

CHAPTER 3- METHODOLOGY

3.1 Introduction

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

Python will be 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 will be done using Streamlit and CSS, which will enable the medical professional to input diagnostic data and get predictions. Additional data visualization libraries such as Plotly will be used in making an informative and interactive radar chart. The machine learning model to be used, logistic regression, will be trained using the Breast Cancer Wisconsin Diagnostic Dataset from Kaggle (<https://www.kaggle.com/uciml/breast-cancer-wisconsin-data>) and will serve as the basis for the predictive model to classify whether a cell cluster is either benign or malignant.

For deployment and hosting, this project will make use of Streamlit Cloud which provides easy deployment and maintenance of the application to keep it accessible via any web browser. The same platform also supports scalability so that the application is accessible across a range of devices without requiring any local installations.

Along with the development tool and framework, this chapter also designates 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 are also provided to make sure of effective planning and resource allocation throughout the project lifecycle.

3.2 Project Design (Waterfall Method)

The project will follow 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 will only move to the next phase once the current one is completed ensuring that no important steps are overlooked.

❖ Justification for the Waterfall Model

1. **Simplicity:** The rigid structure of the waterfall model is easy to understand and implement; hence, it is easy to manage and thus suitable for this project.
2. **Well-defined Milestones:** Each project phase has clear objectives and deliverables, ensuring clarity in the progress tracking.
3. **Focus on Documentation:** The model emphasizes documentation at each stage, something that is essential for this project.
4. **Minimized Risk:** Testing falls in a different phase after development, thus minimizing unexpected problems at deployment.

The **Waterfall Model** will guarantee that the project is 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*** is structured to align with the Waterfall SDLC methodology model. Each step of the process is elaborated on in this section to ensure clarity and alignment with the project objectives and deliverables.

1. Requirements Gathering and Analysis

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

- **Diagnosis Prediction:** It should classify cell clusters as either benign or malignant using a logistic regression model.
- **Visualization:** It should visualize 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 should be *compatible* with the latest web browsers, make fast predictions, and be *responsive on all devices*. *Performance indicators* such as *low latency* in predictions and *seamless integration* of machine learning and visualization components will be *prioritized*.

2. System Architecture

The application architecture will be designed as a *three-layered* structure:

- ✓ **Frontend:** the frontend will be developed using Streamlit and a bit of CSS to provide an interactive dashboard where users input diagnostic data and view predictions along with their probability scores.
- ✓ **Backend:** A Python-based machine learning model (logistic regression) will be implemented using machine learning libraries like Scikit-learn for predictive analysis etcetera.
- ✓ **Database:** Since the application uses a pre-existing dataset (Breast Cancer Wisconsin Diagnostic Data from Kaggle (<https://www.kaggle.com/uciml/breast-cancer-wisconsin-data>)), there won't be a traditional database. The model will directly load and process this dataset. The data flow will involve receiving inputs from the user interface, passing them through the trained model and displaying predictions and related visualizations.
- ✓ **Visualization:** The application will utilize Plotly to generate an interactive radar chart to enhance user comprehension of predictions.

3. Detailed Design

The system shall be designed in a manner to maximize user experience and computational efficiency.

- **User Flow:** The user's journey from inputting diagnostic data to presenting results shall be mapped. Each step in the interaction shall be intuitive thus ensuring that users can understand and trust the predictions provided.
- **Integration:** The backend that is the machine learning model, will interact with the frontend to produce results in real time. The architecture will be modularly integrated to support updates and enhancements.

4. User Interface Design

The UI will be designed with simplicity and functionality in mind.

- **Input Fields:** The user will 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 will present the predictions, the probability score and visualizations in a clear, orderly fashion.
- **Charts:** Interactive visualizations via a radar chart using Plotly will be used to better understand the model predictions thus help in decision making.

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

5. Data Design

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

- **Data Format:** The dataset will be pre-processed for handling missing values, data normalization and preparation of features for the logistic regression model (EDA).
- **Data Flow:** User inputs will be 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 will be displayed on the dashboard.

6. Implementation

- **Backend Development:** Train the logistic regression model using Scikit-learn and preprocess the dataset using Pandas and Numpy.

Develop a Python script to load the model and return predictions based on user inputs.

Use 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:** Build an interactive Streamlit application that captures user inputs and displays model predictions.

Employ Plotly for creating the radar chart for visualization of the results.

Integration of these components will give us the full-stack AI application.

7. Testing and Validation

Full Testing (System Testing) will make sure the application is seamlessly functional:

- **Functionality Testing:** The logistic regression model's accuracy and classification report including the precision call etcetera will be 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
- **Unit Testing:** The system's individual components such as the machine learning model, user input forms, probability score and visualizations will be tested.
- **Integration Testing:** Frontend and backend work in cohesion, and Streamlit allows seamless integration of CSS, Plotly, and the machine learning model.
- **System Testing:** test the system application as a whole.

- **User Acceptance Testing (UAT):** After the system is deployed, it will undergo UAT to ensure that it meets user expectations and works as intended.

8. Deployment

The application will be deployed on Streamlit Cloud, ensuring:

- Public accessibility through a web browser with no need for local installations.
- Scalable to support multiple users.
- Ease in maintenance for future updates and enhancements.

Streamlit Cloud offers a free deployment solution that is efficient and hosts support to share Streamlit applications.

9. Maintenance

After deployment, the system will enter the phase of maintenance where:

- **Bug Fixes:** Any bugs detected post-deployment are fixed.
- **Updates:** Integrate users' feedback to enhance the working and usability of the application.
- **Enhancements:** Visualizations will be enhanced; the predictive capabilities will be extended or new features added to the application based on changes in medical knowledge or users' requirements.

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

3.4 System Requirements

Functional Requirements:

1. **Data Input:** The user (medical professional) should be able to enter the values for different features of the cell cluster, like radius, texture, perimeter and etcetera.
2. **Prediction Output:** The system should predict along with the probability score whether a cell cluster is benign or malignant scores based on the input data fed into it.
3. **Visualization:** The system should visualize the model's predictions through an interactive radar chart.
4. **Model Accuracy:** The machine learning model should have high accuracy in predicting the classification of breast cancer.

Non-Functional Requirements:

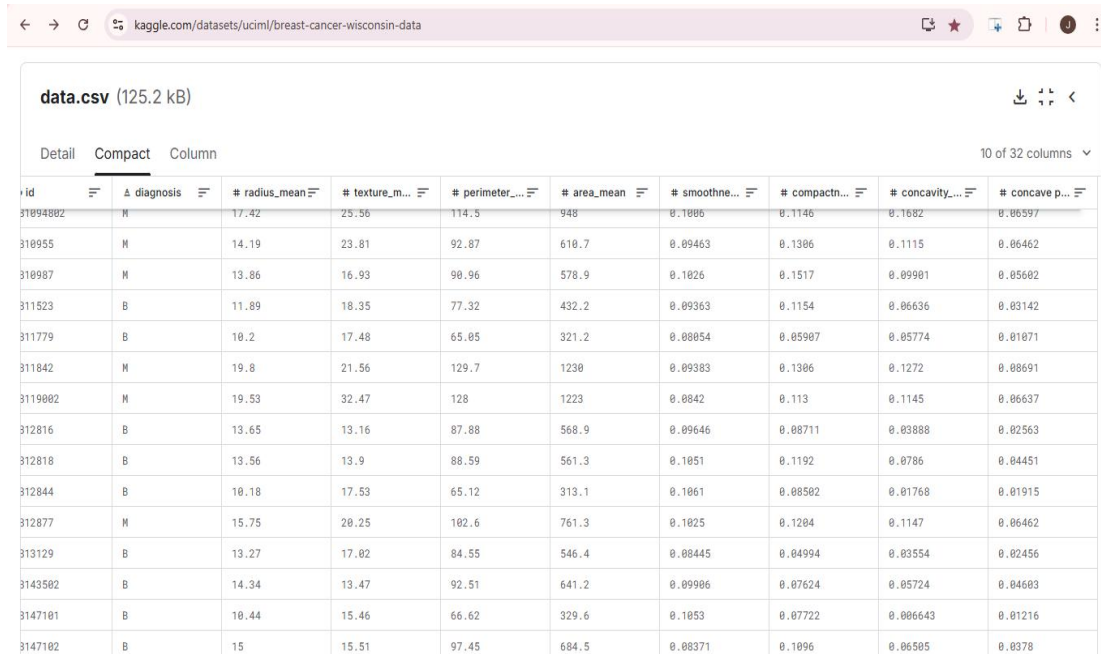
1. **Usability:** The web application should be user-friendly, ensuring that users with minimal technical knowledge can interact with it effectively.
2. **Performance:** The system should be responsive, having little to no delay in the display of the UI components including the dashboard, predictions and the radar visualization.
3. **Security:** The application must securely handle any sensitive user data; in this case, no personally identifiable information is used.
4. **Scalability:**
The application should be able to handle large volume of user data and multiple concurrent live users.
5. **Compatibility:** The system should be able to function on multiple OS platforms and different device architectures for user accessibility.
6. **Reliability:** The system should be 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 is 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



id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
31094002	M	17.42	25.56	114.5	948	0.1086	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 contains 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.

1. ID Column:

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

2. 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.

3. 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:

1. **Mean Values:** Measurements averaged across the sample.
2. **Standard Errors:** Variability of the sample measurements.
3. **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 is well-balanced, with a slight predominance of benign cases, with 357 being benign and 212 malignant.
- All the feature values are 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 is highly suitable for the project due to its:

- ***Real-world Relevance:*** Since it is from medical research, it reflects real clinical data.
- ***Completeness:*** The high-dimensional feature set enables the fine-grained characterization of cell clusters and hence it will improve the predictive capability of the machine learning model.
- ***Quality:*** There are no missing or mismatching data points hence allowing a sound and reliable model training.

Data Analysis:

1. **Preprocessing:** The dataset will undergo necessary preprocessing steps such as normalization feature selection etcetera.
2. **EDA:** Using visualization and statistical analysis with the ML libraries; Pandas, Numpy etcetera in Python to understand the distribution of the data and identify patterns.
3. **Model Training:** Data will be split into both training and testing sets. In this case, the Logistic Regression model will be trained using Scikit-learn.
4. **Model Evaluation:** The model's performance will be evaluated using metrics such as accuracy, precision, recall, and F1-score.

Tools To Be Used:

1. **Python:** The primary language for the backend development and programming including model training and analysis.
2. **Streamlit:** Streamlit shall be employed for building up a web interactive dashboard UI for the app.
3. **Scikit-learn:** Will be use to train the logistic regression of the machine learning model.
4. **Plotly:** For an interactive visualizations of data input and predictions.
5. **Pandas:** For data Manipulation and pre-processing.
6. **Numpy:** For numerical computing and array operations to enhance data preprocessing and statistical computations.

7. **Pickle:** Will be employed to serialize and save the trained machine learning model for efficient performance, deployment and future use.
8. **Altair:** For creating declarative and simple statistical visualization to complement Plotly's interactive visualizations.

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

APPENDICES

APPENDIX A: PROPOSED SCHEDULE

Table 1

Activities	Months							
	1	2	3	4	5	6	7	8
Project Conceptualization and Requirement Gathering								
System Analysis and Design								
Project Proposal Writing and Submission								
System Development								
System Implementation, Testing and Validation								
Deployment and Maintenance								
Project Report Writing								
Final Review and Project Submission and Presentation								

APPENDIX B: PROPOSED BUDGET

Table 2

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