Breast Cancer Prediction Web Application Using Machine Learning

By:

Ankita Banerjee (University Roll No. 10800121011)

Amit Kiran Deb (University Roll No. 10800121111)

Anindita Sircar (University Roll No. 10800121012)

Aniket Das (University Roll No. 10800122202)

Course: Final Year (7th Semester) B.Tech Course 2021-2025 Batch

Course Code: **PROJ-CS781**

Under the Supervision of:

Abhishek Bandyopadhyay (Head of the Department, AIML)

Department of Computer Science and Engineering

Asansol Engineering College

DEC 2024

Problem Statement and Objective

Problem Statement:

- Early detection of breast cancer is critical for improving survival rates. Traditional methods, such as Fine Needle Aspiration (FNA), often provide limited and inconclusive results, leading to delayed or incorrect diagnoses.

Objectives:

- Utilize machine learning algorithms for accurate prediction of breast cancer.
- Develop a user-friendly web application for healthcare professionals to input patient data and receive predictions.
- Provide explainable results to aid in decision-making.

Literature Survey

Existing Work:

- Studies have shown that supervised learning algorithms like SVM, Decision Trees, and Random Forests, Logistic Regression, Knn are effective for tumor classification.
- Limitations include difficulty in processing large datasets and lack of integration with web-based diagnostic tools.

Gap:

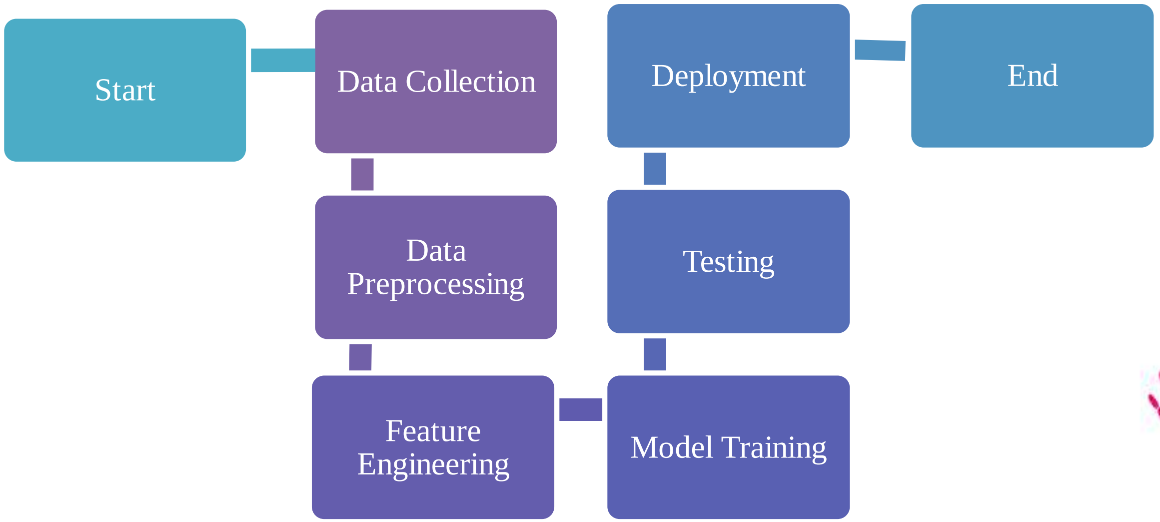
- Lack of accessible tools for real-time predictions integrated with secure databases.
- Limited exploration of web applications as an interface for ML models in breast cancer prediction.

Methodology – Workflow

Workflow Steps:

- **Data Collection:** Breast cancer datasets sourced from UCI ML Repository.
- **Preprocessing:** Cleaning, normalisation, and splitting data into training/testing sets.
- **Feature Engineering:** Selecting key features like mean radius and mean smoothness.
- **Model Training:** Implementing ML algorithms (SVM, Random Forest, Logistic Regression, Knn, Decision Tree).
- **Deployment:** A secure web application with a React.js frontend enables user-friendly input, prediction visualization, and SHAP-based explanations. The Flask backend processes inputs, integrates with the SVM model, and delivers real-time predictions, ensuring accessibility and transparency.

Workflow Diagram:



Why is preprocessing crucial in machine learning for healthcare applications?

We used the Breast Cancer Diagnostic Dataset, which includes 30 features like mean texture and mean area for over 500 records, providing a robust basis for our analysis.

.

DATA PRE-PROCESSING

Handling Missing data

- **Clean missing or inconsistent values to prevent bias or errors in predictions.**

Normalizing Feature values

- **Scale features to ensure no single feature dominates, maintaining balanced data input.**

Splitting Data

- **Divide data into 80% training and 20% testing to evaluate model accuracy and generalization.**

Methodology – Model Training and Testing

- **ML Algorithms Used:**

ML Algorithms

- **Logistic Regression:** Simple, interpretable model.
- **SVM:** Effective for high-dimensional data.
- **Random Forest:** Handles non-linear data and provides feature importance.
- **KNN:** Classifies data based on the majority class of its nearest neighbors.
- **Decision Tree:** Uses a tree-like structure to split data based on feature values for classification.

- **Training and Testing:**

Training Process:	Testing Process:
• Train models using 80% training data. • Hyperparameter optimization using Grid Search.	• Evaluate models on test datasets using metrics like accuracy, precision, recall, and F1-score. • Apply cross-validation to prevent overfitting.

Now let's have a look at System Requirements and System Design-

System Requirements-

Hardware	Software
Processor: i5 or higher	Python for ML model development.
RAM: 8GB or more	MongoDB for database management.
Storage: Minimum 256 GB SSD	React.js and Node.js for web application.

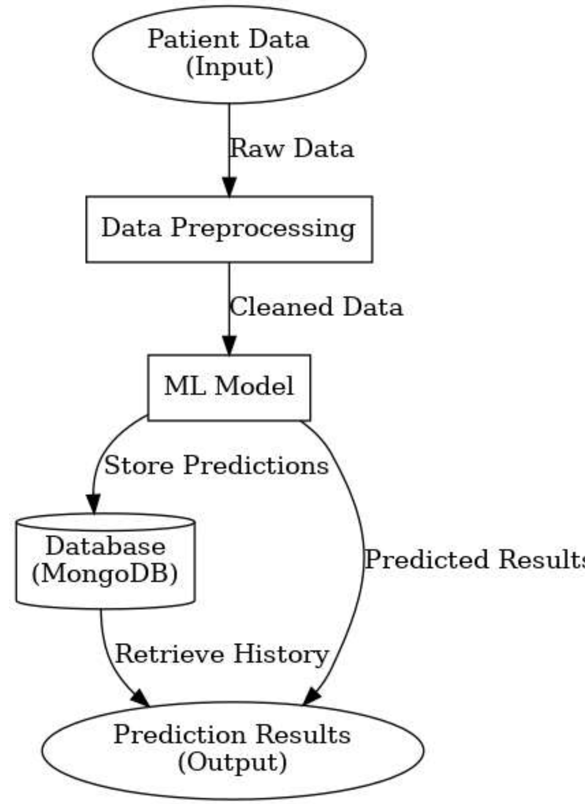
System Design-

The workflow includes:

1. **Frontend:**
   - Collects user input for the 30 diagnostic features via forms in React.js.
   - Sends the input to the backend through RESTful API calls.
2. **Backend:**
   - Receives input via Flask routes and processes it using the trained machine learning model.
   - Generates predictions (e.g., Malignant or Benign) and SHAP explanations.
   - Send the results back to the frontend in JSON format.
3. **Frontend (Output):**
   - Displays predictions and interpretable explanations to users.
   - Updates user dashboards with prediction history.

4. Database

- **Purpose:** Store patient data, predictions, and historical records.
- **Technology:**
  - Firestore for storing:
  - Patient information: Name, age, gender, health metrics.
  - Prediction data: Model results, timestamps, and confidence scores.



Technology Stack:

- **Frontend:** React.js (Dynamic and responsive UI).
- **Backend:** Flask (REST API) and Firebase.
- **Machine Learning:** SVM with SHAP for explainability.
- **Database:** Firebase Firestore stores user data and prediction history.

Flask

Security Measures for the Web Application

Penetration Testing:
- Conduct regular testing to identify and resolve security vulnerabilities.

OWASP Top 10 Compliance:
- Address risks like SQL Injection, Cross-Site Scripting (XSS), and Broken Authentication.

Data Encryption:
- Ensure secure communication using HTTPS and encrypt sensitive data in the database.

Secure Authentication:
- Implement multi-factor authentication (MFA) and store passwords using hashing algorithms.

Input Validation and Sanitization:
- Validate and sanitize user inputs to prevent injection and other attacks.

Access Control:
- Use role-based access control (RBAC) to restrict unauthorized access to sensitive data and features.

Regular Updates and Patching:
- Frequently update software libraries and dependencies to patch vulnerabilities.

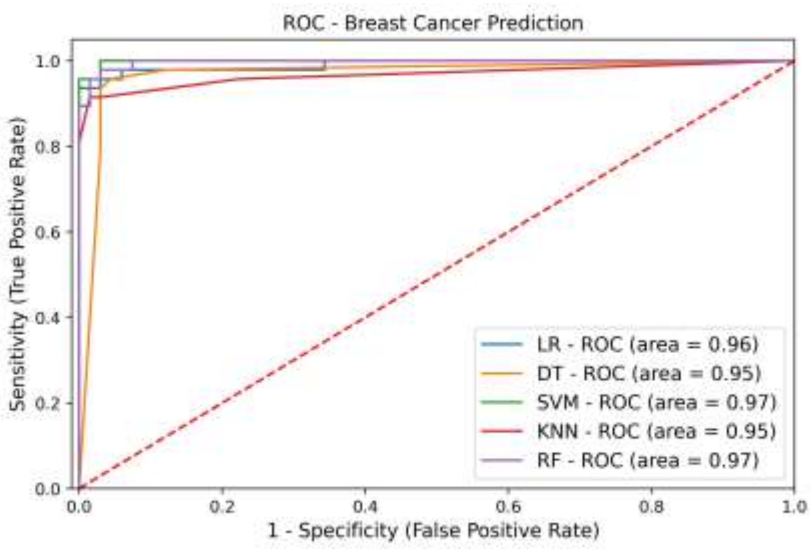
Monitoring and Logging:
- Set up logging and monitoring systems to detect and respond to suspicious activities real time.

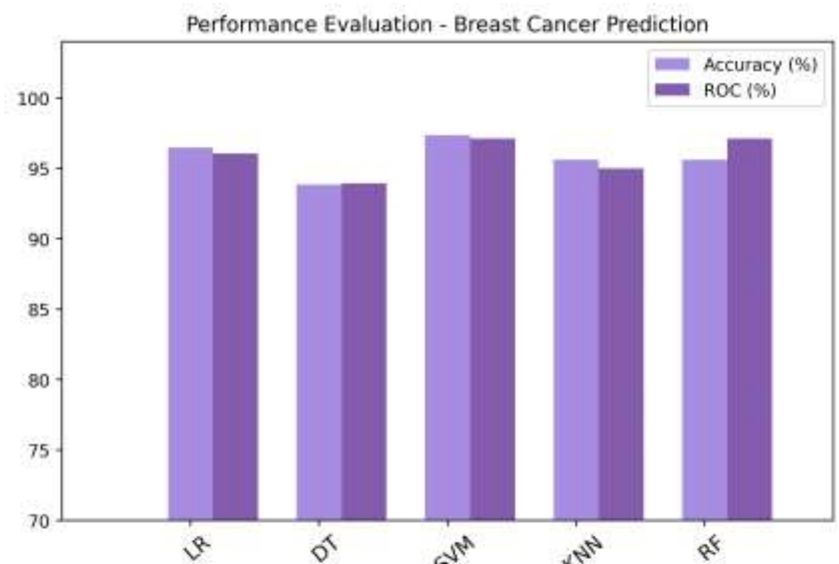
Prediction Results and Insights

Performance Metrics:

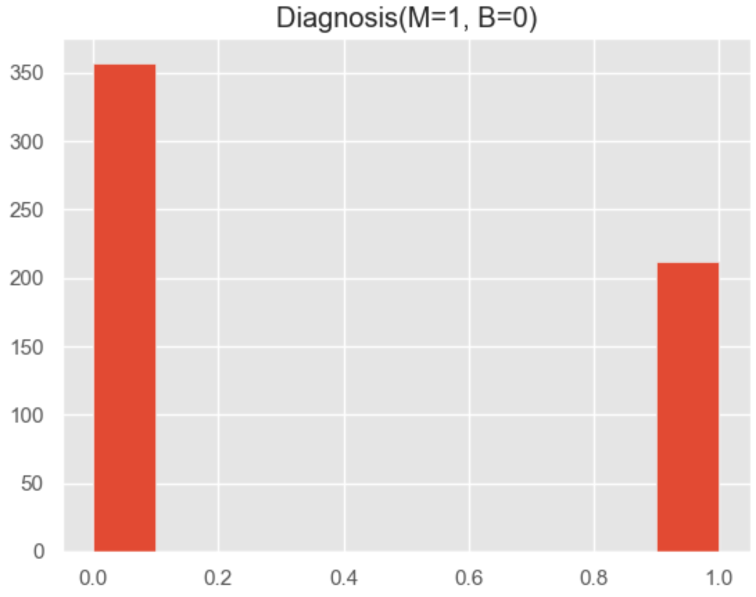
- **SVM**: Accuracy = 97%
- **Logistic Regression**: Accuracy = 95%
- **KNN**: Accuracy = 94%
- **Decision Tree**: Accuracy = 94%
- **Random Forest**: Accuracy = 90%

Visualizations:





Prediction Results and Insights



The chart on the left displays the accuracy comparison among various machine learning algorithms used in the model. The SVM model achieved the highest accuracy at 97%, demonstrating its effectiveness for this task.

The pie chart on the right illustrates the distribution of predictions made by the model. Out of all predictions, 70% were classified as benign, and 30% as malignant.

Conclusion and Future Scope

Conclusion:

- The project successfully demonstrates the potential of machine learning in breast cancer prediction.
- Web-based implementation ensures accessibility and usability for healthcare professionals.

Future Scope:

- Adding features like real-time data analysis and personalized recommendations.
- Analyze mammography images to identify cancerous tumours efficiently.

References

- Fatima Noreen, Haroon Ahmed (2020), Prediction of Breast Cancer, Comparative Review of Machine Learning Techniques, and Their Analysis, IEEE Access, 10.1109/ACCESS.2020.3016715
- Mamta Sai Yarabarla, Lakshmi Kavya Ravi, A.Sivasangari (2019), Breast Cancer Prediction via Machine Learning, IEEE Xplore, 10.1109/ICOEI.2019.8862533
- https://www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470
- "Improving Cancer Detection Using Machine Learning," Journal of Healthcare Informatics, 2023.
- "Web Application Security and Best Practices," OWASP, 2023.

Thank You