**California House Price Prediction Report**

**1. Objective**

The notebook aims to predict California house prices using machine learning models. It explores data preprocessing, visualization, and modeling to build an accurate prediction system.

**2. Dataset**

* **File Used**: housing.csv
* **Instances**: 20,640
* **Missing Values**: total\_bedrooms has 20433 non-null values (missing data present)
* **Features:**
  + longitude, latitude: Geographic coordinates
  + housing\_median\_age
  + total\_rooms, total\_bedrooms, population, households
  + median\_income
  + ocean\_proximity (categorical)
  + median\_house\_value (Target variable)

**3. Data Preprocessing**

* **Handling Missing Values**:
  + The total\_bedrooms feature has missing values, which were filled using the **median** to address right skewness.
* **Feature Encoding**:
  + ocean\_proximity (categorical feature) was likely converted into numerical form.
* **Feature Scaling**:
  + No explicit scaling was observed in the extracted portion, but it may have been included later.

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

* **Distributions**:
  + total\_bedrooms: Right-skewed; missing values filled with median.
  + median\_income: Distribution visualized with a histogram.
* **Correlations**:
  + The notebook likely explores correlations between numerical features and median\_house\_value.
* **Geospatial Data Analysis**:
  + Latitude & longitude were plotted to visualize housing locations.

**5. Feature Engineering**

* Potential transformations include:
  + Creating new features based on existing columns (e.g., rooms\_per\_household, bedrooms\_per\_room).
  + Encoding categorical variables.

**6. Machine Learning Models Used**

The notebook likely implements multiple models, including:

* **Linear Regression**
* **Decision Trees**
* **Random Forests**
* **Gradient Boosting (e.g., XGBoost)**

It may also include hyperparameter tuning using **GridSearchCV**.

**7. Model Evaluation**

* **Metrics Used**:
  + **RMSE (Root Mean Squared Error)**
  + **R² Score (Coefficient of Determination)**
* **Train-Test Split**:
  + The dataset was likely split into **training and testing sets** (e.g., 80%-20%).

**8. Results & Conclusion**

* **Best Model**: Random forest Regressor is the best model. Likely determined based on RMSE and R² score.

**Pickle file is saved and downloaded and used for deployment**

## . Deployment Strategy and API Usage Guide

### Deployment Strategy:

* **Flask API:**
  + Built a REST API using Flask to serve model predictions.
  + The API accepts a JSON request with house features and returns a predicted price.

### Running the API Locally:

1. **Install Dependencies:**

pip install -r requirements.txt

1. **Run Flask Application:**

python app.py

1. **Make a Prediction Request:**
2. curl -X POST "http://127.0.0.1:5000/predict" \
3. -H "Content-Type: application/json" \

-d '{"features": [3, 2, 1500]}'

1. **Expected Response:**

{"predicted\_price": 450000}

## Conclusion:

This project successfully implemented a machine learning model for house price prediction using Random Forest. The API allows users to interact with the model efficiently. Future improvements can include testing with more advanced models like XGBoost or deploying the API using cloud services such as AWS or Azure.