University of Moratuwa, Sri Lanka

Faculty of Engineering

Department of Electronic and Telecommunication Engineering
Semester 5 (Intake 2021)

EN3150 - Pattern Recognition

Assignment 01 : Learning from data and related challenges and linear models for regression

Kodikara U. S. S.
210293K

Contents

1 Data pre-processing **3**
1.1 Scaling method . . . 3

2 Learning from data **4**
2.1 Random data split . . . 4
2.2 Linear regression . . . 4
2.3 Increase the number of data samples . . . 5

3 Linear regression on real world data **6**
3.2 Independent variables and Dependent variables . . . 6
3.3 Possibility of applying linear regression . . . 6
3.4 Removing missing values . . . 6
3.5 Selecting features . . . 7
3.6 Splitting data points . . . 7
3.7 Linear regression model . . . 7
3.8 Highly contributed independent variable . . . 8
3.9 Train a regression model for selected features . . . 8
3.10 Calculations . . . 9
3.11 Discarding features based on p-value . . . 9

4 Performance Evaluation of Linear Regression **10**
4.1 Residual Standard Error (RSE) for Models A and B . . . 10
4.2 Based on RSE, which model performs better? . . . 10
4.3 Compute R-squared (R^2) for Models A and B . . . 10
4.4 Between RSE and R^2, which performance metric is more fair for comparing two models and why? . . . 11

5 Linear regression impact on outliers **11**
5.1 What happens when $a \to 0$? . . . 11
5.2 Suppose we need to minimize the influence of data points with $|r_i| \geq 40$. What value(s) of "a" and what function(s) would you choose, and why? . . . 11

1 Data pre-processing

1.1 Scaling method

To select the most appropriate scaling method for the two features shown in the figure, let's consider the nature of the features and the scaling methods.

- **Feature 1:**
  - **Characteristics:** Feature 1 has a majority of its values clustered around 0, with a few significant outliers reaching values above 5 and below -1.
  - **Scaling Consideration:**
    - **Standard Scaling (Z-score normalization):** This method centers the data around the mean and scales it according to the standard deviation. It is sensitive to outliers, which could make it less suitable for Feature 1 due to its extreme values.
    - **Min-Max Scaling:** This method scales the data to a fixed range, typically [0, 1]. While it normalizes the feature, it could compress the majority of the data points that are close to 0, exaggerating the impact of outliers.
    - **Max-Abs Scaling:** This method scales the data based on the maximum absolute value of the feature, preserving the sparsity of the data. It is more robust to outliers and preserves the structure of sparse data.
  - **Conclusion:** Given the sparse nature of Feature 1 and its outliers, **Max-Abs Scaling** would be the most suitable option. This method will scale the data while maintaining the integrity of the sparse structure, minimizing the impact of outliers without compressing the majority of the data.
- **Feature 2:**
  - **Characteristics:** Feature 2 shows a wider spread of values, ranging from around -40 to nearly 40, with a more uniform distribution of data points.
  - **Scaling Consideration:**
    - **Standard Scaling:** Since Feature 2 has a relatively normal distribution of values, standard scaling would effectively normalize the data by centering it around the mean and scaling it by the standard deviation.
    - **Min-Max Scaling:** While this could normalize the data to a [0, 1] range, it might not preserve the distribution's variance well.
    - **Max-Abs Scaling:** This method would scale Feature 2 based on its maximum absolute value, similar to Feature 1, but this might not be ideal if preserving the feature's variance is important.
  - **Conclusion:** Given that Feature 2 has a relatively normal distribution, **Standard Scaling** would be the most suitable option. This method will effectively normalize the data while preserving its variance and spread.

Final Conclusion:
Feature 1: Use **Max-Abs Scaling** to preserve the sparse structure and minimize the impact of outliers.
Feature 2: Use **Standard Scaling** to normalize the data while preserving its variance.

This combination of scaling methods ensures that the unique properties of each feature are maintained, enabling more effective downstream analysis or modeling.

2 Learning from data

2.1 Random data split

The function `train_test_split` splits the dataset into two subsets: training and testing data. This process is inherently random, and the randomness is controlled by the `random_state` argument. In the first line of Listing 2, a random integer smaller than 104 is generated, and this integer is used as the `random_state`.
When I ran the code four times, I observed that the plots for the training and testing data were different in each run. This difference occurs because each time the code is executed, a different random integer is chosen as the `random_state`, leading to a different random shuffle of the data before it is split. As a result, the specific points selected for training and testing vary, creating different plots each time the code is run.
The use of a random integer as the `random_state` is why the training and testing datasets, and consequently the scatter plots, differ across the four runs. If we want consistent plots across multiple runs,we would need to fix the `random_state` to a specific value instead of generating a random one each time.

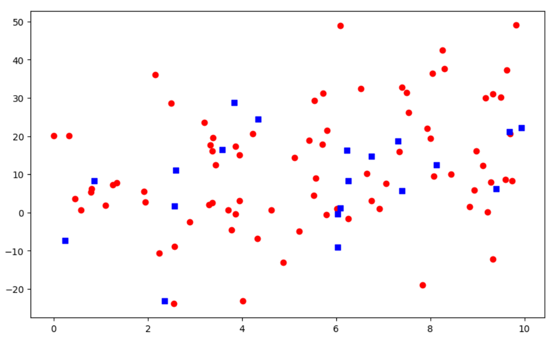

Figure 1: First output

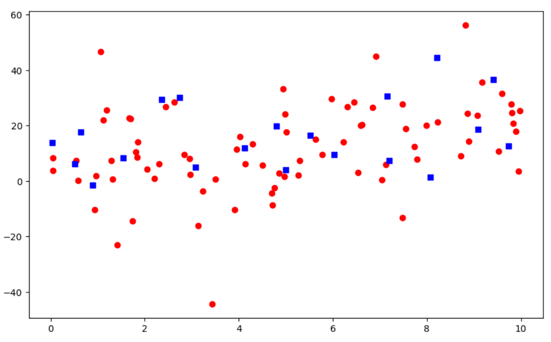

Figure 2: Second output

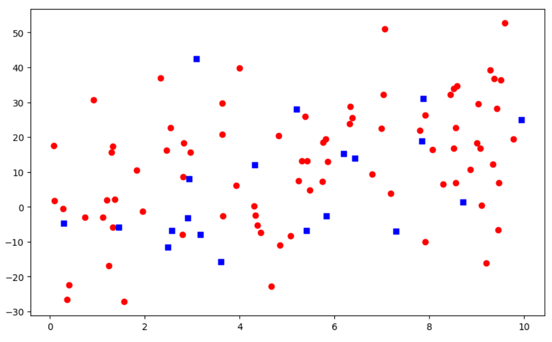

Figure 3: Third output

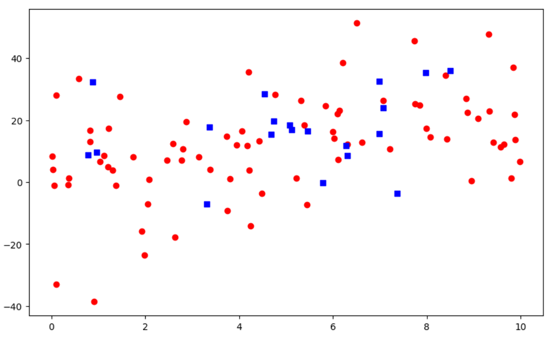

Figure 4: Fourth output

2.2 Linear regression

In Listing 3, ten different linear regression models are trained, each using a different randomly split training dataset. The randomness in the data split is controlled by the `random_state`, which is set to a different random integer in each iteration. Since the training dataset varies with each split, the linear regression model is trained on different data each time. As a result, the model parameters (slope and intercept) differ across the ten instances, leading to different prediction lines.
The graph displays ten different lines, each corresponding to a linear regression model trained on a

distinct dataset split. The variation in the dataset splits is the reason why each model has different parameters and thus produces a different line on the plot.

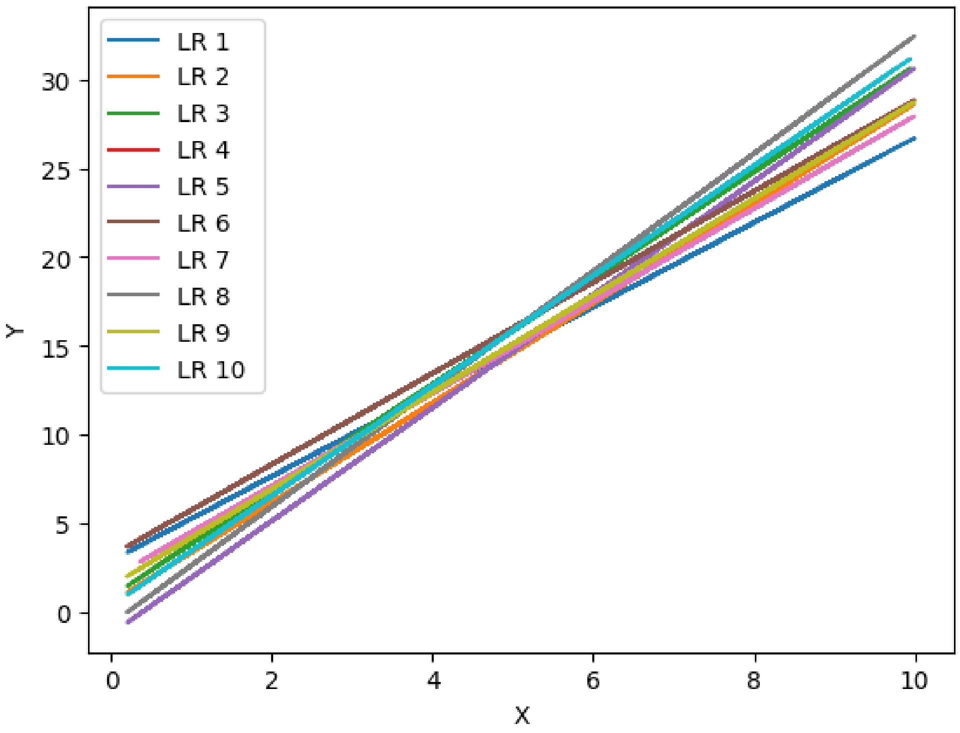

Figure 5: Linear regression models by different data splits

2.3 Increase the number of data samples

When the number of samples increases to 10,000, the linear regression model becomes more stable and consistent across different runs. With a small sample size of 100, the model is more prone to overfitting, making the regression lines vary significantly with each random split due to the noise in the data. However, with 10,000 samples, the model is better able to generalize, and the regression lines across different runs are almost identical, closely approximating the true underlying relationship $Y = 3 + 2X$. This happens because the larger dataset dilutes the effect of random noise, resulting in a more accurate and reliable model estimation.

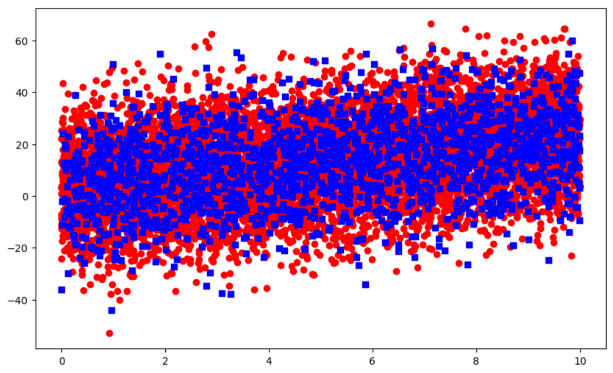

Figure 6: Data splits with 10 000 samples

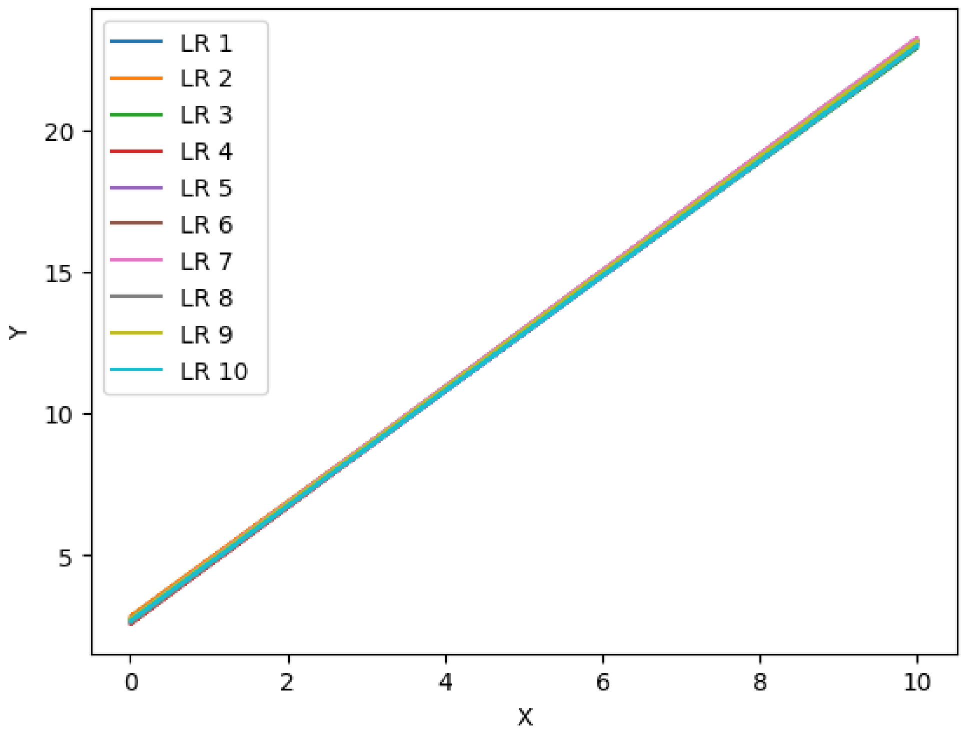

Figure 7: Linear regression models by different data splits for 10 000 samples

3 Linear regression on real world data

3.2 Independent variables and Dependent variables

- Number of Dependent Variables: 2
- Number of Independent Variables: 33

3.3 Possibility of applying linear regression

No
linear regression cannot be performed directly with categorical features. Categorical variables need to be encoded into numerical format first. For example, since 'Age' has an inherent order, label encoding would be suitable. However, for 'Gender' and 'Ethnicity,' which don't have an ordinal relationship, one-hot encoding is recommended. Each method has its own set of advantages and disadvantages depending on the nature of the categorical data.

3.4 Removing missing values

No, the provided approach is not correct. Dropping missing values separately from X and y can lead to misalignment between the features and the target variables, causing incorrect mappings.
Instead, a better approach is to first concatenate X and y into a single DataFrame, remove rows with missing values, and then split them back into X and y.

```python
import pandas as pd

# Concatenate X and y
df = pd.concat([X, y], axis=1)

# Drop rows with missing values
df = df.dropna()

# Separate X and y again
X = df.drop(columns=y.columns)
y = df[y.columns]
```

Listing 1: Correct approach to handle missing values in X and y

3.5 Selecting features

```python
y_sel = df['aveOralM']

# Selecting the independent features
X_sel = df[['Age', 'Humidity','T_Max1', 'T_OR1', 'T_atm']]
```

Listing 2: Selected features

3.6 Splitting data points

```python
X_train , X_test , Y_train , Y_test = train_test_split (X_sel,
    y_sel , test_size =0.2 , random_state = np . random . randint
    (104) )
```

Listing 3: Data split

3.7 Linear regression model

```python
import pandas as pd
df_encoded = pd.get_dummies(X, columns=['Age'], drop_first=True)
X_sel = df_encoded[['Humidity','T_Max1', 'T_OR1', 'T_atm',] + [
    col for col in df_encoded.columns if col.startswith('Age_')]]
X_train , X_test , Y_train , Y_test = train_test_split (X_sel,
    y_sel , test_size =0.2 , random_state = np . random . randint
    (104) )
model = LinearRegression ()
model . fit ( X_train , Y_train )
coefficients = model.coef_
```

Listing 4: Linear regression model

- **Model Coefficients:**

Feature	Coefficient
Humidity	0.000981
T_Max1	0.881195
T_OR1	0.020920
T_atm	-0.043686
Age_21-25	0.008337
Age_21-30	0.093078
Age_26-30	-0.009939
Age_31-40	-0.011477
Age_41-50	0.018673
Age_51-60	-0.010355
Age_¿60	-0.032256

- **Estimated Intercept:** 5.510381

3.8 Highly contributed independent variable

```
    top_feature_index = np.argmax(np.abs(coefficients))
    top_feature = X_sel.columns[top_feature_index]
    top_coefficient = coefficients[top_feature_index]
```

Listing 5: Highly contributed independent variable

T_Max1
Coefficient: 0.8811945763490852
Since it has the highest coefficient, T_max contributes the most to the dependent variable.

3.9 Train a regression model for selected features

```
    y_sel = df['aveOralM']
    # Selecting the independent features
    X_sel = df[['T_OR1', 'T_OR_Max1', 'T_FHC_Max1', 'T_FH_Max1']]
```

Listing 6: Selection of features

- **Estimated Coefficients:**

Feature	Coefficient
T_OR1	0.205457
T_OR_Max1	0.348196
T_FHC_Max1	-0.083718
T_FH_Max1	0.376564

- **Estimated Intercept:** 6.793556

3.10 Calculations

- **Residuals and R-squared:**

Metric	**Value**
Residual Sum of Squares (RSS)	77.974491
Residual Standard Error (RSE)	0.310457
Mean Squared Error (MSE)	0.095792
R-squared (R2) statistic	0.651338

- **Standard Errors for each feature:**

Feature	**Standard Error**
T_OR1	0.892586
T_OR_Max1	0.890373
T_FHC_Max1	0.043574
T_FH_Max1	0.048550

- **T-statistics for each feature:**

Feature	**T-statistic**
T_OR1	0.230183
T_OR_Max1	0.391068
T_FHC_Max1	-1.921303
T_FH_Max1	7.756138

- **P-values for each feature:**

Feature	**P-value**
T_OR1	0.818008
T_OR_Max1	0.695849
T_FHC_Max1	0.055044
T_FH_Max1	2.633993e-14

3.11 Discarding features based on p-value

The regression results reveal that the p-values for the features **T_OR1** - 0.818008 and **T_OR_Max1** - 0.695849 exceed 0.05. This suggests that these features do not significantly contribute to the model at the 5% significance level. Consequently, their influence on the dependent variable is minimal, indicating that they might not be essential predictors for the model.

4 Performance Evaluation of Linear Regression

4.1 Residual Standard Error (RSE) for Models A and B

Residual Standard Error (RSE) is calculated using the formula:

$$\text{RSE} = \sqrt{\frac{\text{SSE}}{N - p}}$$

where:

- **SSE** is the Sum of Squared Errors.
- **N** is the number of data samples.
- **p** is the number of parameters in the model (including the intercept).

Given data:

- **Model A**: SSE = 9, $N = 10000$, $p = 3$ (intercept + w_1, w_2)
- **Model B**: SSE = 2, $N = 10000$, $p = 5$ (intercept + w_1, w_2, w_3, w_4)

Calculations: For Model A:

$$\text{RSE}_A = \sqrt{\frac{9}{10000 - 3}} = \sqrt{\frac{9}{9997}} \approx \sqrt{0.00090027} \approx 0.03$$

For Model B:

$$\text{RSE}_B = \sqrt{\frac{2}{10000 - 5}} = \sqrt{\frac{2}{9995}} \approx \sqrt{0.0002001} \approx 0.01414$$

4.2 Based on RSE, which model performs better?

Since Model B has a lower RSE (0.01414) compared to Model A (0.03), **Model B** performs better.

4.3 Compute R-squared (R^2) for Models A and B

The R^2 value is calculated using the formula:

$$R^2 = 1 - \frac{\text{SSE}}{\text{TSS}}$$

Given data:

- **Model A**: SSE = 9, TSS = 90
- **Model B**: SSE = 2, TSS = 10

Calculations: For Model A:

$$R_A^2 = 1 - \frac{9}{90} = 1 - 0.1 = 0.9$$

For Model B:

$$R_B^2 = 1 - \frac{2}{10} = 1 - 0.2 = 0.8$$

Based on R^2, which model performs better?
Model A has a higher R^2 value (0.9) compared to **Model B** (0.8), indicating that **Model A** performs better in explaining the variability in the data.

4.4 Between RSE and R^2, which performance metric is more fair for comparing two models and why?

R^2 is generally used to evaluate how well the independent variables explain the variability in the dependent variable. However, R^2 can be misleading when comparing models with different numbers of predictors, as it tends to increase with the addition of more variables, even if those variables don't significantly improve the model.

On the other hand, **RSE** provides a direct measure of the model's prediction accuracy in terms of the standard deviation of the residuals, accounting for the number of predictors used. Thus, **RSE** can be considered a more fair comparison metric when evaluating models with different numbers of predictors because it penalizes the addition of unnecessary variables, making **RSE** more reliable for model comparison in this context.

5 Linear regression impact on outliers

5.1 What happens when $a \to 0$?

When the parameter a approaches 0, both modified loss functions $L_1(w)$ and $L_2(w)$ behave differently, focusing more on smaller residuals and potentially ignoring outliers.

For $L_1(w)$:

$$L_1(w) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{r_i^2}{a^2 + r_i^2}\right)$$

As $a \to 0$, the term a^2 in the denominator becomes negligible compared to r_i^2, especially for larger r_i, leading $L_1(w)$ to approach:

$$L_1(w) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{r_i^2}{r_i^2}\right) = \frac{1}{N}\sum_{i=1}^{N} 1 = 1$$

Thus, for large residuals, $L_1(w)$ becomes insensitive to the magnitude of r_i, effectively treating large and small residuals the same, meaning it does not penalize outliers as strongly.

For $L_2(w)$:

$$L_2(w) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \exp\left(\frac{-2|r_i|}{a}\right)\right)$$

As $a \to 0$, the exponential term becomes very small for large r_i, causing $L_2(w)$ to approximate:

$$L_2(w) \approx \frac{1}{N}\sum_{i=1}^{N}(1 - 0) = 1$$

Similar to $L_1(w)$, $L_2(w)$ also becomes insensitive to large residuals when a is close to 0, meaning it would also not penalize outliers effectively.

Conclusion: Both loss functions lose sensitivity to large residuals as a approaches 0, treating large and small residuals similarly, which makes them less effective in handling outliers.

5.2 Suppose we need to minimize the influence of data points with $|r_i| \geq 40$. What value(s) of "a" and what function(s) would you choose, and why?

To minimize the influence of data points with $|r_i| \geq 40$, we want a loss function that sharply reduces the contribution of large residuals. From Figure 2, we observe the behavior of $L_1(w)$ and $L_2(w)$ for different values of a.

For $L_1(w)$:

- As a increases, $L_1(w)$ becomes flatter, meaning the function becomes less sensitive to changes in residuals, especially large ones.
- $L_1(w)$ for $a = 100$ (brown line with circles) shows the least variation, meaning it will significantly reduce the influence of large residuals.

For $L_2(w)$:

- Similarly, $L_2(w)$ for larger a values (like 100) also shows a flatter curve, making it less sensitive to large residuals.
- The green line (stars) for $a = 25$ shows that $L_2(w)$ can start penalizing large residuals more effectively at smaller a compared to $L_1(w)$, making it a better candidate for suppressing the impact of large residuals.

Recommended Value and Function:

- **Function**: $L_2(w)$ would be more effective since it gives a more controlled reduction in the influence of large residuals.
- **Value of** a: Based on Figure 2, $a = 25$ (green line with stars) is a good choice. It significantly penalizes large residuals while maintaining sensitivity to smaller residuals, effectively reducing the impact of outliers.