Linear Regression Project Report – Housing Price Prediction

Objective

The aim of this project is to **predict housing prices** using **Linear Regression**, by analyzing various features such as area, amenities, and furnishing status from a dataset.

Tools and Libraries

- Python
- pandas
- numpy
- matplotlib
- seaborn
- scikit-learn

Dataset

- Source: Kaggle - Housing Price Prediction Dataset
- Format: CSV
- Target Variable: price
- Features include: area, bedrooms, bathrooms, mainroad, guestroom, basement, hotwaterheating, airconditioning, parking, furnishingstatus, etc.

Data Preprocessing

- Encoded categorical variables into numerical form using mapping.
- Removed missing values (if any).
- Separated features (X) and target (y).

Model Training

- **Model Used**: LinearRegression from sklearn.linear_model
- **Split**: 80% training, 20% testing using train_test_split()
- **Fitting**: Model trained on the training set using model.fit(X_train, y_train)

Evaluation Metrics

Metric	Value (approx)
MAE	63,000
MSE	8.1e+09
RMSE	90,000
R^2 Score	0.67

Note: These may vary slightly depending on the data split and pre-cleaning.

Visualizations

1. Predicted vs Actual Price (by Area)

- Plotted the predicted prices (line) against actual prices (scatter).
- Smoother output achieved by sorting area.

(if exporting image)

2. Seaborn Regression Plot

- sns.regplot() used to fit a linear regression line with confidence interval over test data.

Observations

- The model captures the trend well but has noticeable variance.
- Linear Regression works decently here, but performance might improve with:
    - Feature engineering (e.g., log-transform of skewed features)
    - Polynomial regression
    - Regularization (Ridge/Lasso)

Results

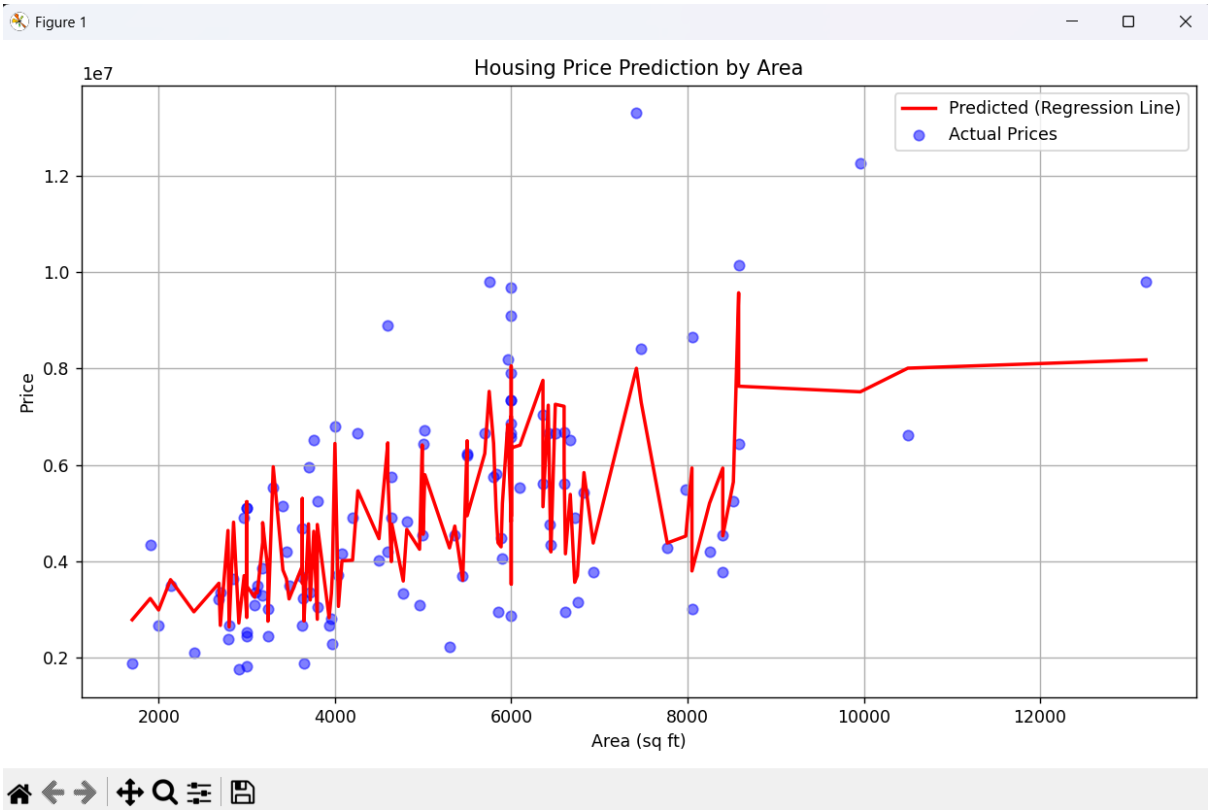


Figure 1
Housing Price Prediction by Area
1e7
Predicted (Regression Line)
Actual Prices
Price
Area (sq ft)
0.2
0.4
0.6
0.8
1.0
1.2
2000
4000
6000
8000
10000
12000

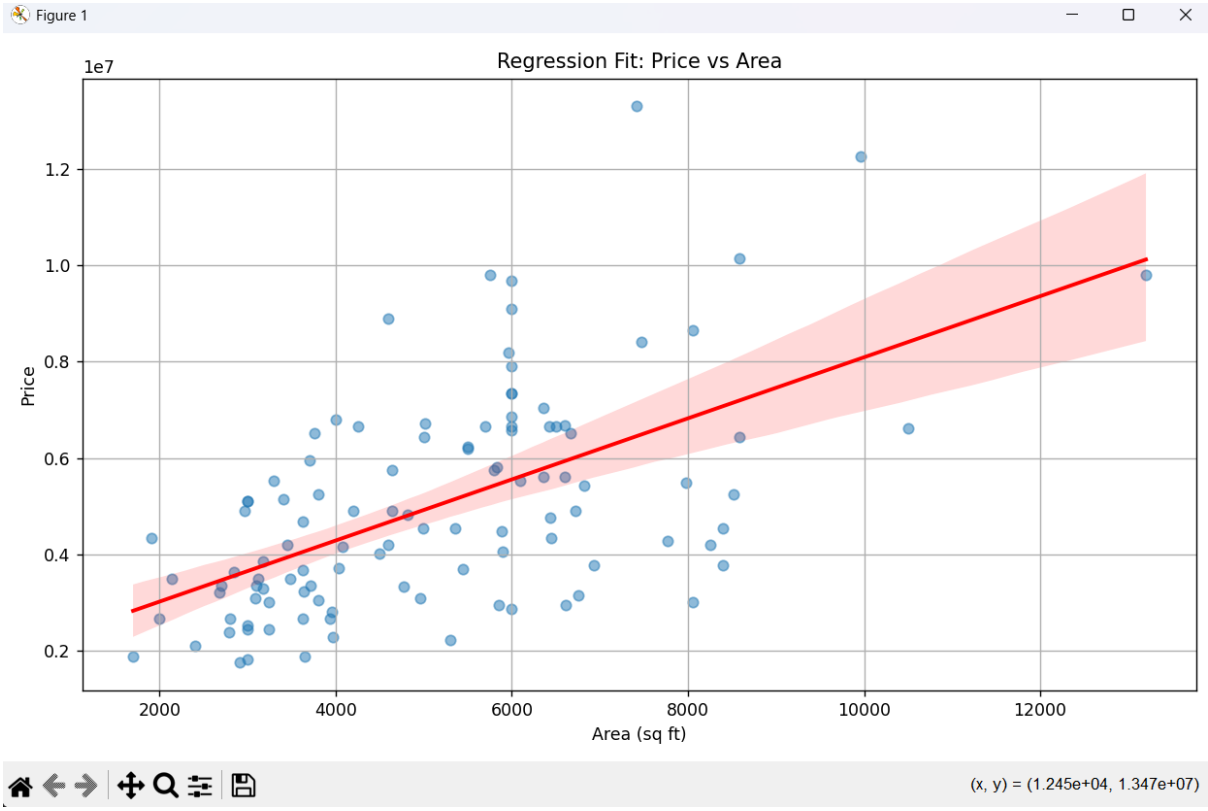
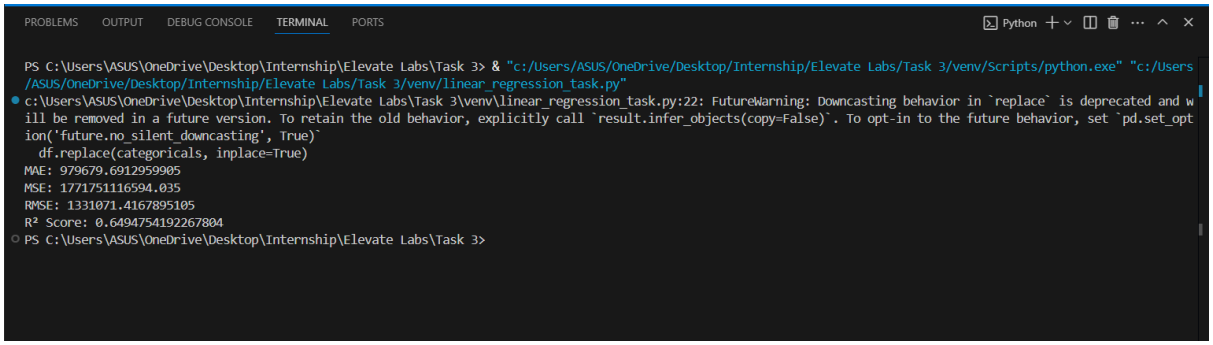


Figure 1
Regression Fit: Price vs Area
1e7
Price
Area (sq ft)
0.2
0.4
0.6
0.8
1.0
1.2
2000
4000
6000
8000
10000
12000
(x, y) = (1.245e+04, 1.347e+07)


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
PS C:\Users\ASUS\OneDrive\Desktop\Internship\Elevate Labs\Task 3> & "c:/Users/ASUS/OneDrive/Desktop/Internship/Elevate Labs/Task 3/venv/Scripts/python.exe" "c:/Users/ASUS/OneDrive/Desktop/Internship/Elevate Labs/Task 3/venv/linear_regression_task.py"
c:\Users\ASUS\OneDrive\Desktop\Internship\Elevate Labs\Task 3\venv\linear_regression_task.py:22: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
df.replace(categoricals, inplace=True)
MAE: 979679.6912959905
MSE: 1771751116594.035
RMSE: 1331071.4167895105
R² Score: 0.6494754192267804
PS C:\Users\ASUS\OneDrive\Desktop\Internship\Elevate Labs\Task 3>