Breast Cancer Classification Using Machine Learning: A Comparative Analysis with Logistic Regression and Random Forest

GitHub Link- https://github.com/YuvarajMandal/Midterm

1. Introduction

Breast cancer is the most prevalent cancer among women globally, accounting for approximately 25% of all cancer cases, with over 2.1 million new diagnoses in 2015. Early and accurate detection is critical for improving patient outcomes, as it enables timely intervention. The Breast Cancer Wisconsin (Diagnostic) Dataset provides a valuable resource for developing machine-learning models to classify breast tumors as malignant (cancerous) or benign (non-cancerous). This project focuses on implementing and comparing two machine learning models—Logistic Regression and Random Forest—for binary classification of breast tumors. Both models were manually coded from scratch.

2. Dataset Source and Description

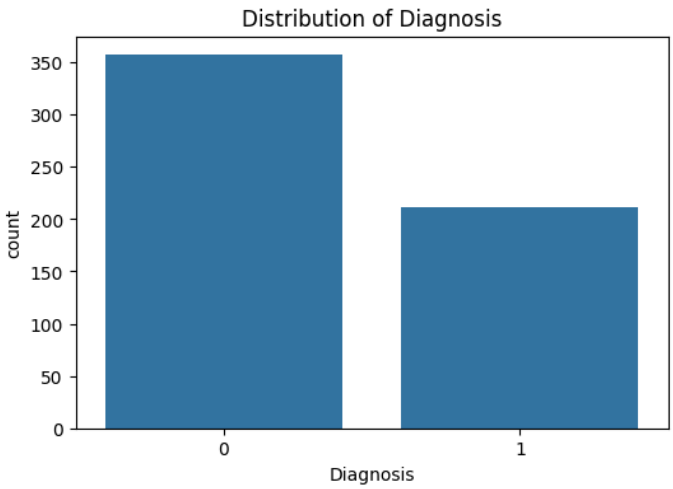
The dataset used in this project is the Breast Cancer Wisconsin (Diagnostic) Dataset, sourced from Kaggle. It contains 569 instances, each representing a breast tumor, with 30 features describing characteristics of cell nuclei, such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. They are provided in three forms: mean, standard error, and worst (largest) values, resulting in 30 total features. The target variable, "Diagnosis," indicates whether the tumor is malignant (M) or benign (B). In preprocessing, the "Diagnosis" column was converted to a binary format: 1 for malignant and 0 for benign.

The dataset has no missing values, as confirmed by the df.info() output, which shows 569 non-null entries for all 31 columns (30 features as float64 and 1 target as int64).

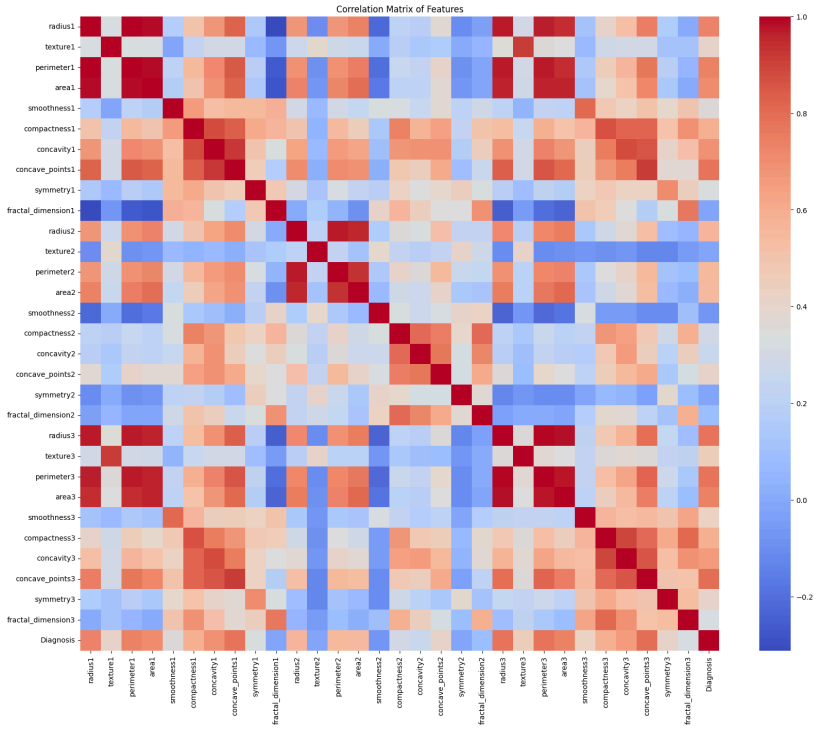
3. Data Exploration and Important Features

Initial exploration of the dataset revealed the following insights:

- The dataset has a class imbalance, with approximately 63% benign (357 samples) and 37% malignant (212 samples) cases.



- Features like radius, perimeter, and area are highly correlated, as they describe similar aspects of tumor size. Diagnosis (M/B) has a good degree of correlation with other features, hence making it a good dataset to perform machine learning.



3. Methods:

The methodology involved the following steps:

A. Data Preprocessing:

The "Diagnosis" column was converted from categorical ("M"/"B") to binary (1/0).

Features were standardized using a standardized function, which computes the mean and standard deviation of each feature in the training set and scales both the training and test sets accordingly. This ensures that features are on the same scale, which is critical for Logistic Regression to prevent features with larger ranges from dominating the model.

B. Model Implementation and Algorithm Explanation:

B1. Logistic Regression:

Logistic Regression is a linear model for binary classification that predicts the probability of a sample belonging to the positive class (malignant = 1) using the sigmoid function. It learns a set of weights and a bias term to form a linear decision boundary, optimizing these parameters by minimizing the log-loss (cross-entropy loss) using gradient descent.

Detailed Algorithm:

For a sample with features X=[x1,x2,...,x30], the model computes a linear combination: z=w0+w1x1+w2x2+...+w30x30 where w0 is the bias (intercept), and w1,w2,...,w30 are the weights for each of the 30 features.

Sigmoid Function: The linear combination z is passed through the sigmoid function to obtain a probability: P(y=1|X)= $\frac{1}{1+e^{-z}}$.

To prevent numerical overflow, z is clipped between -500 and 500 before applying the sigmoid function.

The model minimizes the log-loss, defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

where yi is the true label (0 or 1), y^i is the predicted probability, and N is the number of samples.

The weights and bias are updated iteratively using gradient descent over 10,000 iterations with a learning rate of 0.1. First predicted probabilities y^ are computed. Then gradients are computed as follows:

- For weights: $\frac{\partial L}{\partial w} = \frac{1}{N} X^T(\hat{y} - y)$, where X^T is the transpose of the feature matrix.
- For bias: $\frac{\partial L}{\partial b} = \frac{1}{N} \sum(\hat{y} - y)$.

Update parameters: $w \leftarrow w - \eta \cdot \frac{\partial L}{\partial w}$, $b \leftarrow b - \eta \cdot \frac{\partial L}{\partial b}$, where $\eta = 0.1$.

The model predicts probabilities using the learned weights and bias. A threshold of 0.5 is applied to classify samples: if P(y=1|X)≥0.5 predict 1 (malignant); otherwise, predict 0 (benign).

B2. Random Forest:

Random Forest is an ensemble method that builds multiple decision trees and combines their predictions through majority voting. Each tree is trained on a bootstrap sample of the data.

Detailed Algorithm:

Decision Tree:

Each tree recursively splits the data by selecting the feature and threshold that maximize information gain, computed using entropy:

- Entropy: $H(y) = -\sum_c p_c \log_2(p_c)$, where p_c is the proportion of class c (0 or 1) in the node.
- Information Gain: For a split on feature j at threshold t, the information gain is:

$$IG = H(\text{parent}) - \left(\frac{n_{\text{left}}}{n} H(\text{left}) + \frac{n_{\text{right}}}{n} H(\text{right})\right)$$

where n_{left} and n_{right} are the number of samples in the left and right splits, and n is the total number of samples in the node.

For each node, the algorithm iterates over all features and their unique values to find the split that maximizes information gain. If no valid split is found (e.g., all samples belong to one class or the number of samples is below the minimum split threshold of 2), the node becomes a leaf, and the majority class is assigned.

The maximum depth of 5 limits overfitting by preventing the tree from growing too complex. Hence, no need for pruning.

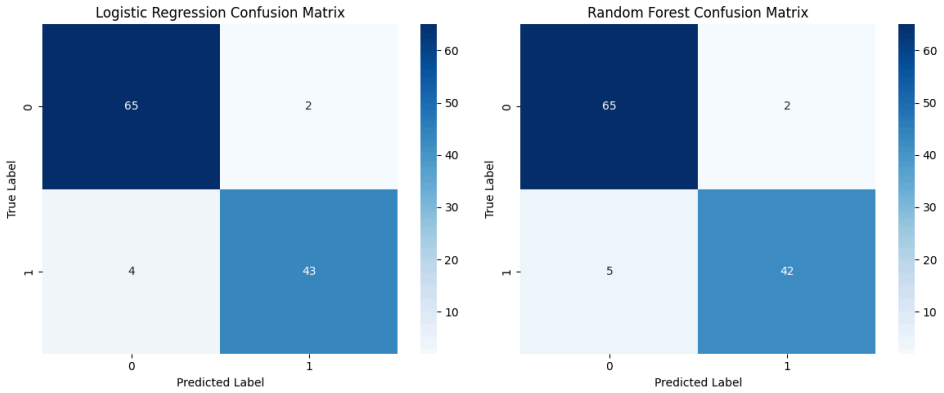
Random Forest:

For each of the 10 trees, a bootstrap sample is created by sampling the training data with replacement. Each tree is trained independently on its bootstrap sample using the DecisionTree class. The random state (set to 42) ensures reproducibility of the bootstrap samples.

For a given sample, each of the 10 trees makes a prediction by traversing its structure. The final prediction is determined by majority voting: the class (0 or 1) with the most votes across all trees is selected.

4. Final Results

- The final results indicate that Logistic Regression achieved the highest performance, with an accuracy of 94.74%, precision of 95.56%, recall of 91.49%, and F1-score of 93.48%.
- Random Forest performed slightly worse, with an accuracy of 93.86%, precision of 95.45%, recall of 89.36%, and F1-score of 92.31%.



5. Conclusion

This project analysed the Breast Cancer Wisconsin (Diagnostic) Dataset to classify tumors as malignant or benign using manually implemented Logistic Regression and Random Forest models. Logistic Regression emerged as the best-performing model, achieving high accuracy and recall, which are crucial for medical diagnostics. Future work could involve hyperparameter tuning (e.g., adjusting the learning rate, number of trees, or maximum depth), performing feature selection to reduce dimensionality, and addressing class imbalance to further improve recall for the malignant class.