exp3

October 4, 2023

[1]:
```
#dataset : https://www.kaggle.com/datasets/mdrazakhan/linear-regression-dataset
```

[16]:
```
import pandas as pd
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
```

[3]:
```
df=pd.read_csv('dataset\cars.csv')
```

[4]:
```
df.head()
```

[4]:

	Car	Model	Volume	Weight	CO2
0	Toyoty	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

[7]:
```
x=df[['Weight', 'Volume']]
y=df['CO2']
```

[19]:
```
combined_df = pd.concat([x, y], axis=1)

# Calculate the correlation matrix
correlation_matrix = combined_df.corr()
print(correlation_matrix)
```

```
          Weight    Volume       CO2
Weight  1.000000  0.753537  0.552150
Volume  0.753537  1.000000  0.592082
CO2     0.552150  0.592082  1.000000
```

[ ]:
```
'''
Weight and CO2:
    The correlation coefficient between 'Weight' and 'CO2' is approximately 0.
↪5522.
```

```
    This suggests a moderate positive correlation between 'Weight' and 'CO2'.
    As the weight of an object increases, the CO2 emissions tend to increase to␣
 ↪some extent.

Volume and CO2:
    The correlation coefficient between 'Volume' and 'CO2' is approximately 0.
 ↪5921.
    This also indicates a moderate positive correlation between 'Volume' and␣
 ↪'CO2'.
    As the volume of an object increases, the CO2 emissions tend to increase to␣
 ↪some extent.
'''
```

[11]:
```
#train test split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=42)
```

[12]:
```
regrLine=linear_model.LinearRegression()
regrLine.fit(X_train, y_train)
```

[12]:
```
LinearRegression()
```

[13]:
```
# Coefficient
print("Coefficient : ",end=' ')
print(regrLine.coef_)
```

```
Coefficient :  [0.00804928 0.00428741]
```

[17]:
```
# Make predictions on the test data
y_pred = regrLine.predict(X_test)

# Calculate mse to evaluate the model's performance
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 58.07928583657771
```