California Housing Prices – Final Report

1. Project Overview

Goal:
The goal of this project was to build a machine learning model that can predict the **median house value** in California based on factors like income, population density, and location. This type of model can be useful for real estate professionals, urban planners, or local governments to better understand housing trends and guide future planning or investment decisions.

Dataset:
I used the **California Housing dataset**, which comes built-in with Scikit-learn. It's based on data from the 1990 U.S. Census.

Target Variable:
`MedianHouseValue` — the median price of houses in a block group (values are scaled in $100,000s).

---

2. Data Cleaning & Preprocessing

The dataset was already in great shape—no missing values or major issues.

Here's what I did:

- Renamed the target column for better clarity
- Double-checked feature data types and distributions
- Since all columns were numeric, I didn't need to perform any categorical encoding or imputations

---

3. Exploratory Data Analysis (EDA)

I spent some time exploring the data visually and statistically to understand how different features relate to housing prices.

Key insights:

- `MedInc` (Median Income) had the **strongest positive correlation** with home prices
- Features like `AveRooms`, `HouseAge`, and `AveOccup` showed **nonlinear relationships** with the target
- A correlation heatmap showed some **multicollinearity**, especially between `AveRooms` and `AveBedrms`
- Geographic features like `Latitude` and `Longitude` clearly impacted prices—homes closer to the coast were more expensive

Visuals I used:

- Histogram of the target variable
- Correlation heatmap
- Scatter plots, especially `MedInc` vs. `MedianHouseValue`

---

4. Modeling Approach

To evaluate performance, I trained two models:

- **Linear Regression**: A straightforward and explainable model to set a performance baseline
- **Random Forest Regressor**: A more powerful ensemble method that handles complex patterns and nonlinearities

I used an 80/20 **train-test split** to evaluate model performance.

---

5. Model Performance

Model	R² Score	RMSE
Linear Regression	0.58	~0.73
Random Forest	0.81	~0.47

The **Random Forest** model performed significantly better in terms of both R² and RMSE.

---

6. Key Findings

Here are the top features that most influenced the model's predictions:

- **MedInc** – Higher median income usually means higher home values
- **AveRooms** – More rooms in a house tends to reflect higher value
- **HouseAge** – Surprisingly, older homes weren't always cheaper
- **AveOccup** – Reflects how crowded households are, which had a complex relationship with price
- **Latitude & Longitude** – Coastal proximity clearly increased housing prices

---

7. Recommendations

- Use **Random Forest** for deployment since it outperformed the simpler model
- Consider adding **more detailed location data** like ZIP codes or school zones to improve prediction accuracy
- Apply **feature scaling and hyperparameter tuning** for better generalization

8. Next Steps

- Run a **Grid Search or Random Search** to tune the Random Forest model
- Bring in **external datasets** (e.g., proximity to highways or schools)
- Try **PCA or other dimensionality reduction** techniques to improve model speed and performance
- Build a **dashboard** for interactive housing price predictions

9. Tools & Technologies

- **Python** (Pandas, NumPy, Scikit-learn)
- **Jupyter Notebook**
- **Matplotlib** and **Seaborn** for visualizations
- **Git & GitHub** for version control and sharing

10. Credits

- Dataset: California Housing Data from Scikit-learn
- Project completed as part of the **Springboard Data Science Career Track – Capstone Two**

Appendix

- `model_metrics.txt`: Includes detailed performance metrics
- `README.md`: GitHub documentation for the full project