24MDT0184_regression_DA_2

February 14, 2025

1 Name: Tufan Kundu

2 Reg no: 24MDT0184

2.1 Digital Assessment 2: Multiple Linear Regression Analysis

2.2 Slot: L29+L30

2.3 Course Name & code : Regression Analysis and Predictive Models Lab & PMDS504P

2.4 Faculty Name : Dr. Jisha Francis

3 Questions:

• Perform regression analysis and compare the models: 1. Fit two simple linear regression models: – One model using car Weight as the predictor for CO2 emissions. – Another model using car Volume as the predictor for CO2 emissions. 2. Fit a multiple linear regression model using both Weight and Volume as predictors for CO2 emissions. 3. Compare the three models based on their R-squared and adjusted R-squared values, discussing the strengths and limitations of each model. 4. Interpret the regression coefficients for each model and discuss which model provides the best fit for predicting CO2 emissions, providing reasoning based on statistical significance and model performance.

3.1 Importing the necessary libraries

[2]:
```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
```

[3]:
```
## loading the dataset
```

[8]:
```
df = pd.read_excel(r"C:\Users\Batch1\Downloads\tk\cardata.xlsx")
df
```

[8]:

	Car	Model	Volume	Weight	CO2
0	Toyota	Aygo	1000	790	99
1	Mitsubishi	Space Star	1200	1160	95
2	Skoda	Citigo	1000	929	95

3	Fiat	500	900	865	90
4	Mini	Cooper	1500	1140	105
5	VW	Up!	1000	929	105
6	Skoda	Fabia	1400	1109	90
7	Mercedes	A-Class	1500	1365	92
8	Ford	Fiesta	1500	1112	98
9	Audi	A1	1600	1150	99
10	Hyundai	I20	1100	980	99
11	Suzuki	Swift	1300	990	101
12	Ford	Fiesta	1000	1112	99
13	Honda	Civic	1600	1252	94
14	Hundai	I30	1600	1326	97
15	Opel	Astra	1600	1330	97
16	BMW	1	1600	1365	99
17	Mazda	3	2200	1280	104
18	Skoda	Rapid	1600	1119	104
19	Ford	Focus	2000	1328	105
20	Ford	Mondeo	1600	1584	94
21	Opel	Insignia	2000	1428	99
22	Mercedes	C-Class	2100	1365	99
23	Skoda	Octavia	1600	1415	99
24	Volvo	S60	2000	1415	99
25	Mercedes	CLA	1500	1465	102
26	Audi	A4	2000	1490	104
27	Audi	A6	2000	1725	114
28	Volvo	V70	1600	1523	109
29	BMW	5	2000	1705	114
30	Mercedes	E-Class	2100	1605	115
31	Volvo	XC70	2000	1746	117
32	Ford	B-Max	1600	1235	104
33	BMW	2	1600	1390	108
34	Opel	Zafira	1600	1405	109
35	Mercedes	SLK	2500	1395	120

[6]:
```
## checking the shape of the dataset
df.shape
```

[6]: (36, 5)

[9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Car     36 non-null     object
 1   Model   36 non-null     object
```

```
 2   Volume  36 non-null     int64
 3   Weight  36 non-null     int64
 4   CO2     36 non-null     int64
dtypes: int64(3), object(2)
memory usage: 1.5+ KB
```

[10]:
```
## checking for null values
df.isnull().sum()
```

[10]:
```
Car       0
Model     0
Volume    0
Weight    0
CO2       0
dtype: int64
```

[11]:
```
## Checking for duplicate values
df.duplicated().sum()
```

[11]:
```
0
```

3.2 Making Simple Linear Regression model using Car Weight as predictor for C02 emissions

[41]:
```
x_weight = df[['Weight']]
y = df[['CO2']]

model_weight = LinearRegression()
model_weight.fit(x_weight,y)
y_pred_weight = model_car.predict(x_weight)

## Intercept for the model
print("Intercept for the model:",model_weight.intercept_)
print("Coefficient for the model:",model_weight.coef_[0])

## R2 score for the model
from sklearn.metrics import r2_score,mean_squared_error

r2_weight = r2_score(y,y_pred_weight)
print("R2 score of the model is:", r2_weight)
print("MSE of the model is:", mean_squared_error(y_pred_weight,y))
```

```
Intercept for the model: [80.05939852]
Coefficient for the model: [0.01699973]
R2 score of the model is: 0.30486966019513084
MSE of the model is: 37.55581115956569
```

[36]:
```
import statsmodels.api as sm
x = df['Weight']
y = df['CO2']

# adds a constant to the independent variable
x_with_const = sm.add_constant(x)
# Fit the OLS regression model using stats model
model_sm_weight = sm.OLS(y,x_with_const).fit()
print(model_sm_weight.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    CO2   R-squared:                       0.305
Model:                            OLS   Adj. R-squared:                  0.284
Method:                 Least Squares   F-statistic:                     14.91
Date:                Fri, 14 Feb 2025   Prob (F-statistic):           0.000481
Time:                        12:19:35   Log-Likelihood:                -116.35
No. Observations:                  36   AIC:                             236.7
Df Residuals:                      34   BIC:                             239.9
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         80.0594      5.785     13.839      0.000      68.302      91.816
Weight         0.0170      0.004      3.862      0.000       0.008       0.026
==============================================================================
Omnibus:                        0.226   Durbin-Watson:                   0.988
Prob(Omnibus):                  0.893   Jarque-Bera (JB):                0.104
Skew:                           0.122   Prob(JB):                        0.949
Kurtosis:                       2.901   Cond. No.                     7.23e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 7.23e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

3.3 Making Simple Linear Regression model using Car Volume as predictor for C02 emissions

[40]:
```
x_vol = df[['Volume']]
y = df[['CO2']]

model_vol = LinearRegression()
model_vol.fit(x_vol,y)
y_pred_vol = model_vol.predict(x_vol)
```

```
## Intercept for the model
print("Intercept for the model:",model_vol.intercept_)
print("Coefficient for the model:",model_vol.coef_[0])

## R2 score for the model
from sklearn.metrics import r2_score,mean_squared_error

r2_vol = r2_score(y,y_pred_vol)
print("R2 score of the model is:", r2_vol)
print("MSE of the model is:", mean_squared_error(y_pred_vol,y))
```

```
Intercept for the model: [83.74643307]
Coefficient for the model: [0.01134704]
R2 score of the model is: 0.3505608516055503
MSE of the model is: 35.08725287919057
```

[44]:
```
import statsmodels.api as sm
x = df['Volume']
y = df['CO2']

# adds a constant to the independent variable
x_with_const = sm.add_constant(x)
# Fit the OLS regression model using stats model
model_sm_volume = sm.OLS(y,x_with_const).fit()
print(model_sm_volume.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    CO2   R-squared:                       0.351
Model:                            OLS   Adj. R-squared:                  0.331
Method:                 Least Squares   F-statistic:                     18.35
Date:                Fri, 14 Feb 2025   Prob (F-statistic):           0.000142
Time:                        12:26:54   Log-Likelihood:                -115.12
No. Observations:                  36   AIC:                             234.2
Df Residuals:                      34   BIC:                             237.4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         83.7464      4.387     19.092      0.000      74.832      92.661
Volume         0.0113      0.003      4.284      0.000       0.006       0.017
==============================================================================
Omnibus:                        7.658   Durbin-Watson:                   0.958
Prob(Omnibus):                  0.022   Jarque-Bera (JB):                2.261
Skew:                           0.076   Prob(JB):                        0.323
Kurtosis:                       1.782   Cond. No.                     7.15e+03
```

```
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 7.15e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

3.4 Fit a multiple linear regression model using both Weight and Volume as predictors for CO2 emissions

[42]:
```
x = df[['Weight','Volume']]
y =  df[['CO2']]

model_combined = LinearRegression()
model_combined.fit(x,y)
y_pred = model_combined.predict(x)

## Intercept for the model
print("Intercept for the model:",model_combined.intercept_)
print("Coefficients for the model:",model_combined.coef_)

## R2 score for the model
from sklearn.metrics import r2_score,mean_squared_error

r2 = r2_score(y,y_pred)
print("R2 score of the combined model is:", r2)
print("MSE of the combined model is:", mean_squared_error(y_pred,y))
```

```
Intercept for the model: [79.69471929]
Coefficients for the model: [[0.00755095 0.00780526]]
R2 score of the combined model is: 0.3765564043619988
MSE of the combined model is: 33.68279098995155
```

[38]:
```
import statsmodels.api as sm
x = df[['Weight','Volume']]
y = df['CO2']

# adds a constant to the independent variable
x_with_const = sm.add_constant(x)
# Fit the OLS regression model using stats model
model_sm_full = sm.OLS(y,x_with_const).fit()
print(model_sm_full.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    CO2   R-squared:                       0.377
Model:                            OLS   Adj. R-squared:                  0.339
```

```
Method:                 Least Squares   F-statistic:                     9.966
Date:                Fri, 14 Feb 2025   Prob (F-statistic):           0.000411
Time:                        12:22:11   Log-Likelihood:                -114.39
No. Observations:                  36   AIC:                             234.8
Df Residuals:                      33   BIC:                             239.5
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         79.6947      5.564     14.322      0.000      68.374      91.016
Weight         0.0076      0.006      1.173      0.249      -0.006       0.021
Volume         0.0078      0.004      1.948      0.060      -0.000       0.016
==============================================================================
Omnibus:                        4.957   Durbin-Watson:                   0.944
Prob(Omnibus):                  0.084   Jarque-Bera (JB):                1.836
Skew:                          -0.025   Prob(JB):                        0.399
Kurtosis:                       1.895   Cond. No.                     1.16e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.16e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

3.5 Conclusion

[45]:
```
## metric for Simple linear reg with weight
print("Intercept for the model with weight:",model_weight.intercept_)
print("Coefficient for the model with weight:",model_weight.coef_[0])
print("R2 score of the model with weight is:", r2_weight)
print("Adjusted R2 score of the model with weight:0.284")  # value we got from
 ↪stats model
print("MSE of the model with weight is:", mean_squared_error(y_pred_weight,y))
print()
print()

## Metric for simple linear reg with vol
print("Intercept for the model with volume:",model_vol.intercept_)
print("Coefficient for the model with volume:",model_vol.coef_[0])
print("R2 score of the model with volume is:", r2_vol)
print("Adjusted R2 score of the model with weight:0.331")  # value we got from
 ↪stats model
print("MSE of the model with volume is:", mean_squared_error(y_pred_vol,y))
print()
print()
```

```
## Metric for Multiple linear model with both weight and volume
print("Intercept for the combined model:",model_combined.intercept_)
print("Coefficients for the combined model:",model_combined.coef_)
print("R2 score of the combined model is:", r2)
print("Adjusted R2 score of the combined model:0.339")  # value we got from␣
 ↪stats model
print("MSE of the combined model is:", mean_squared_error(y_pred,y))
```

```
Intercept for the model with weight: [80.05939852]
Coefficient for the model with weight: [0.01699973]
R2 score of the model with weight is: 0.30486966019513084
Adjusted R2 score of the model with weight:0.284
MSE of the model with weight is: 37.55581115956569
```

```
Intercept for the model with volume: [83.74643307]
Coefficient for the model with volume: [0.01134704]
R2 score of the model with volume is: 0.3505608516055503
Adjusted R2 score of the model with weight:0.331
MSE of the model with volume is: 35.08725287919057
```

```
Intercept for the combined model: [79.69471929]
Coefficients for the combined model: [[0.00755095 0.00780526]]
R2 score of the combined model is: 0.3765564043619988
Adjusted R2 score of the combined model:0.339
MSE of the combined model is: 33.68279098995155
```

Since the R2 score of the multiple linear regression model is the highest (0.37) indicating the model can explain 37% of the variance in CO2 emission. Adjusted R2 also maintains high explainability. Also the mean squared error for the multiple linear regression model is the least. So the multiple linear model performs best in predicting the emission of CO2. Followed by the simple linear regression model using Volume