

<b>EXP NO: 9</b>	<b>MINI PROJECT - HOUSE PRICE PREDICTION API USING FLASK AND DOCKER</b>
------------------	---

**AIM:**

To develop and deploy a **machine learning–based House Price Prediction System** that can estimate housing prices based on key input features. The project seeks to integrate a trained regression model with a **Flask REST API** to enable real-time predictions through web requests and to **containerize the entire application using Docker** for seamless deployment, scalability, and environment independence.

**ALGORITHM:**

1. Data Collection: Load the California Housing Dataset from sklearn.datasets.
2. Data Preprocessing:
3. Handle missing values.
4. Convert date columns to datetime format.
5. Sort data chronologically.
6. Initialize a Linear Regression model from the scikit-learn library.
7. Save the trained model using joblib.dump() as model.pkl.
8. Train-Test Split: Divide the data into training and testing sets (e.g., 80%-20%).
9. Model Training: Fit a Linear Regression model using the training data.
10. Create a Dockerfile specifying the base image, dependencies, and commands to run the Flask app.
11. Visualization: Plot actual vs predicted stock prices to observe forecasting accuracy.

**CODE:**

```
!pip install flask scikit-learn pandas numpy joblib pyngrok
```

```
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import joblib
```

```
# Load dataset
data = fetch_california_housing(as_frame=True)
df = data.frame
```

```
X = df.drop('MedHouseVal', axis=1)
y = df['MedHouseVal']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
model = LinearRegression()
```

```

model.fit(X_train, y_train)

# Save model
joblib.dump(model, 'model.pkl')
print("✅ Model trained and saved as model.pkl")

from flask import Flask, request, jsonify
import joblib
import numpy as np
from pyngrok import ngrok

app = Flask(__name__)
model = joblib.load('model.pkl')

@app.route('/')
def home():
    return "🏠 Welcome to the House Price Prediction API (via Colab)"

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    try:
        features = np.array([
            data['MedInc'], data['HouseAge'], data['AveRooms'],
            data['AveBedrms'], data['Population'], data['AveOccup'],
            data['Latitude'], data['Longitude']
        ])
        prediction = model.predict(features)[0]
        return jsonify({'predicted_price': round(prediction, 2)})
    except Exception as e:
        return jsonify({'error': str(e)})

# Expose Colab's port 5000 to the internet via ngrok
public_url = ngrok.connect(5000)
print(f"Public URL: {public_url}")
app.run(port=5000)

import requests

url = "http://xxxxx.ngrok.io/predict"
payload = {
    "MedInc": 8.3, "HouseAge": 41.0, "AveRooms": 6.98, "AveBedrms": 1.02,
    "Population": 322.0, "AveOccup": 2.55, "Latitude": 37.88, "Longitude": -122.23
}
response = requests.post(url, json=payload)
print(response.json())

```

**OUTPUT:**

Public URL: <http://xxxxx.ngrok.io>

```
{'predicted_price': 2.33}
```

**RESULT:**

The Linear Regression model was successfully trained and used to predict stock prices. The predicted values closely followed the actual prices, showing that the model effectively captured the overall trend of the stock data.