🏠 House Price Prediction - Regression Project

📘 Overview

This project focuses on predicting house prices in California using a classic regression approach on the **California Housing Dataset**. It walks through a complete Machine Learning pipeline including data ingestion, visualization, feature engineering, model training, and evaluation.

📥 1. Data Collection

- Dataset Source: https://github.com/ageron/data/raw/main/housing.tgz
- Data is downloaded and extracted from a .tgz file.
- Data is loaded using Pandas: housing.csv.

🔍 2. Data Exploration

- Basic info: housing.head(), housing.info(), and housing.describe()
- Value counts of categorical attribute ocean_proximity.

📊 3. Visualizations

- **Histograms** of all numerical features.
- **Income category distribution** using pd.cut().
- **Geographical scatter plots**:
  - Basic latitude/longitude visualization
  - Color-coded by house value and population
- **Correlation matrix** and **scatter matrix** to identify key relationships.

🔀 4. Train-Test Splitting

- Random permutation and shuffling
- Hash-based splitting using CRC32
- train_test_split from Scikit-Learn with random_state=42 for reproducibility
- Stratified sampling based on income_cat

🧹 5. Data Cleaning

- Handling missing values in total_bedrooms
    - Drop rows (optional)
    - Fill using median via SimpleImputer

📐 6. Feature Engineering

- Added attributes:
    - rooms_per_household
    - bedrooms_per_room
    - population_per_household
- Created preprocessing pipelines:
    - Pipeline for numerical features
    - ColumnTransformer for combining pipelines
    - OneHotEncoder for categorical variables

🧠 7. Model Training

Algorithms Used:

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

Evaluation:

- mean_squared_error for RMSE calculation
- cross_val_score for k-fold validation
- Grid Search for hyperparameter tuning on Random Forest

🧪 8. Final Evaluation

- Tested on hold-out test set
- Metrics: RMSE, confidence intervals

- Best model performance achieved with tuned **Random Forest Regressor**

📈 9. Visual Outputs

- Attribute histograms
- Scatter plots (geo-locations & correlation plots)
- Bar plot of income categories
- Optional: correlation heatmap

✅ Key Findings

- median_income is the most predictive feature.
- Feature engineering significantly improved model performance.
- Random Forest performed better than both Decision Tree and Linear Regression.

🧩 Tools & Libraries

- Python, Pandas, NumPy, Scikit-learn
- Matplotlib for plotting
- Jupyter Notebook for execution

📌 Next Steps (Suggestions)

- Deploy the model via Flask or FastAPI.
- Store results and predictions in a database.
- Track experiment metrics using MLflow or Weights & Biases.

Project by Praneeth | Guided by Aurelien Geron (Hands-On ML)