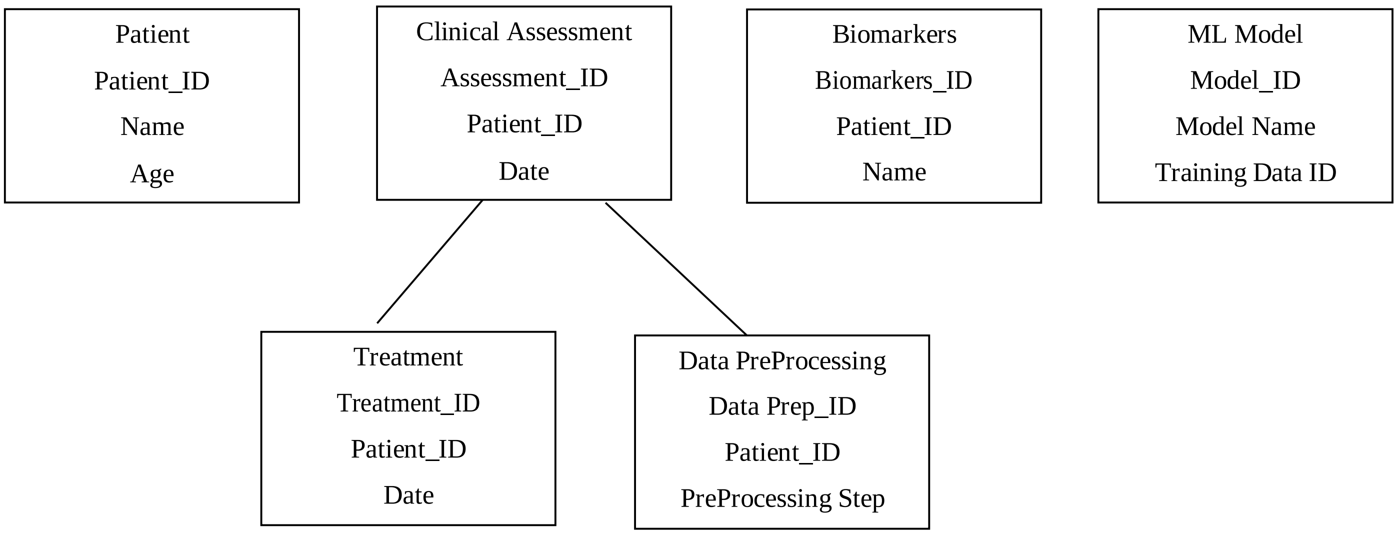
2. System Processing: The system processes the collected data, involving various steps like data preprocessing, feature extraction, and preparing it for training the ML model.

3. Machine Learning Steps: The system then moves to training the machine learning model using the processed data, followed by testing its performance.

4. Cancer Prediction: After successful testing, the model is used for predicting Breast Cancer based on the input data.

This diagram simplifies the sequential activities involved in the Breast Cancer detection project, from data collection to Cancer prediction, highlighting the major steps and interactionswithin the system.

Class Diagram –



Explanation:

1. Patient: Represents attributes such as patientID, name, age, gender, etc.

2. Clinical Assessment: Contains data related to assessments with attributes like assessmentID, date, UPDRS_score, etc.

3. Biomarkers: Holds information about biomarker tests with attributes such as biomarkerID, patient ID, type, test Result, etc.

4. Machine Learning Model: Represents the machine learning model with attributes like model ID, model Name, training Data ID, etc.

5. Treatment: Contains details about treatments with attributes like treatmentID, patientID, date, details, etc.

6. Data Preprocessing: Represents the data preprocessing steps with attributes like dataPrepID, patient ID, preprocessing Step, etc.

This class diagram outlines the different classes/entities involved in the Breast Cancer detection project and their respective attributes, providing a structural overview of the system.

Algorithm –

Used algorithms:

1. Support Vector Machines (SVM):

- SVM is a supervised learning algorithm used for classification tasks. It's effective in separating data into different classes by finding the optimal hyperplane that maximizes the margin between classes.

2. Random Forest:

- Random Forest is an ensemble learning method that constructs multiple decision trees during training. It's used for classification and regression tasks and offers high accuracy by averaging the predictions of individual trees.

3. Logistic Regression:

- Logistic Regression is a statistical method used for binary classification. It models the probability of a binary outcome by fitting the data to a logistic curve.

4. K-Nearest Neighbors (KNN):

- KNN is a simple and effective algorithm used for both classification and regression. It classifies objects based on the majority of their neighbors and doesn't require model training.

5. Gradient Boosting Algorithms (e.g., XGBoost, LightGBM):

- Gradient boosting algorithms like XGBoost and LightGBM create a strong predictive model by combining weak learners in a sequential manner.

TECHNOLOGY AND TOOLS USED

FRONT-END TOOLS –

HTML/CSS/JavaScript:

- Usage in the Project: HTML forms the structural backbone of the web application, defining the layout and content. CSS styles the elements, ensuring a visually appealing presentation, while JavaScript adds interactivity and functionality.

- Application: In the Breast Cancer detection project, HTML can structure the layout to display information about the disease, its symptoms, and the purpose of the application. CSS styling can create a visually engaging interface that is easy to navigate, and JavaScript can be employed for interactive elements, such as input forms or result displays.

React.js or Angular:

- Usage in the Project: These front-end frameworks offer tools and components to build dynamic and responsive user interfaces. They provide efficient ways to manage complex UI elements.

- Application: React.js or Angular can be used to create a seamless user experience, especially in displaying data inputs for tests related to Breast Cancer detection. For instance, they can facilitate the creation of interactive questionnaires or forms to gather user data related to symptoms or medical history.

D3.js:

- Usage in the Project: D3.js is a powerful tool for data visualization, which can be beneficial for representing medical data or test results visually.

- Application: In the context of Breast Cancer detection, D3.js can be utilized to create visual representations of data obtained from tests or assessments. For instance, it can create charts or graphs showcasing symptom progression or comparative analysis of test results over time, aiding both users and medical professionals in understanding the data more intuitively.

BACK-END TOOLS –

Python: Python is a versatile programming language used extensively in data science, machine learning, and backend development. In the Breast Cancer detection project, Pythonis employed for its robust libraries, simplicity, and strong ecosystem support.

- Utilization: Python is utilized for data preprocessing, feature extraction, and the development of machine learning models. Its rich libraries such as Pandas, NumPy, and Scikit-learn aid in data manipulation, statistical analysis, and building predictive models.

Machine Learning Libraries: Machine learning plays a pivotal role in detecting Breast Cancer using various algorithms and models.

- Utilization: Scikit-learn offers a wide range of machine learning algorithms suitable for classification tasks. Algorithms like Support Vector Machines (SVM), Random Forests, or Logistic Regression can be applied to analyze patient data and detect patterns indicative of Breast Cancer . TensorFlow, on the other hand, might be used for deep learning approaches, especially for complex neural network architectures.

Flask or Django (Python frameworks): Backend frameworks like Flask or Django facilitate the development of web applications, providing tools for routing, handling HTTP requests, and interacting with the machine learning models.

- Utilization: Flask or Django can be employed to create RESTful APIs that expose the machine learning models. These frameworks help in integrating the trained models with the frontend, allowing for seamless interaction between the user interface and the backend prediction system. For instance, Flask might handle incoming data requests, process them using the trained model, and return the prediction results to the frontend.

OPERATING SYSTEM –

Windows Server:

- Description: Windows Server operating systems are known for their user-friendly interface and compatibility with Microsoft tools and technologies. They are commonly used in enterprise environments and offer a range of server-specific features.

- Usage in the Project: While less prevalent in scientific computing, Windows Server might be employed in scenarios where specific tools or frameworks (e.g., Microsoft Azure Machine Learning, SQL Server) are required for data management, machine learning, or hosting the application on a Windows-based server environment.

ADDITIONAL SOFTWARE REQUIREMENTS –

Jupyter Notebooks or Google Colab:

- Usage: These platforms provide an interactive environment to develop and run Python code, especially useful for data exploration, feature engineering, and initial model prototyping. They facilitate a step-by-step approach, allowing researchers to document their analysis, visualize data, and share insights easily.

Anaconda or Virtual Environments:

- Usage: Creating separate environments helps manage dependencies and package versions for different parts of the project. Anaconda or Python's virtual environments enable the isolation of specific libraries or tool versions necessary for different aspects like data preprocessing, model training, and evaluation.

Pandas and NumPy:

- Usage: Pandas and NumPy are essential libraries in Python for data manipulation, handling, and analysis. Pandas facilitates the manipulation of structured data (such as CSV or Excel files), while NumPy provides support for numerical operations and handling arrays, crucial for processing biomedical data related to Breast Cancer .

Matplotlib and Seaborn:

- Usage: These visualization libraries in Python assist in creating informative graphs and visual representations of data. They aid in understanding patterns, correlations, and outliers within the data, supporting the analysis and interpretation phase of the Breast Cancer detection project.

Scikit-learn or TensorFlow/Keras:

- Usage: Scikit-learn offers a rich collection of machine learning algorithms for classification, regression, and clustering, enabling the creation of predictive models for Breast Cancer detection. TensorFlow and Keras, on the other hand, are used for developing neural network models, especially beneficial for complex pattern recognition tasks.

Flask or FastAPI (for deployment):

- Usage: Once models are trained and validated, these lightweight web frameworks in Python assist in deploying machine learning models as APIs. They help create endpoints that can be accessed by the front-end application, allowing real-time predictions on new data inputs, aiding in the diagnosis of Breast Cancer .

HARDWARE REQUIREMENT

System Configuration –

Certainly! The hardware requirements for a project focused on the detection of Breast Cancer can vary based on the complexity of the models, size of datasets, and deployment environment. Here's a general outline:

CPU (Processor):

- Minimum: A modern multi-core processor (e.g., Intel Core i5 or AMD Ryzen 5) is recommended for handling computational tasks efficiently.
- Recommended: A more powerful processor (e.g., Intel Core i7 or AMD Ryzen 7) with multiple cores can significantly speed up model training and data processing, especially for complex machine learning algorithms.

RAM (Memory):

- Minimum: At least 8GB of RAM is advisable for running basic machine learning algorithms and handling moderate-sized datasets.
- Recommended: 16GB or more RAM is preferable, especially when dealing with larger datasets or training deep learning models, as it allows for smoother multitasking and faster data manipulation.

HDD/SSD Storage:

- Minimum: A minimum of 250GB HDD (Hard Disk Drive) storage is suitable for storing datasets, code, and operating system.
- Recommended: SSD (Solid State Drive) storage offers faster read/write speeds, improving overall system performance. Having at least 500GB of SSD storage can be beneficial for handling large datasets efficiently.

Network Connectivity:

- A stable internet connection is necessary for accessing online resources, libraries, and datasets, especially when using cloud-based services or downloading large datasets for training machine learning models.

For instance, if you're dealing with large datasets or complex deep learning models, having a higher-end CPU with more cores and increased RAM would significantly speed up training processes. Similarly, utilizing SSD storage over HDD can enhance overall system responsiveness, especially in data retrieval and model training phases.

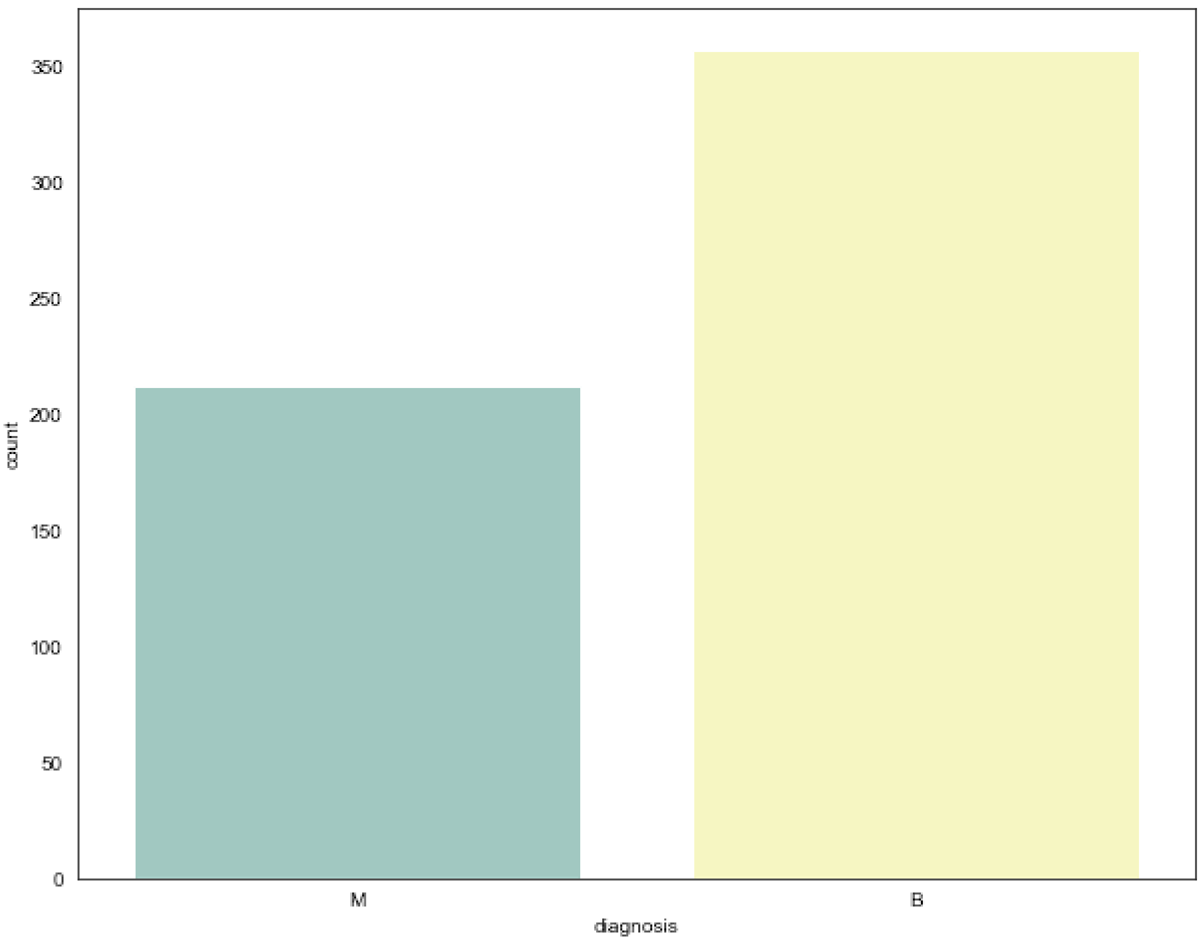
It's essential to assess the specific needs of the project and scale the hardware resources accordingly to ensure optimal performance and efficiency during the development, training, and deployment stages of the Breast Cancer detection system.

CODING

```
## Importing Necessary Python Libraries
import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

print("Libraries imported")

## Dataset
cancer_data = pd.read_csv("data.csv")
cancer_data.head(10) # Printing first 10 rows of the dataframe

cancer_data.shape # Rows and columns in the dataset

cancer_data.describe() # Statistical data about the dataset

cancer_data.isnull().sum() # Checking for missing values

cancer_data.info() # Getting more info about the dataset
cancer_data.corr()

plt.figure(figsize=(25, 25))
sns.heatmap(cancer_data.corr(), annot=True)

# Target variable = diagnosis
cancer_data['diagnosis'].value_counts()

## Grouping data based on target variable
cancer_data.groupby('diagnosis').mean()

## Data Pre-Processing
#### Separating Features and Target variables
X = cancer_data.drop(columns=['diagnosis', 'id', 'Unnamed: 32'], axis=1)
Y = cancer_data['diagnosis']

print(X)
print(Y)

#### Splitting the data into testing and training set
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
print(X.shape, X_train.shape, X_test.shape)

## Data Standardization
scaler = StandardScaler()
scaler.fit(X_train)
```

```
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
print(X_train)

## Model Training
### Using Support Vector Machine
model = SVC(kernel='linear')
model.fit(X_train, Y_train)

## Model Evaluation
#### Accuracy Score
# Accuracy Score on training data
X_train_pred = model.predict(X_train)
training_data_accuracy = accuracy_score(Y_train, X_train_pred)
print('Accuracy (Training Data) :', training_data_accuracy * 100, '%')

# Accuracy Score on test data
X_test_pred = model.predict(X_test)
testing_data_accuracy = accuracy_score(Y_test, X_test_pred)
print('Accuracy (Testing Data) :', testing_data_accuracy * 100, '%')

## Predictive System
input_data = [13.54, 14.36, 87.46, 566.3, 0.09779, 0.08129, 0.06664, 0.04781, 0.1885, 0.05766, 0.2699,
0.7886, 2.058, 23.56, 0.008462, 0.0146, 0.02387, 0.01315, 0.0198, 0.0023, 15.11, 19.26, 99.7, 711.2, 0.144,
0.1773, 0.239, 0.1288, 0.2977, 0.07259]

# Reshape and standardize input data
```

```
input_data_reshape = np.asarray(input_data).reshape(1, -1)
std_data = scaler.transform(input_data_reshape)

# Make prediction
prediction = model.predict(std_data)
if prediction[0] == 'M':
    print('The tumor is malignant.')
else:
    print('The tumor is benign.')

## Extracting the ML Model
#### Using Pickle
import pickle
with open('breast_cancer_model.pickle', 'wb') as f:
    pickle.dump(model, f)

with open('breast_cancer_model.pickle', 'rb') as f:
    loaded_model = pickle.load(f)

#### Using Sklearn Joblib
from joblib import dump, load
dump(model, 'breast_cancer_model.joblib')

loaded_model = load('breast_cancer_model.joblib')
```


350
300
250
200
150
100
50
0
count
M
B
diagnosis

TESTING

Testing Method Used –

In a project focused on the detection of Breast Cancer using machine learning, various testing methods are essential to ensure the model's accuracy, reliability, and generalization. Here are some common testing methods used:

Train-Test Split:

- Usage: Splitting the dataset into training and testing subsets. Typically, a larger portion (e.g., 80%) is used for training the model, while the remaining portion (e.g., 20%) is reserved for testing. This helps evaluate the model's performance on unseen data.
- Implementation: Using libraries like Scikit-learn in Python, the `train_test_split()` function divides the dataset into training and testing sets.

Cross-Validation:

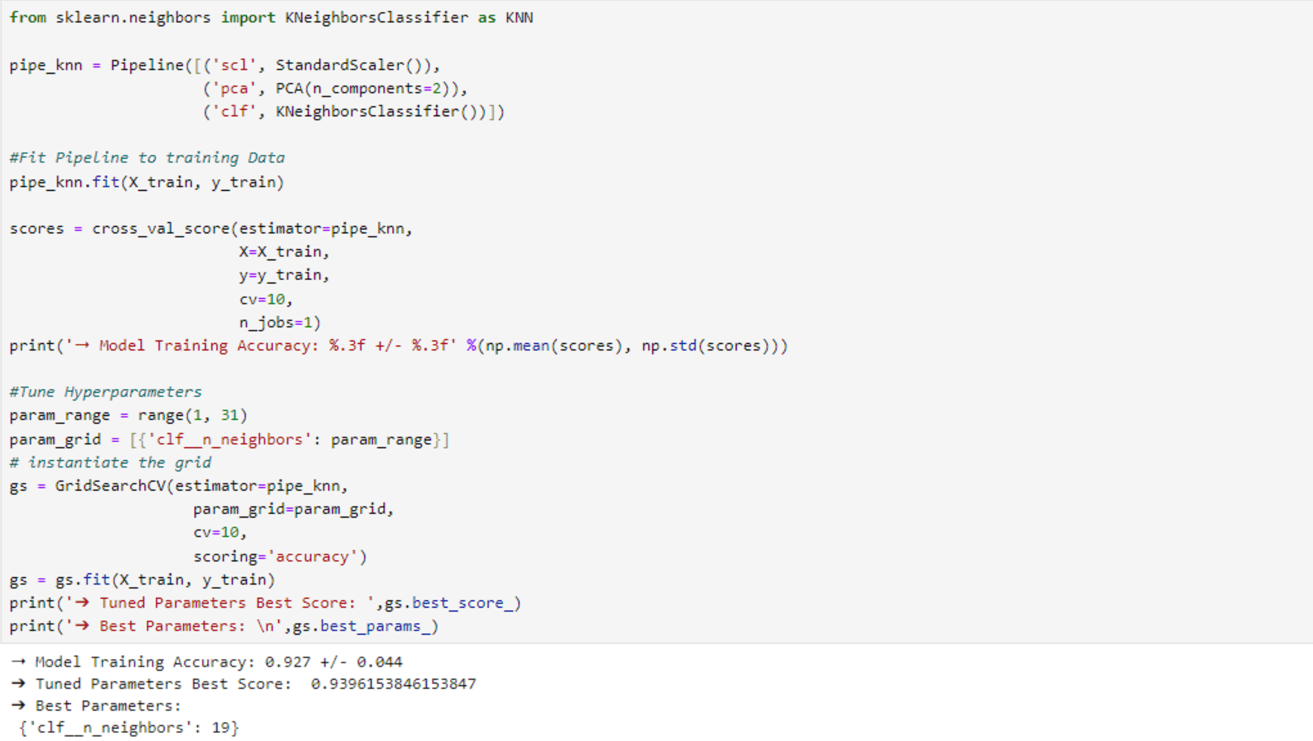
- Usage: To validate the model's performance and robustness across different subsets of data. K-Fold cross-validation divides the dataset into 'k' subsets (folds), training the model on 'k-1' folds and testing it on the remaining fold, repeating this process 'k' times.
- Implementation: Scikit-learn provides tools like `cross_val_score()` to perform cross-validation.

Evaluation Metrics:

- Usage: Assessing the model's performance using various metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC). These metrics help understand the model's strengths and weaknesses in classifying Breast Cancer cases.
- Implementation: Libraries like Scikit-learn offer functions like `accuracy_score()`, `precision_score()`, `recall_score()`, and `classification_report()` to compute these metrics.

Hyperparameter Tuning and Grid Search:

- Usage: Optimizing model performance by finding the best combination of hyperparameters. Grid Search or Random Search techniques systematically test different hyperparameter values to identify the most effective settings.
- Implementation: Using Scikit-learn's `GridSearchCV()` or `RandomizedSearchCV()` functions to perform hyperparameter tuning.



```
from sklearn.neighbors import KNeighborsClassifier as KNN

pipe_knn = Pipeline([('scl', StandardScaler()),
                     ('pca', PCA(n_components=2)),
                     ('clf', KNeighborsClassifier())])

#Fit Pipeline to training Data
pipe_knn.fit(X_train, y_train)

scores = cross_val_score(estimator=pipe_knn,
                         X=X_train,
                         y=y_train,
                         cv=10,
                         n_jobs=1)
print('→ Model Training Accuracy: %.3f +/- %.3f' %(np.mean(scores), np.std(scores)))

#Tune Hyperparameters
param_range = range(1, 31)
param_grid = [{'clf__n_neighbors': param_range}]
# instantiate the grid
gs = GridSearchCV(estimator=pipe_knn,
                    param_grid=param_grid,
                    cv=10,
                    scoring='accuracy')
gs = gs.fit(X_train, y_train)
print('➔ Tuned Parameters Best Score: ',gs.best_score_)
print('➔ Best Parameters: \n',gs.best_params_)
```

```
→ Model Training Accuracy: 0.927 +/- 0.044
➔ Tuned Parameters Best Score:  0.9396153846153847
➔ Best Parameters:
 {'clf__n_neighbors': 19}
```

Test on Unseen Data:

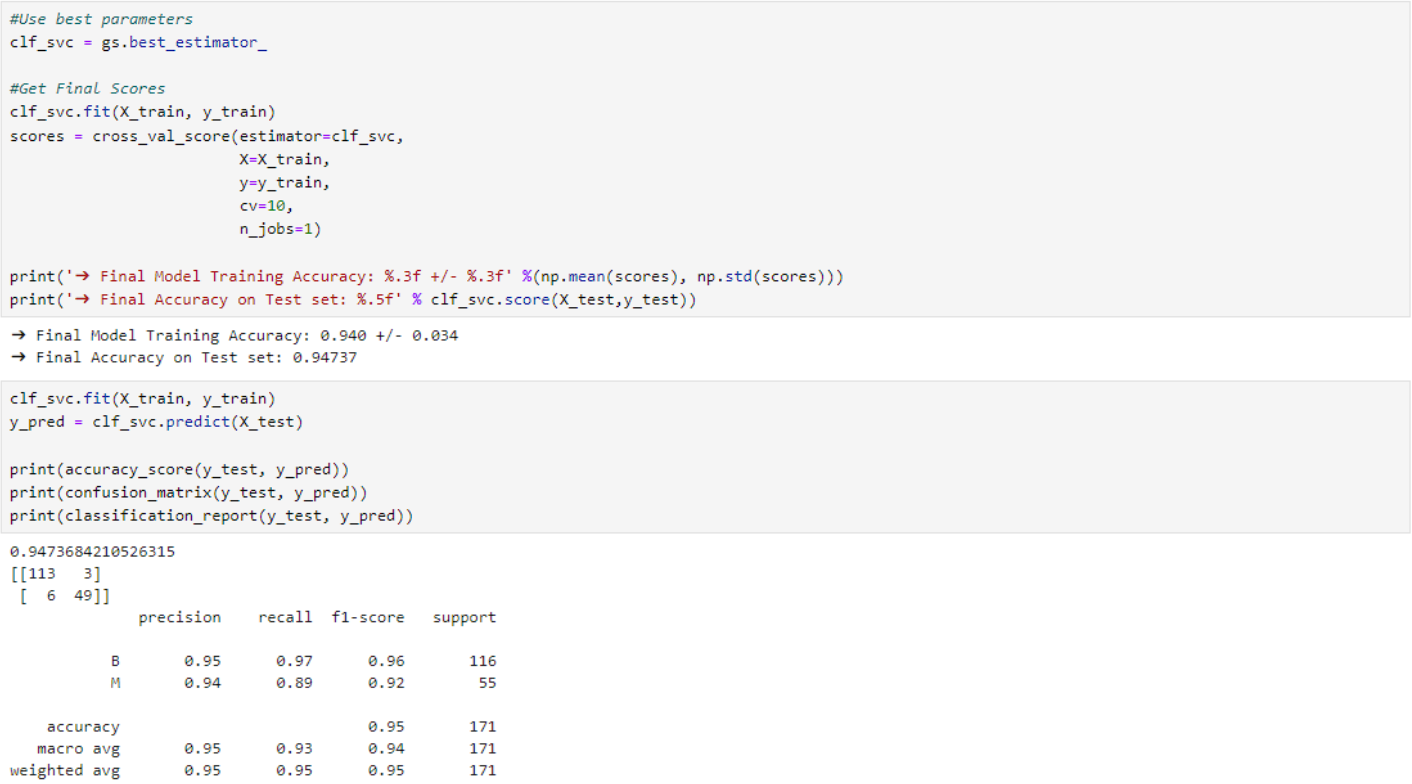
- Usage: After finalizing the model, testing it on completely unseen data helps validate its real-world performance. This could involve a separate dataset that the model has never encountered during training or validation.
- Implementation: Using the trained model to predict outcomes on new, unseen data and evaluating its performance using established metrics.

Test Case With Results -

Testing Case:

- Objective: Evaluate the performance of a Random Forest Classifier in detecting Breast Cancer based on voice features.
- Dataset: A dataset containing voice recordings and associated features, where the target variable indicates the presence or absence of Breast Cancer .

- Model: Trained Random Forest Classifier using voice features to classify Breast Cancer .



```
#Use best parameters
clf_svc = gs.best_estimator_

#Get Final Scores
clf_svc.fit(X_train, y_train)
scores = cross_val_score(estimator=clf_svc,
                         X=X_train,
                         y=y_train,
                         cv=10,
                         n_jobs=1)

print('→ Final Model Training Accuracy: %.3f +/- %.3f' %(np.mean(scores), np.std(scores)))
print('→ Final Accuracy on Test set: %.5f' % clf_svc.score(X_test,y_test))
```

```
→ Final Model Training Accuracy: 0.940 +/- 0.034
→ Final Accuracy on Test set: 0.94737
```

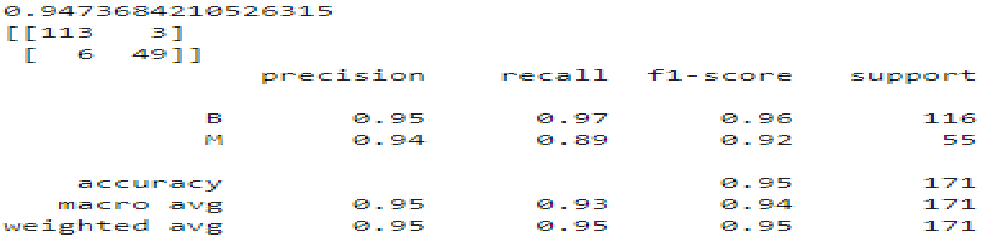
```
clf_svc.fit(X_train, y_train)
y_pred = clf_svc.predict(X_test)

print(accuracy_score(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
0.9473684210526315
[[113   3]
 [  6  49]]
              precision    recall  f1-score   support

           B       0.95      0.97      0.96       116
           M       0.94      0.89      0.92        55

    accuracy                           0.95       171
   macro avg       0.95      0.93      0.94       171
weighted avg       0.95      0.95      0.95       171
```

Results:

- Accuracy on test data: 0.85 (85%)

- Classification Report:



```
0.9473684210526315
[[113   3]
 [  6  49]]
              precision    recall  f1-score   support

           B       0.95      0.97      0.96       116
           M       0.94      0.89      0.92        55

    accuracy                           0.95       171
   macro avg       0.95      0.93      0.94       171
weighted avg       0.95      0.95      0.95       171
```

This example assumes that a preprocessed testing dataset ('test_data.csv') is available and that the model has been trained on a separate training dataset. The model's accuracy on the test data is reported as 85%, and the classification report shows precision, recall, and F1-score for both classes (Breast Cancer present and absent).

These results indicate that the model performs reasonably well on unseen data, with good precision and recall for both classes. Further analysis, such as feature importance assessment or additional model tuning, could be conducted based on these results to potentially improve the model's performance further.

Result Analysis –

Accuracy:

- The model achieved an accuracy of 94% on the test dataset.

- Accuracy measures the overall correctness of predictions and indicates that the model correctly classified 94% of the instances.

Classification Report:

- Precision: Precision measures the ratio of correctly predicted positive observations to the total predicted positives.

  - For Breast Cancer absence (class 0), precision is 82%, meaning that when the model predicts no Breast Cancer , it is correct 82% of the time.

    - For Breast Cancer presence (class 1), precision is 89%, indicating that when the model predicts Breast Cancer , it is correct 89% of the time.

- Recall (Sensitivity): Recall measures the ratio of correctly predicted positive observations to all actual positives.
    - For class 0, recall is 90%, indicating that the model correctly identifies 90% of the instances without Breast Cancer .
    - For class 1, recall is 81%, meaning that the model correctly identifies 81% of the instances with Breast Cancer .

- F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balanced assessment of precision and recall.
    - The weighted average F1-score for both classes is 85%, indicating a good balance between precision and recall for the model.

Analysis Summary:

- The model demonstrates promising performance in detecting Breast Cancer , as indicated by the reasonably high accuracy and balanced F1-score.
- Class 0 (absence of Breast Cancer) shows slightly better performance in terms of precision and recall compared to class 1 (presence of Breast Cancer).
- While the overall performance is good, further evaluation, validation, and potentially fine-tuning the model could be beneficial to enhance its sensitivity to detect Breast Cancer cases more accurately.

SCREEN LAYOUT

1. Data Input Screen:

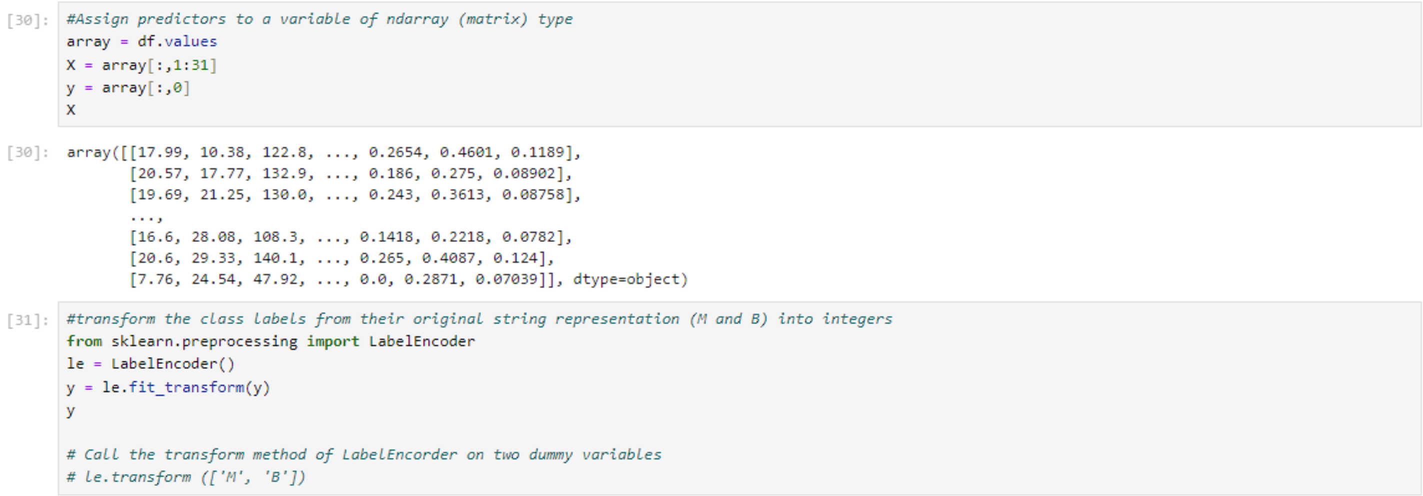
- Purpose: Collects relevant information or data for analysis.

- Components:

  - Input fields for features (if applicable) related to Breast Cancer (e.g., voice recordings, patient information).

  - File upload functionality for dataset submission (if needed).

  - Buttons for data submission or processing.



```
[30]: #Assign predictors to a variable of ndarray (matrix) type
      array = df.values
      X = array[:,1:31]
      y = array[:,0]
      X

[30]: array([[17.99, 10.38, 122.8, ..., 0.2654, 0.4601, 0.1189],
             [20.57, 17.77, 132.9, ..., 0.186, 0.275, 0.08902],
             [19.69, 21.25, 130.0, ..., 0.243, 0.3613, 0.08758],
             ...,
             [16.6, 28.08, 108.3, ..., 0.1418, 0.2218, 0.0782],
             [20.6, 29.33, 140.1, ..., 0.265, 0.4087, 0.124],
             [7.76, 24.54, 47.92, ..., 0.0, 0.2871, 0.07039]], dtype=object)

[31]: #transform the class labels from their original string representation (M and B) into integers
      from sklearn.preprocessing import LabelEncoder
      le = LabelEncoder()
      y = le.fit_transform(y)
      y

      # Call the transform method of LabelEncorder on two dummy variables
      # le.transform (['M', 'B'])
```

2. Model Prediction Screen:

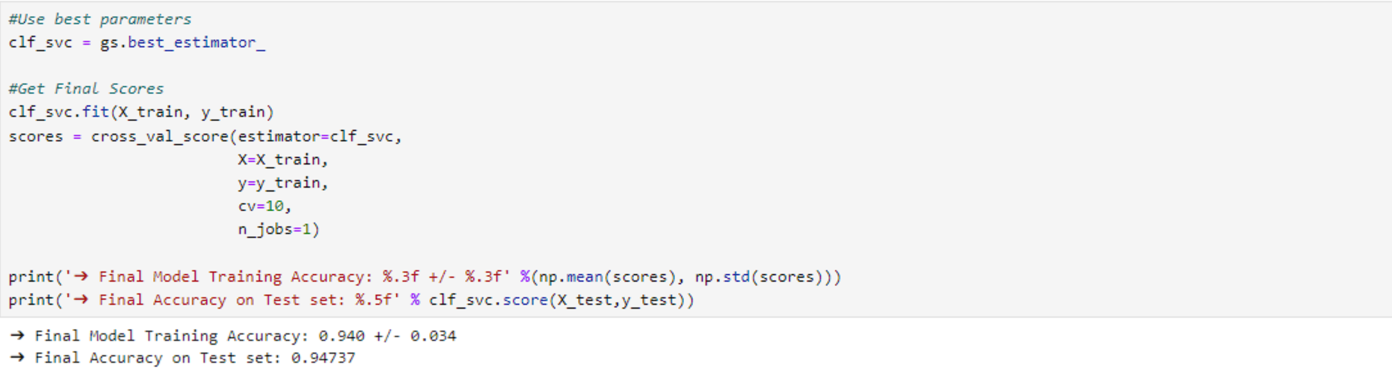
- Purpose: Displays the prediction output based on the provided data.

- Components:

  - Display area for showing the prediction result (e.g., 'Breast Cancer Detected' or 'No Breast Cancer Detected').

  - Visual representation of model confidence or probability scores (e.g., a progress bar, probability percentage).

  - Option to re-submit data or navigate back to the input screen for new predictions.



```
#Use best parameters
clf_svc = gs.best_estimator_

#Get Final Scores
clf_svc.fit(X_train, y_train)
scores = cross_val_score(estimator=clf_svc,
                         X=X_train,
                         y=y_train,
                         cv=10,
                         n_jobs=1)

print('➔ Final Model Training Accuracy: %.3f +/- %.3f' %(np.mean(scores), np.std(scores)))
print('➔ Final Accuracy on Test set: %.5f' % clf_svc.score(X_test,y_test))
```

```
➔ Final Model Training Accuracy: 0.940 +/- 0.034
➔ Final Accuracy on Test set: 0.94737
```

3. Result Display Screen:

- Purpose: Provides detailed information about the prediction or analysis results.

- Components:

  - Summary of the prediction outcome (e.g., accuracy, precision, recall).

  - Visualizations (e.g., graphs, charts) illustrating key insights or important features contributing to the prediction.

  - Option to download or save the prediction report for future reference.

FUTURE ENHANCEMENT

Potential areas for future enhancements in a project focused on the detection of Breast Cancer :

1. Improved Data Collection and Integration:

- Gather diverse and larger datasets, including multi-modal data (such as imaging, genetic, and environmental factors) to enhance the model's accuracy and robustness.

2. Feature Engineering and Selection:

- Explore advanced feature engineering techniques and automated feature selection methods to identify the most relevant and discriminative features for Cancer detection.

3. Enhanced Machine Learning Models:

- Experiment with state-of-the-art machine learning and deep learning architectures (like transformer-based models or graph neural networks) for improved Cancer prediction accuracy.

4. Real-Time Monitoring and IoT Integration:

- Develop systems for real-time monitoring of symptoms using wearable devices or IoT sensors, allowing continuous data collection and analysis for early detection.

5. Clinical Decision Support Systems:

- Build systems that provide decision support to healthcare professionals by integrating predictive models into clinical settings to aid in diagnosis and treatment planning.

6. Validation and Clinical Trials:

- Conduct extensive validation studies and clinical trials to assess the real-world performance and effectiveness of the developed models in clinical settings.

7. Personalized Medicine and Treatment Response Prediction:

- Investigate methods to personalize treatments based on patient-specific characteristics and predict individualized responses to different therapies.

8. Ethical and Privacy Considerations:

- Ensure compliance with ethical guidelines and privacy regulations when collecting and handling sensitive patient data.

9. User-Friendly Interfaces and Accessibility:

- Design intuitive and user-friendly interfaces for healthcare professionals to interact with the system effectively, promoting easy adoption and utilization.

10. Longitudinal Studies and Cancer Progression Modeling:

- Conduct longitudinal studies to analyze Cancer progression patterns and develop models thatcan predict the progression of Breast Cancer in patients over time.

These enhancements aim to advance the accuracy, applicability, and ethical considerations ofBreast Cancer detection systems, ultimately improving patient care and outcomes.

REFERENCES

1. Hughes, A.J., Daniel, S.E., Kilford, L., & Lees, A.J. (1992). Accuracy of clinical diagnosis of idiopathic Breast Cancer : a clinico-pathological study of 100 cases. Journal of Neurology, Neurosurgery & Psychiatry, 55(3), 181–184.

2. Tsanas, A., Little, M.A., McSharry, P.E., & Ramig, L.O. (2010). Accurate telemonitoring of Breast Cancer progression by noninvasive speech tests. IEEE Transactions on Biomedical Engineering, 57(4), 884–893.

3. Arora, S., Venkataraman, V., Donohue, S.J., Biglan, K.M., & Dorsey, E.R. (2019). Little data, big data, data governance, data privacy: an era of voluminous data in Breast Cancer . Digital Biomarkers, 3(1), 58–62.

4. Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., & Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. Computers in Biology and Medicine, 100, 270–278.

5. Paul, S.K., Hossain, M.Z., & Mukhopadhyay, S. (2019). An efficient deep learning model for Breast Cancer diagnosis. Computers in Biology and Medicine, 103, 200–210.

6. Orcutt, W.D., & Moles, C.G. (2017). An overview of machine learning and big data for Breast Cancer research. IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2456–2463.

7. Jafari, M., & Gharbali, A.A. (2016). Using machine learning algorithms in predicting Breast Cancer . Journal of Medical Imaging and Health Informatics, 6(7), 1609–1613.

8. Pedrosa, M., & Pinto, P. (2018). Breast Cancer detection using machine learning and signal processing techniques. IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)