🔍 California Housing Price Prediction – Full Stack ML Project Summary

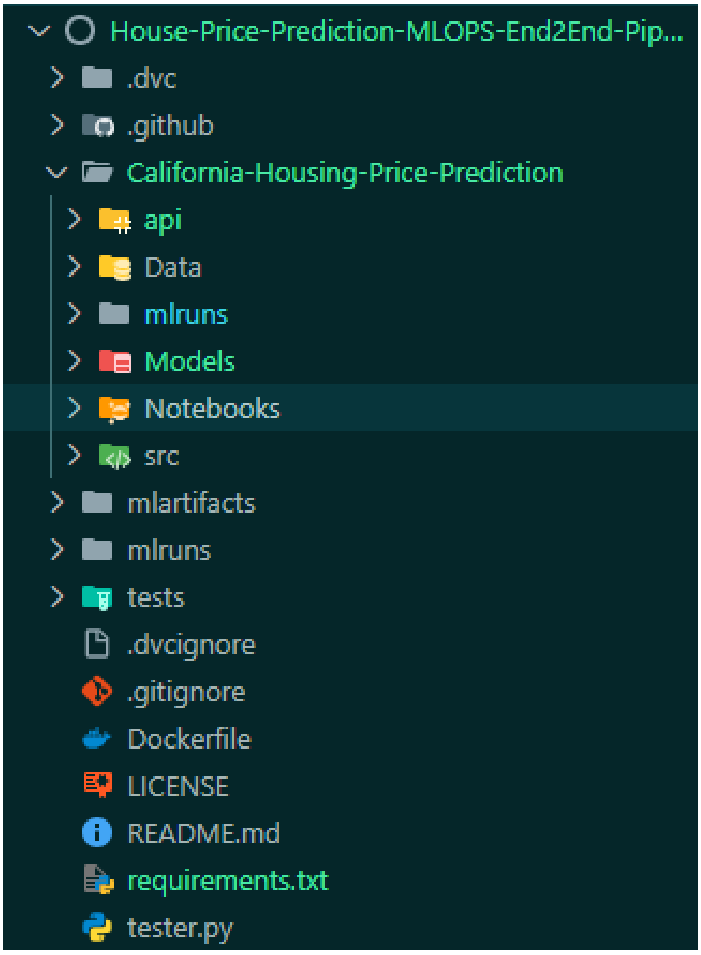
📁 Assignment Structure Overview

Project Details:-

GitHub Repo Link: - https://github.com/Atharv-Chaudhari-Bits/House-Price-Prediction-MLOPS-End2End-Pipeline

Docker Hub Link: - https://hub.docker.com/r/atharvchaudharibits/california-housing-app

Directory Structure –



```
House-Price-Prediction-MLOPS-End2End-Pip...
  .dvc
  .github
  California-Housing-Price-Prediction
    api
    Data
    mlruns
    Models
    Notebooks
    src
  mlartifacts
  mlruns
  tests
  .dvcignore
  .gitignore
  Dockerfile
  LICENSE
  README.md
  requirements.txt
  tester.py
```

Built and deployed an end-to-end California Housing Price Prediction pipeline using MLOps best practices. Implemented model training, tracking, and versioning with Git, DVC, and MLflow. Packaged the model as a REST API using FastAPI and containerized it with Docker. Automated CI/CD via GitHub Actions and set up basic logging with optional monitoring support.

✨ ML Models Implemented

We trained and compared several regression models using scikit-learn:

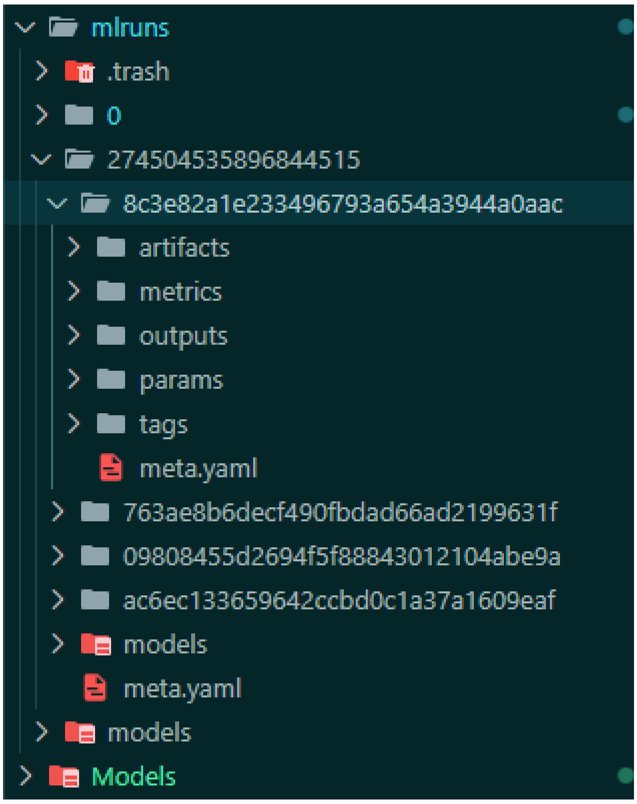
- `LinearRegression`
- `DecisionTreeRegressor`
- `RandomForestRegressor`
- `GradientBoostingRegressor`

Each model was evaluated and logged using **MLflow**, capturing:

- Hyperparameters
- Metrics (R^2, RMSE, MAE)
- Model artifacts
- Training metadata

MLflow created separate directories under `mlruns/<experiment-id>/<run-id>/`, which include:

- `params/`: model hyperparameters
- `metrics/`: performance metrics
- `artifacts/`: saved models
- `tags/`: metadata like model name or author



```
mlruns
  .trash
  0
  274504535896844515
    8c3e82a1e233496793a654a3944a0aac
      artifacts
      metrics
      outputs
      params
      tags
      meta.yaml
    763ae8b6decf490fbdad66ad2199631f
    09808455d2694f5f88843012104abe9a
    ac6ec133659642ccbd0c1a37a1609eaf
    models
    meta.yaml
  models
Models
```

🐳 Docker Setup (Model Packaging + API)

The project is fully containerized using **Docker**, enabling smooth deployment across environments.

Key files used:

- `Dockerfile`: Defines the build, installing FastAPI, MLflow, scikit-learn, etc.
- `.dockerignore`: Ignores unnecessary files like `__pycache__`, notebooks, and MLflow trash.
- Docker builds the image with API code and launches a FastAPI server.

Main functionalities containerized:

1. Load the best MLflow model from registry
2. Serve predictions via REST (JSON) and HTML form (`/form`)
3. Logging for inference time, status, and model used

🌐 FastAPI Features

FastAPI is used to expose the model through:

- `/predict`: accepts JSON inputs and returns prediction
- `/form`: accepts user inputs via HTML form and returns output
- `/retrain`: accepts new CSV uploads for dynamic model retraining

Pydantic schemas are used for request validation.
All logs (including prediction attempts) are enhanced with **emojis/symbols** for better visibility in both console and file logs.

Logs are saved to app_log.log file

🔁 CI/CD with GitHub Actions

In `.github/workflows/`, I've set up a GitHub Actions pipeline that:

- Installs dependencies
- Runs test suite

📦 Data Versioning with DVC

DVC tracks both raw and processed data (`Data/raw`, `Data/processed`) and integrates with Git:

- Ensures reproducibility by locking datasets to specific model versions
- DVC files (`.dvc/`) and `dvc.yaml` track preprocessing and training steps
- Supports pushing large data to remote (e.g., S3, GDrive)

📦 MLflow Tracking and Model Management

MLflow tracks:

- Training experiments
- Model performance
- Parameters and artifacts
- Registered models under unique IDs

✅ Logging and Monitoring

Logging is implemented for:

- API requests
- Inference status
- Retraining results

Logs include timestamps, status symbols (✅, ❌), and are stored in both console and file format.

⚙ Retraining Pipeline via API

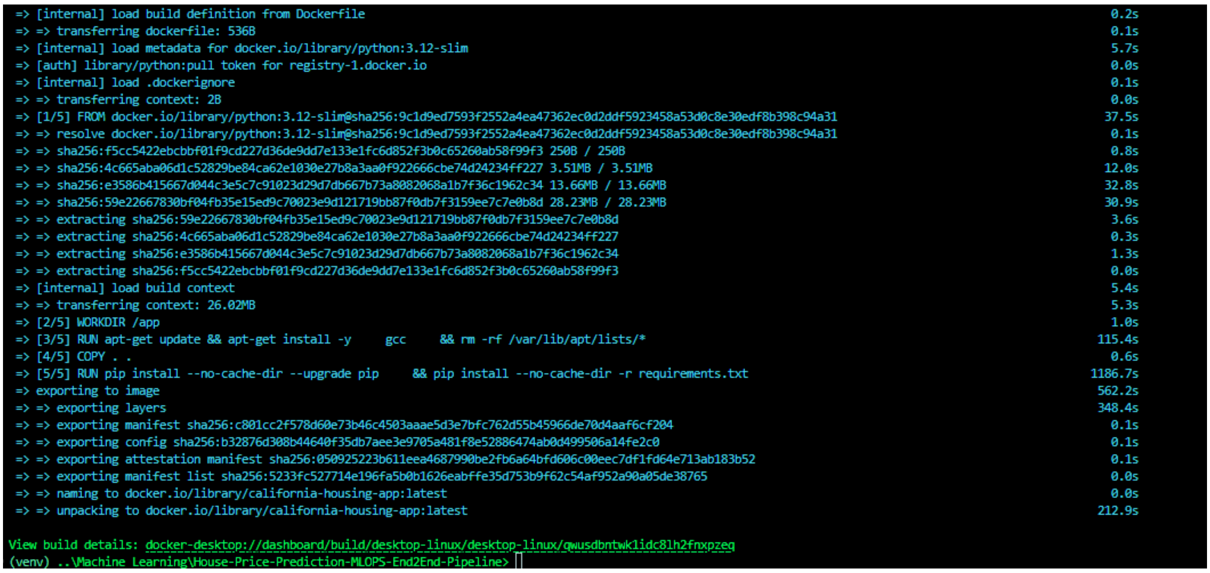
The `/retrain` endpoint supports:

- Uploading a new CSV via HTML form
- Reading and preprocessing new data
- Retraining the best model
- Logging it to MLflow
- Saving the new model to registry

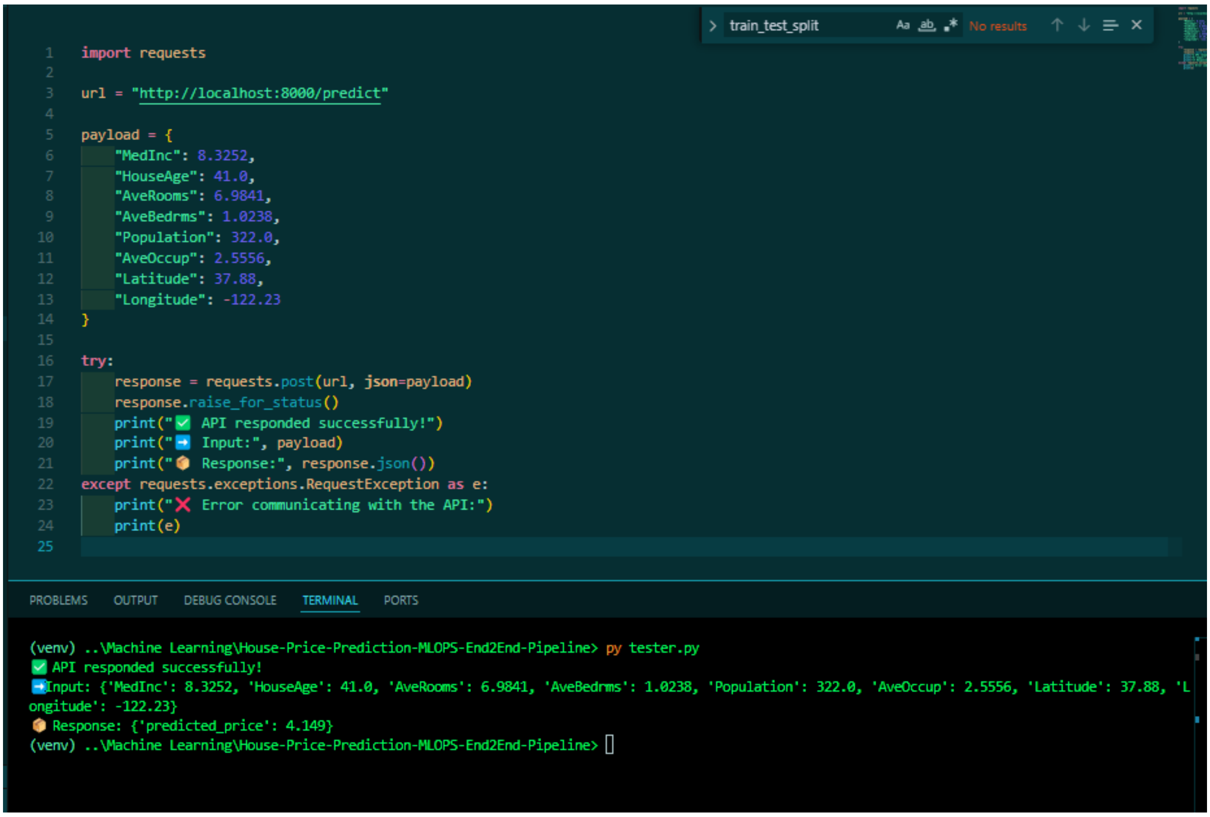
⍰ Testing

Setup for `tests/` directory includes:

- Unit tests for preprocessing and inference
- API integration tests using `pytest`
- Retraining test cases for model updates

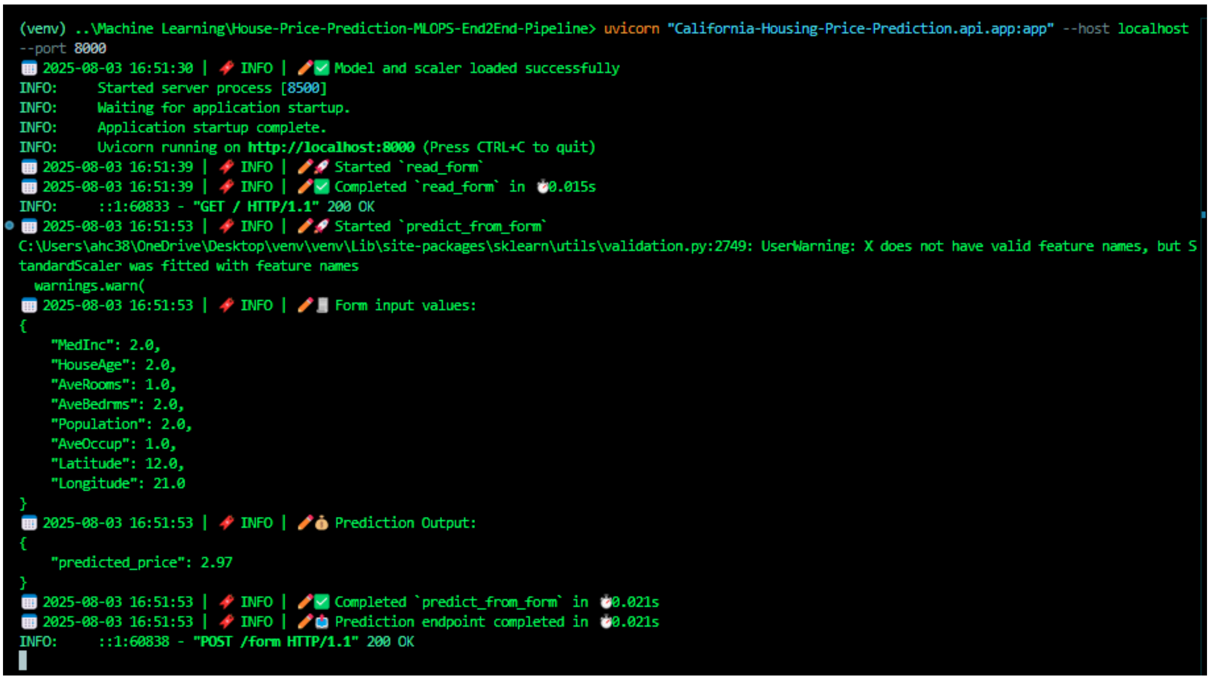


```
=> [internal] load build definition from Dockerfile                                                                                           0.2s
=> => transferring dockerfile: 536B                                                                                                           0.1s
=> [internal] load metadata for docker.io/library/python:3.12-slim                                                                            5.7s
=> [auth] library/python:pull token for registry-1.docker.io                                                                                  0.0s
=> [internal] load .dockerignore                                                                                                              0.1s
=> => transferring context: 2B                                                                                                                0.0s
=> [1/5] FROM docker.io/library/python:3.12-slim@sha256:9c1d9ed7593f2552a4ea47362ec0d2ddf5923458a53d0c8e30edf8b398c94a31                     37.5s
=> => resolve docker.io/library/python:3.12-slim@sha256:9c1d9ed7593f2552a4ea47362ec0d2ddf5923458a53d0c8e30edf8b398c94a31                      0.1s
=> => sha256:f5cc5422ebcbbf01f9cd227d36de9dd7e133e1fc6d852f3b0c65260ab58f99f3 250B / 250B                                                     0.8s
=> => sha256:4c665aba06d1c52829be84ca62e1030e27b8a3aa0f922666cbe74d24234ff227 3.51MB / 3.51MB                                                12.0s
=> => sha256:e3586b415667d044c3e5c7c91023d29d7db667b73a8082068a1b7f36c1962c34 13.66MB / 13.66MB                                              32.8s
=> => sha256:59e22667830bf04fb35e15ed9c70023e9d121719bb87f0db7f3159ee7c7e0b8d 28.23MB / 28.23MB                                              30.9s
=> => extracting sha256:59e22667830bf04fb35e15ed9c70023e9d121719bb87f0db7f3159ee7c7e0b8d                                                      3.6s
=> => extracting sha256:4c665aba06d1c52829be84ca62e1030e27b8a3aa0f922666cbe74d24234ff227                                                      0.3s
=> => extracting sha256:e3586b415667d044c3e5c7c91023d29d7db667b73a8082068a1b7f36c1962c34                                                      1.3s
=> => extracting sha256:f5cc5422ebcbbf01f9cd227d36de9dd7e133e1fc6d852f3b0c65260ab58f99f3                                                      0.0s
=> [internal] load build context                                                                                                              5.4s
=> => transferring context: 26.02MB                                                                                                           5.3s
=> [2/5] WORKDIR /app                                                                                                                         1.0s
=> [3/5] RUN apt-get update && apt-get install -y     gcc     && rm -rf /var/lib/apt/lists/*                                                115.4s
=> [4/5] COPY . .                                                                                                                             0.6s
=> [5/5] RUN pip install --no-cache-dir --upgrade pip     && pip install --no-cache-dir -r requirements.txt                                 1186.7s
=> exporting to image                                                                                                                       562.2s
=> => exporting layers                                                                                                                      348.4s
=> => exporting manifest sha256:c801cc2f578d60e73b46c4503aaae5d3e7bfc762d55b45966de70d4aaf6cf204                                              0.1s
=> => exporting config sha256:b32876d308b44640f35db7aee3e9705a481f8e52886474ab0d499506a14fe2c0                                                0.1s
=> => exporting attestation manifest sha256:050925223b611eea4687990be2fb6a64bfd606c00eec7df1fd64e713ab183b52                                  0.1s
=> => exporting manifest list sha256:5233fc527714e196fa5b0b1626eabffe35d753b9f62c54af952a90a05de38765                                         0.0s
=> => naming to docker.io/library/california-housing-app:latest                                                                               0.0s
=> => unpacking to docker.io/library/california-housing-app:latest                                                                          212.9s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/qwusdbntwk1idc8lh2fnxpzeq
(venv) ..\Machine Learning\House-Price-Prediction-MLOPS-End2End-Pipeline> 
```

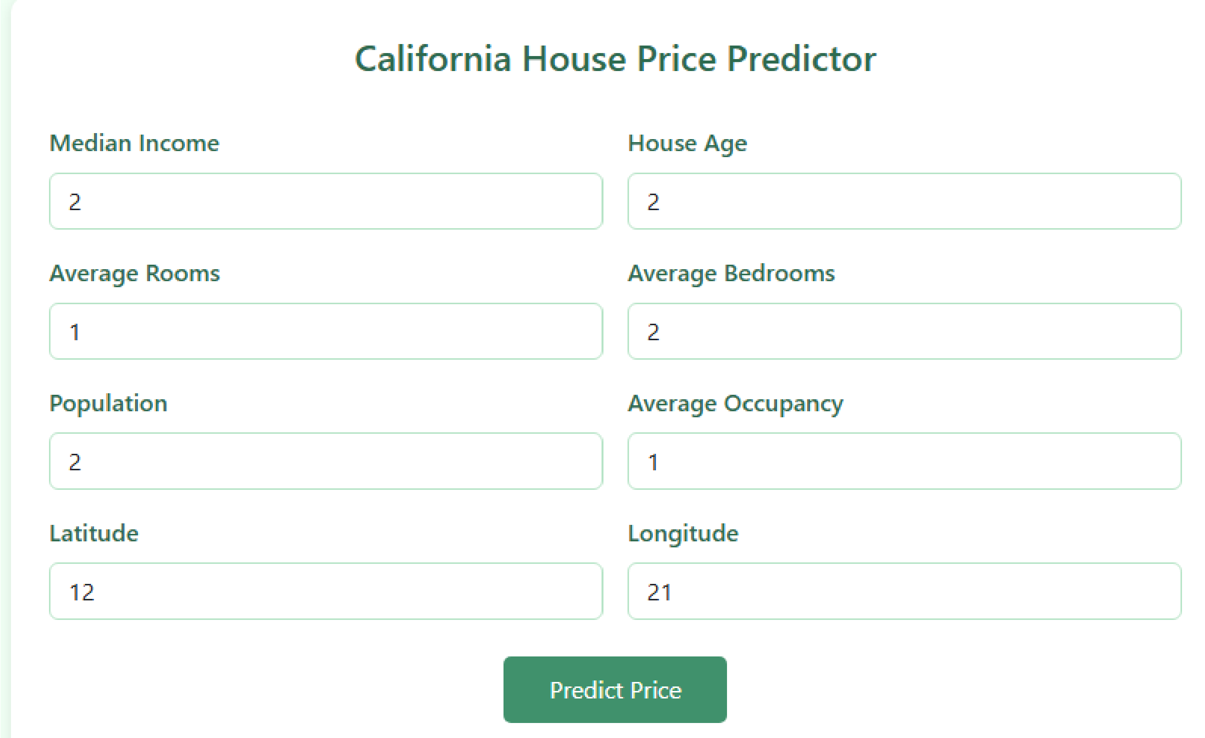


```python
import requests

url = "http://localhost:8000/predict"

payload = {
    "MedInc": 8.3252,
    "HouseAge": 41.0,
    "AveRooms": 6.9841,
    "AveBedrms": 1.0238,
    "Population": 322.0,
    "AveOccup": 2.5556,
    "Latitude": 37.88,
    "Longitude": -122.23
}

try:
    response = requests.post(url, json=payload)
    response.raise_for_status()
    print("✅ API responded successfully!")
    print("➡️ Input:", payload)
    print("📦 Response:", response.json())
except requests.exceptions.RequestException as e:
    print("❌ Error communicating with the API:")
    print(e)

```

```
(venv) ..\Machine Learning\House-Price-Prediction-MLOPS-End2End-Pipeline> py tester.py
✅ API responded successfully!
➡️Input: {'MedInc': 8.3252, 'HouseAge': 41.0, 'AveRooms': 6.9841, 'AveBedrms': 1.0238, 'Population': 322.0, 'AveOccup': 2.5556, 'Latitude': 37.88, 'L
ongitude': -122.23}
📦 Response: {'predicted_price': 4.149}
(venv) ..\Machine Learning\House-Price-Prediction-MLOPS-End2End-Pipeline> 
```



```
(venv) ..\Machine Learning\House-Price-Prediction-MLOPS-End2End-Pipeline> uvicorn "California-Housing-Price-Prediction.api.app:app" --host localhost
--port 8000
📅 2025-08-03 16:51:30 | 📌 INFO | ✏️✅ Model and scaler loaded successfully
INFO:     Started server process [8500]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://localhost:8000 (Press CTRL+C to quit)
📅 2025-08-03 16:51:39 | 📌 INFO | ✏️🚀 Started `read_form`
📅 2025-08-03 16:51:39 | 📌 INFO | ✏️✅ Completed `read_form` in ⏱0.015s
INFO:     ::1:60833 - "GET / HTTP/1.1" 200 OK
📅 2025-08-03 16:51:53 | 📌 INFO | ✏️🚀 Started `predict_from_form`
C:\Users\ahc38\OneDrive\Desktop\venv\venv\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not have valid feature names, but S
tandardScaler was fitted with feature names
  warnings.warn(
📅 2025-08-03 16:51:53 | 📌 INFO | ✏️📄 Form input values:
{
    "MedInc": 2.0,
    "HouseAge": 2.0,
    "AveRooms": 1.0,
    "AveBedrms": 2.0,
    "Population": 2.0,
    "AveOccup": 1.0,
    "Latitude": 12.0,
    "Longitude": 21.0
}
📅 2025-08-03 16:51:53 | 📌 INFO | ✏️💰 Prediction Output:
{
    "predicted_price": 2.97
}
📅 2025-08-03 16:51:53 | 📌 INFO | ✏️✅ Completed `predict_from_form` in ⏱0.021s
📅 2025-08-03 16:51:53 | 📌 INFO | ✏️📤 Prediction endpoint completed in ⏱0.021s
INFO:     ::1:60838 - "POST /form HTTP/1.1" 200 OK
```



California House Price Predictor

Median Income: 2

House Age: 2

Average Rooms: 1

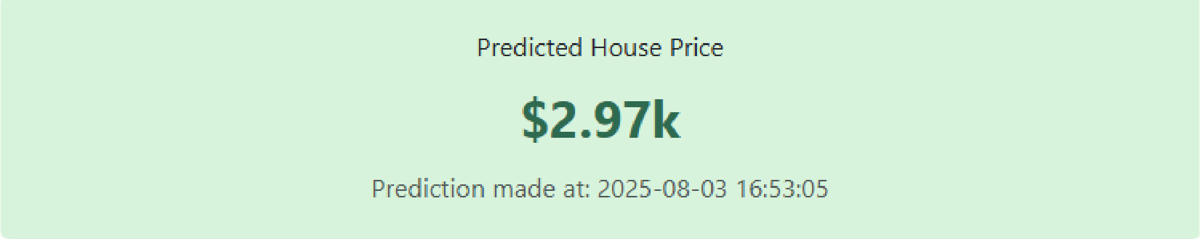
Average Bedrooms: 2

Population: 2

Average Occupancy: 1

Latitude: 12

Longitude: 21

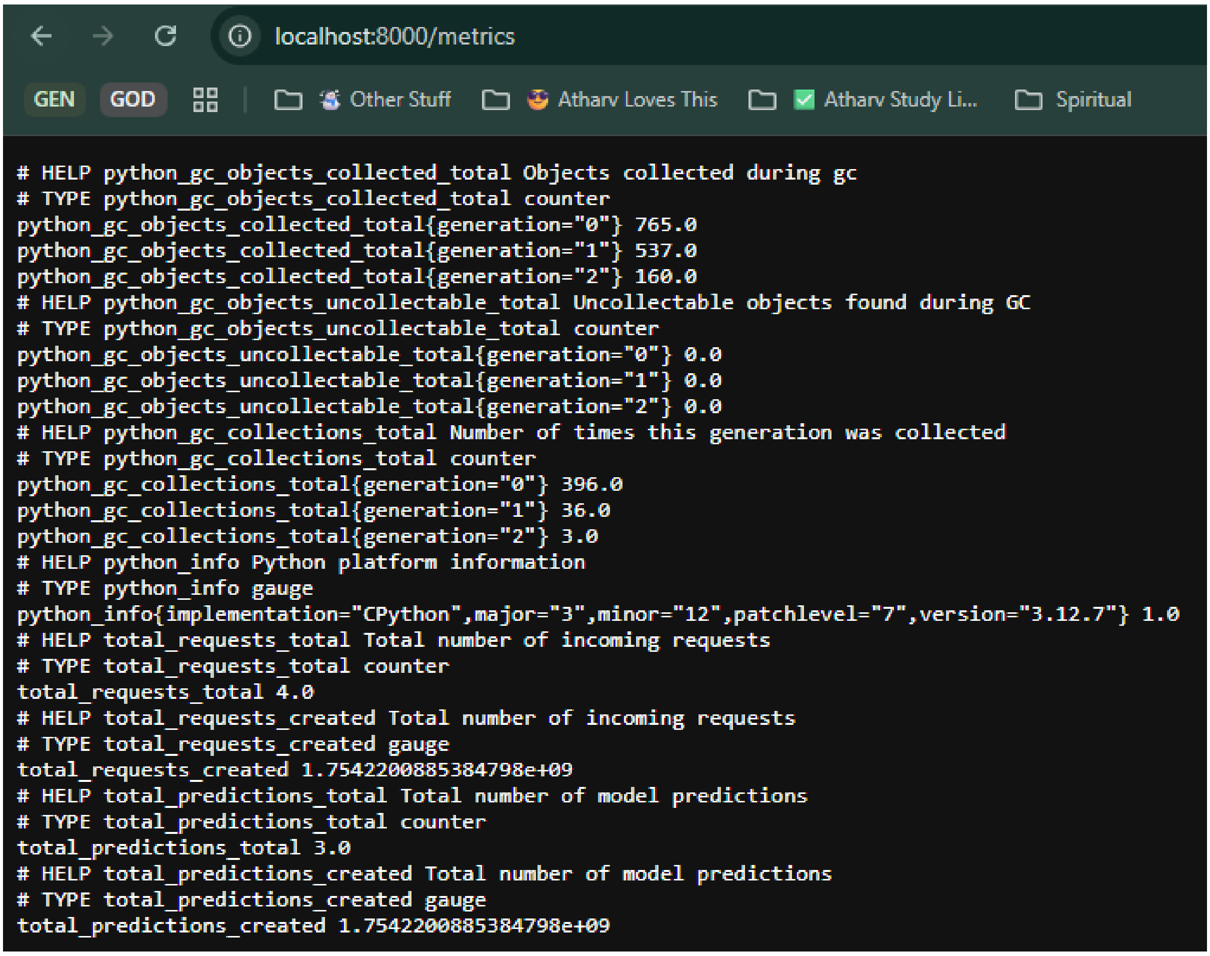


Predicted House Price

$2.97k

Prediction made at: 2025-08-03 16:53:05

Metrics and Prometheus client Usage :-



```
# HELP python_gc_objects_collected_total Objects collected during gc
# TYPE python_gc_objects_collected_total counter
python_gc_objects_collected_total{generation="0"} 765.0
python_gc_objects_collected_total{generation="1"} 537.0
python_gc_objects_collected_total{generation="2"} 160.0
# HELP python_gc_objects_uncollectable_total Uncollectable objects found during GC
# TYPE python_gc_objects_uncollectable_total counter
python_gc_objects_uncollectable_total{generation="0"} 0.0
python_gc_objects_uncollectable_total{generation="1"} 0.0
python_gc_objects_uncollectable_total{generation="2"} 0.0
# HELP python_gc_collections_total Number of times this generation was collected
# TYPE python_gc_collections_total counter
python_gc_collections_total{generation="0"} 396.0
python_gc_collections_total{generation="1"} 36.0
python_gc_collections_total{generation="2"} 3.0
# HELP python_info Python platform information
# TYPE python_info gauge
python_info{implementation="CPython",major="3",minor="12",patchlevel="7",version="3.12.7"} 1.0
# HELP total_requests_total Total number of incoming requests
# TYPE total_requests_total counter
total_requests_total 4.0
# HELP total_requests_created Total number of incoming requests
# TYPE total_requests_created gauge
total_requests_created 1.7542200885384798e+09
# HELP total_predictions_total Total number of model predictions
# TYPE total_predictions_total counter
total_predictions_total 3.0
# HELP total_predictions_created Total number of model predictions
# TYPE total_predictions_created gauge
total_predictions_created 1.7542200885384798e+09
```