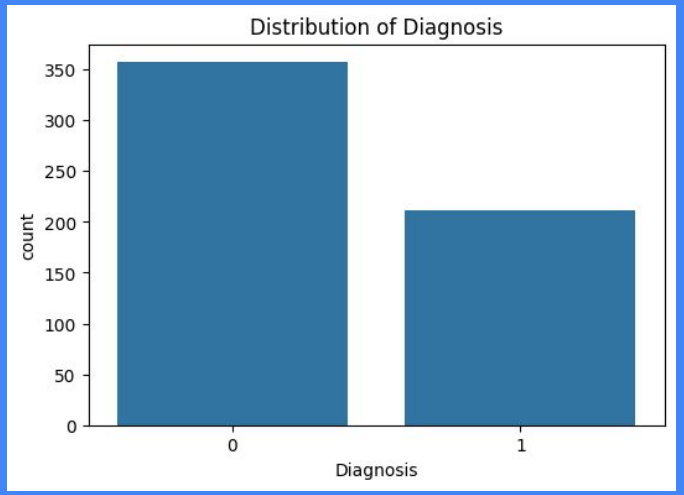
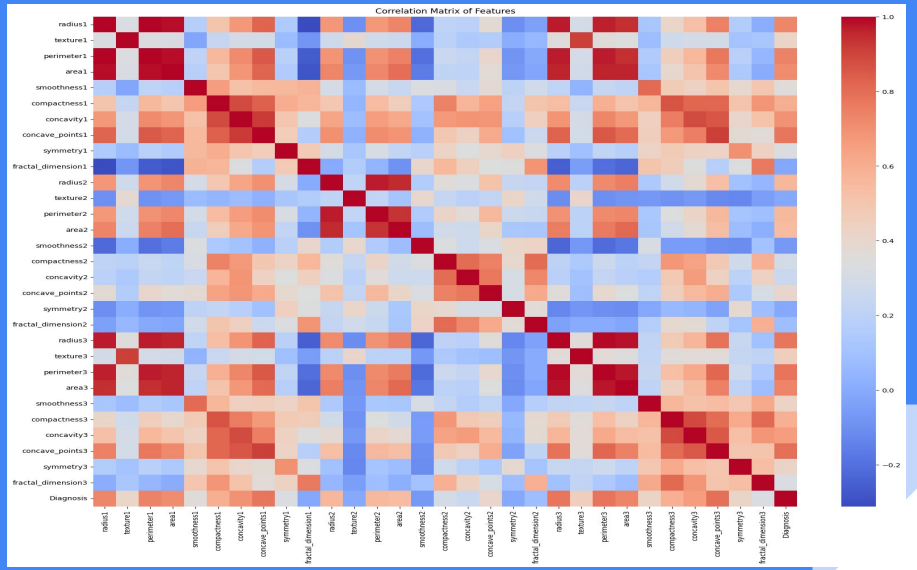
Breast Cancer Classification Using Machine Learning: A Comparative Analysis with Logistic Regression and Random Forest

About dataset:

- Breast Cancer Wisconsin (Diagnostic) Dataset, sourced from Kaggle.
- 569 instances, each representing a breast tumor, with 30 features
- The target variable, "Diagnosis," indicates whether the tumor is malignant (M) or benign (B) -> 1 for M and 0 for B
- The dataset has no missing values





Logistic Regression

Features were standardized using a standardized function

Now, logistic regression

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.

The linear combination z is passed through the sigmoid function to obtain a probability: P(y=1|X)

The model minimizes the log-loss, defined as:

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

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 $\eta = 0.1$.

Random Forest

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.

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.

Random Forest

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

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 final prediction is determined by majority voting: the class (0 or 1) with the most votes across all trees is selected.

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%.