Great! You're working with the **Kaggle House Prices - Advanced Regression** dataset, and you've done a good job cleaning and separating your features (X) and target (Y). Now, when it comes to **visualizing the dataset**, it helps to choose plots that highlight both:

* **Target variable (Y) distribution**
* **Relationships between features and target**
* **Data types and correlations**
* **Missing values (though you've already handled them)**

Here are some **recommended plots** to understand your dataset better:

**🔹 1. Target Variable Distribution**

Start by visualizing the distribution of SalePrice (Y) to understand skewness and outliers.

import seaborn as sns

import matplotlib.pyplot as plt

sns.histplot(Y, kde=True)

plt.title('Distribution of Sale Price')

plt.xlabel('Sale Price')

plt.ylabel('Frequency')

plt.show()

**🔹 2. Correlation Heatmap (Numerical Features)**

This shows how strongly each numerical feature correlates with the target and with each other.

import numpy as np

# Append the target to X for correlation

corr\_matrix = train\_df\_d.corr(numeric\_only=True)

plt.figure(figsize=(16, 10))

sns.heatmap(corr\_matrix, cmap='coolwarm', annot=False, fmt=".2f", linewidths=0.5)

plt.title('Correlation Heatmap')

plt.show()

To focus only on top correlated features with SalePrice:

top\_corr = corr\_matrix['SalePrice'].abs().sort\_values(ascending=False)[1:11]

print(top\_corr)

**🔹 3. Pairplot / Scatterplot (Top Correlated Features)**

Visualize relationships between top features and the target.

top\_features = top\_corr.index.tolist()

sns.pairplot(train\_df\_d[top\_features + ['SalePrice']])

plt.show()

Or just individual scatter plots for continuous features:

for col in top\_features:

plt.figure(figsize=(6, 4))

sns.scatterplot(x=train\_df\_d[col], y=Y)

plt.title(f'SalePrice vs {col}')

plt.xlabel(col)

plt.ylabel('Sale Price')

plt.show()

**🔹 4. Boxplots (Categorical vs Target)**

Helps to see how categorical variables influence the target.

categorical\_cols = train\_df\_d.select\_dtypes(include=['object']).columns

for col in categorical\_cols[:5]: # visualize a few at a time

plt.figure(figsize=(10, 4))

sns.boxplot(x=col, y='SalePrice', data=train\_df\_d)

plt.title(f'SalePrice by {col}')

plt.xticks(rotation=45)

plt.show()

**🔹 5. Countplot for Categorical Variables**

To understand category distributions.

for col in categorical\_cols[:5]: # limit for clarity

plt.figure(figsize=(10, 4))

sns.countplot(x=col, data=train\_df\_d)

plt.title(f'Count of categories in {col}')

plt.xticks(rotation=45)

plt.show()

**🔹 6. Feature Importance (after a quick model)**

If you're eager to visualize which features a model thinks are important:

from sklearn.ensemble import RandomForestRegressor

# Encode categorical variables

X\_encoded = pd.get\_dummies(X)

model = RandomForestRegressor()

model.fit(X\_encoded, Y)

importances = pd.Series(model.feature\_importances\_, index=X\_encoded.columns)

importances.nlargest(10).plot(kind='barh')

plt.title('Top 10 Feature Importances')

plt.show()

**✅ Summary of Plots to Use**

| **Plot Type** | **Purpose** |
| --- | --- |
| Histogram (Target) | Distribution of SalePrice |
| Correlation Heatmap | Numerical feature relationships |
| Scatterplot / Pairplot | Feature vs target trends |
| Boxplot | Categorical feature influence |
| Countplot | Category distribution |
| Feature Importance | Model-driven insights |

Let me know if you'd like code for interactive visualizations (e.g., Plotly) or specific plots for your most correlated variables!