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

EN3150 - Pattern Recognition

Name:	**Gunawardane E.R.N.H.**
Index No:	210200C
Date	03.09.2024

1 Data pre-processing

1. max-abs is the best scaling method in this scenario. In analyzing feature 1 we can see that feature 1 contains many sparse data points. With max-abs sparse data points wouldn't change so the structure won't change in the data distribution. In feature 2 overshoots are not seen. Therefore it is ok to use max-abs. Since there are no visible overshoots, it won't affect the scaling with max-abs.

2 Learning from data

2. Training and testing data sets are different in each run. This is because when the `random_state` parameter is set to a specific integer, the shuffling of data will be consistent across each run. This consistency is important for data reproducibility. However, if we change the `random_state` parameter to a different value in a random manner, the data will be shuffled differently in each run.

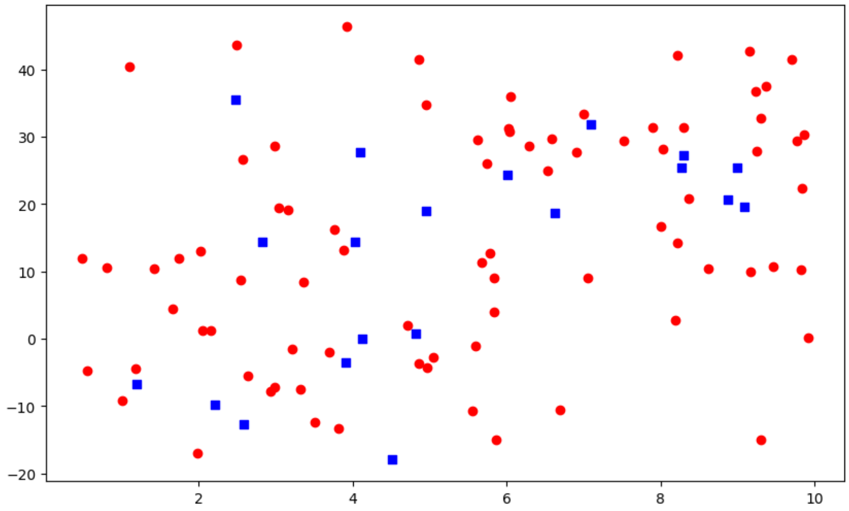


Figure 1: Random data generation and visualization

3. At each iteration the model is fitted to a different data set. Since the model created will vary with the data set the linear regression model is different from one instance to other.

4. The models created are focused on many data points than in the 100 data point instance. Since every model trained in each iteration is trained for 8000 data points(Assuming 80 percent is for training) the variation of the weights of the models are very less compared to the 100 data point model. Therefore, the 10 models that are trained tend to converge closely to one another.

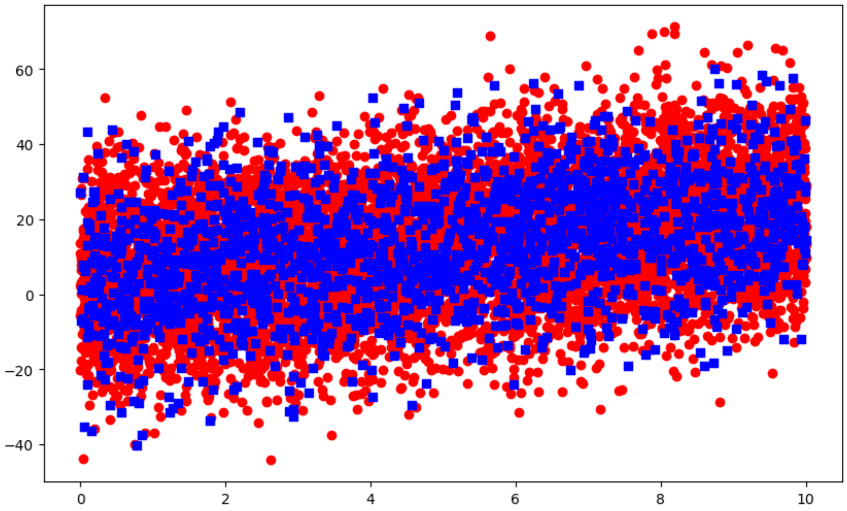


Figure 2: Random data generation and visualization for 10,000 data points

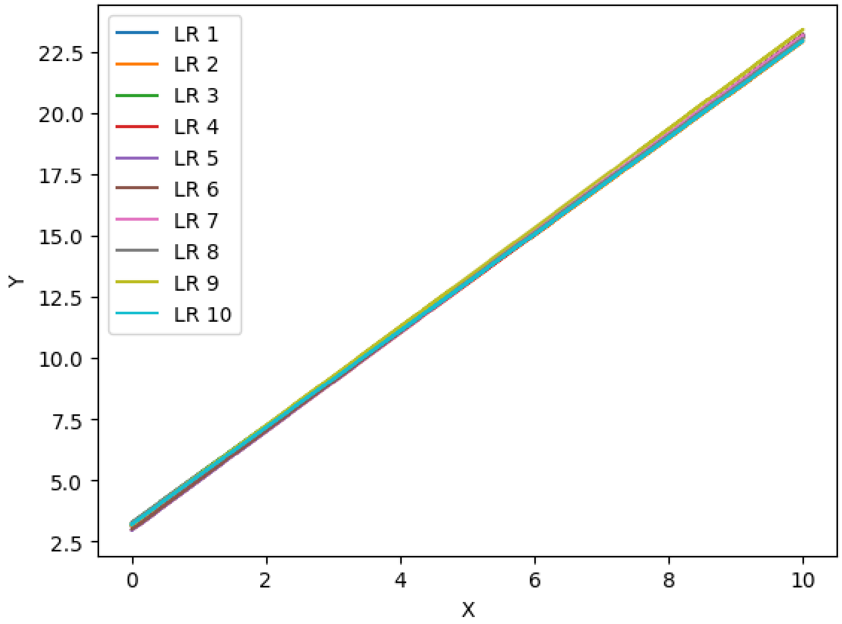


Figure 3: Linear models for 10,000 data point scenario

3 Linear regression on real world data

2. Number of Independent Variables = 33, Number of Dependent Variables = 2

3. No. There are three categorical variables: Gender, Age, and Ethnicity. To convert these categorical data into numerical data, it is appropriate to use label encoding for Age because the categorical data has an inherent order. For Gender and Ethnicity, it is advisable to use one-hot encoding because these variables do not have any ordinal relationship.

4. No. The correct approach is given below.

```
    infrared_thermography_temperature.data.dropna(subset=[’Age’], inplace=True)
    infrared_thermography_temperature.data.dropna(subset=[’Gender’], inplace=True)
```

Listing 1: Python code to drop missing values in ’Age’ and ’Gender’ columns

7.

```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

import pandas as pd

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Apply label encoding to the ’Age’ column
infrared_thermography_temperature.data[’features’][’Age_Encoded’] = label_encoder.fit_transform(infrared_thermography_temperature.data[’features’][’Age’])

X = infrared_thermography_temperature.data[’features’][[’Age_Encoded’, ’T_OR_Max1’, ’T_offset1’, ’Max1R13_1’, ’Max1L13_1’]]

# Initialize the MinMaxScaler
scaler = MinMaxScaler()

# Fit and transform the data
scaled_data = scaler.fit_transform(X)

# Convert the scaled data back to a DataFrame
X_scaled = pd.DataFrame(scaled_data, columns=X.columns)

# Label (Y) - Selecting the label column
Y = infrared_thermography_temperature.data[’targets’][’aveOralM’]

X_train, X_test, Y_train, Y_test = train_test_split(X_scaled, Y, test_size=0.2, random_state=42)

# Create a Linear Regression model
model = LinearRegression()

# Train the model on the training data
model.fit(X_train, Y_train)

# Print the coefficients of the model
print(f"Model Coefficients: {model.coef_}")
print(f"Model Intercept: {model.intercept_}")
```

Listing 2: Python code for preprocessing and linear regression using scikit-learn

- **Model Coefficients**:

Feature	Coefficient
Age_Encoded	0.0608
T_OR_Max1	1.9203
T_offset1	-0.2737
Max1R13_1	1.0628
Max1L13_1	0.6893

Model Intercept: 35.6568

8. T_OR_Max1 has the highest coefficient, indicating that it contributes the most to the dependent variable.

9.

```python
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

import pandas as pd

X = infrared_thermography_temperature.data['features'][['T_OR1', 'T_OR_Max1', 'T_FHC_Max1', 'T_FH_Max1']]

# Initialize the MinMaxScaler
scaler = MinMaxScaler()

# Fit and transform the data
scaled_data = scaler.fit_transform(X)

# Convert the scaled data back to a DataFrame
X_scaled = pd.DataFrame(scaled_data, columns=X.columns)

# Label (Y) - Selecting the label column
Y = infrared_thermography_temperature.data['targets']['aveOralM']

X_train, X_test, Y_train, Y_test = train_test_split(X_scaled, Y, test_size=0.2, random_state=42)

# Create a Linear Regression model
model = LinearRegression()

# Train the model on the training data
trained_model = model.fit(X_train, Y_train)

# Print the coefficients of the model
print(f"Model Coefficients: {model.coef_}")
print(f"Model Intercept: {model.intercept_}")
```

Listing 3: Python code for scaling features and training a linear regression model

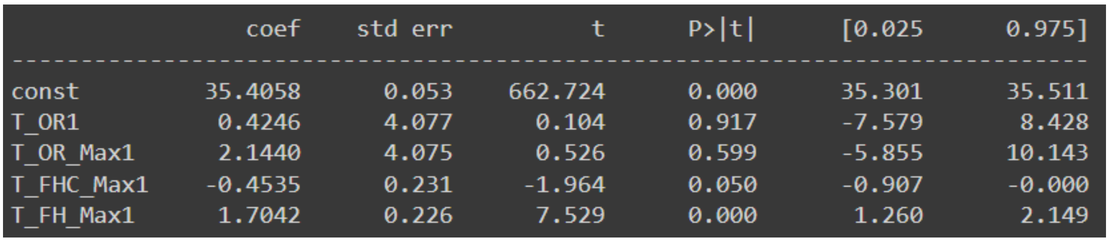
- **Model Coefficients**:

Feature	Coefficient
T_OR1	0.4246
T_OR_Max1	2.1440
T_FHC_Max1	-0.4535
T_FH_Max1	1.7042

Model Intercept: 35.4058

10.

Metric	Value
Residual Sum of Squares (RSS)	79.0280
Root Mean Squared Error (RSE)	0.3122
Mean Squared Error (MSE)	0.0968
R-squared Score	0.6433



```
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         35.4058      0.053    662.724      0.000      35.301      35.511
T_OR1          0.4246      4.077      0.104      0.917      -7.579       8.428
T_OR_Max1      2.1440      4.075      0.526      0.599      -5.855      10.143
T_FHC_Max1    -0.4535      0.231     -1.964      0.050      -0.907      -0.000
T_FH_Max1      1.7042      0.226      7.529      0.000       1.260       2.149
```

Figure 4: S.E., t-statistic, p-value

11. From the regression output, the p-values for the features `T_OR1` and `T_OR_Max1` are higher than 0.05. This indicates that these features do not have significant contributions to the model at the 5% significance level. Therefore, their impact on the dependent variable is minimal, suggesting that they may not be crucial predictors for the model.

4 Performance evaluation of Linear regression

2.

$$\text{RSE} = \sqrt{\frac{SSE}{n-D}}$$

$$\text{model A : RSE}_A = \sqrt{\frac{9}{10000-3}} \approx 0.03$$

$$\text{model B : RSE}_B = \sqrt{\frac{2}{10000-5}} \approx 0.014$$

$$\text{RSE}_A > \text{RSE}_B,\ \text{therefore model B performs better.}$$

3.

$$\begin{aligned} R^2 \text{ for A} &= 1 - \frac{9}{90} = 0.9 \\ R^2 \text{ for B} &= 1 - \frac{2}{10} = 0.8 \\ {R^2}_A > {R^2}_B&, \text{ therefore model A performs better.} \end{aligned}$$

4. R^2 is a more fair metric. Data sets in different ranges greatly affect RSE but not for R^2. Hence R^2 is much more fair.

5 Linear regression impact on outliers

2.

$$\begin{aligned} \lim_{a\to 0} \frac{1}{N} \sum_{i=1}^{N} \left(\frac{r_i^2}{a^2 + r_i^2} \right) &= \frac{1}{N} \sum_{i=1}^{N} \lim_{a\to 0} \left(\frac{r_i^2}{a^2 + r_i^2} \right) \\ &= \frac{1}{N} \sum_{i=1}^{N} 1 = \frac{1}{N} \cdot N = 1 \\ \lim_{a\to 0} \frac{1}{N} \sum_{i=1}^{N} \left(1 - \exp\left(-\frac{2|r_i|}{a} \right) \right) &= \frac{1}{N} \sum_{i=1}^{N} \left(1 - \lim_{a\to 0} \exp\left(-\frac{2|r_i|}{a} \right) \right) \\ &= \frac{1}{N} \sum_{i=1}^{N} (1 - 0) = \frac{1}{N} \cdot N = 1 \end{aligned}$$

3.

$$L_2(w), \quad a = 2.5$$

This has a value of 1 for $|r_i| > 40$ and it reaches 1 as it reaches 40.

$$L_1(w), \quad a = 25$$

This has a value very close to 1 for $|r_i| > 40$ and it behaves in a decent way for values of $|r_i| < 40$.