A

PROJECT REPORT

On

“House Price Prediction Web App”

Submitted by,

Sahil Patil

Vipul Patil

Mansi Prajapati

Atharva Rane

Under the Guidance of

“Prof. U. G. Darekar”

Department of Information Technology
Datta Meghe College of Engineering,
Sector-3, Airoli, Navi Mumbai – 400 708, (M.S.), INDIA

2024-25

Datta Meghe College of Engineering

(AICTE & Govt. of Maharashtra Recognized, Affiliated to University of Mumbai)

Department of Information Technology

CERTIFICATE

This is to certify that Mr. **Sahil Patil, Div-B, Roll No.28/** Mr. **Vipul Patil, Div-B, Roll No.30/** Miss **Mansi Prajapati, Div-B, Roll No.33/** Mr. **Atharva Rane, Div-B, Roll No.34** of TE, of the Information Technology Department has completed the Project work entitled **“House Price Prediction Web App”** working under my guidance and supervision within the institute.

1. Abstract

This project presents a House Price Prediction Web Application that integrates machine learning with web development. The application is built using Python, scikit-learn, and Flask and utilizes the California Housing dataset to train a predictive model. The model is implemented using Linear Regression and is integrated into a Flask-based web interface to allow users to input housing-related features and receive price predictions. This project demonstrates the intersection of data science and web development, providing a foundation for further enhancements such as advanced models, cloud deployment, and enhanced UI/UX features.

2. Contents

3. Chapter-1

3.1 Introduction

House price prediction is an essential tool for real estate agencies, buyers, and sellers. This project develops a web application that predicts house prices based on key features such as income levels, house age, and number of rooms, population density, and geographical location. The predictive model is built using Linear Regression and is deployed using Flask, allowing users to make predictions through an interactive web interface.

3.2 Problem Definition

The real estate market is highly dynamic, influenced by factors such as location, amenities, market trends, and economic conditions. Traditional methods of house price estimation are often inaccurate, time-consuming, and lack real-time updates. Buyers, sellers, and investors struggle to make informed decisions due to fluctuating property prices and the absence of an efficient, data-driven tool.

This project aims to develop a **machine learning-powered web application** that provides accurate, real-time house price predictions based on historical data, live market trends, and user inputs. By integrating **advanced ML models and real-time data sources**, the system will offer a user-friendly platform for price estimation, comparative analysis, and trend visualization.

Key Challenges:

- **Inconsistent data:** House prices vary based on multiple unpredictable factors.
- **Lack of real-time updates:** Traditional models fail to adapt to market fluctuations.
- **User accessibility:** Existing platforms are complex and not user-friendly.

4. Objectives

The primary objective of this project is to develop a machine learning-based web application that accurately predicts house prices based on various factors. The specific goals include:

1. Accurate Price Prediction – Develop a model using the California Housing dataset to predict house prices based on factors like income, house age, and location.
2. User-Friendly Web Application – Create a Flask-based web interface where users can input property details and get instant predictions.
3. Model Performance Optimization – Train and evaluate a Linear Regression model to ensure reliable and efficient predictions.
4. Scalability & Future Enhancements – Design the system to allow integration of advanced models (e.g., Random Forest, XGBoost) for improved accuracy.

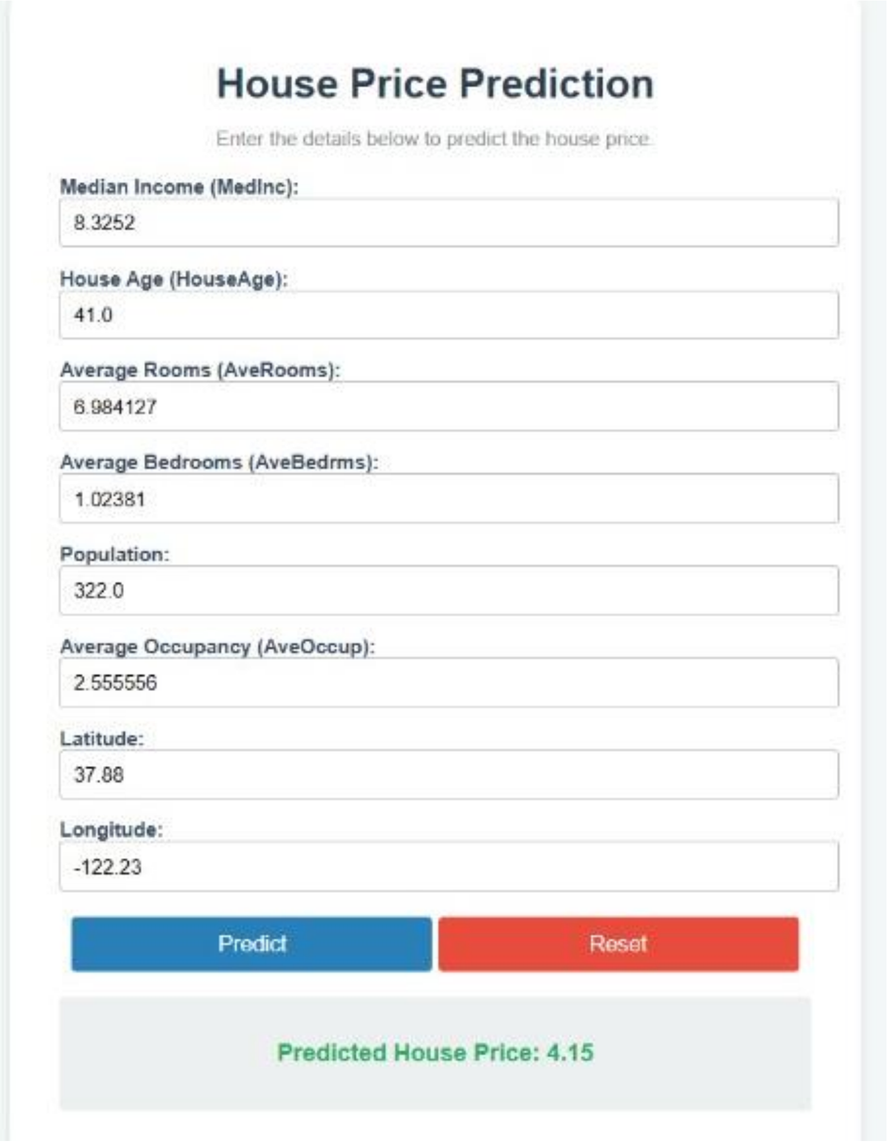
5. Scope

The project focuses on:

- Developing a regression-based model using the California Housing dataset.
- Providing real-time predictions via a web-based interface.
- Enabling quick price estimation for homebuyers, sellers, and real estate agents.
- Laying the foundation for future enhancements, such as integrating geospatial data, additional economic indicators, or deep learning models.

6. Implementation

6.1 Implementation Snapshots



House Price Prediction

Enter the details below to predict the house price.

Median Income (MedInc): 8.3252

House Age (HouseAge): 41.0

Average Rooms (AveRooms): 6.984127

Average Bedrooms (AveBedrms): 1.02381

Population: 322.0

Average Occupancy (AveOccup): 2.555556

Latitude: 37.88

Longitude: -122.23

Predicted House Price: 4.15

6.2 Code Implementation

Model Training Code (train_model.py)

```
import pandas as pd
from sklearn.datasets import fetch_california_housing
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import pickle

# Load dataset
data = fetch_california_housing()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['MedHouseVal'] = data.target  # Target variable

# Define features and target
X = df.drop('MedHouseVal', axis=1)
y = df['MedHouseVal']

# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Train Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Print model performance
print("Training R^2:", model.score(X_train, y_train))
print("Test R^2:", model.score(X_test, y_test))

# Save model
with open("model.pkl", "wb") as f:
    pickle.dump(model, f)
```

Flask Application Code (app.py)

```
from flask import Flask, render_template, request
import pickle
import numpy as np

app = Flask(__name__)

# Load the trained model
with open("model.pkl", "rb") as f:
    model = pickle.load(f)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Retrieve input data from the form (the dataset has 8 features)
        features = ["MedInc", "HouseAge", "AveRooms", "AveBedrms", "Population",
"AveOccup", "Latitude", "Longitude"]
        input_data = [float(request.form.get(feature)) for feature in features]
    except Exception as e:
        return "Invalid input: " + str(e)

    # Prepare data for prediction
    final_features = np.array(input_data).reshape(1, -1)
    prediction = model.predict(final_features)
    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text=f"Predicted House Price:
{output}")

if __name__ == "__main__":
    app.run(debug=True)
```

HTML Template (templates/index.html)

```html
<!DOCTYPE html>
<html>
<head>
    <title>House Price Prediction</title>
</head>
<body>
    <h2>House Price Prediction</h2>
    <form action="/predict" method="post">
        <label>Median Income (MedInc):</label>
        <input type="text" name="MedInc" required><br>
        <label>House Age (HouseAge):</label>
        <input type="text" name="HouseAge" required><br>
        <label>Average Rooms (AveRooms):</label>
        <input type="text" name="AveRooms" required><br>
        <label>Average Bedrooms (AveBedrms):</label>
        <input type="text" name="AveBedrms" required><br>
        <label>Population:</label>
        <input type="text" name="Population" required><br>
        <label>Average Occupancy (AveOccup):</label>
        <input type="text" name="AveOccup" required><br>
        <label>Latitude:</label>
        <input type="text" name="Latitude" required><br>
        <label>Longitude:</label>
        <input type="text" name="Longitude" required><br>
        <br>
        <input type="submit" value="Predict">
    </form>
    {% if prediction_text %}
        <h3>{{ prediction_text }}</h3>
    {% endif %}
</body>
</html>
```

7. CONCLUSION

The House Price Prediction Web App successfully integrates machine learning with web development to provide real-time house price estimations using the California Housing dataset and a Linear Regression model. The Flask-based interface ensures user-friendly interactions, allowing individuals to input key housing details and receive instant predictions. While the model offers reliable estimates, future improvements such as advanced algorithms (e.g., Random Forest, XGBoost), additional real estate factors (e.g., crime rates, infrastructure, economic indicators), and cloud deployment could enhance accuracy and scalability. This project lays a solid foundation for real estate analytics, demonstrating how data-driven insights can aid property buyers, sellers, and investors in making informed decisions.