```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Data Preparation and Analysis Report: Property_Price Dataset

Introduction

The objective of this analysis was to clean and prepare the Property_Price dataset for further analysis. The initial dataset consisted of 1459 rows and 81 columns.

Data Cleaning Process

1. Initial Data Shape:

The original dataset had a shape of (1459, 81).

2. Removing Unnecessary Columns:

The 'ID' column was removed as it did not provide any valuable information for the analysis.

After further analysis, the following columns were identified as unnecessary and were removed: 'Lot_Extent', 'Lane_Type', 'Brick_Veneer_Type', 'Fireplace_Quality', 'Pool_Quality', 'Fence_Quality', and 'Miscellaneous_Feature'.

The dataset shape after removing these columns was (1459, 73).

3. Handling Missing Values:

Missing values in the dataset were handled using appropriate imputation methods:

- Mean: Used for numerical columns.
- Mode: Used for categorical columns.
- Median: Used for columns where data distribution was skewed
- The columns with imputed values include:-
- ["Basement_Height", "Basement_Condition", "Exposure_Level", "BsmtFinType1", "BsmtFinType2", "Electrical_System", "Garage", "Garage_Finish_Year", "Garage_Quality", "Garage_Condition"]

After performing the above steps, the dataset was cleaned and ready for further analysis.

Conclusion

- The data cleaning process involved removing unnecessary columns, handling missing values, and ensuring the dataset was in a usable format. The final shape of the cleaned dataset is (1459, 73). This cleaned dataset provides a solid foundation for subsequent analysis and modeling efforts.

To gain a thorough understanding of our dataset, we utilized the Pandas Profiling tool for comprehensive data analysis. This tool facilitated both univariate and bivariate analyses, as well as correlation assessments.

- link of Pandas Profiling Analysis: https://drive.google.com/file/d/1zGGV5WYWxdVioV2zwHb9-hLP1aSp7ydd/view?usp=drive_link

After completing the pandas profiling analysis, I proceeded to train and test the model using the prepared dataset. This involved splitting the data into training and testing sets, training various models, and evaluating their performance based on the testing data.

```
df_new = pd.read_csv("file_without_null_values.csv")

df_new.shape

(1459, 73)

X = df_new.iloc[:,0:-1]
y = df_new.iloc[:,-1]

X.head(2)

   Building_Class Zoning_Class  Lot_Size Road_Type Property_Shape  \
0              60          RLD      8450     Paved            Reg
1              20          RLD      9600     Paved            Reg

  Land_Outline Utility_Type Lot_Configuration Property_Slope
Neighborhood  \
0          Lvl       AllPub                 I             GS
CollgCr
```

```
1          Lvl       AllPub              FR2P             GS
Veenker

   ... Open_Lobby_Area Enclosed_Lobby_Area Three_Season_Lobby_Area  \
0  ...       69.596115           20.337934                       0
1  ...       74.716033           15.039392                       0

  Screen_Lobby_Area  Pool_Area  Miscellaneous_Value  Month_Sold
Year_Sold  \
0                  0          0                    0           2
2008
1                  0          0                    0           5
2007

  Sale_Type Sale_Condition
0        WD         Normal
1        WD         Normal

[2 rows x 72 columns]
```

```python
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

encoder = OneHotEncoder(sparse_output=False)
categorical_cols = X.select_dtypes(include=['object']).columns
numerical_cols = X.select_dtypes(include=['float64', 'int64']).columns

preprocessor = ColumnTransformer(
    transformers=[('num', Pipeline(steps=[('scaler',
StandardScaler())]), numerical_cols),
        ('cat', OneHotEncoder(sparse_output=False,
handle_unknown="ignore"), categorical_cols)
    ])  # keep the other columns as is

df_encoded = preprocessor.fit_transform(X)

encoded_feature_names1 =
preprocessor.named_transformers_['cat'].get_feature_names_out(categori
cal_cols)
encoded_feature_names2 =
preprocessor.named_transformers_['num'].get_feature_names_out()
all_feature_names = list(encoded_feature_names1) +
list(encoded_feature_names2)

len(all_feature_names)
```

```
267
```

```python
df_encoded.shape
```

```
(1459, 267)

df_encoded = pd.DataFrame(df_encoded, columns=all_feature_names)

df_encoded

      Zoning_Class_Commer  Zoning_Class_FVR  Zoning_Class_RHD  \
0                0.072771         -0.207111          0.650852
1               -0.873090         -0.091895         -0.072372
2                0.072771          0.073415          0.650852
3                0.309236         -0.096904          0.650852
4                0.072771          0.374980          1.374077
...                   ...               ...               ...
1454            -0.873090         -0.302289          0.650852
1455             0.072771         -0.260511         -0.072372
1456            -0.873090          0.266277         -0.072372
1457             0.309236         -0.147800          0.650852
1458            -0.873090         -0.080173         -0.795596

      Zoning_Class_RLD  Zoning_Class_RMD  Road_Type_Gravel
Road_Type_Paved  \
0            -0.516787          1.050507          0.877986
0.510905
1             2.179252          0.156540         -0.430226        -
0.574674
2            -0.516787          0.984287          0.829534
0.322590
3            -0.516787         -1.863163         -0.720940        -
0.574674
4            -0.516787          0.951177          0.732629
1.363861
...                ...               ...               ...
...
1454         -0.516787          1.083617          0.974891        -
0.574674
1455         -0.516787          0.918068          0.732629        -
0.574674
1456          0.381893          0.222760          0.151202
0.084428
1457          3.077931         -1.002306          1.023343        -
0.574674
1458          0.381893         -0.704317          0.538820        -
0.574674

      Property_Shape_IR1  Property_Shape_IR2  Property_Shape_IR3  ...
\
0               0.575950           -0.287744           -0.945245  ...

1               1.172460           -0.287744           -0.641887  ...
```

```
2                 0.093479           -0.287744           -0.302307  ...
3                -0.498645           -0.287744           -0.062338  ...
4                 0.464105           -0.287744           -0.175531  ...
...                    ...                 ...                 ...  ...
1454             -0.073193           -0.287744            0.551169  ...
1455             -0.972344           -0.287744            0.872638  ...
1456              0.760166            0.723464            0.048592  ...
1457             -0.369255           -0.287744            0.700585  ...
1458             -0.864884            6.095898           -1.284824  ...

      Garage_Area  W_Deck_Area  Open_Lobby_Area
Enclosed_Lobby_Area  \
0             0.0          0.0              0.0                    1.0
1             0.0          0.0              0.0                    1.0
2             0.0          0.0              0.0                    1.0
3             0.0          0.0              0.0                    1.0
4             0.0          0.0              0.0                    1.0
...           ...          ...              ...                    ...
1454          0.0          0.0              0.0                    1.0
1455          0.0          0.0              0.0                    1.0
1456          0.0          0.0              0.0                    1.0
1457          0.0          0.0              0.0                    1.0
1458          0.0          0.0              0.0                    1.0

      Three_Season_Lobby_Area  Screen_Lobby_Area  Pool_Area  \
0                         0.0                0.0        0.0
1                         0.0                0.0        0.0
2                         0.0                0.0        0.0
3                         1.0                0.0        0.0
4                         0.0                0.0        0.0
...                       ...                ...        ...
```

```
1454                      0.0                  0.0         0.0
1455                      0.0                  0.0         0.0
1456                      0.0                  0.0         0.0
1457                      0.0                  0.0         0.0
1458                      0.0                  0.0         0.0

      Miscellaneous_Value  Month_Sold  Year_Sold
0                     0.0         1.0        0.0
1                     0.0         1.0        0.0
2                     0.0         1.0        0.0
3                     0.0         0.0        0.0
4                     0.0         1.0        0.0
...                   ...         ...        ...
1454                  0.0         1.0        0.0
1455                  0.0         1.0        0.0
1456                  0.0         1.0        0.0
1457                  0.0         1.0        0.0
1458                  0.0         1.0        0.0

[1459 rows x 267 columns]
```

```
aaa = preprocessor.named_transformers_['num']
["scaler"].get_feature_names_out()
aaa
len(aaa)
```

```
35
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df_encoded, y,
test_size=0.2, random_state=42)
```

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

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

```
LinearRegression()
```

```
from sklearn.metrics import mean_squared_error, r2_score
y_pred = model.predict(X_test)
```

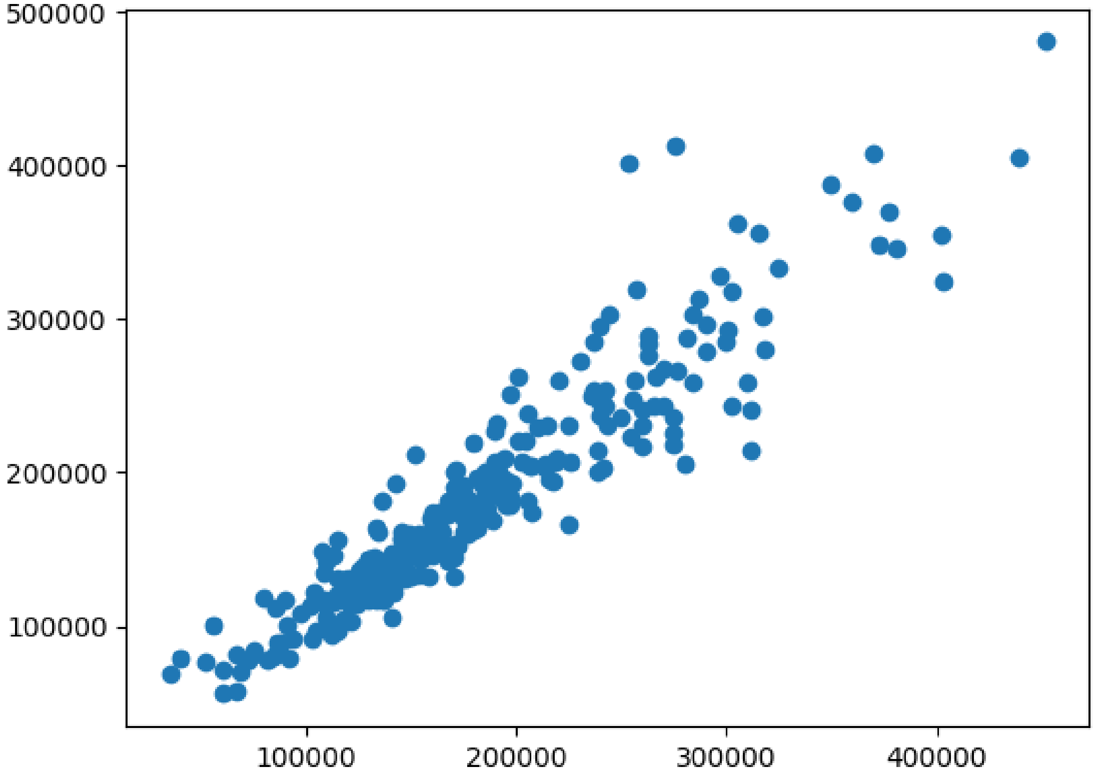
```
# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```
print("R2 score: ", r2)
print("MSE: ", mse)
```

```
R2 score:  -6.340669423254766e+17
MSE:  3.334346388367599e+27
```

```
## Means the data is not linear

columns_to_scale = ['Lot_Size', 'Brick_Veneer_Area', 'BsmtFinSF2',
"Underground_Half_Bathroom", "Kitchen_Above_Grade",
                   "Screen_Lobby_Area", "Miscellaneous_Value"]

new_drop_cols = ["Month_Sold", "Year_Sold"]

df_new = df_new.drop(new_drop_cols, axis=1)

df_new.shape

(1459, 71)

X = df_new.iloc[:,0:-1]
y = df_new.iloc[:,-1]

X.shape

(1459, 70)

encoder = OneHotEncoder(sparse_output=False)
categorical_cols = X.select_dtypes(include=['object']).columns
numerical_cols = X.select_dtypes(include=['float64', 'int64']).columns

preprocessor = ColumnTransformer(
    transformers=[('num', Pipeline(steps=[('scaler',
StandardScaler())]), numerical_cols),
        ('cat', OneHotEncoder(sparse_output=False,
handle_unknown="ignore"), categorical_cols)
    ])  # keep the other columns as is

df_encoded = preprocessor.fit_transform(X)

encoded_feature_names1 =
preprocessor.named_transformers_['cat'].get_feature_names_out(categori
cal_cols)
encoded_feature_names2 =
preprocessor.named_transformers_['num'].get_feature_names_out()
all_feature_names = list(encoded_feature_names1) +
list(encoded_feature_names2)

df_encoded = pd.DataFrame(df_encoded, columns=all_feature_names)

X_train, X_test, y_train, y_test = train_test_split(df_encoded, y,
test_size=0.2, random_state=42)

# Evaluation with xgboost

from sklearn.linear_model import LinearRegression
import xgboost as xgb

model = LinearRegression()
```

```
xgb = xgb.XGBRegressor(objective='reg:squarederror',
eval_metric='rmse')

model.fit(X_train, y_train)
xgb.fit(X_train, y_train)

XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None,
early_stopping_rounds=None,
             enable_categorical=False, eval_metric='rmse',
feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None,
max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan,
monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)

from sklearn.metrics import mean_squared_error, r2_score
y_pred = model.predict(X_test)
y_pred2 = xgb.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
mse2 = mean_squared_error(y_test, y_pred2)

r2 = r2_score(y_test, y_pred)
r22 = r2_score(y_test, y_pred2)

print("R2 score: ", r2)
print("MSE: ", mse)

print("R2 score: ", r22)
print("MSE: ", mse2)

R2 score:  -5.154654906119597e+18
MSE:  2.7106609447995225e+28
R2 score:  0.8714097120812617
MSE:  676213398.7437927

# accuracy with xgboost is 87.14%

plt.scatter(y_test, y_pred2)

<matplotlib.collections.PathCollection at 0x2202250f9b0>
```




```
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import Ridge, Lasso, ElasticNet
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
import pandas as pd

# Define models
models = {
    'Decision Tree': DecisionTreeRegressor(),
    'Random Forest': RandomForestRegressor(),
    'Gradient Boosting': GradientBoostingRegressor(),
    'SVR': SVR(),
    'k-NN': KNeighborsRegressor(),
    'Ridge': Ridge(),
    'Lasso': Lasso(),
    'ElasticNet': ElasticNet(),
    'Neural Network': MLPRegressor(max_iter=1000)
}
```

```
# Evaluate models
results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    results[name] = {'MSE': mse, 'R²': r2}

# Print results
for name, metrics in results.items():
    print(f'{name} - MSE: {metrics["MSE"]:.4f}, R²:
{metrics["R²"]:.4f}')
```

```
C:\Users\kumar\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\linear_model\_coordinate_descent.py:678: ConvergenceWarning:

Objective did not converge. You might want to increase the number of iterations, check the scale of the features or consider increasing regularisation. Duality gap: 1.749e+10, tolerance: 7.671e+08


Decision Tree - MSE: 1546437316.3527, R²: 0.7059
Random Forest - MSE: 561817282.8983, R²: 0.8932
Gradient Boosting - MSE: 489784031.8279, R²: 0.9069
SVR - MSE: 5558806379.5032, R²: -0.0571
k-NN - MSE: 1125523321.1477, R²: 0.7860
Ridge - MSE: 579639488.3872, R²: 0.8898
Lasso - MSE: 579548159.6933, R²: 0.8898
ElasticNet - MSE: 578628630.1335, R²: 0.8900
Neural Network - MSE: 6304341107.6862, R²: -0.1988

C:\Users\kumar\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:691: ConvergenceWarning:

Stochastic Optimizer: Maximum iterations (1000) reached and the optimization hasn't converged yet.
```

```
# I have checked with the various models
# the r2 score with Gradient Boosting model is hightest of 90.6%
```

Applied OneHotEncoding and Standard Scaling using a ColumnTransformer in a pipeline.

Initially used a linear regression model, but the R^2 score was unsatisfactory due to the data's non-linear nature.

Removed two columns and achieved an R^2 score of 87% with XGBoost.

I further evaluated various models and found that Gradient Boosting provided the highest R^2 score of 91.47%, while models like SVR and Neural Networks performed poorly.

```
df = pd.read_csv("file_without_null_values.csv")

df.shape

(1459, 73)
```

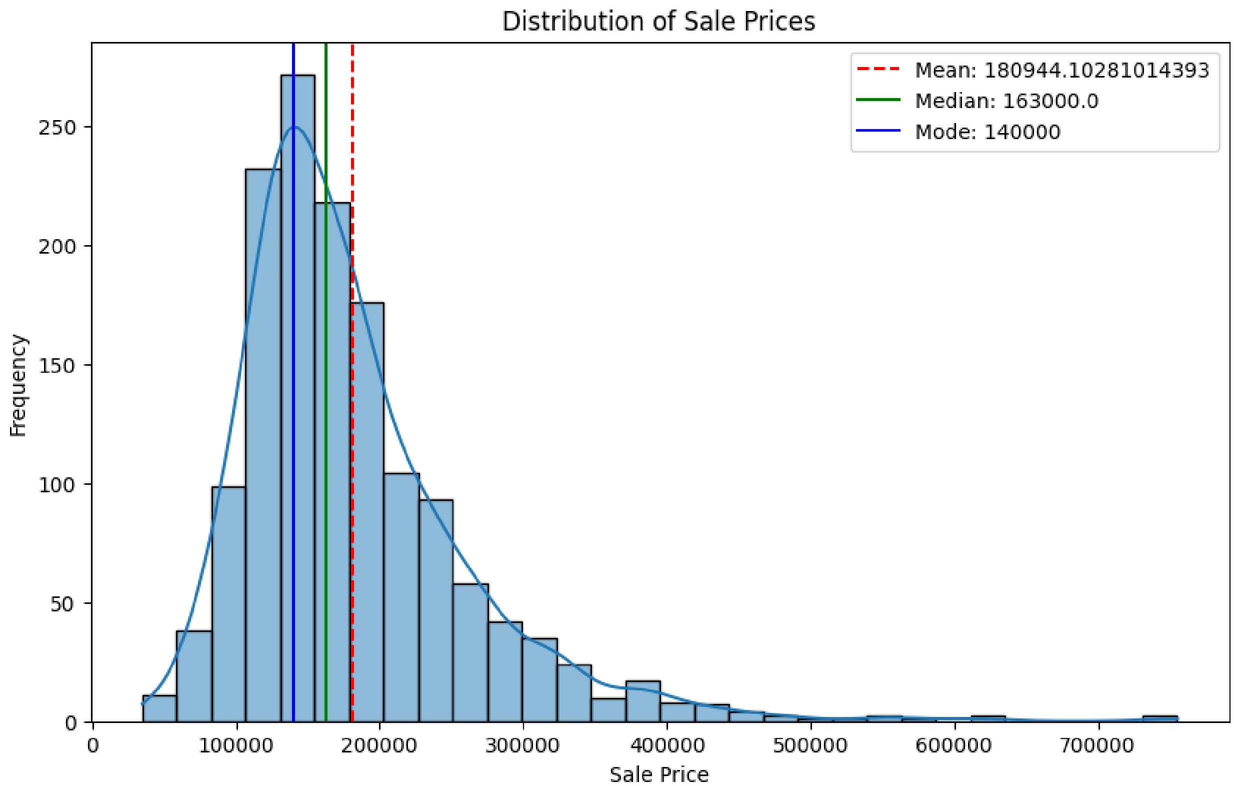
After evaluating various models, I performed Exploratory Data Analysis(EDA) on the data.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 73 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   Building_Class               1459 non-null   int64 
 1   Zoning_Class                 1459 non-null   object
 2   Lot_Size                     1459 non-null   int64 
 3   Road_Type                    1459 non-null   object
 4   Property_Shape               1459 non-null   object
 5   Land_Outline                 1459 non-null   object
 6   Utility_Type                 1459 non-null   object
 7   Lot_Configuration            1459 non-null   object
 8   Property_Slope               1459 non-null   object
 9   Neighborhood                 1459 non-null   object
 10  Condition1                   1459 non-null   object
 11  Condition2                   1459 non-null   object
 12  House_Type                   1459 non-null   object
 13  House_Design                 1459 non-null   object
 14  Overall_Material             1459 non-null   int64 
 15  House_Condition              1459 non-null   int64 
```

```
 16  Construction_Year          1459 non-null   int64
 17  Remodel_Year               1459 non-null   int64
 18  Roof_Design                1459 non-null   object
 19  Roof_Quality               1459 non-null   object
 20  Exterior1st                1459 non-null   object
 21  Exterior2nd                1459 non-null   object
 22  Brick_Veneer_Area          1459 non-null   float64
 23  Exterior_Material          1459 non-null   object
 24  Exterior_Condition         1459 non-null   object
 25  Foundation_Type            1459 non-null   object
 26  Basement_Height            1459 non-null   object
 27  Basement_Condition         1459 non-null   object
 28  Exposure_Level             1459 non-null   object
 29  BsmtFinType1               1459 non-null   object
 30  BsmtFinSF1                 1459 non-null   int64
 31  BsmtFinType2               1459 non-null   object
 32  BsmtFinSF2                 1459 non-null   int64
 33  BsmtUnfSF                  1459 non-null   int64
 34  Total_Basement_Area        1459 non-null   int64
 35  Heating_Type               1459 non-null   object
 36  Heating_Quality            1459 non-null   object
 37  Air_Conditioning           1459 non-null   object
 38  Electrical_System          1459 non-null   object
 39  First_Floor_Area           1459 non-null   int64
 40  Second_Floor_Area          1459 non-null   int64
 41  LowQualFinSF               1459 non-null   int64
 42  Grade_Living_Area          1459 non-null   int64
 43  Underground_Full_Bathroom  1459 non-null   int64
 44  Underground_Half_Bathroom  1459 non-null   int64
 45  Full_Bathroom_Above_Grade  1459 non-null   int64
 46  Half_Bathroom_Above_Grade  1459 non-null   int64
 47  Bedroom_Above_Grade        1459 non-null   int64
 48  Kitchen_Above_Grade        1459 non-null   int64
 49  Kitchen_Quality            1459 non-null   object
 50  Rooms_Above_Grade          1459 non-null   int64
 51  Functional_Rate            1459 non-null   object
 52  Fireplaces                 1459 non-null   int64
 53  Garage                     1459 non-null   object
 54  Garage_Built_Year          1459 non-null   int64
 55  Garage_Finish_Year         1459 non-null   object
 56  Garage_Size                1459 non-null   int64
 57  Garage_Area                1459 non-null   float64
 58  Garage_Quality             1459 non-null   object
 59  Garage_Condition           1459 non-null   object
 60  Pavedd_Drive               1459 non-null   object
 61  W_Deck_Area                1459 non-null   float64
 62  Open_Lobby_Area            1459 non-null   float64
 63  Enclosed_Lobby_Area        1459 non-null   float64
 64  Three_Season_Lobby_Area    1459 non-null   int64
```

```
 65  Screen_Lobby_Area          1459 non-null   int64
 66  Pool_Area                  1459 non-null   int64
 67  Miscellaneous_Value        1459 non-null   int64
 68  Month_Sold                 1459 non-null   int64
 69  Year_Sold                  1459 non-null   int64
 70  Sale_Type                  1459 non-null   object
 71  Sale_Condition             1459 non-null   object
 72  Sale_Price                 1459 non-null   int64
dtypes: float64(5), int64(31), object(37)
memory usage: 832.2+ KB
```

```
mean_price = df['Sale_Price'].mean()
median_price = df['Sale_Price'].median()
mode_price = df['Sale_Price'].mode()[0]

plt.figure(figsize=(10, 6))
sns.histplot(df['Sale_Price'], bins=30, kde=True)
plt.axvline(mean_price, color='r', linestyle='--', label=f'Mean:
{mean_price}')
plt.axvline(median_price, color='g', linestyle='-', label=f'Median:
{median_price}')
plt.axvline(mode_price, color='b', linestyle='-', label=f'Mode:
{mode_price}')
plt.legend()
plt.title('Distribution of Sale Prices')
plt.xlabel('Sale Price')
plt.ylabel('Frequency')
plt.show()
```




```
print(mean_price, median_price, mode_price)
```

```
180944.10281014393 163000.0 140000
```

Mean (Average) Price: 180944.10

The mean price is the average housing price.

This measure is useful because it provides a general idea of the overall market price.

However, it can be affected by extremely high or low prices (outliers).

Median Price: 163000.0

The median represents the price at which half the houses are priced higher and half are priced lower.

Mode Price: 140000

The mode is the most frequently occurring price in the dataset.

This measure is useful to identify the most common price point in the market.

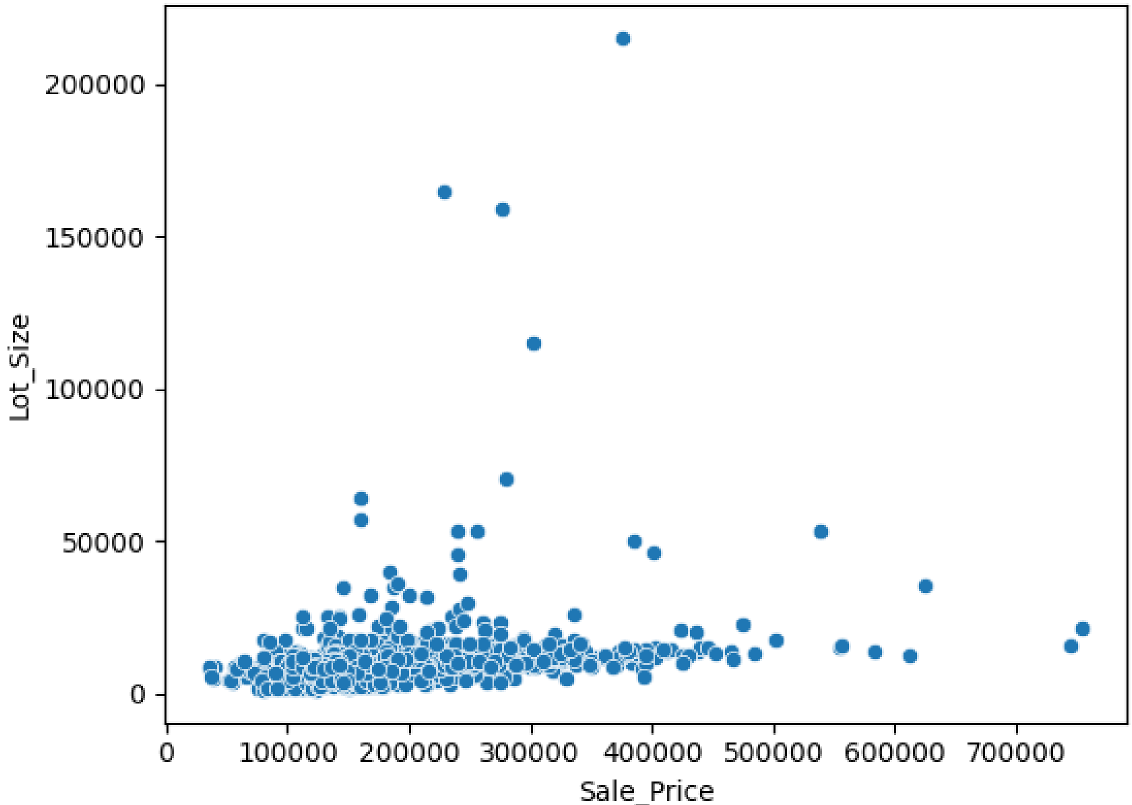
```
import matplotlib.pyplot as plt
import seaborn as sns

# Sample data for illustration
prices = [140000, 150000, 160000, 163000, 165000, 170000, 180000,
180000, 200000, 250000]

mean_price = sum(prices) / len(prices)
median_price = sorted(prices)[len(prices) // 2]
mode_price = 140000

plt.figure(figsize=(10, 6))
sns.histplot(prices, bins=10, kde=True)
plt.axvline(mean_price, color='r', linestyle='--', label=f'Mean: $
{mean_price:,.2f}')
plt.axvline(median_price, color='g', linestyle='-', label=f'Median: $
{median_price:,.2f}')
plt.axvline(mode_price, color='b', linestyle='-', label=f'Mode: $
{mode_price:,.2f}')
plt.legend()
plt.title('Distribution of Housing Prices')
plt.xlabel('Price')
```

```
plt.ylabel('Frequency')
plt.show()
```

Above picture provides a clear and comprehensive explanation of the central tendency measures helping to understand the key aspects of the sales price data.

```
sns.scatterplot(data = df, y = "Lot_Size", x = "Sale_Price")
```

```
<Axes: xlabel='Sale_Price', ylabel='Lot_Size'>
```




```
correlation = df['Lot_Size'].corr(df['Sale_Price'])
correlation
```

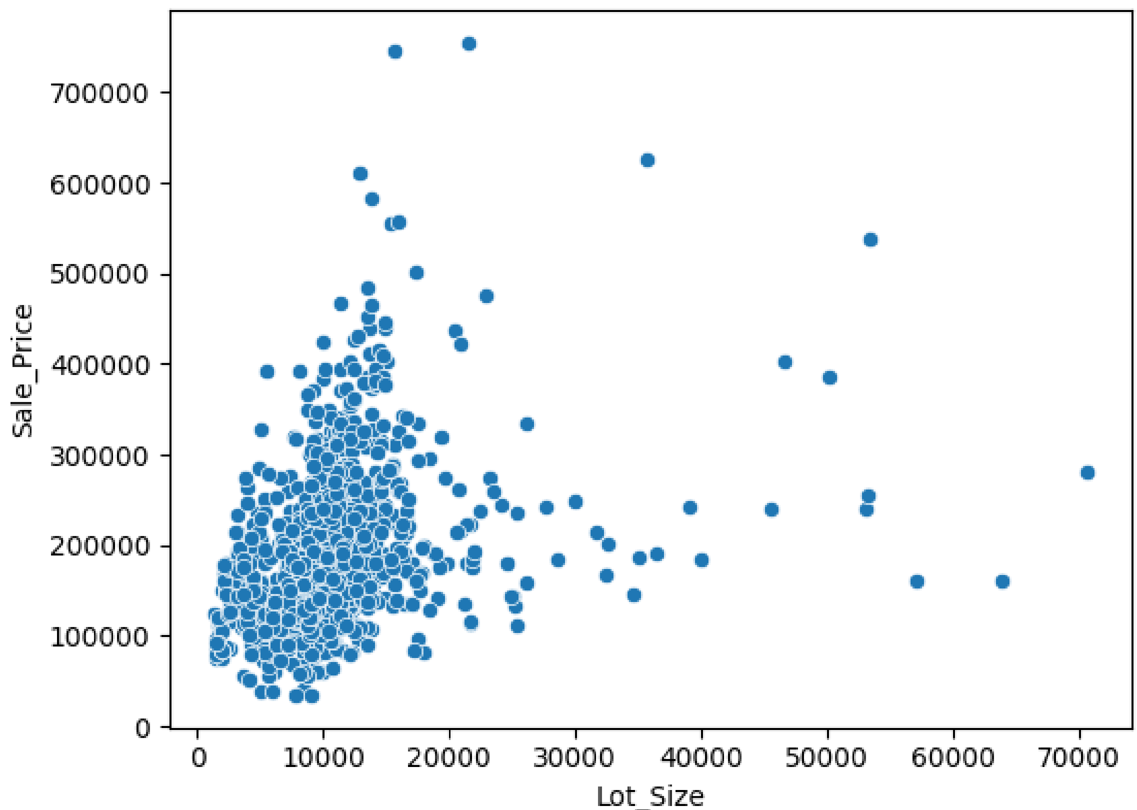
```
0.2638429115653823
```

```
lot_sale_df = df[["Lot_Size", "Sale_Price"]]
lot_sale_df = lot_sale_df[lot_sale_df["Lot_Size"] < 100000]
correlation = lot_sale_df['Lot_Size'].corr(lot_sale_df['Sale_Price'])
correlation
```

```
0.3545076735098013
```

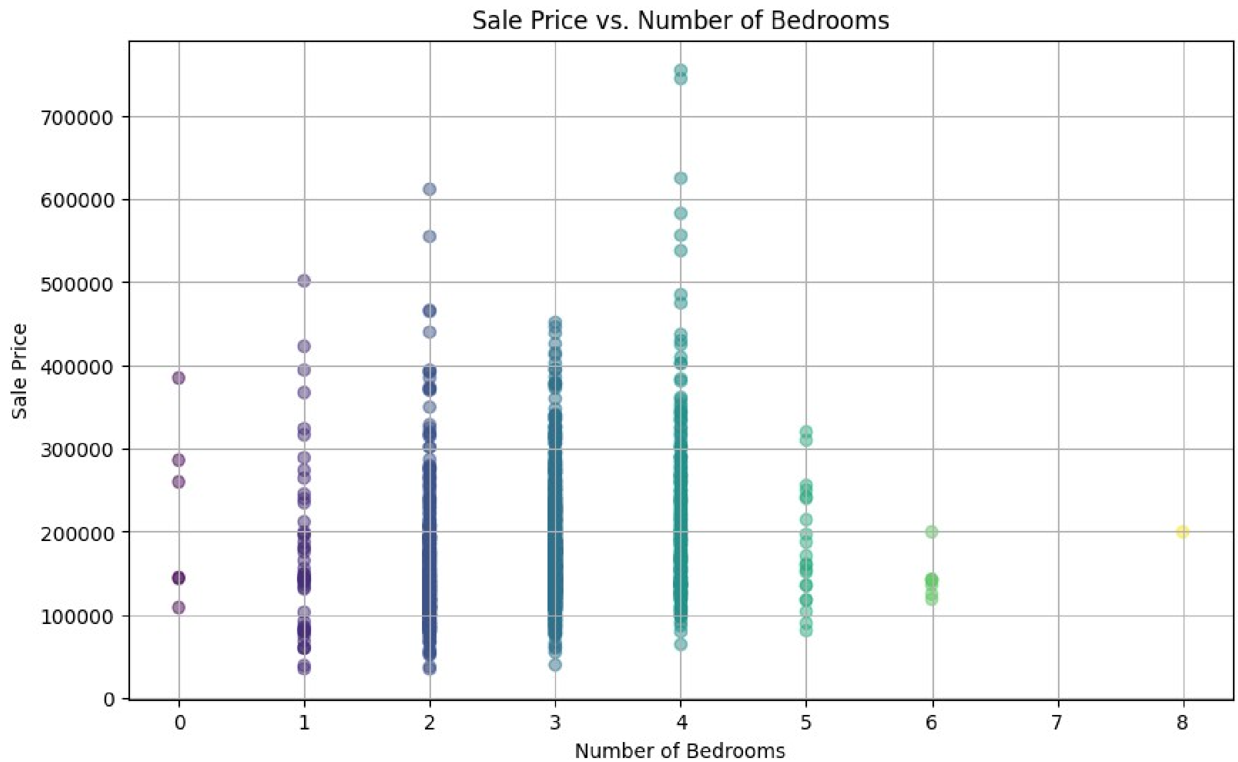
```
sns.scatterplot(data = lot_sale_df, x = "Lot_Size", y = "Sale_Price")
```

```
<Axes: xlabel='Lot_Size', ylabel='Sale_Price'>
```



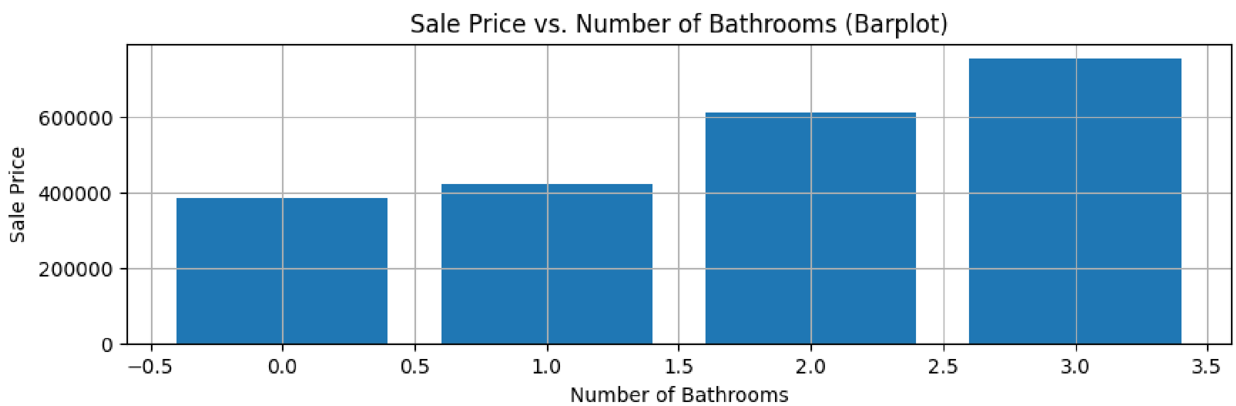
larger lots sell for higher prices. But, sometimes, there are exceptions where a big lot might not sell for as much as expected or a small lot might sell for more.

```
plt.figure(figsize=(10, 6))
plt.scatter(df['Bedroom_Above_Grade'], df['Sale_Price'], alpha=0.5,
c=df["Bedroom_Above_Grade"])
plt.title('Sale Price vs. Number of Bedrooms')
plt.xlabel('Number of Bedrooms')
plt.ylabel('Sale Price')
plt.grid(True)
plt.show()
```



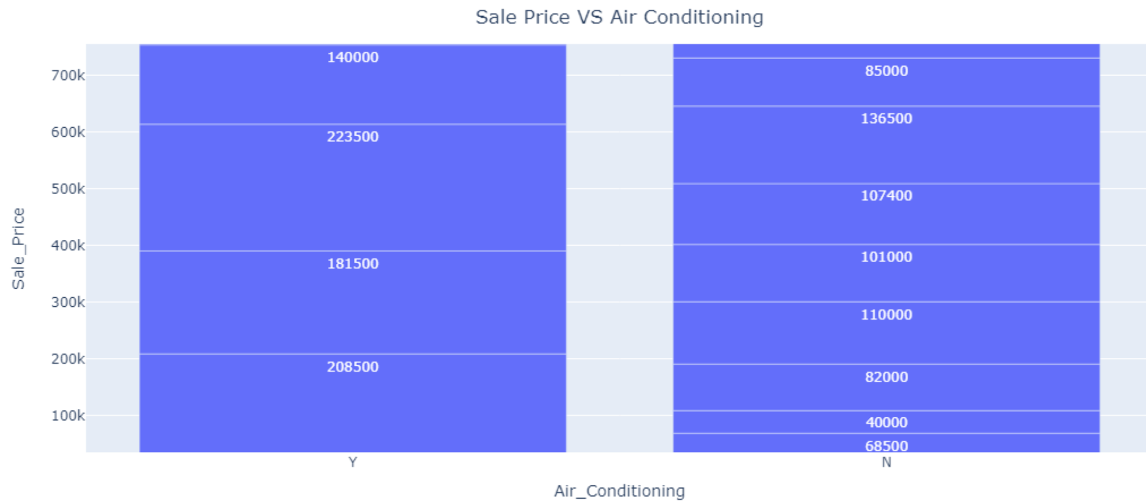
properties with more bedrooms tend to sell for more money. However, there are also some dots that fall outside of this trend. These are called outliers. Outliers are properties that have a high or low sale price compared to other properties with a similar number of bedrooms. There could be a few reasons for this, such as the location of the property, its condition, or special features that the property has.

```
plt.figure(figsize=(10, 6))
plt.subplot(2,1,1)
plt.scatter(df['Full_Bathroom_Above_Grade'], df['Sale_Price'],
alpha=0.5, c=df["Full_Bathroom_Above_Grade"])
plt.title('Sale Price vs. Number of Bathrooms (ScatterPlot)')
plt.xlabel('Number of Bathrooms')
plt.ylabel('Sale Price')
plt.grid(True)

plt.figure(figsize=(10, 6))
plt.subplot(2,1,2)
plt.bar(df['Full_Bathroom_Above_Grade'], df['Sale_Price'])
plt.title('Sale Price vs. Number of Bathrooms (Barplot)')
plt.xlabel('Number of Bathrooms')
plt.ylabel('Sale Price')
plt.grid(True)

plt.show()
```



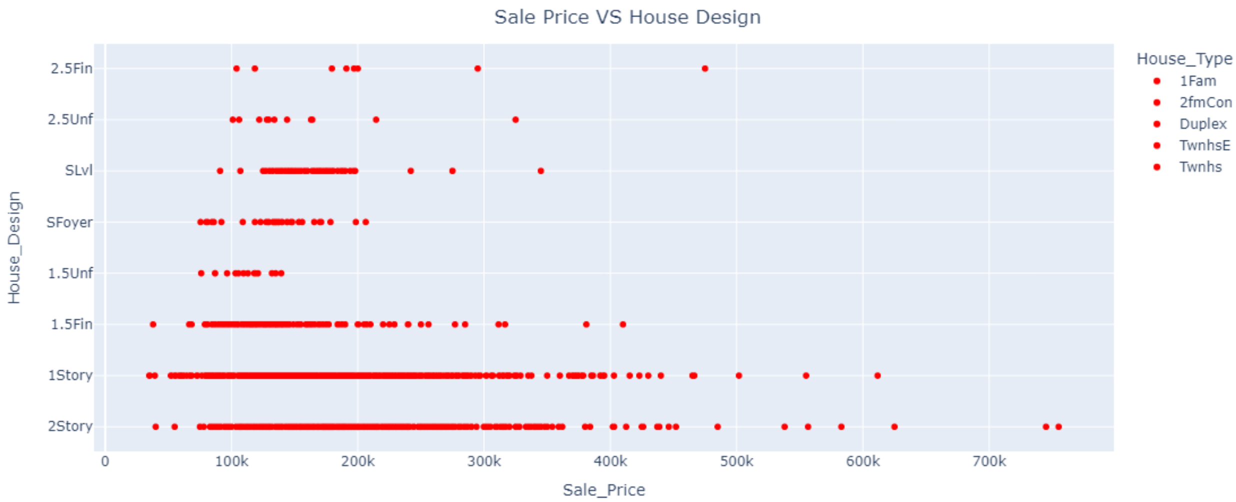
Properties with more bathrooms tend to sell for more money. However, there are also some dots that fall outside this trend.

```
fig = px.bar(df, x = 'Air_Conditioning', y =
'Sale_Price' ,text="Sale_Price")
fig.update_layout(title = "Sale Price VS Air
Conditioning",title_x=0.5,title_y=0.94, autosize=False,
width=1000,height=500)
#fig.update_traces(marker_color = 'violet', marker_line_color =
'black',marker_line_width = 2)
y_min = df["Sale_Price"].min()
y_max = df["Sale_Price"].max()
fig.update_layout(yaxis=dict(range=[y_min, y_max]))
fig.show()
```



As property prices increase, the price difference between perperties with air conditioning and those without also grows. In other words, more expensive properties tend to show a bigger price gap based on whether they have air conditioning installed.

```
df["Sale_Price"].max()

755000

fig = px.scatter(df, x = "Sale_Price", y = "House_Design",
color="House_Type")
fig.update_layout(title = "Sale Price VS House
Design",title_x=0.5,title_y=0.94, autosize=False,
width=1000,height=500)
fig.update_traces(marker_color = 'red', marker_line_color = 'black')
fig.show()
```



- 2.5Fin and 2.5Unf categories have very few records (only 20 each), indicating these designs are less common.
- 1Story and 2Story houses are the most common designs with a wide range of sale prices.
- 2Story houses generally command higher prices, indicating a likely positive correlation between having multiple stories and higher sale prices.
- The spread of prices in 1Story houses suggests a broad market segment with varying prices based on other factors (like size, location, etc.).

```
df.shape
```

```
(1459, 73)
```

Then I performed feature selection on the data. The initial dataset had a shape of (1459, 73).

```
import scipy.stats as stats

categorical_columns = df.select_dtypes(include=['object']).columns

# Perform ANOVA for each categorical column
anova_results = {}
for col in categorical_columns:
    groups = [df[df[col] == category]['Sale_Price'] for category in
df[col].unique()]
    anova_result = stats.f_oneway(*groups)
    anova_results[col] = anova_result

# Print ANOVA results
for col, result in anova_results.items():
    print(f"ANOVA result for {col}: F-statistic = {result.statistic},
p-value = {result.pvalue}")
```

```
ANOVA result for Zoning_Class: F-statistic = 43.85461265705704, p-
value = 8.623678334677968e-35
ANOVA result for Road_Type: F-statistic = 2.4601325688217863, p-value
= 0.11698606402169083
ANOVA result for Property_Shape: F-statistic = 40.059946792250386, p-
value = 7.137696914217586e-25
ANOVA result for Land_Outline: F-statistic = 12.840259882227091, p-
value = 2.7817438435761282e-08
ANOVA result for Utility_Type: F-statistic = 0.2989507717386339, p-
value = 0.5846246651848929
ANOVA result for Lot_Configuration: F-statistic = 7.7952425864006205,
p-value = 3.250291720935249e-06
ANOVA result for Property_Slope: F-statistic = 1.9526600191586492, p-
value = 0.14226753347490803
ANOVA result for Neighborhood: F-statistic = 71.73067858473284, p-
value = 2.3465866522923626e-225
ANOVA result for Condition1: F-statistic = 6.119869778841551, p-value
= 8.850238347455908e-08
ANOVA result for Condition2: F-statistic = 2.073107344843178, p-value
= 0.043510119433217
ANOVA result for House_Type: F-statistic = 13.01849279371417, p-value
= 2.0291429199969593e-10
ANOVA result for House_Design: F-statistic = 19.575876839683175, p-
value = 3.588437772489042e-25
ANOVA result for Roof_Design: F-statistic = 17.77627787904895, p-value
= 3.908164572338989e-17
ANOVA result for Roof_Quality: F-statistic = 6.721750843613629, p-
value = 7.356309273248755e-08
ANOVA result for Exterior1st: F-statistic = 18.587888208110293, p-
value = 2.977413814634284e-43
ANOVA result for Exterior2nd: F-statistic = 17.480878924817794, p-
value = 5.494558715253583e-43
ANOVA result for Exterior_Material: F-statistic = 444.28743060749485,
p-value = 7.440829120284835e-205
ANOVA result for Exterior_Condition: F-statistic = 8.80279307664569,
p-value = 5.068975761586581e-07
ANOVA result for Foundation_Type: F-statistic = 100.1377669559078, p-
value = 7.320259470842929e-91
ANOVA result for Basement_Height: F-statistic = 413.5657552033554, p-
value = 3.0551393861901977e-194
ANOVA result for Basement_Condition: F-statistic = 13.786199700269902,
p-value = 7.225349255486377e-09
ANOVA result for Exposure_Level: F-statistic = 76.47513098425473, p-
value = 6.190506171729479e-46
ANOVA result for BsmtFinType1: F-statistic = 70.42629068768852, p-
value = 4.371412794979286e-66
ANOVA result for BsmtFinType2: F-statistic = 2.362123661083842, p-
value = 0.03800421757201356
ANOVA result for Heating_Type: F-statistic = 4.259844075668412, p-
value = 0.0007534604836132154
```

```
ANOVA result for Heating_Quality: F-statistic = 88.28422610389843, p-value = 3.216320938669905e-67
ANOVA result for Air_Conditioning: F-statistic = 98.31917541469363, p-value = 1.7996181382228056e-22
ANOVA result for Electrical_System: F-statistic = 23.080700930485012, p-value = 1.6238358620842179e-18
ANOVA result for Kitchen_Quality: F-statistic = 407.4280686001325, p-value = 4.454356363811786e-192
ANOVA result for Functional_Rate: F-statistic = 6.940722716594368, p-value = 3.772707386918715e-08
ANOVA result for Garage: F-statistic = 49.59716858101429, p-value = 1.27964735338011e-55
ANOVA result for Garage_Finish_Year: F-statistic = 304.6507349889101, p-value = 2.9752698638318936e-111
ANOVA result for Garage_Quality: F-statistic = 8.457571802260452, p-value = 9.589593109138726e-07
ANOVA result for Garage_Condition: F-statistic = 8.204792968902872, p-value = 1.5286359731104103e-06
ANOVA result for Pavedd_Drive: F-statistic = 42.03604743966769, p-value = 1.784823505685642e-18
ANOVA result for Sale_Type: F-statistic = 28.835416346334068, p-value = 5.5577569033580856e-42
ANOVA result for Sale_Condition: F-statistic = 45.538709413564796, p-value = 8.737390258964316e-44
```

```
most_correlated_column = min(anova_results, key=anova_results.get)

most_correlated_column
```

```
'Utility_Type'
```

```
type(anova_results)
```

```
dict
```

```
p_values_df = pd.DataFrame(anova_results)
p_values_df = p_values_df.T
p_values_df = p_values_df.rename(columns={0:"statistic", 1:"pvalue"})

p_values_df
```

	statistic	pvalue
Zoning_Class	43.854613	8.623678e-35
Road_Type	2.460133	1.169861e-01
Property_Shape	40.059947	7.137697e-25
Land_Outline	12.840260	2.781744e-08
Utility_Type	0.298951	5.846247e-01
Lot_Configuration	7.795243	3.250292e-06
Property_Slope	1.952660	1.422675e-01
Neighborhood	71.730679	2.346587e-225
Condition1	6.119870	8.850238e-08

```
Condition2             2.073107   4.351012e-02
House_Type            13.018493   2.029143e-10
House_Design          19.575877   3.588438e-25
Roof_Design           17.776278   3.908165e-17
Roof_Quality           6.721751   7.356309e-08
Exterior1st           18.587888   2.977414e-43
Exterior2nd           17.480879   5.494559e-43
Exterior_Material    444.287431  7.440829e-205
Exterior_Condition     8.802793   5.068976e-07
Foundation_Type      100.137767   7.320259e-91
Basement_Height      413.565755  3.055139e-194
Basement_Condition    13.786200   7.225349e-09
Exposure_Level        76.475131   6.190506e-46
BsmtFinType1          70.426291   4.371413e-66
BsmtFinType2           2.362124   3.800422e-02
Heating_Type           4.259844   7.534605e-04
Heating_Quality       88.284226   3.216321e-67
Air_Conditioning      98.319175   1.799618e-22
Electrical_System     23.080701   1.623836e-18
Kitchen_Quality      407.428069  4.454356e-192
Functional_Rate        6.940723   3.772707e-08
Garage                49.597169   1.279647e-55
Garage_Finish_Year   304.650735  2.975270e-111
Garage_Quality         8.457572   9.589593e-07
Garage_Condition       8.204793   1.528636e-06
Pavedd_Drive          42.036047   1.784824e-18
Sale_Type             28.835416   5.557757e-42
Sale_Condition        45.538709   8.737390e-44
```

```
p_values_df[p_values_df["pvalue"] > 0.0001].index
```

```
Index(['Road_Type', 'Utility_Type', 'Property_Slope', 'Condition2',
       'BsmtFinType2', 'Heating_Type'],
      dtype='object')
```

```
p_cols = ['Road_Type', 'Utility_Type', 'Property_Slope', 'Condition2',
       'BsmtFinType2', 'Heating_Type']
```

```
numerical_columns = ['Building_Class', 'Lot_Size', 'Overall_Material',
'House_Condition',
       'Construction_Year', 'Remodel_Year', 'Brick_Veneer_Area',
'BsmtFinSF1',
       'BsmtFinSF2', 'BsmtUnfSF', 'Total_Basement_Area',
'First_Floor_Area',
       'Second_Floor_Area', 'LowQualFinSF', 'Grade_Living_Area',
       'Underground_Full_Bathroom', 'Underground_Half_Bathroom',
       'Full_Bathroom_Above_Grade', 'Half_Bathroom_Above_Grade',
       'Bedroom_Above_Grade', 'Kitchen_Above_Grade',
'Rooms_Above_Grade',
       'Fireplaces', 'Garage_Built_Year', 'Garage_Size',
```

```
'Garage_Area',
       'W_Deck_Area', 'Open_Lobby_Area', 'Enclosed_Lobby_Area',
       'Three_Season_Lobby_Area', 'Screen_Lobby_Area', 'Pool_Area',
       'Miscellaneous_Value', 'Month_Sold', 'Year_Sold']

df.shape

(1459, 73)

df[p_cols]

     Road_Type Utility_Type Property_Slope Condition2 BsmtFinType2  \
0        Paved       AllPub             GS       Norm          Unf
1        Paved       AllPub             GS       Norm          Unf
2        Paved       AllPub             GS       Norm          Unf
3        Paved       AllPub             GS       Norm          Unf
4        Paved       AllPub             GS       Norm          Unf
...        ...          ...            ...        ...          ...
1454     Paved       AllPub             GS       Norm          Unf
1455     Paved       AllPub             GS       Norm          Unf
1456     Paved       AllPub             GS       Norm          Rec
1457     Paved       AllPub             GS       Norm          Unf
1458     Paved       AllPub             GS       Norm          Rec

     Heating_Type
0            GasA
1            GasA
2            GasA
3            GasA
4            GasA
...           ...
1454         GasA
1455         GasA
1456         GasA
1457         GasA
1458         GasA

[1459 rows x 6 columns]

new_df = df.drop(columns=p_cols)

new_df

      Building_Class Zoning_Class  Lot_Size Property_Shape
Land_Outline  \
0                 60          RLD      8450            Reg
Lvl
1                 20          RLD      9600            Reg
Lvl
2                 60          RLD     11250            IR1
Lvl
```

```
3                  70           RLD      9550            IR1
Lvl
4                  60           RLD     14260            IR1
Lvl
...               ...           ...       ...            ...          ..
.
1454               20           FVR      7500            Reg
Lvl
1455               60           RLD      7917            Reg
Lvl
1456               20           RLD     13175            Reg
Lvl
1457               70           RLD      9042            Reg
Lvl
1458               20           RLD      9717            Reg
Lvl

     Lot_Configuration Neighborhood Condition1 House_Type House_Design
...  \
0                    I      CollgCr       Norm       1Fam       2Story
...
1                 FR2P      Veenker      Feedr       1Fam       1Story
...
2                    I      CollgCr       Norm       1Fam       2Story
...
3                    C      Crawfor       Norm       1Fam       2Story
...
4                 FR2P      NoRidge       Norm       1Fam       2Story
...
...                ...          ...        ...        ...          ...
...
1454                 I      Somerst       Norm       1Fam       1Story
...
1455                 I      Gilbert       Norm       1Fam       2Story
...
1456                 I       NWAmes       Norm       1Fam       1Story
...
1457                 I      Crawfor       Norm       1Fam       2Story
...
1458                 I        NAmes       Norm       1Fam       1Story
...

      Enclosed_Lobby_Area  Three_Season_Lobby_Area  Screen_Lobby_Area
\
0               20.337934                        0                  0

1               15.039392                        0                  0

2              -46.232198                        0                  0
```

```
3            60.921821                      0                   0

4            21.788818                      0                   0

...                ...                    ...                 ...

1454        126.676547                      0                   0

1455        125.521880                      0                   0

1456        148.266666                      0                   0

1457         54.320896                      0                   0

1458         19.498763                      0                   0


      Pool_Area Miscellaneous_Value Month_Sold Year_Sold Sale_Type  \
0             0                   0          2      2008        WD
1             0                   0          5      2007        WD
2             0                   0          9      2008        WD
3             0                   0          2      2006        WD
4             0                   0         12      2008        WD
...         ...                 ...        ...       ...       ...
1454          0                   0         10      2009        WD
1455          0                   0          8      2007        WD
1456          0                   0          2      2010        WD
1457          0                2500          5      2010        WD
1458          0                   0          4      2010        WD

      Sale_Condition Sale_Price
0             Normal     208500
1             Normal     181500
2             Normal     223500
3            Abnorml     140000
4             Normal     250000
...              ...        ...
1454          Normal     185000
1455          Normal     175000
1456          Normal     210000
1457          Normal     266500
1458          Normal     142125

[1459 rows x 67 columns]

# VIF for numerical columns
numeric_columns = []
for i in df.columns:
    if df[i].dtype != 'object' and i not in ['Sale_Price']:
        numeric_columns.append(i)
```

```python
# VIF sequentially check
vif_data = df[numeric_columns]
total_columns = vif_data.shape[1]
columns_to_be_kept = []
columns_to_remove = []
column_index = 0

from statsmodels.stats.outliers_influence import
variance_inflation_factor
for i in range (0,total_columns):

    vif_value = variance_inflation_factor(vif_data, column_index)
    print (column_index,'---',vif_value)


    if vif_value <= 6:
        columns_to_be_kept.append( numeric_columns[i] )
        column_index = column_index+1

    else:
        columns_to_remove.append(numeric_columns[i])
        vif_data = vif_data.drop([ numeric_columns[i] ] , axis=1)
```

```
0 --- 4.148455331669338
1 --- 2.554651924012021
2 --- 65.82703795275853
2 --- 40.103600251427345
2 --- 15472.946790220098
2 --- 17172.36471532405
2 --- 1.801553229835182
3 --- inf
3 --- 1.2454852917348698
4 --- 6.596344457208478
4 --- 24.228416189474
4 --- inf
4 --- 6.276880893216082
4 --- 1.1275718705532585
5 --- 49.90757655726292
5 --- 1.9976189564967686
6 --- 1.1270295712801177
7 --- 19.648306529407897
7 --- 2.0310570212310646
8 --- 27.715322494579294
8 --- 31.296402215611838
8 --- 24.736135169599343
8 --- 2.454610769324991
9 --- 7751.424691960003
9 --- 8.423280886091545
9 --- 6.051566976102079
9 --- 1.5698570770681988
```

```
10 --- 1.521739053824365
11 --- 1.1769985456955605
12 --- 1.0227222246128613
13 --- 1.1360177844475854
14 --- 1.0287547797447814
15 --- 1.0147100763692771
16 --- 6.532812241386163
16 --- 6.636596661162874

C:\Users\kumar\AppData\Local\Programs\Python\Python312\Lib\site-
packages\statsmodels\stats\outliers_influence.py:197: RuntimeWarning:

divide by zero encountered in scalar divide

C:\Users\kumar\AppData\Local\Programs\Python\Python312\Lib\site-
packages\statsmodels\stats\outliers_influence.py:197: RuntimeWarning:

divide by zero encountered in scalar divide


vif_value

6.636596661162874

vif_data

      Building_Class  Lot_Size  Brick_Veneer_Area  BsmtFinSF2
LowQualFinSF  \
0                 60      8450              196.0           0
0
1                 20      9600                0.0           0
0
2                 60     11250              162.0           0
0
3                 70      9550                0.0           0
0
4                 60     14260              350.0           0
0
...              ...       ...                ...         ...
...
1454              20      7500                0.0           0
0
1455              60      7917                0.0           0
0
1456              20     13175              119.0         163
0
1457              70      9042                0.0           0
0
1458              20      9717                0.0        1029
0
```

```
      Underground_Full_Bathroom  Underground_Half_Bathroom  \
0                             1                          0
1                             0                          1
2                             1                          0
3                             1                          0
4                             1                          0
...                         ...                        ...
1454                          1                          0
1455                          0                          0
1456                          1                          0
1457                          0                          0
1458                          1                          0

      Half_Bathroom_Above_Grade  Fireplaces  W_Deck_Area
Open_Lobby_Area  \
0                             1           0   163.788080
69.596115
1                             0           1   198.900074
74.716033
2                             1           1    26.127533
32.085268
3                             0           1    46.948018
40.181415
4                             1           1   -10.626105
20.755323
...                         ...         ...          ...
...
1454                          0           0    -9.973961        -
9.267967
1455                          1           1   -80.348891
113.043436
1456                          0           2    36.180338
221.514480
1457                          0           2    88.568242
110.888690
1458                          0           0   144.036562       -
33.654857

      Enclosed_Lobby_Area  Three_Season_Lobby_Area  Screen_Lobby_Area
\
0               20.337934                        0                  0

1               15.039392                        0                  0

2              -46.232198                        0                  0

3               60.921821                        0                  0

4               21.788818                        0                  0
```

```
...                  ...                      ...                 ...

1454         126.676547                         0                   0

1455         125.521880                         0                   0

1456         148.266666                         0                   0

1457          54.320896                         0                   0

1458          19.498763                         0                   0


      Pool_Area  Miscellaneous_Value
0             0                    0
1             0                    0
2             0                    0
3             0                    0
4             0                    0
...         ...                  ...
1454          0                    0
1455          0                    0
1456          0                    0
1457          0                 2500
1458          0                    0

[1459 rows x 16 columns]
```

```
correlation_matrix = new_df[numeric_columns].corr()
correlation_matrix
```

```
                           Building_Class  Lot_Size  Overall_Material
\
Building_Class                   1.000000 -0.139852          0.032168

Lot_Size                        -0.139852  1.000000          0.105797

Overall_Material                 0.032168  0.105797          1.000000

House_Condition                 -0.059106 -0.005621         -0.091749

Construction_Year                0.027734  0.014220          0.572342

Remodel_Year                     0.040029  0.013754          0.550453

Brick_Veneer_Area                0.022559  0.103950          0.410062

BsmtFinSF1                      -0.069364  0.214190          0.240239

BsmtFinSF2                      -0.064813  0.111317         -0.058354
```

BsmtUnfSF	-0.141426	-0.002658	0.307794
Total_Basement_Area	-0.238327	0.260870	0.538210
First_Floor_Area	-0.251685	0.299491	0.476468
Second_Floor_Area	0.307557	0.050965	0.295187
LowQualFinSF	0.046414	0.004774	-0.030502
Grade_Living_Area	0.074583	0.263119	0.592916
Underground_Full_Bathroom	0.004156	0.158265	0.111773
Underground_Half_Bathroom	-0.002477	0.048037	-0.040291
Full_Bathroom_Above_Grade	0.131076	0.126035	0.550358
Half_Bathroom_Above_Grade	0.178227	0.014316	0.274328
Bedroom_Above_Grade	-0.023346	0.119698	0.101789
Kitchen_Above_Grade	0.281672	-0.017793	-0.184040
Rooms_Above_Grade	0.040201	0.190009	0.427386
Fireplaces	-0.046165	0.271411	0.396455
Garage_Built_Year	0.098340	-0.042228	0.437818
Garage_Size	-0.040749	0.154886	0.600458
Garage_Area	-0.023996	0.006646	0.019580
W_Deck_Area	-0.000923	0.030025	0.034516
Open_Lobby_Area	-0.024260	0.032699	-0.036792
Enclosed_Lobby_Area	0.030441	0.003951	0.018161
Three_Season_Lobby_Area	-0.043906	0.020418	0.030314
Screen_Lobby_Area	-0.026199	0.043151	0.064755
Pool_Area	0.008244	0.077670	0.065143
Miscellaneous_Value	-0.007738	0.038064	-0.031461
Month_Sold	-0.013660	0.001200	0.070766
Year_Sold	-0.021330	-0.014256	-0.027277

```
                           House_Condition  Construction_Year  
Remodel_Year  \
Building_Class                   -0.059106           0.027734   
0.040029   
Lot_Size                         -0.005621           0.014220   
0.013754   
Overall_Material                 -0.091749           0.572342   
0.550453   
House_Condition                   1.000000          -0.375953   
0.074021   
Construction_Year                -0.375953           1.000000   
0.592915   
Remodel_Year                      0.074021           0.592915   
1.000000   
Brick_Veneer_Area                -0.127660           0.314706   
0.178886   
BsmtFinSF1                       -0.046466           0.249689   
0.129082   
BsmtFinSF2                        0.039867          -0.048931      -
0.066836   
BsmtUnfSF                        -0.136637           0.148952   
0.180605   
Total_Basement_Area              -0.171237           0.391550   
0.291477   
First_Floor_Area                 -0.144276           0.282030   
0.240620   
Second_Floor_Area                 0.029158           0.010197   
0.139574   
LowQualFinSF                      0.025527          -0.183805      -
0.062519   
Grade_Living_Area                -0.079567           0.198959   
0.287178   
Underground_Full_Bathroom        -0.055257           0.187838   
0.120290   
Underground_Half_Bathroom         0.117892          -0.038197      -
0.012500   
Full_Bathroom_Above_Grade        -0.193961           0.468301   
0.438667   
Half_Bathroom_Above_Grade        -0.061125           0.242960   
0.184294   
Bedroom_Above_Grade               0.012938          -0.070630      -
0.040486   
Kitchen_Above_Grade              -0.086951          -0.174836      -
0.149787   
Rooms_Above_Grade                -0.057505           0.095549   
0.191597   
Fireplaces                       -0.023580           0.147629   
0.112024   
```

```
Garage_Built_Year                  -0.299097          0.700110
0.571223
Garage_Size                        -0.185565          0.537906
0.420230
Garage_Area                        -0.009511          0.023549
0.010012
W_Deck_Area                         0.045563          0.007441
0.045620
Open_Lobby_Area                     0.004575         -0.045182     -
0.061111
Enclosed_Lobby_Area                -0.042148         -0.006849     -
0.010810
Three_Season_Lobby_Area             0.025535          0.031339
0.045224
Screen_Lobby_Area                   0.054885         -0.050405     -
0.038932
Pool_Area                          -0.001967          0.004940
0.005786
Miscellaneous_Value                 0.068803         -0.034396     -
0.010347
Month_Sold                         -0.003480          0.012382
0.021418
Year_Sold                           0.043916         -0.013598
0.035846

                           Brick_Veneer_Area  BsmtFinSF1
BsmtFinSF2  \
Building_Class                      0.022559   -0.069364   -0.064813

Lot_Size                            0.103950    0.214190    0.111317

Overall_Material                    0.410062    0.240239   -0.058354

House_Condition                    -0.127660   -0.046466    0.039867

Construction_Year                   0.314706    0.249689   -0.048931

Remodel_Year                        0.178886    0.129082   -0.066836

Brick_Veneer_Area                   1.000000    0.264012   -0.071773

BsmtFinSF1                          0.264012    1.000000   -0.051047

BsmtFinSF2                         -0.071773   -0.051047    1.000000

BsmtUnfSF                           0.113850   -0.494968   -0.208515

Total_Basement_Area                 0.362698    0.522298    0.104430

First_Floor_Area                    0.342302    0.445841    0.096945
```

```
Second_Floor_Area                  0.173764  -0.136680  -0.098536
LowQualFinSF                      -0.069124  -0.064449   0.014943
Grade_Living_Area                  0.389776   0.208527  -0.009137
Underground_Full_Bathroom          0.085538   0.649001   0.157721
Underground_Half_Bathroom          0.026577   0.067576   0.071255
Full_Bathroom_Above_Grade          0.275458   0.059175  -0.075468
Half_Bathroom_Above_Grade          0.201414   0.003552  -0.033461
Bedroom_Above_Grade                0.102494  -0.107477  -0.015910
Kitchen_Above_Grade               -0.037451  -0.080905  -0.040565
Rooms_Above_Grade                  0.279943   0.044513  -0.034925
Fireplaces                         0.247637   0.260708   0.047957
Garage_Built_Year                  0.211982   0.119663  -0.093667
Garage_Size                        0.363547   0.224786  -0.037244
Garage_Area                        0.020672   0.027699  -0.001745
W_Deck_Area                        0.014964   0.036518   0.006073
Open_Lobby_Area                    0.001159   0.038410  -0.024431
Enclosed_Lobby_Area                0.013640   0.007652  -0.016625
Three_Season_Lobby_Area            0.018751   0.026525  -0.029897
Screen_Lobby_Area                  0.061355   0.062194   0.089223
Pool_Area                          0.011697   0.140566   0.041813
Miscellaneous_Value               -0.029853   0.003623   0.005034
Month_Sold                        -0.005987  -0.015662  -0.015099
Year_Sold                         -0.008130   0.014282   0.031587

                           BsmtUnfSF  ...  Garage_Area  W_Deck_Area  \
Building_Class             -0.141426  ...    -0.023996    -0.000923
Lot_Size                   -0.002658  ...     0.006646     0.030025
Overall_Material            0.307794  ...     0.019580     0.034516
```

House_Condition	-0.136637	...	-0.009511	0.045563
Construction_Year	0.148952	...	0.023549	0.007441
Remodel_Year	0.180605	...	0.010012	0.045620
Brick_Veneer_Area	0.113850	...	0.020672	0.014964
BsmtFinSF1	-0.494968	...	0.027699	0.036518
BsmtFinSF2	-0.208515	...	-0.001745	0.006073
BsmtUnfSF	1.000000	...	-0.003586	-0.010075
Total_Basement_Area	0.415828	...	0.024540	0.030046
First_Floor_Area	0.318259	...	-0.001461	0.046259
Second_Floor_Area	0.003939	...	-0.037720	0.005865
LowQualFinSF	0.028096	...	-0.053175	0.004136
Grade_Living_Area	0.240025	...	-0.037327	0.039288
Underground_Full_Bathroom	-0.422475	...	0.020634	0.021359
Underground_Half_Bathroom	-0.095999	...	0.047924	0.007716
Full_Bathroom_Above_Grade	0.288398	...	-0.033428	0.023177
Half_Bathroom_Above_Grade	-0.040330	...	-0.007799	0.023130
Bedroom_Above_Grade	0.166809	...	-0.032253	0.009506
Kitchen_Above_Grade	0.029955	...	-0.028872	-0.021768
Rooms_Above_Grade	0.250524	...	-0.029437	0.027160
Fireplaces	0.050971	...	0.009095	0.040411
Garage_Built_Year	0.172023	...	-0.005049	0.004376
Garage_Size	0.213635	...	0.012685	0.007406
Garage_Area	-0.003586	...	1.000000	-0.013644
W_Deck_Area	-0.010075	...	-0.013644	1.000000
Open_Lobby_Area	-0.036021	...	0.029191	-0.029247
Enclosed_Lobby_Area	0.007724	...	-0.003719	-0.054954
Three_Season_Lobby_Area	0.020693	...	0.010523	-0.020934
Screen_Lobby_Area	-0.012765	...	0.017073	-0.027602
Pool_Area	-0.035150	...	0.009449	-0.004103
Miscellaneous_Value	-0.023903	...	-0.011799	-0.000344
Month_Sold	0.034820	...	0.023323	0.025610
Year_Sold	-0.041179	...	0.008717	0.033614

	Open_Lobby_Area	Enclosed_Lobby_Area \
Building_Class	-0.024260	0.030441
Lot_Size	0.032699	0.003951
Overall_Material	-0.036792	0.018161
House_Condition	0.004575	-0.042148
Construction_Year	-0.045182	-0.006849
Remodel_Year	-0.061111	-0.010810
Brick_Veneer_Area	0.001159	0.013640
BsmtFinSF1	0.038410	0.007652
BsmtFinSF2	-0.024431	-0.016625
BsmtUnfSF	-0.036021	0.007724
Total_Basement_Area	-0.005323	0.009623
First_Floor_Area	0.030272	0.025440
Second_Floor_Area	-0.037295	-0.005074
LowQualFinSF	-0.031299	-0.007902
Grade_Living_Area	-0.011602	0.013771

Underground_Full_Bathroom	-0.003263	0.025902
Underground_Half_Bathroom	0.011794	-0.032112
Full_Bathroom_Above_Grade	-0.007260	0.001873
Half_Bathroom_Above_Grade	-0.024593	-0.008560
Bedroom_Above_Grade	0.001606	-0.005546
Kitchen_Above_Grade	0.023516	0.003137
Rooms_Above_Grade	-0.002276	-0.007173
Fireplaces	0.042723	0.049195
Garage_Built_Year	-0.042241	-0.000272
Garage_Size	-0.014276	0.021136
Garage_Area	0.029191	-0.003719
W_Deck_Area	-0.029247	-0.054954
Open_Lobby_Area	1.000000	0.006178
Enclosed_Lobby_Area	0.006178	1.000000
Three_Season_Lobby_Area	-0.015663	0.032717
Screen_Lobby_Area	0.066338	0.051263
Pool_Area	0.008181	0.013138
Miscellaneous_Value	-0.020327	-0.025895
Month_Sold	-0.027465	-0.027018
Year_Sold	0.021557	-0.019800

\	Three_Season_Lobby_Area	Screen_Lobby_Area
Building_Class	-0.043906	-0.026199
Lot_Size	0.020418	0.043151
Overall_Material	0.030314	0.064755
House_Condition	0.025535	0.054885
Construction_Year	0.031339	-0.050405
Remodel_Year	0.045224	-0.038932
Brick_Veneer_Area	0.018751	0.061355
BsmtFinSF1	0.026525	0.062194
BsmtFinSF2	-0.029897	0.089223
BsmtUnfSF	0.020693	-0.012765
Total_Basement_Area	0.037423	0.084581
First_Floor_Area	0.056125	0.088807
Second_Floor_Area	-0.024426	0.040469
LowQualFinSF	-0.004305	0.026778

Grade_Living_Area	0.020606	0.101430
Underground_Full_Bathroom	-0.000018	0.023363
Underground_Half_Bathroom	0.035095	0.032078
Full_Bathroom_Above_Grade	0.035284	-0.008299
Half_Bathroom_Above_Grade	-0.004877	0.072693
Bedroom_Above_Grade	-0.024465	0.044332
Kitchen_Above_Grade	-0.024618	-0.051655
Rooms_Above_Grade	-0.006709	0.059327
Fireplaces	0.011185	0.184416
Garage_Built_Year	0.015858	-0.089138
Garage_Size	0.035696	0.050323
Garage_Area	0.010523	0.017073
W_Deck_Area	-0.020934	-0.027602
Open_Lobby_Area	-0.015663	0.066338
Enclosed_Lobby_Area	0.032717	0.051263
Three_Season_Lobby_Area	1.000000	-0.031458
Screen_Lobby_Area	-0.031458	1.000000
Pool_Area	-0.007997	0.051296
Miscellaneous_Value	0.000347	0.031930
Month_Sold	0.029465	0.023196
Year_Sold	0.018656	0.010720

	Pool_Area	Miscellaneous_Value	Month_Sold \
Building_Class	0.008244	-0.007738	-0.013660
Lot_Size	0.077670	0.038064	0.001200
Overall_Material	0.065143	-0.031461	0.070766
House_Condition	-0.001967	0.068803	-0.003480

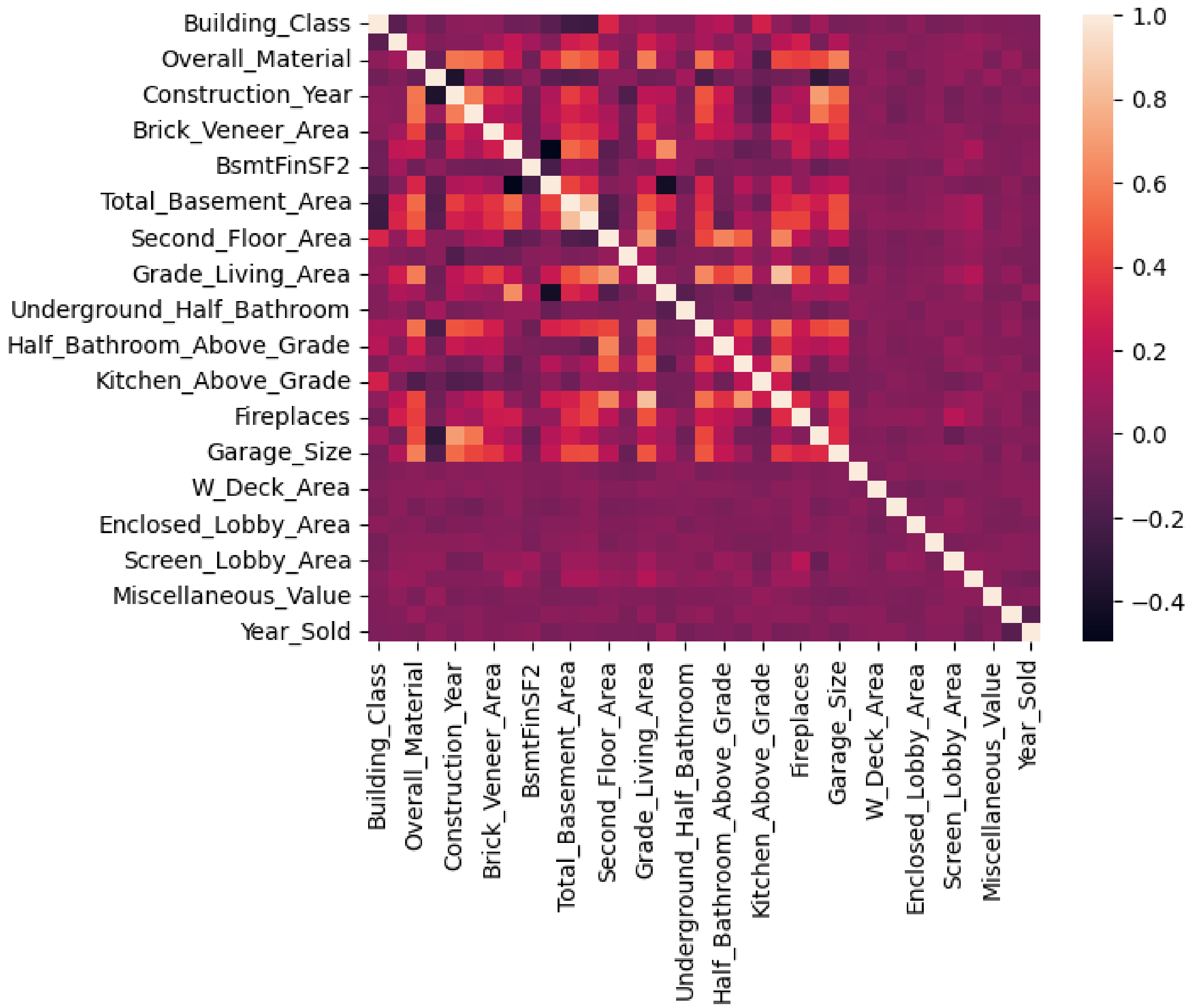
Construction_Year	0.004940	-0.034396	0.012382
Remodel_Year	0.005786	-0.010347	0.021418
Brick_Veneer_Area	0.011697	-0.029853	-0.005987
BsmtFinSF1	0.140566	0.003623	-0.015662
BsmtFinSF2	0.041813	0.005034	-0.015099
BsmtUnfSF	-0.035150	-0.023903	0.034820
Total_Basement_Area	0.126084	-0.018453	0.013234
First_Floor_Area	0.131539	-0.021082	0.031392
Second_Floor_Area	0.081467	0.016153	0.035107
LowQualFinSF	0.062152	-0.003800	-0.022184
Grade_Living_Area	0.170197	-0.002446	0.050204
Underground_Full_Bathroom	0.067696	-0.022990	-0.025281
Underground_Half_Bathroom	0.020014	-0.007381	0.032854
Full_Bathroom_Above_Grade	0.049573	-0.014357	0.055809
Half_Bathroom_Above_Grade	0.022451	0.001365	-0.008954
Bedroom_Above_Grade	0.070711	0.007777	0.046558
Kitchen_Above_Grade	-0.014535	0.062329	0.026572
Rooms_Above_Grade	0.083745	0.024745	0.036883
Fireplaces	0.095058	0.001352	0.046294
Garage_Built_Year	-0.018168	-0.029698	0.000332
Garage_Size	0.020893	-0.043158	0.040453
Garage_Area	0.009449	-0.011799	0.023323
W_Deck_Area	-0.004103	-0.000344	0.025610
Open_Lobby_Area	0.008181	-0.020327	-0.027465
Enclosed_Lobby_Area	0.013138	-0.025895	-0.027018
Three_Season_Lobby_Area	-0.007997	0.000347	0.029465

Screen_Lobby_Area	0.051296	0.031930	0.023196
Pool_Area	1.000000	0.029665	-0.033742
Miscellaneous_Value	0.029665	1.000000	-0.006502
Month_Sold	-0.033742	-0.006502	1.000000
Year_Sold	-0.059683	0.004915	-0.145712

	Year_Sold
Building_Class	-0.021330
Lot_Size	-0.014256
Overall_Material	-0.027277
House_Condition	0.043916
Construction_Year	-0.013598
Remodel_Year	0.035846
Brick_Veneer_Area	-0.008130
BsmtFinSF1	0.014282
BsmtFinSF2	0.031587
BsmtUnfSF	-0.041179
Total_Basement_Area	-0.015013
First_Floor_Area	-0.013627
Second_Floor_Area	-0.028631
LowQualFinSF	-0.028910
Grade_Living_Area	-0.036482
Underground_Full_Bathroom	0.066972
Underground_Half_Bathroom	-0.046502
Full_Bathroom_Above_Grade	-0.019578
Half_Bathroom_Above_Grade	-0.010391
Bedroom_Above_Grade	-0.036030
Kitchen_Above_Grade	0.031708
Rooms_Above_Grade	-0.034487
Fireplaces	-0.024013
Garage_Built_Year	0.001821
Garage_Size	-0.039034
Garage_Area	0.008717
W_Deck_Area	0.033614
Open_Lobby_Area	0.021557
Enclosed_Lobby_Area	-0.019800
Three_Season_Lobby_Area	0.018656
Screen_Lobby_Area	0.010720
Pool_Area	-0.059683
Miscellaneous_Value	0.004915
Month_Sold	-0.145712
Year_Sold	1.000000

[35 rows x 35 columns]

```
sns.heatmap(correlation_matrix)
plt.show()
```




```
threshold = 0.5
mask = (correlation_matrix.abs() > threshold) &
(correlation_matrix.abs() < 1)

columns_to_drop = set()
for col in mask.columns:
    if mask[col].any():
        columns_to_drop.add(col)

df_filtered = df.drop(columns=columns_to_drop)

columns_to_drop

{'Bedroom_Above_Grade',
 'BsmtFinSF1',
```

```
 'Construction_Year',
 'First_Floor_Area',
 'Full_Bathroom_Above_Grade',
 'Garage_Built_Year',
 'Garage_Size',
 'Grade_Living_Area',
 'Half_Bathroom_Above_Grade',
 'Overall_Material',
 'Remodel_Year',
 'Rooms_Above_Grade',
 'Second_Floor_Area',
 'Total_Basement_Area',
 'Underground_Full_Bathroom'}
```

```
df_filtered
```

```
      Building_Class Zoning_Class  Lot_Size Road_Type
Property_Shape  \
0                 60          RLD      8450     Paved            Reg

1                 20          RLD      9600     Paved            Reg

2                 60          RLD     11250     Paved            IR1

3                 70          RLD      9550     Paved            IR1

4                 60          RLD     14260     Paved            IR1

...              ...          ...       ...       ...            ...

1454              20          FVR      7500     Paved            Reg

1455              60          RLD      7917     Paved            Reg

1456              20          RLD     13175     Paved            Reg

1457              70          RLD      9042     Paved            Reg

1458              20          RLD      9717     Paved            Reg


     Land_Outline Utility_Type Lot_Configuration Property_Slope
Neighborhood  \
0             Lvl       AllPub                 I             GS
CollgCr
1             Lvl       AllPub              FR2P             GS
Veenker
2             Lvl       AllPub                 I             GS
CollgCr
3             Lvl       AllPub                 C             GS
Crawfor
```

```
4              Lvl       AllPub              FR2P            GS
NoRidge
...            ...          ...               ...           ...
...
1454           Lvl       AllPub                 I            GS
Somerst
1455           Lvl       AllPub                 I            GS
Gilbert
1456           Lvl       AllPub                 I            GS
NWAmes
1457           Lvl       AllPub                 I            GS
Crawfor
1458           Lvl       AllPub                 I            GS
NAmes

      ... Enclosed_Lobby_Area Three_Season_Lobby_Area
Screen_Lobby_Area  \
0     ...          20.337934                        0
0
1     ...          15.039392                        0
0
2     ...         -46.232198                        0
0
3     ...          60.921821                        0
0
4     ...          21.788818                        0
0
...   ...                ...                      ...                 ..
.
1454  ...         126.676547                        0
0
1455  ...         125.521880                        0
0
1456  ...         148.266666                        0
0
1457  ...          54.320896                        0
0
1458  ...          19.498763                        0
0

     Pool_Area  Miscellaneous_Value Month_Sold Year_Sold Sale_Type  \
0            0                    0          2      2008        WD
1            0                    0          5      2007        WD
2            0                    0          9      2008        WD
3            0                    0          2      2006        WD
4            0                    0         12      2008        WD
...        ...                  ...        ...       ...       ...
1454         0                    0         10      2009        WD
1455         0                    0          8      2007        WD
```

```
1456        0                  0          2      2010        WD
1457        0               2500          5      2010        WD
1458        0                  0          4      2010        WD

     Sale_Condition  Sale_Price
0            Normal      208500
1            Normal      181500
2            Normal      223500
3           Abnorml      140000
4            Normal      250000
...             ...         ...
1454         Normal      185000
1455         Normal      175000
1456         Normal      210000
1457         Normal      266500
1458         Normal      142125

[1459 rows x 58 columns]
```

```
new_df.shape
```

```
(1459, 67)
```

```
columns_to_be_kept
```

```
['Building_Class',
 'Lot_Size',
 'Brick_Veneer_Area',
 'BsmtFinSF2',
 'LowQualFinSF',
 'Underground_Full_Bathroom',
 'Underground_Half_Bathroom',
 'Half_Bathroom_Above_Grade',
 'Fireplaces',
 'W_Deck_Area',
 'Open_Lobby_Area',
 'Enclosed_Lobby_Area',
 'Three_Season_Lobby_Area',
 'Screen_Lobby_Area',
 'Pool_Area',
 'Miscellaneous_Value']
```

```
new_df[columns_to_be_kept]
```

```
      Building_Class  Lot_Size  Brick_Veneer_Area  BsmtFinSF2
LowQualFinSF  \
0                 60      8450              196.0           0
0
1                 20      9600                0.0           0
0
2                 60     11250              162.0           0
```

```
0
3                70     9550                   0.0           0
0
4                60    14260                 350.0           0
0
...             ...      ...                   ...         ...
...
1454             20     7500                   0.0           0
0
1455             60     7917                   0.0           0
0
1456             20    13175                 119.0         163
0
1457             70     9042                   0.0           0
0
1458             20     9717                   0.0        1029
0

      Underground_Full_Bathroom  Underground_Half_Bathroom  \
0                             1                          0
1                             0                          1
2                             1                          0
3                             1                          0
4                             1                          0
...                         ...                        ...
1454                          1                          0
1455                          0                          0
1456                          1                          0
1457                          0                          0
1458                          1                          0

      Half_Bathroom_Above_Grade  Fireplaces  W_Deck_Area
Open_Lobby_Area  \
0                             1           0   163.788080
69.596115
1                             0           1   198.900074
74.716033
2                             1           1    26.127533
32.085268
3                             0           1    46.948018
40.181415
4                             1           1   -10.626105
20.755323
...                         ...         ...          ...
...
1454                          0           0    -9.973961        -
9.267967
1455                          1           1   -80.348891
113.043436
```

```
1456                          0           2    36.180338
221.514480
1457                          0           2    88.568242
110.888690
1458                          0           0   144.036562        -
33.654857

      Enclosed_Lobby_Area  Three_Season_Lobby_Area  Screen_Lobby_Area
\
0                20.337934                        0                  0

1                15.039392                        0                  0

2               -46.232198                        0                  0

3                60.921821                        0                  0

4                21.788818                        0                  0

...                    ...                      ...                ...

1454            126.676547                        0                  0

1455            125.521880                        0                  0

1456            148.266666                        0                  0

1457             54.320896                        0                  0

1458             19.498763                        0                  0


      Pool_Area  Miscellaneous_Value
0             0                    0
1             0                    0
2             0                    0
3             0                    0
4             0                    0
...         ...                  ...
1454          0                    0
1455          0                    0
1456          0                    0
1457          0                 2500
1458          0                    0

[1459 rows x 16 columns]

new_df.head()

   Building_Class Zoning_Class  Lot_Size Property_Shape
Land_Outline  \
```

```
0              60          RLD      8450            Reg          Lvl

1              20          RLD      9600            Reg          Lvl

2              60          RLD     11250            IR1          Lvl

3              70          RLD      9550            IR1          Lvl

4              60          RLD     14260            IR1          Lvl


  Lot_Configuration Neighborhood Condition1 House_Type
House_Design  ...  \
0                 I      CollgCr       Norm       1Fam
2Story  ...
1              FR2P      Veenker      Feedr       1Fam
1Story  ...
2                 I      CollgCr       Norm       1Fam
2Story  ...
3                 C      Crawfor       Norm       1Fam
2Story  ...
4              FR2P      NoRidge       Norm       1Fam
2Story  ...

   Enclosed_Lobby_Area  Three_Season_Lobby_Area  Screen_Lobby_Area
Pool_Area  \
0            20.337934                        0                  0
0
1            15.039392                        0                  0
0
2           -46.232198                        0                  0
0
3            60.921821                        0                  0
0
4            21.788818                        0                  0
0

  Miscellaneous_Value Month_Sold Year_Sold Sale_Type
Sale_Condition  \
0                   0          2      2008        WD         Normal

1                   0          5      2007        WD         Normal

2                   0          9      2008        WD         Normal

3                   0          2      2006        WD        Abnorml

4                   0         12      2008        WD         Normal


  Sale_Price
```

```
0     208500
1     181500
2     223500
3     140000
4     250000

[5 rows x 67 columns]
```

```
columns_to_remove
```

```
['Overall_Material',
 'House_Condition',
 'Construction_Year',
 'Remodel_Year',
 'BsmtFinSF1',
 'BsmtUnfSF',
 'Total_Basement_Area',
 'First_Floor_Area',
 'Second_Floor_Area',
 'Grade_Living_Area',
 'Full_Bathroom_Above_Grade',
 'Bedroom_Above_Grade',
 'Kitchen_Above_Grade',
 'Rooms_Above_Grade',
 'Garage_Built_Year',
 'Garage_Size',
 'Garage_Area',
 'Month_Sold',
 'Year_Sold']
```

```
new_df.drop(columns=columns_to_remove)
```

```
      Building_Class Zoning_Class  Lot_Size Property_Shape
Land_Outline  \
0                 60          RLD      8450            Reg
Lvl
1                 20          RLD      9600            Reg
Lvl
2                 60          RLD     11250            IR1
Lvl
3                 70          RLD      9550            IR1
Lvl
4                 60          RLD     14260            IR1
Lvl
...              ...          ...       ...            ...            ..
.
1454              20          FVR      7500            Reg
Lvl
1455              60          RLD      7917            Reg
Lvl
```

```
1456              20          RLD     13175            Reg
Lvl
1457              70          RLD      9042            Reg
Lvl
1458              20          RLD      9717            Reg
Lvl

     Lot_Configuration Neighborhood Condition1 House_Type House_Design
...  \
0                    I      CollgCr       Norm       1Fam       2Story
...
1                 FR2P      Veenker      Feedr       1Fam       1Story
...
2                    I      CollgCr       Norm       1Fam       2Story
...
3                    C      Crawfor       Norm       1Fam       2Story
...
4                 FR2P      NoRidge       Norm       1Fam       2Story
...
...                ...          ...        ...        ...          ...
...
1454                 I      Somerst       Norm       1Fam       1Story
...
1455                 I      Gilbert       Norm       1Fam       2Story
...
1456                 I       NWAmes       Norm       1Fam       1Story
...
1457                 I      Crawfor       Norm       1Fam       2Story
...
1458                 I        NAmes       Norm       1Fam       1Story
...

     W_Deck_Area Open_Lobby_Area Enclosed_Lobby_Area
Three_Season_Lobby_Area  \
0     163.788080       69.596115           20.337934
0
1     198.900074       74.716033           15.039392
0
2      26.127533       32.085268          -46.232198
0
3      46.948018       40.181415           60.921821
0
4     -10.626105       20.755323           21.788818
0
...          ...             ...                 ...
...
1454   -9.973961       -9.267967          126.676547
0
1455  -80.348891      113.043436          125.521880
```

```
0
1456   36.180338      221.514480         148.266666
0
1457   88.568242      110.888690          54.320896
0
1458  144.036562      -33.654857          19.498763
0

      Screen_Lobby_Area Pool_Area Miscellaneous_Value Sale_Type  \
0                     0         0                   0        WD
1                     0         0                   0        WD
2                     0         0                   0        WD
3                     0         0                   0        WD
4                     0         0                   0        WD
...                 ...       ...                 ...       ...
1454                  0         0                   0        WD
1455                  0         0                   0        WD
1456                  0         0                   0        WD
1457                  0         0                2500        WD
1458                  0         0                   0        WD

     Sale_Condition Sale_Price
0            Normal     208500
1            Normal     181500
2            Normal     223500
3           Abnorml     140000
4            Normal     250000
...             ...        ...
1454         Normal     185000
1455         Normal     175000
1456         Normal     210000
1457         Normal     266500
1458         Normal     142125

[1459 rows x 48 columns]
```

```
new_df.shape
```

```
(1459, 67)
```

```
ll = df[columns_to_be_kept]
```

```
ll["Sale_Price"] = new_df["Sale_Price"]
```

```
C:\Users\kumar\AppData\Local\Temp\ipykernel_24672\2298138157.py:1:
SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy


ll

      Building_Class  Lot_Size  Brick_Veneer_Area  BsmtFinSF2
LowQualFinSF  \
0                 60      8450              196.0           0
0
1                 20      9600                0.0           0
0
2                 60     11250              162.0           0
0
3                 70      9550                0.0           0
0
4                 60     14260              350.0           0
0
...              ...       ...                ...         ...
...
1454              20      7500                0.0           0
0
1455              60      7917                0.0           0
0
1456              20     13175              119.0         163
0
1457              70      9042                0.0           0
0
1458              20      9717                0.0        1029
0

      Underground_Full_Bathroom  Underground_Half_Bathroom  \
0                             1                          0
1                             0                          1
2                             1                          0
3                             1                          0
4                             1                          0
...                         ...                        ...
1454                          1                          0
1455                          0                          0
1456                          1                          0
1457                          0                          0
1458                          1                          0

      Half_Bathroom_Above_Grade  Fireplaces  W_Deck_Area
Open_Lobby_Area  \
0                             1           0   163.788080
69.596115
1                             0           1   198.900074
```

```
74.716033
2                               1            1    26.127533
32.085268
3                               0            1    46.948018
40.181415
4                               1            1   -10.626105
20.755323
...                           ...          ...          ...
...
1454                            0            0    -9.973961         -
9.267967
1455                            1            1   -80.348891
113.043436
1456                            0            2    36.180338
221.514480
1457                            0            2    88.568242
110.888690
1458                            0            0   144.036562        -
33.654857

      Enclosed_Lobby_Area  Three_Season_Lobby_Area  Screen_Lobby_Area
\
0               20.337934                        0                  0

1               15.039392                        0                  0

2              -46.232198                        0                  0

3               60.921821                        0                  0

4               21.788818                        0                  0

...                   ...                      ...                ...

1454           126.676547                        0                  0

1455           125.521880                        0                  0

1456           148.266666                        0                  0

1457            54.320896                        0                  0

1458            19.498763                        0                  0


      Pool_Area  Miscellaneous_Value  Sale_Price
0             0                    0      208500
1             0                    0      181500
2             0                    0      223500
3             0                    0      140000
```

```
4               0                     0      250000
...           ...                   ...         ...
1454            0                     0      185000
1455            0                     0      175000
1456            0                     0      210000
1457            0                  2500      266500
1458            0                     0      142125

[1459 rows x 17 columns]
```

```
co_matrix = ll.corr()

correlations_with_target = co_matrix['Sale_Price'].drop('Sale_Price')

correlations_with_target
```

```
Building_Class               -0.084563
Lot_Size                      0.263843
Brick_Veneer_Area             0.475160
BsmtFinSF2                   -0.010952
LowQualFinSF                 -0.025642
Underground_Full_Bathroom     0.227551
Underground_Half_Bathroom    -0.016915
Half_Bathroom_Above_Grade     0.284626
Fireplaces                    0.466828
W_Deck_Area                   0.042814
Open_Lobby_Area              -0.010131
Enclosed_Lobby_Area           0.020789
Three_Season_Lobby_Area       0.044553
Screen_Lobby_Area             0.111378
Pool_Area                     0.092389
Miscellaneous_Value          -0.021216
Name: Sale_Price, dtype: float64
```

```
correlation_threshold = 0.3
features_to_keep =
correlations_with_target[abs(correlations_with_target) >
correlation_threshold].index.tolist()

features_to_keep
```

```
['Brick_Veneer_Area', 'Fireplaces']
```

```
categorical_columns
```

```
Index(['Zoning_Class', 'Road_Type', 'Property_Shape', 'Land_Outline',
       'Utility_Type', 'Lot_Configuration', 'Property_Slope',
'Neighborhood',
       'Condition1', 'Condition2', 'House_Type', 'House_Design',
'Roof_Design',
       'Roof_Quality', 'Exterior1st', 'Exterior2nd',
```

```
'Exterior_Material',
       'Exterior_Condition', 'Foundation_Type', 'Basement_Height',
       'Basement_Condition', 'Exposure_Level', 'BsmtFinType1',
'BsmtFinType2',
       'Heating_Type', 'Heating_Quality', 'Air_Conditioning',
       'Electrical_System', 'Kitchen_Quality', 'Functional_Rate',
'Garage',
       'Garage_Finish_Year', 'Garage_Quality', 'Garage_Condition',
       'Pavedd_Drive', 'Sale_Type', 'Sale_Condition'],
      dtype='object')
```

```
cat_columns = new_df.select_dtypes(include="object").columns.tolist()
```

```
from scipy.stats import f_oneway
anova_results = {}
for feature in new_df[cat_columns].columns:  # Exclude the target
column
    categories = new_df[feature].unique()
    groups = [new_df[new_df[feature] == category]['Sale_Price'] for
category in categories]
    anova_results[feature] = f_oneway(*groups)

# Collect p-values
p_values = {feature: result.pvalue for feature, result in
anova_results.items()}

# Select features with p-value below significance threshold (e.g.,
0.05)
significance_threshold = 0.05
selected_features = [feature for feature, p_value in p_values.items()
if p_value < significance_threshold]

# Display selected features and their p-values
selected_features_pvalues = {feature: p_values[feature] for feature in
selected_features}
print("Selected features based on ANOVA p-values:",
selected_features_pvalues)
```

```
Selected features based on ANOVA p-values: {'Zoning_Class':
8.623678334677968e-35, 'Property_Shape': 7.137696914217586e-25,
'Land_Outline': 2.7817438435761282e-08, 'Lot_Configuration':
3.250291720935249e-06, 'Neighborhood': 2.3465866522923626e-225,
'Condition1': 8.850238347455908e-08, 'House_Type':
2.0291429199969593e-10, 'House_Design': 3.588437772489042e-25,
'Roof_Design': 3.908164572338989e-17, 'Roof_Quality':
7.356309273248755e-08, 'Exterior1st': 2.977413814634284e-43,
'Exterior2nd': 5.494558715253583e-43, 'Exterior_Material':
7.440829120284835e-205, 'Exterior_Condition': 5.068975761586581e-07,
'Foundation_Type': 7.320259470842929e-91, 'Basement_Height':
3.0551393861901977e-194, 'Basement_Condition': 7.225349255486377e-09,
```

```
'Exposure_Level': 6.190506171729479e-46, 'BsmtFinType1':
4.371412794979286e-66, 'Heating_Quality': 3.216320938669905e-67,
'Air_Conditioning': 1.7996181382228056e-22, 'Electrical_System':
1.6238358620842179e-18, 'Kitchen_Quality': 4.454356363811786e-192,
'Functional_Rate': 3.772707386918715e-08, 'Garage': 1.27964735338011e-
55, 'Garage_Finish_Year': 2.9752698638318936e-111, 'Garage_Quality':
9.589593109138726e-07, 'Garage_Condition': 1.5286359731104103e-06,
'Pavedd_Drive': 1.784823505685642e-18, 'Sale_Type':
5.5577569033580856e-42, 'Sale_Condition': 8.737390258964316e-44}
```

```
selected_features_pvalues
```

```
{'Zoning_Class': 8.623678334677968e-35,
 'Property_Shape': 7.137696914217586e-25,
 'Land_Outline': 2.7817438435761282e-08,
 'Lot_Configuration': 3.250291720935249e-06,
 'Neighborhood': 2.3465866522923626e-225,
 'Condition1': 8.850238347455908e-08,
 'House_Type': 2.0291429199969593e-10,
 'House_Design': 3.588437772489042e-25,
 'Roof_Design': 3.908164572338989e-17,
 'Roof_Quality': 7.356309273248755e-08,
 'Exterior1st': 2.977413814634284e-43,
 'Exterior2nd': 5.494558715253583e-43,
 'Exterior_Material': 7.440829120284835e-205,
 'Exterior_Condition': 5.068975761586581e-07,
 'Foundation_Type': 7.320259470842929e-91,
 'Basement_Height': 3.0551393861901977e-194,
 'Basement_Condition': 7.225349255486377e-09,
 'Exposure_Level': 6.190506171729479e-46,
 'BsmtFinType1': 4.371412794979286e-66,
 'Heating_Quality': 3.216320938669905e-67,
 'Air_Conditioning': 1.7996181382228056e-22,
 'Electrical_System': 1.6238358620842179e-18,
 'Kitchen_Quality': 4.454356363811786e-192,
 'Functional_Rate': 3.772707386918715e-08,
 'Garage': 1.27964735338011e-55,
 'Garage_Finish_Year': 2.9752698638318936e-111,
 'Garage_Quality': 9.589593109138726e-07,
 'Garage_Condition': 1.5286359731104103e-06,
 'Pavedd_Drive': 1.784823505685642e-18,
 'Sale_Type': 5.5577569033580856e-42,
 'Sale_Condition': 8.737390258964316e-44}
```

```
significance_threshold = 0.05
features_to_keep = [feature for feature, p_value in p_values.items()
if p_value < significance_threshold]

len(features_to_keep)
```

```
31

# means we should keep all the above cat features

new_new_df = new_df.drop(columns=columns_to_remove)

new_new_df.shape

(1459, 48)

len(cat_columns)

31

new_df["Zoning_Class"].unique()

array(['RLD', 'RMD', 'Commer', 'FVR', 'RHD'], dtype=object)

# Ordinal features
# House_Design, House_Type, Exterior_Material, Exterior_Condition,
Basement_Height, Basement_Condition
# Exposure_Level, BsmtFinType1, Heating_Quality, Electrical_System,
Kitchen_Quality, Functional_Rate, Garage_Finish_Year
# Garage_Quality, Garage_Condition, Pavedd_Drive, Sale_Condition

new_df["House_Design"].unique()

array(['2Story', '1Story', '1.5Fin', '1.5Unf', 'SFoyer', 'SLvl',
'2.5Unf',
       '2.5Fin'], dtype=object)

categories = {
    'House_Design': ['1Story', '1.5Fin', '1.5Unf', '2Story', '2.5Fin',
'2.5Unf', 'SFoyer', 'SLvl'],
    'House_Type': ['1Fam', '2fmCon', 'Duplex', 'Twnhs', 'TwnhsE'],
    'Exterior_Material': ['Po', 'Fa', 'TA', 'Gd', 'Ex'],  # Reversed
order
    'Exterior_Condition': ['Po', 'Fa', 'TA', 'Gd', 'Ex'],  # Reversed
order
    'Basement_Height': ['NA', 'Po', 'Fa', 'TA', 'Gd', 'Ex'],  #
Reversed order
    'Basement_Condition': ['NA', 'Po', 'Fa', 'TA', 'Gd', 'Ex'],  #
Reversed order
    'Exposure_Level': ['NA', 'No', 'Mn', 'Av', 'Gd'],  # Reversed
order
    'BsmtFinType1': ['Unf', 'LwQ', 'Rec', 'BLQ', 'ALQ', 'GLQ'],  #
Reversed order
    'Heating_Quality': ['Po', 'Fa', 'TA', 'Gd', 'Ex'],  # Reversed
order
    'Electrical_System': ['Mix', 'FuseP', 'FuseF', 'FuseA', 'SBrkr'],
# Reversed order
    'Kitchen_Quality': ['Po', 'Fa', 'TA', 'Gd', 'Ex'],  # Reversed
```

```
order
    'Functional_Rate': ['S', 'SD', 'MajD2', 'MajD1', 'MD', "MS",
'MD2', 'MD1', 'TF'],  # Reversed order
    'Garage_Finish_Year': ['NA', 'Unf', 'RFn', 'Fin'],  # Reversed
order
    'Garage_Quality': ['NA', 'Po', 'Fa', 'TA', 'Gd', 'Ex'],  #
Reversed order
    'Garage_Condition': ['NA', 'Po', 'Fa', 'TA', 'Gd', 'Ex'],  #
Reversed order
    'Pavedd_Drive': ['N', 'P', 'Y'],  # Reversed order
    'Sale_Condition': ['Partial', 'Family', 'Alloca', 'AdjLand',
'Abnorml', 'Normal']  # Reversed order
}

categories.keys()
```

```
dict_keys(['House_Design', 'House_Type', 'Exterior_Material',
'Exterior_Condition', 'Basement_Height', 'Basement_Condition',
'Exposure_Level', 'BsmtFinType1', 'Heating_Quality',
'Electrical_System', 'Kitchen_Quality', 'Functional_Rate',
'Garage_Finish_Year', 'Garage_Quality', 'Garage_Condition',
'Pavedd_Drive', 'Sale_Condition'])
```

```
columns_to_encode = ['House_Design', 'House_Type',
'Exterior_Material', 'Exterior_Condition', 'Basement_Height',
                     'Basement_Condition', 'Exposure_Level',
'BsmtFinType1', 'Heating_Quality', 'Electrical_System',
'Kitchen_Quality',
                     'Functional_Rate', 'Garage_Finish_Year',
'Garage_Quality', 'Garage_Condition', 'Pavedd_Drive',
'Sale_Condition']

from sklearn.preprocessing import OrdinalEncoder
encoder = OrdinalEncoder(categories=[categories['House_Design'],
categories['House_Type'],
                                      categories['Exterior_Material'],
categories['Exterior_Condition'],
                                      categories['Basement_Height'],
categories['Basement_Condition'],
                                      categories['Exposure_Level'],
categories['BsmtFinType1'],
                                      categories['Heating_Quality'],
categories['Electrical_System'],
                                      categories['Kitchen_Quality'],
categories['Functional_Rate'],

categories['Garage_Finish_Year'], categories['Garage_Quality'],
                                      categories['Garage_Condition'],
categories['Pavedd_Drive'],
                                      categories['Sale_Condition']])
```

```
encoded_columns = encoder.fit_transform(new_new_df[columns_to_encode])
encoded_df = pd.DataFrame(encoded_columns, columns=columns_to_encode)

df[columns_to_encode] = encoded_df

df_encoded = pd.concat([new_new_df.drop(columns=columns_to_encode),
encoded_df], axis=1)

print(encoded_df)
```

```
      House_Design  House_Type  Exterior_Material  Exterior_Condition
\
0              3.0         0.0                3.0                 2.0

1              0.0         0.0                2.0                 2.0

2              3.0         0.0                3.0                 2.0

3              3.0         0.0                2.0                 2.0

4              3.0         0.0                3.0                 2.0

...            ...         ...                ...                 ...

1454           0.0         0.0                3.0                 2.0

1455           3.0         0.0                2.0                 2.0

1456           0.0         0.0                2.0                 2.0

1457           3.0         0.0                4.0                 3.0

1458           0.0         0.0                2.0                 2.0


      Basement_Height  Basement_Condition  Exposure_Level
BsmtFinType1  \
0                 4.0                 3.0             1.0
5.0
1                 4.0                 3.0             4.0
4.0
2                 4.0                 3.0             2.0
5.0
3                 3.0                 4.0             1.0
4.0
4                 4.0                 3.0             3.0
5.0
...               ...                 ...             ...            ..
.
1454              4.0                 3.0             1.0
```

```
5.0
1455              4.0                   3.0              1.0
0.0
1456              4.0                   3.0              1.0
4.0
1457              3.0                   4.0              1.0
5.0
1458              3.0                   3.0              2.0
5.0

      Heating_Quality  Electrical_System  Kitchen_Quality
Functional_Rate  \
0                 4.0                  4.0              3.0
8.0
1                 4.0                  4.0              2.0
8.0
2                 4.0                  4.0              3.0
8.0
3                 3.0                  4.0              3.0
8.0
4                 4.0                  4.0              3.0
8.0
...               ...                  ...              ...
...
1454              4.0                  4.0              3.0
8.0
1455              4.0                  4.0              2.0
8.0
1456              2.0                  4.0              2.0
7.0
1457              4.0                  4.0              3.0
8.0
1458              3.0                  3.0              3.0
8.0

      Garage_Finish_Year  Garage_Quality  Garage_Condition
Pavedd_Drive  \
0                    2.0             3.0               3.0
2.0
1                    2.0             3.0               3.0
2.0
2                    2.0             3.0               3.0
2.0
3                    1.0             3.0               3.0
2.0
4                    2.0             3.0               3.0
2.0
...                  ...             ...               ...            .
..
```

```
1454                 2.0             3.0               3.0
2.0
1455                 2.0             3.0               3.0
2.0
1456                 1.0             3.0               3.0
2.0
1457                 2.0             3.0               3.0
2.0
1458                 1.0             3.0               3.0
2.0

      Sale_Condition
0                5.0
1                5.0
2                5.0
3                4.0
4                5.0
...              ...
1454             5.0
1455             5.0
1456             5.0
1457             5.0
1458             5.0

[1459 rows x 17 columns]
```

```
new_new_df.head()
```

```
   Building_Class Zoning_Class  Lot_Size Property_Shape
Land_Outline  \
0              60          RLD      8450            Reg          Lvl

1              20          RLD      9600            Reg          Lvl

2              60          RLD     11250            IR1          Lvl

3              70          RLD      9550            IR1          Lvl

4              60          RLD     14260            IR1          Lvl


  Lot_Configuration Neighborhood Condition1 House_Type
House_Design  ...  \
0                 I      CollgCr       Norm       1Fam
2Story  ...
1              FR2P      Veenker      Feedr       1Fam
1Story  ...
2                 I      CollgCr       Norm       1Fam
2Story  ...
3                 C      Crawfor       Norm       1Fam
```

```
2Story  ...
4              FR2P      NoRidge       Norm       1Fam
2Story  ...

  W_Deck_Area Open_Lobby_Area Enclosed_Lobby_Area
Three_Season_Lobby_Area  \
0  163.788080       69.596115           20.337934
0
1  198.900074       74.716033           15.039392
0
2   26.127533       32.085268          -46.232198
0
3   46.948018       40.181415           60.921821
0
4  -10.626105       20.755323           21.788818
0

   Screen_Lobby_Area Pool_Area Miscellaneous_Value Sale_Type
Sale_Condition  \
0                  0         0                   0        WD
Normal
1                  0         0                   0        WD
Normal
2                  0         0                   0        WD
Normal
3                  0         0                   0        WD
Abnorml
4                  0         0                   0        WD
Normal

  Sale_Price
0     208500
1     181500
2     223500
3     140000
4     250000

[5 rows x 48 columns]
```

```
#new_new_df.to_csv("file_after_feature_selection.csv", index=False)

new_new_df
```

```
      Building_Class Zoning_Class  Lot_Size Property_Shape
Land_Outline  \
0                 60          RLD      8450            Reg
Lvl
1                 20          RLD      9600            Reg
Lvl
2                 60          RLD     11250            IR1
```

```
Lvl
3                 70          RLD      9550            IR1
Lvl
4                 60          RLD     14260            IR1
Lvl
...              ...          ...       ...            ...          ..
.
1454              20          FVR      7500            Reg
Lvl
1455              60          RLD      7917            Reg
Lvl
1456              20          RLD     13175            Reg
Lvl
1457              70          RLD      9042            Reg
Lvl
1458              20          RLD      9717            Reg
Lvl

     Lot_Configuration Neighborhood Condition1 House_Type House_Design
...  \
0                    I      CollgCr       Norm       1Fam       2Story
...
1                 FR2P      Veenker      Feedr       1Fam       1Story
...
2                    I      CollgCr       Norm       1Fam       2Story
...
3                    C      Crawfor       Norm       1Fam       2Story
...
4                 FR2P      NoRidge       Norm       1Fam       2Story
...
...                ...          ...        ...        ...          ...
...
1454                 I      Somerst       Norm       1Fam       1Story
...
1455                 I      Gilbert       Norm       1Fam       2Story
...
1456                 I       NWAmes       Norm       1Fam       1Story
...
1457                 I      Crawfor       Norm       1Fam       2Story
...
1458                 I        NAmes       Norm       1Fam       1Story
...

     W_Deck_Area Open_Lobby_Area Enclosed_Lobby_Area
Three_Season_Lobby_Area  \
0     163.788080       69.596115           20.337934
0
1     198.900074       74.716033           15.039392
0
```

```
2      26.127533      32.085268         -46.232198
0
3      46.948018      40.181415          60.921821
0
4     -10.626105      20.755323          21.788818
0
...          ...            ...                ...
...
1454   -9.973961      -9.267967         126.676547
0
1455  -80.348891     113.043436         125.521880
0
1456   36.180338     221.514480         148.266666
0
1457   88.568242     110.888690          54.320896
0
1458  144.036562     -33.654857          19.498763
0

      Screen_Lobby_Area Pool_Area Miscellaneous_Value Sale_Type  \
0                     0         0                   0        WD
1                     0         0                   0        WD
2                     0         0                   0        WD
3                     0         0                   0        WD
4                     0         0                   0        WD
...                 ...       ...                 ...       ...
1454                  0         0                   0        WD
1455                  0         0                   0        WD
1456                  0         0                   0        WD
1457                  0         0                2500        WD
1458                  0         0                   0        WD

     Sale_Condition Sale_Price
0            Normal     208500
1            Normal     181500
2            Normal     223500
3           Abnorml     140000
4            Normal     250000
...             ...        ...
1454         Normal     185000
1455         Normal     175000
1456         Normal     210000
1457         Normal     266500
1458         Normal     142125

[1459 rows x 48 columns]
```

After performing feature selection, the data shape is now 1459 rows × 48 columns. The removed columns included those identified by ANOVA and correlation tests, such as 'Overall_Material', 'House_Condition', and 'Garage_Size', among others, as well as categorical features like 'Road_Type', 'Utility_Type', and 'Heating_Type'.

```
new_new_df.select_dtypes(include=["float", "int"]).columns
```

```
Index(['Building_Class', 'Lot_Size', 'Brick_Veneer_Area',
'BsmtFinSF2',
       'LowQualFinSF', 'Underground_Full_Bathroom',
       'Underground_Half_Bathroom', 'Half_Bathroom_Above_Grade',
'Fireplaces',
       'W_Deck_Area', 'Open_Lobby_Area', 'Enclosed_Lobby_Area',
       'Three_Season_Lobby_Area', 'Screen_Lobby_Area', 'Pool_Area',
       'Miscellaneous_Value', 'Sale_Price'],
      dtype='object')
```

```
columns_to_normalize = ['Building_Class', 'Lot_Size',
'Brick_Veneer_Area', 'BsmtFinSF2',
       'LowQualFinSF', 'Underground_Full_Bathroom',
       'Underground_Half_Bathroom', 'Half_Bathroom_Above_Grade',
'Fireplaces',
       'W_Deck_Area', 'Open_Lobby_Area', 'Enclosed_Lobby_Area',
       'Three_Season_Lobby_Area', 'Screen_Lobby_Area', 'Pool_Area',
       'Miscellaneous_Value', 'Sale_Price']

nominal_features = ["Zoning_Class", "Property_Shape", "Land_Outline",
"Lot_Configuration", "Neighborhood", "Condition1",
                    "Roof_Design", "Roof_Quality", "Exterior1st",
"Exterior2nd", "Foundation_Type", "Garage", "Sale_Type"]

from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder(sparse_output=False, drop='first')
encoded_data = encoder.fit_transform(new_new_df[nominal_features])

encoded_df1 = pd.DataFrame(encoded_data,
columns=encoder.get_feature_names_out(nominal_features))

df_encoded = pd.concat([df_encoded.drop(columns=nominal_features),
encoded_df1], axis=1)

df_encoded.shape
```

```
(1459, 141)
```

```
encoded_df1.shape
```

```
(1459, 106)
```

```
df_encoded
```

```
      Building_Class  Lot_Size  Brick_Veneer_Area  BsmtFinSF2  \
0                 60      8450              196.0           0
1                 20      9600                0.0           0
2                 60     11250              162.0           0
3                 70      9550                0.0           0
4                 60     14260              350.0           0
...              ...       ...                ...         ...
1454              20      7500                0.0           0
1455              60      7917                0.0           0
1456              20     13175              119.0         163
1457              70      9042                0.0           0
1458              20      9717                0.0        1029

     Air_Conditioning  LowQualFinSF  Underground_Full_Bathroom  \
0                   Y             0                          1
1                   Y             0                          0
2                   Y             0                          1
3                   Y             0                          1
4                   Y             0                          1
...               ...           ...                        ...
1454                Y             0                          1
1455                Y             0                          0
1456                Y             0                          1
1457                Y             0                          0
1458                Y             0                          1

      Underground_Half_Bathroom  Half_Bathroom_Above_Grade  Fireplaces
...  \
0                             0                          1           0
...
1                             1                          0           1
...
2                             0                          1           1
...
3                             0                          0           1
...
4                             0                          1           1
...
...                         ...                        ...         ...
...
1454                          0                          0           0
...
1455                          0                          1           1
...
1456                          0                          0           2
...
1457                          0                          0           2
...
1458                          0                          0           0
...
```

```
      Garage_CarPort  Garage_Detchd  Sale_Type_CWD  Sale_Type_Con  \
0                0.0            0.0            0.0            0.0
1                0.0            0.0            0.0            0.0
2                0.0            0.0            0.0            0.0
3                0.0            1.0            0.0            0.0
4                0.0            0.0            0.0            0.0
...              ...            ...            ...            ...
1454             0.0            0.0            0.0            0.0
1455             0.0            0.0            0.0            0.0
1456             0.0            0.0            0.0            0.0
1457             0.0            0.0            0.0            0.0
1458             0.0            0.0            0.0            0.0

      Sale_Type_ConLD  Sale_Type_ConLI  Sale_Type_ConLw  Sale_Type_New
\
0                 0.0              0.0              0.0            0.0

1                 0.0              0.0              0.0            0.0

2                 0.0              0.0              0.0            0.0

3                 0.0              0.0              0.0            0.0

4                 0.0              0.0              0.0            0.0

...               ...              ...              ...            ...

1454              0.0              0.0              0.0            0.0

1455              0.0              0.0              0.0            0.0

1456              0.0              0.0              0.0            0.0

1457              0.0              0.0              0.0            0.0

1458              0.0              0.0              0.0            0.0


      Sale_Type_Oth  Sale_Type_WD
0               0.0           1.0
1               0.0           1.0
2               0.0           1.0
3               0.0           1.0
4               0.0           1.0
...             ...           ...
1454            0.0           1.0
1455            0.0           1.0
1456            0.0           1.0
1457            0.0           1.0
1458            0.0           1.0
```

```
[1459 rows x 141 columns]

new_new_df.shape

(1459, 48)

mapping_air_conditioning = {'Y': 1, 'N': 0}

df_encoded['Air_Conditioning_Binary'] =
df_encoded['Air_Conditioning'].map(mapping_air_conditioning)

df_encoded = df_encoded.drop(columns=['Air_Conditioning'])

df_encoded

      Building_Class  Lot_Size  Brick_Veneer_Area  BsmtFinSF2
LowQualFinSF  \
0                 60      8450              196.0           0
0
1                 20      9600                0.0           0
0
2                 60     11250              162.0           0
0
3                 70      9550                0.0           0
0
4                 60     14260              350.0           0
0
...              ...       ...                ...         ...
...
1454              20      7500                0.0           0
0
1455              60      7917                0.0           0
0
1456              20     13175              119.0         163
0
1457              70      9042                0.0           0
0
1458              20      9717                0.0        1029
0

      Underground_Full_Bathroom  Underground_Half_Bathroom  \
0                             1                          0
1                             0                          1
2                             1                          0
3                             1                          0
4                             1                          0
...                         ...                        ...
1454                          1                          0
1455                          0                          0
1456                          1                          0
```

```
1457                             0                           0
1458                             1                           0

      Half_Bathroom_Above_Grade  Fireplaces  W_Deck_Area  ...
Garage_Detchd  \
0                             1           0   163.788080  ...
0.0
1                             0           1   198.900074  ...
0.0
2                             1           1    26.127533  ...
0.0
3                             0           1    46.948018  ...
1.0
4                             1           1   -10.626105  ...
0.0
...                         ...         ...          ...  ...
...
1454                          0           0    -9.973961  ...
0.0
1455                          1           1   -80.348891  ...
0.0
1456                          0           2    36.180338  ...
0.0
1457                          0           2    88.568242  ...
0.0
1458                          0           0   144.036562  ...
0.0

      Sale_Type_CWD  Sale_Type_Con  Sale_Type_ConLD
Sale_Type_ConLI  \
0               0.0            0.0              0.0              0.0

1               0.0            0.0              0.0              0.0

2               0.0            0.0              0.0              0.0

3               0.0            0.0              0.0              0.0

4               0.0            0.0              0.0              0.0

...             ...            ...              ...              ...

1454            0.0            0.0              0.0              0.0

1455            0.0            0.0              0.0              0.0

1456            0.0            0.0              0.0              0.0

1457            0.0            0.0              0.0              0.0
```

```
1458            0.0            0.0                0.0                0.0


      Sale_Type_ConLw  Sale_Type_New  Sale_Type_Oth  Sale_Type_WD  \
0                 0.0            0.0            0.0           1.0
1                 0.0            0.0            0.0           1.0
2                 0.0            0.0            0.0           1.0
3                 0.0            0.0            0.0           1.0
4                 0.0            0.0            0.0           1.0
...               ...            ...            ...           ...
1454              0.0            0.0            0.0           1.0
1455              0.0            0.0            0.0           1.0
1456              0.0            0.0            0.0           1.0
1457              0.0            0.0            0.0           1.0
1458              0.0            0.0            0.0           1.0

      Air_Conditioning_Binary
0                           1
1                           1
2                           1
3                           1
4                           1
...                       ...
1454                        1
1455                        1
1456                        1
1457                        1
1458                        1

[1459 rows x 141 columns]
```

```
columns_to_normalize
```

```
['Building_Class',
 'Lot_Size',
 'Brick_Veneer_Area',
 'BsmtFinSF2',
 'LowQualFinSF',
 'Underground_Full_Bathroom',
 'Underground_Half_Bathroom',
 'Half_Bathroom_Above_Grade',
 'Fireplaces',
 'W_Deck_Area',
 'Open_Lobby_Area',
 'Enclosed_Lobby_Area',
 'Three_Season_Lobby_Area',
 'Screen_Lobby_Area',
 'Pool_Area',
 'Miscellaneous_Value',
 'Sale_Price']
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

df_encoded[columns_to_normalize] =
scaler.fit_transform(df_encoded[columns_to_normalize])

df_encoded
```

```
      Building_Class  Lot_Size  Brick_Veneer_Area  BsmtFinSF2
LowQualFinSF  \
0           0.072771 -0.207111           0.510905   -0.287744     -
0.120284
1          -0.873090 -0.091895          -0.574674   -0.287744     -
0.120284
2           0.072771  0.073415           0.322590   -0.287744     -
0.120284
3           0.309236 -0.096904          -0.574674   -0.287744     -
0.120284
4           0.072771  0.374980           1.363861   -0.287744     -
0.120284
...              ...       ...                ...         ...
...
1454       -0.873090 -0.302289          -0.574674   -0.287744     -
0.120284
1455        0.072771 -0.260511          -0.574674   -0.287744     -
0.120284
1456       -0.873090  0.266277           0.084428    0.723464     -
0.120284
1457        0.309236 -0.147800          -0.574674   -0.287744     -
0.120284
1458       -0.873090 -0.080173          -0.574674    6.095898     -
0.120284

      Underground_Full_Bathroom  Underground_Half_Bathroom  \
0                      1.108656                  -0.241148
1                     -0.819269                   3.947370
2                      1.108656                  -0.241148
3                      1.108656                  -0.241148
4                      1.108656                  -0.241148
...                         ...                        ...
1454                   1.108656                  -0.241148
1455                  -0.819269                  -0.241148
1456                   1.108656                  -0.241148
1457                  -0.819269                  -0.241148
1458                   1.108656                  -0.241148

      Half_Bathroom_Above_Grade  Fireplaces  W_Deck_Area  ...
Garage_Detchd  \
0                      1.228641   -0.951848     0.567296  ...
0.0
```

```
1                       -0.760912    0.599824     0.848746  ...
0.0
2                        1.228641    0.599824    -0.536161  ...
0.0
3                       -0.760912    0.599824    -0.369268  ...
1.0
4                        1.228641    0.599824    -0.830770  ...
0.0
...                           ...         ...          ...  ...
...
1454                    -0.760912   -0.951848    -0.825542  ...
0.0
1455                     1.228641    0.599824    -1.389652  ...
0.0
1456                    -0.760912    2.151495    -0.455580  ...
0.0
1457                    -0.760912    2.151495    -0.035650  ...
0.0
1458                    -0.760912   -0.951848     0.408972  ...
0.0

      Sale_Type_CWD  Sale_Type_Con  Sale_Type_ConLD
Sale_Type_ConLI  \
0               0.0            0.0              0.0              0.0

1               0.0            0.0              0.0              0.0

2               0.0            0.0              0.0              0.0

3               0.0            0.0              0.0              0.0

4               0.0            0.0              0.0              0.0

...             ...            ...              ...              ...

1454            0.0            0.0              0.0              0.0

1455            0.0            0.0              0.0              0.0

1456            0.0            0.0              0.0              0.0

1457            0.0            0.0              0.0              0.0

1458            0.0            0.0              0.0              0.0


      Sale_Type_ConLw  Sale_Type_New  Sale_Type_Oth  Sale_Type_WD  \
0                 0.0            0.0            0.0           1.0
1                 0.0            0.0            0.0           1.0
2                 0.0            0.0            0.0           1.0
```

```
3                  0.0            0.0            0.0           1.0
4                  0.0            0.0            0.0           1.0
...                ...            ...            ...           ...
1454               0.0            0.0            0.0           1.0
1455               0.0            0.0            0.0           1.0
1456               0.0            0.0            0.0           1.0
1457               0.0            0.0            0.0           1.0
1458               0.0            0.0            0.0           1.0

      Air_Conditioning_Binary
0                           1
1                           1
2                           1
3                           1
4                           1
...                       ...
1454                        1
1455                        1
1456                        1
1457                        1
1458                        1

[1459 rows x 141 columns]
```

After feature selection, I transformed the columns using appropriate encoding techniques: OneHotEncoding for categorical features, binary mapping for binary features, and ordinal encoding for ordinal features.

```
X = df_encoded.drop(columns="Sale_Price")
y = df_encoded["Sale_Price"]

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=23)

from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(X_train, y_train)

LinearRegression()

y_pred = lr.predict(X_test)

from sklearn.metrics import r2_score, mean_squared_error
print("R2 score: ", np.mean(r2_score(y_test, y_pred))*100)
print("MSE: ", mean_squared_error(y_test, y_pred))

R2 score:  69.42975456152712
MSE:  0.2469884707847605
```

After applying linear regression on the transformed data, the R^2 score improved to 69%, which is better than the previous score.

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import Ridge, Lasso, ElasticNet
from sklearn.neural_network import MLPRegressor

models = {
    'Decision Tree': DecisionTreeRegressor(),
    'Random Forest': RandomForestRegressor(),
    'Gradient Boosting': GradientBoostingRegressor(),
    'SVR': SVR(),
    'k-NN': KNeighborsRegressor(),
    'Ridge': Ridge(),
    'Lasso': Lasso(),
    'ElasticNet': ElasticNet(),
    'Neural Network': MLPRegressor(max_iter=1000)
}

results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    results[name] = {'MSE': mse, 'R²': r2}

# Print results
for name, metrics in results.items():
    print(f'{name} - MSE: {metrics["MSE"]:.4f}, R²:
{metrics["R²"]:.4f}')
```

```
Decision Tree - MSE: 0.2806, R²: 0.6527
Random Forest - MSE: 0.1429, R²: 0.8231
Gradient Boosting - MSE: 0.1372, R²: 0.8302
SVR - MSE: 0.1263, R²: 0.8437
k-NN - MSE: 0.1887, R²: 0.7665
Ridge - MSE: 0.1674, R²: 0.7928
Lasso - MSE: 0.8123, R²: -0.0055
ElasticNet - MSE: 0.7845, R²: 0.0290
Neural Network - MSE: 0.2134, R²: 0.7358
```

```
import xgboost as xgb

xgb = xgb.XGBRegressor(objective='reg:squarederror',
eval_metric='rmse')
```

```
xgb.fit(X_train, y_train)

XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None,
early_stopping_rounds=None,
             enable_categorical=False, eval_metric='rmse',
feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None,
max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan,
monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)

y_pred2 = xgb.predict(X_test)

# Evaluate the model
mse2 = mean_squared_error(y_test, y_pred2)
r22 = r2_score(y_test, y_pred2)

print("R2 score: ", r22)
print("MSE: ", mse2)

R2 score:  0.8062251458983503
MSE:  0.15655796741125982
```

Before Transformation (High Feature Dimensionality - 1459 features):

- Decision Tree (DT): High MSE (1.34 billion) and moderate R^2 (0.72) indicate a poor fit. The large number of features might be causing overfitting.
- Random Forest (RF): Lower MSE (494 million) and high R^2 (0.89) suggest a better fit than DT. However, there's still room for improvement.
- Gradient Boosting (GB): Even lower MSE (412 million) and a very high R^2 (0.91) show the best fit among tree-based models before transformation.
- Support Vector Regression (SVR): Very high MSE (513 billion) and negative R^2 indicate a complete failure to capture the relationship. This model likely doesn't suit this data.
- k-Nearest Neighbors (k-NN): Moderate MSE (820 million) and R^2 (0.83) suggest a decent fit, but there's potential for improvement.
- Ridge, Lasso, ElasticNet: These regularization models show moderate MSEs (around 500 million) and R^2s (around 0.89), indicating a trade-off between underfitting and overfitting due to regularization.

- Neural Network (NN): Very high MSE (610 billion) and negative R^2 imply a complete failure to learn from the data. The high dimensionality might be overwhelming the network.
- XGBoost: Similar performance to Gradient Boosting with slightly lower MSE and slightly higher R^2.

After Transformation (Reduced Feature Dimensionality - 48 features):

All Models: A significant decrease in MSE (all below 1) and a general increase in R^2 suggest a much better fit for all models. The data transformation likely addressed the issue of high dimensionality.

- DT, RF, GB, k-NN: These models show substantial improvements with much lower MSEs and higher R^2s, indicating the benefit of dimensionality reduction for tree-based and nearest neighbor methods.
- SVR: Still shows a high MSE, although lower than before. This suggests SVR might not be suitable for this data even after transformation.
- Ridge, Lasso: Both models show a significant increase in MSE and a negative R^2. This indicates overfitting due to the regularization being too strong for the lower dimensional data. We might need to adjust the regularization parameters.
- ElasticNet: Similar to Ridge and Lasso, it also suffers from overfitting with high MSE and negative R^2.
- NN: Shows improvement with lower MSE and positive R^2, but still performs worse than other models. The reduced dimensionality might have helped, but the network architecture might still need adjustments.
- XGBoost: Maintains a good balance with a low MSE and a high R^2, suggesting it benefits from the reduced dimensionality while avoiding overfitting.

Best Performer:

Based on the scores after data transformation, Gradient Boosting and XGBoost are the clear winners with the lowest MSEs and highest R^2s. They seem to be most effective in capturing the underlying relationships after dimensionality reduction. Here's a breakdown:

- Gradient Boosting & XGBoost (MSE: 0.1372 & 0.1565, R^2: 0.8302 & 0.806): These models perform very similarly, making it difficult to definitively choose one. You might need to perform further hyperparameter tuning or compare them on a hold-out validation set to determine the absolute best.
- Random Forest (MSE: 0.1429, R^2: 0.8231): A close contender, performing slightly worse than the top two.
- k-NN & SVR (MSE: 0.1887 & 0.1263, R^2: 0.7665 & 0.8437): These models show decent performance, but fall behind the top performers. SVR's high R^2 might be misleading due to its overall high MSE.

- Other Models (Ridge, Lasso, ElasticNet, Neural Network): These models suffered from overfitting after dimensionality reduction and require further adjustments.

More Exploratory Data Analysis on the clean Data

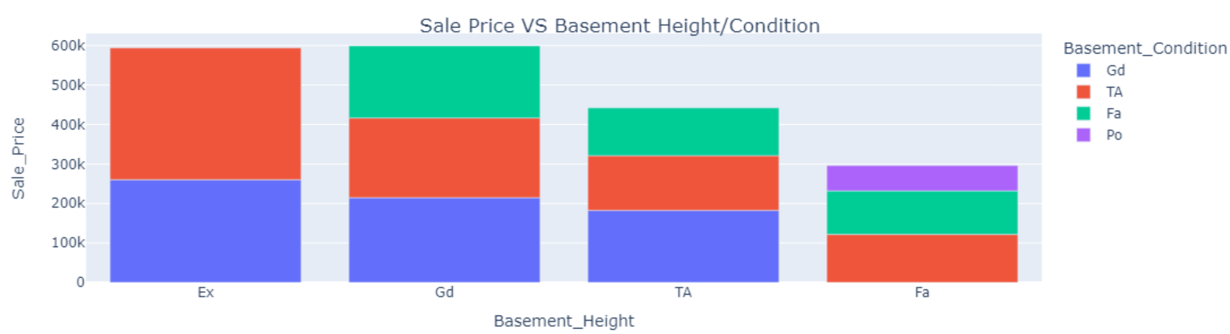
```
fig = px.scatter(new_new_df, x = "Sale_Price", y = "House_Design",
color="House_Type", size=df_encoded["Air_Conditioning_Binary"],
width=1000, height=500)
fig.update_layout(title="Sale Price VS House Design/Type",
title_x=0.5, title_y=0.94)
fig.show()
```

- It shows that single-family homes (1Fam) dominate the market and that 2.5Fin houses tend to have higher prices, while 2Story and 1Story houses are more common in the lower to mid-price ranges.

```
aggregated_data = new_new_df.groupby(['Roof_Design', "Roof_Quality"])
['Sale_Price'].mean().reset_index()

fig = px.bar(aggregated_data, x='Roof_Design', y='Sale_Price',
color="Roof_Quality")
fig.update_layout(title="Sale Price VS Roof Design/Quality",
title_x=0.5, title_y=0.94)
fig.show()
```

- from above graph gabled roof tends to be the most expensive design, with an average sale price of around $137k. Homes with a flat roof or a hip roof tend to be the least expensive on average.

```
aggregated_data = new_new_df.groupby(['Basement_Height',
'Basement_Condition'])['Sale_Price'].mean().reset_index()

fig = px.bar(aggregated_data, x='Basement_Height', y='Sale_Price',
color="Basement_Condition")
fig.update_layout(title="Sale Price VS Basement Height/Condition",
title_x=0.5, title_y=0.87)
fig.show()
```

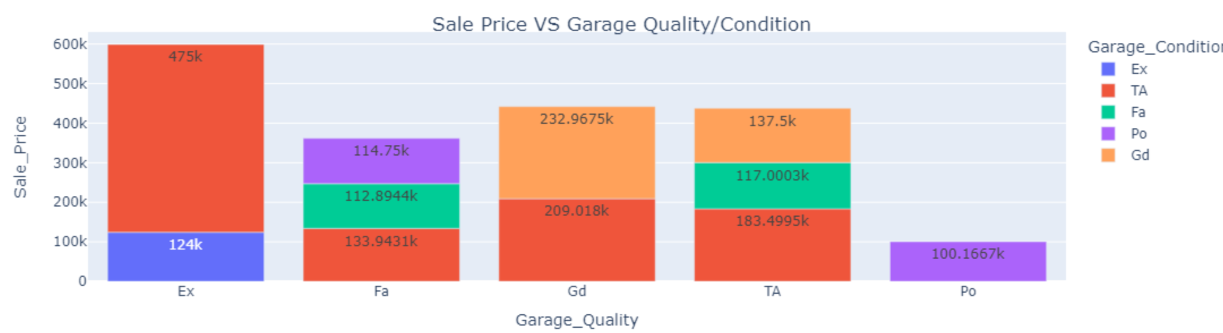


- There are houses with high basements (presumably rated Excellent or Good) that sold for a lower price than houses with shorter basements (possibly rated Fair or Poor).
- Similarly, there are houses with Poor or Fair condition basements that sold for a higher price than those with Excellent or Good condition basements.

```
aggregated_data = new_new_df.groupby(["Kitchen_Quality"])
['Sale_Price'].mean().reset_index()

fig = px.bar(aggregated_data, x='Kitchen_Quality', y='Sale_Price')
fig.update_layout(title="Mean Sale Price VS Kitchen Quality",
title_x=0.5, title_y=0.87)
fig.show()
```

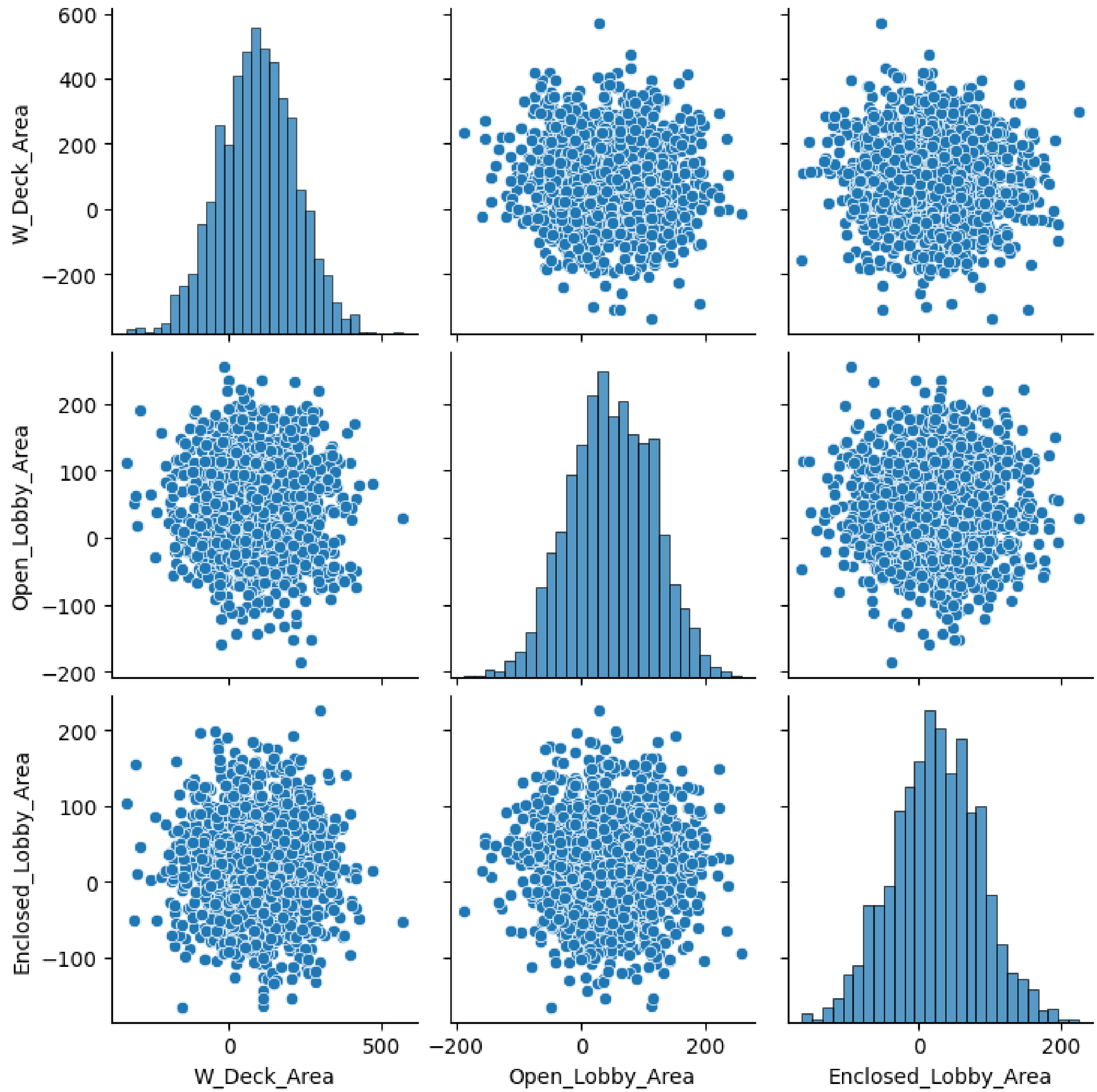
- graph you sent shows a trend where the average sale price is relatively low for kitchens rated Good while Excellent kitchens have a higher average sale price.
- Time: The data may not account for the age of the kitchens. An Excellent kitchen might be a recent renovation, while a Good kitchen might be older.

```
aggregated_data = new_new_df.groupby(['Garage_Quality',
'Garage_Condition'])['Sale_Price'].mean().reset_index()

fig = px.bar(aggregated_data, x='Garage_Quality', y='Sale_Price',
color="Garage_Condition", text_auto=True)
fig.update_layout(title="Sale Price VS Garage Quality/Condition",
title_x=0.5, title_y=0.87)
fig.show()
```



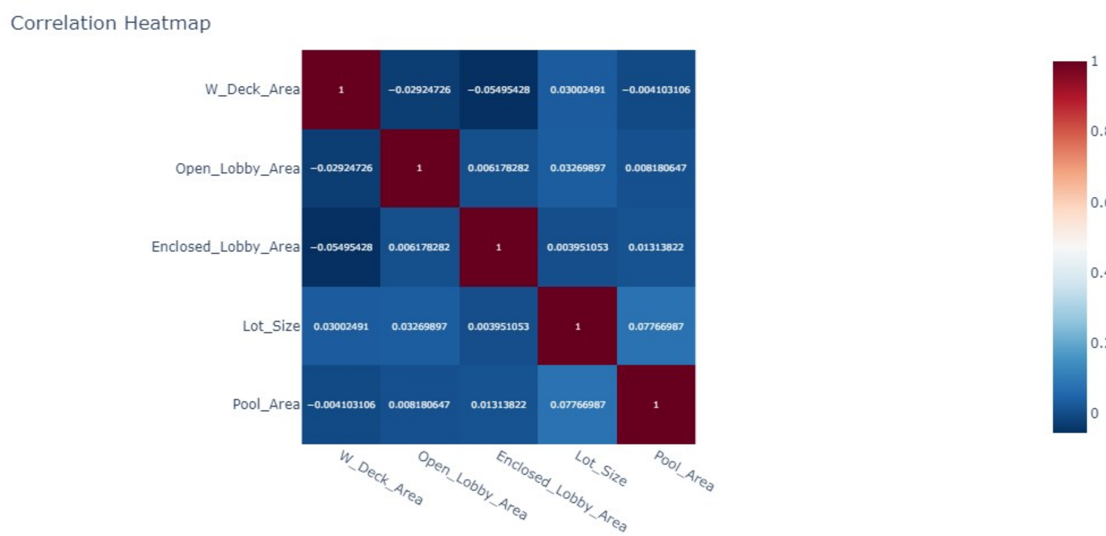
- There is a positive correlation between sale price and garage condition. This means that houses with garages in better condition tend to sell for more than houses with garages in poorer condition.
- For example, a house with a garage in Excellent condition (Ex) might sell for an average price around 600k, ##### whereas a house with a garage in Poor condition (Po) might sell for an average price around 100k.

```
sns.pairplot(new_new_df, vars=["W_Deck_Area", "Open_Lobby_Area",
"Enclosed_Lobby_Area"])
plt.show()
```




```
new_new_df.columns

Index(['Building_Class', 'Zoning_Class', 'Lot_Size', 'Property_Shape',
       'Land_Outline', 'Lot_Configuration', 'Neighborhood',
'Condition1',
       'House_Type', 'House_Design', 'Roof_Design', 'Roof_Quality',
       'Exterior1st', 'Exterior2nd', 'Brick_Veneer_Area',
'Exterior_Material',
       'Exterior_Condition', 'Foundation_Type', 'Basement_Height',
       'Basement_Condition', 'Exposure_Level', 'BsmtFinType1',
'BsmtFinSF2',
       'Heating_Quality', 'Air_Conditioning', 'Electrical_System',
```

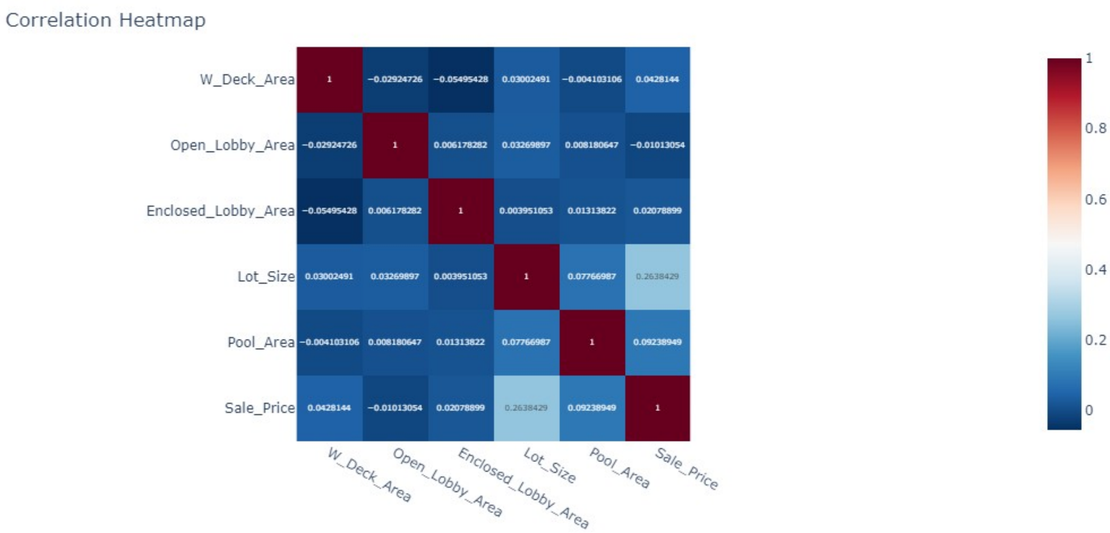
```
       'LowQualFinSF', 'Underground_Full_Bathroom',
       'Underground_Half_Bathroom', 'Half_Bathroom_Above_Grade',
       'Kitchen_Quality', 'Functional_Rate', 'Fireplaces', 'Garage',
       'Garage_Finish_Year', 'Garage_Quality', 'Garage_Condition',
       'Pavedd_Drive', 'W_Deck_Area', 'Open_Lobby_Area',
'Enclosed_Lobby_Area',
       'Three_Season_Lobby_Area', 'Screen_Lobby_Area', 'Pool_Area',
       'Miscellaneous_Value', 'Sale_Type', 'Sale_Condition',
'Sale_Price'],
      dtype='object')

#df_encoded.to_csv("train.csv", index=False)

corelation_matrix = new_new_df[["W_Deck_Area", "Open_Lobby_Area",
"Enclosed_Lobby_Area", "Lot_Size", "Pool_Area"]].corr()

fig = px.imshow(corelation_matrix, text_auto=True,
color_continuous_scale='RdBu_r', width=1000, height=500)
fig.update_layout(title="Correlation Heatmap", title_x=0.1,
title_y=0.94)
fig.show()
```



- There is a positive correlation between Lot_Size and Pool_Area, which means that houses with larger lots tend to also have larger pools.
- There is a negative correlation between Enclosed_Lobby_Area and Open_Lobby_Area, which means that houses with larger enclosed lobbies tend to have smaller open lobbies.

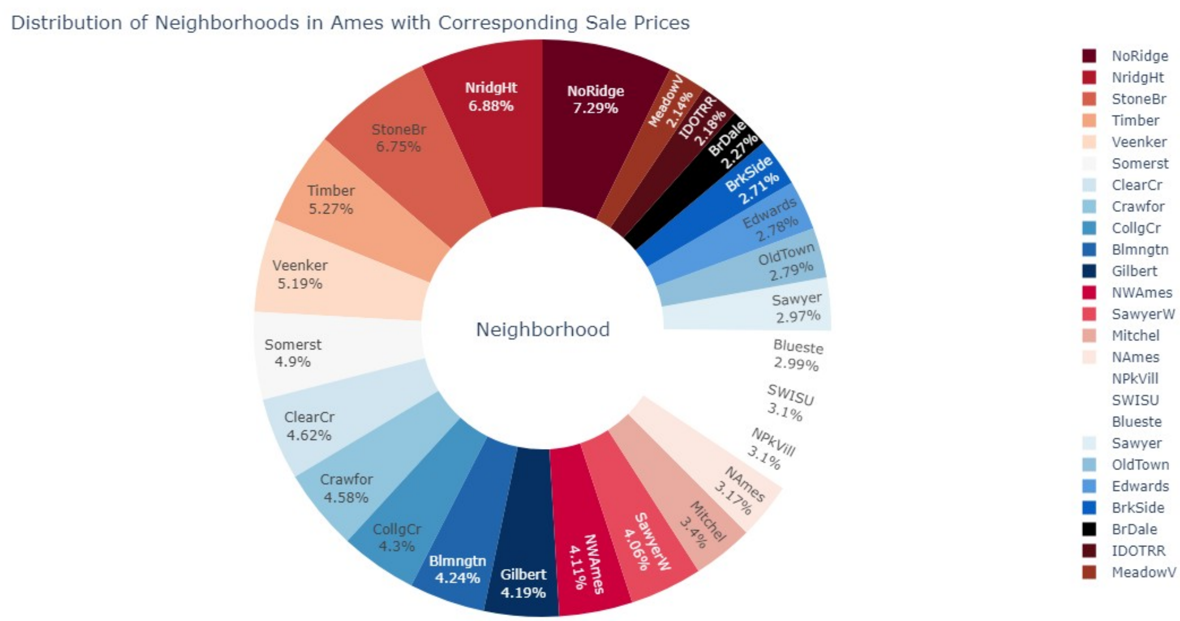
```
corelation_matrix = new_new_df[["W_Deck_Area", "Open_Lobby_Area",
"Enclosed_Lobby_Area", "Lot_Size", "Pool_Area", "Sale_Price"]].corr()

fig = px.imshow(corelation_matrix, text_auto=True,
color_continuous_scale='RdBu_r', width=1000, height=500)
fig.update_layout(title="Correlation Heatmap", title_x=0.1,
```

```
title_y=0.94)
fig.show()
```



- we can see that there is a strong positive correlation between sale price and lot size. This means that houses with larger lots tend to sell for more money.
- There is also a weak positive correlation between sale price and open lobby area. This means that houses with larger open lobby areas tend to sell for more money
- but the correlation is not as strong as the correlation between sale price and lot size. There is a weak positive correlation between sale price and enclosed lobby area
- There is a negative correlation between sale price and enclosed lobby area. This means that houses with larger enclosed lobby areas tend to sell for less money. However, it's important to note that the correlation is not very strong.

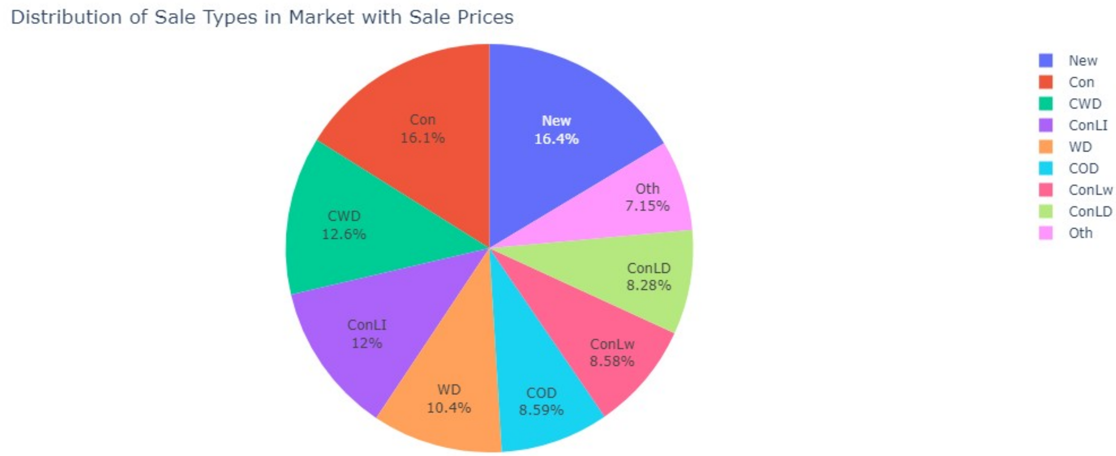
```
aggregated_data = new_new_df.groupby(['Neighborhood'])
['Sale_Price'].mean().reset_index()
fig = px.pie(aggregated_data, values='Sale_Price',
names='Neighborhood', hole=.42,
color_discrete_sequence=px.colors.sequential.RdBu)
fig.update_layout(title="Distribution of Neighborhoods in Ames with
Corresponding Sale Prices", title_x=0.05, title_y=0.94, width=1000,
height=650,
                 annotations=[dict(text='Neighborhood', x=0.5, y=0.5,
font_size=17, showarrow=False)])
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.show()
```



The pie chart represents the percentage distribution of neighborhoods within Ames city limits, along with their respective sale prices. Here's an analysis of the trends depicted in the chart:

- NoRidge (7.29%) and NridgHt (6.88%) have the highest percentages, indicating that these neighborhoods are the most common or have the highest sales volumes within the dataset.
- StoneBr (6.75%) and Timber (5.27%) also constitute a significant portion of the sales, suggesting that these neighborhoods are also popular or have a high frequency of sales.
- Veenker (5.19%), Somerst (4.9%), ClearCr (4.62%), and Crawfor (4.58%) show notable percentages, reflecting their importance in the housing market.
- Other neighborhoods such as CollgCr, Blmngtn, Gilbert, NWAmes, SawyerW, Mitchel, NPkVill, SWISU, Blueste, and others have varying percentages, with the lower end around 2.7% (BrDale, BrkSide).

```
aggregated_data = new_new_df.groupby(['Sale_Type'])
['Sale_Price'].mean().reset_index()
fig = px.pie(aggregated_data, values='Sale_Price', names='Sale_Type')
fig.update_layout(title="Distribution of Sale Types in Market with
Sale Prices", title_x=0.1, title_y=0.94, width=1000, height=500)
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.show()
```



The pie chart illustrates the distribution of different sale types with their corresponding percentages.

- New (16.4%) and Con (16.1%) have the highest percentage, indicating that newly constructed homes and properties sold through a contract with a 15% down payment on regular terms are the most common sale types.
- CWD (12.6%) and ConLI (12%) also constitute a significant portion of the sales. Cash sales (CWD) and sales through a contract with low interest (ConLI) are popular among buyers.
- WD (10.4%) and COD (8.59%) follow next, showing that conventional warranty deed sales and court officer deed/estate sales are also relatively common.
- ConLw (8.58%) and ConLD (8.28%) have similar proportions, suggesting that sales through contracts with low down payment and low interest or low down payment are also utilized options.
- Oth (7.15%) is the least common, indicating other types of sales are less frequent.

```
aggregated_data = new_new_df.groupby(['Functional_Rate'])
['Sale_Price'].mean().reset_index()
fig = px.treemap(aggregated_data, path = ["Functional_Rate"], color =
"Sale_Price")
fig.update_layout(title="Impact of Functional Rate on Sale Prices",
title_x=0.5, title_y=0.94, width=1000, height=500)
fig.show()
```

The tree map visualizes the distribution of different functional rates of homes along with their sale prices.

- The chart indicates that the condition of the home (functional rate) has a noticeable impact on the sale price. Homes with typical functionality command the highest prices, while those with severe damage or major deductions have the lowest sale prices.