DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

UNIVERSITY OF MORATUWA

# EN3150 - Pattern Recognition


## Assignment 01

Learning from data,related challenges and linear models for regression


| | |
|---|---|
| Name: | Mirihagalla M.K.D.M. |
| Index: | 200397A |
| Date: | 9th Sep. 2023. |

# Table of Contents

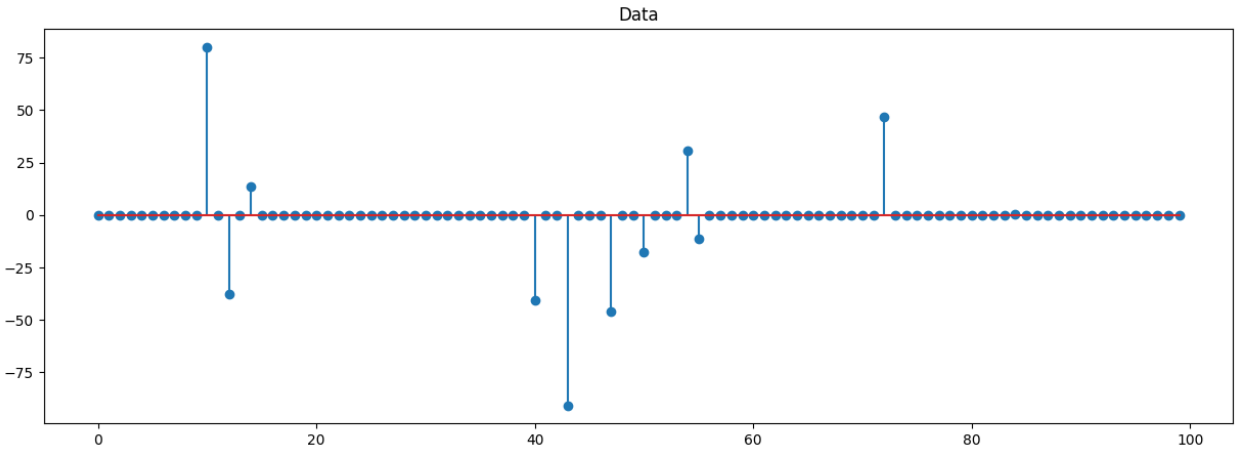
# Part 1- Data Preprocessing

## Data Generation

Using the Index number = 200397A I have generated the data and plotted them as follows.

```python
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 = 50*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=200397 # Enter without english letter and without leading zeros
signal = generate_signal(signal_length, num_nonzero)
signal[10] = (your_index_no % 10)*10 + 10
if  your_index_no % 10 == 0:
  signal[10] = np.random.randn(1) + 30


signal=signal.reshape(signal_length,1)
plt.figure(figsize=(15,5))
plt.subplot(1, 1, 1)
plt.title("Data")
plt.stem(signal)
```

## Plot generated data.



## Applying Normalization Methods.

- MaxAbsScaler (preprocessing. MaxAbsScaler() from sklearn. preprocessing)
- Implementation of min-max and standard normalization using handwritten functions.

**Normalization using sklearn**

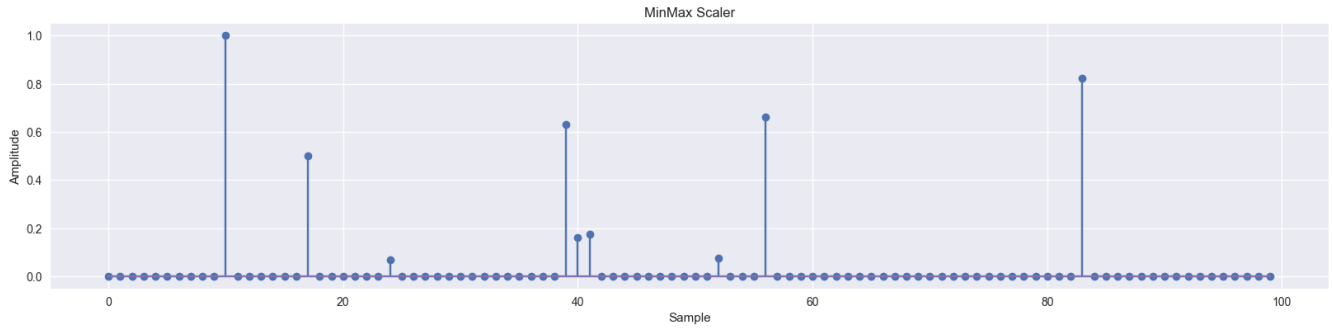
```
import numpy as np
from sklearn.preprocessing import MaxAbsScaler
max_abs_scaler = MaxAbsScaler()

signal_1 = max_abs_scaler.fit_transform(signal)  # This is the Maxabs scaler
```

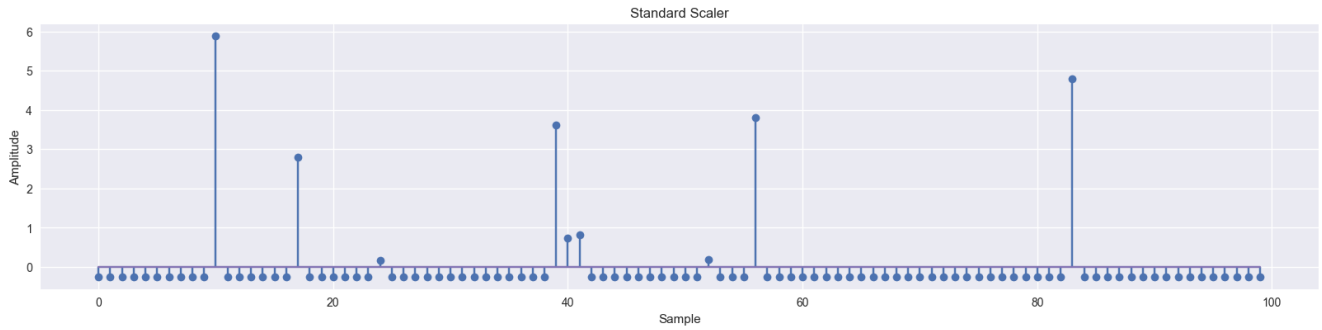
**Defining MinMaxScaler and Standard scaler functions**

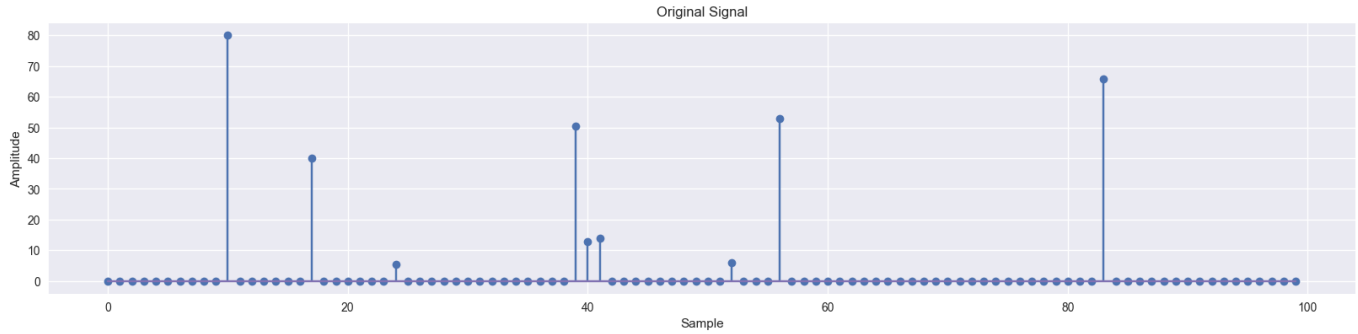
+ Code + Markdown

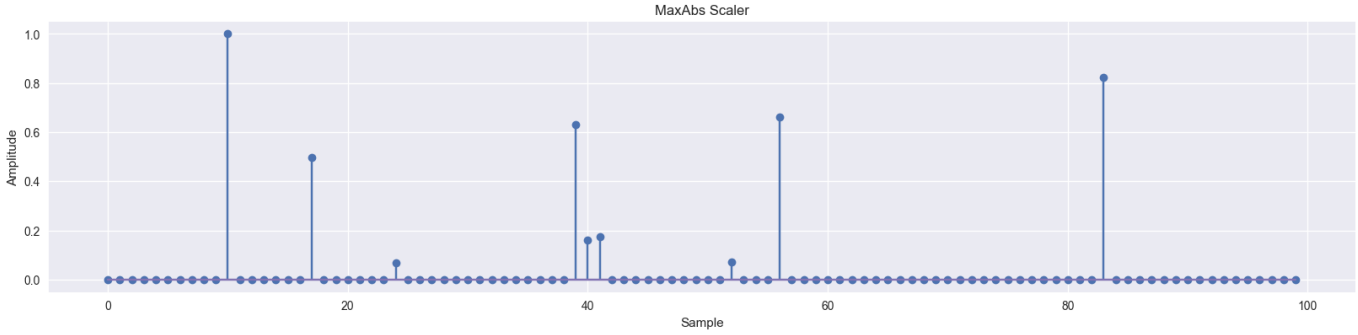
```
def min_max_scale_function(data):
    min_val = np.min(data)
    max_val = np.max(data)
    data = (data-min_val)/(max_val-min_val)
    return data

def standard_scaler_function(data):
    miu = np.mean(data)
    stdiv = np.std(data)
    return (data-miu)/stdiv

signal_2 = min_max_scale_function(signal)
signal_3 = standard_scaler_function(signal)
```

## Visualize the data before and after normalization.






## Non-Zero Elements

Non Zero Elements of the Signal (Before Normalization): 10

Non Zero Elements of the Signal (After MaxAbs Normalization): 10

Non Zero Elements of the Signal (After MinMax Normalization): 99

Non Zero Elements of the Signal (After Standard Normalization): 100

## Normalization scales the data, Impact on structure of the data.

6. Compare how each normalization method scales the data and its impact on structure of the data.

### Maximum Absolute Normalization

In the signal given we can see the data has been distributed over the range of [0,1] since we divide the values by the maximum absolute value of the data set. This normalization method is useful when we need to preserve the shape of the data distribution. Even though the values get mapped into the range [0,1], it keeps the original data distribution shape steady.

### Min-Max Normalization

This normalization method always maps the signal amplitude to the range of [0,1]. This normalization method significantly changes the shape of the distribution and does not preserve the original shape of the distribution. This is useful when we need to compress the data ranges into specific ranges.

### Standard Normalization

This normalization typically shifts the data to zero, meaning the standard deviation of 1. Therefore, this follows a normal distribution. This can be useful when we need to compare datasets with different means and standard deviations. This does not preserve the original distribution and changes the distribution shape significantly.

## Effects of each normalization method

7. Discuss the effects of each normalization method on the data's distribution, structure, and scale. Which normalization approach do you recommend for this kind of data and what is the reason behind this?

| | MAX-ABS NORMALIZATION | MIN-MAX NORMALIZATION | STANDARD NORMALIZATION |
|---|---|---|---|
| DISTRIBUTION | typically preserves the shape of the distribution, as it scales all values by the maximum absolute value. It maintains the relative ordering of data points | compresses the range of values to [0, 1]. This can lead to a significant change in the distribution shape | aims to make the data follow a standard normal distribution (mean=0, std=1). It significantly changes the distribution shape if it deviates from a normal distribution. |
| STRUCTURE | minimally affected. The data points remain in their original order, and only their scale is adjusted. | change considerably, as values are compressed into a smaller range. The ordering of data points is preserved, but the scale is entirely transformed. | Changed, as it centers the data around zero and adjusts the spread of values. The original ordering of data points is maintained. |
| SCALE | The scale of the data is adjusted. more like a scaling factor. | adjusted to a fixed range [0,1], useful for comparing datasets with different scales. | adjusted to have a mean of 0 and a standard deviation of 1. useful for statistical analysis. Original scale is broken. |

# Part 2 – Linear regression on real world data

## Procedure and parameters

First, we need to import the data from the .csv file as a pandas' data frame (that is easy to manage). Then we split them into training set and testing set getting 20% for testing and 80% for training. Then we create a linear regression model (object) and we train the model parameters using the data we split.

After training the model I could get the co-efficient for each feature as follows.

```
Co-efficient corresponding to independent variables
---------------------------------------------------

TV          :     0.044729517468716326
radio       :     0.18919505423437652
newspaper   :     0.0027611143413671935
Intercept   :     2.979067338122629
```

## Performance Metrices

After Evaluating the model depending on the following statistics, I could get the following.

```
_____________________________________________________
RSE         :     1.8779709363435915
MSE         :     3.1740973539761033
R2          :     0.899438024100912
_____________________________________________________
Standard Error (SE) for each feature
---------------------------------------------------
TV          :     0.0014563733599258976
radio       :     0.010864334024725128
newspaper   :     0.007922459304137928
_____________________________________________________
T-Statistics for each feature
---------------------------------------------------
TV          :     36.466716506203646
radio       :     20.138066864833657
newspaper   :     3.01108835995514
_____________________________________________________
P-Values for each feature
---------------------------------------------------
TV          :     1.5241085751107518e-78
radio       :     2.3135664129241866e-45
newspaper   :     0.0030344754530055016
```

To calculate the given statics these are the typical parts of the code.

```
# Residual sum of squares (RSS)
RSS = np.sum((y_test_predicted-y_test)**2)
# Residual Standard Error (RSE)
N=len(y_test)
d=(len(model.coef_)+1) # including interecept
RSE = np.sqrt (RSS/(N-d))
#Mean Squared Error
MSE = mean_squared_error(y_test,y_test_predicted)
#R2 Score
R2 = r2_score(y_test,y_test_predicted)
#creating ols model
ols_model = sm.OLS(y_train,X_train).fit()
#Standard Error (SE)
params = ols_model.params
SE = ols_model.bse
#T-Statistics
t_values = ols_model.tvalues
#P-Values
p_values = ols_model.pvalues
```

## Relationship between advertising budgets and sales

5. Is there a relationship between advertising budgets and sales?

There is a relation with TV and radio but there is unlikely to have a relation with newspapers.

When we look at the p values it explains that TV and radio have extremely low values compared to newspapers. Therefor we can say that there is strong evidence to deny the null hypothesis in TV and radio. Also, t values, newspapers have low t value that also confirms that.

6. Which independent variable contributes highly on sales?

TV.

Because it has the lowest p value compared to other features. ($1.5 \times 10^{-78}$)

7. One may argue that possibly, allocating 25, 000 dollars both television advertising and radio advertising individually (i.e., 25, 000 dollars for TV and 25, 000 dollars for radio) yields higher sales compared to investing 50, 000 dollars in either television or radio advertising individually. Based on your trained model, comment on this argument. Here, assume that the budget allocated for newspapers is zero.

No, it is not always true.

Let us consider three cases,

- Only TV – 50 000$
- Only Radio – 50 000$
- Both TV and Radio 25 000$ each

Now let us use the model to predict each case.

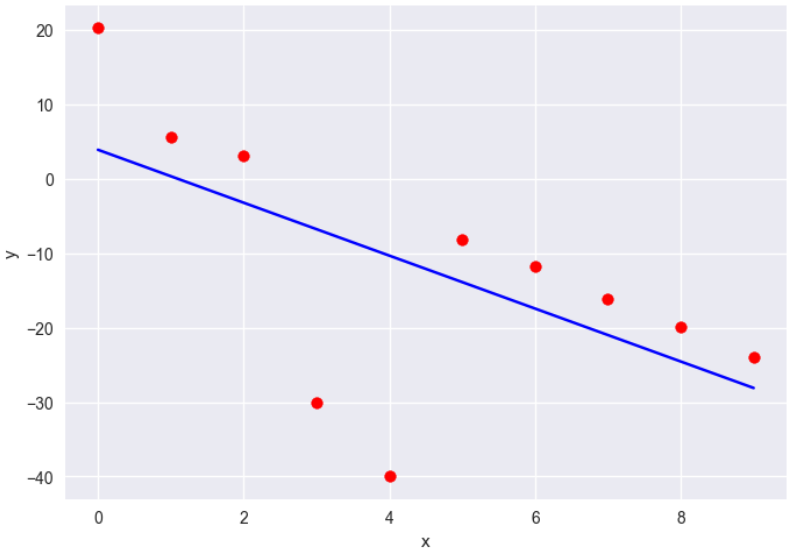
```
# x = ["TV","Radio","Newspaper"]
x_test_cases = np.array([
    [50_000,0,0],
    [0,50_000,0],
    [25_000,25_000,0]
    ])

y_test_cases = np.array([model.predict([x_test_case]) for x_test_case in x_test_cases])

for i in range(len(x_test_cases)):
    print(f"Predicted Sales for {x_test_cases[i]} is {y_test_cases[i]}")
```

```
Predicted Sales for [50000     0     0] is [2239.45494077]
Predicted Sales for [    0 50000     0] is [9462.73177906]
Predicted Sales for [25000 25000     0] is [5851.09335992]
```

According to predicted values we can see that if we use 50 000$ only for radios we can get a higher value rather than doing both.

## Part 3: Impact on Outliers

2. Plot x, y as a scatter and plot your linear regression model in the same scatter plot.



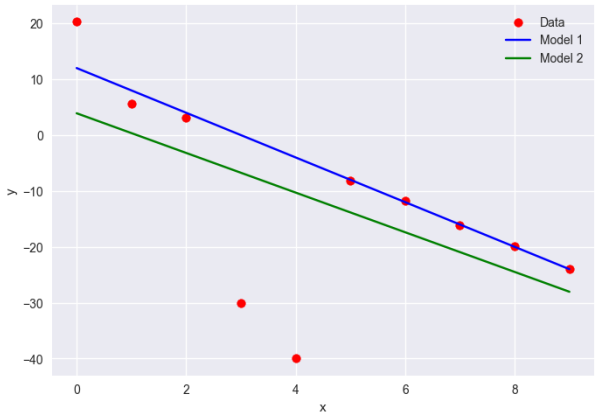
4. using the loss function given we can calculate the loss for each model

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

```
Loss for Model 1:  0.435416262490386
Loss for Model 2:  0.9728470518681676
```

5.

The first model seems to have a lower loss value that indicates it is a good model. If we plot the 2 regression models, we can see the model 1 has evaded the outliers when finding the regression model.

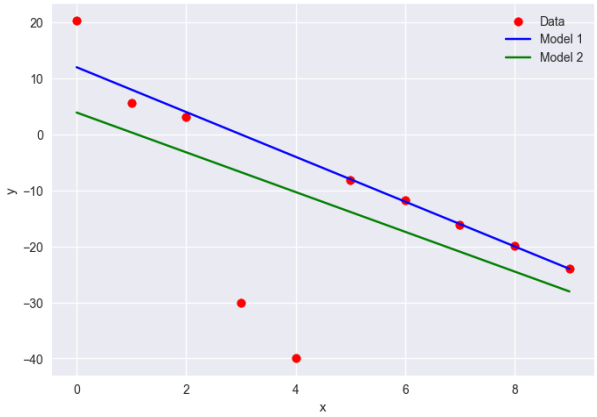



With the plot we can also justify that the model 1 is more suitable. Model 2 has deviated from the datapoints because it has considered the outliers when minimizing the loss.

6.

The robust estimator makes sure that extreme or unusual data points (outliers) do not mess up our model. It does this by giving less importance to those unusual points when figuring out the model. β is a hyper-parameter. If we make it bigger the model cares less about outliers, and if make it smaller, the model pays more attention to outliers. So, by adjusting this β, we can control how much the unusual data points affect our model. This helps us find a balance between making our model resistant to outliers and fitting it well to the rest of the data.

7.




8.

In equation (4), β controls how much outliers affect the model.

- A big β makes the model care less about outliers, making it more stable but potentially less accurate.
- A small β makes the model pay more attention to outliers, which can lead to overfitting.

Picking the right β depends on data set and we need to balance between stability and accuracy.