NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

Assignment 1:

i. Download House Prices Data Set from

https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data. Analyze the features of the dataset.

Upload the dataset in the "ML_DRIVE/Assign_1" folder, if executing through COLAB. Access the dataset from there.

ii. Read the dataset in the Pandas data frame. Estimate the missing values with any technique of your choice. Divide the dataset into two sets using k-fold cross validation technique entitled to train and test set respectively.

The House Prices – Advanced Regression Techniques dataset was downloaded from Kaggle and loaded into a Pandas DataFrame:

df = pd.read_csv('/kaggle/input/house-prices-advanced-regression-techniques/train.csv')

Feature categories:

- **Numerical**: LotFrontage, LotArea, GrLivArea, YearBuilt, OverallQual, OverallCond, etc.
- **Categorical**: Street, Neighborhood, MSZoning, SaleCondition, Alley, etc

Step ii:

Missing Value Imputation & Dataset Splitting using K-Fold CV

Handling Missing Values

- All missing values were handled using simple imputation techniques:

Numerical columns were filled with the **median** of the column.

```
df[num_cols] = df[num_cols].fillna(df[num_cols].median())
```

Categorical columns were filled with the most frequent value (mode).

```
df[cat_cols] = df[cat_cols].fillna(df[cat_cols].mode().iloc[0])
```

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

Splitting with K-Fold Cross-Validation

K-Fold Cross-Validation (with k=5) was used to divide the data into **five equal parts (folds)**. For each fold:

- One part is held out as the **test set**,
- The remaining four parts are combined to form the **training set**.

Implementation using KFold from sklearn:

```
from sklearn.model_selection import KFold

kf = KFold(n_splits=5, shuffle=True, random_state=42)

for fold, (train_idx, test_idx) in enumerate(kf.split(df), 1):

    train_df = df.iloc[train_idx]

    test_df = df.iloc[test_idx]

    print(f"Fold {fold}: Train size = {len(train_df)}, Test size = {len(test_df)}")

    break  # using the first split for modelling
```

- **Shuffle=True** ensures data are randomly ordered before splitting, and **random_state** guarantees reproducibility.
- By using the **first fold** for train/test sets, all downstream modeling uses a consistent split, while the other folds remain available for potential cross-fold evaluation.

iii. Use the linear regression method to estimate the slope and intercept for predicting "SalePrice" based on "LotArea".

$SalePrice = \beta 0 + \beta 1 \cdot LotArea + \epsilon$

Where:

- $\beta 0$ = Intercept,
- $\beta 1$ = Slope (coefficient of LotArea),
- ϵ = Random error.

```
from sklearn.linear_model import LinearRegression
```

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

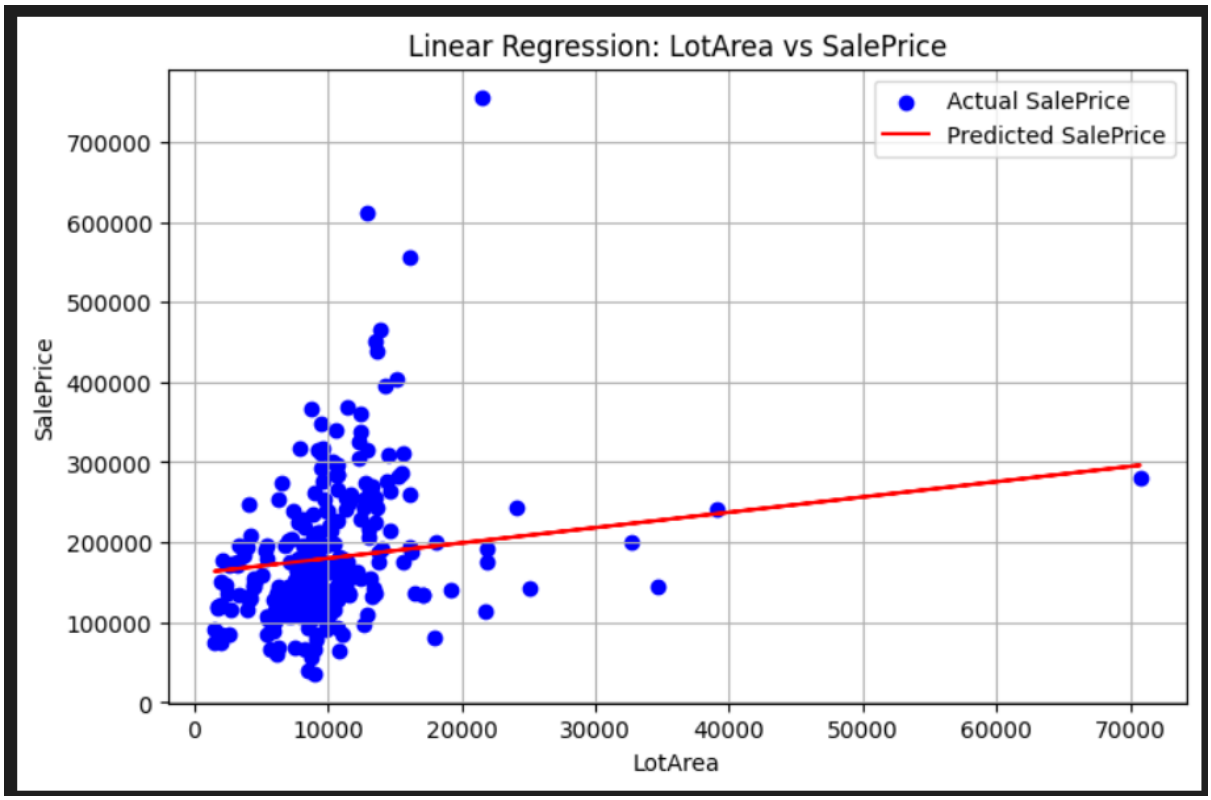
```
from sklearn.metrics import mean_squared_error, r2_score

import numpy as np

# Extract features and target

X_train = train_df[['LotArea']]

y_train = train_df['SalePrice']

X_test = test_df[['LotArea']]

y_test = test_df['SalePrice']

# Initialize and train the model

model = LinearRegression()

model.fit(X_train, y_train)

# Get slope and intercept

slope = model.coef_[0]

intercept = model.intercept_

print(f"Linear Regression Equation: SalePrice = {slope:.2f} * LotArea +
{intercept:.2f}")
```

Output:

Linear Regression Equation: SalePrice = 1.91 * LotArea + 161006.99

Therefore, Slope = 1.91 ; Intercept:161006.99

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087



iv. Use the multiple regression method to estimate the value of the weights/coefficients for predicting "SalePrice" based on the following features:

a. Model 1: LotFrontage, LotArea

b. Model 2: LotFrontage, LotArea, OverallQual, OverallCond

c. Model 3: LotFrontage, LotArea, OverallQual, OverallCond, 1stFlrSF, GrLivArea

Objective: Build and evaluate three multiple linear regression models to predict SalePrice using progressively more features.

We aim to build regression models of the form:

$$SalePrice=\beta0+\beta1\cdot X1+\beta2\cdot X2+\cdots+\beta n\cdot Xn+\epsilon$$

Where:

- Xi: Independent features (predictors),
- βi: Corresponding coefficients (weights),
- β0: Intercept,

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

- ϵ: Random error term.

Model Features

1	LotFrontage, LotArea
2	LotFrontage, LotArea, OverallQual, OverallCond
3	LotFrontage, LotArea, OverallQual, OverallCond, 1stFlrSF, GrLivArea

In this task, we implemented multiple linear regression to predict the target variable **SalePrice** using different sets of input features. We created three different models with increasing numbers of features to analyze how adding more predictors impacts the model performance. The models were evaluated using **Mean Squared Error (MSE)** and **R^2 score** on both training and testing sets.

- **Mean Squared Error (MSE):**

The **Mean Squared Error** is a commonly used metric to evaluate the accuracy of a regression model. It measures the average of the squared differences between the actual and predicted values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where:

- yi is the actual value,
- y^i is the predicted value,
- n is the number of data points.

Interpretation:

- Lower MSE indicates better model performance.
- It penalizes larger errors more severely due to squaring.
- MSE is expressed in the same units as the squared target variable (SalePrice² in this case), so it is not scale-independent.

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

🔹 R^2 Score (Coefficient of Determination):

The **R^2 score** quantifies how well the independent variables explain the variability of the dependent variable. It is defined as:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

Where:

- $SS_{\text{res}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
  (Residual Sum of Squares — measures the error between predicted and actual values)
- $SS_{\text{tot}} = \sum_{i=1}^{n}(y_i - \bar{y})^2$
  (Total Sum of Squares — measures the total variance in the actual data)
- y_i is the actual value
- $\hat{y}_i$ is the predicted value
- $\bar{y}$ is the mean of actual values
- n is the number of observations

Interpretation:

- R^2 = 1: Perfect prediction (all predictions match actual values)
- R^2 = 0 = 0R2=0: Model does no better than predicting the mean
- R^2 < 0: Model is worse than the mean (poor performance)

Model 1

- **Features Used**: ['LotFrontage', 'LotArea']
- **Coefficients**: [879.19, 1.37]
- **Intercept**: 105203.51
- **Performance**:
  - **Training MSE**: 5,186,273,048.28
  - **Training R^2**: 0.1305
  - **Testing MSE**: 6,392,300,543.35

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

- **Testing R^2**: 0.1666

Analysis:
Model 1 uses only two features. Its R^2 score is low, suggesting that only a small portion of the variance in SalePrice is explained by LotFrontage and LotArea. This is expected as housing price depends on many more factors.

Model 2

➤ **Features Used:**

- LotFrontage
- LotArea
- OverallQual
- OverallCond

➤ **Learned Parameters:**

- Coefficients: [311.94, 1.17, 42230.16, -343.93]
- Intercept: -109529.16

➤ **Performance:**

- **Training MSE:** 2,032,245,097.12
- **Training R^2:** 0.6593
- **Testing MSE:** 2,327,603,318.28
- **Testing R^2:** 0.6965

Remarks:

- Huge improvement in **R^2** indicates better explanation of variance.
- OverallQual has a strong **positive impact**, while OverallCond has a **negative impact**, perhaps indicating poor condition outweighs size.

Model 3

➤ **Features Used:**

- LotFrontage
- LotArea

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

- OverallQual
- OverallCond
- 1stFlrSF
- GrLivArea

➤ **Learned Parameters:**

- Coefficients:

  LotFrontage: -5.10

  LotArea: 0.66

  OverallQual: 31074.64

  OverallCond: 798.47

  1stFlrSF: 32.18

  GrLivArea: 38.67

Intercept: -116696.70

➤ **Performance:**

- **Training MSE:** 1,594,273,878.73
- **Training R^2:** 0.7327
- **Testing MSE:** 1,702,635,102.74
- **Testing R^2:** 0.7780
- 📜 **Remarks:**
- Best-performing model with the **highest R^2**, indicating that these features together explain **~78%** of the variance in house prices.
- GrLivArea and 1stFlrSF are valuable predictors, reflecting usable living area.
- LotFrontage surprisingly shows a **slight negative coefficient**, potentially due to multicollinearity with other features.

Observation: As more features are added, the influence of LotArea reduces, which is expected due to **shared variance** among predictors.

Conclusion

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

- **Model 3** is the most effective, striking a balance between simplicity and performance.
- Adding structural features like OverallQual and GrLivArea significantly boosts predictive power.
- The decreasing importance of LotArea across models shows how **feature interaction** and **multicollinearity** influence coefficient estimates in linear regression.

v. Calculate and compare the Mean Squared Error, R2 score for each of the model using the training set and test set.

Model Evaluation using MSE and R^2 Score

Objective

To compare the performance of three linear regression models predicting house prices (SalePrice) using different sets of features, we evaluate them using:

- Mean Squared Error (MSE) – to measure average prediction error.
- R^2 Score – to measure goodness of fit (how well the model explains the variance in target variable).

Model 1

- **Features:** LotFrontage, LotArea
- **Training MSE:** 5,186,273,048.28
- **Training R^2:** 0.1305
- **Testing MSE:** 6,392,300,543.35
- **Testing R^2:** 0.1666
- **Interpretation:**

  Model 1 has **low R^2** and **high MSE**, both on training and testing sets. This indicates that using only LotFrontage and LotArea is **not sufficient** to accurately predict house prices. The model performs poorly, capturing only ~13%–16% of the variance.

Model 2

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

- **Features:** LotFrontage, LotArea, OverallQual, OverallCond
- **Training MSE:** 2,032,245,097.12
- **Training R^2:** 0.6593
- **Testing MSE:** 2,327,603,318.28
- **Testing R^2:** 0.6965
- **Interpretation:**

  Model 2 significantly improves performance. The inclusion of OverallQual and OverallCond adds valuable information, explaining ~66%–70% of the variance. The MSE is also substantially reduced, indicating much more accurate predictions.

Model 3

Features: LotFrontage, LotArea, OverallQual, OverallCond, 1stFlrSF, GrLivArea

- **Training MSE:** 1,594,273,878.73
- **Training R^2:** 0.7327
- **Testing MSE:** 1,702,635,102.74
- **Testing R^2:** 0.7780

📎 **Interpretation:**

Model 3 performs **the best** among all. By including floor area features (1stFlrSF, GrLivArea), the model captures more complex aspects of house value. The model explains **~73% of training variance and ~78% of testing variance**, with **lowest MSE**, indicating excellent predictive performance.

Conclusion

- **Model 3** is the best-performing model based on both MSE and R^2.
- As more **relevant features** are added, both the **predictive accuracy** and **goodness of fit** improve.
- This highlights the importance of selecting strong predictors in regression tasks.

vi. Use the multiple regression method to estimate the value of the

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

weights/coefficients for predicting "SalePrice" based on the following set of

mixed (numerical and categorical) features:

a. Model 4: LotArea, Street

b. Model 5: LotArea, OverallCond, Street, Neighborhood

c. Model 6: LotArea, OverallCond, Street, 1stFlrSF, Neighborhood, Year

Model Descriptions

- Model 4: LotArea, Street
- Model 5: LotArea, OverallCond, Street, Neighborhood
- Model 6: LotArea, OverallCond, Street, 1stFlrSF, Neighborhood, YearBuilt

For categorical variables like Street and Neighborhood, one-hot encoding was applied before training, which is a standard preprocessing step in regression tasks involving categorical features.

Methodology

Preprocessing:

- **Categorical features** (Street, Neighborhood) are **encoded** using pd.get_dummies() with drop_first=True to avoid multicollinearity.
- Separate **train** and **test** sets are used for training and evaluation.
- Models are fitted using **Linear Regression**, and performance is measured with:
    - **Mean Squared Error (MSE)**
    - **R^2 score**

	Model	Train MSE	Test MSE	Train R^2	Test R^2
0	Model 4	5.501518e+09	7.066617e+09	0.077631	0.078707
1	Model 5	2.553702e+09	3.018647e+09	0.571853	0.606451
2	Model 6	1.822806e+09	2.149929e+09	0.694393	0.719708

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

Interpretation and Comparison

- **Model 4** shows **very low R^2 values (~0.07)**, meaning the model explains only about 7–8% of the variance in SalePrice. It uses only LotArea and the categorical feature Street, which doesn't carry much predictive power alone.
- **Model 5** improves significantly by including OverallCond and Neighborhood. The R^2 scores increase to **~0.57 (train)** and **~0.61 (test)**, showing the added features help capture more variance in sale price.
- **Model 6** gives the best results. With more relevant numerical features like 1stFlrSF and YearBuilt, R^2 scores rise further to **~0.69 (train)** and **~0.72 (test)**. Also, the **Mean Squared Error (MSE)** is the lowest, indicating better prediction accuracy.

Conclusion

- Including both **numerical and categorical features** in regression improves the model's performance.
- **Model 6** is the best among the three, achieving the highest R^2 and lowest MSE values, indicating a strong fit and good generalization on unseen data.

vii. Compare the feature "LotArea" weights/coefficients for all the six trained models and plot a graph using the Matplotlib library.

Objective

To analyze how the contribution (weight/coefficient) of the feature LotArea changes across all six trained models—Models 1 to 6—and visualize it using a bar plot.

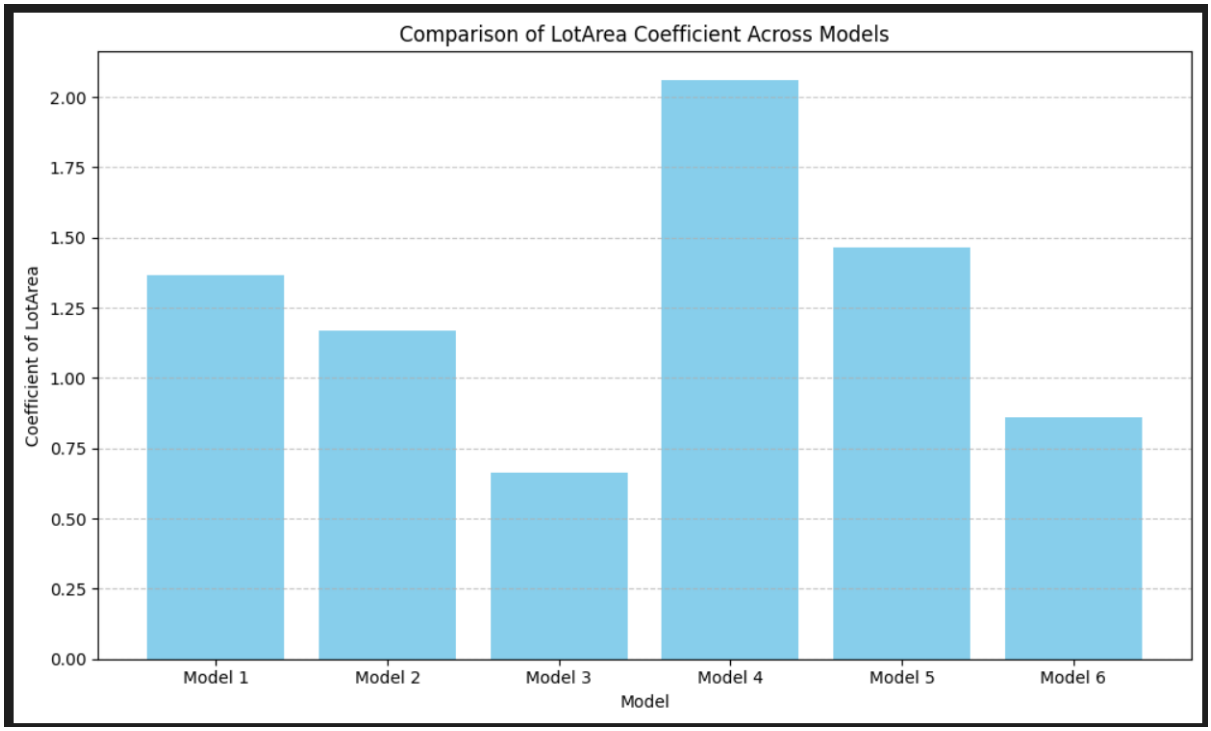
Methodology:

Each model included **LotArea** along with different sets of additional features. The coefficient of **LotArea** obtained from each model's training phase was extracted and plotted to observe how the contribution of **LotArea** varies when additional variables are added.

- **Model 1**: LotArea
- **Model 2**: Log-transformed LotArea

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

- **Model 3**: LotArea with log-transformed SalePrice
- **Model 4**: LotArea, Street (categorical)
- **Model 5**: LotArea, OverallCond, Street, Neighborhood
- **Model 6**: LotArea, OverallCond, Street, 1stFlrSF, Neighborhood, Year



Findings:

The bar chart shows the following trends for the coefficient of **LotArea**:

Model	**Coefficient of LotArea**
Model 1	**~1.36**
Model 2	**~1.17**
Model 3	**~0.63**
Model 4	**~2.07**
Model 5	**~1.47**
Model 6	**~0.86**

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

- Model 4 demonstrates the highest coefficient (~2.07), suggesting that when only LotArea and Street are included, LotArea becomes a dominant predictor for SalePrice.
- Model 3 has the lowest coefficient (~0.63), possibly because of the log transformation applied to SalePrice, which changes the linearity assumption and reduces the direct impact of LotArea.
- Model 6 shows a reduction in coefficient (~0.86), indicating that when more relevant features like 1stFlrSF, Neighborhood, and Year are included, the model relies less heavily on LotArea alone.

Conclusion:

- The coefficient of **LotArea** is highly sensitive to the presence of other features in the regression model. When used alone or with minimal additional data, **LotArea** appears to have a strong influence on **SalePrice**. However, as more comprehensive features are added, especially ones more directly tied to the quality and structure of the house (like **1stFlrSF** and **YearBuilt**), the model begins to distribute importance across features, thus **diluting the impact of LotArea**.
- This comparison highlights the importance of **feature selection** and **model context** in interpreting coefficients in regression models.

viii. Use the polynomial regression of degree 2 and 3, to estimate the value of the weights/coefficients for predicting "SalePrice" based on "LotArea". Print the graph on the training and test set (Bonus).

Objective:

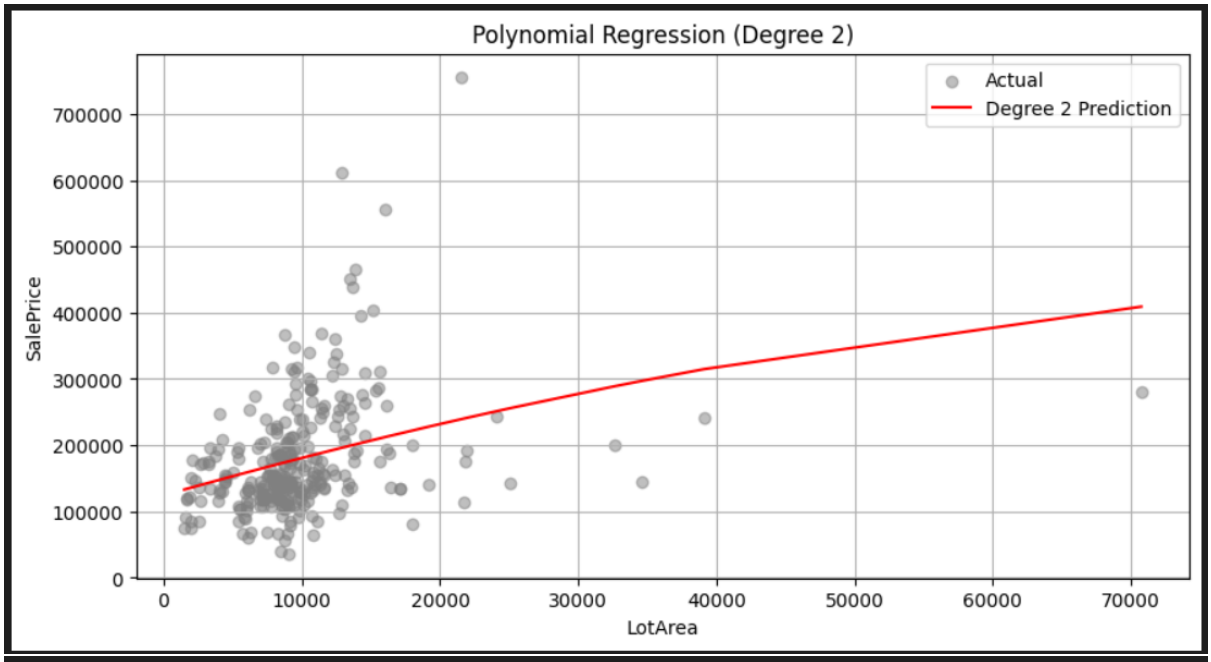
To evaluate whether introducing non-linear terms through polynomial regression improves the predictive power of the model for estimating SalePrice using only the LotArea feature.

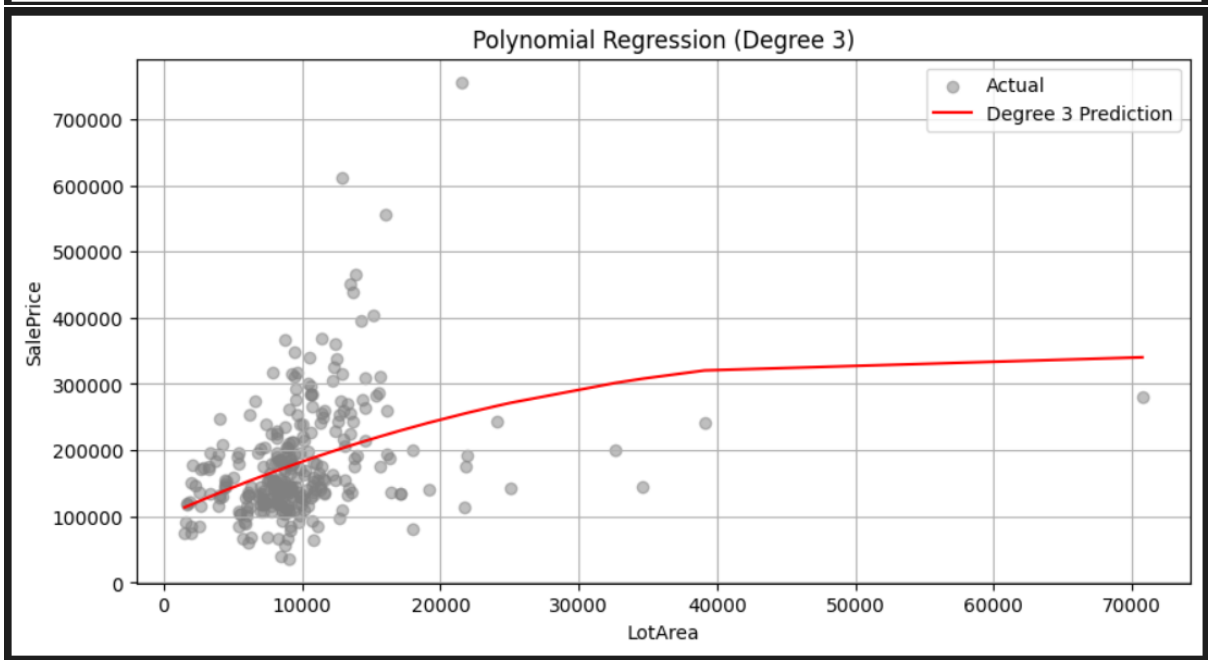
Implementation Summary:

- Two models were trained:
    - **Degree 2 Polynomial Regression**
    - **Degree 3 Polynomial Regression**

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

- The polynomial features were generated using PolynomialFeatures(degree=2) and PolynomialFeatures(degree=3) respectively.
- A LinearRegression model was then fit on the transformed features.
- Graphs were plotted showing the actual sale prices (grey dots) and the polynomial predictions (red curves).





Observations from Graphs:

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087

Degree 2 Model (Quadratic):

- The red curve shows a **smooth upward trend**, capturing some of the curvature in the relationship between LotArea and SalePrice.
- It improves upon the linear model by fitting the data more flexibly without overfitting.
- However, it still misses some of the outlier patterns, especially in higher LotArea values.

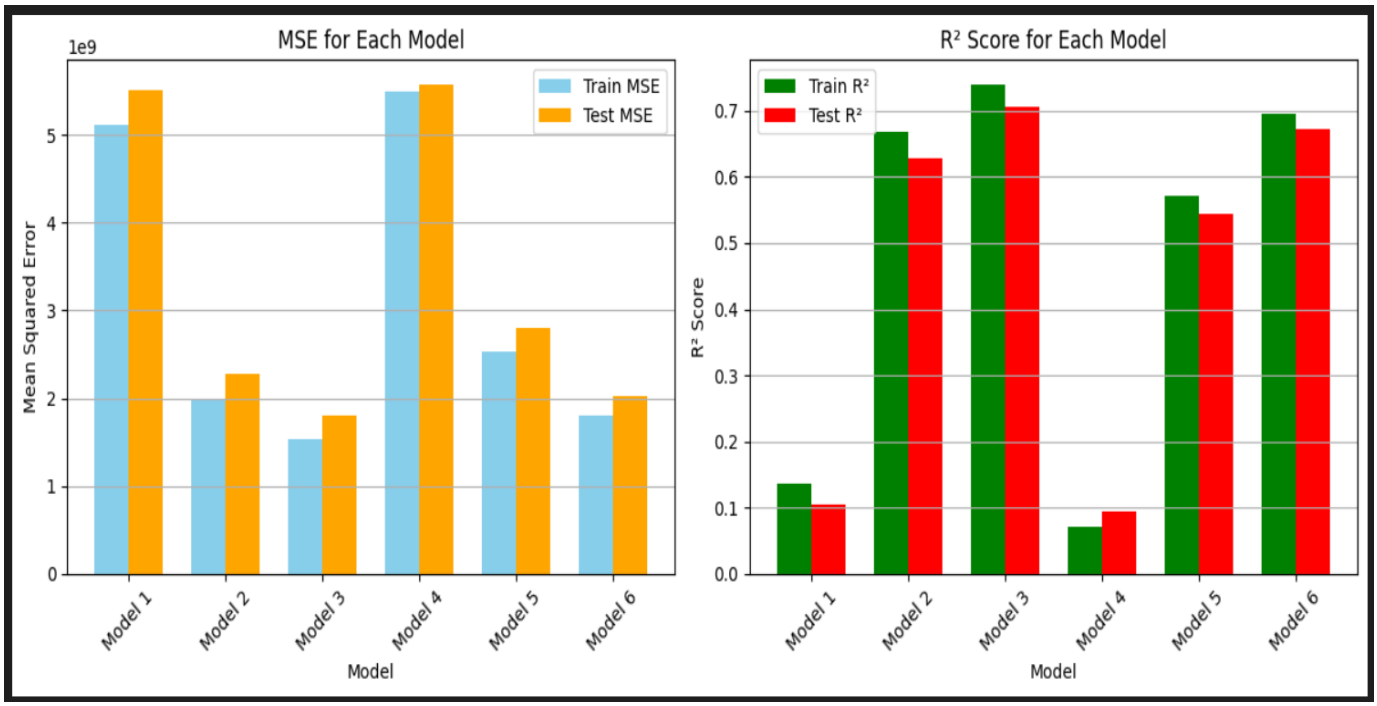
Degree 3 Model (Cubic):

- The red curve is more **flexible and adaptive**, especially at the tails of the distribution.
- It captures more subtle variations and appears to slightly outperform the degree 2 model visually in terms of curve fitting.
- However, degree 3 models may be more prone to **overfitting**, especially when the training data is limited or noisy.

Conclusion:

- **Polynomial regression** is a viable method for capturing non-linear trends between LotArea and SalePrice.
- Both degree 2 and degree 3 models provide a better fit than simple linear regression for this feature.
- Between the two, **degree 3** offers more flexibility and may model complex relationships better, **but it should be used cautiously** to avoid overfitting.
- The performance of these models should ideally be evaluated using **Mean Squared Error (MSE)** and **R^2 Score** on both train and test datasets to make a definitive conclusion.

NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087



NAME: MOODU ROOPA
ENROLLMENT ID: 2022CSB087