Linear Regression from Scratch: California Housing

Roll Number: g25ait1010

1. Assumptions

- Features are standardized to mean 0 and variance 1; target is unscaled.
- Dataset has no missing values and features follow a linear relationship with target.

2. Resources Used

- Scikit-learn for data loading and train/test split.
- NumPy for numerical operations.
- Matplotlib for plotting.
- GeeksforGeeks Gradient Descent formulas[1].
- FutureAI blog on MSE and R^2 definitions[2].

3. Implementation

```
class LinearRegression:
    def __init__(self, learning_rate=0.01, n_iterations=1000, lambda_param=0):
        self.learning_rate = learning_rate
        self.n_iterations = n_iterations
        self.lambda_param = lambda_param
        self.weights = None
        self.bias = None
        self.cost_history = []

    def fit(self, X, y):
        n_samples, n_features = X.shape
        self.weights = np.zeros(n_features)
        self.bias = 0
        for i in range(self.n_iterations):
            y_pred = X.dot(self.weights) + self.bias
            cost = np.mean((y_pred - y)**2)
            dw = (1/n_samples)*(X.T.dot(y_pred-y) + self.lambda_param*self.weights)
            db = np.mean(y_pred-y)
            self.weights -= self.learning_rate*dw
            self.bias -= self.learning_rate*db
            self.cost_history.append(cost)

    def predict(self, X):
        return X.dot(self.weights) + self.bias
```

4. Evaluation Metrics

Metric	Definition
MSE	Mean Squared Error: average squared error[3]
R^2	Variance explained: squared correlation[4]

5. Experimental Results

5.1 Base Model ($\lambda=0$)

Dataset	MSE	R^2
Train	0.5457	0.5765
Test	0.5238	0.6405

5.2 Ridge Model ($\lambda=10$)

Dataset	MSE	R^2
Test	0.5179	0.6471

6. Output Plots

6.1 Learning Curve (MSE vs. Iterations)

Plots cost reduction during training iterations, showing steep drop and plateau.

6.2 Actual vs. Predicted

Scatter of predictions vs. actual values, points near 45° indicate good fit.

6.3 Learning Rate Comparison

Comparison of cost history for LR=1.0, 0.01, 0.0001.

7. Observations

- **Learning Curve:** Converges by ~1000 iterations.
- **Actual vs Predicted:** $R^2 \approx 0.64$ means 64% variance explained.
- **Regularization:** Ridge ($\lambda=10$) slightly improved test MSE and R^2.
- **Learning Rates:** 0.01 was optimal; 1.0 diverged, 0.0001 slow.

8. Conclusions

Implementation meets assignment requirements. Regularization and hyperparameter experiments documented.

References:
[1] GeeksforGeeks on Gradient Descent
[2] FutureAI blog on MSE and R^2
[3] GeeksforGeeks on MSE[3]
[4] GfG on R^2[4]