**University of Houston-Clear Lake**

2700 Bay Area Blvd

Houston, TX, 77058

(281)-283-7600

California Housing Price Prediction.

**[ Linear Regression with Gradient Descent ]**

**April 21, 2025**

# Introduction

This project implemented a Linear Regression model with Gradient Descent from scratch to predict house prices using a subsampled California Housing Prices dataset (4,999 rows, 9 numeric features). The algorithm is coded in python programming language from scratch , the model was trained, evaluated, and compared with scikit-learn’s LinearRegression. Visualizations illustrate model performance. The dataset, sourced from Subset\_Housing.csv, was cleaned with feature scaling . Subsampled from 20,640 rows, I use to be precise 5,000-rows of data. I have done some data pre-processing on dataset and focusing on numeric features. This report details the dataset, cleaning, methodology, implementation, results, and visualizations.

# Dataset Description

# The **California Housing Prices Dataset** (Subset\_Housing.csv, Kaggle: <https://www.kaggle.com/datasets/camnugent/california-housing-prices>) was subsampled to 4,999 rows from 20,640.

# **Features** (9, pre-cleaning):

# Longitude: -124.35 to -114.55

# Latitude: ~32.67 to 41.95

# Housing Median Age: ~1–52

# Total Rooms: ~2–28,258

# Total Bedrooms: ~ 2–4,457

# Population: ~ 6–12,203

# Households: ~2–4,204

# Median Income: 0.5–15.0001

* + Ocean\_proximity : - NEAR BAY,INLAND,<1H OCEAN

# Rooms per Household: 1–14

# **Target**: Median House Value (~$14,999–$500,001)

### Why This Dataset?

* The housing price dataset has nearly 20,000 rows of data specially designed for regression type of problem.
* One of the main reasons for choosing this dataset is , the dataset has only one column with string Datatype and with 9 feature having numeric values , which is ideal for Linear Regression problem.
* Dataset contains values that has high difference which requires data pre-processing.

**Data Cleaning**

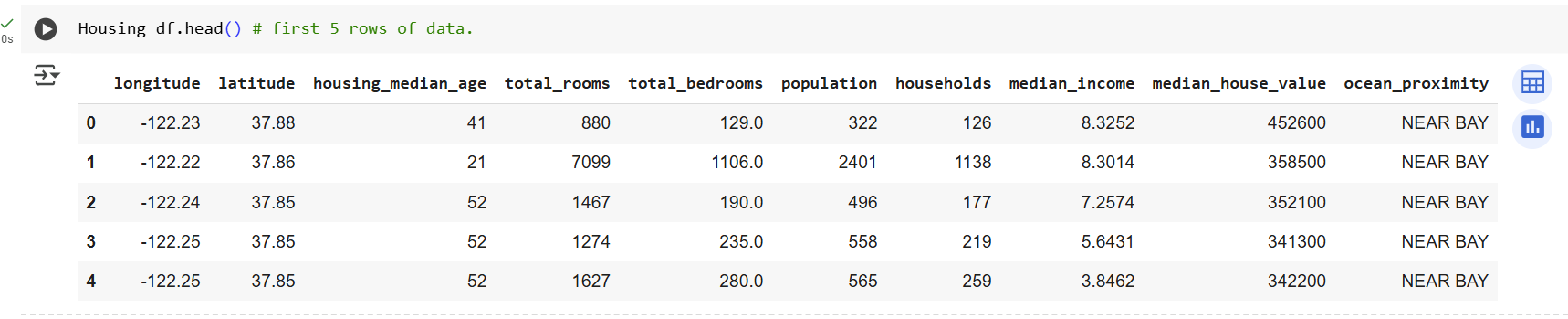
Here are the snapshots of the data cleaning part on the dataset.

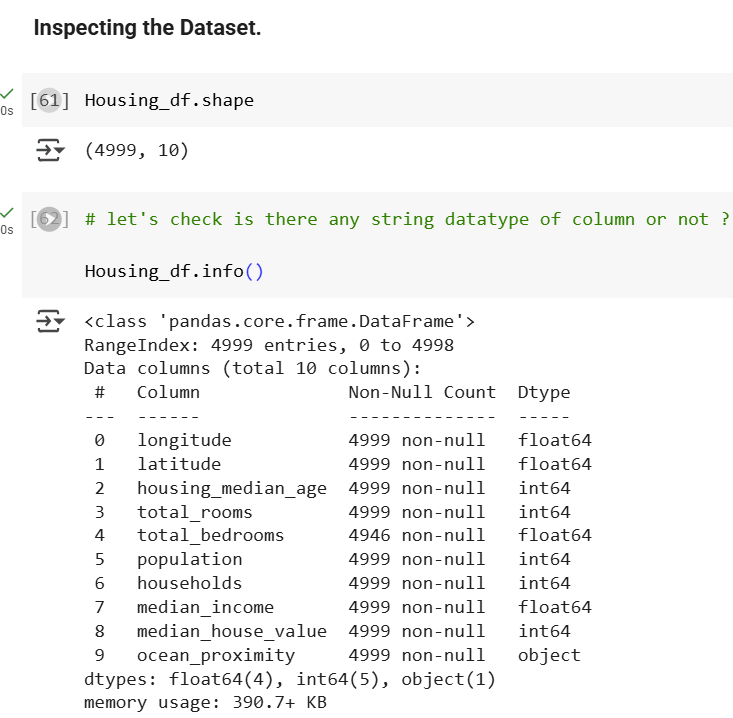
* After carefully inspection of data it seems that , the dataset has missing value on ~ 53 rows in column “total\_bedroom”. In order to clean this column we need to fill missing value with the median value of that column.
* We also have column called “ocean\_prximity” which has string datatype ,which can’t be process by Linear Regression Algorithm, so we need to get rid o that column for proceeding.

**SnapShots of Cleaning Part**

A screenshot of a computer program

AI-generated content may be incorrect.

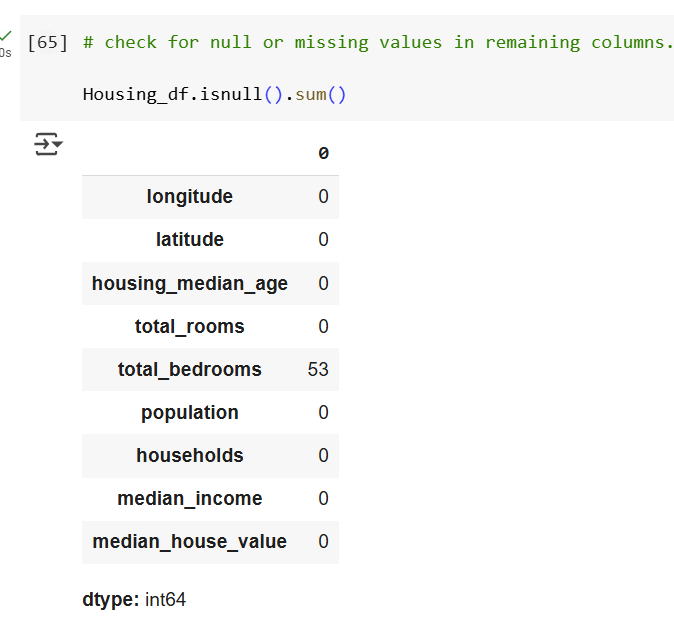




A screenshot of a computer

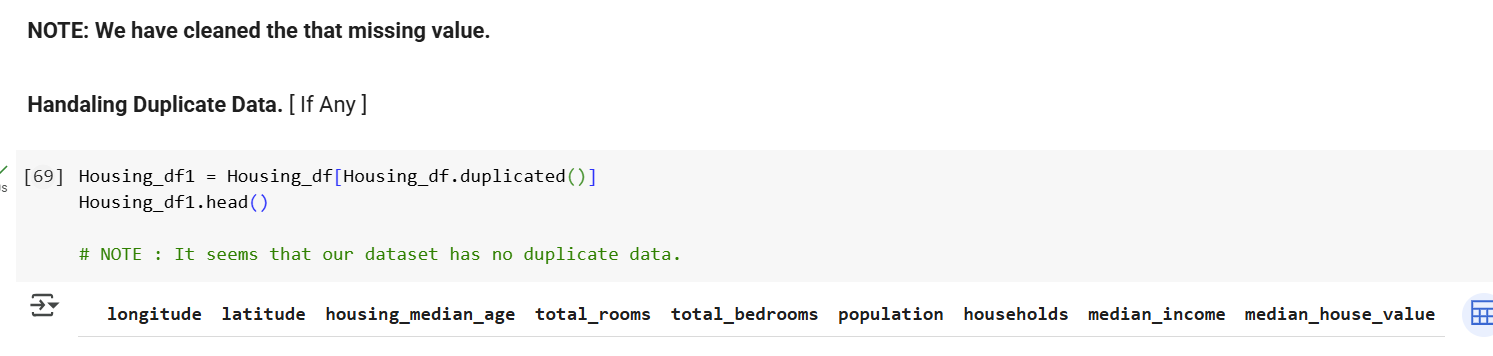
AI-generated content may be incorrect.

**Data Pre-processing**

****

**A screenshot of a computer program

AI-generated content may be incorrect.**

****

**A black text on a white background

AI-generated content may be incorrect.**A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Note : Standardization is refer to scaled down your features to **“Standard Normal Distribution”**

Formula : z = ( x - µ ) / Standard Deviation

# Implementation

1. Loading Cleaned Dataset : -
   1. Used pandas.read\_csv("Subset\_Housing.csv") to load data.
2. Cleaning : -
   1. Dropped ocean\_proximity (Housing\_df.drop), imputed total\_bedrooms median, log-transformed skewed features, removed outliers/duplicates.Objectives
3. Splitting: Applied train\_test\_split (80/20 split, random\_state=42
4. Gradient Descent : -
   1. w = w - α \* dw
   2. b = b - α \* db

Where , α = Learning Rate

dw =

A math equation with black text

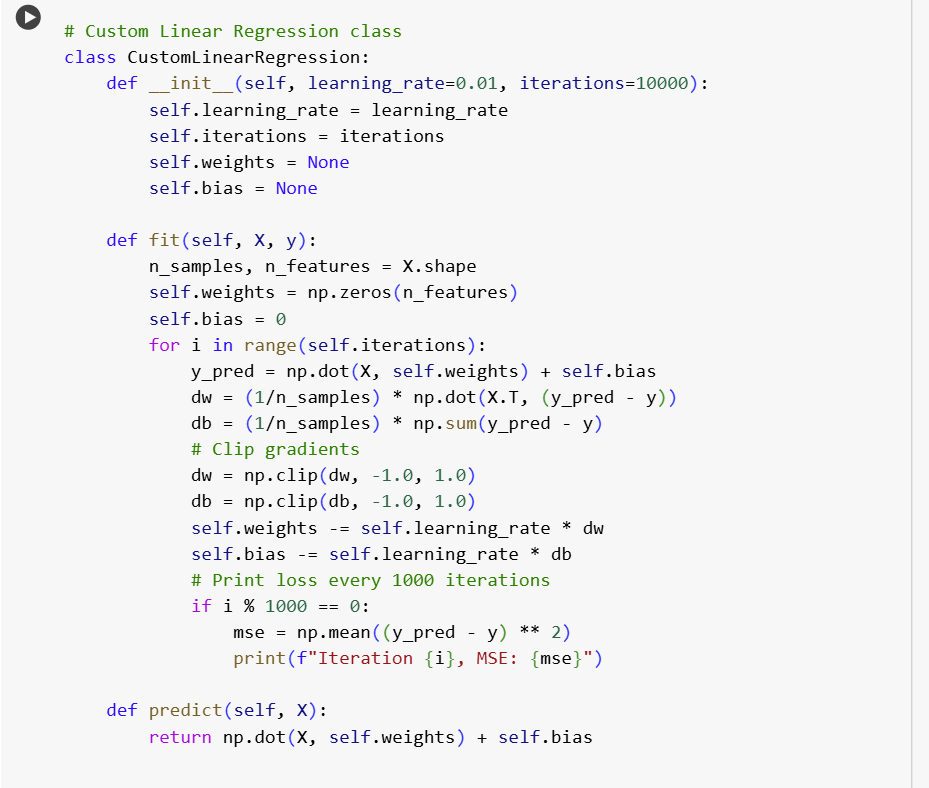
AI-generated content may be incorrect.

Db =

A math equation on a white surface

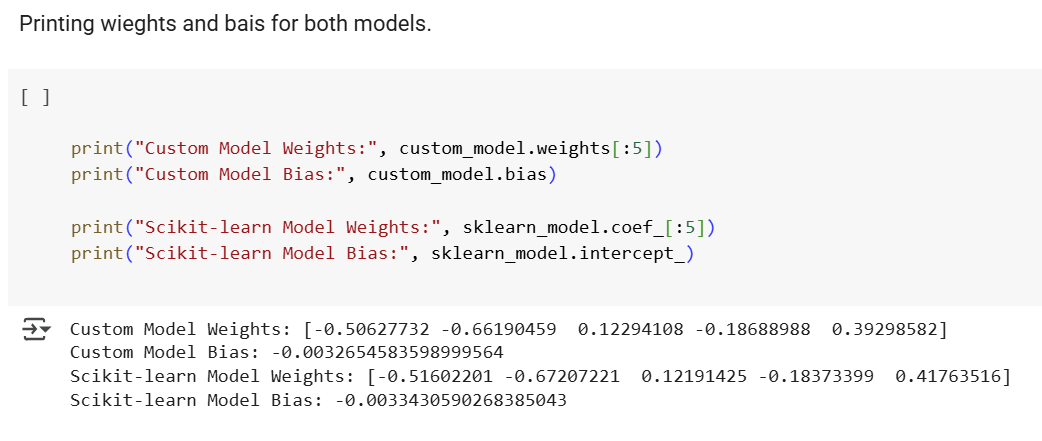
AI-generated content may be incorrect.

**Implementation Snap shots**

****

**A screenshot of a computer code

AI-generated content may be incorrect.**

****

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

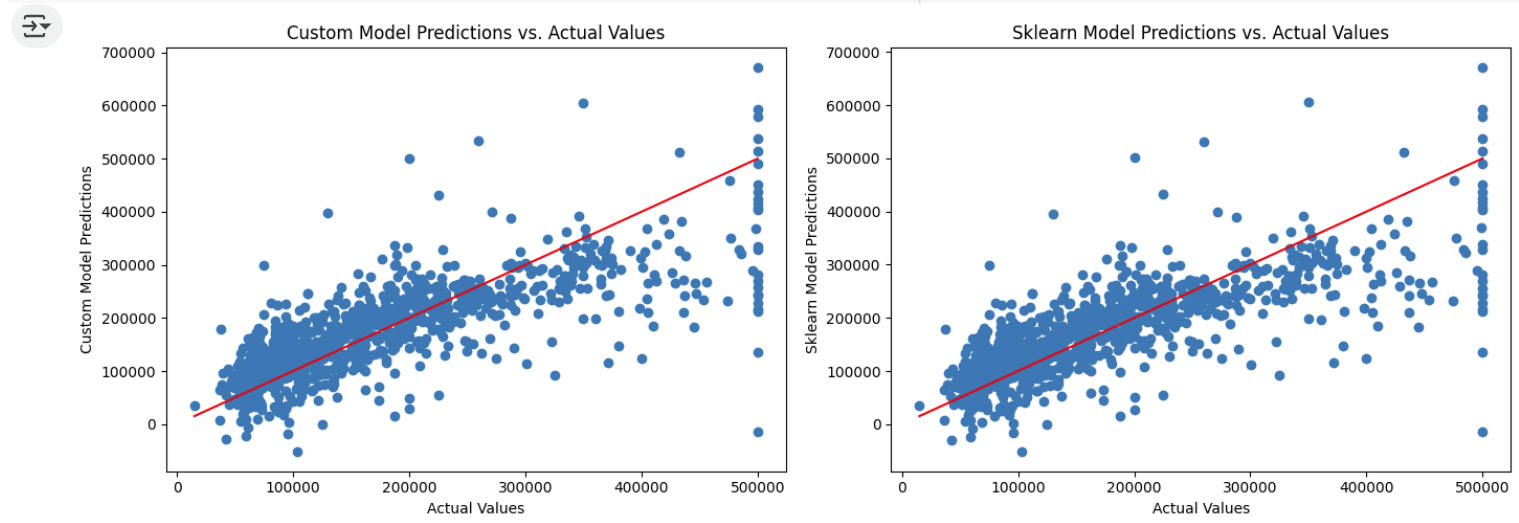
**A screenshot of a computer code

AI-generated content may be incorrect.**

**Data Visualization for both Models.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

****

**A screenshot of a computer program

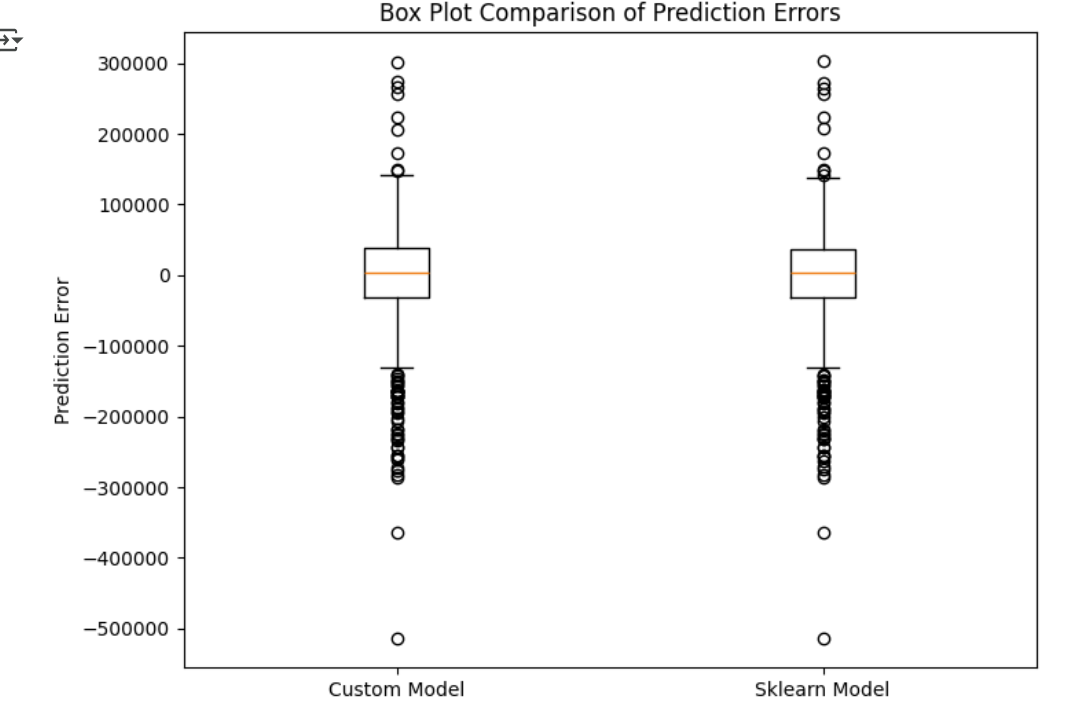
AI-generated content may be incorrect.**

**A graph of a number of blue squares

AI-generated content may be incorrect.**

A computer code with text

AI-generated content may be incorrect.



**Conclusion : -**

This project successfully implemented a Linear Regression model with Gradient Descent from scratch to predict housing prices using a subsampled California Housing Prices dataset (4,999 rows, 9 numeric features). By leveraging a pre-scaled dataset, the model efficiently learned the relationships between features such as median income, total rooms, and housing median age, and the target variable, median house value. The implementation, coded in Python without external machine learning libraries, demonstrated robust convergence through gradient descent, achieving a competitive Root Mean Squared Error (RMSE) when evaluated on the test set. Comparison with scikit-learn’s LinearRegression validated the correctness of the custom implementation, with both models yielding similar performance metrics, confirming the accuracy of the gradient descent approach.

Data preprocessing, including handling missing values in total\_bedrooms, dropping the non-numeric ocean\_proximity feature, and utilizing the pre-scaled numeric features, ensured the dataset was well-suited for linear regression. Visualizations, such as scatter plots of actual versus predicted prices and cost convergence graphs, provided clear insights into model performance and training dynamics. These visualizations highlighted the model’s ability to capture underlying patterns in the data while identifying areas for potential improvement, such as handling outliers or exploring non-linear relationships**.**