California Housing Price Prediction

A comprehensive regression analysis project built using Python and **Streamlit** to predict median housing prices in California. This application allows users to explore data, train models with flexible configurations, evaluate model performance, and make predictions — all through an interactive web interface.

Table of Contents

Project Objectives

The main goals of this project are:

- Predict the **Median House Value** in California using demographic and geographic features.
- Implement **end-to-end regression modeling**, including EDA, feature engineering, model building, and evaluation.
- Provide a **user-friendly interface** for training, evaluating, and deploying models interactively using Streamlit.

Dataset Description

The dataset is sourced from `sklearn.datasets.fetch_california_housing` and includes:

- **Instances**: 20,640

- **Features** (original):
  - `MedInc`: Median income in block
  - `HouseAge`: Median house age
  - `AveRooms`: Average number of rooms
  - `AveBedrms`: Average number of bedrooms
  - `Population`: Block population
  - `AveOccup`: Average occupants per household
  - `Latitude`, `Longitude`: Geographical coordinates
- **Target**:
  - `MedHouseVal`: Median house value in $100,000s

Data Preprocessing & Feature Engineering

Cleaning

- Checked for and confirmed the absence of **missing values**.
- Applied **outlier capping** (winsorization) to reduce the effect of extreme values using the 1st and 99th percentiles.

Feature Engineering

- Added multiple domain-driven features:
  - `RoomsPerPopulation = AveRooms * AveOccup`
  - `BedroomsRatio = AveBedrms / AveRooms`
  - `IncomePerRoom = MedInc / AveRooms`
  - `LatxLong = Latitude * Longitude` (interaction term to reflect spatial effects)

Normalization

- Standardized features using `StandardScaler`.

Application Architecture

Built using **Streamlit**, the app is divided into four main pages:

1. Data Exploration

- Descriptive statistics

- Missing values check
- Correlation matrix (heatmap)
- Distribution plots (histograms, boxplots)
- Geospatial plot (colored scatter plot with population-based size)

2. Model Training & Evaluation

- **Model options**:
  - Linear Regression
  - Ridge, Lasso, ElasticNet
  - Polynomial Regression (degree=2)
- **Feature Selection**: via `SelectKBest(f_regression)`
- **Dimensionality Reduction**: Optional PCA
- **Cross-Validation**: Customizable folds
- **Metrics Calculated**:
  - MSE, RMSE, MAE, R^2 (for both train/test)
- **Model Analysis**:
  - Coefficient ranking
  - Feature importance visualization
  - Residual plot & Q-Q plot
- **Model Export**: Trained models are saved using `joblib`.

3. Price Prediction

- Dynamic form to input feature values
- Reconstructs engineered features automatically
- Shows predicted price in both $100,000s and $ units
- (If supported by model): Visualizes **feature contributions**

4. Project Report

- Presents project overview, findings, and technical notes

Model Training & Evaluation

Modeling Pipeline:

1. **Preprocessing Steps**:
   a. Scaling

   b. Optional PCA
   c. Optional K-Best feature selection
2. **Regression Model** (selectable)
3. **Optional PolynomialFeatures transformation**

Regularization Support:

- Ridge, Lasso, and ElasticNet allow tuning of `alpha` via Streamlit sliders.

Cross-Validation:

- Implemented using `cross_val_score()` from Scikit-learn.
- Configurable number of folds (3 to 10).

Evaluation Metrics:

Metric	Description
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
R^2	Coefficient of Determination

Prediction Module

- Loads the trained model from disk.
- Takes input features from a web form.
- Automatically computes engineered features.
- Outputs:
  - Predicted median house value in both 100k and dollar units.
  - If model supports `.coef_`, shows:
    - Contribution of each feature
    - Bar plot of top contributors

Results Summary

- **Best Performing Model**: Ridge Regression
- **Test R^2 Score**: ~0.62
- **Top Predictive Features**:
  - `MedInc` (median income)

- `Latitude`, `Longitude`
    - `AveRooms`
    - `RoomsPerPopulation`
- **Model saved** as `housing_model.pkl`

Challenges & Solutions

Challenge	Solution
Multicollinearity (AveRooms, AveBedrms)	Used Lasso/Ridge and SelectKBest
Outliers skewing the distribution	Winsorization (1st, 99th percentile)
Non-linearity in features	Polynomial regression + PCA
Model interpretability in high-dim. space	Restricted polynomial degree to 2 and visualized top features

Future Enhancements

- Integrate **XGBoost**, **Random Forest**, or **Gradient Boosting**
- Perform **hyperparameter tuning** using GridSearchCV/Optuna
- Add **real-time map visualizations** with Folium or Plotly
- Enable **model comparison dashboards**
- Convert to **full-stack app** using Flask + Streamlit front-end

Dependencies

```
numpy
pandas
matplotlib
seaborn
scikit-learn

statsmodels
joblib
streamlit
pillow
```