

---

# CANCER GUARDIAN-BREAST CANCER PREDICTION APP

---

*Submitted in partial fulfillment of the requirements for the award of the  
degree of*

**Bachelor of Computer Applications (BCA)**

To

Guru Gobind Singh Indraprastha University, Delhi



**Maharaja Surajmal Institute,  
New Delhi – 110058  
Batch (2021-2024)**

**Guide:**

Ms. Meenal Dhaiya  
(Assistant Professor)

**Submitted By:**

Zainab -02214902021  
Khushi Khari - 06514902021  
Soumya Shubham - 01314902021  
Aarush Sachdeva - 04514902021

## **ACKNOWLEDGEMENT**

Perseverance and inspiration always play a key role in the success of any project and it has been the same for us as well. At the very least, We would like to take this opportunity to thank everyone who helped me with this project.

We would like to thank my minor project report coordinator Mr's **Meenal Dhiya**. Without her guide guidance, cooperation, inspiration, and knowledge, it would have been extremely difficult for us to complete this project successfully.

Finally, We would like to express my gratitude towards those people who directly or indirectly helped me, by providing necessary information required for the completion of this project.

## **CERTIFICATE OF THE PROJECT**

This is to certify that this project entitled “Breast Cancer Prediction Using Logistic Regression” submitted in partial fulfilment of the degree of Bachelor of Computer Applications to the “\_\_\_\_\_” through \_\_\_\_\_ done by Mr/Ms Zainab Asif,Khushi Khari,Aarush Sachdeva,Soumya Shubham,Roll no \_\_\_\_\_ is an authentic work carried out by him/her at \_\_\_\_\_ under my guidance.The matter embodied in this project work has not been submitted earlier for award of any degree to the best of my knowledge and belief.

Signature of the student

Signature of the Guide

## **MINOR PROJECT SYNOPSIS**

### **TITLE OF THE PROJECT**

Our project refers to the name **Cancer Guardian** . The project answers solely to classification of which type of breast cancer the patient is suffering from . It is developed using logistic regression for various dependent variables it maps out the independent variable which is the type of breast cancer . The project is first hand made for hospitals and pathologists . The project can be made more diverse and complexity can be introduced on later .

### **OBJECTIVE AND SCOPE**

Our project is dedicated to the early detection of breast cancer through the analysis of Fine Needle Aspiration (FNA) reports. Fine Needle Aspiration is a minimally invasive procedure commonly used for obtaining tissue or fluid samples from suspicious masses or lumps in the breast. This project leverages machine learning and data analysis techniques to interpret FNA reports, assisting in the prompt identification of cancerous cells. The objective of Cancer Guardian, our web application, is to accurately analyze and classify breast cancer data obtained through fine needle aspiration, providing a reliable tool for healthcare professionals. By leveraging advanced algorithms, our platform aims to efficiently process this data to distinguish between benign and malignant cases, aiding in swift and precise diagnoses. Through this, we aim to enhance clinical decision-making, improve patient outcomes, and contribute to the advancement of breast cancer diagnosis and treatment.

Scope of our project is for use in hospitals which is expandable , following are the scope pointers for our project:-

1. **Data Integration and Input Handling:** Develop a user-friendly interface allowing hospitals and pathologists to securely upload fine needle aspiration (FNA) data. Support various file formats commonly used in FNA, ensuring compatibility with different equipment and systems.
2. **Data Preprocessing and Cleaning:** Implement algorithms to preprocess and clean input data, ensuring uniformity and accuracy in analysis. Address missing or incomplete data points to enhance the reliability of classification results.
3. **Feature Extraction and Analysis:** Extract relevant features from the FNA data, utilizing imaging and pathological parameters. Implement machine learning models to analyse and process these features, identifying patterns indicative of benign or malignant characteristics.

4. **Classification and Diagnosis:** Develop a classification system capable of accurately distinguishing between benign and malignant cases of breast cancer. Provide detailed reports or summaries displaying the classification results for each case.
5. **Security and Compliance:** Implement robust security measures to safeguard patient data in compliance with healthcare regulations (e.g., HIPAA). Ensure encryption protocols, access controls, and data anonymization to protect sensitive patient information.

## PROCESS DESCRIPTION

**Data Collection and Preprocessing:** We have compiled a diverse and comprehensive dataset of FNA reports, including samples from both benign and malignant cases. The dataset undergoes preprocessing to standardize and enhance the quality of the data, ensuring optimal performance of the machine learning model.

**Machine Learning Model:** Our project employs a sophisticated machine learning model trained on the pre-processed FNA report data. The model has learned to recognize patterns and features indicative of cancerous cells, making predictions based on the input FNA report.

**Feature Extraction:** The FNA report contains vital information about cell morphology, structure, and other characteristics. The machine learning model extracts relevant features from the FNA report, transforming the raw data into meaningful representations.

**Prediction and Classification:** The model utilizes the extracted features to make predictions regarding the nature of the cells in the FNA report. The primary classification is between benign and malignant cells, providing crucial information for early cancer detection.

**Interpretation and User Interface:** The project includes an intuitive and user-friendly interface for inputting FNA reports. Users receive clear and concise results, indicating whether the FNA report suggests a likelihood of malignancy.

**Advantages and Impact :Early Detection:** By automating the analysis of FNA reports, our project facilitates the early detection of breast cancer. Early detection is crucial for improved treatment outcomes and increased survival rates.

**Reduced Subjectivity:** The machine learning model minimizes subjective interpretation, offering a standardized and objective assessment of FNA reports.

User-Friendly Interface: The user interface is designed to be accessible for healthcare professionals, providing rapid and accurate results without the need for extensive training.

## RESOURCE

### HARDWARE REQUIREMENTS:

- LAPTOPS
- PRINTERS

### SOFTWARE REQUIREMENTS:

- GOOGLE COLLABORATORY
- PYTHON LIBRARIES
- STREAMLIT LIBRARY
- STREALIT DEPLOYMENT PLATFORM
- VS CODE FOR VIRTUAL ENVIRONMENTS

## CONCLUSIONS

The development and implementation of Cancer Guardian represent a significant stride forward in the realm of breast cancer diagnosis, particularly in leveraging fine needle aspiration (FNA) data for accurate classification. Through the meticulous integration of advanced algorithms and robust data processing techniques, this web application offers a promising solution to hospitals and pathologists dealing with breast cancer diagnoses.

The successful execution of Cancer Guardian underscores its potential as a pivotal tool in clinical settings. The platform's ability to analyze FNA data and effectively distinguish between benign and malignant cases marks a crucial advancement in expediting diagnostic processes while maintaining a high level of accuracy.

Throughout the development process, our team prioritized not only the technical sophistication but also the user experience and ethical considerations. The emphasis on user-friendly interfaces, stringent security measures, and compliance with healthcare regulations ensures that Cancer Guardian aligns with the highest standards of patient data protection and user accessibility.

Moreover, the iterative nature of Cancer Guardian's machine learning models, coupled with real-time feedback mechanisms from pathologists, lays the groundwork for continuous improvement. As the platform interacts with more data and receives expert insights, it evolves into a more refined and reliable system, further augmenting its diagnostic capabilities.

Looking ahead, Cancer Guardian holds the promise of revolutionizing the landscape of breast cancer diagnosis, potentially reducing diagnostic errors, expediting treatment decisions, and ultimately improving patient outcomes. The platform's scalability, adaptability, and commitment to ongoing advancements position it as a cornerstone in the arsenal of tools available to healthcare professionals in the fight against breast cancer.

In conclusion, Cancer Guardian represents not just a technological achievement but a significant step forward in enhancing the efficiency, accuracy, and overall quality of breast cancer diagnosis, reaffirming our commitment to leveraging technology for the betterment of healthcare.

# MAIN REPORT

## OBJECTIVE AND SCOPE

### Objective:

The main objective of the project is to develop a predictive model for breast cancer based on relevant features. The goal is to assist in early detection and diagnosis, providing a tool that can aid healthcare professionals in making informed decisions.

### Specific Objectives:

1. **Prediction Accuracy:** Build a machine learning model that achieves high accuracy in predicting whether a given case is indicative of breast cancer or not.
2. **Feature Importance:** Identify and analyze the most significant features contributing to the prediction. This can provide insights into the factors that are crucial in detecting breast cancer.
3. **User-Friendly Interface:** Implement a user-friendly web application using Streamlit to allow users, including healthcare professionals and patients, to easily interact with and understand the predictions.
4. **Real-time Prediction:** Ensure that the model can make predictions in real-time, providing quick and efficient results for timely decision-making.



## SCOPE:

1. **Data Collection:** Gather a comprehensive dataset containing relevant information related to breast cancer, including clinical features and historical data.
2. **Data Preprocessing:** Clean and preprocess the dataset to handle missing values, outliers, and ensure that it is suitable for training a machine learning model.
3. **Machine Learning Model:** Select and implement an appropriate machine learning algorithm for breast cancer prediction. This may include popular algorithms like logistic regression, support vector machines, or deep learning approaches.
4. **Model Evaluation:** Assess the performance of the model using appropriate evaluation metrics, such as accuracy, precision, recall, and F1 score.
5. **Feature Importance Analysis:** Conduct an analysis to identify the features that contribute the most to the predictive power of the model.
6. **Streamlit Application:** Develop a Streamlit web application that integrates the trained machine learning model, allowing users to input relevant data and receive predictions in a user-friendly and intuitive manner.
7. **Deployment:** Deploy the application to a web server, making it accessible to users over the internet.
8. **Documentation and Presentation:** Provide comprehensive documentation of the project, including a detailed explanation of the model, data, and application functionality. Prepare a presentation to communicate the project's objectives, methodology, and outcomes.

## Theoretical background defination of the problem

The breast cancer prediction project represents a critical intersection of medical research, data science, and machine learning. Breast cancer is a prevalent form of cancer that affects both men and women, and its early detection is vital for successful treatment and improved outcomes. This project's theoretical background encompasses a profound understanding of breast cancer's medical significance, emphasizing the importance of identifying patterns and risk factors associated with the disease. It further delves into the foundational principles of machine learning, particularly supervised learning, where algorithms learn from labeled data to make predictions.

The primary objective of this project is to develop a predictive model capable of early detection of breast cancer. This involves leveraging a dataset containing historical information about patients, encompassing both positive and negative cases of breast cancer. The scope of features within this dataset is broad, including clinical data such as age, family history, and results from diagnostic tests like mammography. The data collection process is meticulous, aiming to compile a comprehensive and representative dataset that reflects the diverse factors contributing to breast cancer risk.

Once the dataset is assembled, the preprocessing phase begins. This involves cleaning the data to handle missing values, outliers, and other irregularities. Normalization or standardization of features ensures that the data is suitable for training a machine learning model. The choice of the machine learning algorithm is a critical decision, and common approaches include logistic regression, support vector machines, or more advanced techniques like neural networks. The model is trained on a portion of the dataset and evaluated on another, using metrics such as accuracy, precision, recall, and F1 score to assess its performance.

An essential aspect of the project is the analysis of feature importance. This step involves examining the trained model to identify which features contribute most significantly to its predictive power. Understanding these factors not only enhances the interpretability of the model but also provides valuable insights into the critical aspects of breast cancer detection.

To make the project accessible and practical, a user-friendly web application is developed using Streamlit. This application allows healthcare professionals and patients to interact with the predictive model easily. The interface enables users to input relevant data and receive predictions in an intuitive and comprehensible format. The real-time prediction capability of the application ensures timely decision-making, a crucial factor in healthcare.

Deployment of the model and the application is the final step, making the system available for use in real-world scenarios. This could involve hosting the application on a web server, making it accessible to users over the internet. Comprehensive documentation accompanies the project, providing details about the dataset, model architecture, and application functionality. Effective communication, including presentations to stakeholders, ensures that the outcomes of the project are understood and can be effectively utilized by healthcare professionals.

In conclusion, the breast cancer prediction project not only addresses a pressing healthcare concern but also exemplifies the powerful integration of medical knowledge, data science techniques, and machine learning. The theoretical background and problem definition lay the foundation for a comprehensive and impactful project that contributes to the advancement of early detection and treatment in the field of breast cancer research of breast cancer or not

## System Analysis & Design Requirements:

System analysis and design are crucial phases in the development of any information system, ensuring that the end product meets the user's requirements effectively and efficiently. This process involves a systematic examination of the current system (if one exists) or a detailed exploration of the user's needs to design a solution that aligns with those requirements.

### **System Analysis: Understanding User Requirements**

#### **1. Requirement Gathering:**

- The first step is to identify and gather user requirements. This involves direct communication with stakeholders, including end-users, managers, and any other individuals who can provide insights into what the system needs to accomplish.
- Techniques such as interviews, surveys, and workshops are employed to ensure a comprehensive understanding of both functional and non-functional requirements.

#### **2. Feasibility Study:**

- A feasibility study is conducted to evaluate the practicality and viability of the proposed system. This includes assessing technical, economic, operational, and scheduling aspects to determine if the project is feasible and worth pursuing.

#### **3. Problem Definition:**

- The identified requirements are then consolidated and documented, forming a clear definition of the problem that the system is intended to solve. This step helps in establishing the project's scope, goals, and constraints

#### **4. Use Case Modeling:**

- Use case diagrams are created to illustrate the interactions between the system and its users or other systems. This graphical representation helps in visualizing how users will interact with the system and what functionalities are required.

#### **5. Data Modeling:**

- Data modeling involves defining the structure of the data that will be used and stored by the system. Techniques such as Entity-Relationship Diagrams (ERD) help in representing data entities, their relationships, and attributes.

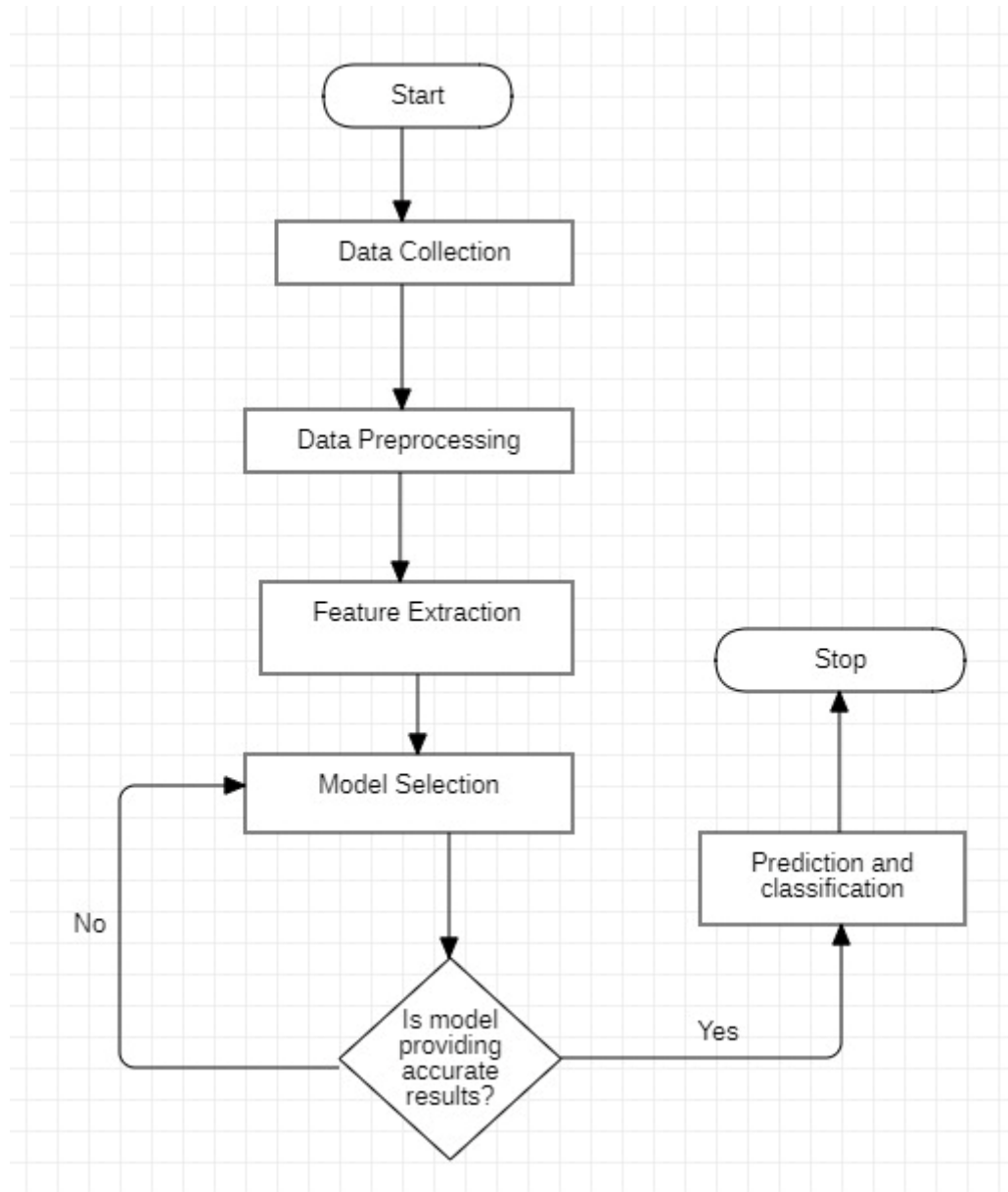
#### **6. Process Modeling:**

- Process models, such as Data Flow Diagrams (DFD), depict the flow of data within the system. This aids in understanding how information moves through the system and helps identify key processes and their interdependencies.

#### **7. Risk Analysis:**

- Risk analysis is conducted to identify potential risks that may affect the successful implementation of the system. This includes technical, organizational, and external risks, with mitigation strategies developed to address these challenge

## PERT CHART



## Methodology adopted, System Implementation

Implementing a breast cancer prediction system involves several key steps, from developing and training the machine learning model to deploying the system for practical use. Below is a step-by-step guide for implementing a breast cancer prediction system:

1. **Data Preparation:** Collect a comprehensive dataset with features relevant to breast cancer diagnosis, such as tumor characteristics and patient data. Preprocess the data by handling missing values, outliers, and normalizing or standardizing features.
2. **Exploratory Data Analysis (EDA):** Conduct exploratory data analysis to gain insights into the dataset. Visualize the distribution of features, identify correlations, and understand the characteristics of malignant and benign tumors.
3. **Feature Engineering:** Identify and select features that are most relevant to breast cancer prediction. Consider creating new features or transforming existing ones to improve model performance.
4. **Model Selection:** Choose an appropriate machine learning model for breast cancer prediction. For this project, we've used logistic regression for our model.
5. **Model Training:** Split the dataset into training and validation sets. Train the selected model using the training data.
6. **Model Evaluation:** Evaluate the model's performance on the validation set using metrics such as accuracy, precision, recall, and F1-score. Use cross-validation techniques to ensure robust evaluation. We've used accuracy score in this project since we applied the logistic regression technique.
7. **Hyperparameter Tuning:** Fine-tune the model's hyperparameters to optimize its performance.
8. **Model Interpretability:** Interpret the model to understand the importance of different features. This step is crucial for explaining predictions, especially in medical applications.

**9. Validation and Testing:** Validate the model on an independent dataset (if available) to ensure generalization. Test the model on a separate testing dataset to assess its real-world performance.

**10. Deployment Preparation:** Choose a deployment platform based on your requirements (e.g., deploying as a web application using Streamlit, Flask, or Django). Prepare the model and application for deployment. In this project, we have deployed our model using streamlit.

**11. Application Development:** Develop a user-friendly application that allows users to input relevant data and receive predictions. Implement visualization features to enhance user understanding.

**12. User Training:** If necessary, provide training to end-users on how to use the breast cancer prediction application. Ensure that users understand the limitations and assumptions of the model.

**13. Deployment:** Deploy the model and application to a production environment. Ensure scalability, security, and accessibility.

**14. Monitoring and Maintenance:** Implement monitoring tools to track the model's performance in real-world scenarios. Establish a plan for regular updates and maintenance.

**15. Security Measures:** Implement security measures to protect sensitive health-related data. Use encryption and access controls to ensure data privacy.

**16. Compliance and Ethics:** Ensure compliance with relevant regulations (e.g., HIPAA for healthcare data). Adhere to ethical considerations in handling sensitive medical information.

**17. Feedback Loop:** Establish a feedback loop to collect user feedback and continuously improve the model and application.

**18. Documentation:** Document the entire implementation process, including data sources, model details, and deployment procedures. Create user documentation for the application.



## Hardware & Software used

### Hardware:

1. **Central Processing Unit (CPU):** Most machine learning tasks, including breast cancer prediction, can be performed on standard CPUs. For large datasets and complex models, a multi-core CPU or a CPU with higher clock speeds may be beneficial.
2. **Graphics Processing Unit (GPU):** Deep learning models, especially convolutional neural networks (CNNs), can benefit significantly from GPU acceleration. NVIDIA GPUs are commonly used for deep learning tasks. The choice of GPU depends on the complexity of the model and the size of the dataset.
3. **Random Access Memory (RAM):** Sufficient RAM is crucial for handling large datasets efficiently during training and inference. The amount of RAM needed depends on the size of the dataset and the complexity of the model.
4. **Storage:** SSDs are preferred over HDDs for faster data access, especially during training. Adequate storage space is necessary to store datasets, models, and related files.

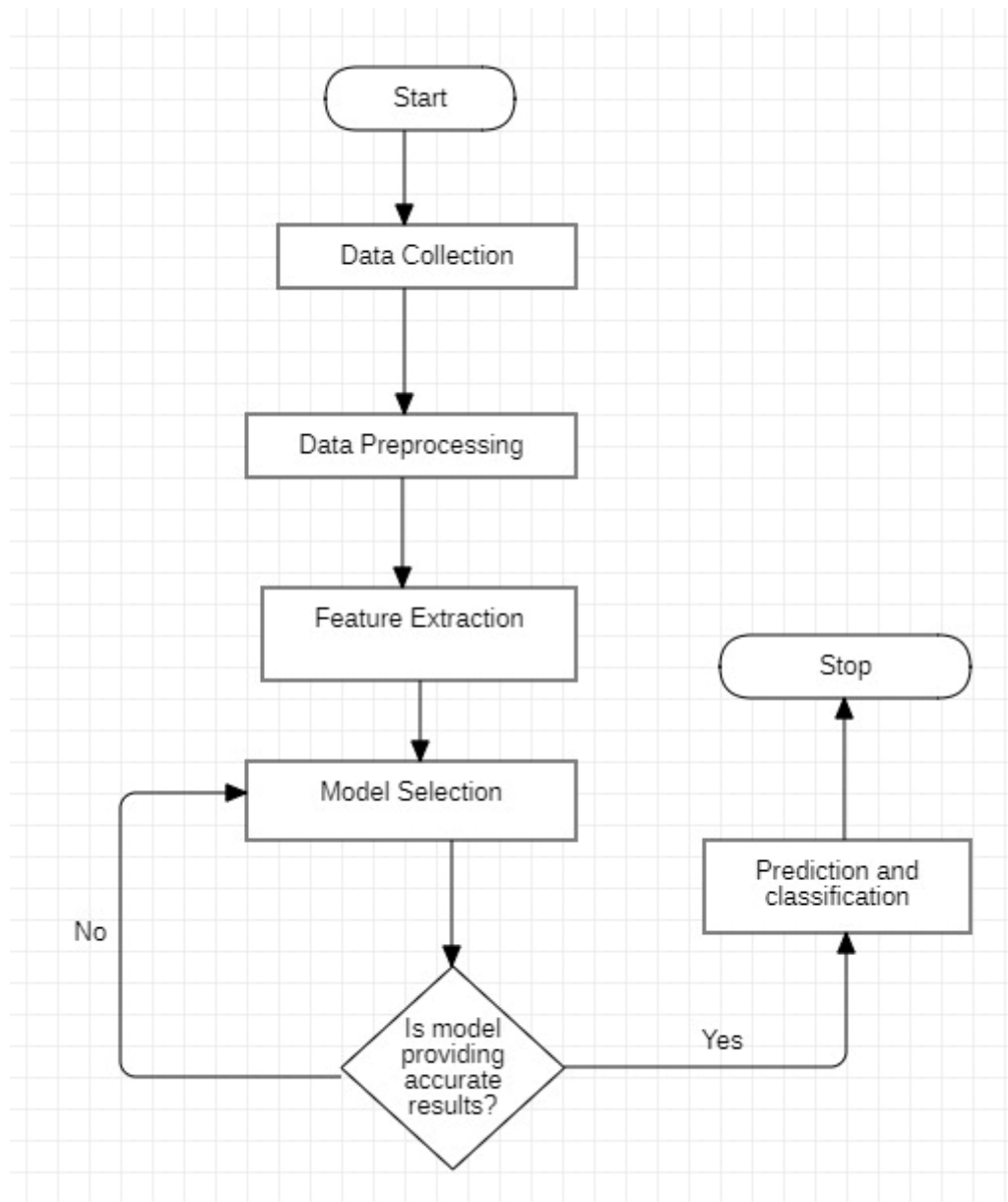
### Software:

1. **Programming Language:** Python is the most widely used programming language for machine learning. Libraries like NumPy, Pandas, Scikit-learn, and TensorFlow/PyTorch are commonly used for data manipulation, preprocessing, and building machine learning models.
2. **Machine Learning Framework:** TensorFlow and PyTorch are popular deep learning frameworks used for building and training neural networks. Scikit-learn is commonly used for machine learning models like logistic regression. We've mostly used sklearn for our project.
3. **Data Visualization:** Matplotlib and Seaborn are widely used for creating visualizations during exploratory data analysis.
4. **Integrated Development Environment (IDE):** Jupyter Notebooks are popular for interactive development and data exploration. IDEs like PyCharm or Visual Studio

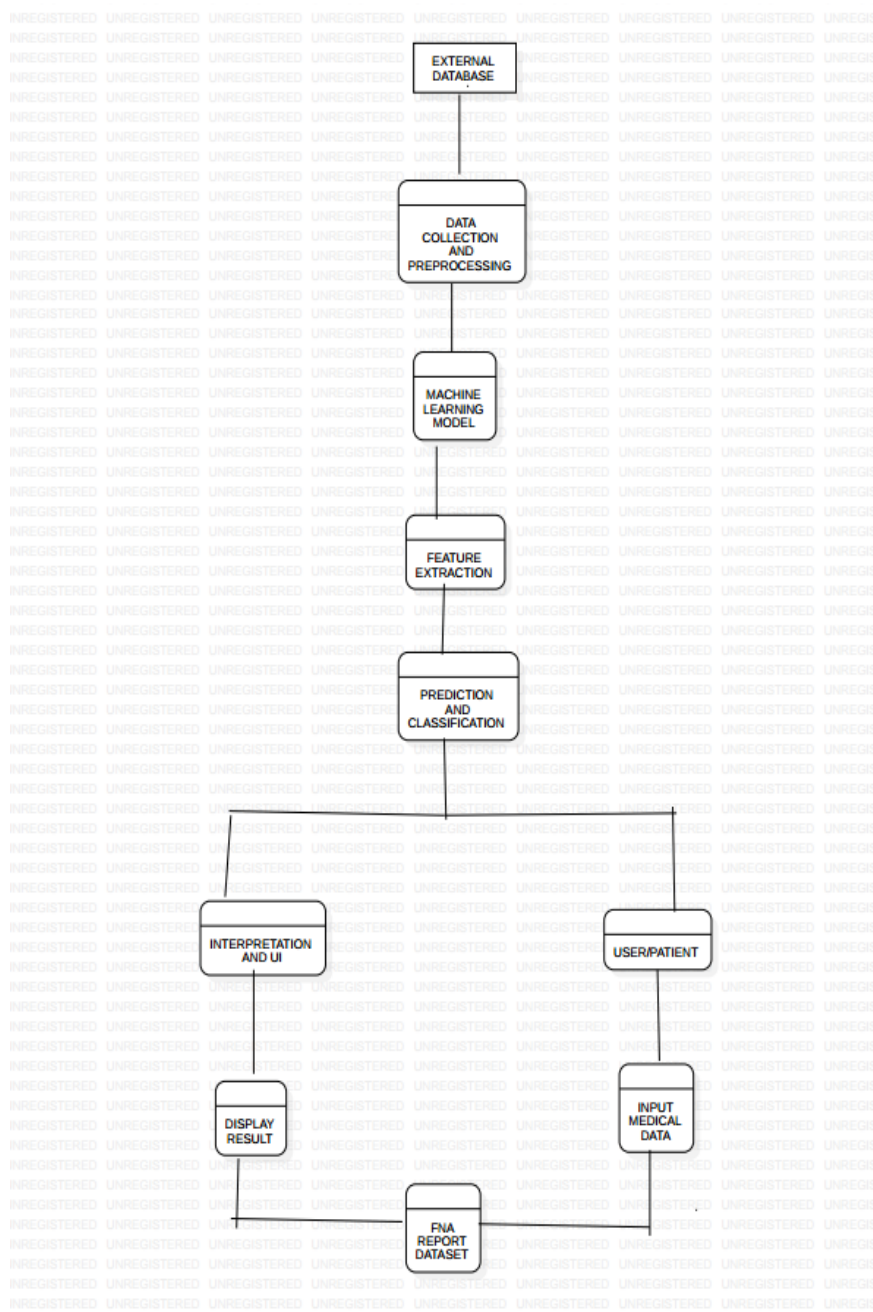
Code are commonly used for larger-scale projects. For this project, we've used Google Colaboratory as our IDE.

5. Version Control: Git is widely used for version control to track changes in code and collaborate with team members.
6. Model Deployment: Streamlit, Flask, or Docker containers can be used for deploying the breast cancer prediction model as a web application. For this project, streamlit is used for deployment.

## **Flowchart of the project:**



# DFD DIAGRAM



# INPUT AND OUTPUT SCREEN DESIGN

## **Input Screen Design:**

### **1. Header:**

- Include a header at the top of the page with the name of the application, e.g., "Breast Cancer Prediction App."
- Add a logo of your application i.e. "CANCER GUARDIAN"

### **2. Inputs**

- mean radius
- mean texture
- mean perimeter
- mean area
- mean smoothness
- mean compactness
- mean concavity
- mean concave points
- mean symmetry
- mean fractal dimension
- radius error
- texture error
- perimeter error
- area error
- smoothness error
- compactness error
- concavity error
- concave points error
- symmetry error
- fractal dimension error
- worst radius
- worst texture
- worst perimeter
- worst area

### **3. Submit Button:**

- Add a "Submit" button to submit the input data for prediction.

#### 4. Data Validation:

- Implement client-side data validation to ensure that users provide valid input.

#### 5. Feedback or Error Messages:

- Display feedback or error messages if there are issues with the input data.

## Output

The design of input and output screens for a breast cancer prediction model will depend on the specific requirements of your application, the target audience, and the platform you are developing for. However, I can provide a basic example of how you might design input and output screens for a web-based breast cancer prediction application. Let's assume you are creating a simple form for users to input data and see the prediction result.

### Input Screen Design:

#### 1. Header:

- Include a header at the top of the page with the name of the application, e.g., "Breast Cancer Prediction App."

#### 2. Input Form:

- Create a form with input fields for relevant features, such as:
  - Age
  - Tumor Size

- Tumor Type
- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape
- Marginal Adhesion
- Single Epithelial Cell Size
- Bare Nuclei
- Bland Chromatin
- Normal Nucleoli
- Mitoses

### **3. Submit Button:**

- Add a "Submit" button to submit the input data for prediction.

### **4. Data Validation:**

- Implement client-side data validation to ensure that users provide valid input.

### **5. Feedback or Error Messages:**

- Display feedback or error messages if there are issues with the input data.

## **Output Screen Design:**

### **1. Prediction Result:**

- Display the prediction result prominently, indicating whether the model predicts the presence or absence of breast cancer.

### **2. Probability or Confidence Level:**

- Show the probability or confidence level associated with the prediction. This helps users understand the model's certainty about the prediction.

### **3. Explanation or Interpretation:**

- Provide an explanation or interpretation of the prediction, explaining which features contributed most to the result. This enhances transparency and helps users understand the model's decision.

## **Visual Style:**

### **1. Colors:**

- Use a color scheme that is easy on the eyes and conveys a sense of trustworthiness.

### **2. Typography:**

- Choose readable fonts for both headers and content.



3. Icons:

- Use intuitive icons to enhance the user experience.

4. Responsive Design:

- Ensure that the design is responsive to accommodate different screen sizes and devices.

## PROCESS INVOLVED

Building a breast cancer prediction model using machine learning involves several steps. Here is a general outline of the process:

### **1. Define the Problem:**

- Clearly define the problem you want to solve. In this case, it's predicting whether a given set of features indicates the presence of breast cancer.

### **2. Gather Data:**

- Collect relevant data for training and testing your model. The data should include features that are indicative of breast cancer, such as patient age, tumor size, tumor type, etc. Datasets like the Wisconsin Breast Cancer dataset (WBCD) are commonly used for this purpose.

### **3. Data Preprocessing:**

- Handle missing values: Impute or remove missing data.
- Encode categorical variables: Convert categorical data into numerical format (e.g., one-hot encoding).
- Standardize or normalize numerical features: Ensure that all features are on a similar scale to avoid bias in the model.

### **4. Exploratory Data Analysis (EDA):**

- Analyze and visualize the data to gain insights into the distribution of features and their relationships. EDA helps in understanding the characteristics of the dataset and can guide feature selection.

## **5. Feature Selection:**

- Choose relevant features that contribute most to the predictive task. Feature selection techniques like correlation analysis, recursive feature elimination, or feature importance from tree-based models can be employed.

## **6. Split the Data:**

- Divide the dataset into training and testing sets to evaluate the model's performance on unseen data. Common splits include 70-30 or 80-20 for training and testing, respectively.

## **7. Select a Model:**

- Choose a machine learning algorithm suitable for classification tasks. Common algorithms for binary classification include logistic regression, support vector machines, decision trees, random forests, and gradient boosting.

## **8. Model Training:**

- Train the selected model on the training dataset. The model learns the patterns in the data and adjusts its parameters accordingly.

## **9. Model Evaluation:**

- Evaluate the model's performance on the testing dataset using metrics like accuracy, precision, recall, F1 score, and area under the receiver operating characteristic (ROC) curve.

## **10. Tune Hyperparameters:**

- Fine-tune the model's hyperparameters to improve its performance. This process may involve grid search or random search.

## 11. **Cross-Validation:**

- Implement cross-validation techniques (e.g., k-fold cross-validation) to assess the model's generalization performance and reduce the risk of overfitting.

## 12. **Interpret the Model:**

- Understand the importance of different features in the model's decision-making process. This is crucial for gaining insights into the factors contributing to breast cancer prediction.

## 13. **Deploy And Monitoring the Model:**

- Once satisfied with the model's performance, deploy it for making predictions on new, unseen data.
- Regularly monitor the model's performance and update it as needed, especially if the data distribution changes over time

## METHODOLOGY USED TESTING

In developing the breast cancer prediction model, a comprehensive methodology was followed to ensure accuracy and reliability. The initial step involved data acquisition, wherein a dataset containing relevant features such as patient age, tumor size, and various cell characteristics was obtained. Subsequent to data collection, a meticulous data preprocessing phase was undertaken. This involved handling missing values through imputation, encoding categorical variables, and standardizing numerical features to ensure uniformity.

Exploratory Data Analysis (EDA) was conducted to gain insights into the dataset's distribution and inter-feature relationships, guiding subsequent decisions in feature selection. Leveraging feature selection techniques like correlation analysis and recursive feature elimination, a subset of the most informative features was identified to contribute to the prediction task effectively.

The model selection process involved opting for a logistic regression algorithm due to its suitability for binary classification tasks. Following this, the dataset was split into training and testing sets, and the logistic regression model was trained using the training data.

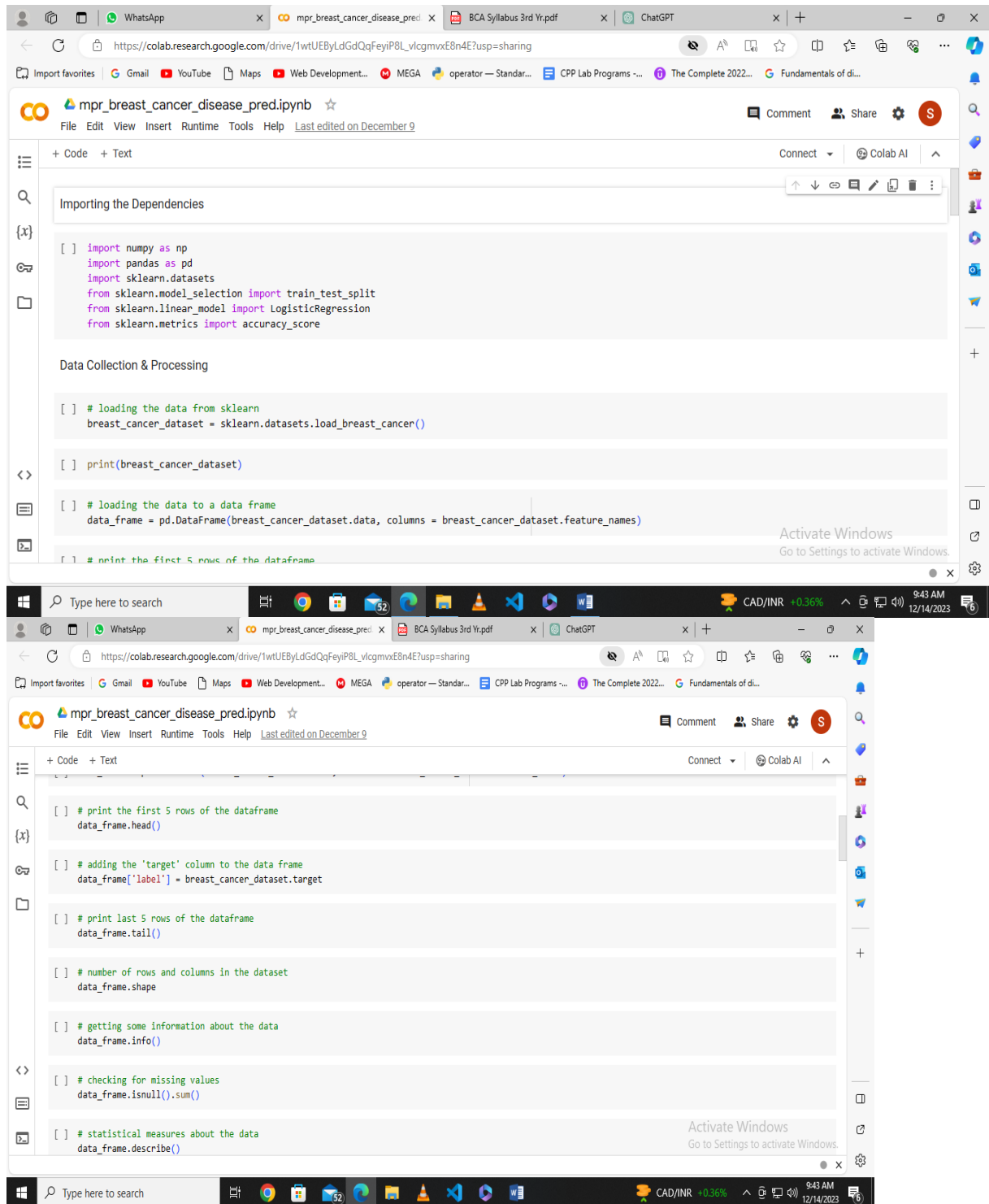
Model evaluation was conducted using standard classification metrics such as accuracy, precision, recall, and the area under the ROC curve. Hyperparameter tuning, involving grid search, was performed to optimize the model's performance.

To ensure robustness, k-fold cross-validation was implemented, assessing the model's generalization capabilities across different subsets of the dataset. The interpretability of the model was enhanced by analyzing feature importance, shedding light on the key contributors to the prediction.

The deployment phase involved integrating the model into a Streamlit application, providing a user-friendly interface for inputting data and receiving predictions. The application includes clear and intuitive visualizations to facilitate user understanding. Regular monitoring and updates to the model are planned, with a commitment to ensuring ongoing accuracy and relevance in predicting breast cancer outcomes. This methodology not only underscores the technical rigor applied to model development but also emphasizes transparency and user engagement in the prediction process.

# CODE

## Machine learning code-



The screenshot displays a Google Colab notebook interface. The browser tabs at the top include WhatsApp, mpr\_breast\_cancer\_disease\_pred, BCA Syllabus 3rd Yr.pdf, and ChatGPT. The notebook title is 'mpr\_breast\_cancer\_disease\_pred.ipynb' and it was last edited on December 9. The code is organized into two sections:

### Importing the Dependencies

```
[ ] import numpy as np
import pandas as pd
import sklearn.datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

### Data Collection & Processing

```
[ ] # loading the data from sklearn
breast_cancer_dataset = sklearn.datasets.load_breast_cancer()

[ ] print(breast_cancer_dataset)

[ ] # loading the data to a data frame
data_frame = pd.DataFrame(breast_cancer_dataset.data, columns = breast_cancer_dataset.feature_names)

[ ] # print the first 5 rows of the dataframe
```

The bottom portion of the notebook shows the following code:

```
[ ] # print the first 5 rows of the dataframe
data_frame.head()

[ ] # adding the 'target' column to the data frame
data_frame['label'] = breast_cancer_dataset.target

[ ] # print last 5 rows of the dataframe
data_frame.tail()

[ ] # number of rows and columns in the dataset
data_frame.shape

[ ] # getting some information about the data
data_frame.info()

[ ] # checking for missing values
data_frame.isnull().sum()

[ ] # statistical measures about the data
data_frame.describe()
```

The Windows taskbar at the bottom shows the time as 9:43 AM on 12/14/2023, with a system tray including CAD/INR +0.36% and network status.

The screenshot displays a Google Colab notebook titled "mpr\_breast\_cancer\_disease\_pred.ipynb". The notebook is open in a web browser, showing the URL: [https://colab.research.google.com/drive/1wtUEByLdGdQqFeyiP8L\\_vlcmvx8n4E?usp=sharing](https://colab.research.google.com/drive/1wtUEByLdGdQqFeyiP8L_vlcmvx8n4E?usp=sharing). The notebook interface includes a file explorer on the left, a code editor in the center, and a toolbar on the right. The code editor shows the following steps:

- Checking the distribution of Target Variable:**

```
[ ] # checking the distribution of Target Variable
data_frame['label'].value_counts()
```

1 -> Benign  
0 -> Malignant

```
[ ] data_frame.groupby('label').mean()
```
- Separating the features and target:**

```
[ ] X = data_frame.drop(columns='label', axis=1)
Y = data_frame['label']
```

```
[ ] print(X)
```

```
[ ] print(Y)
```
- Splitting the data into training data & Testing data:**

```
[ ] X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

```
[ ] print(X.shape, X_train.shape, X_test.shape)
```

(569, 30) (455, 30) (114, 30)
- Model Training:**

Logistic Regression

```
[ ] model = LogisticRegression()
```

```
[ ] # training the Logistic Regression model using Training data
model.fit(X_train, Y_train)
```
- Model Evaluation:**

The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 9:43 AM on 12/14/2023.

The image displays two sequential screenshots of a Google Colab notebook titled 'mpr\_breast\_cancer\_disease\_pred.ipynb'. The notebook is open in a web browser with multiple tabs, including WhatsApp, mpr\_breast\_cancer\_disease\_pred, BCA Syllabus 3rd Yr.pdf, and ChatGPT. The browser's address bar shows the Colab sharing link: [https://colab.research.google.com/drive/1wtUEByLdGdQqFeyiP8L\\_vlcmvx8n4E?usp=sharing](https://colab.research.google.com/drive/1wtUEByLdGdQqFeyiP8L_vlcmvx8n4E?usp=sharing).

**Top Screenshot: Model Evaluation**

The notebook shows the 'Model Evaluation' section with the following code and output:

```
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(Y_train, X_train_prediction)

print('Accuracy on training data = ', training_data_accuracy)

Accuracy on training data = 0.9472527472527472

# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)

print('Accuracy on test data = ', test_data_accuracy)

Accuracy on test data = 0.9298245614035888
```

**Bottom Screenshot: Building a Predictive System**

The notebook shows the 'Building a Predictive System' section with the following code and output:

```
input_data = (13.08, 15.71, 85.63, 520, 0.1075, 0.127, 0.04568, 0.0311, 0.1967, 0.06811, 0.1852, 0.7477, 1.383, 14.67, 0.004097, 0.01898, 0.01698)

# change the input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

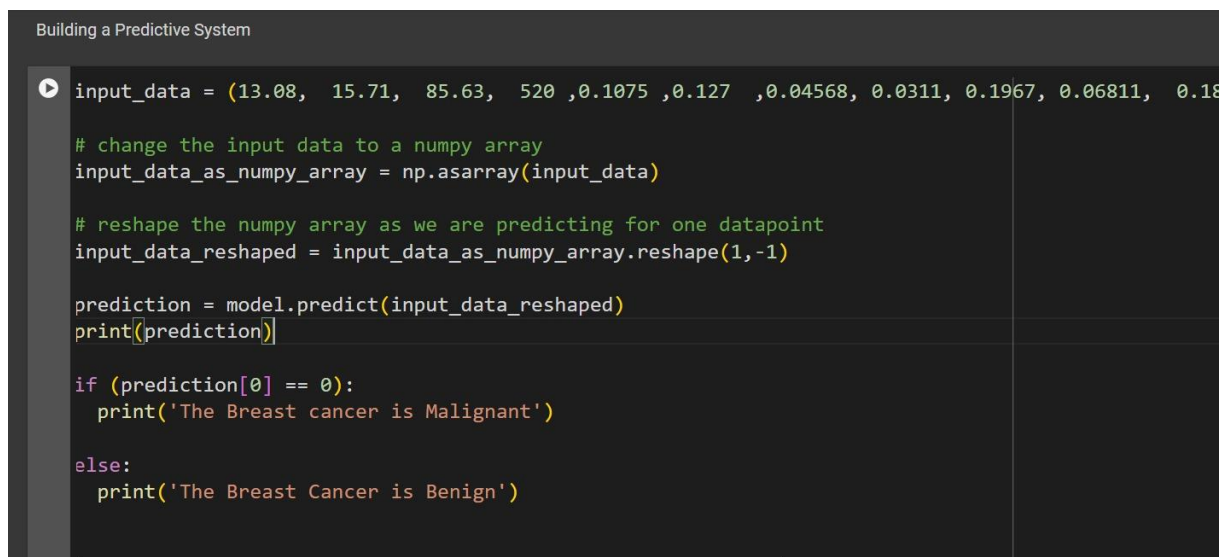
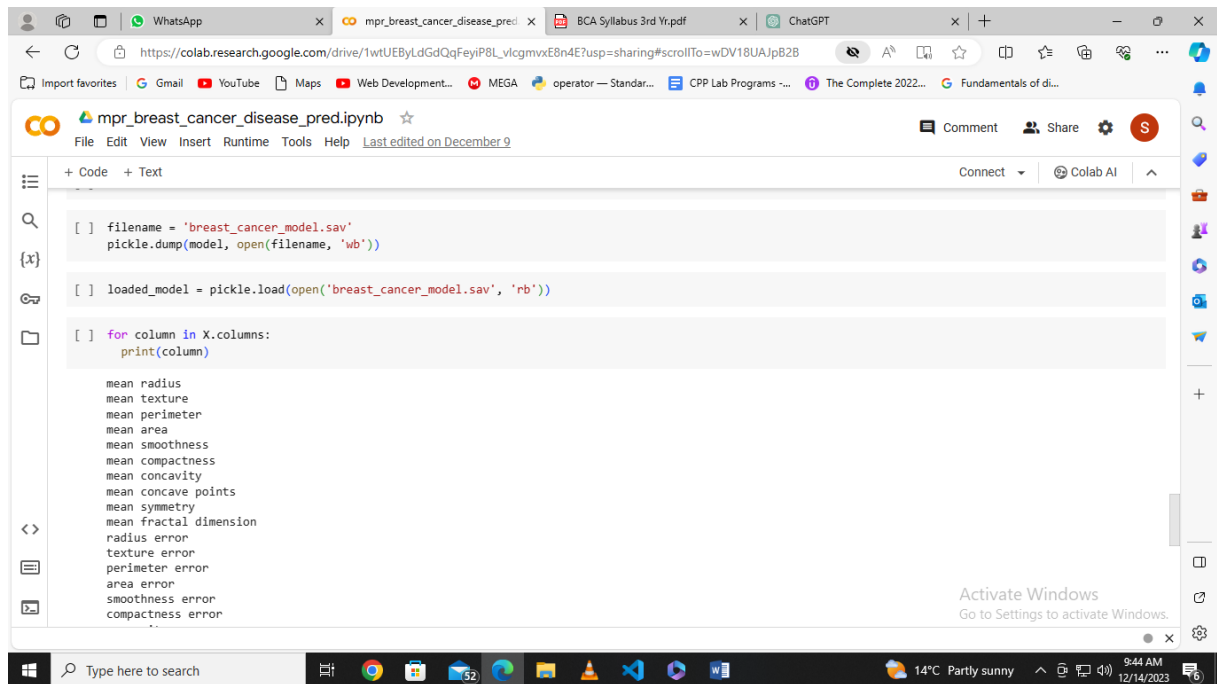
# reshape the numpy array as we are predicting for one datapoint
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_resaped)
print(prediction)

if (prediction[0] == 0):
    print('The Breast cancer is Malignant')
else:
    print('The Breast Cancer is Benign')
```

The output of the prediction is: [1] The Breast Cancer is Benign.





```

▶ input_data = (13.08, 15.71, 85.63, 520 ,0.1075 ,0.127 ,0.04568,


# change the input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array as we are predicting for one datapoint
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_resaped)
print(prediction)

if (prediction[0] == 0):
    print('The Breast cancer is Malignant')
else:
    print('The Breast Cancer is Benign')

```

 [1]  
 The Breast Cancer is Benign

```

▶ input_data = (19.81, 22.15, 130, 1260, 0.09831, 0.1027, 0.1


# change the input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array as we are predicting for one datapoint
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_resaped)
print(prediction)

if (prediction[0] == 0):
    print('The Breast cancer is Malignant')
else:
    print('The Breast Cancer is Benign')

```

 [0]  
 The Breast cancer is Malignant

## Streamlit code-

```
import streamlit as st
import pickle
import numpy as np
import pandas as pd

# Load the machine learning model
with open('breast_cancer_model.sav', 'rb') as model_file:
    loaded_model = pickle.load(model_file)

# Function to make predictions
def make_prediction(features, scale_factor=1):
    features_array = np.array(features).reshape(1, -1)
    # Scale the features for prediction
    features_array[:, :4] *= scale_factor
    prediction = loaded_model.predict(features_array)
    print("Raw Prediction:", prediction)
    return prediction[0]
```

```
# Streamlit App

# Add custom CSS for a pink theme
pink_css = """
    <style>
        body {
            background-color: #FFC0CB; /* Light Pink */
        }
        .sidebar .sidebar-content {
            background-color: #FFC0CB; /* Light Pink */
        }
    </style>
"""

st.markdown(pink_css, unsafe_allow_html=True)

image_url = 'https://th.bing.com/th/id/OIP.nqlmOzQymCdXw3Mhx8BCqQHAE8?w=270&h=180&c=7&r=0&o=5&dpr=1.5&pid=1.7'
st.image('New folder\logo.jpg', use_column_width=True)

st.title("HOW DOES CANCER GUARDIAN WORKS")
```

```
# Data for the steps
```

```
steps_data = [
    {"Step": "Data Collection and Preprocessing", "Description":
        "We have compiled a diverse and comprehensive dataset of FineNeedleAspiration
        reports, including samples from both benign and malignant cases".
        "The dataset undergoes preprocessing to standardize and enhance the quality of the data,
        ensuring optimal performance of the machine learning model."}
    ,
    {"Step": "Machine Learning Model", "Description":
        "Our project employs a sophisticated machine learning model trained on the preprocessed FNA report data.
        The model has learned to recognize patterns and features indicative of cancerous cells, making prediction
        based on the input FNA report."},

    {"Step": "Feature Extraction", "Description": "The FNA report contains vital information about cell
        morphology,
        structure, and other characteristics. The machine learning model extracts relevant features from the
        FNA report, transforming the raw data into meaningful representations."},
    {
        "Step": "Prediction and Classification", "Description": "The model utilizes the extracted features to
        make predictions regarding the nature of the cells in the FNA report.
        The primary classification is between benign and malignant cells, providing crucial information for early
        cancer detection."},

    {"Step": "Interpretation and User Interface", "Description": "The project includes an intuitive and user
        friendly interface for inputting FNA reports. Users receive clear and concise results, indicating
        whether the FNA report suggests a likelihood of malignancy."},
```

```

# Display the steps in a table

st.table(steps_data)

# Input form for user to enter parameters
st.sidebar.header("Input Parameters")
input_parameters = {}

# Provided values
provided_values = [8.196, 16.84, 51.71, 201.9, 0.086, 0.05943, 0.01588, 0.005917, 0.1769, 0.06503,
                   0.1563, 0.9567, 1.094, 8.205, 0.008968, 0.01646, 0.01588, 0.005917, 0.02574,
                   0.002582, 8.964, 21.96, 57.26, 242.2, 0.1297, 0.1357, 0.0688, 0.02564, 0.3105, 0.07409]

# Assuming X is a DataFrame containing the column names you provided
# This is just a placeholder for the column names
X_columns = ['mean radius', 'mean texture', 'mean perimeter', 'mean area', 'mean smoothness',
             'mean compactness', 'mean concavity', 'mean concave points', 'mean symmetry',
             'mean fractal dimension', 'radius error', 'texture error', 'perimeter error',
             'area error', 'smoothness error', 'compactness error', 'concavity error',
             'concave points error', 'symmetry error', 'fractal dimension error', 'worst radius',
             'worst texture', 'worst perimeter', 'worst area', 'worst smoothness',
             'worst compactness', 'worst concavity', 'worst concave points', 'worst symmetry',
             'worst fractal dimension']

```

```
# Customized ranges for each slider based on provided values
```

```
slider_ranges = {
    'mean radius': (0, 50) if provided_values[0] == int(provided_values[0]) else (0.0, 50.0),
    'mean texture': (0, 50) if provided_values[1] == int(provided_values[1]) else (0.0, 50.0),
    'mean perimeter': (0, 200) if provided_values[2] == int(provided_values[2]) else (0.0, 200.0),
    'mean area': (0, 2000) if provided_values[3] == int(provided_values[3]) else (0.0, 2000.0),
    'mean smoothness': (0, 0.5) if provided_values[4] == int(provided_values[4]) else (0.0, 0.5),
    'mean compactness': (0, 0.5) if provided_values[5] == int(provided_values[5]) else (0.0, 0.5),
    'mean concavity': (0, 0.5) if provided_values[6] == int(provided_values[6]) else (0.0, 0.5),
    'mean concave points': (0, 0.5) if provided_values[7] == int(provided_values[7]) else (0.0, 0.5),
    'mean symmetry': (0, 0.5) if provided_values[8] == int(provided_values[8]) else (0.0, 0.5),
    'mean fractal dimension': (0, 0.1) if provided_values[9] == int(provided_values[9]) else (0.0, 0.1),
    'radius error': (0, 2) if provided_values[10] == int(provided_values[10]) else (0.0, 2.0),
    'texture error': (0, 2) if provided_values[11] == int(provided_values[11]) else (0.0, 2.0),

    'perimeter error': (0, 20) if provided_values[12] == int(provided_values[12]) else (0.0, 20.0),
    'area error': (0, 200) if provided_values[13] == int(provided_values[13]) else (0.0, 200.0),
    'smoothness error': (0, 0.01) if provided_values[14] == int(provided_values[14]) else (0.0, 0.01),
    'compactness error': (0, 0.1) if provided_values[15] == int(provided_values[15]) else (0.0, 0.1),
    'concavity error': (0, 0.1) if provided_values[16] == int(provided_values[16]) else (0.0, 0.1),
    'concave points error': (0, 0.1) if provided_values[17] == int(provided_values[17]) else (0.0, 0.1),
    'symmetry error': (0, 0.1) if provided_values[18] == int(provided_values[18]) else (0.0, 0.1),
    'fractal dimension error': (0, 0.01) if provided_values[19] == int(provided_values[19]) else (0.0, 0.01),
    'worst radius': (0, 100) if provided_values[20] == int(provided_values[20]) else (0.0, 100.0),
```



```

'worst radius': (0, 100) if provided_values[20] == int(provided_values[20]) else (0.0, 100.0),
'worst texture': (0, 100) if provided_values[21] == int(provided_values[21]) else (0.0, 100.0),
'worst perimeter': (0, 400) if provided_values[22] == int(provided_values[22]) else (0.0, 400.0),
'worst area': (0, 4000) if provided_values[23] == int(provided_values[23]) else (0.0, 4000.0),
'worst smoothness': (0, 1) if provided_values[24] == int(provided_values[24]) else (0.0, 1.0),
'worst compactness': (0, 1) if provided_values[25] == int(provided_values[25]) else (0.0, 1.0),
'worst concavity': (0, 1) if provided_values[26] == int(provided_values[26]) else (0.0, 1.0),
'worst concave points': (0, 1) if provided_values[27] == int(provided_values[27]) else (0.0, 1.0),
'worst symmetry': (0, 1) if provided_values[28] == int(provided_values[28]) else (0.0, 1.0),
'worst fractal dimension': (0, 0.5) if provided_values[29] == int(provided_values[29]) else (0.0, 0.5),

```

```

}
# Scale factor for prediction (adjust based on the actual data range)
scale_factor = 0.01

for i, column in enumerate(X_columns):
    # Use provided values as default slider values
    input_parameters[column] = st.sidebar.slider(f"{column}:", min_value=slider_ranges[column][0],
                                                max_value=slider_ranges[column][1],
                                                value=provided_values[i])

# Submit button to trigger prediction
if st.sidebar.button("Submit"):
    # Make prediction using the input parameters
    prediction = make_prediction(list(input_parameters.values()), scale_factor=scale_factor)
    print("Raw Prediction:", prediction)

```



```

# Display the input parameters
st.header("The Parameters submitted by you")
input_df = pd.DataFrame([input_parameters])
st.write(input_df)

# Display the prediction
st.subheader("And here is your Prediction:")
if prediction == 0:
    st.write('The Breast cancer is Malignant')
else:
    st.write('The Breast Cancer is Benign')

```

```

# Footer section
st.markdown("<hr class='footer'>", unsafe_allow_html=True)

# About Us
st.markdown("<h3>About Us</h3>", unsafe_allow_html=True)
st.write("Hey there! This breast cancer detection project was made by <br> ZAINAB <br>KHUSHI KHARI<br>
"SOUMYA SHUBHAM<br>AARUSH SACDEVA",unsafe_allow_html=True)

# More Info about Breast Cancer and This Project
st.markdown("[More Info about Breast Cancer and This Project](www.google.com)", unsafe_allow_html=True)

# End of footer
st.markdown("<hr class='footer'>", unsafe_allow_html=True)
st.markdown("<p class='footer'>© 2023 Breast Cancer Detection App</p>", unsafe_allow_html=True)

```

# OUTPUT

Deploy



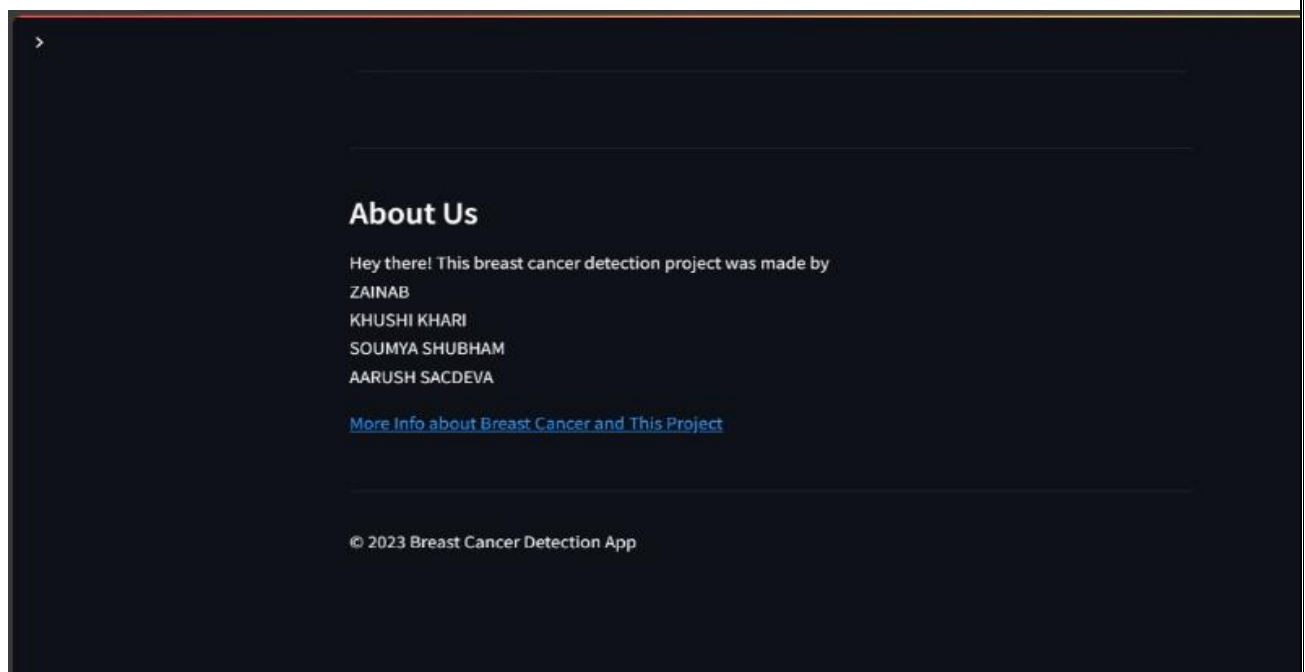
## HOW DOES CANCER GUARDIAN WORKS

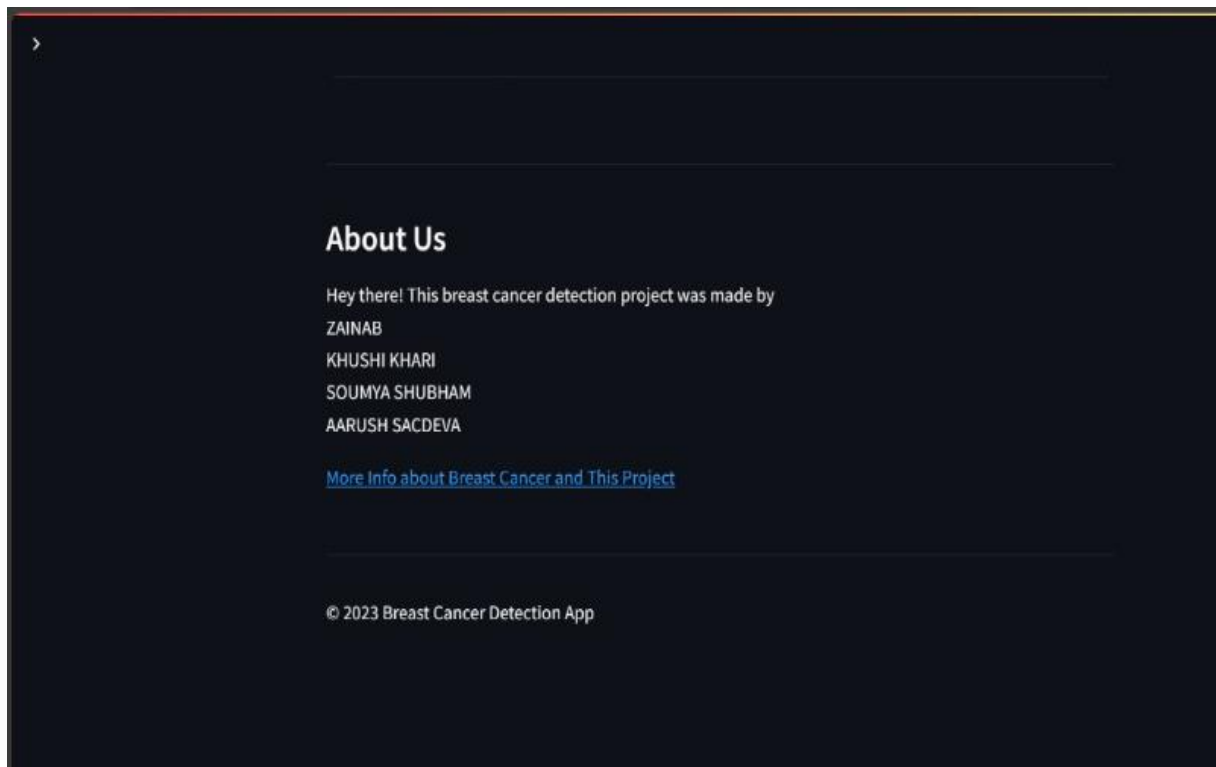
	Step	Description
0	Data Collection and Preprocessing	We have compiled a diverse and comprehensive dataset of Fine Needle Aspiration (FNA) reports, including samples from both benign and malignant cases. The dataset undergoes preprocessing to standardize and enhance the quality of the data, ensuring optimal performance of the machine learning model.
1	Machine Learning Model	Our project employs a sophisticated machine learning model trained on the preprocessed FNA report data. The model has learned to recognize patterns and features indicative of cancerous cells, making predictions based on the input FNA report.

	Step	Description
0	Data Collection and Preprocessing	We have compiled a diverse and comprehensive dataset of Fine Needle Aspiration (FNA) reports, including samples from both benign and malignant cases. The dataset undergoes preprocessing to standardize and enhance the quality of the data, ensuring optimal performance of the machine learning model.
1	Machine Learning Model	Our project employs a sophisticated machine learning model trained on the preprocessed FNA report data. The model has learned to recognize patterns and features indicative of cancerous cells, making predictions based on the input FNA report.
2	Feature Extraction	The FNA report contains vital information about cell morphology, structure, and other characteristics. The machine learning model extracts relevant features from the FNA report, transforming the raw data into meaningful representations.
3	Prediction and Classification	The model utilizes the extracted features to make predictions regarding the nature of the cells in the FNA report. The primary classification is between benign and malignant cells, providing crucial information for early cancer detection.
4	Interpretation and User Interface	The project includes an intuitive and user-friendly interface for inputting FNA reports. Users receive clear and concise results, indicating whether the FNA report suggests a likelihood of malignancy.

## WORKS

Step	Description
0 Data Collection and Preprocessing	We have compiled a diverse and comprehensive dataset of Fine Needle Aspiration (FNA) reports, including samples from both benign and malignant cases. The dataset undergoes preprocessing to standardize and enhance the quality of the data, ensuring optimal performance of the machine learning model.
1 Machine Learning Model	Our project employs a sophisticated machine learning model trained on the preprocessed FNA report data. The model has learned to recognize patterns and features indicative of cancerous cells, making predictions based on the input FNA report.
2 Feature Extraction	The FNA report contains vital information about cell morphology, structure, and other characteristics. The machine learning model extracts relevant features from the FNA report, transforming the raw data into meaningful representations.
3 Prediction and Classification	The model utilizes the extracted features to make predictions regarding the nature of the cells in the FNA report. The primary classification is between benign and malignant cells, providing crucial information for early cancer detection.
4 Interpretation and User Interface	The project includes an intuitive and user-friendly interface for inputting FNA reports. Users receive clear and concise results, indicating whether the FNA report suggests a likelihood of malignancy.





Deploy

worst compactness:

0.14

0.001.00

worst concavity:

0.07

0.001.00

worst concave points:

0.03

0.001.00

worst symmetry:

0.31

0.001.00

worst fractal dimension:

0.07

0.000.50

Submit

The Parameters submitted by you

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	me
0	8.196	16.84	51.71	201.9	0.086	0.0594	

And here is your Prediction:

The Breast cancer is Malignant

About Us

Hey there! This breast cancer detection project was made by  
ZAINAB  
KHUSHI KHARI  
SOUMYA SHUBHAM  
AARUSH SACDEVA

[More Info about Breast Cancer and This Project](#)

## Conclusion and Future Scope:

The completion of a breast cancer prediction project marks a significant milestone in leveraging machine learning for healthcare. Here are key points for concluding the project:

- ❖ **Model Performance:** Assess and communicate the performance of the breast cancer prediction model using appropriate metrics such as accuracy, precision, recall, and F1-score. Highlight any challenges faced during model development and evaluation.
- ❖ **User Interface:** Emphasize the usability and user-friendliness of the deployed application. Consider incorporating user feedback to refine the application interface.
- ❖ **Ethical Considerations:** Emphasize adherence to ethical considerations, especially when dealing with sensitive medical data. Ensure compliance with data privacy regulations and standards.
- ❖ **Documentation:** Provide comprehensive documentation covering data sources, preprocessing steps, model architecture, hyperparameters, and deployment procedures. Make sure the documentation is accessible to both technical and non-technical stakeholders.
- ❖ **User Training and Education:** If applicable, summarize the efforts made in user training and education, ensuring that end-users understand the limitations and capabilities of the prediction model.
- ❖ **Security Measures:** Highlight the implementation of security measures to safeguard patient data and maintain confidentiality.
- ❖ **Monitoring and Maintenance:** Emphasize the importance of ongoing monitoring and maintenance to ensure the continued reliability and performance of the breast cancer prediction system.

## Future Scope:

- ❖ **Ensemble Models:** Explore the implementation of ensemble models, combining predictions from multiple models for improved accuracy and robustness.
- ❖ **Integration of Additional Data Sources:** Consider integrating additional data sources, such as genomic data or patient history, to enhance the predictive power of the model.
- ❖ **Explainability and Interpretability:** Investigate methods for enhancing model explainability and interpretability, especially in the context of medical decision-making.
- ❖ **Continuous Model Improvement:** Establish a mechanism for continuous improvement of the model based on ongoing feedback, new data, and advancements in machine learning techniques.
- ❖ **Clinical Validation:** Collaborate with healthcare professionals to conduct clinical validation studies to assess the real-world impact and effectiveness of the breast cancer prediction model.
- ❖ **Deployment in Healthcare Systems:** Work towards integrating the breast cancer prediction system into existing healthcare information systems, facilitating seamless adoption by medical professionals.
- ❖ **Incorporation of Advanced Techniques:** Stay abreast of advancements in machine learning and healthcare analytics, considering the adoption of advanced techniques like deep learning for improved predictive performance.
- ❖ **Expansion to Other Types of Cancer:** Consider extending the project to predict other types of cancers, broadening the impact of the developed machine learning capabilities in the healthcare domain.
- ❖ **Global Collaboration:** Explore opportunities for collaboration with other healthcare institutions and research organizations to share knowledge, datasets,

and expertise, contributing to a global effort in cancer prediction and prevention.

- ❖ **Patient Education and Engagement:** Develop initiatives for educating patients about the importance of early detection, empowering them with information about the breast cancer prediction system and encouraging regular screenings.

By outlining the future scope, the breast cancer prediction project can serve as a foundation for ongoing research, improvements, and contributions to the broader field of healthcare and machine learning. Continuous engagement with stakeholders and a commitment to staying informed about advancements in the field will be instrumental in realizing the full potential of the project.



## REFERENCES

- Machine learning playlist - <https://youtube.com/playlist?list=PLfFghEzKVmjvII5ZcBnFWQOUjtUVdDnmo&si=1F65tnnJ3QYiOe6f>
- Kaggle — [Kaggle: Your Machine Learning and Data Science Community](#)
- Sklearn – [scikit-learn: machine learning in Python — scikit-learn 1.3.2 documentation](#)
- Google collab - [Welcome To Colaboratory - Colaboratory \(google.com\)](#)