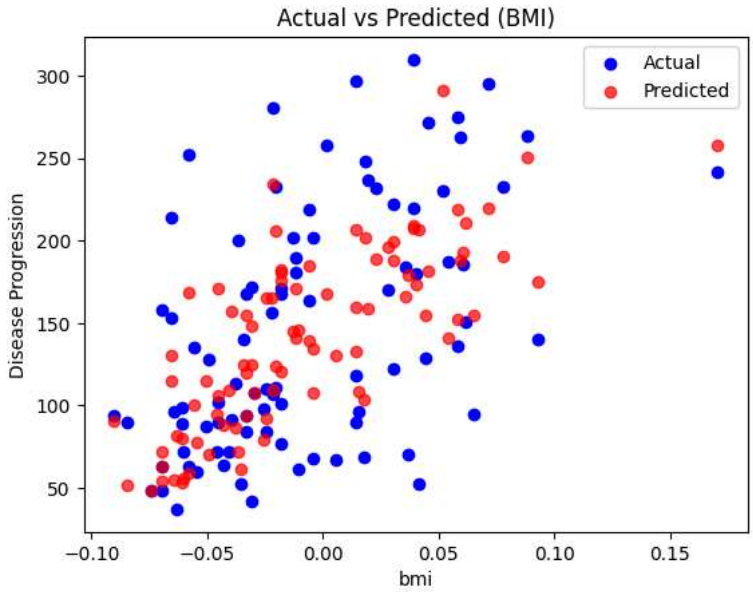
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Example dataset: load from sklearn
from sklearn.datasets import load_diabetes

data = load_diabetes()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target)

# (Optional preprocessing if needed, e.g., scaling, encoding — skipped for simplicity)

# 2. Split data into train-test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 3. Fit a Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# 4. Evaluate model
y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"MAE: {mae:.2f}")
print(f"MSE: {mse:.2f}")
print(f"R²: {r2:.2f}")

# 5. Plot regression line (for one feature for simplicity, e.g., 'bmi')
feature = 'bmi'
plt.scatter(X_test[feature], y_test, color='blue', label='Actual')
plt.scatter(X_test[feature], y_pred, color='red', label='Predicted', alpha=0.7)
plt.xlabel(feature)
plt.ylabel('Disease Progression')
plt.title('Actual vs Predicted (BMI)')
plt.legend()
plt.show()

# Interpret coefficients
coef_df = pd.DataFrame({
    'Feature': X.columns,
    'Coefficient': model.coef_
})
print(coef_df)
print(f"Intercept: {model.intercept_:.2f}")
```

```
MAE: 42.79
MSE: 2900.19
R²: 0.45
```



	Feature	Coefficient
0	age	37.904021
1	sex	-241.964362
2	bmi	542.428759
3	bp	347.703844
4	s1	-931.488846
5	s2	518.062277
6	s3	163.419983
7	s4	275.317902
8	s5	736.198859