

Question 1: Linear Regression — Predicting House Prices

Dataset: [House Prices - Advanced Regression Techniques | Kaggle](#)

Objective: Build a Linear Regression model from scratch and compare its performance with a prebuilt model from sklearn.

1. Preprocess the Data

- Download and load train.csv.
- Use the feature OverallQual to predict the target SalePrice.
- Handle missing values if any and prepare the dataset.

2. Train Linear Regression from Scratch

- Implement a function that calculates the slope and intercept using the least squares formula.
- Train this model using the train.csv data.

3. Test Your Custom Model

- Download and load test.csv.
- Use the same feature (OverallQual) from the test set to make predictions.
- Calculate the Mean Squared Error (MSE) on the test predictions.

4. Train and Test Sklearn's Linear Regression Model

- Use `sklearn.linear_model.LinearRegression` to train a model on train.csv using the same feature.
- Predict on test.csv and compute the MSE for this model as well.

5. Create the Following Plots

- Plot 1 (Train Comparison): Plot the custom model vs sklearn model predictions on training data.
- Plot 2 (Test Comparison): Plot the custom model vs sklearn model predictions on test data.
- In each plot, include both, the data points and the fitted lines. Add clear titles, axis labels and legends.

Question 2: Logistic Regression — Predicting Student Pass/Fail Outcome

Dataset: [Students Performance in Exams](#)

Objective: Build a Logistic Regression model from scratch and compare its performance with a prebuilt model from sklearn.

1. Preprocess the Data

- Download and load StudentsPerformance.csv (from the Kaggle dataset).
- Create a new column PassedMath (1 if math score ≥ 50 and 0 otherwise)
- Use "reading score" as the only feature to predict, PassedMath.
- Split the dataset into training and test sets (80% train, 20% test).

2. Train Logistic Regression from Scratch

- Implement logistic regression using the sigmoid function and gradient descent.
- Train the model on the training data.

3. Test Your Custom Model

- Use your model to predict outcomes on the test set.
- If a predicted probability is ≥ 0.5 , classify it as 1; otherwise, classify as 0.
- Calculate accuracy of your model and create a confusion matrix (you may use `sklearn.metrics.confusion_matrix` for visualization).

4. Train and Test Sklearn's LogisticRegression Model

- Use `sklearn.linear_model.LogisticRegression` to train a model using the same feature.
- Predict on the test set using the sklearn model, calculate accuracy, and plot the confusion matrix.

5. Create the Following Plots

- Plot 1 (Train Comparison): Plot the custom model vs sklearn model predictions on training data.
- Plot 2 (Test Comparison): Plot the custom model vs sklearn model predictions on test data.
- In each plot, include both, the data points and the fitted lines. Add clear titles, axis labels and legends.