In this project, we set out to predict housing prices in California using linear regression. The dataset includes various features about houses and their neighborhoods — such as the number of rooms, population, income, and how close they are to the ocean. Our goal was to use this data to train a model that could accurately estimate the median house value in any given area.

To get started, we first cleaned the data by removing rows with missing values. This step is important because missing or incomplete data can throw off the model’s learning process. One of the columns, ocean\_proximity, contains text values (like "NEAR BAY" or "INLAND"), which aren't directly usable in a numerical model. So, we used label encoding to convert these categories into numbers. Here’s how they were mapped:

<1H OCEAN → 0

INLAND → 1

ISLAND → 2

NEAR BAY → 3

NEAR OCEAN → 4

To help the model understand the data more deeply, we added some new features based on existing ones. These are called engineered features, and they often boost performance in predictive tasks. We added:

rooms\_per\_household – the average number of rooms per household

population\_per\_household – how many people live in an average household

bedrooms\_per\_room – the proportion of bedrooms to total rooms

These extra features helped provide more context about living conditions, which is useful for predicting prices.

Next, we standardized all the numeric features using Z-score normalization. This means we adjusted each feature so that it has a mean of 0 and a standard deviation of 1. Doing this ensures that no single feature dominates the others just because it has larger numbers.

Once the data was ready, we split it into a training set (80%) and a test set (20%). The training set was used to teach the model, and the test set was used to check how well it performs on unseen data.

For the model itself, we used Linear Regression from scikit-learn, a simple but effective algorithm that tries to find the best-fit line through the data. After training the model, we used it to make predictions on the test set. To evaluate the model’s performance, we calculated the Mean Squared Error (MSE) — a standard metric that shows how far off our predictions were from the actual values. Lower MSE means better performance.

Conclusion:

Used label encoding for the categorical ocean\_proximity feature

Added 3 new engineered features to improve prediction accuracy

Applied Z-score normalization for balanced learning

Used LinearRegression from sklearn for modeling

Evaluation was done using Mean Squared Error (MSE)