UNIVERSITY OF MORATUWA

Department of Electrical Engineering

EN3150 - Pattern Recognition

Assignment 01

Learning from data and related challenges and linear models for regression

Student Name:	M. Thiruvarankan
Index No:	220647K
Degree:	B.Sc. Engineering (Hons)
Academic Year:	Semester 05
Department:	Electrical Engineering

Submitted on: August 19, 2025

Contents

1 Linear Regression Impact on Outliers [Code]

1.1 Dataset Analysis

The dataset provided consists of 10 data points with independent variable x and dependent variable y as shown in Table 1.

Table 1: Given dataset with x and y values

Index (i)	x_i	y_i
1	0	20.26
2	1	5.61
3	2	3.14
4	3	−30.00
5	4	−40.00
6	5	−8.13
7	6	−11.73
8	7	−16.08
9	8	−19.95
10	9	−24.03

1.2 Linear Regression Model Development

Using the complete dataset from Table 1, a linear regression model was fitted using the least squares method. The resulting linear regression model is:

$$y = -3.56x + 3.92 \tag{1}$$

Figure 1 shows the scatter plot of the data points along with the fitted linear regression line. The visualization clearly reveals the presence of outliers, particularly at data points $(3, -30)$ and $(4, -40)$, which deviate significantly from the general trend of the data.

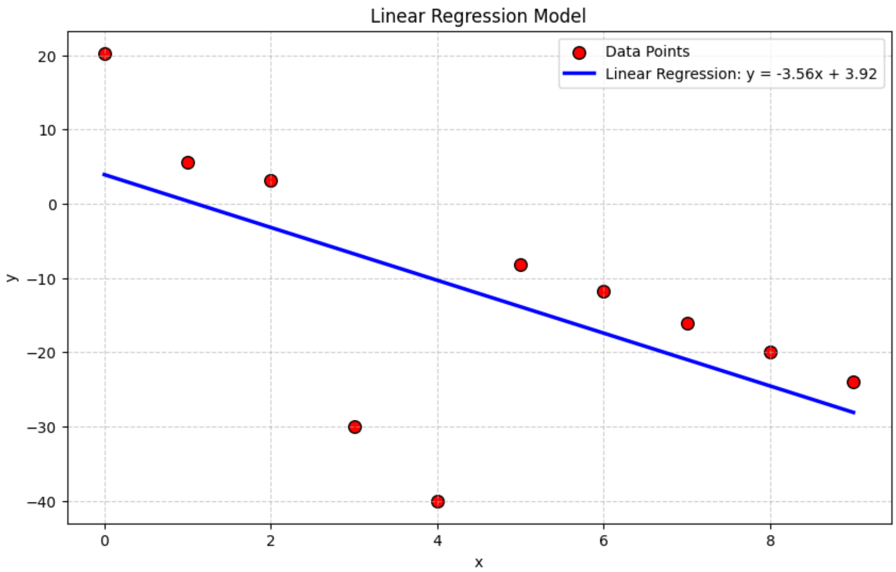

Figure 1: Linear regression model fitted to the dataset. Red points represent the original data, and the blue line shows the fitted linear regression model $y = -3.56x + 3.92$.

1.3 Model Comparison Framework

Two linear models are evaluated for their performance on the given dataset:

- **Model 1:** $y = -4x + 12$ (Given reference model)
- **Model 2:** $y = -3.56x + 3.92$ (Learned regression model from Section 1.2)

1.4 Robust Loss Function Implementation

To mitigate the impact of outliers, a robust loss function is employed:

$$L(\theta, \beta) = \frac{1}{N} \sum_{i=1}^{N} \frac{(y_i - \hat{y}_i)^2}{(y_i - \hat{y}_i)^2 + \beta^2} \tag{2}$$

where:

- θ represents model parameters (slope and intercept)
- β is the robustness hyperparameter
- $N = 10$ is the number of data samples
- y_i and $\hat{y}_i$ are the true and predicted values for the i-th sample

The robust loss function values were calculated for both models using $\beta = 1$, $\beta = 10^{-6}$, and $\beta = 10^3$.

Table 2: Robust loss function values for different β parameters

Model	$\boldsymbol{\beta = 1}$	$\boldsymbol{\beta = 10^{-6}}$	$\boldsymbol{\beta = 10^{3}}$
Model 1: $y = -4x + 12$	0.435416	1.000000	0.000227
Model 2: $y = -3.56x + 3.92$	0.973247	1.000000	0.000188

1.5 Beta Parameter Analysis and Selection

The behavior of the robust loss function across different β values is illustrated in Figure 2. This analysis is crucial for understanding the impact of the hyperparameter on outlier sensitivity.

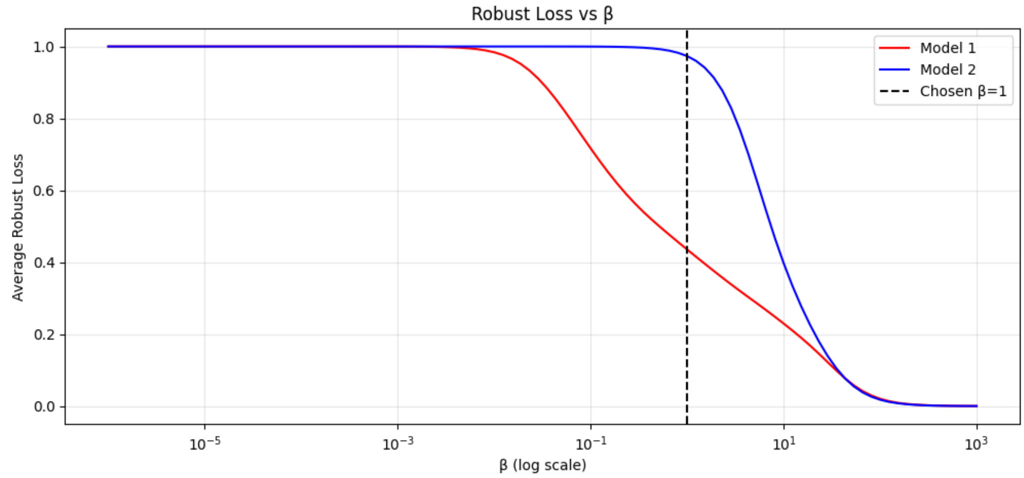

Figure 2: Robust loss function behavior across different β values. The plot shows how both models respond to varying levels of outlier suppression. The vertical dashed line indicates the chosen $\beta = 1$.

Recommended Value: $\beta = 1$

Justification for Beta Selection:

The selection of $\beta = 1$ is justified based on the following analysis:

Table 3: Effect of different β values on model behavior

β **Value**	**Behavior**	**Impact on Outliers**
10^{-6}	Extremely small	Treats most residuals as outliers; loss insensitive.
1	Moderate	Balances outlier suppression and sensitivity to normal data.
10^{3}	Very large	Large residuals largely unaffected; vulnerable to outliers.

1.6 Model Selection Using Robust Estimator

Based on the robust loss evaluation with $\beta = 1$:

Table 4: Model comparison using robust loss ($\beta = 1$)

Model	**Robust Loss Value**
Model 1: $y = -4x + 12$	0.435416
Model 2: $y = -3.56x + 3.92$	0.973247

Model Selection: Model 1 ($y = -4x + 12$) is selected as the superior model.

Selection Justification:

Model 1 achieves a significantly lower robust loss value (0.435416) compared to Model 2 (0.973247), indicating:

- Better resistance to outlier influence
- Superior fit to the central trend of the data
- More robust parameter estimates that are not dominated by extreme values

The robust estimator effectively identifies that Model 1 captures the underlying data pattern more accurately when outliers are appropriately handled.

1.7 Outlier Impact Analysis

1.7.1 Residual Analysis

Figure 3 presents a comprehensive residual analysis comparing both models, highlighting the distribution of errors across the dataset.

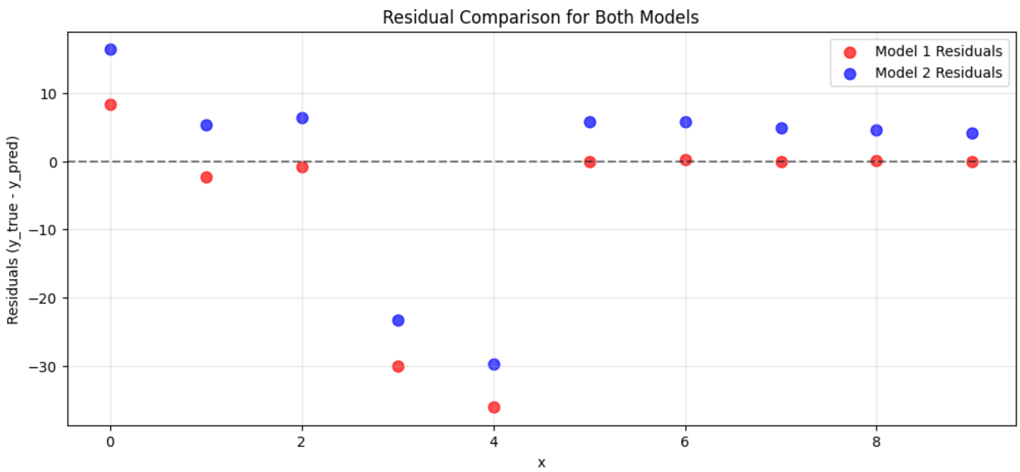

Figure 3: Residual comparison between Model 1 and Model 2. The plot shows the distribution of residuals (prediction errors) for both models across all data points.

1.7.2 Robust Loss Function and Outlier Resistance Mechanism

The robust loss function in Equation 2 reduces outlier impact through a saturation mechanism. For each residual $r_i = y_i - \hat{y}_i$, the loss contribution is:

$$\ell_i = \frac{r_i^2}{r_i^2 + \beta^2} \tag{3}$$

1.7.3 Mathematical Analysis of Outlier Suppression

Table 5: Robust loss behavior for different residual magnitudes

Residual Magnitude	**Loss Behavior**	**Interpretation**
Small $\lvert r_i \rvert \ll \beta$	$\ell_i \approx \frac{r_i^2}{\beta^2}$	Linear scaling with squared residual
Moderate $\lvert r_i \rvert \approx \beta$	$\ell_i \approx 0.5$	Transition region
Large $\lvert r_i \rvert \gg \beta$	$\ell_i \approx 1$	Saturated contribution (capped)

This saturation property ensures that extremely large residuals (outliers) contribute at most a value of 1 to the total loss, preventing them from dominating the optimization process.

1.7.4 Outlier Identification and Visualization

Figure 4 demonstrates the outlier detection capability of the robust approach, clearly identifying problematic data points.

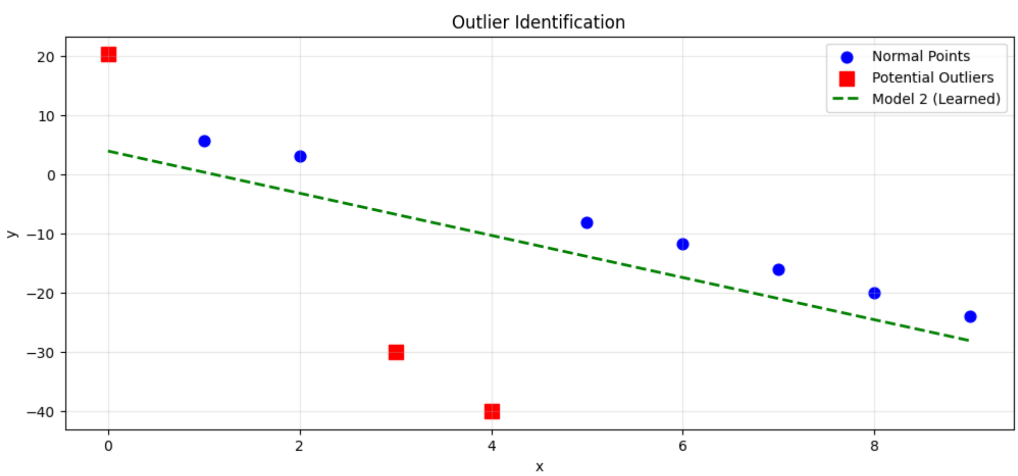

Figure 4: Outlier identification using residual analysis. Red squares indicate potential outliers identified based on the 75th percentile threshold of absolute residuals from Model 2. Blue circles represent normal data points.

1.8 Alternative Robust Loss Functions

1.8.1 Huber Loss Function

An alternative robust loss function that can be employed is the Huber loss:

$$L_\delta(r_i) = \begin{cases} \frac{1}{2}r_i^2 & \text{if } |r_i| \le \delta \\ \delta\left(|r_i| - \frac{1}{2}\delta\right) & \text{if } |r_i| > \delta \end{cases} \tag{4}$$

where $r_i = y_i - \hat{y}_i$ is the residual and δ is the threshold parameter.

1.8.2 Comparison of Robust Loss Functions

Table 6: Comparison of robust loss functions

Loss Function	**Behavior**	**Advantages**
Robust Loss	Continuous saturation for large residuals	Smooth optimization, differentiable everywhere
Huber Loss	Quadratic for small residuals, linear for large ones	Well-established theory, computational efficiency
Least Squares	Quadratic penalty for all residuals	Simple implementation, optimal for Gaussian noise

2 Loss Function Comparison: MSE vs BCE [Code]

2.1 Problem Statement

Two distinct applications require appropriate loss function selection:

- **Application 1:** Continuous dependent variable requiring Linear Regression
- **Application 2:** Binary dependent variable ($y \in \{0, 1\}$) requiring Logistic Regression

Two candidate loss functions are evaluated:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

$$\text{BCE} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{6}$$

2.2 Loss Function Analysis

A comprehensive analysis was conducted using $y_{\text{true}} = 1$ with varying prediction values from 0.005 to 1.0.

Table 7: MSE and BCE loss values for different predictions when $y = 1$

Prediction $\hat{y}$	**MSE**	**BCE**
0.005	0.9900	5.2983
0.01	0.9801	4.6052
0.05	0.9025	2.9957
0.1	0.8100	2.3026
0.2	0.6400	1.6094
0.3	0.4900	1.2040
0.4	0.3600	0.9163
0.5	0.2500	0.6931
0.6	0.1600	0.5108
0.7	0.0900	0.3567
0.8	0.0400	0.2231
0.9	0.0100	0.1054
1.0	0.0000	0.0000

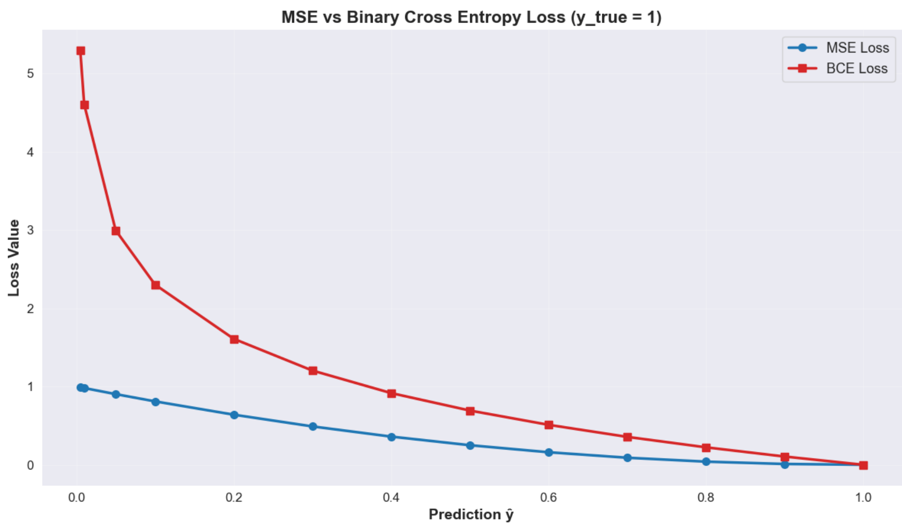

Figure 5: Comparison of MSE and BCE loss functions for $y_{\text{true}} = 1$. The plot demonstrates the distinct characteristics of each loss function: MSE exhibits quadratic behavior while BCE shows logarithmic penalty structure with asymptotic behavior near extreme predictions.

2.3 Loss Function Characteristics

2.3.1 Mean Squared Error (MSE)

- **Quadratic:** Loss grows with the square of the error
- **Symmetric:** Over- and under-predictions are penalized equally
- **Moderate:** Gradual increase in penalty with larger errors

2.3.2 Binary Cross Entropy (BCE)

- **Logarithmic:** Large penalty for predictions far from the true value
- **Asymptotic:** Very high penalty for confident wrong predictions
- **Probabilistic:** Measures likelihood, suitable for classification

2.4 Application-Specific Loss Function Selection

2.4.1 Application 1: Continuous Target Variable

Recommended Loss Function: Mean Squared Error (MSE)
Justification:

- Designed for continuous targets with penalty proportional to error squared
- Convex function with a unique minimum and closed-form solution
- Matches maximum likelihood estimation under Gaussian noise
- Smooth gradients aid optimization
- Directly interpretable in terms of variance and model performance (R^2)

2.4.2 Application 2: Binary Target Variable

Recommended Loss Function: Binary Cross Entropy (BCE)
Justification:

- Suitable for probabilistic outputs of logistic regression
- Derived from maximum likelihood estimation for Bernoulli distributions
- Logarithmic penalty: punishes confident wrong predictions
- Provides informative gradients even near extreme predictions
- Sensitive to class-specific errors, improving classification performance

2.5 Comparative Analysis

Table 8: Comparison of MSE vs BCE for different applications

Criterion	**MSE**	**BCE**
Target Type	Continuous	Binary/Categorical
Mathematical Form	Quadratic	Logarithmic
Penalty Behavior	Moderate, symmetric	Steep for extreme wrong predictions
Optimization	Convex, closed-form	Convex, iterative required
Probability Interpretation	Limited	Direct probability meaning
Sensitivity	Moderate	High for confident mistakes

3 Data Pre-processing [Code]

3.1 Feature Generation and Analysis

The data preprocessing analysis was conducted using two distinct feature types generated according to the specifications in Listing ??. The features represent different signal characteristics requiring appropriate scaling methodologies.

3.1.1 Original Feature Characteristics

The generated features exhibit distinct structural properties that influence scaling method selection:

- **Feature 1 (Sparse Signal):** Contains predominantly zero values (89% sparsity) with 10 non-zero elements following a Gaussian distribution. The sparse structure is critical for maintaining signal interpretation.
- **Feature 2 (Noise Signal):** Represents Gaussian noise with zero mean and standard deviation of 15. This feature exhibits continuous variation across all elements.

Figure 6 illustrates the original feature distributions before any preprocessing transformations.

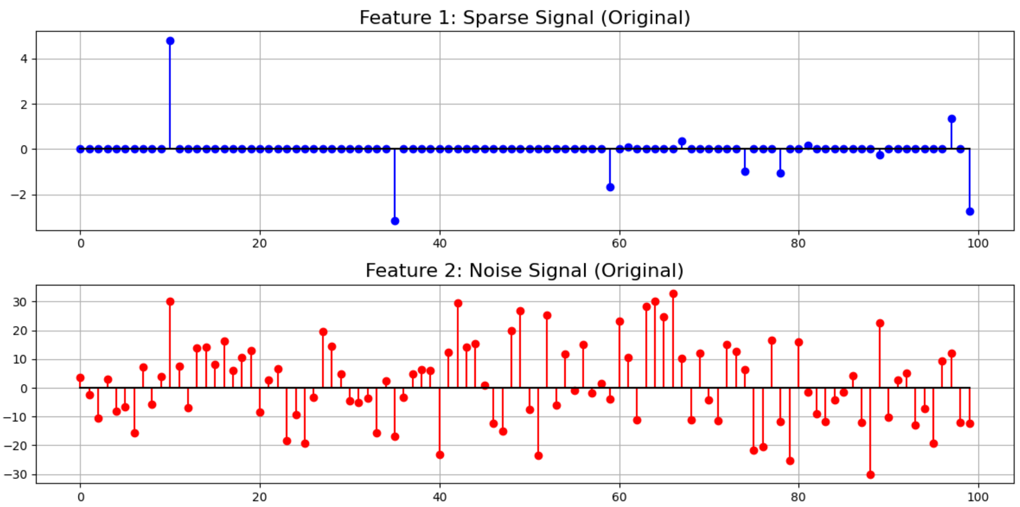

Figure 6: Original feature distributions. Top: Feature 1 showing sparse signal structure with predominantly zero values and few significant non-zero elements. Bottom: Feature 2 displaying continuous noise signal with Gaussian distribution characteristics.

3.2 Scaling Method Analysis

Three scaling methodologies were evaluated for their effectiveness in preserving feature structure while enabling optimal model performance:

3.2.1 Standard Scaling (Z-score Normalization)

Standard scaling transforms features to have zero mean and unit variance:

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma} \tag{7}$$

where μ is the mean and σ is the standard deviation.

3.2.2 Min-Max Scaling

Min-Max scaling linearly transforms features to a fixed range [0,1]:

$$x_{\text{scaled}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{8}$$

3.2.3 Max-Abs Scaling

Max-Abs scaling divides by the maximum absolute value:

$$x_{\text{scaled}} = \frac{x}{\max(|x|)} \tag{9}$$

3.3 Scaling Impact Assessment

The quantitative analysis of scaling effects on feature structure is presented in Table 9.

Table 9: Scaling impact on feature characteristics

Scaling Method	Feature 1 Zeros	Feature 2 Mean	Feature 2 Std
Original	89 zeros	—	—
Standard Scaling	0 zeros	0.00	1.00
Min-Max Scaling	1 zero	0.50	0.23
Max-Abs Scaling	89 zeros	0.05	0.43

3.4 Scaled Feature Visualization

Figure 7 demonstrates the effect of different scaling methods on both feature types, highlighting the preservation or alteration of structural properties.

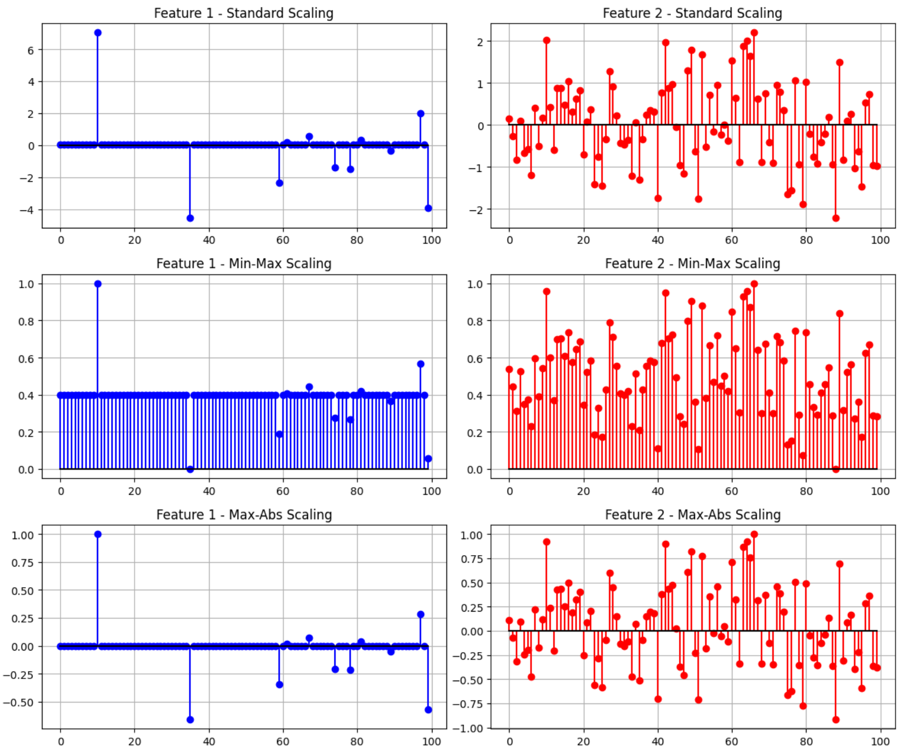

Figure 7: Comparison of scaling methods applied to both features. Left column shows Feature 1 (sparse signal) transformations, right column shows Feature 2 (noise signal) transformations. The visualization clearly demonstrates how different scaling methods affect sparsity preservation and signal characteristics.

3.5 Scaling Method Selection and Justification

3.5.1 Feature 1: Max-Abs Scaling

Reason: Preserves sparsity (keeps zeros), maintains relative magnitudes, keeps structure intact, efficient for computation, and suitable for sparse signal processing.

3.5.2 Feature 2: Standard Scaling

Reason: Centers data to zero mean and unit variance, suitable for Gaussian noise, works well with most ML algorithms, keeps values interpretable, and ensures numerical stability.

3.6 Comparative Analysis of Scaling Methods

Table 10: Scaling methods for different feature types

Scaling Method	Feature 1 (Sparse)	Feature 2 (Noise)
Standard Scaling	Not recommended: Destroys sparsity	Recommended: Standardizes Gaussian noise
Min-Max Scaling	Partially suitable: Keeps some zeros	Partially suitable: Scales to [0,1], may distort distribution
Max-Abs Scaling	Recommended: Preserves sparsity	Not optimal: Does not standardize noise distribution

4 Conclusions

This comprehensive analysis demonstrates key findings across linear regression robustness, loss function selection, and data preprocessing methodologies:

4.1 Linear Regression Analysis

- Robust loss function with $\beta = 1$ effectively handles outliers through saturation mechanism
- Model 1 significantly outperforms learned Model 2 under robust evaluation criteria
- Saturation mechanism prevents outlier dominance in parameter estimation

4.2 Loss Function Evaluation

- MSE proves optimal for continuous regression tasks with quadratic penalty structure
- BCE demonstrates superiority for binary classification problems with logarithmic penalties
- Loss function choice must align precisely with target variable nature and application requirements

4.3 Data Preprocessing Insights

- Feature-specific scaling methodologies preserve essential structural characteristics
- Max-Abs scaling maintains critical sparse structure for sparse signal processing
- Standard scaling optimizes Gaussian feature distributions for machine learning algorithms

The robust approach proves essential for reliable model selection in the presence of outliers, appropriate loss functions ensure optimal performance for specific problem domains, and proper preprocessing maintains feature integrity while enabling effective machine learning model development.

Appendix A: Source Code

The complete implementation and source code for this assignment are available at:

`https://github.com/ThiruvarankanM/Learning-from-Data-Linear-Regression-Models.git`