

```
In [28]: import pandas

df = pandas.read_csv('car.csv')

print(df)
```

	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
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	216	1600	1390	108
34	Opel	Zafira	1600	1405	109
35	Mercedes	SLK	2500	1395	120

```
In [29]: import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [30]: X = df[['Weight', 'Volume']]
y = df['CO2']
```

```
In [31]: # Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [32]: # Step 2: Fit the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
Out[32]: ▼ LinearRegression
LinearRegression()
```

```
In [33]: # Step 3: Make predictions on the testing set
y_pred = model.predict(X_test)
```

```
In [34]: # Step 4: Calculate SSE
SSE = np.sum((y_test - y_pred) ** 2)
```

```
In [35]: # Step 5: Calculate SST
SST = np.sum((y_test - np.mean(y_test)) ** 2)
```

```
In [36]: # Step 6: Calculate R-squared
R_squared = 1 - (SSE / SST)
```

```
In [37]: # Step 7: Calculate Adjusted R-squared
n = len(X_test)
p = X_test.shape[1] # Number of predictors
adjusted_R_squared = 1 - (1 - R_squared) * (n - 1) / (n - p - 1)
```

```
In [38]: print("Sum of Squared Errors (SSE):", SSE)
print("Total Sum of Squares (SST):", SST)
print("R-squared:", R_squared)
print("Adjusted R-squared:", adjusted_R_squared)
```

```
Sum of Squared Errors (SSE): 464.6342866926212
Total Sum of Squares (SST): 692.875
R-squared: 0.3294110962401282
Adjusted R-squared: 0.06117553473617954
```

```
In [ ]:
```