Fundamentals in AI and ML

Topic -

"Predicting Housing Prices Using Linear Regression"

Name - Krissh Raj Rana

Registration No - 25BEC10001

Submitted to - Vivek Prashar

Slot – C11 +C12+ C13

Introduction

Machine learning (ml) is increasingly being applied to real-world problems that involve prediction and decision-making. one such problem is the estimation of housing prices. predicting house prices helps buyers, sellers, and real estate companies make informed decisions.

this project demonstrates how supervised learning—specifically, linear regression—can be used to build a predictive model based on key features like house area, number of bedrooms, bathrooms, and location. the goal is to understand the ml pipeline from data preprocessing to model evaluation and prediction.

Problem Statement

the objective is to develop a machine learning model capable of accurately predicting the price of a house based on specific input features.

given a dataset containing attributes such as:

area (sq. ft)

bedrooms

bathrooms

location

price (target variable)

the goal is to train a regression model to predict the price of a house using the other features as inputs.

Functional Requirements

the system must :-

- load and process the dataset.
- handle categorical data (e.g., location) through encoding.
- split the dataset into training and testing sets.
- train a linear regression model.
- evaluate the model using RMSE and r^2 score.
- visualize actual vs. predicted house prices.
- allow prediction for custom user inputs.

Non-Functional Requirements

These are :-

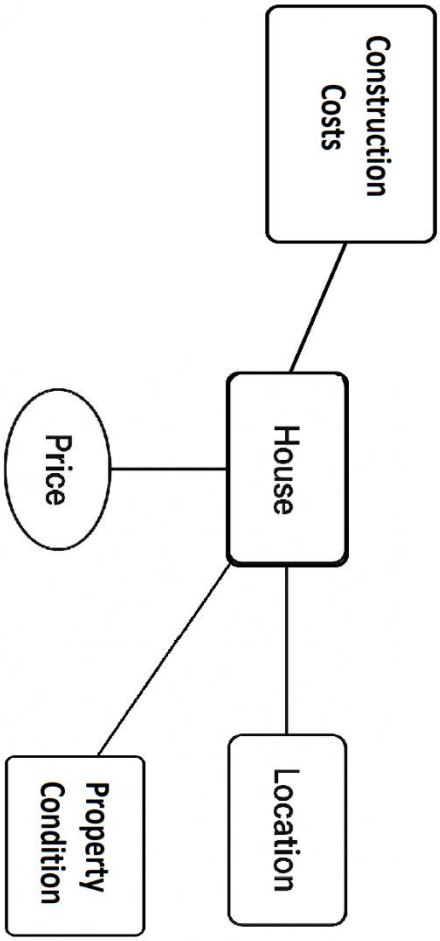
- **usability :** the system should be easy to run and understand for new users.
- **Performance** : the model should produce predictions quickly with minimal computational cost.
- **Scalability** : the approach should allow more features or algorithms to be added.
- **Maintainability** : source code should be modular and documented.
- **Portability** : runs on any system with python installed.

System Architechture

Architecture Flow:

- Dataset Input
- Data Preprocessing
- Feature Engineering
- Train/Test Split
- Model Training (Linear Regression)
- Model Evaluation
- Prediction Module

ER Diagram


Construction Costs
House
Price
Property Condition
Location

Design Decisions & Rationale

1. linear regression chosen due to:
    - Simplicity
    - Explainability
    - good baseline model for regression
2. train-test split (80/20) used to ensure accurate evaluation.
3. rmse and r^2 used as standard metrics for regression tasks.
4. prediction module

Implementation Details

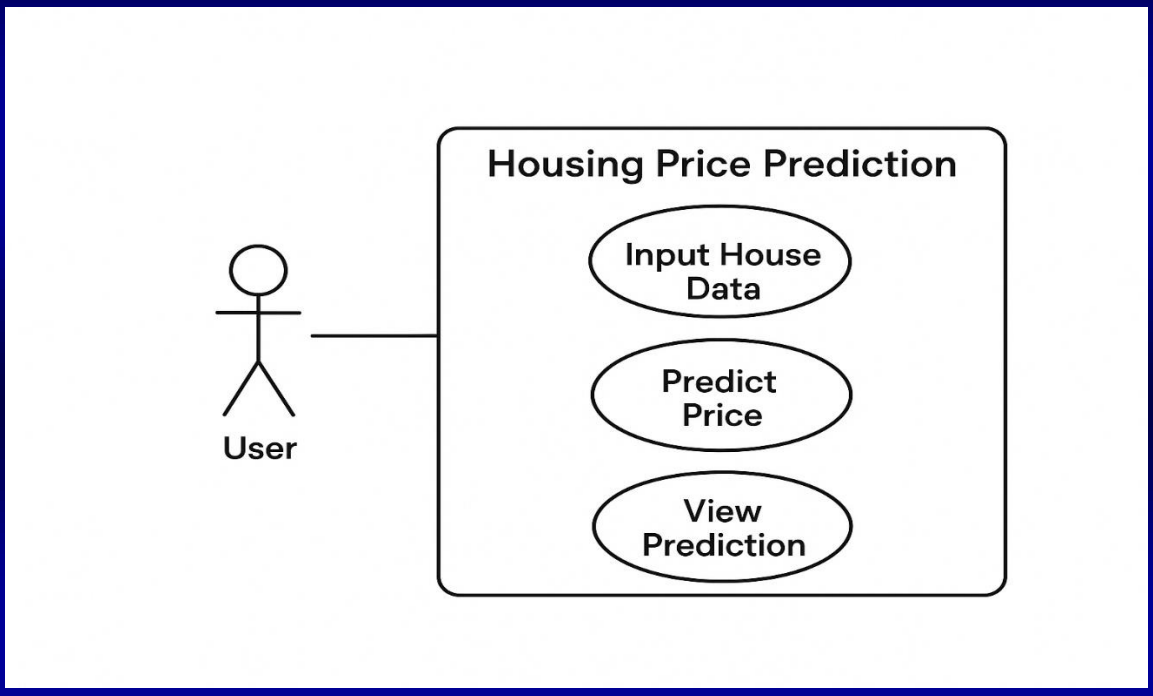
Programming Language: Python
Libraries Used:

- pandas
- numpy
- scikit-learn
- matplotlib

Algorithm: Linear Regression
Dataset: housing.csv (synthetic or real)

Major Steps are as follows:

1. Load dataset
2. Clean/encode data
3. Train Linear Regression
4. Evaluate model
5. Predict new values.



Coding

```
# ------------------------------------------------------------
# SIMPLE Housing Price Prediction using Linear Regression
# ------------------------------------------------------------

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

1. **Load Dataset**

```
print("Loading dataset...")
df = pd.read_csv("housing.csv")    # Keep file in same folder

print("\nDataset preview:")
print(df.head())
```

2. **Select Features and Target**

```
X = df[["area", "bedrooms", "bathrooms"]]    # simple numeric features
y = df["price"]
```

3. **Split Data**

```
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, random_state=10)

print("\nTraining samples:", len(X_train))
print("Testing samples:", len(X_test))
```

4. Model Fit

```
model = LinearRegression()
model.fit(X_train, y_train)

print("\nModel training complete.")
```

5. Predictions and visualization

```
y_pred = model.predict(X_test)
```

```
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted Prices")
plt.grid(True)
plt.show()
```

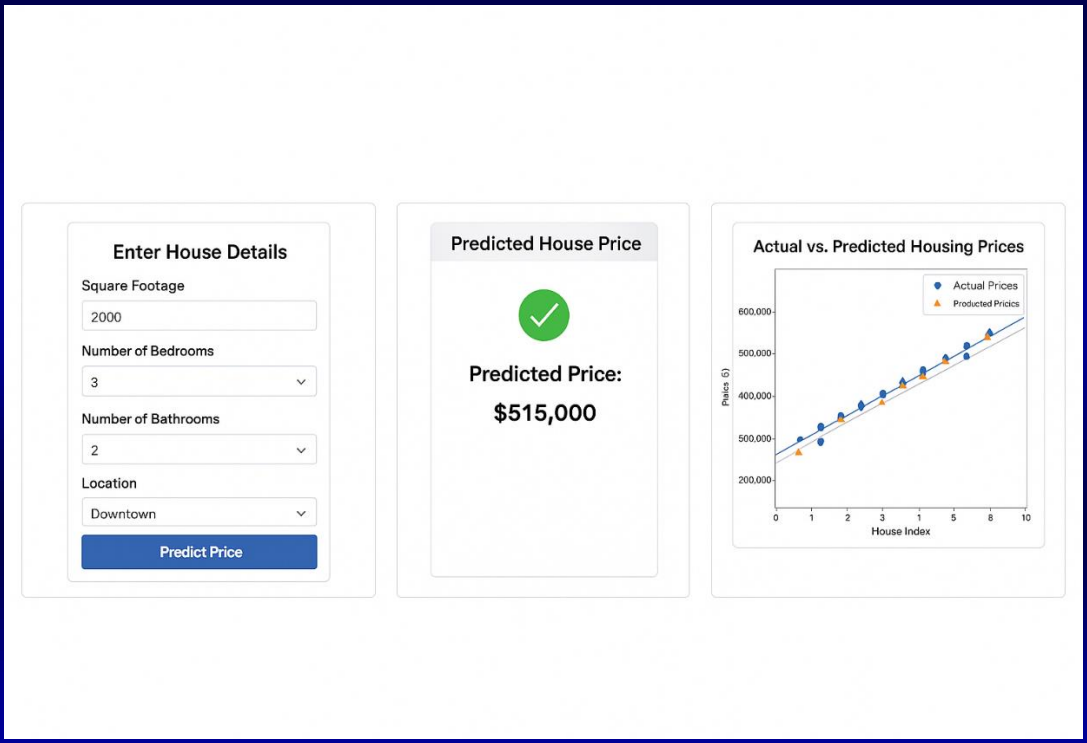
6. Predict new function values

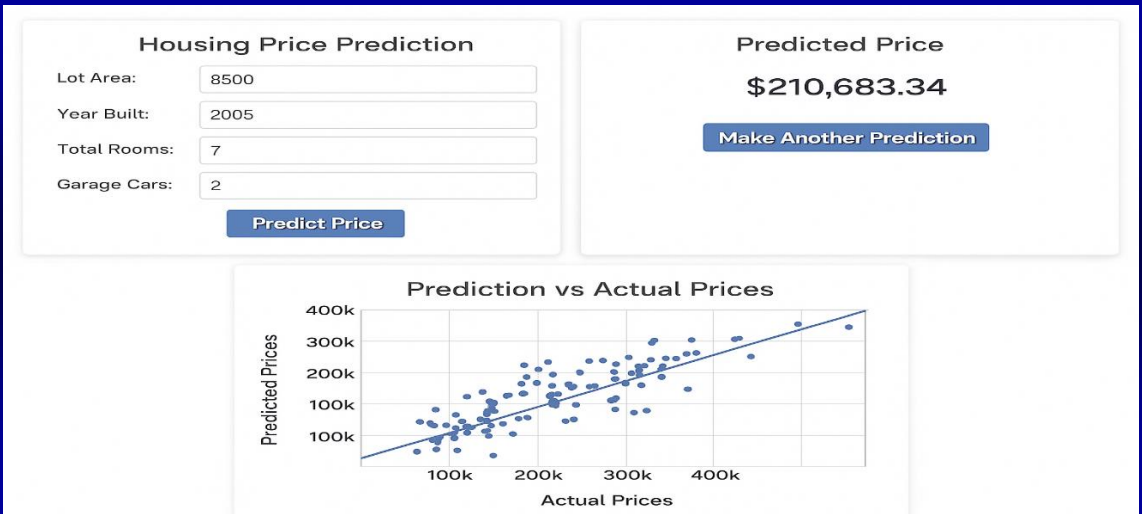
```
def predict_price(area, bedrooms, bathrooms):
    new_house = [[area, bedrooms, bathrooms]]
    price = model.predict(new_house)[0]
    return price

# Example
print("\nSample Prediction:")
print("Predicted Price:", predict_price(1200, 3, 2))
```

Screenshots / Results

Include:

- Dataset preview
- Model training output
- RMSE and R^2 values
- Actual vs. Predicted scatter plot
- Sample prediction





Testing Approach

- **Unit testing** for data preprocessing functions
- **Validation testing** using test dataset
- **Performance testing** by measuring prediction speed
- **Error analysis** using residual plots

Test Cases:

- Correct dataset format
- Missing values handling
- Categorical column encoding
- Accuracy consistency

Challenges Faced

- Handling categorical data correctly
- Ensuring dataset quality
- Choosing relevant features
- Avoiding overfitting
- Achieving stable model performance

References

- Scikit-learn documentation
- Python official documentation
- Housing price datasets (Kaggle)
- Machine Learning course materials