THE CO-OPERATIVE UNIVERSITY OF KENYA (CUK)

SCHOOL OF COMPUTING & MATHEMATICS PROJECT

PROJECT REPORT

BREAST CANCER DIAGNOSIS PREDICTOR APPLICATION

BY

JOHNSON KANYI WAMWEYA

BCSC01/0019/2021

SUPERVISOR: ***MS. EDNA CHEPKEMOI***

Project Report submitted in partial fulfillment of the requirements for the award of the Bachelor of Science in Computer Science

©DECEMBER, 2024

DECLARATION AND APPROVAL

DECLARATION

I declared that this Project Report was my own work and had, to the best of my knowledge, not been submitted to any other institution of higher learning for any award.

Student Name: Johnson Kanyi Wamweya **Registration Number:** BCSC01/0019/2021

Signature: .... Date: ....

APPROVAL

This project Report had been submitted with my approval as the University supervisor.

Supervisor Name: Ms. Edna Chepkemoi

Signature: .... Date: ....

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who contributed to the successful completion of this project. First and foremost, I am deeply grateful to Ms. Edna, my university supervisor, for her continuous support, expert guidance, and valuable feedback throughout the project. Her mentorship played a crucial role in shaping the direction and successful implementation of this work. I also wish to acknowledge the faculty and staff of my institution for providing a conducive learning environment and the necessary resources to undertake this project. Special thanks go to my course mates for their unwavering encouragement and motivation throughout the development phase.

To all who have supported me directly or indirectly, thank you.

ABSTRACT

Breast cancer is the most frequent cancer in women, one in eight women worldwide, and continues to be the leading cause of cancer-related deaths among female patients. Despite the increase in diagnosis and treatment success rates, timely and efficient diagnosis remains a big concern because not all breast lesions are malignant and a benign lesion does not always further develop into cancer. Traditional diagnostic approaches like physical examinations, mammography, and fine-needle aspiration cytology, though reliable, can be complemented by the use of machine learning technologies in improving diagnostic accuracy. This project dealt with developing a web-based application for the prediction of breast cancer diagnosis using machine learning to classify breast masses as either benign or malignant based on diagnostic cell nuclei measurements from a breast tissue. The system was implemented based on the publicly available, Breast Cancer Wisconsin (Diagnostic) Data Set from Kaggle (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data) which contained 569 samples, each with ten real-valued cell features: radius, texture, compactness etcetera. A logistic regression model was trained in predicting the malignancy likelihood for an optimal balance between precision and recall. The model achieved a high accuracy of 98%. The application was developed in Python and using the Streamlit framework, it provided a web-based interactive dashboard for medical professionals to enter the data collected from a breast sample tissue and obtain predictions accompanied by visualizations based on Plotly for intuitive understanding. The expected outcome was a very accurate, user-friendly tool that simplifies diagnosis. The project demonstrated how ML can enhance diagnostic procedures, aligning with global health goals to reduce cancer mortality by supporting early detection and decision-making. By integrating technology into healthcare, this project bridged the gap between traditional diagnostics and modern AI-driven solutions, offering a scalable and accessible tool for clinical and educational use.

TABLE OF CONTENT

LIST OF ABBREVIATIONS

AI: Artificial Intelligence

BRCA1: Breast Cancer 1 Gene

BRCA2: Breast Cancer 2 Gene

CADx: Computer-Aided Diagnosis

CSS: Cascading Style Sheets

EDA: Exploratory Data Analysis

FNA: Fine Needle Aspiration

GDPR: General Data Protection Regulation

HIPAA: Health Insurance Portability and Accountability Act

IBM: International Business Machines

ML: Machine Learning

SDG: Sustainable Development Goals

SDLC: Software Development Life Cycle

UI: User Interface

UAT: User Acceptance Testing

UNSDGs: United Nations Sustainable Development Goals

DFD: Data Flow Diagram

OS: Operating System

ERD: Entity-Relationship Diagram

UML: Unified Modelling Language

CSV: Comma-Separated Values

RDBMS: Relational Database Management System

TC: Test Case

SVM: Support Vector Machine

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1- INTRODUCTION

1.1 Background to the Study

Breast cancer is one of the most common cancers causing death in women all over the world. Early detection and diagnosis of the disease is important in its management and improvement of survival rates among patients. Traditionally, diagnosis has been done by analysing a breast mass through clinical examination imaging and cytology, which can be very time-consuming and prone to human error. More recently, the emerging use of Artificial Intelligence and Machine Learning has made promising new tools available that will help medical professionals diagnose breast cancer more accurately and with greater efficiency. Machine learning models can analyse big datasets of medical information for patterns that might be too subtle for human observation. These models can then be trained to predict whether a breast mass is benign or malignant based on different cell measurements taken as input for diagnosis. The Breast Cancer Diagnosis Predictor Application, therefore, tried to put this technology into application to avail the tool with which medical professionals can effectively conduct early detection of cancerous cases, thereby upgrading the accuracy of diagnosis and decisiveness with speed.

1.2 Statement of the Problem

While medical technology is improving, the diagnosis of breast cancer remains a very challenging process often requiring time-consuming manual analysis of diagnostic data. Medical professionals are specifically tasked with reviewing complex cytology results to determine whether a breast mass is benign or malignant. The methods currently used are prone to delay, human error, and lack of uniformity in diagnosis, which can adversely affect patient care and treatment outcomes. What was needed was an intelligent system that could predict diagnoses of breast cancer with rapidity and accuracy to provide decision support to medical professionals.

1.3 Objectives

1.3.1 Main Objective

To develop a full-stack AI application powered by machine learning to aid medical professionals in diagnosing breast cancer.

1.3.2 Specific Objectives

i. Analyse various breast cell nuclei measurements from a dataset and use them to train the ML model using logistic regression.
ii. Develop and integrate a Streamlit application (UI/Frontend) with the backend(ML model)
iii. Implement and test the breast cancer diagnosis predictor to automatically identify whether a breast tissue is either benign or malignant of cancer.
iv. Design a full-stack AI application for the breast cancer diagnosis predictor

1.4. Significance of the Study

This study is significant because it addressed one of the most critical challenges to healthcare: early-and-accurate diagnosis of cases of breast cancer. The project introduced the use of a machine-learning-powered tool in assisting diagnosis for improvement in patient outcomes through early detection and more informed decision-making.

Alignment with United Nations Sustainable Development Goal (UNSDGs)

SDG 3: Good Health and Well-Being

The application is important to ascertain healthy lives, ensure and promote well-being to all hence supporting SDG 3 through better diagnostic quality care, early treatment of diseases, detection and efficiency in the delivery of healthcare. This could also reduce mortality rates from cancer, hence leading to the best health outcomes.

SDG 5: Gender Equality

Breast cancer predominantly affects women and providing an advanced diagnostic tool that this application addresses, it promotes and advances gender equality in health. By improving early diagnosis and enhancing healthcare delivery for conditions that disproportionately affect women, this application contributes to SDG 5, by ensuring equal access to quality health care for all genders. Such a tool can help bridge gaps in healthcare for women and girls, especially in resource-poor settings where specialized diagnostics are limited.

SDG 9: Industry, Innovation and Infrastructure

The development of a diagnostic tool using machine learning techniques falls under SDG 9, since this endeavour fosters innovation within the healthcare sector. A representative example of this, show casing advanced technology for changing conventional diagnostic practices, the to be developed application is considered one more step toward modernizing infrastructure within healthcare. Using AI for better diagnostics in efficiency and quality will further ensure building resilient healthcare systems-innovative and responsive towards

1.5 Scope of the Study

The scope of the work involved the design, development and deployment of a machine learning-based Breast Cancer Diagnosis Predictor Application that would take specific diagnostic cell measurements as input and make predictions (probability) about the malignancy of the breast mass. The work focused on developing an easy-to-use interface by medical professionals for manual input, prediction and visualization. The scope of the project didn’t include the integration with external system such as cytology lab machines to automatically input data. In the future though, this may be considered in iterations of the application, but data input would be manual for this project.

1.6 Assumptions

The Breast Cancer Diagnosis Predictor Application worked under some crucial assumptions, forming the bedrock of the application's design, functionality and the expected outcomes. These assumptions made the application achieve its objectives and create realistic expectations toward the development and use of the application in clinical environments. They included the following:

Compliance with Data Privacy and Security Standards

The application assumed compliance with medical data privacy regulations, such as HIPAA or GDPR to ensure that all patient data entered into the system is handled securely. This is crucial for the application's acceptance and ethical use within clinical settings.

Model Stability and Predictive Consistency

It is assumed that the machine learning model developed continued to perform consistently over time and yield reliable predictions within the range of expected breast mass characteristics. If any of these distributions of input data change significantly (like new types of measurements or infrequent diagnostic patterns), the model may need retraining for predictive accuracy.

Continuous Model Improvement as New Data Becomes Available

It is hoped that over time, the model could be refined and re-trained with new data to enhance the diagnostic accuracy and relevance of diagnoses to modern medical findings. This continuous improvement process will allow the application to stay up-to-date with diagnostic advancements in breast cancer detection.

1.7 Limitations and Delimitations (Challenges and Counter Measures)

One of the main limitations of the study is the ***reliance on manual data input***, which can introduce human error and affect the accuracy of predictions. While the machine learning model exhibits an accuracy of 98%, any inaccuracies in the input data could lead to less reliable diagnoses. Future versions of the application could ***mitigate*** this limitation by ***integrating the system directly with laboratory equipment or machine for automatic data collection.***

Another limitation is the performance of the machine learning model across diverse patient populations. The ***initial training dataset*** may not have captured all variations in breast cancer diagnostics, leading to ***potential biases*** in the prediction model. To ***counter*** this challenge, ***continuous model improvement*** and retraining with diverse datasets will be necessary.

Lastly, the application was only meant to be a ***decision support tool*** and not a replacement for medical expertise. Medical professionals must interpret the results in the context of a comprehensive diagnostic process

CHAPTER 2- LITERATURE REVIEW

2.1 Introduction

At this point in time, breast cancer remains one of the most common types of cancer in the world; thus, it required a high level of diagnosis using advanced tools to assist in early detection and treatment planning. Machine learning and AI have shown tremendous potential in medical diagnostics, especially in the detection of cancer and has picked up considerable speed over the last decade. Machine learning algorithms formed a basis for developing predictive models to assist healthcare professionals in diagnostic decision-making. This development has facilitated the creation of tools that can process huge datasets to identify patterns and trends that may not be immediately apparent to human analysts. This chapter highlighted the current existing systems available for breast cancer prediction, their limitations and what improvements the proposed solution achieved.

2.2 Related Systems

Several systems have been developed to facilitate the early detection and diagnosis of breast cancer. Below are three prominent systems discussed in detail:

Breast Cancer Risk Assessment System Models

Breast cancer system risk assessment models such as the Gail Model and the Tyrer-Cuzick Model are statistical tools designed to estimate an individual's risk of developing breast cancer. They typically utilize data on genetic predispositions, lifestyle factors, and personal and family medical histories (Costantino et al., 1999).

The Gail Model for instance incorporates age, family history of breast cancer, reproductive history and previous biopsy results in order to produce a 5-year and lifetime percentage risk of cancer (Bondy et al., 2005). Likewise, the Tyrer-Cuzick Model further expands the aforementioned elements through the incorporation of information regarding hormonal variables, as well as genetic alterations such as BRCA1 and BRCA2 (Tyrer et al., 2004).

IBM Watson for Oncology

IBM Watson for Oncology uses AI algorithms to analyse medical literature and patient data, providing recommendations for cancer diagnosis and treatment. It considers patient-specific factors and aligns its suggestions with available clinical guidelines and best practices (Jiang et al., 2017).

Computer-Aided Diagnosis (CAD) Systems for Mammography

Computer-Aided Diagnosis (CAD) systems are designed to assist radiologists in the interpretation of mammographic images by highlighting areas that may indicate the presence of breast cancer. These systems use image recognition techniques to identify suspicious regions and highlight them for further analysis by medical professionals (Doi, 2007).

2.3 Limitations

Despite their advancements, the above systems have significant weaknesses that hinder their effectiveness in certain contexts:

Breast Cancer Risk Assessment System Models:

Generalization Issues

Most of the breast cancer risk assessment models like the Gail Model suffer from generalization problems on diverse populations since they are validated mostly on Caucasian datasets (Bondy et al., 2005).

Narrow Predictive Power

The risk models provide probabilities and do not diagnose a patient; therefore, diagnostic tests must be conducted to reach a conclusion (Costantino et al., 1999).

Genetic Risks Underestimation

Although the Tyrer-Cuzick Model incorporates genetic risk factors, it does not fully capture the complexities of polygenic risks or other unknown genetic factors (Tyrer et al., 2004).

Computer-Aided Diagnosis (CADx) Systems for Mammography:

High False Positive Rates

CADx systems are known to generate a high number of false positives, leading to unnecessary biopsies and patient anxiety (Kooi et al., 2017).

Limited to Imaging Data

CADx systems are primarily focused on mammographic imaging, not taking into consideration other diagnosis parameters such as cell feature characteristics from biopsy results or even the history of the patient; this is important for full diagnosis (Doi 2007; Lehman et al. 2015).

Effectiveness depends on radiologists.

While CADx systems offer value in terms of insights, their effectiveness can vary from case to case and is most pronounced when used as an assistive tool with the expertise of radiologists. (Taylor et al. 2019).

IBM Watson for Oncology:

Complexity and Cost

The high computational power and related infrastructure costs required for IBM Watson for Oncology make it an unsuitable approach for small healthcare settings found in low- and middle-income countries (Jiang et al., 2017).

Limited Adaptation to Local Guidelines

While the recommendations of IBM Watson will be based on global best practices, it does not adapt the recommendations to the local healthcare system and therefore, the constraints and often reduces its applicability (Lee et al., 2018).

Data Privacy Concerns

The fact that there is a lot of sensitive medical data being manipulated increases ethical and regulatory complexities, especially in jurisdictions that have tight data protection legislation (Amann et al., 2020).

2.4 How the proposed solution handled these weaknesses

The Breast Cancer Diagnosis Predictor Application addressed the identified limitations of the existing systems as follows:

Improved Generalizability

The application employed the publicly available Breast Cancer Wisconsin (Diagnostic) Dataset from Kaggle (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data) that included varied features derived from FNA procedures offering better representation of characteristics related to breast cancer. In addition, machine learning allowed the model to be retrained or with the regional population dataset to solve generalization problems across different demographics.

Customizable for Local Contexts

While IBM Watson is bound to global standards, the modular architecture of the predictor application allowed customization based on region-specific datasets or a new ML model and design architecture in the future, hence being more relevant to the local healthcare systems making it a better fit for diverse healthcare environments.

Direct Diagnostic Classification

Unlike the traditional risk assessment models, the application made use of a trained logistic regression model to directly classify cell clusters as either benign or malignant. This diagnostic capability eliminated the reliance solely on probabilistic risk scores and offers actionable insights, reducing the need for extensive confirmatory testing.

Feature-Based Complementary Insights

While the system did not explicitly incorporate genetic risks, its focus on FNA-derived features such as texture, concavity, symmetry and etcetera provided very useful predictive insights that complemented genetic risk factors. The modular design allowed for the integration of genetic data in the future if datasets with such features become available.

Optimized Predictions

The logistic regression model, trained and tested on well-curated data reduced false positives by optimizing hyperparameters and performance metrics during the training process. Besides this, the predictions were given with their probability scores, which allowed the stakeholders to get an idea about the confidence level of the diagnosis and avoid unnecessary alarms.

Multimodal Diagnosis

The predictor application used non-imaging FNA data as from the Kaggle dataset, focusing on critical cellular characteristics such as concavity and compactness. This made it a complementary tool to imaging-focused CADx systems, broadening the diagnostic scope by incorporating biopsy-derived parameters and user-input clinical data.

Standalone Diagnostic Accessibility

Unlike CADx systems which rely heavily on radiologists for interpretation, the predictor application was designed as a stand-alone diagnostic aid accessible to any medical cancer professional without extensive expertise in mammographic analysis. A streamlined interface and interpretative output enabled users with minimal technical training to make informed decisions.

Cost-Effective Framework Deployment

The application leveraged lightweight frameworks, Streamlit and Python, which are computationally efficient and low-cost. The tech stack was designed to be deployable on consumer-grade hardware to make it accessible for small clinics and hospitals in low- and middle-income regions.

Secure On-Premise Data Handling

Data privacy was ensured by offering secure local data storage and optional anonymization of patient inputs. The app was designed for on-premise deployment, sensitive patient data would remain with the healthcare institution reducing the probability of data breaches and full compliance with privacy regulations.

User-Friendly Interface

The application included a well-designed user interface to address usability enabling medical practitioners to input diagnostic data into the system manually, predict and interpret the results through a visual aid that is a radar chart. The inclusion of a radar chart clearly visualized diagnostic data and helped medical professionals to interpret the features influencing the prediction-a requirement that is often lacking in most existing current systems

CHAPTER 3- METHODOLOGY

3.1 Introduction

This chapter outlined the development methodology that was followed for the Breast Cancer Diagnosis Predictor Application. It gave the actual roadmap on how project objectives was achieved; right from the requirements gathering to the final deployment of the project. The project was developed using the Software Development Life Cycle (SDLC) methodology to make sure that each phase of the development process followed a structured and systematic approach.

Python was used as the backend principal development language for machine learning and data processing. Development of the UI, that is the frontend including an interactive web-based dashboard was done using Streamlit and CSS, which enabled medical professionals to input diagnostic data and get predictions. Additional data visualization libraries such as Plotly was used in making an informative and interactive radar chart. The machine learning model used, logistic regression, was trained using the Breast Cancer Wisconsin Diagnostic Dataset from Kaggle (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data) and served as the basis for the predictive model to classify whether a cell cluster is either benign or malignant.

For deployment and hosting, the project made use of Streamlit Cloud which provided easy deployment and maintenance of the application to keep it accessible via any web browser. The same platform also supported compatibility so that the application would be accessible across a range of devices without requiring any local installations.

Along with the development tool and framework, this chapter also designated the procedure for designing the system architecture, defining the system requirements, both functional and non-functional, and establishing data collection and analysis methods. The project schedule and budget were also provided to make sure of effective planning and resource allocation throughout the project lifecycle.

3.2 Project Design (Waterfall Method)

The project followed the **Waterfall Model** of the Software Development Life Cycle (SDLC) methodology. The Waterfall model is a linear and sequential approach to software development, where each phase of the project flows downward, much like a waterfall, from one stage to the next. The project only moved to the next phase once the current one is completed ensuring that no important steps are overlooked.

Justification for the Waterfall Model:

Simplicity: The rigid structure of the waterfall model was easy to understand and implement; hence, it was easy to manage and thus suitable for this project.

Well-defined Milestones: Each project phase had clear objectives and deliverables, ensuring clarity in the progress tracking.

Focus on Documentation: The model emphasized documentation at each stage, something that is essential for this project.

Minimized Risk: Testing fell in a different phase after development, thus minimizing unexpected problems at deployment.

The **Waterfall Model** guaranteed that the project was done systematically in a timely manner to ensure the creation of a high-quality, user-friendly, functional Breast Cancer Diagnosis Predictor Application.

3.3 Design Procedures

The design procedures of the ***Breast Cancer Diagnosis Predictor Application*** was structured to align with the Waterfall SDLC methodology model. Each step of the process was elaborated on in this section to ensure clarity and alignment with the project objectives and deliverables. The steps include the following:

1. Requirements Gathering and Analysis:

This phase focused on understanding the needs of the stakeholders including the technical requirements needed for the application. The stakeholders include ***patients***, ***certified medical professionals*** (radiologists, oncologists or lab technicians), ***hospitals and clinics*** and ***myself (as the developer).*** The primary objective was to develop a user-friendly application capable of:

Diagnosis Prediction: It classified cell clusters as either benign or malignant using a logistic regression model.

Visualization: It visualized predictions and insights through interactive visualization using a radar chart.

User Interaction: A smooth and usable interface for medical professionals to input diagnostic data.

Technical Requirements

The application was ***compatible*** with the latest web browsers, make fast predictions and ***responsive on all devices***. ***Performance indicators*** such as ***low latency*** in predictions and ***seamless integration*** of machine learning and visualization components was ***prioritized.***

2. System Architecture:

The application architecture was designed in a ***three-layered*** structure:

Frontend: the frontend was developed using Streamlit and a bit of CSS providing an interactive dashboard where users input diagnostic data and viewed predictions along with their probability scores.

Backend: A Python-based machine learning model (logistic regression) was implemented using machine learning libraries like Scikit-learn for predictive analysis etcetera.

Database: Since the application used a pre-existing dataset (Breast Cancer Wisconsin Diagnostic Data from Kaggle (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data)), there wasn't need of a traditional database. The model directly loaded and processed the dataset. The data flow involved receiving inputs from the user interface, passing them through the trained model and displaying predictions and related visualizations.

Visualization: The application utilized Plotly to generate an interactive radar chart to enhance user comprehension of predictions.

3. Detailed Design:

The system was designed in a manner to maximize user experience and computational efficiency;

User Flow: The user's journey from inputting diagnostic data to presenting results was mapped. Each step in the interaction was intuitive thus ensuring that users can understand and trust the predictions provided.

Integration: The backend that is the machine learning model, interacted with the frontend to produce results in real time. The architecture was modularly integrated to support updates and enhancements.

4. User Interface Design:

The UI was designed with simplicity and functionality in mind as follows:

Input Fields: The user input diagnostic metrics, such as radius, texture, perimeter, etcetera., via sliding bars which will be created for every cell feature characteristic.

Interactive Dashboard: The dashboard presented the predictions, the probability score and visualizations in a clear, orderly fashion.

Charts: Interactive visualizations via a radar chart using Plotly was used to better understand the model predictions thus help in decision making.

The UI was also tested for usability and responsiveness across different devices and browsers.

5. Data Design:

The project used the Breast Cancer Wisconsin Diagnostic Data Set, from Kaggle (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data);

Data Format: The dataset was pre-processed for handling missing values, data normalization and preparation of features for the logistic regression model (EDA).

Data Flow: User inputs were passed from the UI to the backend for the trained logistic regression model to process the input data and predict whether a cell cluster is benign or malignant. Results including the probability score and visualizations was displayed on the dashboard.

6. Implementation:

Backend Development

Trained the logistic regression model using Scikit-learn and preprocessed the dataset using Pandas and Numpy and developed a Python script to load the model and return predictions based on user inputs.

Used Pickle machine learning library to export the model to the frontend to prevent re-training of the model thus reducing latency and improving performance of the application.

Frontend Development

Bid on an interactive Streamlit application that captured user inputs and displayed model predictions. Employed Plotly for creating the radar chart for visualization of the results.

Integration of all the components above gave us the full-stack AI application.

7. Testing and Validation:

Full Testing (System Testing) made sure the application was seamlessly functional:

Functionality Testing

The logistic regression model's accuracy and classification report including the precision call etcetera was validated by splitting the dataset into training and testing subsets, to test the correctness of predictions from the Logistic Regression model, ensuring it performed well on unseen data

Unit Testing

The system's individual components such as the machine learning model, user input forms, probability score and visualizations were tested.

Integration Testing

Frontend and backend worked in cohesion, and Streamlit allowed seamless integration of CSS, Plotly, and the machine learning model.

System Testing

Tested the system application as a whole.

User Acceptance Testing (UAT)

After the system is deployed, it underwent UAT to ensure that it met user expectations and worked as intended.

8. Deployment

The application was deployed on Streamlit Cloud, ensuring; ***public accessibility*** with no need for local installations, ***scalability*** to support multiple users and ***ease in maintenance*** in case of future updates and enhancements. Streamlit Cloud offered a free deployment solution that is efficient and hosted support to share Streamlit applications.

9. Maintenance

After deployment, the system entered the phase of maintenance where:

Bug Fixes: Any bugs detected post-deployment were fixed.

Updates: Integrated users' feedback to enhance the working and usability of the application.

Enhancements: Visualizations was enhanced; the predictive capabilities was extended an new features added to the application based on changes in medical knowledge or users' requirements.

This structured design process ensured that the Breast Cancer Diagnosis Predictor Application was built efficiently, was user-friendly and achieved its objectives of assisting medical professionals in diagnosing breast cancer.

3.4 System Requirements

Functional Requirements:

Data Input

The user (medical professional) was able to enter the values for different features of the cell cluster, like radius, texture, perimeter and etcetera.

Prediction Output

The system predicted along with the probability score whether a cell cluster is benign or malignant scores based on the input data fed into it.

Visualization

The system visualized the model's predictions through an interactive radar chart.

Model Accuracy

The machine learning model had a high accuracy of 98 % in predicting the classification of breast cancer.

Non-Functional Requirements:

Usability

The web application was user-friendly, ensuring that users with minimal technical knowledge can interact with it effectively.

Performance

The system was responsive, had little to no delay in the display of the UI components including the dashboard, predictions and the radar visualization.

Security

The application securely handled any sensitive user data; in this case, no personally identifiable information is used.

Scalability

The application was able to handle large volume of user data and multiple concurrent live users.

Compatibility

The system was able to function on multiple OS platforms and different device architectures for user accessibility.

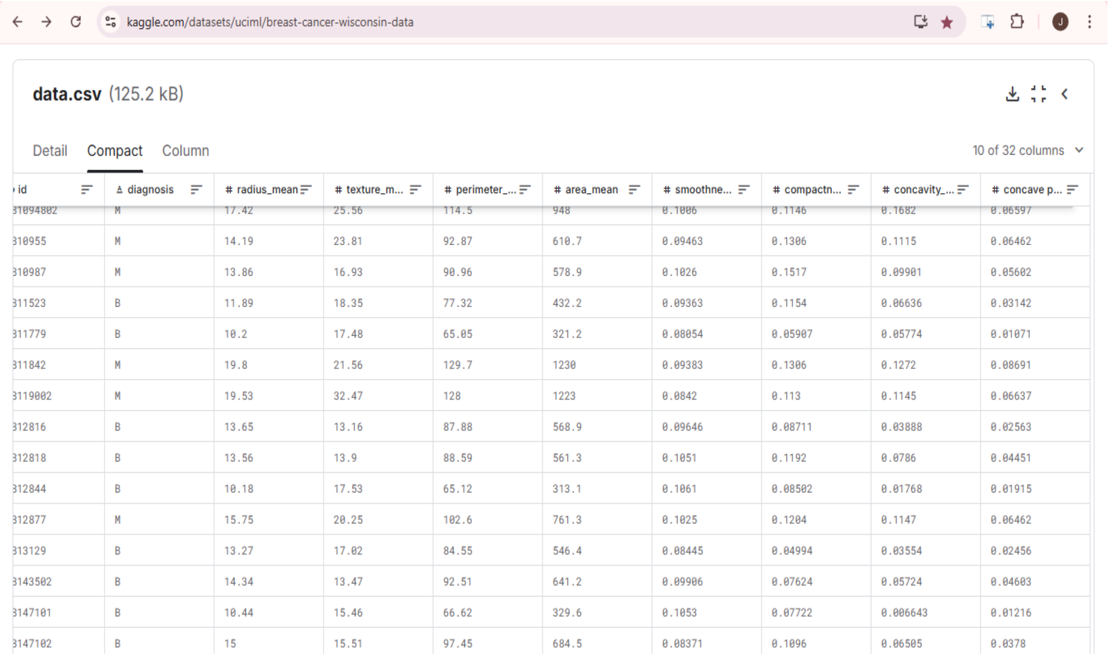
Reliability

The system was reliable and readily available to users with minimal downtime and disruptions

3.5 Data Collection and Analysis (needs assessment) methods and tools

The dataset used for the project was the Breast Cancer Wisconsin (Diagnostic) Dataset which is freely available on Kaggle (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data). The dataset originates from the University of Wisconsin Hospitals, Madison compiled by Dr. William H. Wolberg (Wolberg, 1995). It has been quite popular in machine learning research for breast cancer prediction because of its well-organized structure and good quality features.

Figure 1-Breast Cancer Wisconsin Dataset



data.csv (125.2 kB)

id	diagnosis	radius_mean	texture_m...	perimeter_...	area_mean	smoothne...	compactn...	concavity_...	concave p...
31094802	M	17.42	25.56	114.5	948	0.1006	0.1146	0.1682	0.06597
310955	M	14.19	23.81	92.87	610.7	0.09463	0.1306	0.1115	0.06462
310987	M	13.86	16.93	90.96	578.9	0.1026	0.1517	0.09901	0.05602
311523	B	11.89	18.35	77.32	432.2	0.09363	0.1154	0.06636	0.03142
311779	B	10.2	17.48	65.05	321.2	0.08054	0.05907	0.05774	0.01071
311842	M	19.8	21.56	129.7	1230	0.09383	0.1306	0.1272	0.08691
3119002	M	19.53	32.47	128	1223	0.0842	0.113	0.1145	0.06637
312816	B	13.65	13.16	87.88	568.9	0.09646	0.08711	0.03888	0.02563
312818	B	13.56	13.9	88.59	561.3	0.1051	0.1192	0.0786	0.04451
312844	B	10.18	17.53	65.12	313.1	0.1061	0.08502	0.01768	0.01915
312877	M	15.75	20.25	102.6	761.3	0.1025	0.1204	0.1147	0.06462
313129	B	13.27	17.02	84.55	546.4	0.08445	0.04994	0.03554	0.02456
3143502	B	14.34	13.47	92.51	641.2	0.09906	0.07624	0.05724	0.04603
3147101	B	10.44	15.46	66.62	329.6	0.1053	0.07722	0.006643	0.01216
3147102	B	15	15.51	97.45	684.5	0.08371	0.1096	0.06505	0.0378

Dataset Characteristics

The dataset contained 569 rows and 32 columns. Each row represents a single instance of a diagnosis for a cluster of breast cells while the columns represent the features of that diagnosis.The columns include the following:

ID Column:

- ***`id`:*** A unique identifier for each sample.

Target Variable:

The dataset contains a ***diagnosis*** column which is the target variable that classifies each sample as either:

- ***M*** (Malignant): Representing malignant or cancerous cluster.
- ***B*** (Benign): Representing non-cancerous or benign cluster.

Feature Variables:

There are 30 numeric features; every single feature in the list originally generated from the fine-needle aspirate (FNA) of a tumour in a breast mass. They fall into three fundamental classes:

Mean Values**:** Measurements averaged across the sample.

Standard Errors**:** Variability of the sample measurements.

Worst Values: Most extreme measurements within the sample.

The columns describe specific characteristics for each of the 3 classes above of cell nuclei, derived from the FNA images:

Radius: The average distance from the center to points on the perimeter.

Texture: The standard deviation of gray-scale values.

Perimeter: The circumference of the cell nucleus.

Area: The total size of the cell nucleus.

Smoothness: The variation in the radius lengths.

Compactness: The ratio of the perimeter squared to the area (indicating circularity).

Concavity: The severity of concave portions of the cell nucleus.

Concave Points: The number of concave portions.

Symmetry: The degree of symmetry in the nucleus shape.

Fractal Dimension: A measure of complexity, describing changes in detail at different scales.

Each of the above 10 features is described as the mean, standard error and worst value thus forming 30 feature columns.

Dataset Properties

The dataset was well-balanced, with a slight predominance of benign cases, with 357 being benign and 212 malignant.

All the feature values were numeric making it an apt dataset for machine learning use.

There are no missing or mismatching values in the dataset reducing the time and complexity of preprocessing.

Importance of the Dataset (Why this Dataset)

The Breast Cancer Wisconsin (Diagnostic) Dataset was highly suitable for the project due to its:

Real-world Relevance: Since it is from medical research, it reflected real clinical data.

Completeness: The high-dimensional feature set enabled the fine-grained characterization of cell clusters and hence it improved the predictive capability of the machine learning model.

Quality: There were no missing or mismatching data points hence allowing a sound and reliable model training.

Data Analysis:

Preprocessing: The dataset underwent necessary preprocessing steps such as normalization, feature selection etcetera.

EDA: Used visualization and statistical analysis with the ML libraries; Pandas, Numpy etcetera in Python to understand the distribution of the data and identify patterns.

Model Training: Data was split into both training and testing sets. In this case, the Logistic Regression model was trained using Scikit-learn.

Model Evaluation: The model's performance was evaluated using metrics such as accuracy, precision, recall, and F1-score.

Tools To Be Used:

Python: The primary language for the backend development and programming including model training and analysis.

Streamlit: Streamlit was employed for building up a web interactive dashboard UI for the app.

Scikit-learn: Used to train the logistic regression of the machine learning model.

Plotly: For an interactive visualizations of data input and predictions.

Pandas: For data Manipulation and pre-processing.

Numpy: For numerical computing and array operations which enhanced data preprocessing and statistical computations.

Pickle: Employed to serialize and save the trained machine learning model for efficient performance, deployment and future use.

Altair: Created declarative and simple statistical visualization to complement Plotly's interactive visualizations.

The above tools collectively ensured a robust, efficient and visually appealing Breast Cancer Diagnosis Predictor application that integrates machine learning, data visualization and interactive user experiences.

CHAPTER 4: SYSTEM ANALYSIS

4.1 Detailed Analysis of the Current System

The evaluation of the existing system focused on determining the status of breast cancer diagnostic equipment today and the deficiencies that the Breast Cancer Diagnosis Predictor Application aimed to address.

Traditional diagnostic methods such as Gail and the Tyrer-Cuzick models heavily relied on clinical evaluation, biopsy studies and radiological imaging modalities such as mammography, ultrasound, and MRI scanning. While these methods were good, they had limitations in terms of *accuracy of early detection*, *human error* and *diagnostic subjectivity*. Manual histopathological examination was labour intensive and prone to misclassification of malignant and benign tumours and often leading to additional tests and increased levels of patient anxiety. While Computer-Aided Diagnosis (CAD) technology existed, it was primarily on medical images rather than tabular data, so it was *not present* in resource-poor settings. Legacy CAD systems, also, had no sophisticated predictive analytics and interactive visualization capabilities limiting their functionality for enhancing decision making among healthcare specialists.

Limitations of the Existing System

Diagnostic Errors

Radiological interpretations led to false positives and false negatives, impacting therapy.

Lack of Interactivity

Most systems were devoid of interactive data visualization, which rendered it difficult for healthcare professionals to navigate predictions.

Limited Automation

The necessity for human image assessment resulted in longer diagnosis times.

Scalability Problems

The traditional systems were computationally demanding and not able to process big data effectively.

Non-Personalized Predictions

The current systems failed to provide personalized risk estimates based on the patient data.

The Proposed System

With a view to counteracting the imperfections of present diagnostic methods, the Breast Cancer Diagnosis Predictor Application was developed. This system integrated Machine Learning (ML) and Artificial Intelligence (AI) to provide accurate automated classification of breast cancer tumours.

The proposed application:

Employed logistic regression for tumour classification as benign or malignant based on diagnostic features.

Built an interactive Streamlit dashboard through which medical professionals can input patient data and receive real-time forecasts.

Included interactive visualizations using Plotly to present the correlation between predictions and tumour characteristics.

Ensured scalability and accessibility by deploying on Streamlit Cloud for web-based access.

System Modelling and Flow Representation

Flowchart

The flowchart depicted the sequential flow of the process of the Breast Cancer Diagnosis Predictor Application. The flowchart visually representation of how user input is mapped, how the machine learning model generates prediction and the manner in which results are output to the user (medical professional)

Flowchart Process:

The flowchart comprises the following principal steps:

Start – The process begins whenever the user launches the web application.

User Inputs– The user (medical professional) enters the required cell cluster features (e.g., radius, texture, perimeter, smoothness, etc.) into the Streamlit web interface.

Validate Input Data – The system checks whether all required fields are filled correctly.

If Invalid, an error message is displayed, and the user is requested to re-enter the values.

If Valid, the input data is passed to the next stage.

Preprocess Data – Entered data is transformed and normalized if required for the logistic regression model.

Load Trained Model – The trained logistic regression model from a serialized file (.pkl file) is loaded by the system.

Prediction – Machine learning model processes the input features and identifies whether the cell cluster is Benign (Non-Cancerous) or Malignant (Cancerous).

Probability Score – The computer generates a confidence score (i.e., 85% chance of malignancy).

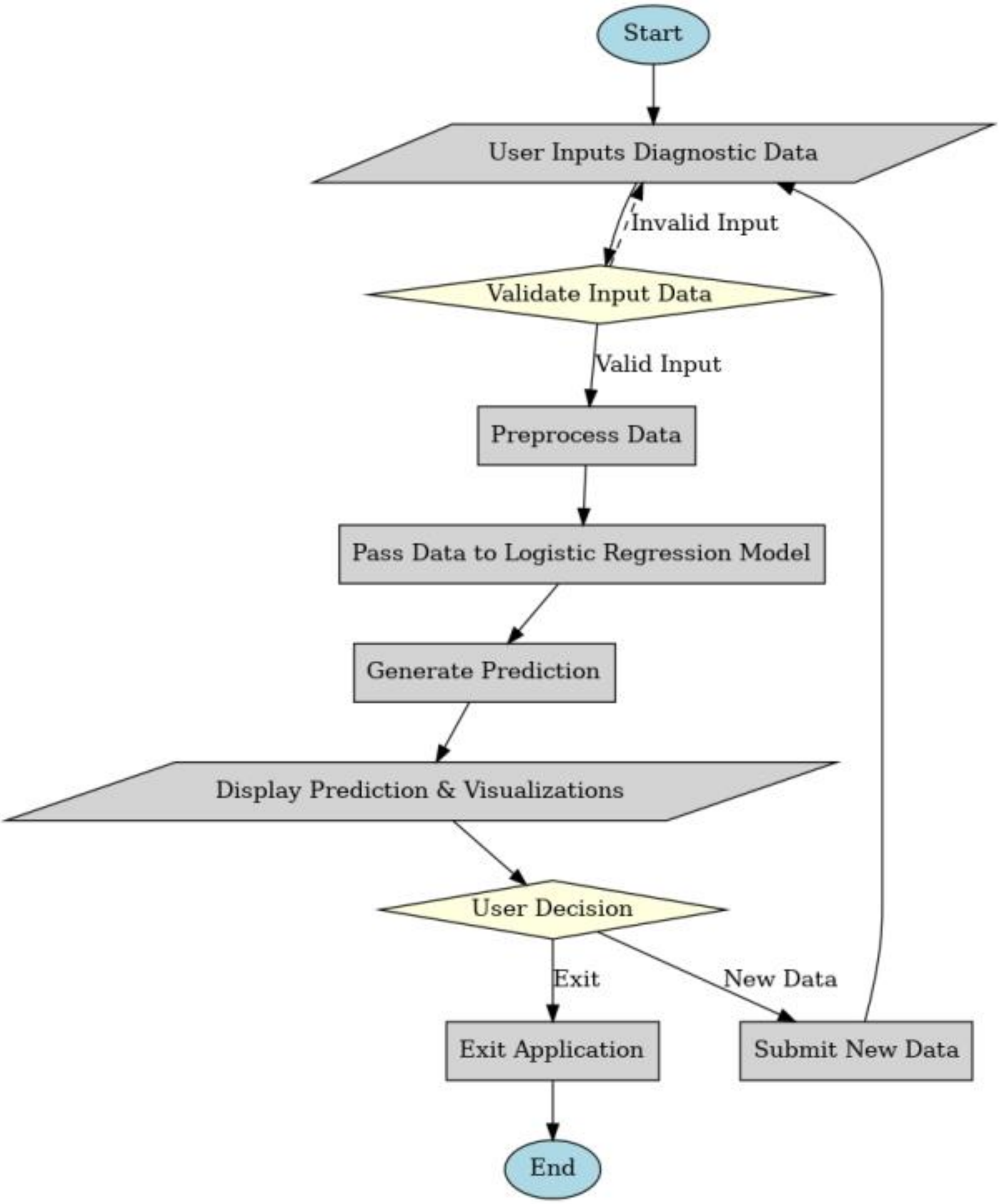
Visualization – The prediction output and probability score are presented together with interactive visualizations using Plotly; a *radar chart* is used to display feature importance comparisons.

Prediction Output – The user is presented with the final diagnosis (Benign or Malignant) together with visualizations.

User Decision – The doctor processes the prediction and decides what to do next (further clinical evaluation, biopsy, treatment recommendation, etc.).

End – The process is finished, and the user may resume the forecast or exit the application.

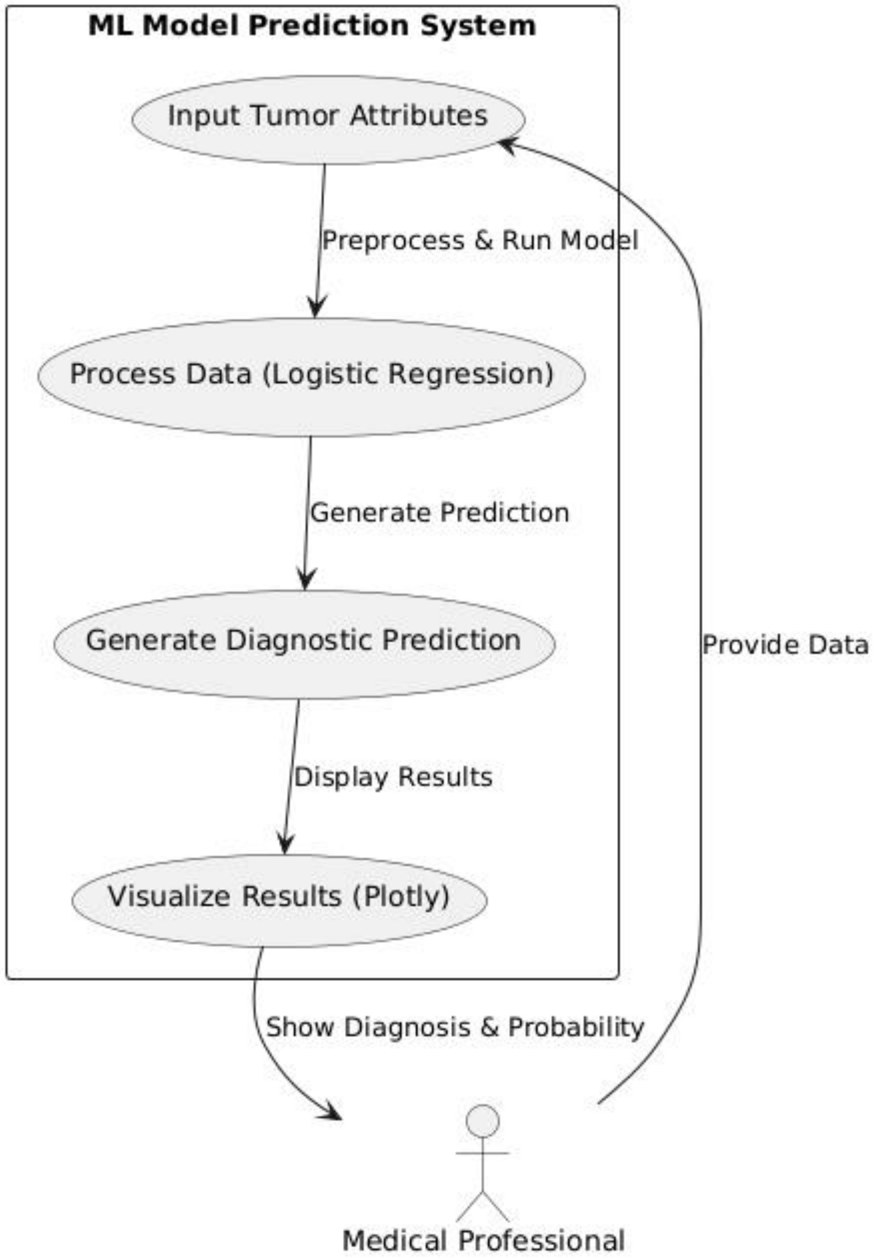
Figure 2- System Flowchart



Context Diagram

A context diagram was drawn to illustrate the high-level interaction of the user (medical professional) with the ML Model Prediction System. It indicated how the user inputs tumour attributes (e.g., radius, texture, and perimeter), which are run through a trained logistic regression model. The system renders a diagnostic prediction—benign or malignant with a probability score—and visualizes the results using Plotly.

Figure 3- Context Diagram

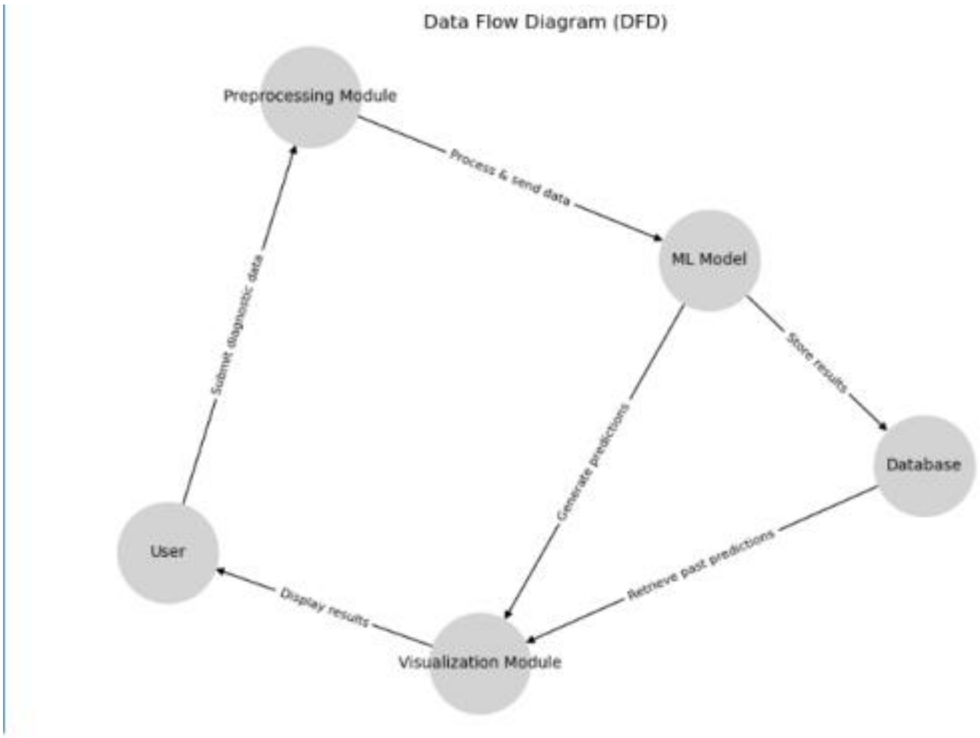

ML Model Prediction System
Input Tumor Attributes
Preprocess & Run Model
Process Data (Logistic Regression)
Generate Prediction
Generate Diagnostic Prediction
Display Results
Visualize Results (Plotly)
Provide Data
Show Diagnosis & Probability
Medical Professional

Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) indicated the major components and interactions of the system.

The DFD displayed the high-level data flow, with the user entering tumour feature values, the system converting the input, and the trained logistic regression model generating a classification output (Benign or Malignant). The system then plotted the results in Plotly and presented them to the user through the interactive Streamlit interface.

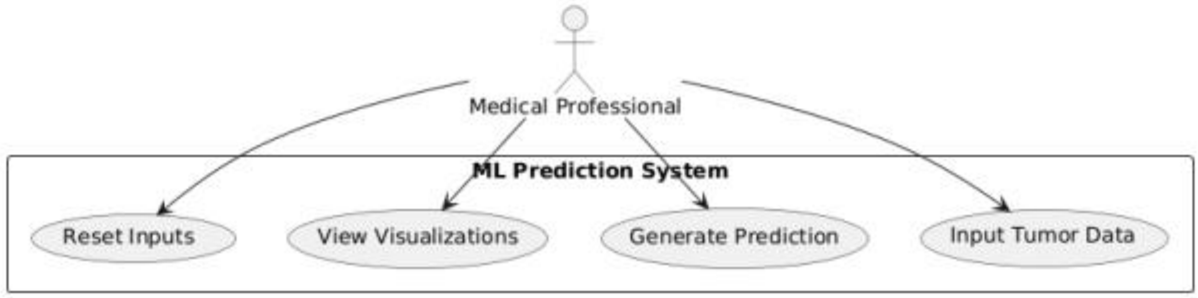
Figure 4-DFD



Use Case Diagram

The *Use Case Diagram* identified principal system actors: the medical professional (user) who provided input and verified predictions, and the system, which handled information and displayed results. Key use cases were *Input Tumor Data, Generate Prediction, View Visualizations and Reset Inputs.*

Figure 5-Use Case Diagram



4.2 System Requirements

4.2.1 Functional Requirements:

Data Input

The user (medical professional) was able to enter values for different features of the cell cluster such as radius, texture and perimeter.

Prediction Output

The system predicted whether a cell cluster was benign or malignant, along with a probability score based on the input data provided.

Visualization

The system visualized the model's predictions through an interactive radar chart.

Model Accuracy

The machine learning model achieved a high accuracy of 98% in classifying breast cancer.

4.2.2 Non-Functional Requirements:

Usability

The web application was designed to be user-friendly, ensuring that users with minimal technical knowledge could interact with it effectively.

Performance

The system was responsive, with little to no delay in displaying UI components, including the dashboard, predictions and radar visualizations.

Security

The application securely handled any sensitive user data; however, no personally identifiable information was used.

Scalability

The application was capable of handling a large volume of user data and multiple concurrent live users.

Compatibility

The system functioned on multiple OS platforms and different device architectures for user accessibility.

Reliability

The system was reliable and readily available to users, with minimal downtime and disruptions.

CHAPTER 5: SYSTEM DESIGN

This chapter presents the system components and design environment for the Breast Cancer Diagnosis Predictor Application. It includes the system architectural design, database design, and user interface design. The system was designed taking into consideration efficiency, usability, scalability, and maintainability. Structured tools, including architectural diagrams, data flow diagrams (DFD), Unified Modelling Language (UML) diagrams and Entity-Relationship Diagrams (ERD) guided the design to visualize and illustrate system components and interactions.

5.1 Architectural Design

The Breast Cancer Diagnosis Predictor Application was based on a three-tier architecture that divided the system into the frontend (presentation layer), backend (processing layer), and model layer (data analysis layer) for modularity, maintainability, and performance improvement. The design architecture outlined the system's structure, including its major constituents and how they interacted, for efficient processing of data and seamless user experience. The application adapted the following **three-tier architecture** as described below:

1. **Frontend (Presentation/User Interface Layer)**

   Developed using Streamlit, allowing users to enter diagnostic features, view predictions, and interact with visualizations.

   Integrated Plotly and Seaborn for dynamic and interactive data visualization.

   Designed to be responsive and compatible with multiple devices.

2. **Backend (Application/Processing Layer)**

Implemented using Python to manage data processing and handle user inputs.

Used Scikit-learn for implementing the logistic regression model.

Integrated NumPy and Pandas for data preprocessing and manipulation.

3. **Model Layer (Data Layer)**

Trained a logistic regression model using the Breast Cancer Wisconsin (Diagnostic) Dataset.

Pickled the trained model using Pickle for easy deployment and real-time predictions for performance.

Provided classification results with probability scores to enhance interpretability.

System Components:

User Interface (UI): Handled user input and input gathering.

Prediction Engine/ML Model: Applied the trained logistic regression model to the input data.

Visualization Module: Generated graphs and interactive plots through Plotly.

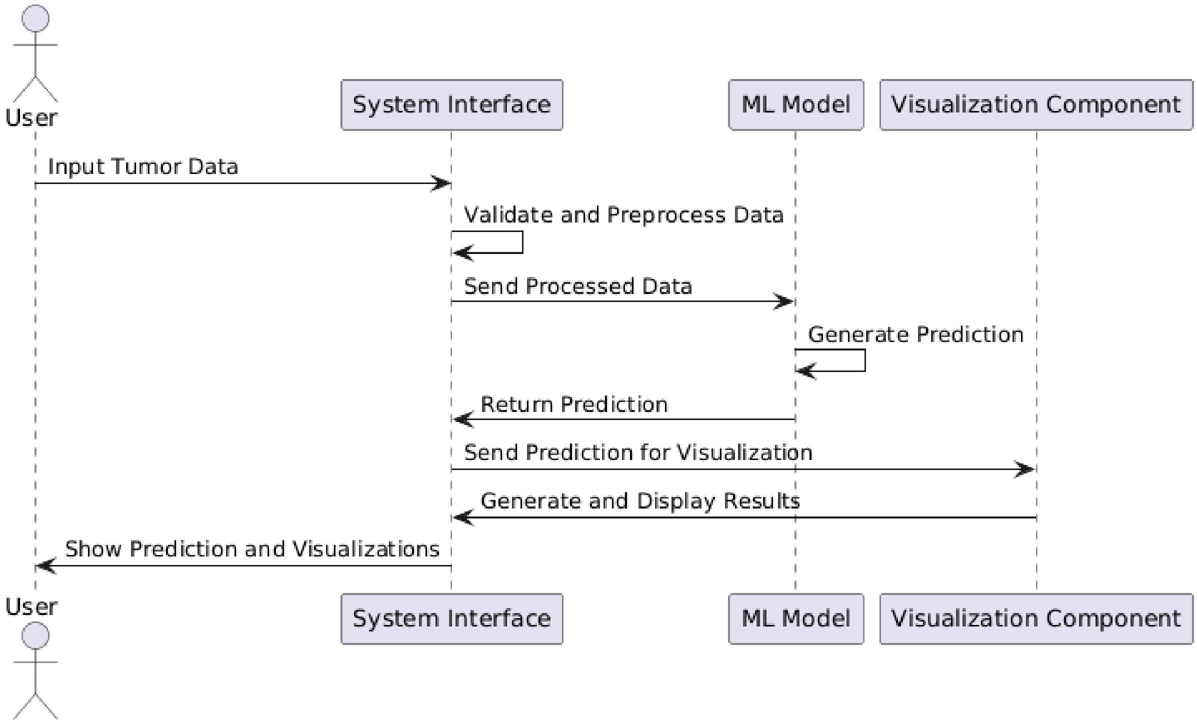
Data Management Module: Employed Pandas and Pickle for dataset handling and model persistence.

System Modelling and Flow Representation

Sequence Diagram:

The Sequence Diagram mapped the interaction flow between the User, System Interface, ML Model, and Visualization Component, defining the order of activities from user input to prediction output.

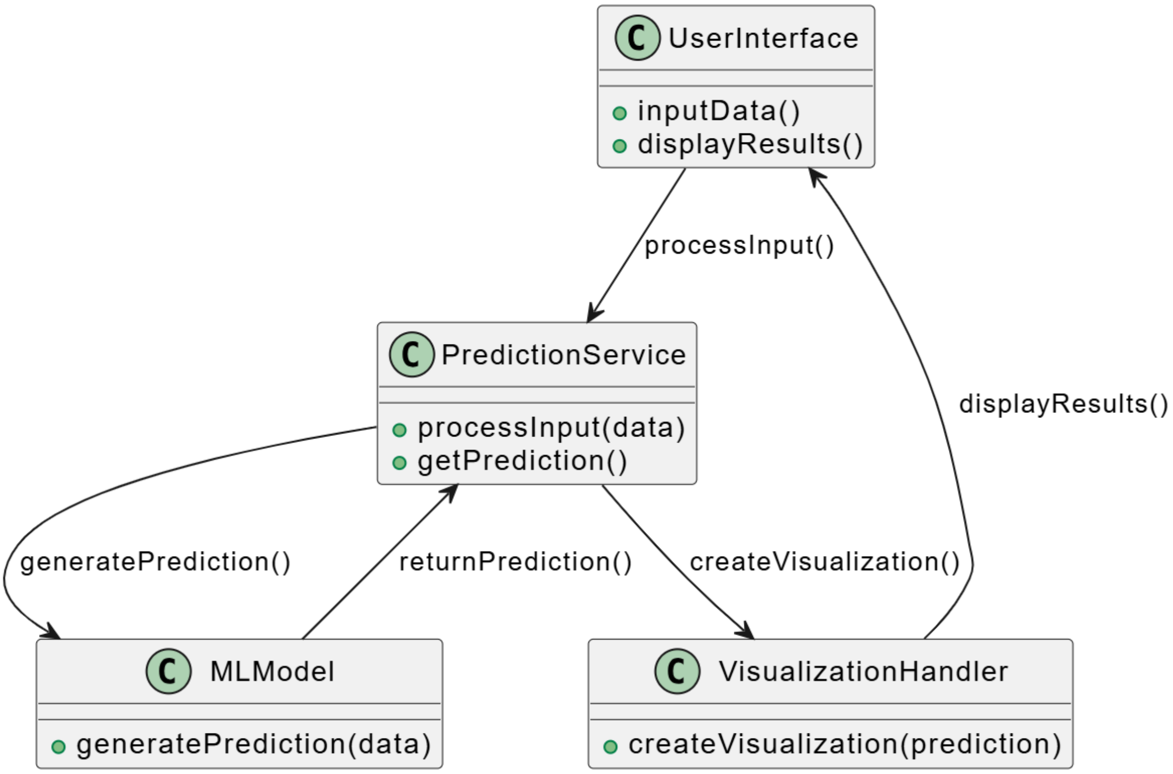
Figure 6-Sequence Diagram



Class Diagram:

The Class Diagram defined the system's key components, including UserInterface, MLModel, PredictionService and VisualizationHandler, representing the interactions and roles of each module in processing input data, generating predictions, and presenting results.

Figure 7- Class Diagram



5.2 Database Design

Since the system utilized the pre-existing Breast Cancer Wisconsin Diagnostic Dataset (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data) instead of a typical relational database, the design of the data flow and storage mechanism was crucial. The dataset is originates from the University of Wisconsin Hospitals, Madison compiled by Dr.William H.Wolberg (Wolberg, 1995) and freely available on Kaggle.

It contained 569 records and 32 columns consisting of diagnostic attributes mean, radius, texture, perimeter, area, compactness, concavity, concave points, symmetry and fractal dimension. There are 30 numeric features; every single feature in the list originally generated from the fine-needle aspirate (FNA) of a tumour in a breast mass. They fall into three fundamental classes: mean values, standard errors and worst values. Although the system did not implement a relational database management system (RDBMS), data retrieval and storage were structured appropriately with CSV-based storage and Pickle serialization of the Wisconsin dataset.

Data Storage and Handling

Dataset Storage

The **Breast Cancer Wisconsin (Diagnostic) Dataset** was stored as a CSV file and loaded into the system using Pandas.

Data Preprocessing including EDA and feature engineering was performed on relevant attributes to feed into the machine learning model.

Model Training

The trained logistic regression model was serialized and stored using Pickle, enabling quick loading and real-time predictions enhancing performance

Data Flow and Processing

The system accepted user inputs (diagnostic features) via the frontend.

The backend processed the inputs, normalized them and fed them into the model.

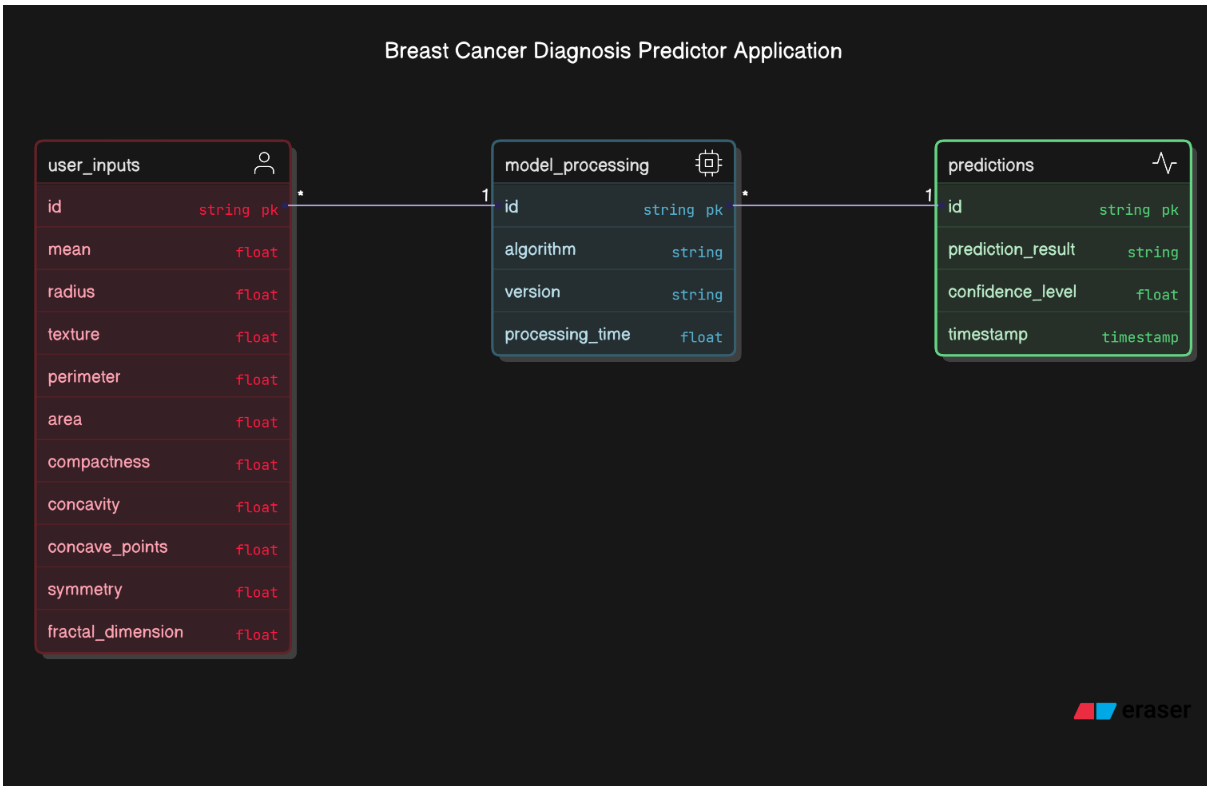
The trained model returned a prediction (benign or malignant) along with a probability score.

The results were then visualized using interactive charts.

Entity-Relationship Diagram (ERD)

Although the system did not utilize a conventional database, an **ERD** was generated to illustrate the key entities and relationships between **user inputs, model processing, and predictions.**

Figure 8- ERD



Breast Cancer Diagnosis Predictor Application

user_inputs	
id	string pk
mean	float
radius	float
texture	float
perimeter	float
area	float
compactness	float
concavity	float
concave_points	float
symmetry	float
fractal_dimension	float

model_processing	
id	string pk
algorithm	string
version	string
processing_time	float

predictions	
id	string pk
prediction_result	string
confidence_level	float
timestamp	timestamp

5.3 User Interface (UI) Design

The user interface was designed to be intuitive, user-friendly and visually appealing with a streamlined dashboard that enabled medical professionals to input diagnostic features and receive real-time predictions. Built using Streamlit, the UI provided a clean, interactive and seamless experience for efficient data entry and result visualization.

UI Design Principles:

Minimalist Design: Clean and simple layout with only essential input fields, visualization and predictions

Interactivity: Interactive radar chart developed using Plotly for better insights.

Responsiveness: Adaptive design for accessibility across different screen sizes and devices.

*Real-Time Feedback***:** Instant model predictions and visualizations with efficient performance.

UI Components

1. **Sidebar Section**

   This is the input panel where medical professionals can enter cell measurements collected using sliders for each cell diagnostic attribute.

2. **Home Section**

   Provided an introduction to the application, its purpose and instructions on how to enter diagnostic data.

3. **Visualization Section**

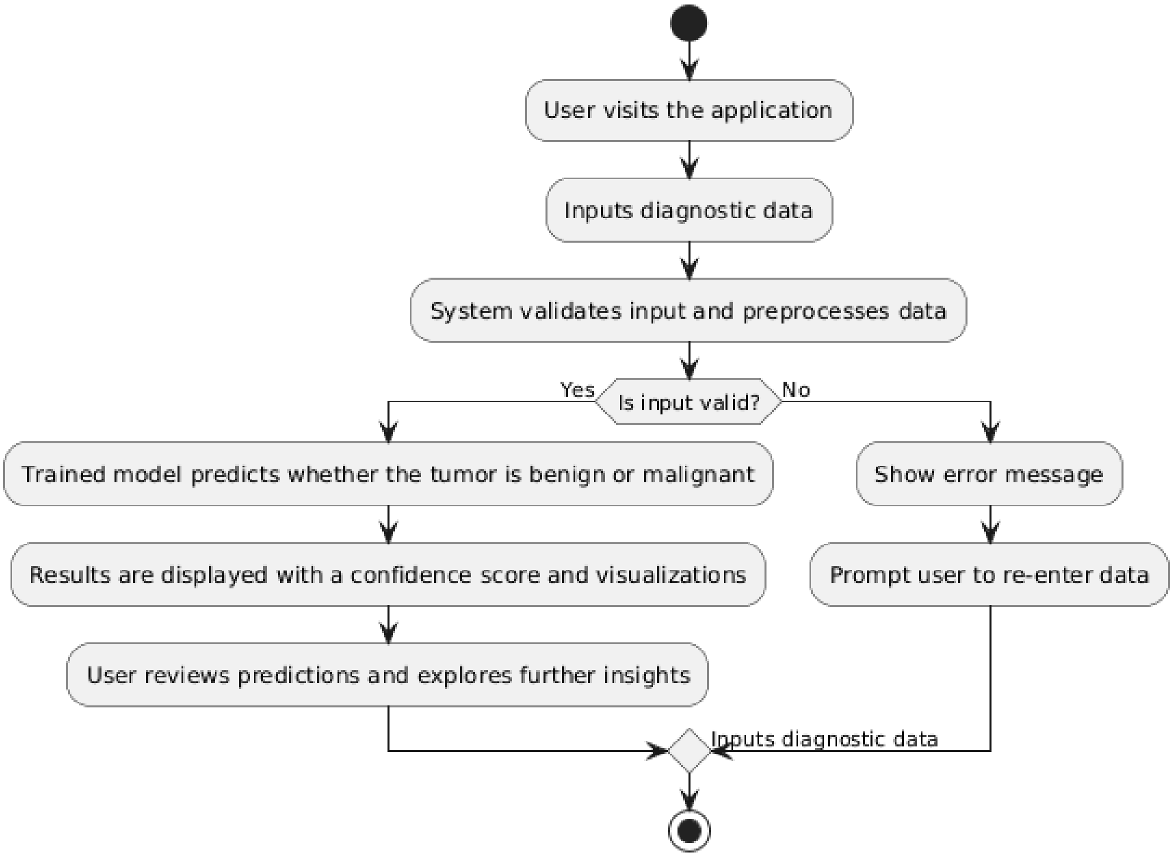
   Displayed a radar chart generated using Plotly and Seaborn to help medical professionals understand data distributions and model predictions visually.

4. **Prediction Section**

   Displayed whether a cell cluster was benign or malignant along with the probability score to indicate the prediction reliability.

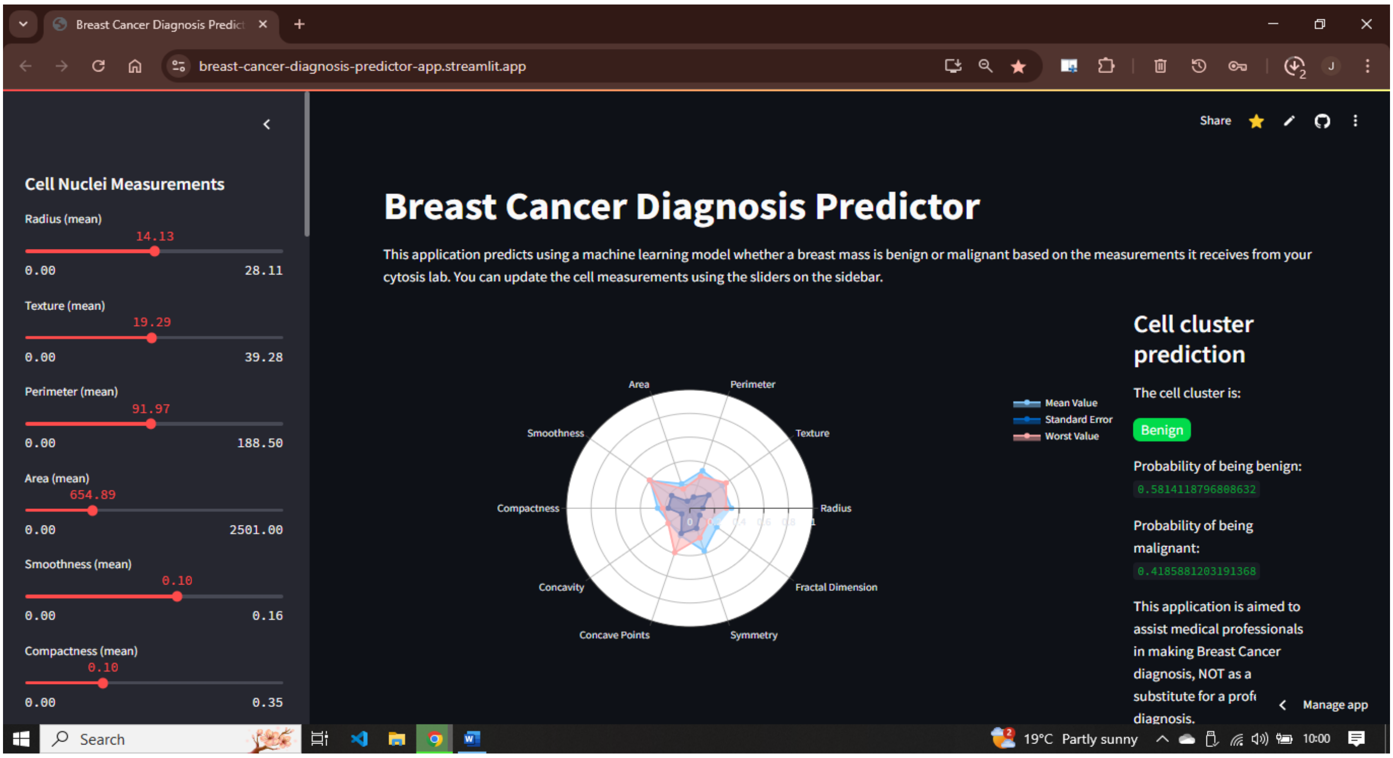
User Flow Diagram

Figure 9- User Flow Diagram


User visits the application
Inputs diagnostic data
System validates input and preprocesses data
Is input valid?
Yes
No
Trained model predicts whether the tumor is benign or malignant
Results are displayed with a confidence score and visualizations
User reviews predictions and explores further insights
Show error message
Prompt user to re-enter data
Inputs diagnostic data

CHAPTER 6: IMPLEMENTATION AND TESTING

This chapter describes the development environment, system components and testing procedures for the **Breast Cancer Diagnosis Predictor Application**. The implementation phase focused on building a web-based machine-learning application that could predict breast cancer diagnoses using patient data.



Cell Nuclei Measurements

Radius (mean) 14.13 (0.00 – 28.11)

Texture (mean) 19.29 (0.00 – 39.28)

Perimeter (mean) 91.97 (0.00 – 188.50)

Area (mean) 654.89 (0.00 – 2501.00)

Smoothness (mean) 0.10 (0.00 – 0.16)

Compactness (mean) 0.10 (0.00 – 0.35)

Breast Cancer Diagnosis Predictor

This application predicts using a machine learning model whether a breast mass is benign or malignant based on the measurements it receives from your cytosis lab. You can update the cell measurements using the sliders on the sidebar.

Cell cluster prediction

The cell cluster is:

Benign

Probability of being benign:

0.5814118796808632

Probability of being malignant:

0.4185881203191368

This application is aimed to assist medical professionals in making Breast Cancer diagnosis, NOT as a substitute for a profe diagnosis.

Figure 10- Breast Cancer Diagnosis Predictor Application

6.1 Development Environment

The development environment was configured to support machine learning model deployment, data visualization and user interaction. The following tools and technologies were used:

Programming Language: Python will be used as the primary language for backend development and programming including model training and analysis.

Framework: Streamlit (for web-based UI development)

Scikit-learn – For machine learning model training and prediction

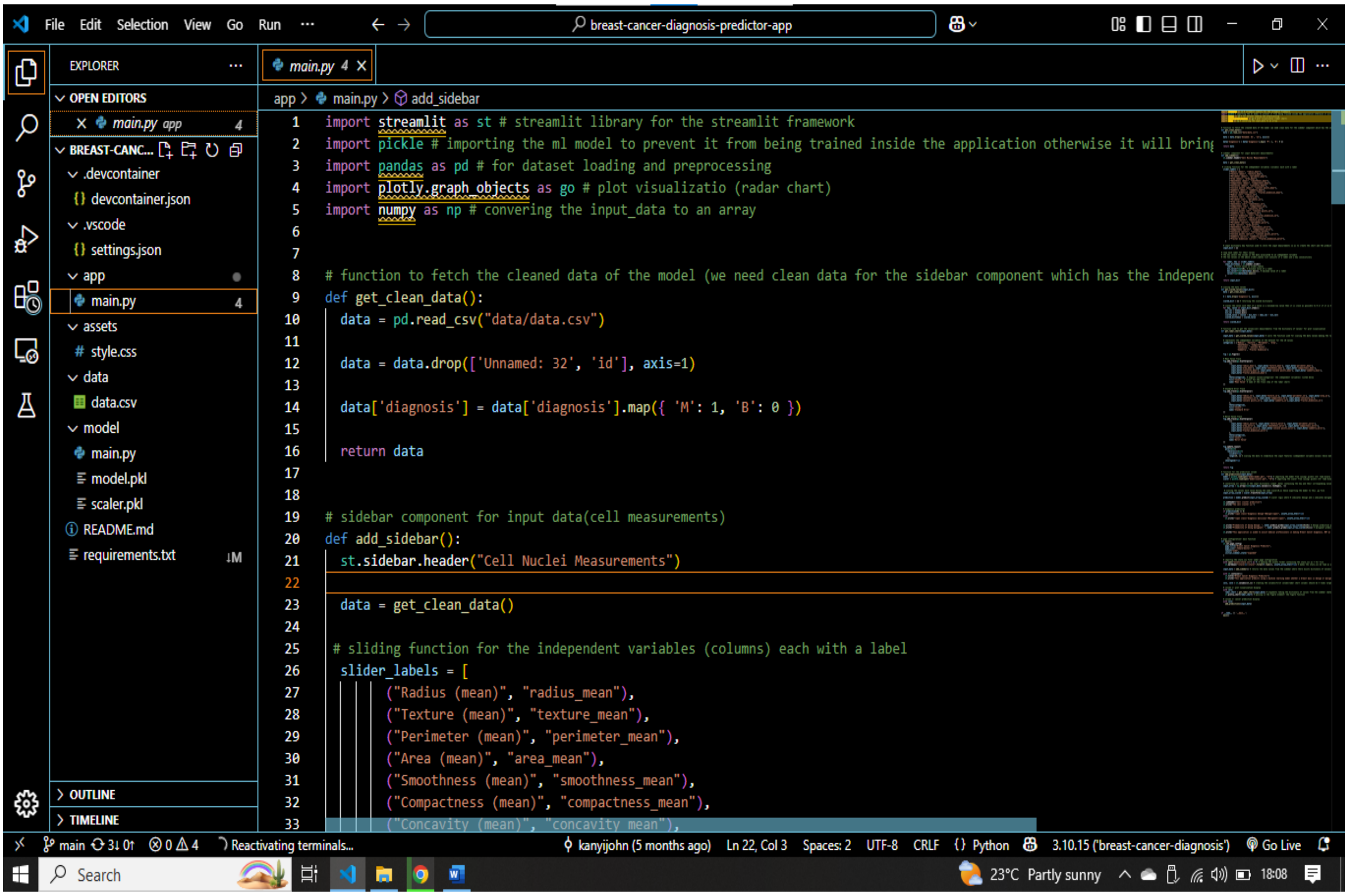
Pandas – For dataset handling and preprocessing

NumPy – For numerical operations and array manipulations

Plotly – For interactive radar chart visualization

Pickle – For model serialization and loading to enhance performance

Altair – For additional data visualization features



```python
import streamlit as st # streamlit library for the streamlit framework
import pickle # importing the ml model to prevent it from being trained inside the application otherwise it will bring
import pandas as pd # for dataset loading and preprocessing
import plotly.graph_objects as go # plot visualizatio (radar chart)
import numpy as np # convering the input_data to an array


# function to fetch the cleaned data of the model (we need clean data for the sidebar component which has the independ
def get_clean_data():
  data = pd.read_csv("data/data.csv")

  data = data.drop(['Unnamed: 32', 'id'], axis=1)

  data['diagnosis'] = data['diagnosis'].map({ 'M': 1, 'B': 0 })

  return data


# sidebar component for input data(cell measurements)
def add_sidebar():
  st.sidebar.header("Cell Nuclei Measurements")

  data = get_clean_data()

 # sliding function for the independent variables (columns) each with a label
  slider_labels = [
        ("Radius (mean)", "radius_mean"),
        ("Texture (mean)", "texture_mean"),
        ("Perimeter (mean)", "perimeter_mean"),
        ("Area (mean)", "area_mean"),
        ("Smoothness (mean)", "smoothness_mean"),
        ("Compactness (mean)", "compactness_mean"),
        ("Concavity (mean)", "concavity_mean"),
```

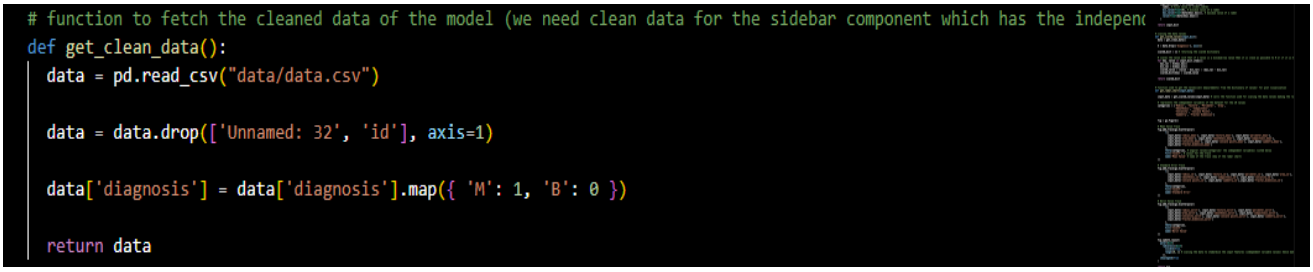
Figure 11- Development Environment

6.2 System Components

The Breast Cancer Diagnosis Predictor Application consisted of several key components that facilitated its operation:

1. Data Processing Component

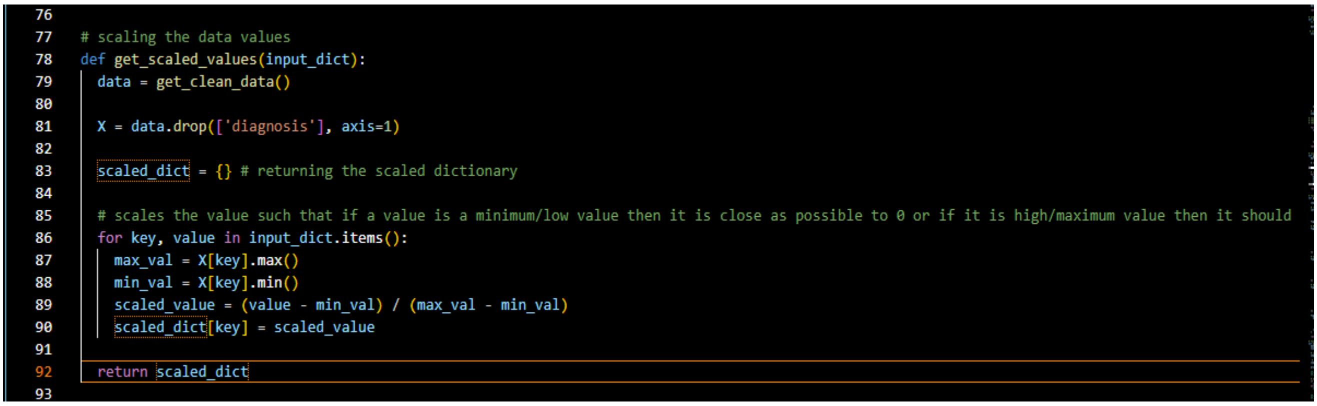
The application loaded the dataset from a CSV file and performed preprocessing tasks such as removing irrelevant columns (Unnamed: 32, id) and encoding categorical values (diagnosis column: Malignant = 1, Benign = 0).



```
# function to fetch the cleaned data of the model (we need clean data for the sidebar component which has the indepen
def get_clean_data():
  data = pd.read_csv("data/data.csv")

  data = data.drop(['Unnamed: 32', 'id'], axis=1)

  data['diagnosis'] = data['diagnosis'].map({ 'M': 1, 'B': 0 })

  return data
```

Figure 12- Encoding

The data was then scaled to ensure uniformity in feature distribution before being used in model prediction. Scaling(normalization) was done in a manner that for if a feature value is a minimum/low value then it is close as possible to 0 and if high/maximum value then it should be close as possible to 1.

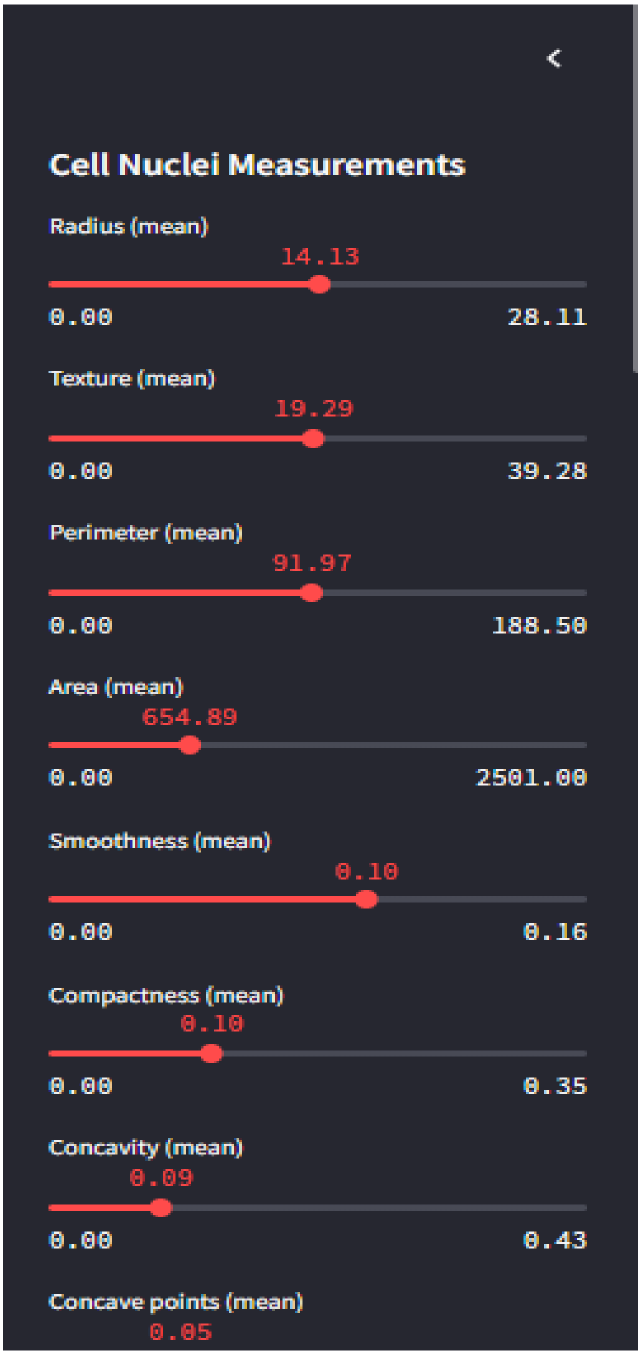


```

# scaling the data values
def get_scaled_values(input_dict):
  data = get_clean_data()

  X = data.drop(['diagnosis'], axis=1)

  scaled_dict = {} # returning the scaled dictionary

  # scales the value such that if a value is a minimum/low value then it is close as possible to 0 or if it is high/maximum value then it should
  for key, value in input_dict.items():
    max_val = X[key].max()
    min_val = X[key].min()
    scaled_value = (value - min_val) / (max_val - min_val)
    scaled_dict[key] = scaled_value

  return scaled_dict

```

Figure 13- Feature Scaling

2. Sidebar Component (User Input Panel)

The **sidebar panel** provided input sliders for medical professionals to adjust the **30 independent variables (cell nucleus measurements)** required for prediction.



Cell Nuclei Measurements

Radius (mean)
14.13
0.00 28.11

Texture (mean)
19.29
0.00 39.28

Perimeter (mean)
91.97
0.00 188.50

Area (mean)
654.89
0.00 2501.00

Smoothness (mean)
0.10
0.00 0.16

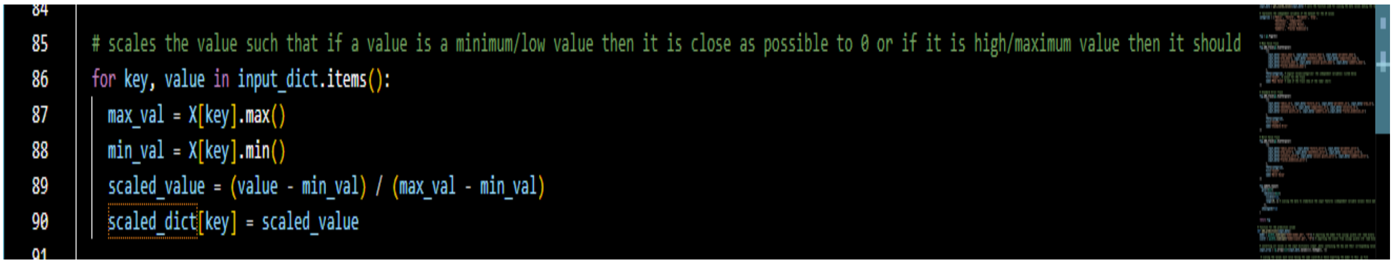
Compactness (mean)
0.10
0.00 0.35

Concavity (mean)
0.09
0.00 0.43

Concave points (mean)
0.05

Figure 14- Sidebar UI

Each slider corresponded to a feature such as radius mean, texture mean, perimeter mean and etcetera. The input values were then normalized before feeding into the model.

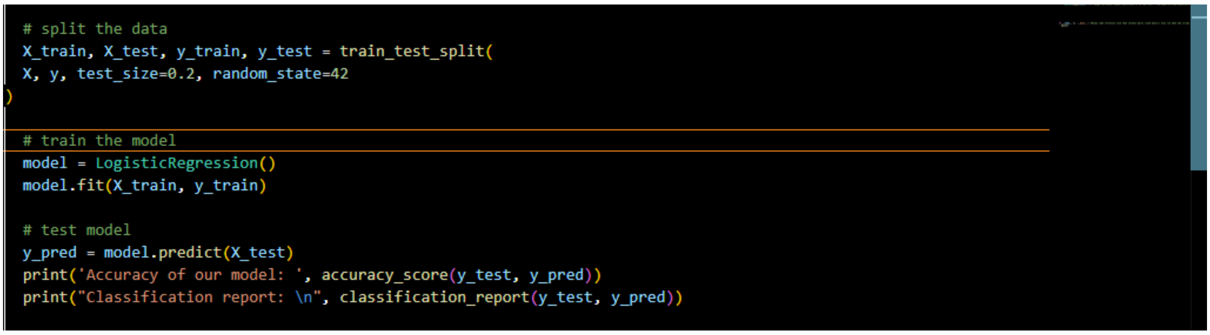


```
# scales the value such that if a value is a minimum/low value then it is close as possible to 0 or if it is high/maximum value then it should
for key, value in input_dict.items():
    max_val = X[key].max()
    min_val = X[key].min()
    scaled_value = (value - min_val) / (max_val - min_val)
    scaled_dict[key] = scaled_value
```

Figure 15

3. Machine Learning Model Component

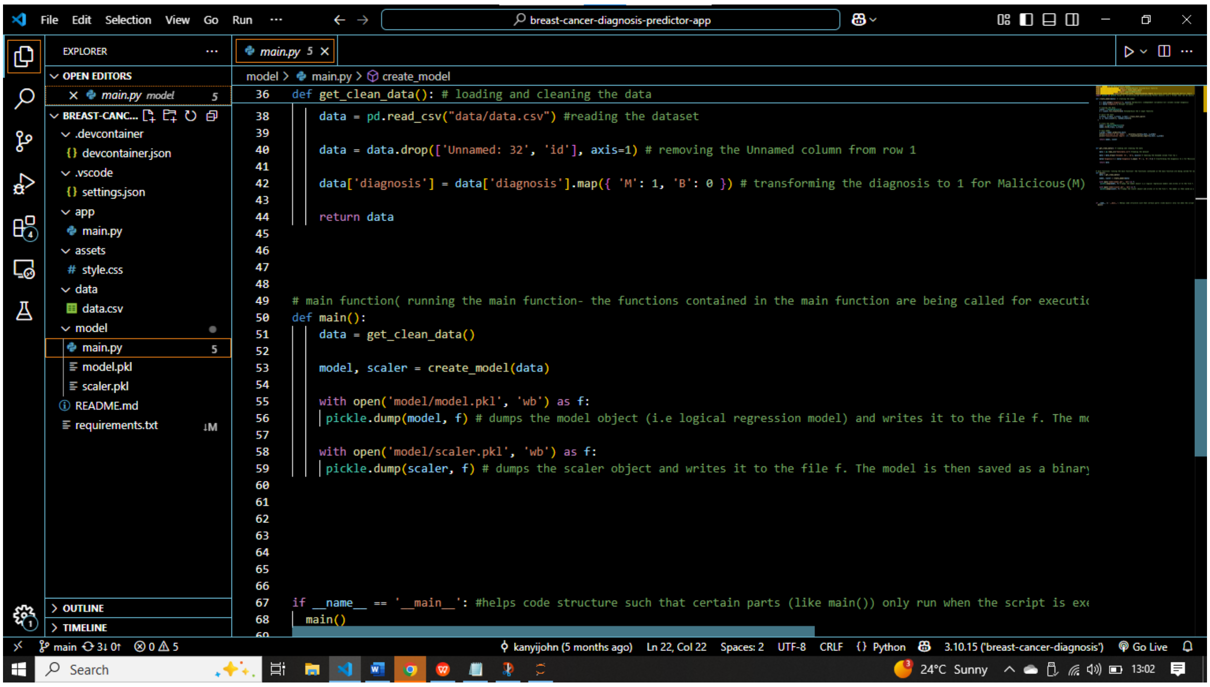
A **logistic regression model** was trained on the Breast Cancer Wisconsin dataset and saved using **Pickle** for future use. The model achieved an accuracy of 98%.



```
    # split the data
    X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

    # train the model
    model = LogisticRegression()
    model.fit(X_train, y_train)

    # test model
    y_pred = model.predict(X_test)
    print('Accuracy of our model: ', accuracy_score(y_test, y_pred))
    print("Classification report: \n", classification_report(y_test, y_pred))
```

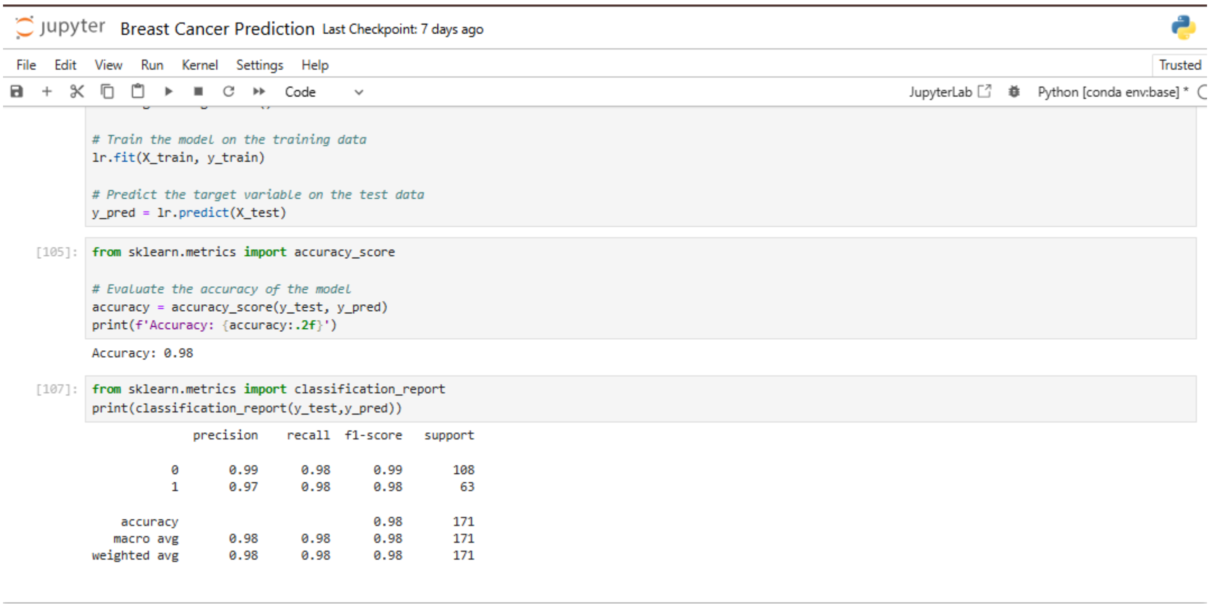
Figure 16-Logistic Regression Model

Figure 17- Pickle



```python
def get_clean_data(): # loading and cleaning the data
    data = pd.read_csv("data/data.csv") #reading the dataset

    data = data.drop(['Unnamed: 32', 'id'], axis=1) # removing the Unnamed column from row 1

    data['diagnosis'] = data['diagnosis'].map({ 'M': 1, 'B': 0 }) # transforming the diagnosis to 1 for Malicicous(M)

    return data



# main function( running the main function- the functions contained in the main function are being called for executic
def main():
    data = get_clean_data()

    model, scaler = create_model(data)

    with open('model/model.pkl', 'wb') as f:
      pickle.dump(model, f) # dumps the model object (i.e logical regression model) and writes it to the file f. The mc

    with open('model/scaler.pkl', 'wb') as f:
      pickle.dump(scaler, f) # dumps the scaler object and writes it to the file f. The model is then saved as a binary







if __name__ == '__main__': #helps code structure such that certain parts (like main()) only run when the script is exe
  main()
```

Figure 18- Model Performance Analysis



```python
# Train the model on the training data
lr.fit(X_train, y_train)

# Predict the target variable on the test data
y_pred = lr.predict(X_test)
```

[105]:
```python
from sklearn.metrics import accuracy_score

# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

```
Accuracy: 0.98
```

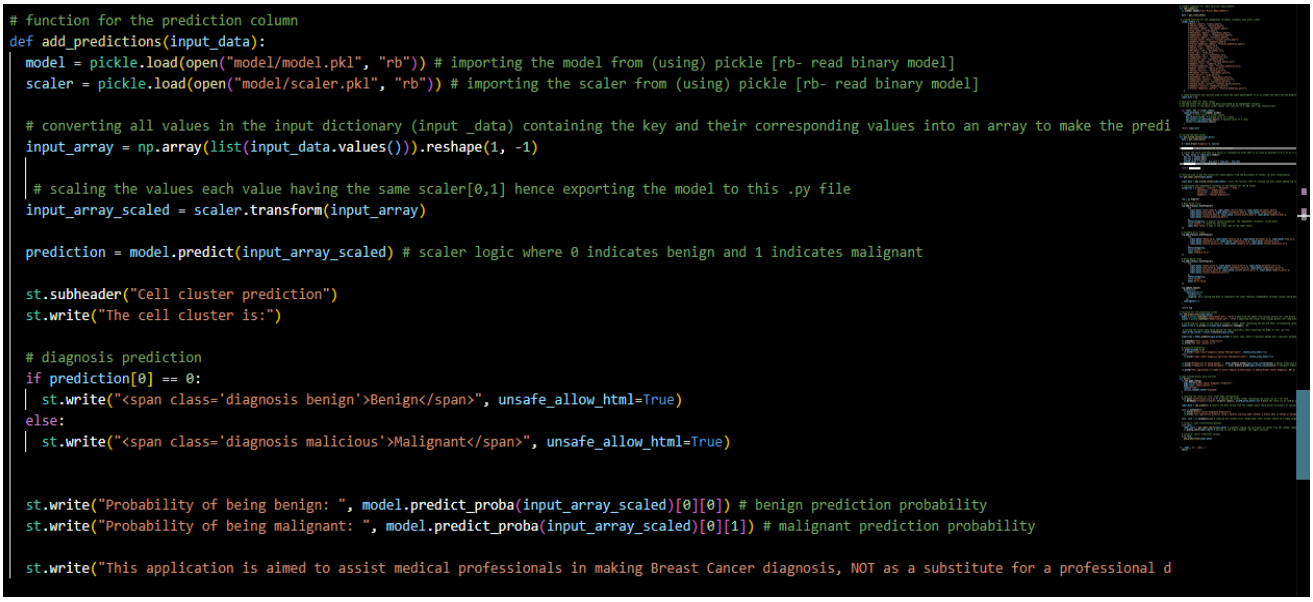
[107]:
```python
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99       108
           1       0.97      0.98      0.98        63

    accuracy                           0.98       171
   macro avg       0.98      0.98      0.98       171
weighted avg       0.98      0.98      0.98       171
```

The model accepted **scaled numerical input** and returned a binary prediction:

0 → Benign

1 → Malignant

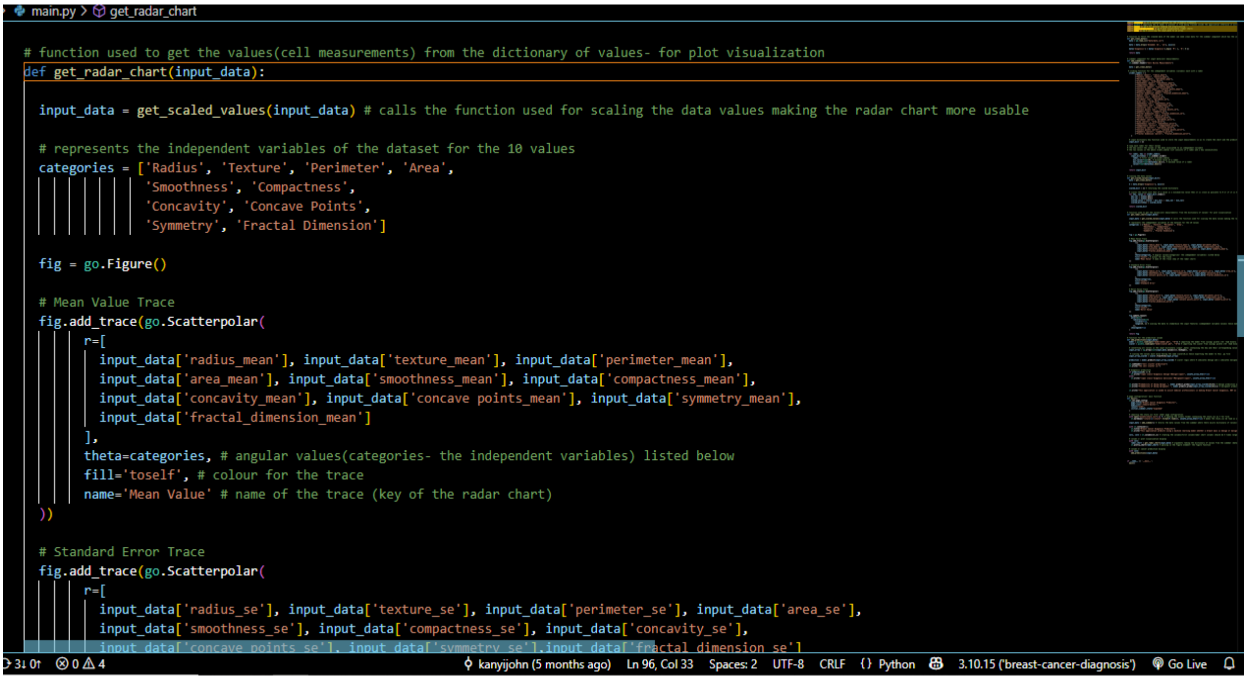


```python
# function for the prediction column
def add_predictions(input_data):
  model = pickle.load(open("model/model.pkl", "rb")) # importing the model from (using) pickle [rb- read binary model]
  scaler = pickle.load(open("model/scaler.pkl", "rb")) # importing the scaler from (using) pickle [rb- read binary model]

  # converting all values in the input dictionary (input _data) containing the key and their corresponding values into an array to make the predi
  input_array = np.array(list(input_data.values())).reshape(1, -1)

   # scaling the values each value having the same scaler[0,1] hence exporting the model to this .py file
  input_array_scaled = scaler.transform(input_array)

  prediction = model.predict(input_array_scaled) # scaler logic where 0 indicates benign and 1 indicates malignant

  st.subheader("Cell cluster prediction")
  st.write("The cell cluster is:")

  # diagnosis prediction
  if prediction[0] == 0:
    st.write("<span class='diagnosis benign'>Benign</span>", unsafe_allow_html=True)
  else:
    st.write("<span class='diagnosis malicious'>Malignant</span>", unsafe_allow_html=True)


  st.write("Probability of being benign: ", model.predict_proba(input_array_scaled)[0][0]) # benign prediction probability
  st.write("Probability of being malignant: ", model.predict_proba(input_array_scaled)[0][1]) # malignant prediction probability

  st.write("This application is aimed to assist medical professionals in making Breast Cancer diagnosis, NOT as a substitute for a professional d
```

Figure 19

4. Data Visualization Component

A **radar chart** was generated using **Plotly**, visualizing the input measurements for Mean, Standard Error and WorstCase values.

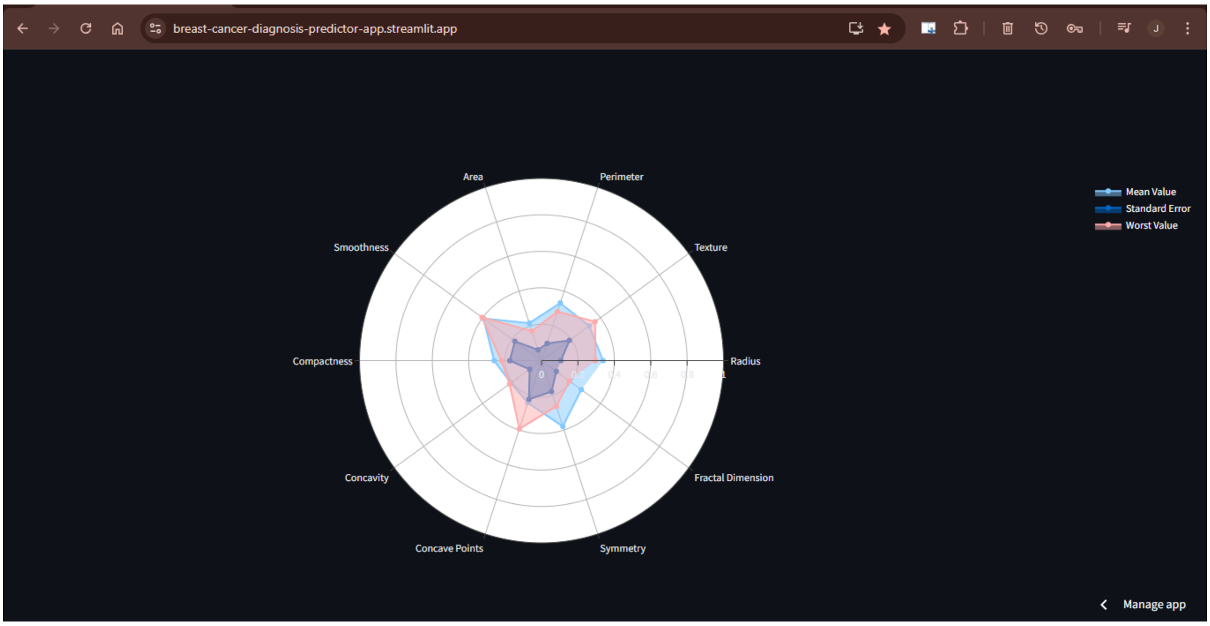


```python
# function used to get the values(cell measurements) from the dictionary of values- for plot visualization
def get_radar_chart(input_data):

  input_data = get_scaled_values(input_data) # calls the function used for scaling the data values making the radar chart more usable

  # represents the independent variables of the dataset for the 10 values
  categories = ['Radius', 'Texture', 'Perimeter', 'Area',
                'Smoothness', 'Compactness',
                'Concavity', 'Concave Points',
                'Symmetry', 'Fractal Dimension']

  fig = go.Figure()

  # Mean Value Trace
  fig.add_trace(go.Scatterpolar(
        r=[
          input_data['radius_mean'], input_data['texture_mean'], input_data['perimeter_mean'],
          input_data['area_mean'], input_data['smoothness_mean'], input_data['compactness_mean'],
          input_data['concavity_mean'], input_data['concave points_mean'], input_data['symmetry_mean'],
          input_data['fractal_dimension_mean']
        ],
        theta=categories, # angular values(categories- the independent variables) listed below
        fill='toself', # colour for the trace
        name='Mean Value' # name of the trace (key of the radar chart)
  ))

  # Standard Error Trace
  fig.add_trace(go.Scatterpolar(
        r=[
          input_data['radius_se'], input_data['texture_se'], input_data['perimeter_se'], input_data['area_se'],
          input_data['smoothness_se'], input_data['compactness_se'], input_data['concavity_se'],
          input_data['concave points_se'], input_data['symmetry_se'],input_data['fractal dimension se']
```

Figure 20

The **interactive plot** allowed users to compare different cell features in a single view.

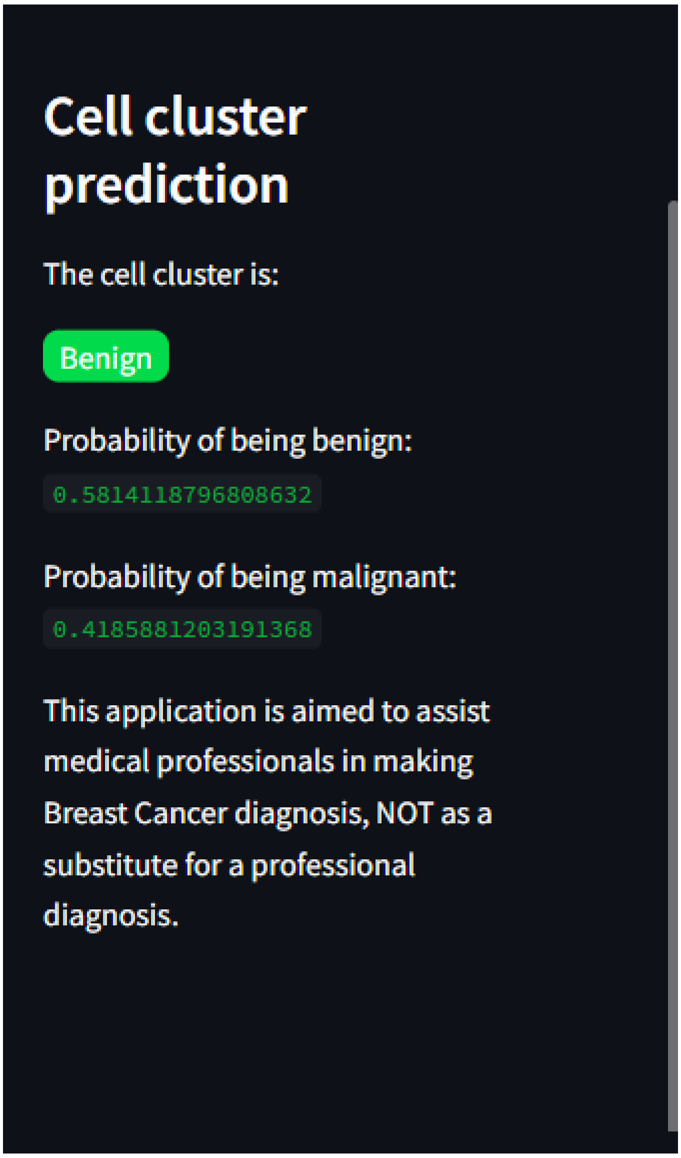
Figure 21- Radar Chart



5. Prediction Display Component

The system displayed **real-time predictions** based on user inputs, along with probability score for each prediction, indicating the confidence level in the classification. A disclaimer was included stating that the application was intended to **assist** medical professionals, not replace them. The output was presented clearly with the inclusion of CSS to distinguishing between benign and malignant classifications for usability and readability.

Figure 22- Prediction



Cell cluster prediction

The cell cluster is:

Benign

Probability of being benign:

```
0.5814118796808632
```

Probability of being malignant:

```
0.4185881203191368
```

This application is aimed to assist medical professionals in making Breast Cancer diagnosis, NOT as a substitute for a professional diagnosis.

6.3 Test Plan

Testing was conducted to validate the functionality, usability, and accuracy of the application. The test plan included:

1. Test Data

The test data consisted of sample patient records from the Breast Cancer Wisconsin dataset.

Various values were fed into the system to simulate real-world scenarios.

2. Test Cases

The following test cases were executed:

Table 1

Test Case ID	**Test Scenario**	**Expected Outcome**	**Actual Outcome**	**Status**
TC01	Load dataset and preprocess	Dataset loaded successfully	Dataset cleaned and ready for use	*Pass*
TC02	User inputs values via sliders	System updates values dynamically	Input values updated correctly	*Pass*
TC03	Model makes a prediction	System returns "Benign" or "Malignant" with probability scores	Predictions displayed correctly	*Pass*
TC04	Radar chart visualization	Chart updates dynamically based on inputs	Chart displayed correctly	*Pass*
TC05	Prediction probability calculation	Probabilities sum to 1	Probabilities correctly displayed	*Pass*
TC06	Model accuracy validation	Accuracy should be above 90%. It achieved a 98% accuracy	Model achieved high accuracy	*Pass*
TC07	Application responsiveness	UI should load without delays	UI loaded smoothly	*Pass*

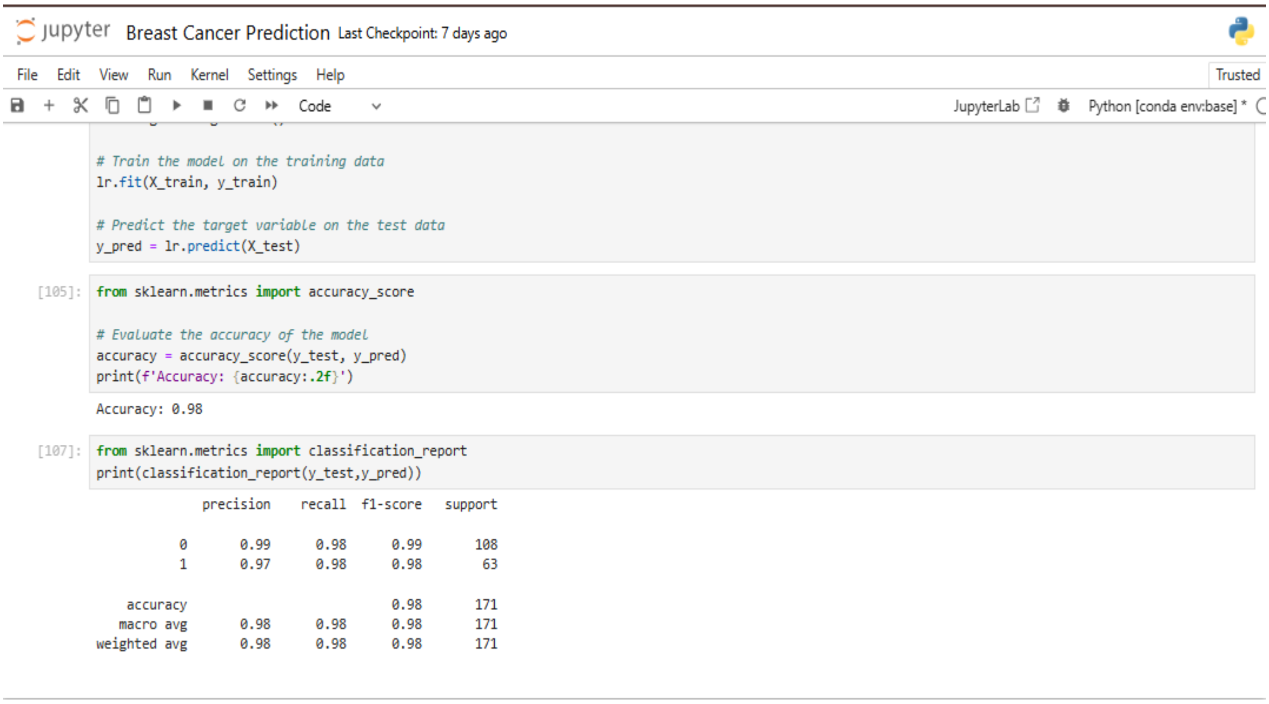
6.4 System Testing

The system underwent various testing phases to ensure reliability and accuracy:

Functionality Testing

The logistic regression model's accuracy and classification report including the precision call etcetera was validated by splitting the dataset into training and testing subsets, to test the correctness of predictions from the Logistic Regression model, ensuring it performs well on unseen data.

Figure 23


jupyter
Breast Cancer Prediction Last Checkpoint: 7 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *

```
# Train the model on the training data
lr.fit(X_train, y_train)

# Predict the target variable on the test data
y_pred = lr.predict(X_test)
```

```
[105]: from sklearn.metrics import accuracy_score

       # Evaluate the accuracy of the model
       accuracy = accuracy_score(y_test, y_pred)
       print(f'Accuracy: {accuracy:.2f}')
```

```
Accuracy: 0.98
```

```
[107]: from sklearn.metrics import classification_report
       print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99       108
           1       0.97      0.98      0.98        63

    accuracy                           0.98       171
   macro avg       0.98      0.98      0.98       171
weighted avg       0.98      0.98      0.98       171
```

Unit Testing

Each function, including data preprocessing, feature scaling, model loading, and prediction generation, was tested individually.

Sample input values were passed through the application to validate correct execution.

Integration Testing

The frontend (Streamlit UI) and backend (ML model and data processing pipeline) were tested together.

The goal was to ensure seamless interaction between different components.

Usability Testing

The application was tested by to verify ease of use. Feedback was gathered regarding UI clarity, navigation and interpretability of results.

Performance Testing

The system was tested under different workloads to measure response time and scalability.The radar chart and prediction modules were optimized to ensure smooth user interaction.

The implementation and testing phases successfully validated the Breast Cancer Diagnosis Predictor Application. The system demonstrated **high accuracy, usability, and performance**, ensuring it could effectively assist medical professionals in breast cancer diagnosis. The application was prepared for **deployment on Streamlit Cloud (https://breast-cancer-diagnosis-predictor-app.streamlit.app)**, making it accessible for real-world usage.

CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS

7.1 Achievements and Lessons Learnt

The development of the Breast Cancer Diagnosis Predictor Application resulted in the successful implementation of a functional, interactive, and user-friendly machine learning system capable of predicting whether a breast tumor was benign or malignant using the Breast Cancer Wisconsin Diagnostic dataset. Key technical achievements included training a high-accuracy logistic regression model using Scikit-learn, integrating it into a responsive Streamlit-based web interface, and deploying interactive visualizations through Plotly and Altair. The application enabled real-time predictions with probability scores and displayed results using an informative radar chart. Throughout the project, knowledge was gained in full-stack ML application development, including model integration, data preprocessing, web interface development, and system testing. All project objectives were met except for advanced model comparison; only logistic regression was used without benchmarking it against other algorithms such as Random Forest or SVM due to time constraints.

7.2 Conclusions

In conclusion, the project successfully demonstrated the practical use of machine learning in supporting medical decision-making for breast cancer diagnosis. The Waterfall SDLC methodology ensured a structured and phased approach to system design, implementation, and testing, resulting in a reliable and accessible diagnostic tool. The system proved to be effective, scalable and easy to use especially in resource-limited settings.

7.3 Recommendations

For future improvement, it is recommended to enhance the system by incorporating more machine learning models for comparative performance analysis expanding the application with real-time database support for storing prediction history and integration with external systems, such as cytology lab machines with user authentication for automatic data input.

REFERENCES

1. Amann, J., Blasimme, A., Vayena, E., Frey, D., & Madai, V. I. (2020). Explainability for artificial intelligence in healthcare: A multidisciplinary perspective. BMC Medical Informatics and Decision Making, 20(1), 310. https://doi.org/10.1186/s12911-020-01332-6

2. Bondy, M. L., et al. (2005). Race, ethnicity, and breast cancer: The debatable role of genetics. Oncologist, 10(1), 59–69.

3. Costantino, J. P., et al. (1999). Validation studies for models projecting the risk of invasive and total breast cancer incidence. Journal of the National Cancer Institute, 91(18), 1541–1548. https://doi.org/10.1093/jnci/91.18.1541

4. Doi, K. (2007). Computer-aided diagnosis in medical imaging: Historical review, current status, and future potential. Computerized Medical Imaging and Graphics, 31(4–5), 198–211. https://doi.org/10.1016/j.compmedimag.2007.02.002

5. Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., ... & Wang, Y. (2017). Artificial intelligence in healthcare: Past, present, and future. Stroke and Vascular Neurology, 2(4), 230–243. https://doi.org/10.1136/svn-2017-000101

6. Kooi, T., Litjens, G., van Ginneken, B., Gubern-Mérida, A., Sánchez, C. I., Mann, R., & den Heeten, A. (2017). Large scale deep learning for computer-aided detection of mammographic lesions. Medical Image Analysis, 35, 303–312.

7. Lee, C. H., Yoon, H. J., & Kim, D. H. (2018). Application of artificial intelligence in cancer diagnosis and predictive models. Cancer Informatics, 17, 1–7.

8. Lehman, C. D., Wellman, R. D., Buist, D. S. M., Kerlikowske, K., Tosteson, A. N. A., & Miglioretti, D. L. (2015). Diagnostic accuracy of digital screening mammography with and without computer-aided detection. JAMA Internal Medicine, 175(11), 1828–1837. https://doi.org/10.1001/jamainternmed.2015.5231

9. Taylor, P., Potts, H. W. W., & Benson, R. (2019). Computer-aided detection in mammography: An overview and future outlook. Radiology Research and Practice, 2019, 5876543.

10. Tyrer, J., et al. (2004). A breast cancer prediction model incorporating familial and personal risk factors. Statistics in Medicine, 23(7), 1111–1130. https://doi.org/10.1002/sim.1668

11. Wolberg, W. H. (1995). Breast Cancer Wisconsin (Diagnostic) Dataset. University of Wisconsin Hospitals, Madison. Retrieved from https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

APPENDICES

APPENDIX A: PROPOSED SCHEDULE

Table 2

Activities	**Months**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Project Conceptualization and Requirements Gathering	■							
Project Research and Literature Review		■						
System Methodology			■					
System Analysis and Design				■	■	■		
System Implementation and Testing					■	■	■	
Project Proposal Writing and Submission		■	■	■	■	■	■	■
Corrections and Final Submission				■	■	■	■	■

APPENDIX B: PROPOSED BUDGET

Table 3

S. No	**Items/Activities**	**Quantity**	**@Ksh**	**Amount in Ksh**
1.	Paper Printing	6 dozen	60	360
2.	Spiral Binder	2	70	140
3.	Paper Punch	1	700	700
4.	Miscellaneous			500
		Total		**sh.1750**