**Category C , task 1**

**1. Problem Definition**

The project aims to predict housing prices in California based on features like location, population, number of rooms, etc. The goal is to create a regression model that accurately estimates the median\_house\_value for each district, using 3 different training models (linear regression, decision tree regression and random forest regression) we determine which one is the most useful and apply it to our clean data set.

**2. Data Collection**

The dataset (housing.csv) was loaded from a file in Google Colab. It contains various features such as:

* longitude, latitude
* total\_rooms, total\_bedrooms
* population, households
* median\_income, ocean\_proximity
* median\_house\_value (target variable)

Datasets can also be downloaded from Kaggle and other sources.

**3. Data Preprocessing**

Key preprocessing steps included:

* Handling missing values in total\_bedrooms using **median imputation**.
* Separating numerical and categorical attributes.
* Encoding categorical data using **One-Hot Encoding** and **Ordinal Encoding**.
* Scaling numerical features with **StandardScaler** to bring them to a common scale.
* Creating new features (feature engineering), such as:
  + bedrooms\_per\_household
  + population\_per\_household
  + rooms\_per\_household

By using median imputation instead of dropping the rows with missing values we are able to keep important information and rows so that we do not lose out on valuable and insightful data .

One hot encoding also allows us to use data as numbers so its possible to apply our models to it because values like “INLAND” cannot be used and we need numerical values

**4. Train-Test Splitting**

To train and evaluate the model properly:

* A random split (train\_test\_split) was done first.
* Then, **Stratified Sampling** based on income\_cat was used for a more balanced train-test distribution, ensuring the median\_income distribution is preserved.
* The ratio was 33% percent for test dataset and 67% for training data set (ratio was given by author as he has a better understanding of data even though a 20: 80 split is usually used)

**5. Exploratory Data Analysis (EDA)**

The dataset was visualized using:

* Histograms
* Correlation heatmaps
* Geographical scatter plots showing median house values and population  
  These visualizations revealed relationships and trends that informed feature engineering decisions.

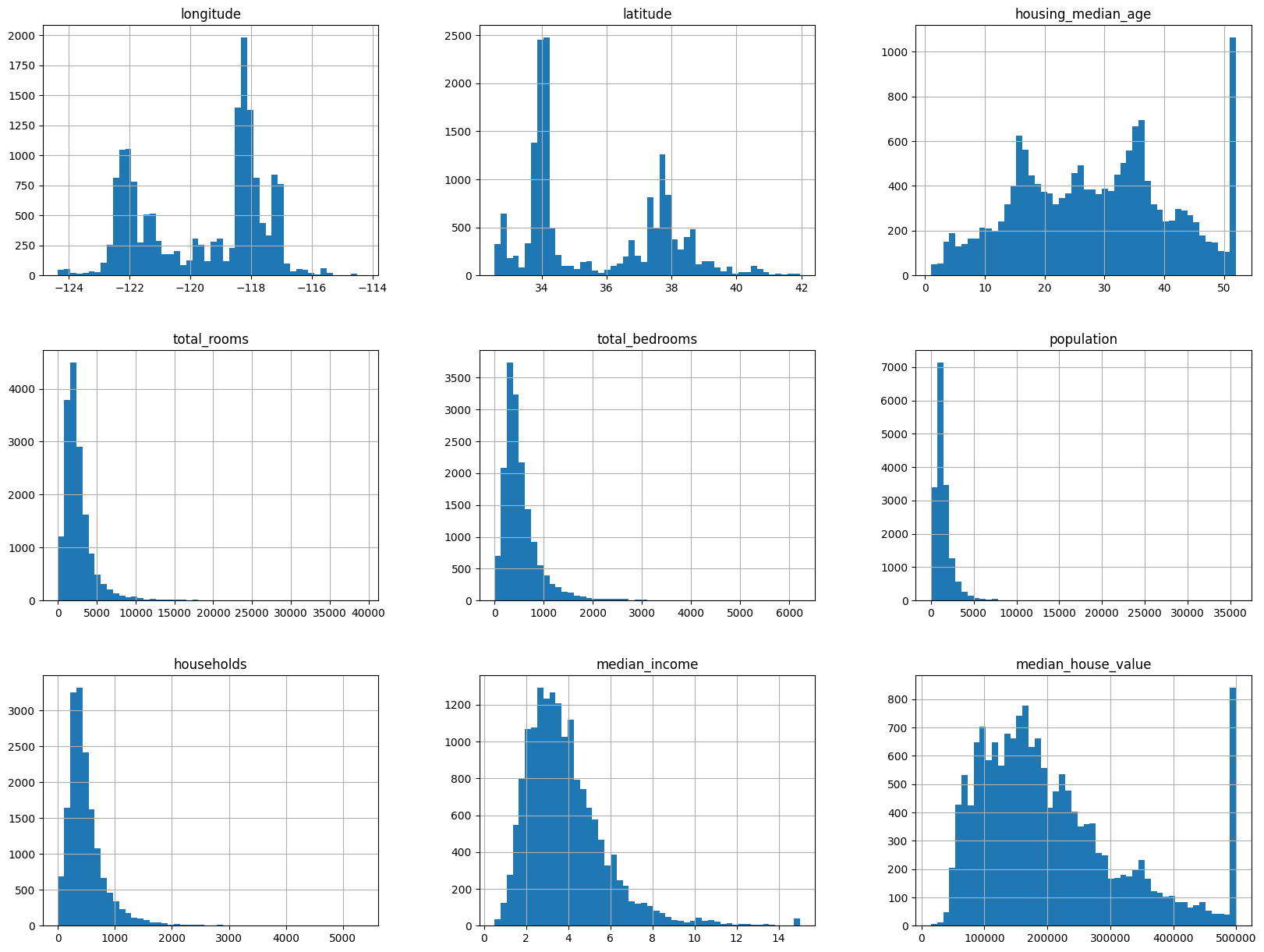


Figure 1. Histograms for each feature

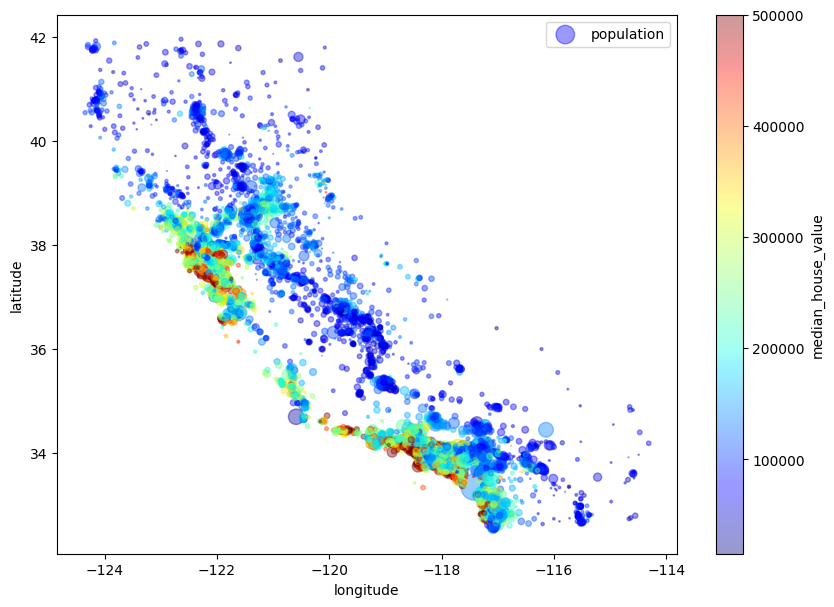


Figure 2. Geological scatter plots

**6. Feature Engineering Pipeline**

You created a robust pipeline that:

* Fills missing values
* Adds new derived features
* Encodes categorical variables
* Scales numerical data  
  This pipeline is modular and reusable across training and testing sets.

**7. Model Training**

Three models were trained and compared:

* **Linear Regression**
* **Decision Tree Regressor**
* **Random Forest Regressor**

Each model was trained using the transformed training data.

**8. Model Evaluation**

Models were evaluated using:

* **RMSE (Root Mean Squared Error)**
* **MAE (Mean Absolute Error)**
* **Cross-Validation** (10-fold CV) to assess stability

You also visualized predictions vs. actual values and plotted residuals to assess the errors.

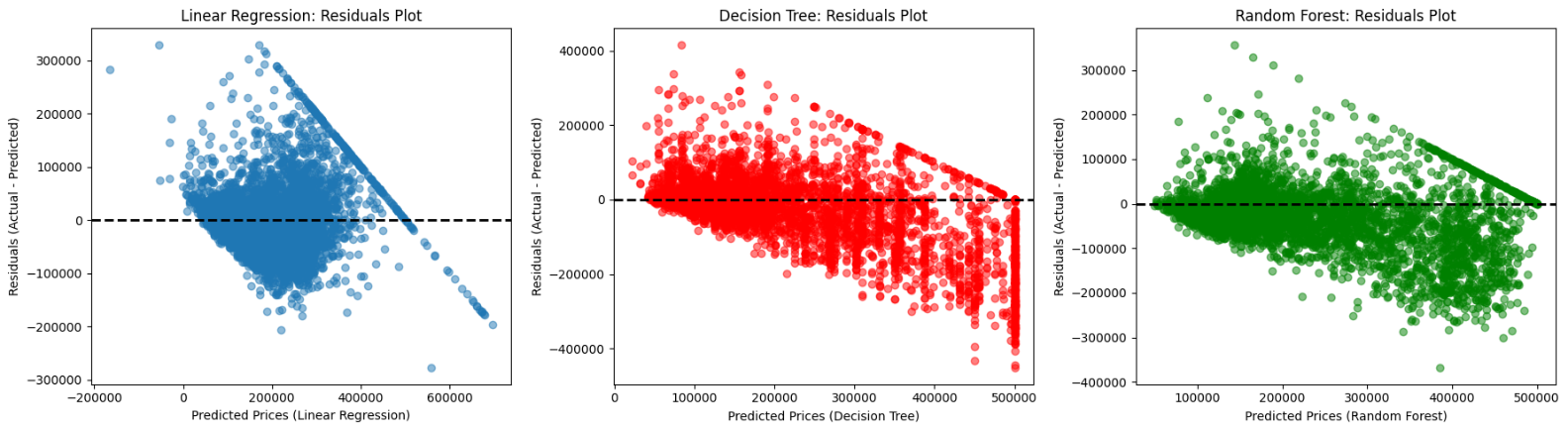


Figure 3. Actual vs predicted values

**9. Model Selection**

From the evaluation:

* Linear Regression underperformed (high bias)
* Decision Tree had low training error but overfit the data
* Random Forest provided the best balance of bias-variance tradeoff

**9. Deployment**

Using the streamlit library and their token we get a URL for a website where we can deploy the model for use



Figure 4. Stream\_lit web dashboard

**How it works:**

The user changes the sliders which are all of are input features , based on the input features the model runs a random forest regression and outputs the predicted median house value