Title: California Housing Price Prediction Using Machine Learning

Student Name: Anshul Narendra Parate

Roll No: A1_14

Department: Artificial Intelligence and Machine Learning

Academic Year: 2025

---

1. Problem Analysis

The goal of this mini project is to predict the median house value in California districts using multiple regression and ensemble machine learning algorithms. The dataset used for this project is the **California Housing Dataset**, which contains details such as geographical location, population, number of rooms, income, and proximity to the ocean.

The problem is a **regression problem**, where the output variable is continuous — the **median_house_value**.

The project aims to:

- Analyze relationships between socioeconomic and geographic variables and housing prices.
- Apply multiple regression and ensemble algorithms for predictive modeling.
- Perform comparative analysis to identify the best-performing model.
- Build a user-friendly GUI using React and connect it with FastAPI for model inference.

---

2. Data Preprocessing and Exploratory Data Analysis (EDA)

Dataset Source: California Housing Dataset from Scikit-learn / Kaggle.

Data Understanding:
The dataset contains 10 major attributes:

1. longitude – west-east location

2. latitude – north-south location
3. housingMedianAge – median house age
4. totalRooms – total rooms per block
5. totalBedrooms – total bedrooms per block
6. population – total number of residents
7. households – total number of housing units
8. medianIncome – median income of residents
9. medianHouseValue – target variable
10. oceanProximity – categorical variable (INLAND, NEAR BAY, etc.)

Data Cleaning Steps:

- Handled missing values by imputing with mean/median where required.
- Converted the categorical variable “oceanProximity” into numerical format using one-hot encoding.
- Detected and handled outliers using interquartile range (IQR) method.
- Scaled numerical features using StandardScaler to normalize ranges.

Exploratory Data Analysis Findings:

- Strong positive correlation between median_income and median_house_value.
- total_rooms, total_bedrooms, population, and households are interrelated features.
- ocean_proximity strongly affects both income and price distribution.
- Geographic visualization revealed that houses near the coast are significantly higher in value.

3. Model Development and Hyperparameter Tuning

All models were trained using **Google Colab** for computational efficiency and **VS Code** for backend and API integration.

Models Implemented:

1. Linear Regression
2. Multiple Linear Regression
3. Decision Tree Regressor
4. Random Forest Regressor
5. XGBoost Regressor
6. LightGBM Regressor
7. Gradient Boosting Regressor

Each model was trained and evaluated on R^2, MAE, MSE, and RMSE metrics.

Hyperparameter Tuning Techniques Used:

- GridSearchCV
- RandomizedSearchCV
- Manual tuning for tree depth, learning rate, and number of estimators in ensemble models.

Model Comparison Results:
Linear models provided a good baseline but struggled with non-linear patterns.
Random Forest, XGBoost, and LightGBM achieved the highest accuracies.

Best Model: **LightGBM (Tuned)**
R^2 Score: 0.8484
MAE: 29,601.66
MSE: 1,986,363,475

RMSE: 44,568.64
Average Prediction Accuracy: 83.19%

---

4. Innovation in Project

Technology Stack and Implementation Details:

- **Frontend:** React.js
    - Developed a clean and responsive GUI for house price prediction.
    - Integrated input forms for all feature values (longitude, latitude, median income, etc.).
    - Displayed predicted price dynamically after API response.
- **Backend:** FastAPI
    - Designed RESTful API endpoints for model inference.
    - Integrated serialized LightGBM model using Pickle.
    - Implemented data validation using Pydantic models.
- **Model Training and Experimentation:** Google Colab
    - Utilized Colab GPU runtime for faster model training.
    - Saved trained models as .pkl files for backend deployment.
- **Development Environment:** Visual Studio Code
    - Used for writing and testing FastAPI code and integrating with frontend.
    - Managed virtual environments and dependencies.
- **Deployment:**
    - The project is designed for global deployment.

- The FastAPI backend can be hosted on Render or HuggingFace Spaces.
- The React frontend can be deployed on Vercel or Netlify.

Innovative Features:

- Real-time prediction interface with React and FastAPI.
- Dynamic error checking and user input validation.
- Modular pipeline for easy future expansion (e.g., image or location-based prediction).
- Option to retrain model with user-provided data for customization.

---

5. Project Knowledge and Question Answering

Key Learning Outcomes:

- Understood end-to-end ML pipeline: data preprocessing, model selection, tuning, and deployment.
- Learned to handle both structured and categorical data efficiently.
- Gained hands-on experience integrating machine learning models with modern web frameworks.
- Applied advanced ensemble algorithms like XGBoost and LightGBM for regression.
- Learned deployment workflow involving backend APIs and frontend communication.

Sample Questions and Answers:

1. **Q:** Why did you choose LightGBM as your final model?
   A: LightGBM provided the best trade-off between accuracy, speed, and generalization. It also performed well on large feature sets and was less prone to overfitting compared to XGBoost in our experiments.

2. **Q:** Why use React and FastAPI instead of Flask?
   A: FastAPI offers faster performance, type checking with Pydantic, and better async support. React was chosen for its modular component structure and modern UI capabilities.

3. **Q:** What are the major factors influencing house prices?
   A: Median income, ocean proximity, and geographic coordinates (latitude, longitude) were found to be the most significant factors.

4. **Q:** How did you validate your model performance?
   A: The dataset was split into training (80%) and testing (20%) sets. We used cross-validation and performance metrics like R^2, MAE, MSE, and RMSE.

5. **Q:** What improvements can be made in the future?
   A: Adding geospatial visualization, integrating live API data (e.g., Zillow), and deploying as a full-scale web app with cloud database support.

---

6. Conclusion

The project successfully demonstrates the application of multiple regression algorithms to predict California housing prices. Through proper data preprocessing, model tuning, and feature analysis, we achieved an R^2 score of 0.84 using the LightGBM model.

The integration of a React-based GUI and a FastAPI backend makes this a complete end-to-end solution. The system is scalable, interpretable, and can be further extended for real-world applications like real-estate price estimation or rental prediction systems.

This project showcases a complete Machine Learning development lifecycle — from problem analysis to deployment.