HOUSE PRICE PREDICTION

Using Machine learning

PROJECT GUIDE: Prof. Suvarna Ranade

PROJECT BY–

- PRASANNA KAILAS SHINDE (ROLL NO 36)
- MRUNALI MANOHAR PETAKAR (ROLL NO 30)

Index :

goal / aim :

The goal of the project is to develop a robust and accurate machine learning model that can predict house prices based on various features related to real estate properties in Ames city. By leveraging advanced analytics and predictive modeling, our objective is to create a model that outperforms traditional methods, providing accurate and reliable predictions of house sale prices.

data overview :

- The dataset comprises real estate data from Ames city, including a comprehensive set of features related to residential properties.
- **Key Features:**
  Property Details:

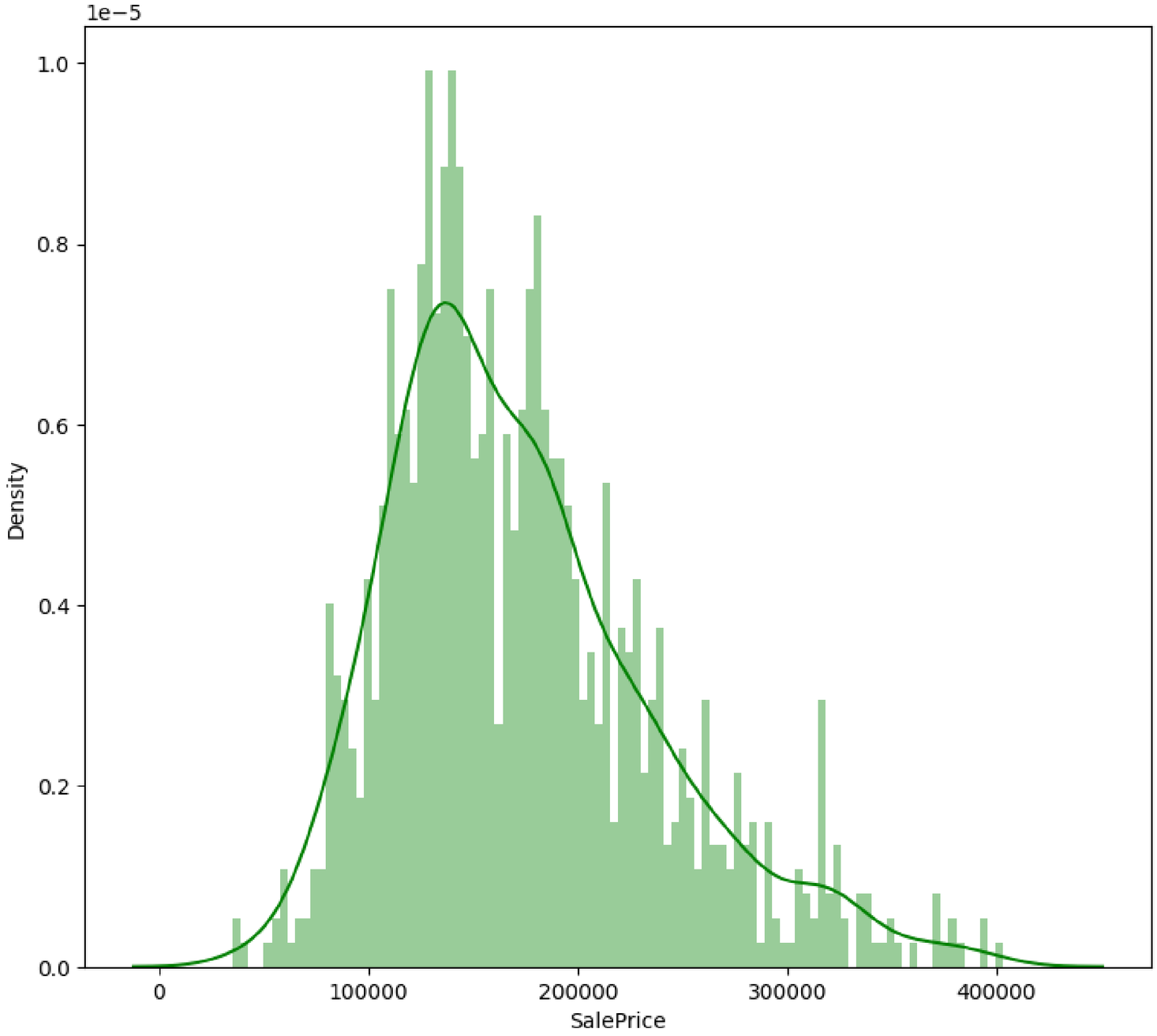
1. SalePrice: This is the target variable that the model aims to predict.
2. LotArea: Indicates the size of the lot on which the property is situated.
3. Utilities: Describes the types of utilities accessible at the property.

and so on...

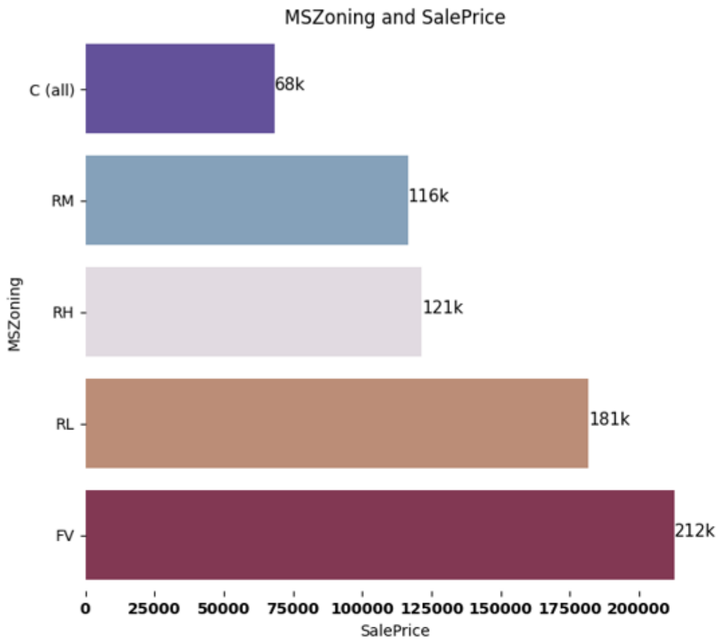
the link for dataset is given here

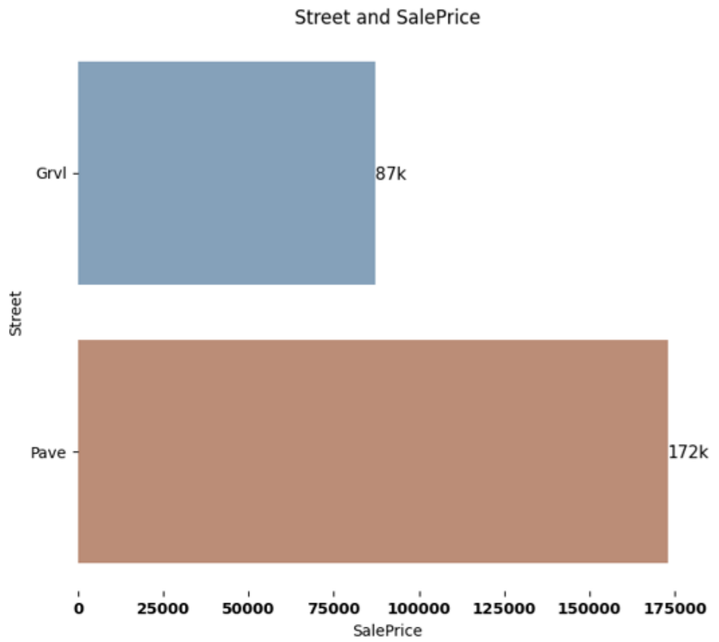
1460
rows

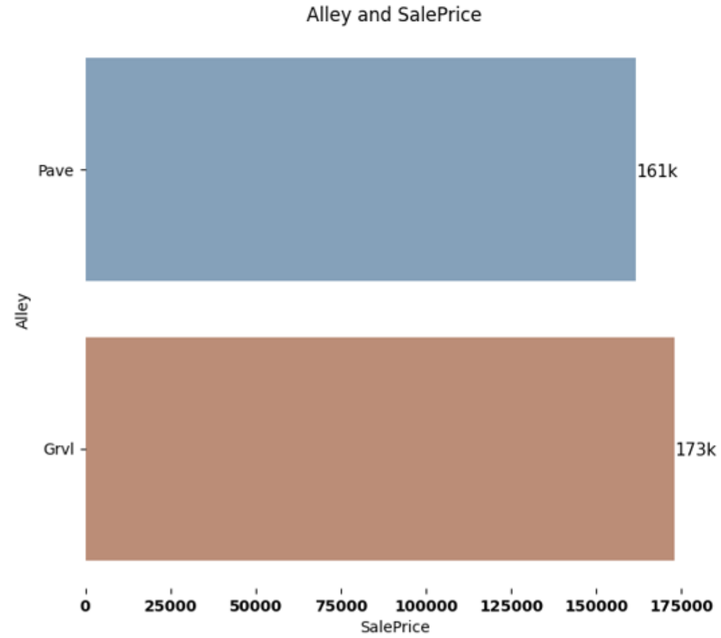
81
columns

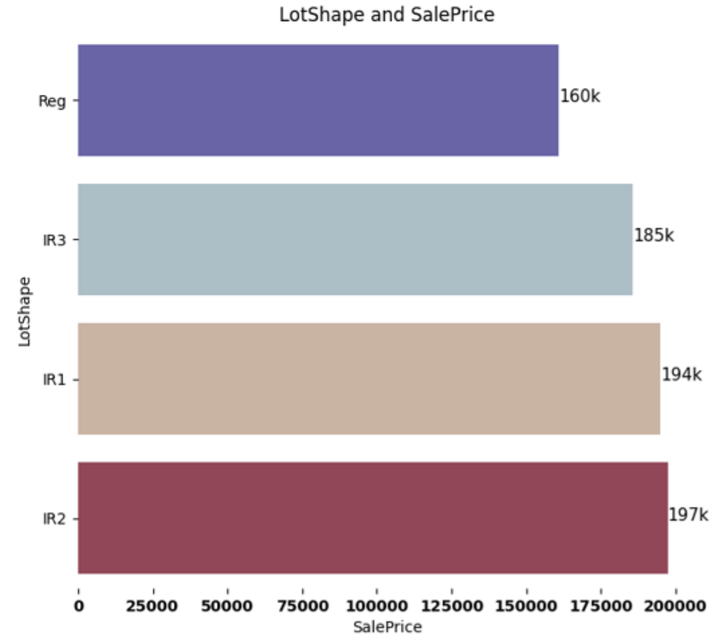


Saleprice is positively distributed



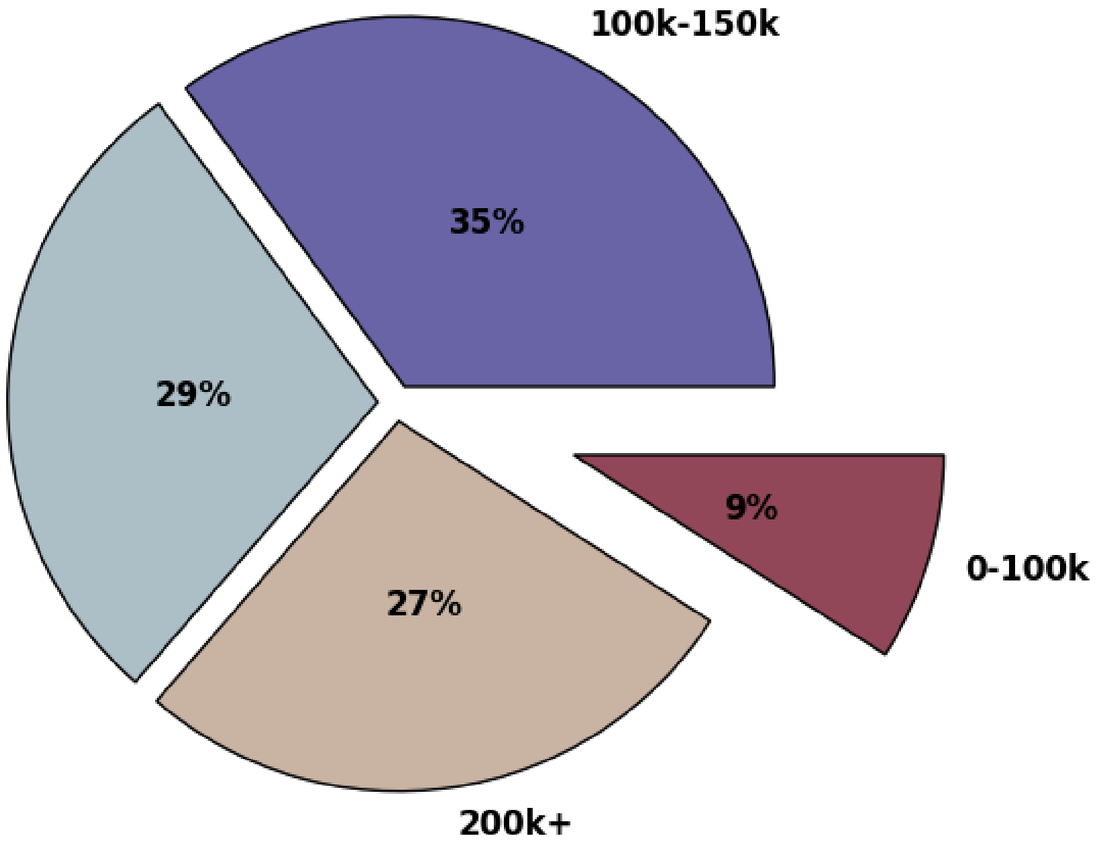


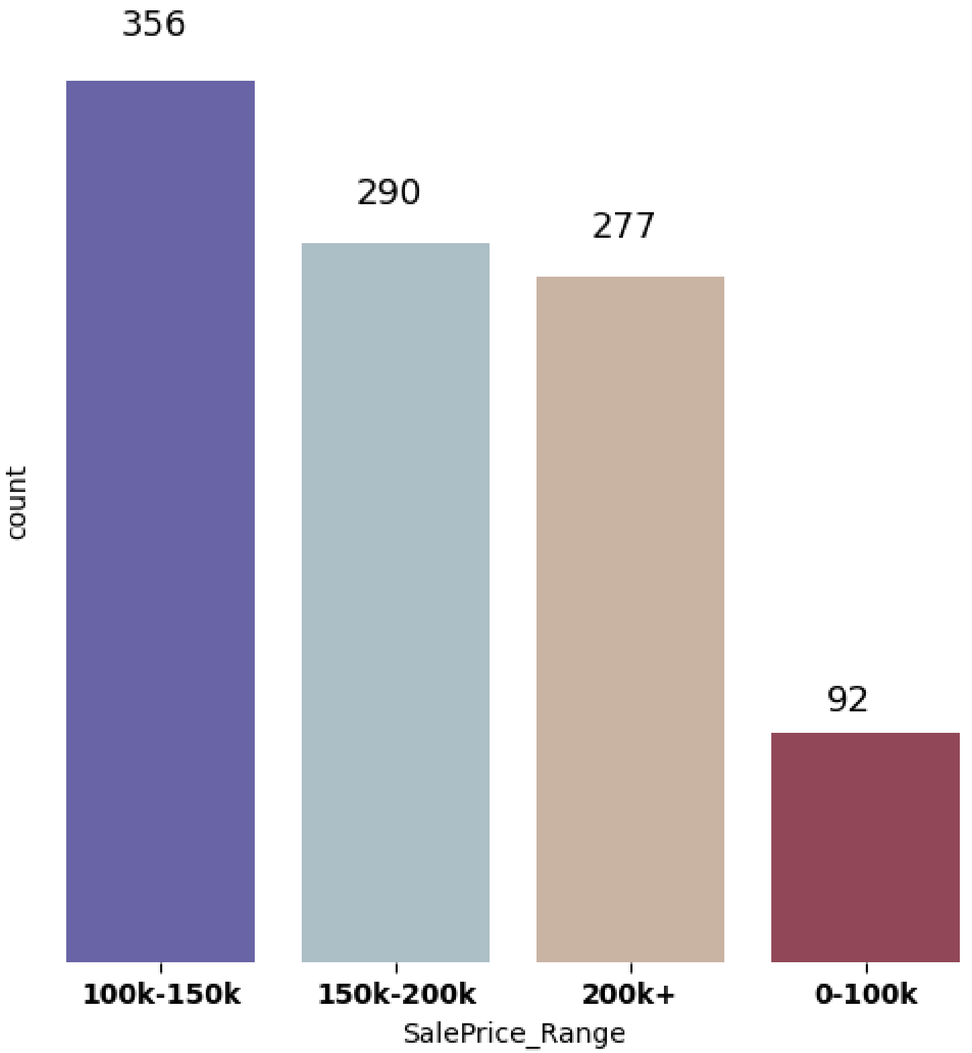




We can observe the price distributions across categorical variables through this visualization. It's evident how the prices are influenced by certain unique features of houses.

► SalePrice Distribution ◄





We created a temporary categorical variable to check the distribution of the target variable. As seen, our dataset predominantly contains houses from the middle and upper segments.

data exploration & cleaning:

first of all we imported the important liabraries like pandas, numpy, scikit-learn, matplotlib, seaborn etc.

1.Handling Missing Values:

For numerical features: Imputation with mean.

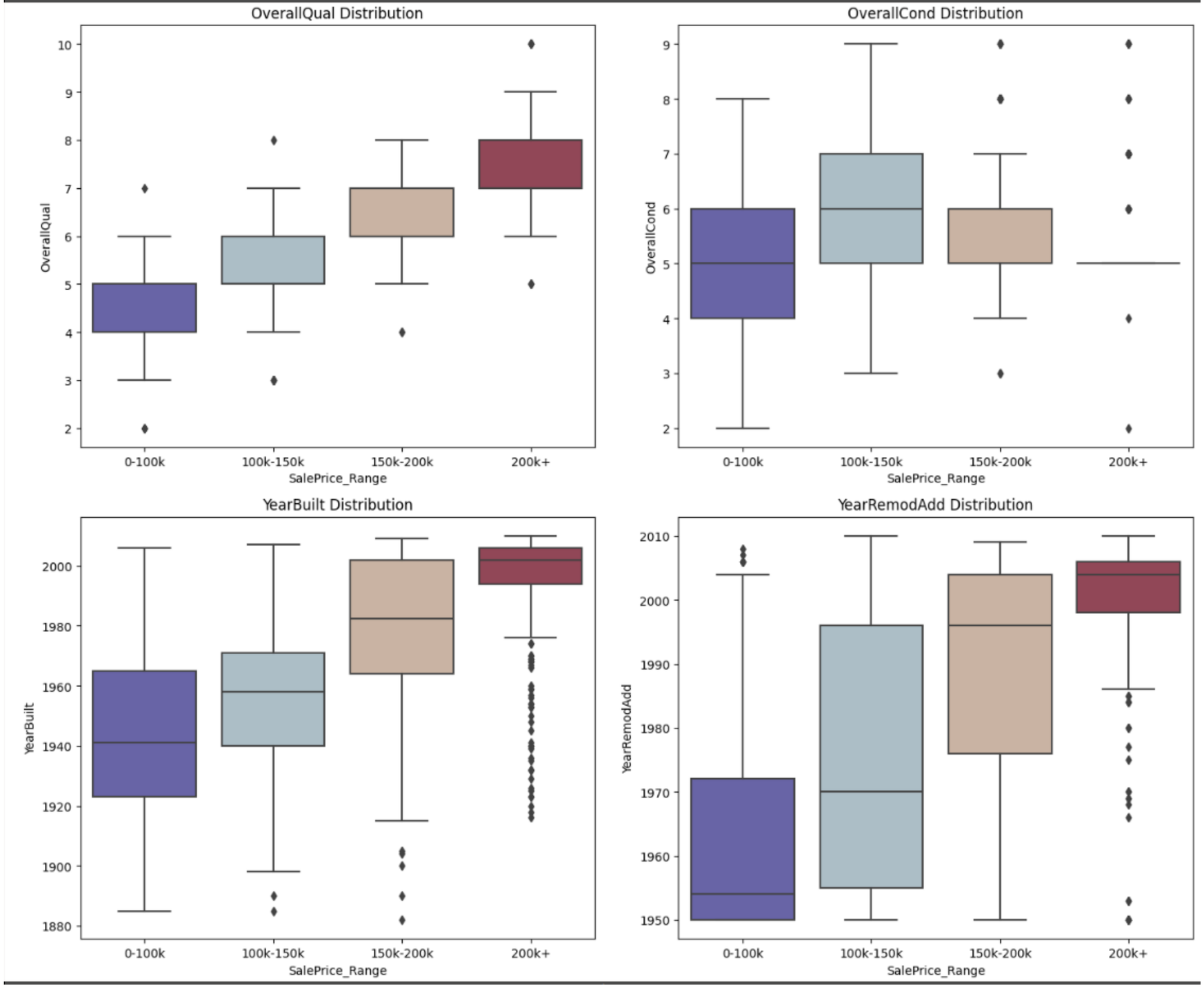
For categorical features: Imputation with the mode.

2. Outlier Detection and Removal:

- Divided columns into categorical (cat)(43)and continuous (con)(38)variables.
- Used Z-score (scipy.stats.zscore) to identify outliers in continuous variables.
- Detected outliers using a threshold (Z-score > 3 or Z-score < -3).

Handling Strategy:

- Removed rows containing outliers.
- Reindexed the dataset for consistency.
- 1015 rows × 24 columns

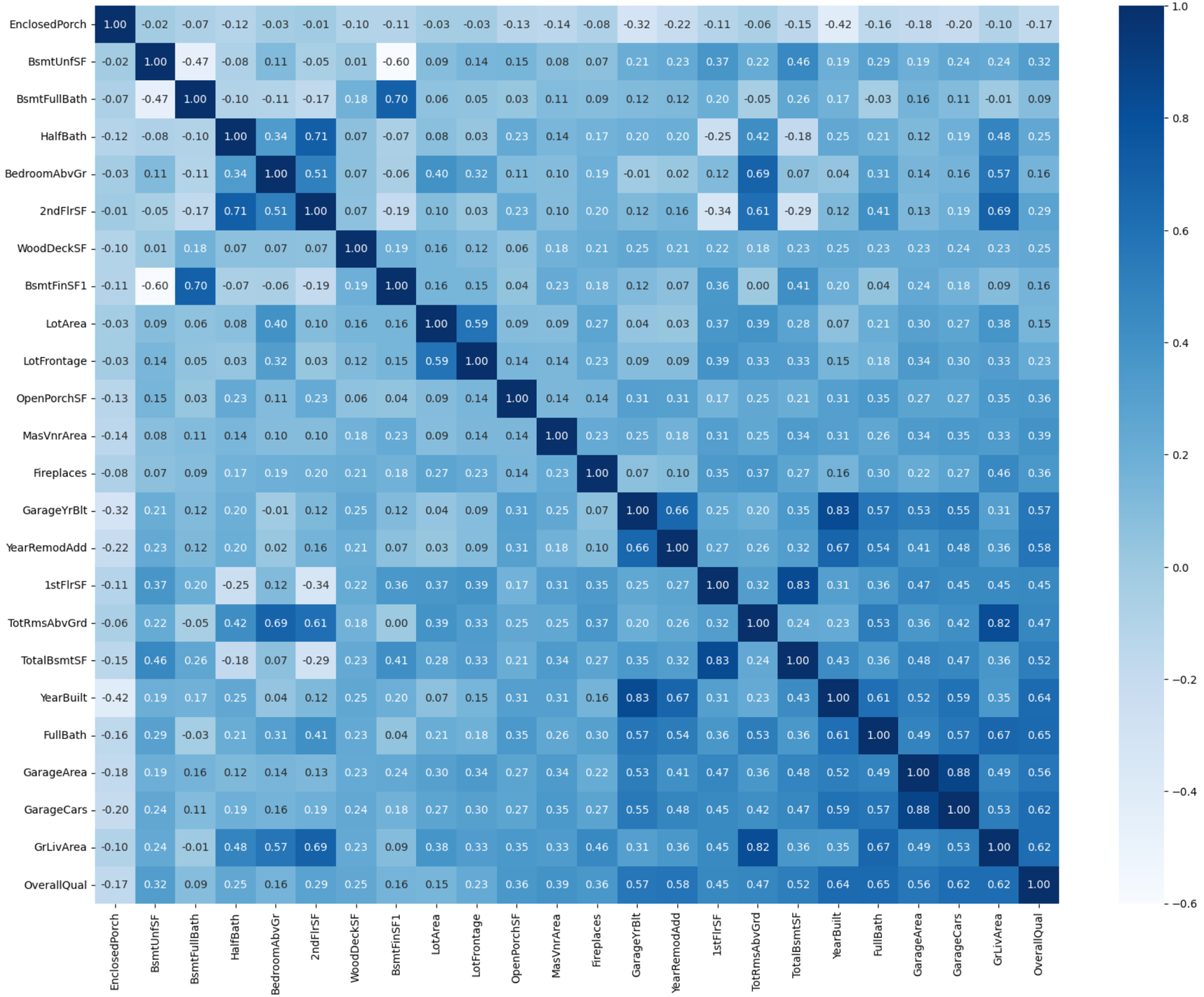

OverallQual Distribution
OverallQual
SalePrice_Range
0-100k
100k-150k
150k-200k
200k+
OverallCond Distribution
OverallCond
YearBuilt Distribution
YearBuilt
YearRemodAdd Distribution
YearRemodAdd

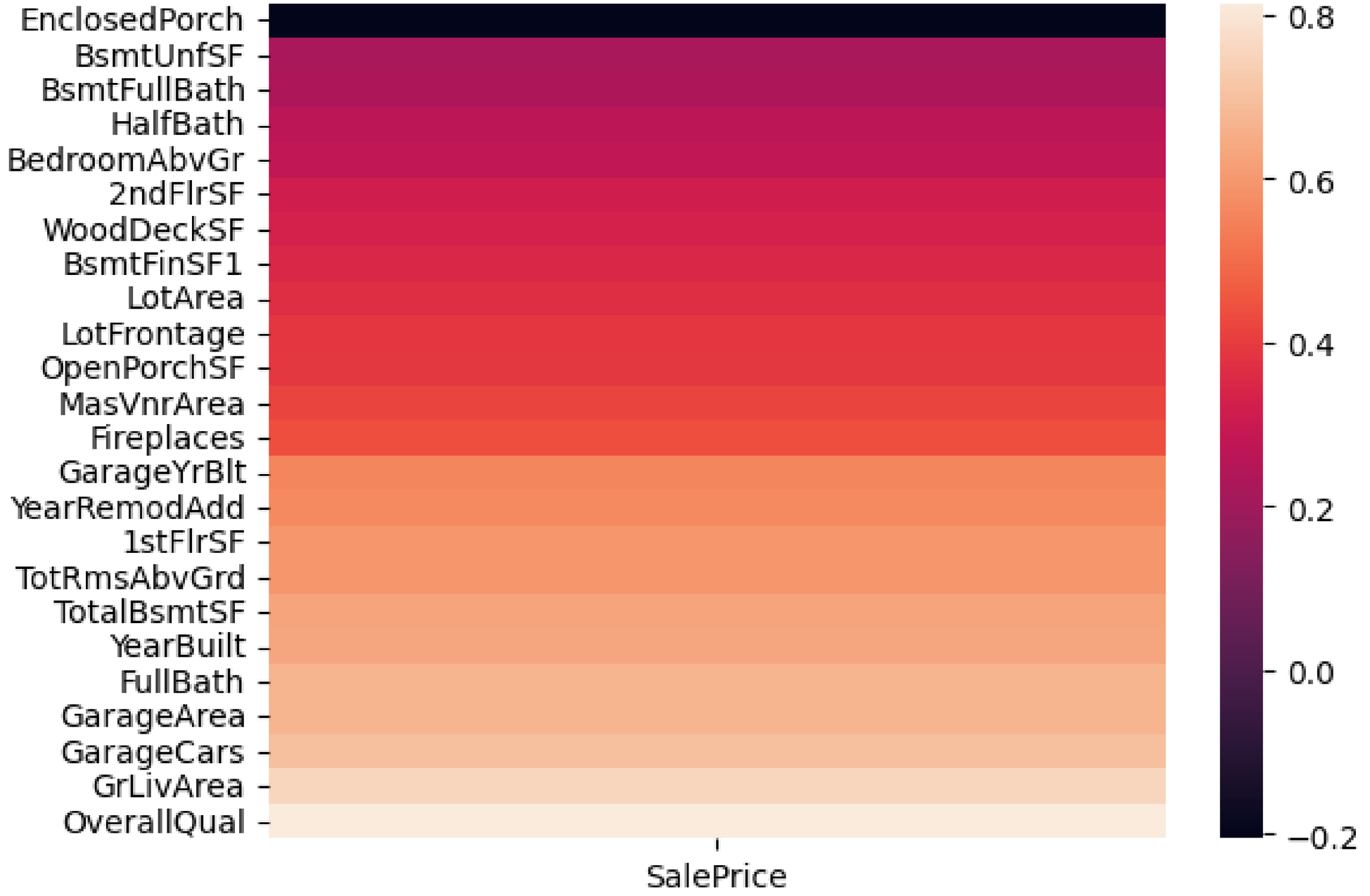
EDA (exploratory data analysis)

Exploratory Data Analysis (EDA) plays a crucial role in understanding the relationships within the dataset and identifying patterns that can guide the feature selection process.

- Determined which features are strongly correlated with the target variable (**SalePrice**).
- Selected features(24 features) with correlations above and below threshold.
- A correlation heatmap visually represents the correlation matrix.
- Features with stronger correlations are highlighted using a color gradient.

Heatmap:


EnclosedPorch 1.00 -0.02 -0.07 -0.12 -0.03 -0.01 -0.10 -0.11 -0.03 -0.03 -0.13 -0.14 -0.08 -0.32 -0.22 -0.11 -0.06 -0.15 -0.42 -0.16 -0.18 -0.20 -0.10 -0.17
BsmtUnfSF -0.02 1.00 -0.47 -0.08 0.11 -0.05 0.01 -0.60 0.09 0.14 0.15 0.08 0.07 0.21 0.23 0.37 0.22 0.46 0.19 0.29 0.19 0.24 0.24 0.32
BsmtFullBath -0.07 -0.47 1.00 -0.10 -0.11 -0.17 0.18 0.70 0.06 0.05 0.03 0.11 0.09 0.12 0.12 0.20 -0.05 0.26 0.17 -0.03 0.16 0.11 -0.01 0.09
HalfBath -0.12 -0.08 -0.10 1.00 0.34 0.71 0.07 -0.07 0.08 0.03 0.23 0.14 0.17 0.20 0.20 -0.25 0.42 -0.18 0.25 0.21 0.12 0.19 0.48 0.25
BedroomAbvGr -0.03 0.11 -0.11 0.34 1.00 0.51 0.07 -0.06 0.40 0.32 0.11 0.10 0.19 -0.01 0.02 0.12 0.69 0.07 0.04 0.31 0.14 0.16 0.57 0.16
2ndFlrSF -0.01 -0.05 -0.17 0.71 0.51 1.00 0.07 -0.19 0.10 0.03 0.23 0.10 0.20 0.12 0.16 -0.34 0.61 -0.29 0.12 0.41 0.13 0.19 0.69 0.29
WoodDeckSF -0.10 0.01 0.18 0.07 0.07 0.07 1.00 0.19 0.16 0.12 0.06 0.18 0.21 0.25 0.21 0.22 0.18 0.23 0.25 0.23 0.23 0.24 0.23 0.25
BsmtFinSF1 -0.11 -0.60 0.70 -0.07 -0.06 -0.19 0.19 1.00 0.16 0.15 0.04 0.23 0.18 0.12 0.07 0.36 0.00 0.41 0.20 0.04 0.24 0.18 0.09 0.16
LotArea -0.03 0.09 0.06 0.08 0.40 0.10 0.16 0.16 1.00 0.59 0.09 0.09 0.27 0.04 0.03 0.37 0.39 0.28 0.07 0.21 0.30 0.27 0.38 0.15
LotFrontage -0.03 0.14 0.05 0.03 0.32 0.03 0.12 0.15 0.59 1.00 0.14 0.14 0.23 0.09 0.09 0.39 0.33 0.33 0.15 0.18 0.34 0.30 0.33 0.23
OpenPorchSF -0.13 0.15 0.03 0.23 0.11 0.23 0.06 0.04 0.09 0.14 1.00 0.14 0.14 0.31 0.31 0.17 0.25 0.21 0.31 0.35 0.27 0.27 0.35 0.36
MasVnrArea -0.14 0.08 0.11 0.14 0.10 0.10 0.18 0.23 0.09 0.14 0.14 1.00 0.23 0.25 0.18 0.31 0.25 0.34 0.31 0.26 0.34 0.35 0.33 0.39
Fireplaces -0.08 0.07 0.09 0.17 0.19 0.20 0.21 0.18 0.27 0.23 0.14 0.23 1.00 0.07 0.10 0.35 0.37 0.27 0.16 0.30 0.22 0.27 0.46 0.36
GarageYrBlt -0.32 0.21 0.12 0.20 -0.01 0.12 0.25 0.12 0.04 0.09 0.31 0.25 0.07 1.00 0.66 0.25 0.20 0.35 0.83 0.57 0.53 0.55 0.31 0.57
YearRemodAdd -0.22 0.23 0.12 0.20 0.02 0.16 0.21 0.07 0.03 0.09 0.31 0.18 0.10 0.66 1.00 0.27 0.26 0.32 0.67 0.54 0.41 0.48 0.36 0.58
1stFlrSF -0.11 0.37 0.20 -0.25 0.12 -0.34 0.22 0.36 0.37 0.39 0.17 0.31 0.35 0.25 0.27 1.00 0.32 0.83 0.31 0.36 0.47 0.45 0.45 0.45
TotRmsAbvGrd -0.06 0.22 -0.05 0.42 0.69 0.61 0.18 0.00 0.39 0.33 0.25 0.25 0.37 0.20 0.26 0.32 1.00 0.24 0.23 0.53 0.36 0.42 0.82 0.47
TotalBsmtSF -0.15 0.46 0.26 -0.18 0.07 -0.29 0.23 0.41 0.28 0.33 0.21 0.34 0.27 0.35 0.32 0.83 0.24 1.00 0.43 0.36 0.48 0.47 0.36 0.52
YearBuilt -0.42 0.19 0.17 0.25 0.04 0.12 0.25 0.20 0.07 0.15 0.31 0.31 0.16 0.83 0.67 0.31 0.23 0.43 1.00 0.61 0.52 0.59 0.35 0.64
FullBath -0.16 0.29 -0.03 0.21 0.31 0.41 0.23 0.04 0.21 0.18 0.35 0.26 0.30 0.57 0.54 0.36 0.53 0.36 0.61 1.00 0.49 0.57 0.67 0.65
GarageArea -0.18 0.19 0.16 0.12 0.14 0.13 0.23 0.24 0.30 0.34 0.27 0.34 0.22 0.53 0.41 0.47 0.36 0.48 0.52 0.49 1.00 0.88 0.49 0.56
GarageCars -0.20 0.24 0.11 0.19 0.16 0.19 0.24 0.18 0.27 0.30 0.27 0.35 0.27 0.55 0.48 0.45 0.42 0.47 0.59 0.57 0.88 1.00 0.53 0.62
GrLivArea -0.10 0.24 -0.01 0.48 0.57 0.69 0.23 0.09 0.38 0.33 0.35 0.33 0.46 0.31 0.36 0.45 0.82 0.36 0.35 0.67 0.49 0.53 1.00 0.62
OverallQual -0.17 0.32 0.09 0.25 0.16 0.29 0.25 0.16 0.15 0.23 0.36 0.39 0.36 0.57 0.58 0.45 0.47 0.52 0.64 0.65 0.56 0.62 0.62 1.00
EnclosedPorch
BsmtUnfSF
BsmtFullBath
HalfBath
BedroomAbvGr
2ndFlrSF
WoodDeckSF
BsmtFinSF1
LotArea
LotFrontage
OpenPorchSF
MasVnrArea
Fireplaces
GarageYrBlt
YearRemodAdd
1stFlrSF
TotRmsAbvGrd
TotalBsmtSF
YearBuilt
FullBath
GarageArea
GarageCars
GrLivArea
OverallQual
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


EnclosedPorch
BsmtUnfSF
BsmtFullBath
HalfBath
BedroomAbvGr
2ndFlrSF
WoodDeckSF
BsmtFinSF1
LotArea
LotFrontage
OpenPorchSF
MasVnrArea
Fireplaces
GarageYrBlt
YearRemodAdd
1stFlrSF
TotRmsAbvGrd
TotalBsmtSF
YearBuilt
FullBath
GarageArea
GarageCars
GrLivArea
OverallQual
SalePrice
0.8
0.6
0.4
0.2
0.0
−0.2

Preprocessing and Feature Selection

- Preprocessing and feature selection are critical steps to ensure that the data is in a suitable format for machine learning models and to identify the most relevant features for accurate predictions.

1. Scaling of Continuous Features:

Ensured that continuous features are on a similar scale to prevent certain features from dominating others.

Utilized MinMaxScaler to scale continuous features between 0 and 1.

2. Dealing with categorical and continuous data:

- We created new dataframe (X1) containing all important continuous features(total 24).
- Then we converted categorical columns into a format suitable for machine learning algorithms.
- Employed pd.get_dummies() to create binary columns for each category within a variable.

- Created a new DataFrame (X2) containing these one-hot encoded categorical features(total 224).
- Then joined Categorical and Continuous Data.
- This consolidated dataset (Xnew) is used for subsequent steps in the machine learning process.

Benefits:
The effective handling of categorical data ensures that all features are in a format suitable for machine learning algorithms.
One-hot encoding enables the incorporation of categorical information without introducing ordinal relationships that could impact model performance.

3. **Divide the dataset into training and testing sets(75%-25%) for model training and evaluation.**
4. **Feature Selection using OLS Regression:**
   Utilized Ordinary Least Squares (OLS) regression and adjusted R-squared values to iteratively drop less significant features.

Model building & evaluation

Model building and evaluation represent the core of the predictive analysis, involving the construction of machine learning models and assessing their performance on both training and testing datasets.

1. ***Linear Regression Model:***

Constructed a baseline linear regression model.
Training:
Used the LinearRegression model from scikit-learn to train on the training set (xtrain, ytrain).
Testing Evaluation:
Evaluated the model on the testing set (xtest, ytest).
used Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared, and Adjusted R-squared value to check accuracy.
Accuracy - R_squared: 0.89
Adj_R2: 0.83

2. Ridge and Lasso Regression Models

- Ridge and Lasso introduce a penalty term to the linear regression objective function, discouraging the model from assigning excessively large weights to any particular feature. This helps prevent overfitting by promoting simpler models.
- There are many features, and some are believed to be less relevant, Lasso can be used to automatically select a subset of the most important features.
  Used the Ridge and Lasso models from scikit-learn to train on the training set.
- Ridge and Lasso can improve a model's generalization performance by preventing it from fitting the noise in the training data too closely.

Accuracy - *Ridge*(R_squared: 0.92 , Adj_R2: 0.87)
Lasso(R_squared: 0.90 , Adj_R2: 0.84)

3. *Comparison and Model Selection*

Compared performance metrics (MAE, MSE, R-squared, Adjusted R-squared) across linear, Ridge, and Lasso models.

- After thorough model training and evaluation, the Ridge regression model has been selected as the final model for predicting house prices based on a combination of performance and interpretability factors.

- Performance:
  - Evaluation Metrics: Comparative evaluation of performance metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared, and Adjusted R-squared, across linear, Ridge, and Lasso models.
  - Balanced Performance: Ridge regression demonstrated a balance between bias and variance, minimizing overfitting while maintaining a good fit to the data.
- Interpretability:
  - Feature Coefficients: Ridge regression, being a regularized linear model, assigns coefficients to features. The interpretability of these coefficients allows for a better understanding of the impact each feature has on the predicted house prices.
  - Model Simplicity: Ridge regression strikes a balance between simplicity and complexity, making it interpretable for stakeholders without sacrificing predictive performance.
- Conclusion: The Ridge regression model is selected for its overall balanced performance and the interpretability it provides, aligning well with the goals of accurate prediction and model transparency.

Predictions and Future Steps:

- The final Ridge regression model was applied to the test set for predicting house prices, and results were prepared for submission.
- The model's predictions are ready for real-world application.
- By exploring these avenues for improvement, our team is committed to staying at the forefront of data science, ensuring our models are not only accurate but also adaptive to emerging challenges and opportunities in the real estate domain. We look forward to the continuous evolution of our predictive capabilities.
- The Github link.

thank you !.