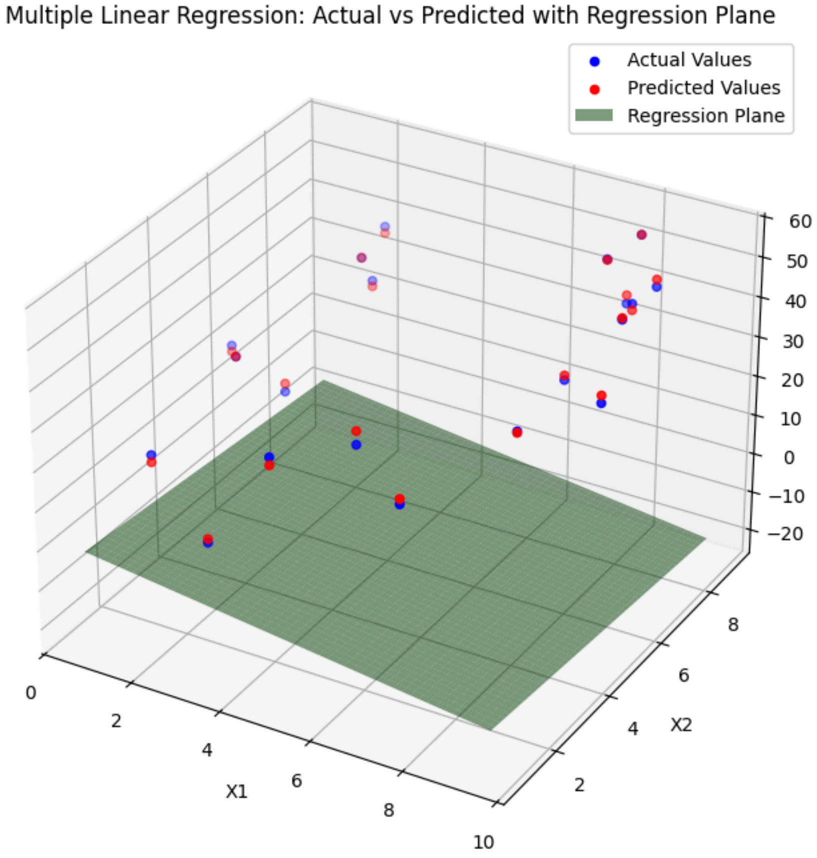
Multiple Linear Regression

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Generate synthetic data
X = np.random.rand(100, 3) * 10
y_mlr = 3.0 * X[:, 0] + 1.5 * X[:, 1] + 2.0 * X[:, 2] + np.random.randn(100) * 2

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y_mlr, test_size=0.2, random_state=42)

# Fit the multiple linear regression model
mlr_model = LinearRegression()
mlr_model.fit(X_train, y_train)

# Predict on the test set
y_pred_mlr = mlr_model.predict(X_test)

# Print the results
print("\nMultiple Linear Regression Results Before Overfitting Check:")
print(f"Coefficients: {mlr_model.coef_}")
print(f"Intercept: {mlr_model.intercept_}")
print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred_mlr)}")
print(f"R-squared: {r2_score(y_test, y_pred_mlr)} \n")

# Create a 3D plot
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(X_test[:, 0], X_test[:, 1], y_test, color='blue', label='Actual Values')

ax.scatter(X_test[:, 0], X_test[:, 1], y_pred_mlr, color='red', label='Predicted Values')

x0, x1 = np.meshgrid(np.linspace(X_test[:, 0].min(), X_test[:, 0].max(), 100),
                     np.linspace(X_test[:, 1].min(), X_test[:, 1].max(), 100))
x2 = (mlr_model.intercept_ + mlr_model.coef_[0] * x0 + mlr_model.coef_[1] * x1) / (-mlr_model.coef_[2])

# Plot the regression plane
ax.plot_surface(x0, x1, x2, color='green', alpha=0.5, label='Regression Plane')

# Set labels and title
ax.set_xlabel('X1')
ax.set_ylabel('X2')
ax.set_zlabel('y')
ax.set_title('Multiple Linear Regression: Actual vs Predicted with Regression Plane')

ax.legend()
plt.show()


#Overfitting Check
r2_train_mlr = r2_score(y_train, mlr_model.predict(X_train))
r2_test_mlr = r2_score(y_test, y_pred_mlr)
print(f"\nR-squared Train (MLR): {r2_train_mlr}")
print(f"R-squared Test (MLR): {r2_test_mlr}")
overfitting_mlr = r2_train_mlr > r2_test_mlr
print(f"Overfitting in Multiple Linear Regression: {'Yes' if overfitting_mlr else 'No'}")

if overfitting_mlr:
    ridge_model_mlr = Ridge(alpha=1.0)
    ridge_model_mlr.fit(X_train, y_train)
    y_pred_ridge_mlr = ridge_model_mlr.predict(X_test)
    print("\nRidge Regression Results for MLR:")
    print(f"Coefficients: {ridge_model_mlr.coef_}")
    print(f"Intercept: {ridge_model_mlr.intercept_}")
    print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred_ridge_mlr)}")
    print(f"R-squared: {r2_score(y_test, y_pred_ridge_mlr)}")
```

```
Multiple Linear Regression Results Before Overfitting Check:
Coefficients: [2.9516421  1.53960422 2.04096319]
Intercept: 0.19127280879830266
Mean Squared Error: 2.3907832544564442
R-squared: 0.984367126997609
```



```
R-squared Train (MLR): 0.9739964735623241
R-squared Test (MLR): 0.984367126997609
Overfitting in Multiple Linear Regression: No
```