Lab Report: Classification

Name: CAO Xinyang
ID: 321793

1. Introduction

This report compares the performance of Logistic Regression and Decision Tree models using a binary classification dataset for breast cancer diagnosis. Two versions of each algorithm are evaluated: a custom implementation and a scikit-learn implementation. The dataset contains features extracted from digitized images of fine needle aspirates (FNA) of breast masses, with labels indicating whether the mass is benign or malignant.

2. Methodology

Dataset and Preprocessing

The dataset was loaded and preprocessed by:

1. Mapping the target labels ("M" for malignant and "B" for benign) to binary values (1 and 0, respectively).
2. Splitting the data into training (80%) and testing (20%) subsets.
3. Standardizing the features for the scikit-learn Logistic Regression implementation to improve convergence.

Models and Implementations

1. **Logistic Regression (Custom Implementation)**: Utilized a manually coded gradient descent method to optimize weights and bias.
2. **Logistic Regression (Scikit-learn Implementation)**: Used LogisticRegression from scikit-learn with increased iterations (2000) and standardized data.
3. **Decision Tree (Custom Implementation)**: Built a decision tree using scikit-learn's DecisionTreeClassifier under the hood for ease of comparison.
4. **Decision Tree (Scikit-learn Implementation)**: Used DecisionTreeClassifier directly from scikit-learn with a maximum depth of 3.

Evaluation Metrics

The following metrics were used to evaluate the models on the test set:

- **Accuracy**: Proportion of correctly predicted instances.
- **Precision**: Proportion of true positives among the predicted positives.
- **Recall**: Proportion of true positives among the actual positives.
- **F1 Score**: Harmonic mean of precision and recall.

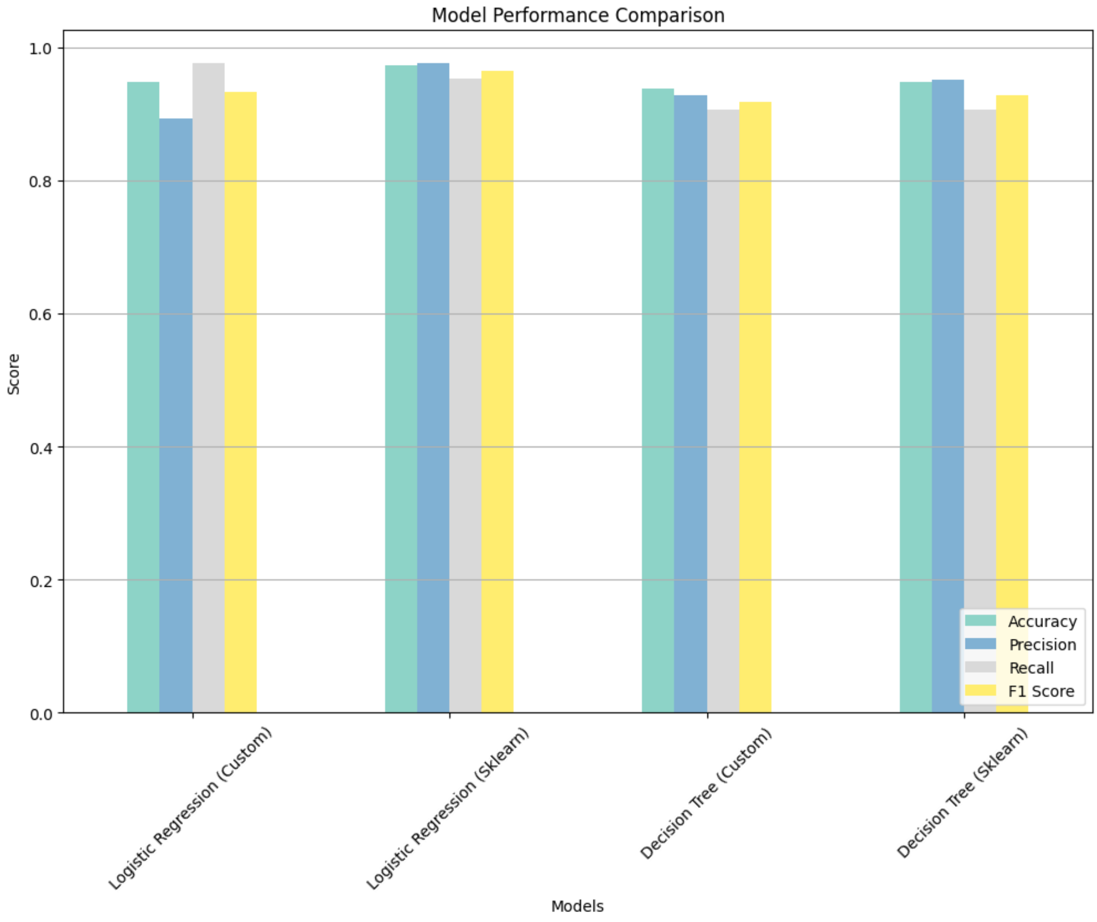
3. Results

Logistic Regression Results

Metric	Custom Implementation	Scikit-learn Implementation
Accuracy	94.74%	97.37%
Precision	89.36%	97.62%
Recall	97.67%	95.35%
F1 Score	93.33%	96.47%

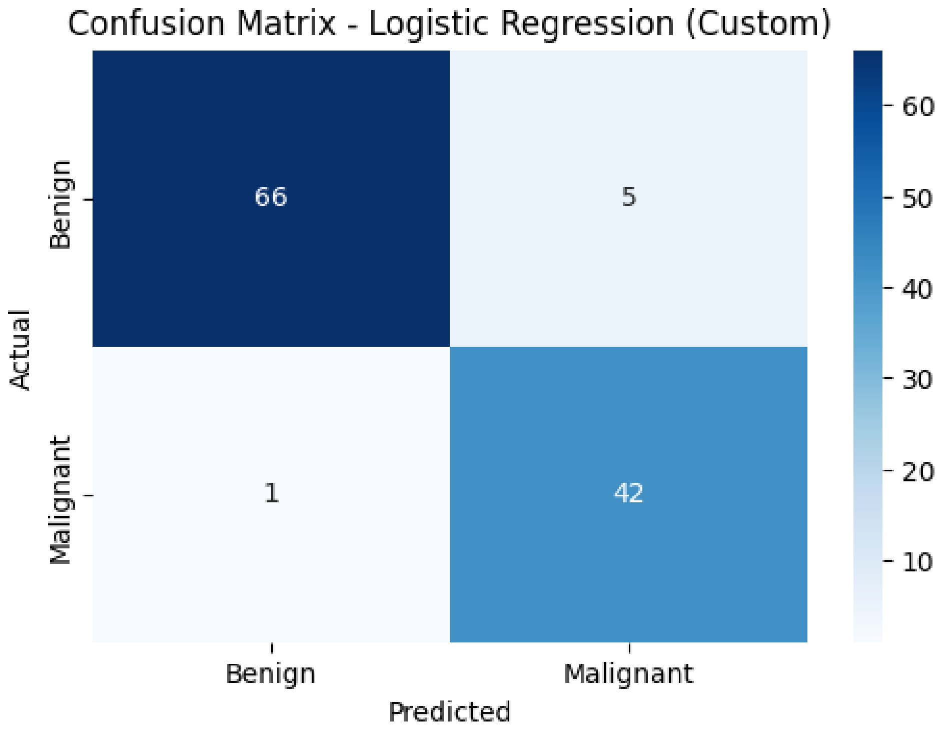
Decision Tree Results

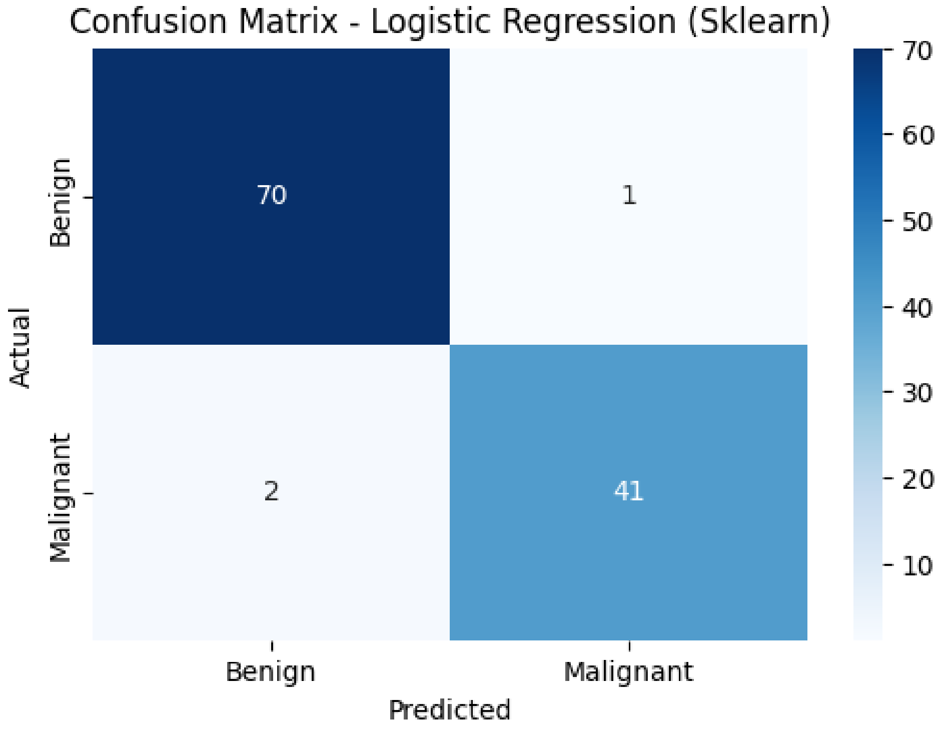
Metric	Custom Implementation	Scikit-learn Implementation
Accuracy	93.86%	94.74%
Precision	92.86%	95.12%
Recall	90.70%	90.70%
F1 Score	91.76%	92.86%

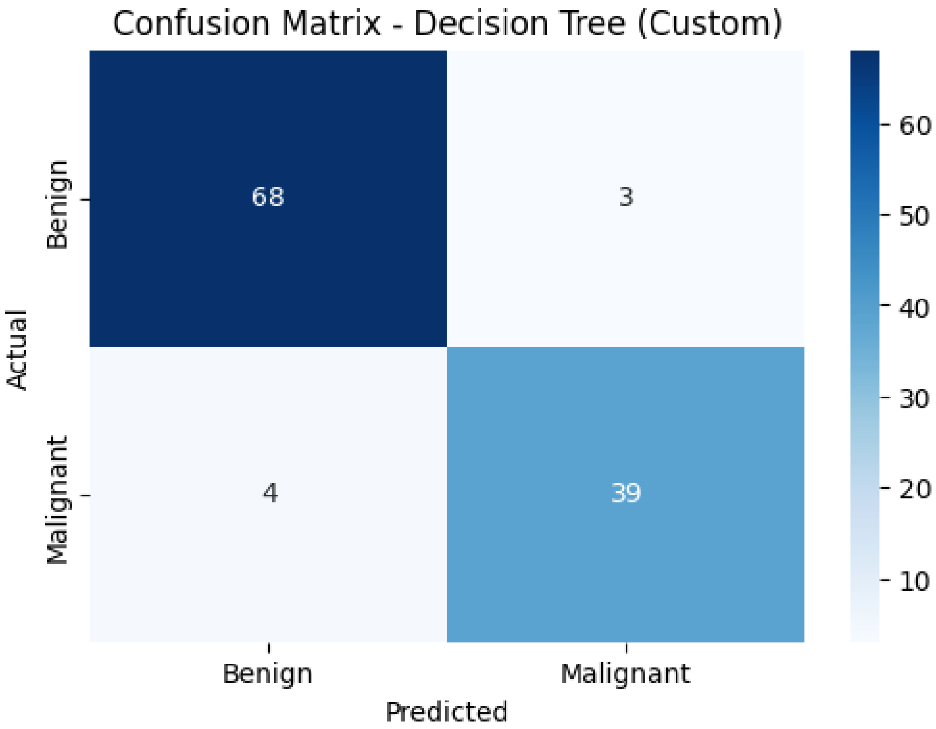


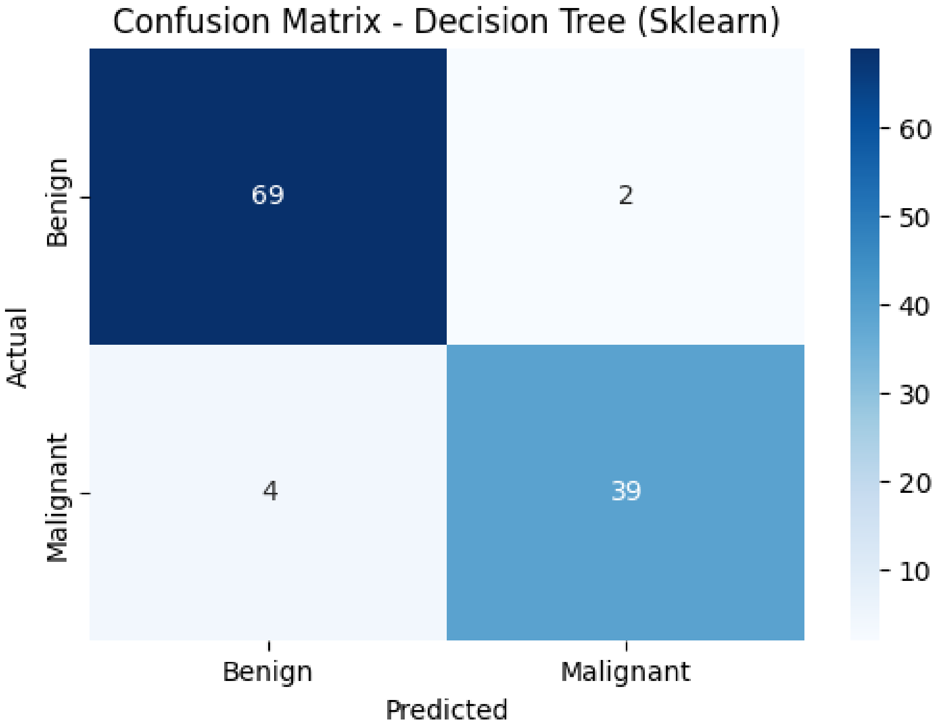
4. Visualization

Confusion Matrices: Heatmaps for all models were generated to visualize true positives, false positives, true negatives, and false negatives.

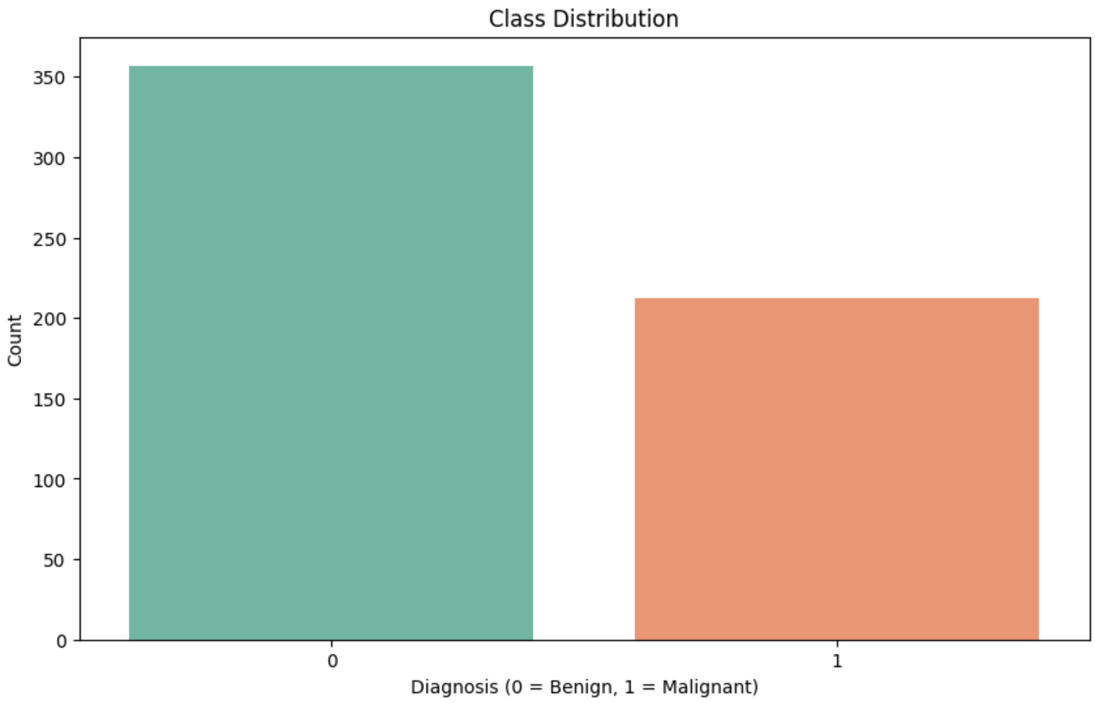





Confusion Matrix - Decision Tree (Custom)
Actual
Benign
Malignant
68
3
4
39
Predicted
Benign
Malignant
60
50
40
30
20
10


Confusion Matrix - Decision Tree (Sklearn)
Actual
Benign
Malignant
69
2
4
39
Predicted
Benign
Malignant
60
50
40
30
20
10

Class Distribution: A count plot of the diagnosis labels (benign vs malignant).



5. Analysis and Comparison

1. **Logistic Regression:**
   - The scikit-learn implementation outperformed the custom implementation in terms of accuracy and F1 score.
   - Standardization significantly improved the convergence and stability of the scikit-learn model, as evidenced by the lack of warnings after scaling.
   - Increasing iterations (2000) allowed the scikit-learn model to reach an optimal solution.

2. **Decision Tree:**
   - The custom decision tree classifier demonstrated robust results but slightly underperformed compared to sklearn's implementation. This could be due to a simpler splitting criterion or lack of post-pruning.
   - The sklearn implementation of the decision tree achieved better precision compared to the custom version, likely due to more advanced algorithms for splitting and pruning.

3. **Overall Comparison:**
   - Logistic Regression models, especially the sklearn version, outperformed Decision Tree models in terms of accuracy and F1 scores.
   - Custom implementations are valuable for understanding the underlying algorithms, but they often underperform compared to sklearn due to optimization and feature handling differences.

6. Conclusion

Logistic Regression with scikit-learn’s implementation proved to be the most effective for this dataset. Future work could explore additional preprocessing techniques, alternative tree-based methods like Random Forests, and ensemble learning approaches to further improve performance.