EN3150 Assignment 01
Learning from data and related challenges and linear models for regression
B.Sc. Engineering, Semester 05

M. T. U. Sampath K. Perera
Department of Electronic & Telecommunication Engineering
University of Moratuwa

August 4, 2025

1 Linear regression impact on outliers

1. You are given set of data points related to independent variable (x) and dependent variable (y) in Table 1.

Table 1: Data set.

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

2. Use all data given in Table 1 to find a linear regression model. Plot x, y as a scatter plot and plot your linear regression model in the same scatter plot. [10 marks]

3. You are given two linear models as follows.

   - Model 1: $y = -4x + 12$
   - Model 2: $y = -3.55x + 3.91$

Here, model 2 is your linear regression model which is learned in task 2.
A robust estimator is introduced to reduce the impact of the outliers. The robust estimator finds model parameters which minimize the following loss function

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

Here, θ represents model parameters, β is a hyper parameter and number of data samples $N = 10$, respectively. Note the y_i and $\hat{y}_i$ are true and predicted i−th data sample, respectively.

4. For the given two models in task 3, calculate the loss function $L(\theta, \beta)$ values for all data samples using eq. (1) for $\beta = 1$, $\beta = 10^{-6}$ and $\beta = 10^3$ (you may use a computer program to calculate this). [20 marks]

5. What is the suitable β value to mitigate the impact of the outliers. Justify your answer. [40 marks]

6. Utilizing this robust estimator with selected β value, determine the most suitable model from the models specified in task 3 for the provided dataset. Justify your selection. [30 marks]

7. How does this robust estimator reduce the impact of the outliers? [20 marks]

8. Identify another loss function that can be used for this robust estimator. [10 marks]

2 Loss Function

Suppose you have two applications namely Application 1 and 2.

- **Application 1:** The dependent variable is continuous.

- **Application 2:** The dependent variable is discrete and binary (only takes values 0 or 1 i.e., $y \in \{0, 1\}$).

You plan to train:

- A **Linear Regression** model for Application 1.

- A **Logistic Regression** model for Application 2.

Two common loss functions are:

- **Mean Squared Error (MSE):**

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

- **Binary Cross Entropy (BCE):**

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

1. Fill the following table and plot the both loss functions. [10 marks]

Table 2: MSE and BCE loss values for different predictions when $y = 1$.

True $y = 1$	Prediction $\hat{y}$	MSE	BCE
1	0.005		
1	0.01		
1	0.05		
1	0.1		
1	0.2		
1	0.3		
1	0.4		
1	0.5		
1	0.6		
1	0.7		
1	0.8		
1	0.9		
1	1.0		

2. Which loss function (MSE or BCE) would you select for each of the applications (Application 1 and 2)? Justify your answer. [30 marks]

3 Data pre-processing

1. Generate feature values of two features using the code given in listing 1. Considering scaling methods of (a) standard scaling, (b) min-max scaling, and (c) max-abs scaling. Select one scaling method for feature 1 and 2, ensuring that the chosen method preserves the structure/properties of the features. Justify your answer. [30 marks]

```
import numpy as np
import matplotlib.pyplot as plt

def generate_signal(signal_length, num_nonzero):

signal = np.zeros(signal_length)
nonzero_indices = np.random.choice(signal_length, num_nonzero,
    replace=False)
nonzero_values = 10*np.random.randn(num_nonzero)
signal[nonzero_indices] = nonzero_values
```

```
    return signal

signal_length = 100  # Total length of the signal
num_nonzero = 10    # Number of non-zero elements in the
    signal
your_index_no= # Enter your index no without english letters
    and without leading zeros
sparse_signal = generate_signal(signal_length, num_nonzero)
sparse_signal[10] = (your_index_no % 10)*2 + 10
if  your_index_no % 10 == 0:
sparse_signal[10] = np.random.randn(1) + 30
sparse_signal=sparse_signal/5
epsilon = np.random.normal(0, 15, signal_length )


#epsilon=epsilon[:, np.newaxis]
plt.figure(figsize=(15,10))
plt.subplot(2, 1, 1)
plt.xlim(0, signal_length)
plt.title("Feature 1", fontsize=18)
plt.xticks(fontsize=18)  # Adjust x-axis tick label font size
plt.yticks(fontsize=18)
plt.stem(sparse_signal)
plt.subplot(2, 1, 2)
plt.xlim(0, signal_length)
plt.title("Feature 2", fontsize=18)
plt.stem(epsilon)
plt.xticks(fontsize=18)  # Adjust x-axis tick label font size
plt.yticks(fontsize=18)

plt.show()
```

Listing 1: Feature data generation.

4 Additional Resources

1. Scikit-learn preprocessing data
2. Introduction to sparsity in signal processing
3. sklearn linear regression

5 Submission

- Upload a report as a pdf file named as "your_indexno_EN3150_A01.pdf". Include the index number and the name within the report as well. Please include all your answers in the report.

- Pay careful attention to formatting such as font size, spacing, and margins.
- Include a title page with necessary information (e.g., title, author, date, index no).
- Use consistent and professional formatting throughout the document.
- Plagiarism will be checked and in cases of plagiarism, an extra penalty of 50% will be applied. In case of copying from each other, both parties involved will receive a grade of zero for the assignment. Academic integrity is of utmost importance, and any form of plagiarism[1] or cheating will not be tolerated.
- An extra penalty of 15% is applied for late submission.

[1] https://en.wikipedia.org/wiki/Plagiarism