California Housing Price Prediction Model - Documentation

Overview

This project uses machine learning to predict California house prices based on various features such as location, income, number of rooms, and proximity to the ocean.

The workflow consists of two main stages:

1. Training - building and saving a predictive model.

2. Inference - loading the model and using it to predict new house prices.

How the Script Works

1. Imports

- Uses libraries like pandas, numpy, and sklearn for data handling and modeling.

- Uses joblib to save and load models efficiently.

2. Build Pipeline

- The build_pipeline() function prepares data before training.

- It handles missing values, scales numbers, and converts categorical text data into numeric form.

3. Training Phase

- The dataset is loaded from housing.csv.

- A new column income_cat is created to perform stratified sampling.

- Data is split into training (80 percent) and testing (20 percent) sets.

- Features (inputs) and labels (outputs) are separated.

- A Random Forest model is trained using the cleaned data.

- The trained model and preprocessing pipeline are saved using joblib (model.pkl, pipeline.pkl).

4. Inference Phase

- If the model already exists, it is loaded from the saved files.

- The saved pipeline transforms new data (from input.csv).

- The model predicts house prices.

- The predictions are added to the data and saved as output.csv.

Files Created

- housing.csv -> Original dataset

- input.csv -> Cleaned training data used for prediction

- model.pkl -> Trained Random Forest model

- pipeline.pkl -> Preprocessing pipeline

- output.csv -> Final file with predictions

Key Concepts

- Stratified Sampling ensures fair representation of income categories.

- Pipelines automate data cleaning and transformation.

- Random Forest is used for robust, accurate regression predictions.

- joblib enables saving and reusing trained models.

Inference Summary

- If MODEL_FILE does not exist -> Train model.

- If MODEL_FILE exists -> Load model and make predictions.

Recommendations

- Use new, unseen data in input.csv for real predictions.

- Combine model and pipeline into one single Pipeline for simplicity.

- Use cross-validation or GridSearchCV to further improve accuracy.

Author: Satyam Gajjar