Linear Regression Model

Overview

Linear regression is a statistical modeling technique used to model the relationship between a dependent variable and one or more independent variables. The goal is to find the best-fitting linear equation that describes the relationship between the variables. The equation takes the form:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \varepsilon$

Where:

y is the dependent variable

$x_1, x_2, ..., x_p$ are the independent variables

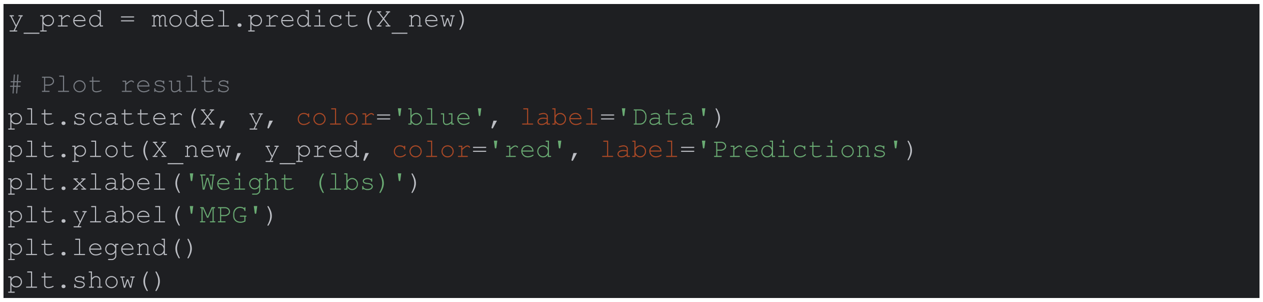
$\beta_0, \beta_1, ..., \beta_p$ are the coefficients that represent the change in y for a one unit change in the corresponding x

ε is the error term representing the variation in y not explained by the model

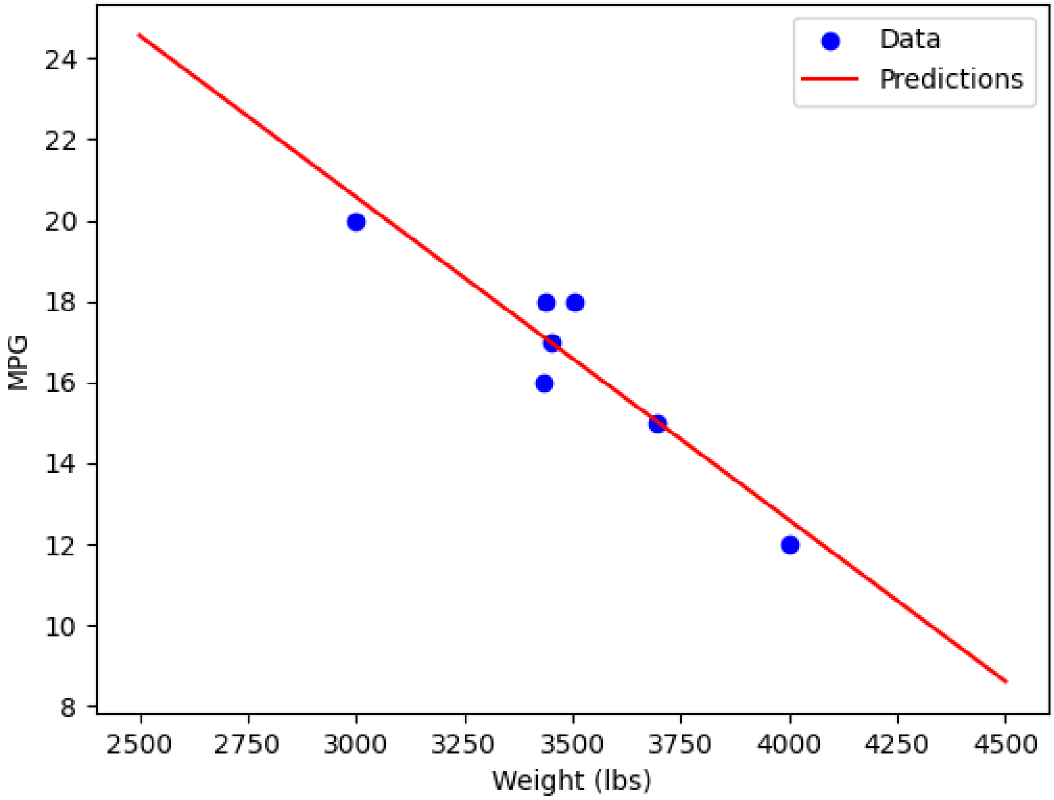
The coefficients are estimated using the ordinary least squares (OLS) method, which minimizes the sum of the squared residuals between the observed values and the values predicted by the model.

Code:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Create sample data
data = pd.DataFrame({
    'weight': [3504, 3693, 3436, 3433, 3449, 3000, 4000],
    'mpg': [18.0, 15.0, 18.0, 16.0, 17.0, 20.0, 12.0]
})

# Extract variables
X = data['weight'].values.reshape(-1, 1)
y = data['mpg'].values

# Fit linear regression model
model = LinearRegression()
model.fit(X, y)

# Print model summary
print(f"Intercept: {model.intercept_:.3f}")
print(f"Coefficient: {model.coef_[0]:.3f}")

# Make predictions
X_new = np.array([[2500], [3000], [3500], [4000], [4500]])
```



```
y_pred = model.predict(X_new)

# Plot results
plt.scatter(X, y, color='blue', label='Data')
plt.plot(X_new, y_pred, color='red', label='Predictions')
plt.xlabel('Weight (lbs)')
plt.ylabel('MPG')
plt.legend()
plt.show()
```


Data
Predictions
MPG
Weight (lbs)