Optimization for Data Science (MAL7070)

Topic：Optimization of Logistic Regression using Gradient Descent and newton method

Submitted By:

Pooja Naveen Poonia (M24CSA020)
Shivani Tiwari (M24CSA029)
Suvigya Sharma (M24CSA033)

Submitted To:

Dr. Kuntal Som

Objective:

The primary objective of this project was to:

- **Optimize the performance of logistic regression using two different optimization techniques**: Gradient Descent with Backtracking and Newton's Method.
- **Evaluate and compare the accuracy and convergence rates** of each method, along with the performance of Scikit-Learn's built-in logistic regression model.

Dataset Details:

Variable	**Description**
Area	Total pixel area of the raisin sample.
Perimeter	Total perimeter length around the raisin sample.
MajorAxisLength	Longest axis of the raisin.
MinorAxisLength	Shortest axis of the raisin.
Eccentricity	Shape feature measuring elongation.
ConvexArea	The area of the convex shape surrounding the raisin.
Extent	Ratio of pixels in the raisin area to pixels in the bounding box.
Class	Indicates the type of raisin, either "Kecimen" or "Besni."

Methodology:

1. **Data Exploration**
    - Dataset contains 900 rows and 8 columns
    - Numerical Features : Area, MajorAxisLength, MinorAxisLength, Eccentricity, ConvexArea, Extent, Perimeter
    - Categorical Features : Class (Kecimen, Besni)
    - Missing Values : No missing values in this dataset

2. Data Preprocessing

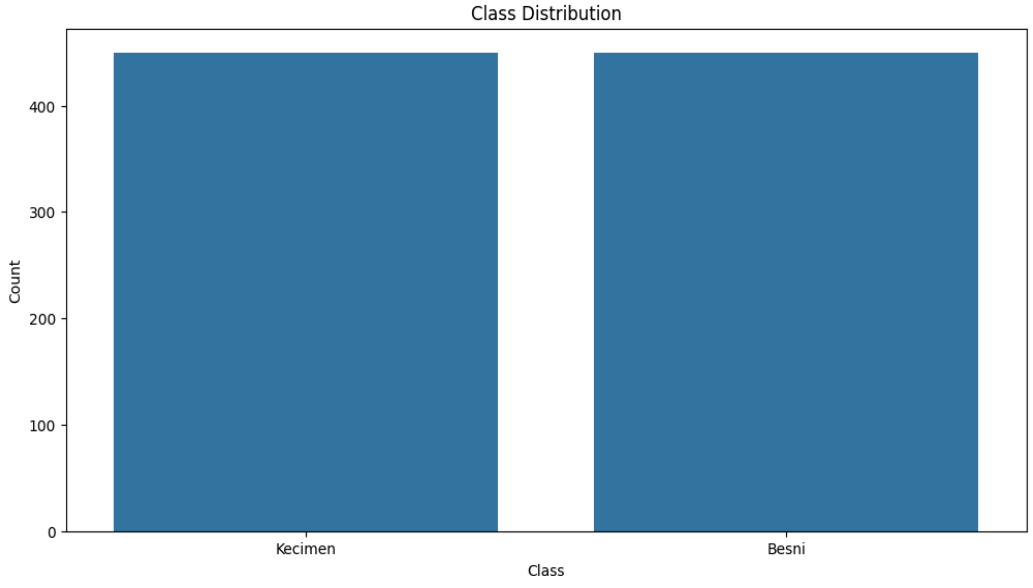
- Impute missing value
- Categorical encoding : Convert the Class feature to a binary encoded format (0 for Kecimen , 1 for Besni)
- Scaling and normalization : Standardization was applied to the numerical features to scale them to a standardized distribution with a mean of zero and a standard deviation of one, ensuring consistency across features regardless of their original scales.

3. Exploratory Data Analysis (EDA)

- Descriptive Statistics : Generate summary statistics (mean, median, standard deviation, etc.) for numerical features to understand basic distribution patterns.

	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimeter
count	900.000000	900.000000	900.000000	900.000000	900.000000	900.000000	900.000000
mean	87804.127778	430.929950	254.488133	0.781542	91186.090000	0.699508	1165.906636
std	39002.111390	116.035121	49.988902	0.090318	40769.290132	0.053468	273.764315
min	25387.000000	225.629541	143.710872	0.348730	26139.000000	0.379856	619.074000
25%	59348.000000	345.442898	219.111126	0.741766	61513.250000	0.670869	966.410750
50%	78902.000000	407.803951	247.848409	0.798846	81651.000000	0.707367	1119.509000
75%	105028.250000	494.187014	279.888575	0.842571	108375.750000	0.734991	1308.389750
max	235047.000000	997.291941	492.275279	0.962124	278217.000000	0.835455	2697.753000

- Class Distribution

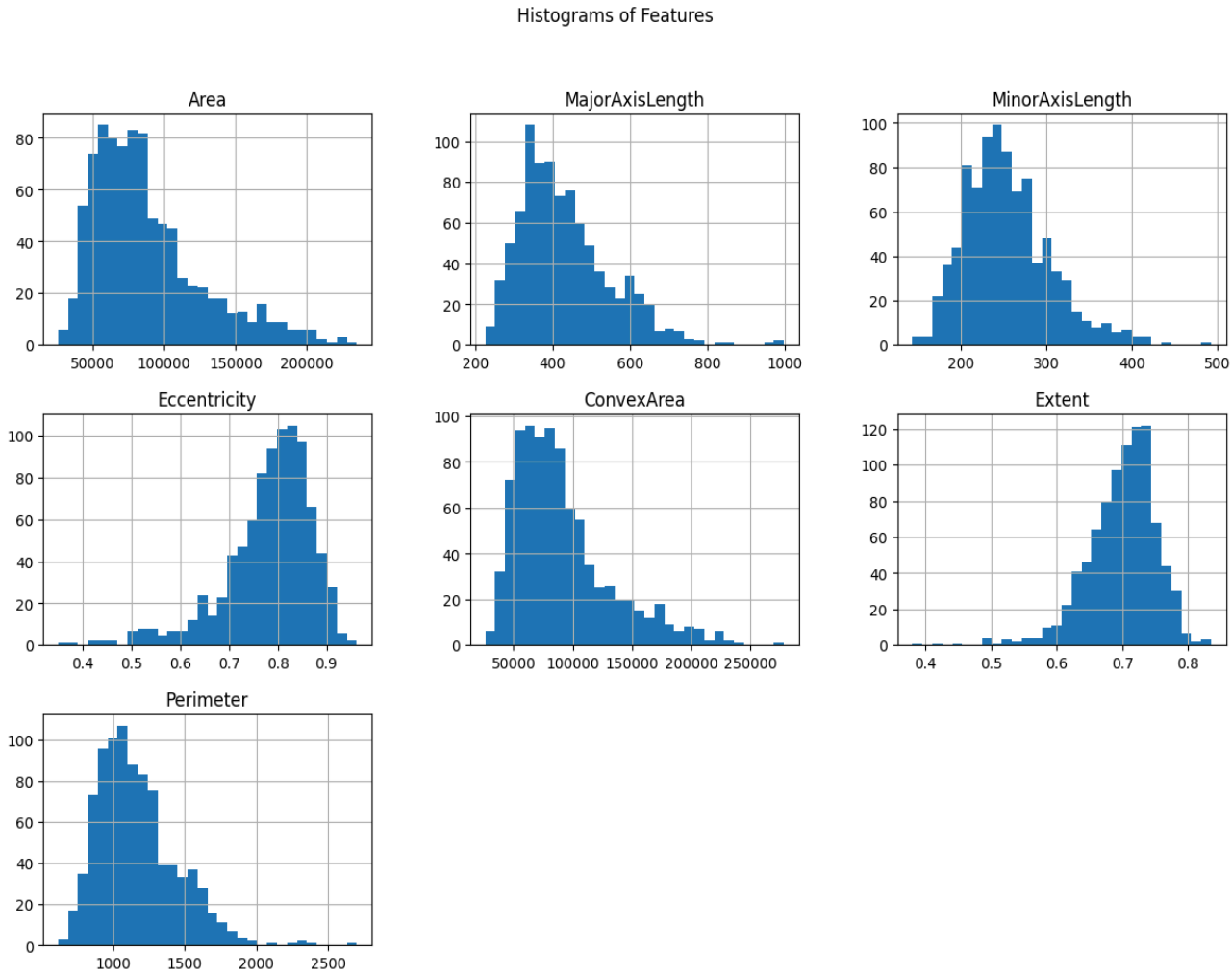


-

The dataset is balanced, with 450 samples per class

Histograms of each feature reveal the following insight

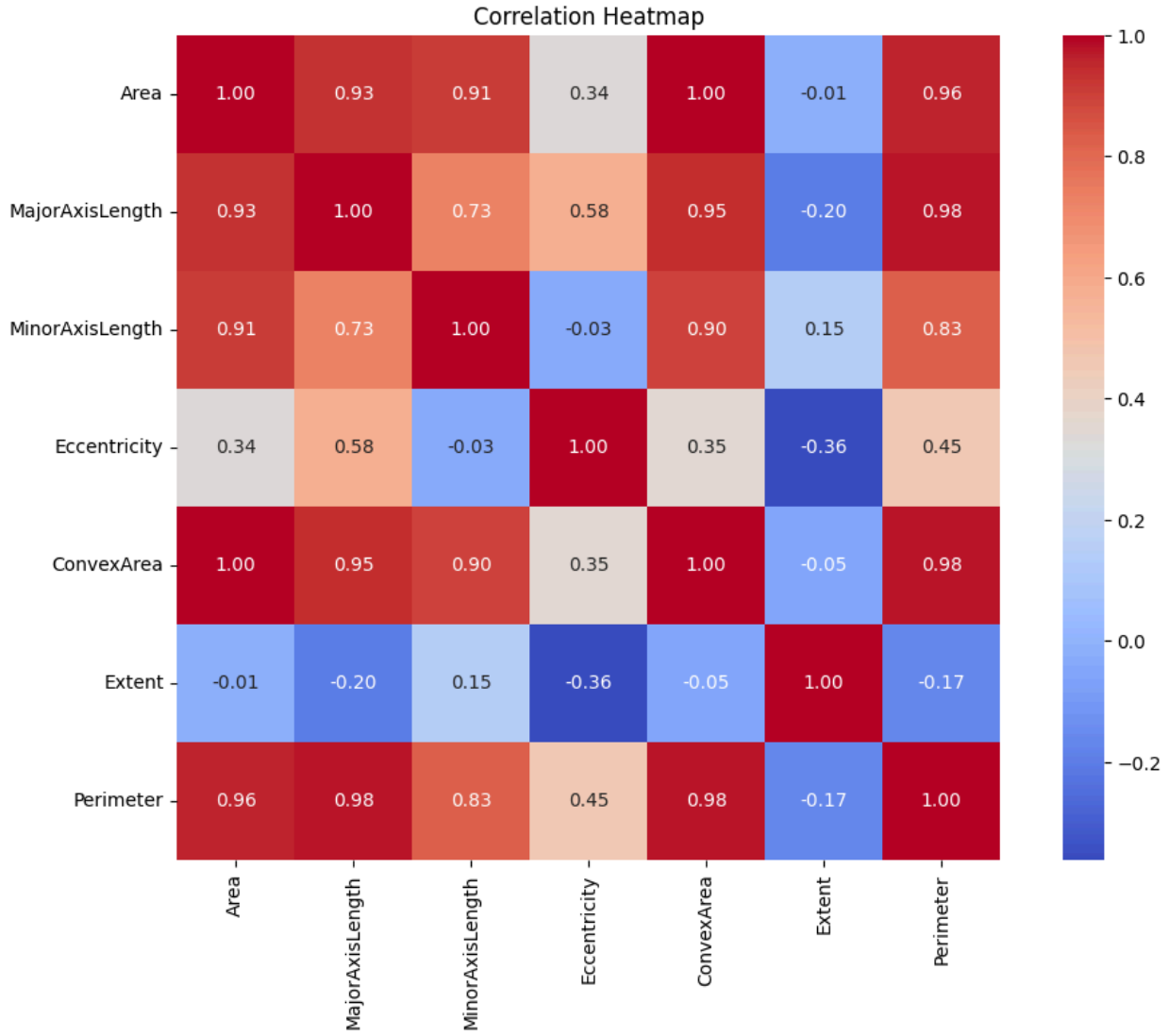
- Right-skewed Distributions: Features like Area, MajorAxisLength, MinorAxisLength, ConvexArea, and Perimeter are right-skewed, with some extreme values. This suggests that larger raisins are less common, which could affect model stability if not handled appropriately.
- Concentrated Distributions: Eccentricity and Extent show more concentrated distributions, indicating less variability in shape elongation and compactness.



Pairwise Feature Relationships

- A pairplot was used to examine the relationships between features and their separability with respect to the target classes.
- Strong Positive Correlations: There is a high positive correlation among Area, ConvexArea, MajorAxisLength, MinorAxisLength, and Perimeter. These features all capture different aspects of the raisin's size or boundary length.
- Class Separation: Eccentricity and Extent appear to provide moderate separation between classes, suggesting that these features could contribute useful information for distinguishing between Kecimen and Besni raisin types.

Correlation Heatmap



- Perfect Correlation: Area and ConvexArea are nearly perfectly correlated (correlation coefficient close to 1), suggesting these features are redundant. This redundancy could lead to multicollinearity, which may cause instability in models sensitive to feature collinearity.
- High Correlation Among Size Features: MajorAxisLength, MinorAxisLength, and Perimeter show strong correlations. This is expected given that larger raisins will have greater lengths and perimeter values.

4. Model Implementation with Optimization Techniques

4.1 Logistic Regression Model Formulation

- Logistic regression models the probability of belonging to one class given the input features. The model utilizes a sigmoid function to map predictions to the range [0, 1].
- The logistic regression objective function was formulated as follows:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(h_\theta(x_i)) + (1 - y_i)\log(1 - h_\theta(x_i))\right]$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

4.2 Optimization Techniques

- Gradient Descent with Backtracking
  - ➔ **Objective**: Minimize the logistic loss function by iteratively adjusting model weights.
  - ➔ **Learning Rate Control**: The learning rate was controlled using a backtracking line search to adaptively reduce the step size when updates were too aggressive.
  - ➔ **Stopping Criteria**: Iterations were stopped based on a convergence threshold or a maximum iteration limit, whichever occurred first.
  - ➔ **Outcome**: This method showed slower convergence, requiring 503,974 iterations to reach a final accuracy of 86.67%.
- Newton's Method
  - ➔ **Objective**: Utilize the second derivative (Hessian matrix) of the cost function to accelerate convergence.
  - ➔ **Steps**: Each update step was calculated as:

$$\theta = \theta - H^{-1}\nabla J(\theta)$$

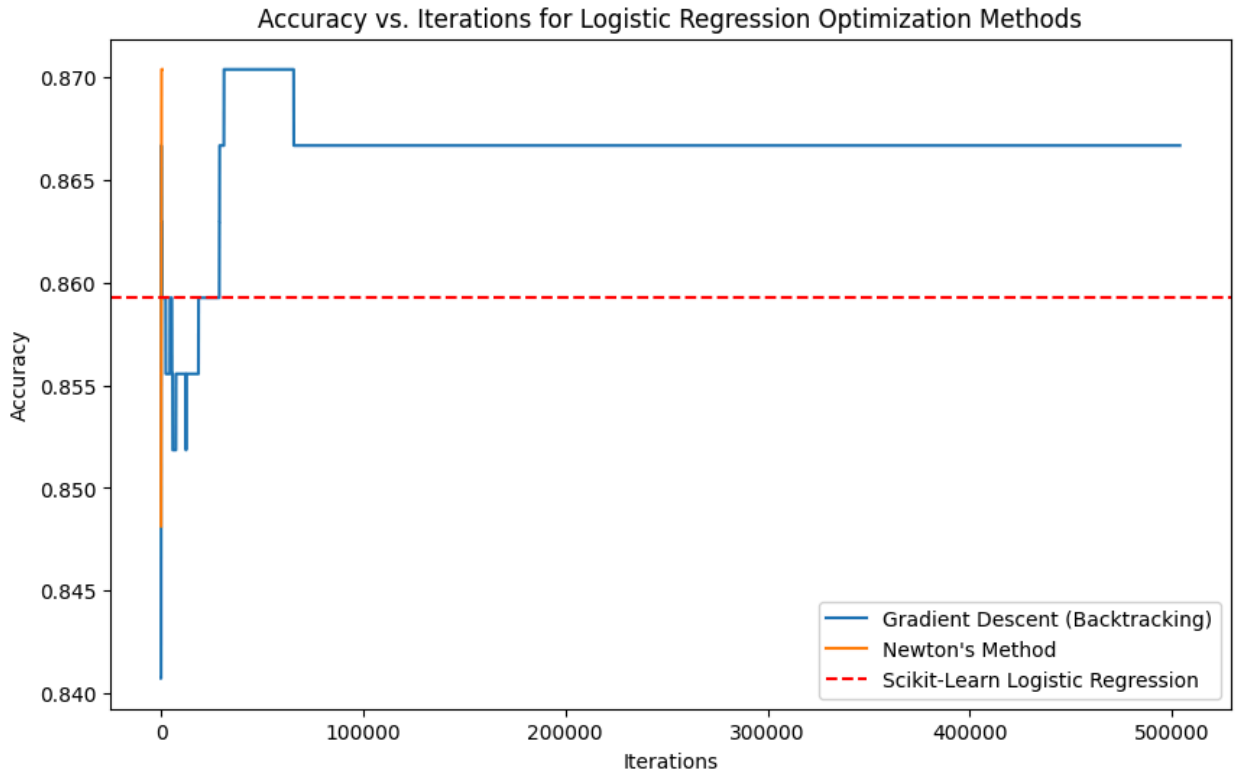
  - ➔ **Stopping Criteria**: This method achieved faster convergence, stopping at 532 iterations with a final accuracy of 87.04%.

➔ **Outcome**: Newton's Method exhibited much faster convergence than gradient descent due to the use of second-order derivative information, albeit with higher computational costs per iteration.

5. Performance Comparison and Analysis

Accuracy vs. Iterations: The accuracy achieved by each method across iterations

- Gradient Descent: Required a high number of iterations and careful step-size tuning to stabilize. Its accuracy eventually matched Newton's Method but only after substantial computational effort.
- Newton's Method: Converged quickly to high accuracy but was limited by the singular Hessian, which caused early stopping. This issue could potentially be mitigated by feature selection or dimensionality reduction to address multicollinearity.
- Scikit-Learn Logistic Regression: Achieved stable accuracy around 86.0% with minimal tuning, reinforcing its utility as a reliable baseline.



Conclusion:

- **Newton's Method** is highly efficient in reaching convergence with fewer iterations, but it requires numerical stability.
- **Gradient Descent with Backtracking Line Search** provides a flexible approach but at the cost of high computational demand, especially with large iteration counts and step-size tuning. This makes it less practical for small datasets or when computational resources are limited.
- **Scikit-Learn Logistic Regression** serves as an effective baseline with competitive accuracy and simplicity. It achieved comparable results to custom optimization methods without requiring extensive tuning, making it a strong choice for general-purpose applications.