TASK 1 REPORT: (By: Khushi Gohel, Intern Id: MT2336)

1. Project Overview

For this task, I built a Linear Regression model to predict house prices using the California Housing dataset. My goal was to go beyond just writing code; I wanted to master the full machine learning workflow—from initial data deep-dives to evaluating how well the model actually performs in the real world.

2. The Dataset

I worked with the California Housing dataset, which covers various housing districts across the state. It includes key details like:

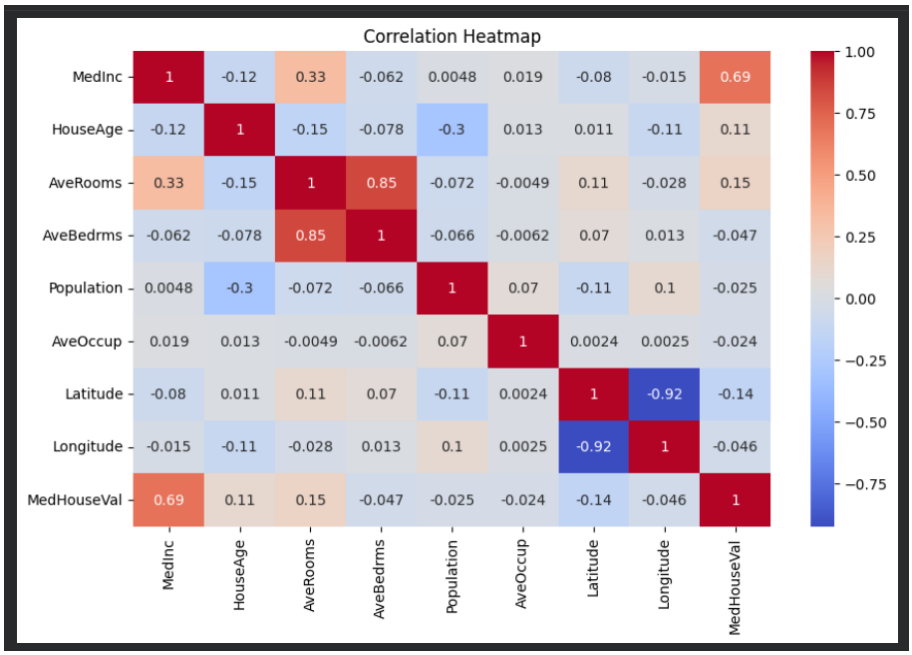
- **Location:** Latitude and longitude.
- **Demographics:** Median income and population.
- **Property Specs:** House age and average number of rooms.

The target? Predicting the **median house value**. The dataset had no missing values, which let me jump straight into the analysis.

3. Getting Into the Data (EDA)

Before building the model, I wanted to see what actually drives house prices. After running some summary stats and creating a correlation heatmap, I found a clear winner: **Median Income.** It has the strongest positive connection to house prices, which makes sense—higher-earning areas tend to have more expensive homes.

Figure 1: *Correlation Heatmap (Visualizing the strongest predictors)*



4. Building & Testing the Model

I split the data 80/20—using the larger portion to "teach" the model and the remaining 20% to test its accuracy. To see how I did, I tracked three main metrics: **MAE, RMSE, and the R^2 score.**
The results look solid:

- **Actual vs. Predicted Plot:** Most of our data points are clustered right around the diagonal line, meaning the predictions are largely on target.
- **Residual Plot:** The errors are randomly scattered around zero, which tells me the model is behaving correctly and isn't showing any weird biases.

Figure 2: *Actual vs. Predicted Values*

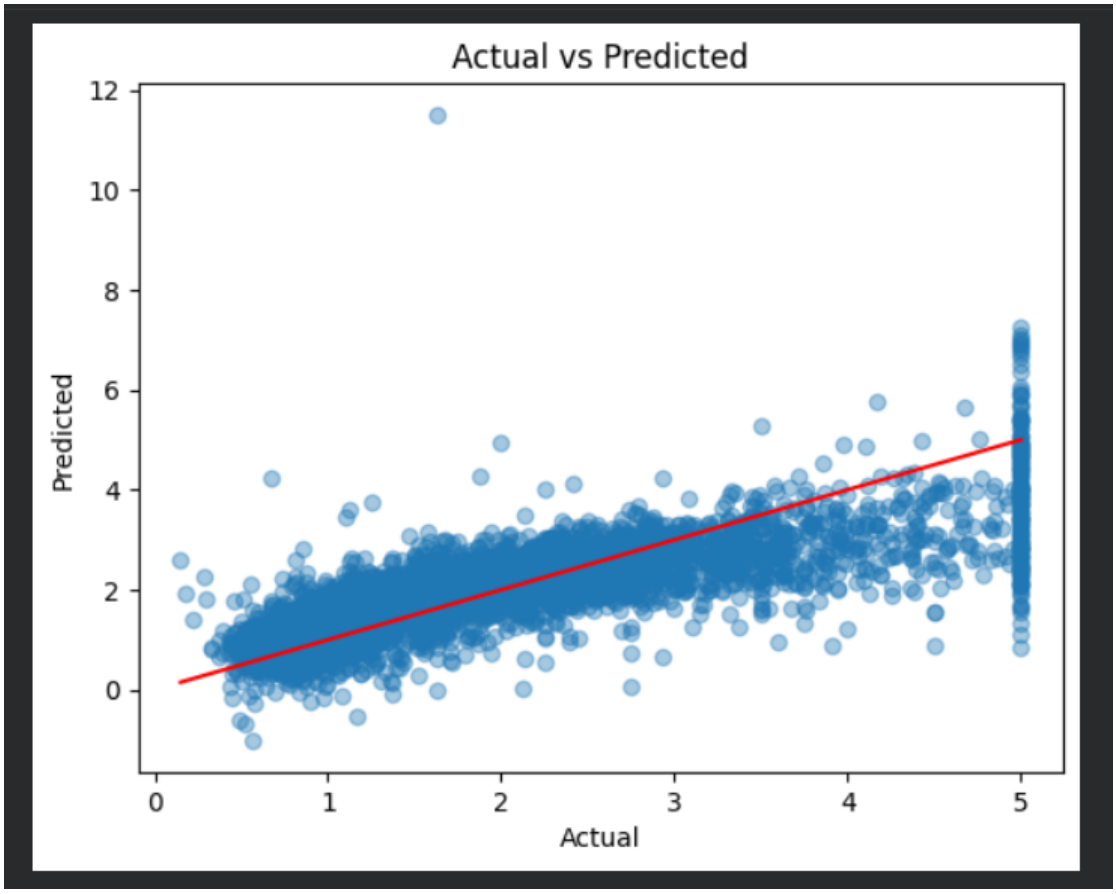
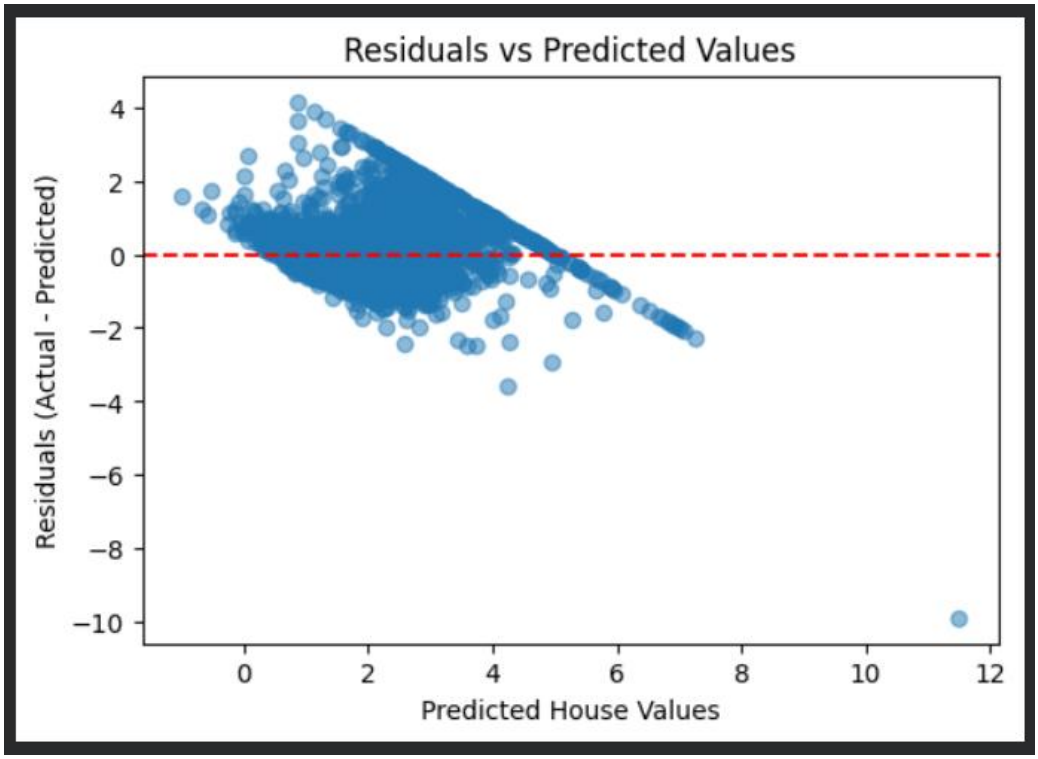


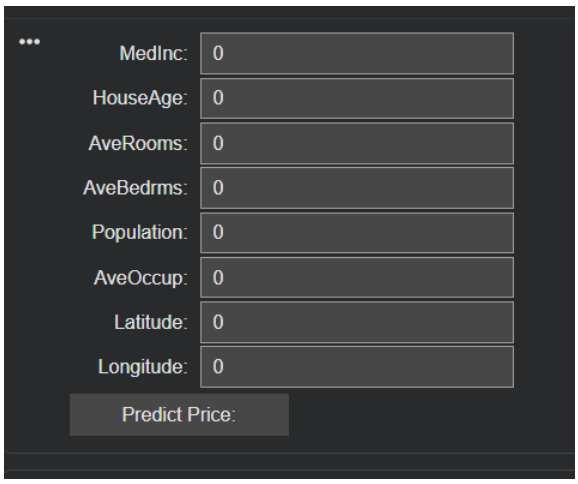
Figure 3: *Residual Analysis*



5. Bonus: Adding a User Interface

I wanted to make the model more "usable," so I built an interactive interface in Google Colab using ipywidgets. Instead of looking at raw code, you can now use sliders to input different neighborhood features and get an instant price prediction from the model.

Figure 4: *UI Output:*



NOTICE: Here, I have used Google colab, as my Jupyter notebook was functioning unwell. Otherwise, the task was completed as assigned.