Predicting House Prices using Linear Regression

Sooraj Veer R
2511AI18

1. Introduction

This project aims to predict house prices based on three key features: Size (sqft), Number of Bedrooms, and Age (years). A Linear Regression model is trained and evaluated on a synthetic dataset. Visualization techniques, including 2D scatter plots and a 3D regression plane, are used to illustrate model performance.

2. Dataset

A synthetic dataset of 50 samples was generated. The target variable, Price (in lakhs), was computed using a weighted combination of the features with added noise. Features include Size (sqft), Bedrooms, and Age (years).

3. Methodology

1. Data Preparation: The dataset was split into features (X) and target (y).
2. Train-Test Split: 80% of the data was used for training and 20% for testing.
3. Model Training: A Linear Regression model was trained using scikit-learn.
4. Evaluation: Model accuracy was measured using Mean Squared Error (MSE) and R^2 Score.
5. Visualization: Scatter plots and a 3D regression plane were generated to compare actual vs predicted values.

4. Results

The Linear Regression model successfully captured the relationship between house features and price. Coefficients were consistent with expectations:
- Larger house size increases price.
- More bedrooms increase price.
- Higher age reduces price.

Evaluation metrics indicated a good model fit, with low MSE and a high R^2 score.

```
Predicted price for house [[2500, 4, 10]] = 160.77 lakhs
```

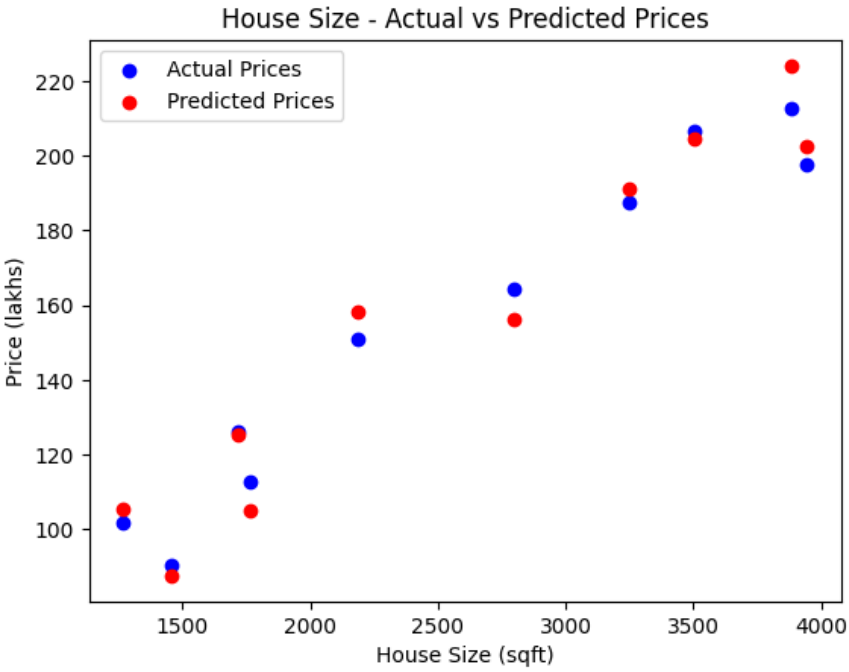
House Price Prediction

```
Sample of dataset:
   Size (sqft)  Bedrooms  Age (years)  Price (lakhs)
0         3674         1           15         184.20
1         1360         3           13          84.50
2         1794         5            1         146.20
3         1630         3           25          88.00
4         1595         5            7         133.25

Model Coefficients:
Size (sqft): 0.05
Bedrooms: 10.63
Age (years): -1.53
Intercept: 5.6990280368450215

Model Evaluation:
Mean Squared Error: 37.86050393043571
R² Score: 0.9796461805295054
```

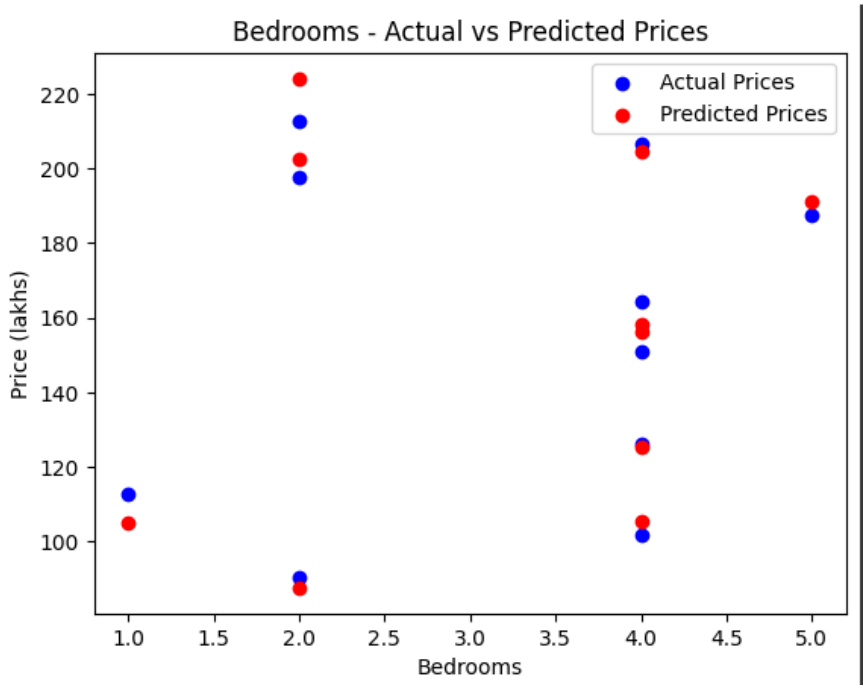
Model Parameters

5. Visualizations

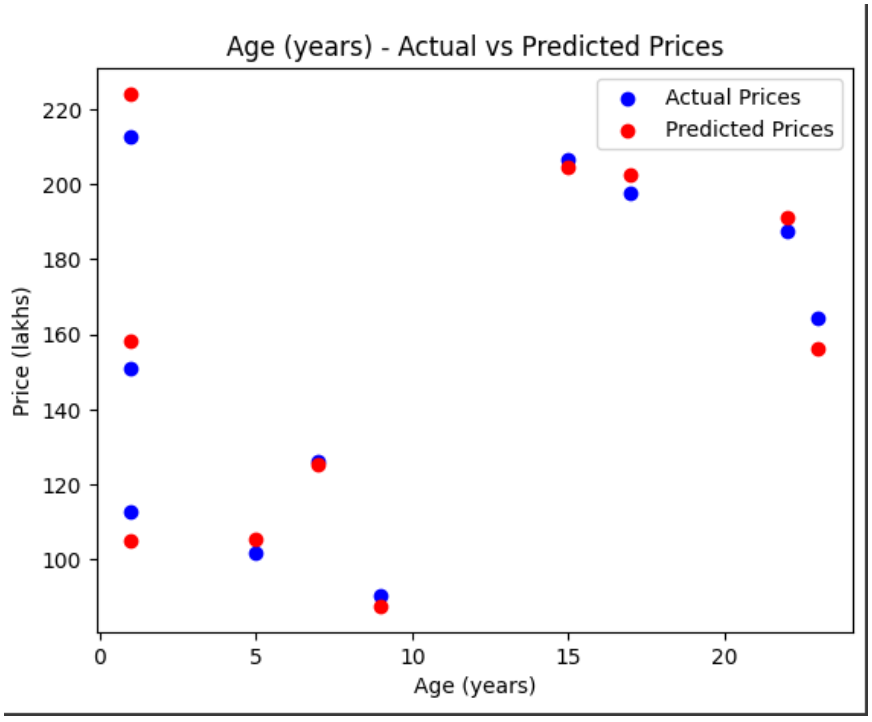
The following figures illustrate the results:



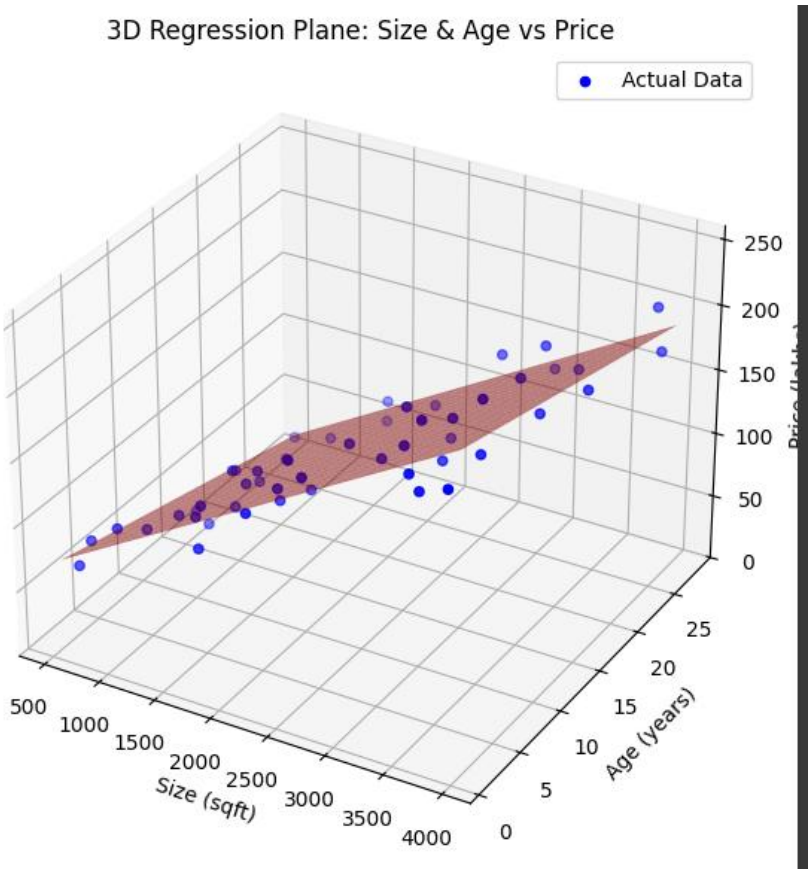
Size vs Price (Actual vs Predicted)



Bedrooms vs Price (Actual vs Predicted)



Age vs Price (Actual vs Predicted)



3D Regression Plane (Size & Age vs Price)

6. Conclusion

The project demonstrated the effectiveness of Linear Regression in predicting house prices using basic features. The model results aligned with real-world intuition. Future improvements could involve using larger real-world datasets, adding more features (e.g., location, amenities), and experimenting with advanced regression techniques like Ridge, Lasso, or Polynomial Regression.

7. Code

```
# Predicting the Price of a House based on 3 input features

# Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from mpl_toolkits.mplot3d import Axes3D
```

```
# Create the dataset
np.random.seed(42)

size = np.random.randint(500, 4000, 50)           # House size in
sqft
bedrooms = np.random.randint(1, 6, 50)            # Number of
bedrooms
age = np.random.randint(1, 30, 50)                # Age of house in
years

# Target (house price in lakhs)
price = (size * 0.05) + (bedrooms * 10) - (age * 1.5) +
np.random.randint(0, 20, 50)

# Create DataFrame
df = pd.DataFrame({
    'Size (sqft)': size,
    'Bedrooms': bedrooms,
    'Age (years)': age,
    'Price (lakhs)': price
})

print("Sample of dataset:")
print(df.head())

# Split features and target
X = df[['Size (sqft)', 'Bedrooms', 'Age (years)']]
y = df['Price (lakhs)']

# Train-test split (80% training, 20% testing)
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)

# Model coefficients
print("\nModel Coefficients:")
print(f"Size (sqft): {model.coef_[0]:.2f}")
print(f"Bedrooms: {model.coef_[1]:.2f}")
print(f"Age (years): {model.coef_[2]:.2f}")
print("Intercept:", model.intercept_)

# Predictions
y_pred = model.predict(X_test)

# Model evaluation
print("\nModel Evaluation:")
```

```
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("R² Score:", r2_score(y_test, y_pred))

#3D Visualization (Size, Age vs Price)
X_2d = df[['Size (sqft)', 'Age (years)']]
y_2d = df['Price (lakhs)']

# Fit model with only 2 features
model_2d = LinearRegression()
model_2d.fit(X_2d, y_2d)

# Create a meshgrid for Size and Age
size_range = np.linspace(X_2d['Size (sqft)'].min(), X_2d['Size
(sqft)'].max(), 20)
age_range = np.linspace(X_2d['Age (years)'].min(), X_2d['Age
(years)'].max(), 20)
size_grid, age_grid = np.meshgrid(size_range, age_range)

# Flatten and predict
grid_points = np.c_[size_grid.ravel(), age_grid.ravel()]
price_pred = model_2d.predict(grid_points)
price_grid = price_pred.reshape(size_grid.shape)

# Plot 3D scatter + regression plane
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Scatter actual points
ax.scatter(X_2d['Size (sqft)'], X_2d['Age (years)'], y_2d,
color='blue', label='Actual Data')

# Plot regression plane
ax.plot_surface(size_grid, age_grid, price_grid, color='red',
alpha=0.5)

# Labels
ax.set_xlabel("Size (sqft)")
ax.set_ylabel("Age (years)")
ax.set_zlabel("Price (lakhs)")
ax.set_title("3D Regression Plane: Size & Age vs Price")
plt.legend()
plt.show()


# Visualization (only Size vs Price)
plt.scatter(X_test['Size (sqft)'], y_test, color='blue',
label='Actual Prices')
plt.scatter(X_test['Size (sqft)'], y_pred, color='red',
label='Predicted Prices')
plt.xlabel("House Size (sqft)")
```

```
plt.ylabel("Price (lakhs)")
plt.title("House Size - Actual vs Predicted Prices")
plt.legend()
plt.show()

# Visualization (only Bedrooms vs Price)
plt.scatter(X_test['Bedrooms'], y_test, color='blue', label='Actual
Prices')
plt.scatter(X_test['Bedrooms'], y_pred, color='red',
label='Predicted Prices')
plt.xlabel("Bedrooms")
plt.ylabel("Price (lakhs)")
plt.title("Bedrooms - Actual vs Predicted Prices")
plt.legend()
plt.show()

# Visualization (only Bedrooms vs Price)
plt.scatter(X_test['Age (years)'], y_test, color='blue',
label='Actual Prices')
plt.scatter(X_test['Age (years)'], y_pred, color='red',
label='Predicted Prices')
plt.xlabel("Age (years)")
plt.ylabel("Price (lakhs)")
plt.title("Age (years) - Actual vs Predicted Prices")
plt.legend()
plt.show()

# Predict for a new house
new_house = np.array([[2500, 4, 10]])  # [Size, Bedrooms, Age]
predicted_price = model.predict(new_house)
print(f"\nPredicted price for house {new_house.tolist()} =
{predicted_price[0]:.2f} lakhs")
```