House Price Prediction

Avesta Narimani
12/ 12/ 2023

Data

Source: Kaggel → House Prices - Advanced Regression Techniques

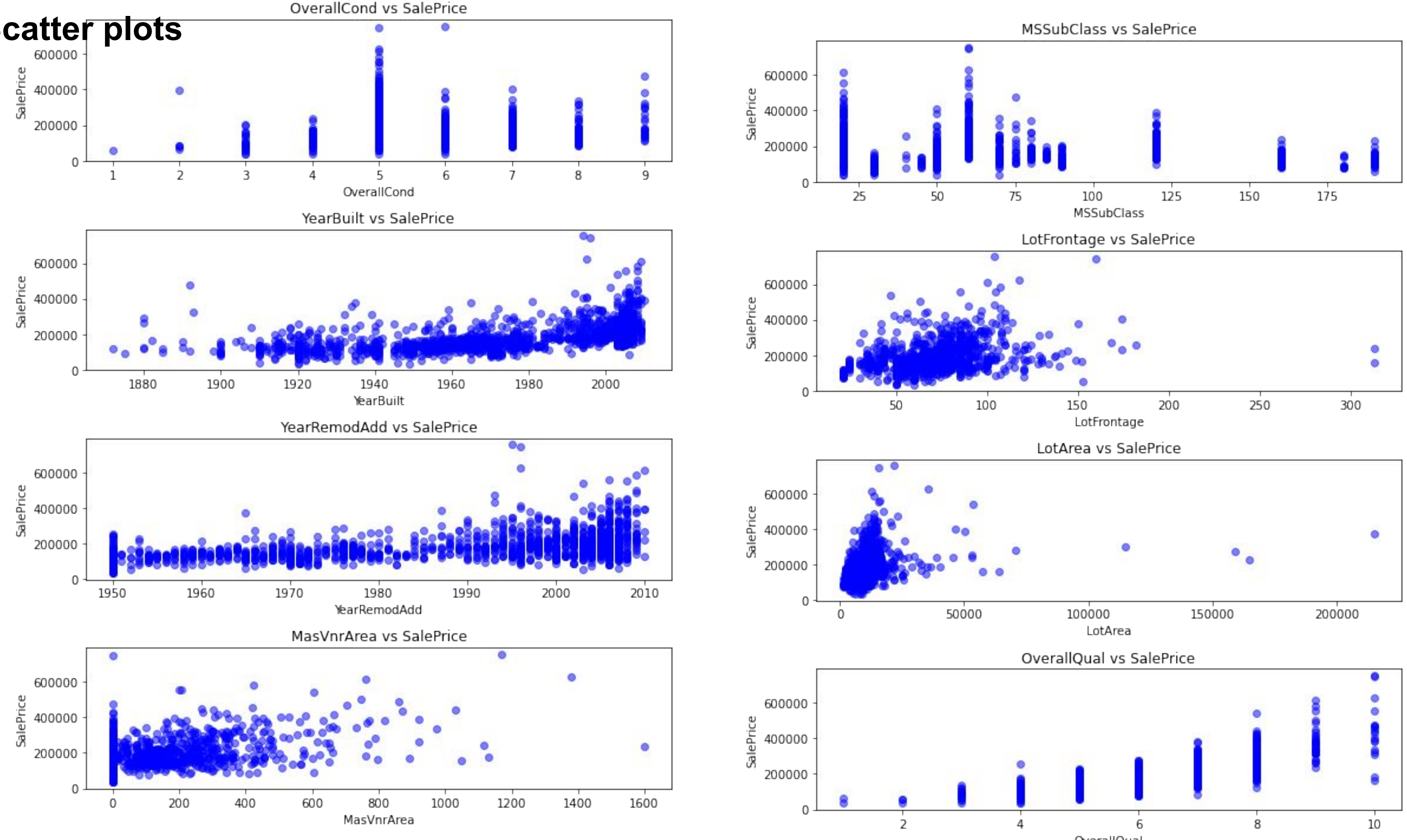
Df.shape = (1460, 80)

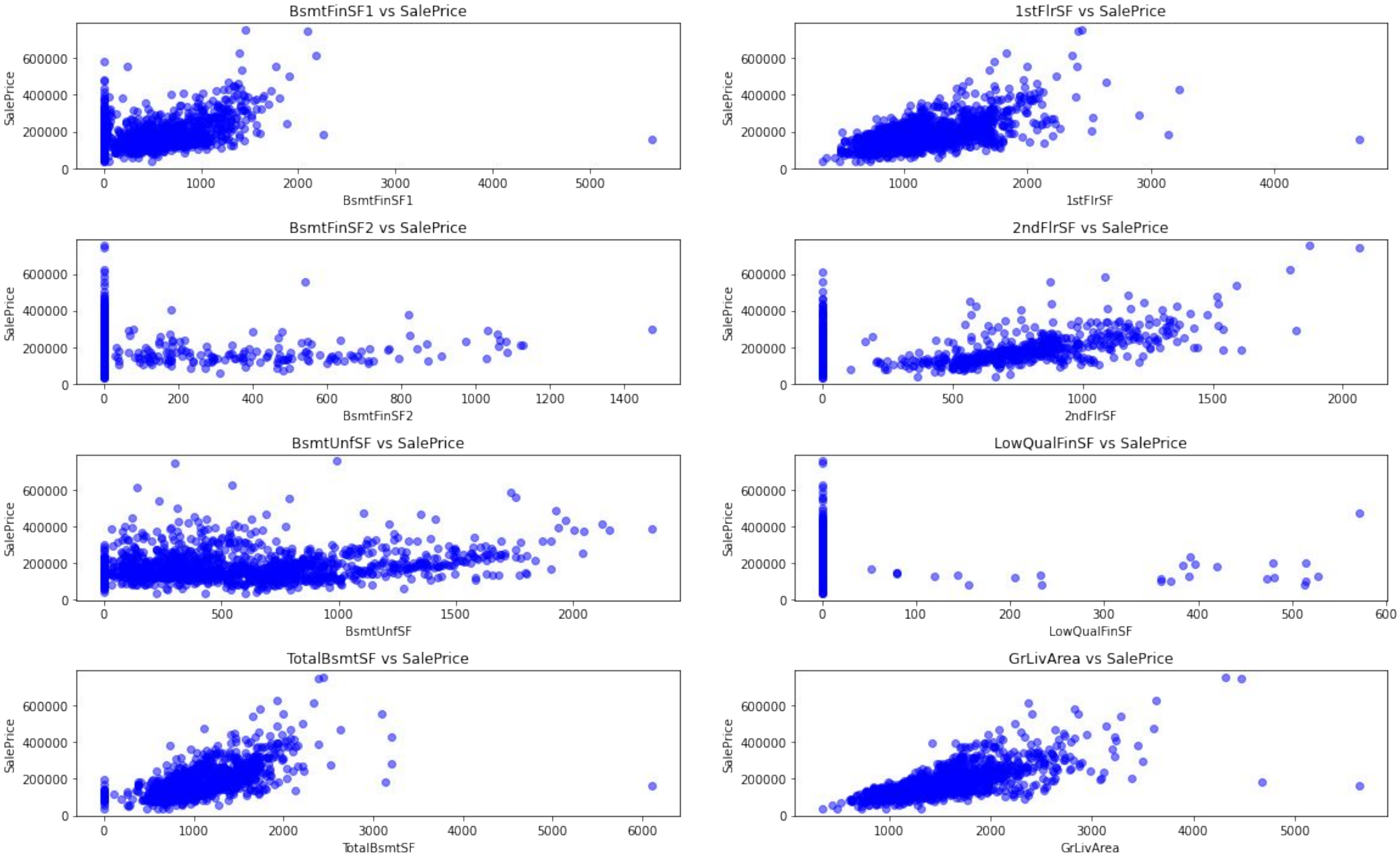
The target variable is SalePrice.

There are 37 numeric columns.

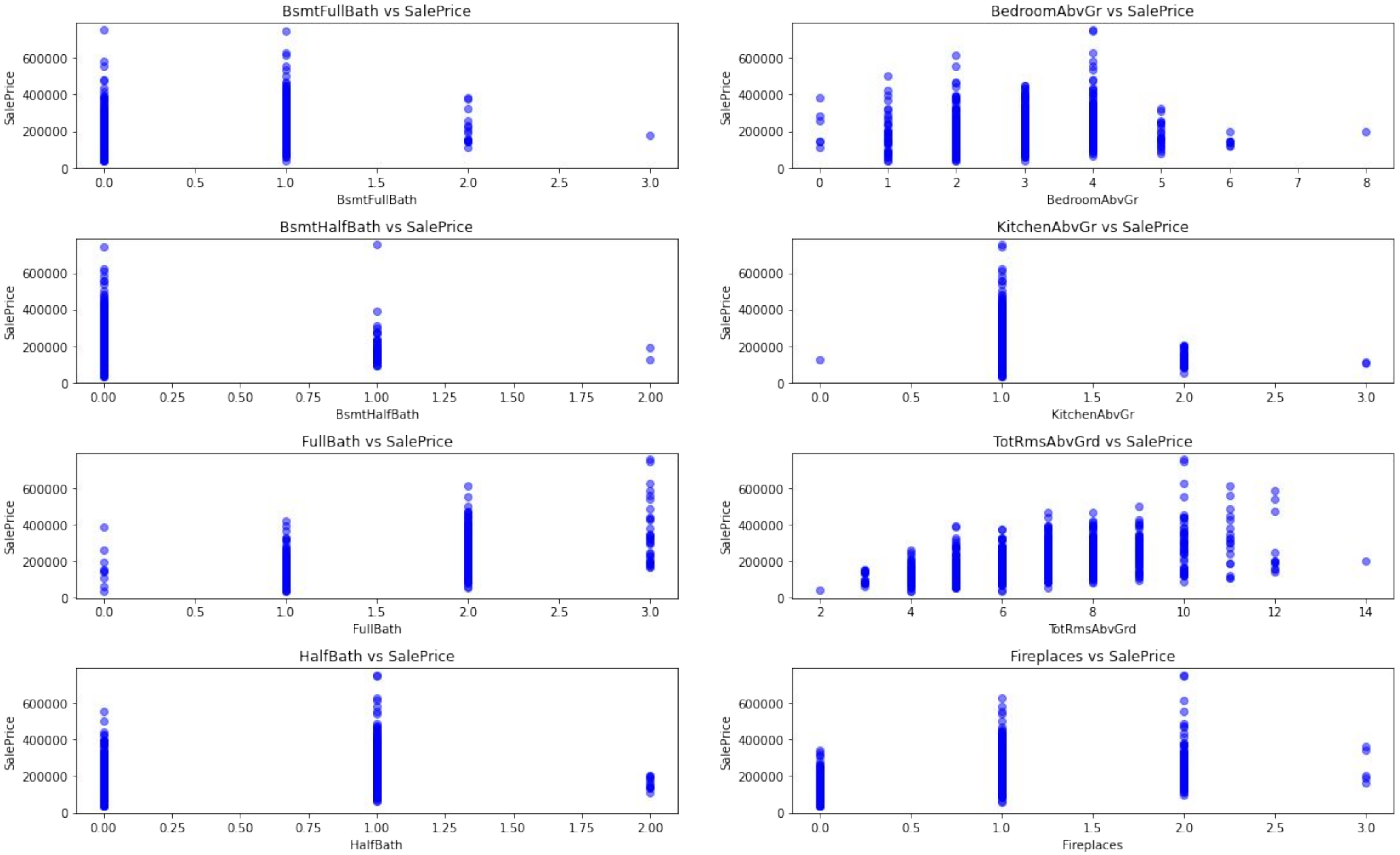
There are 43 categorical columns.

Scatter plots





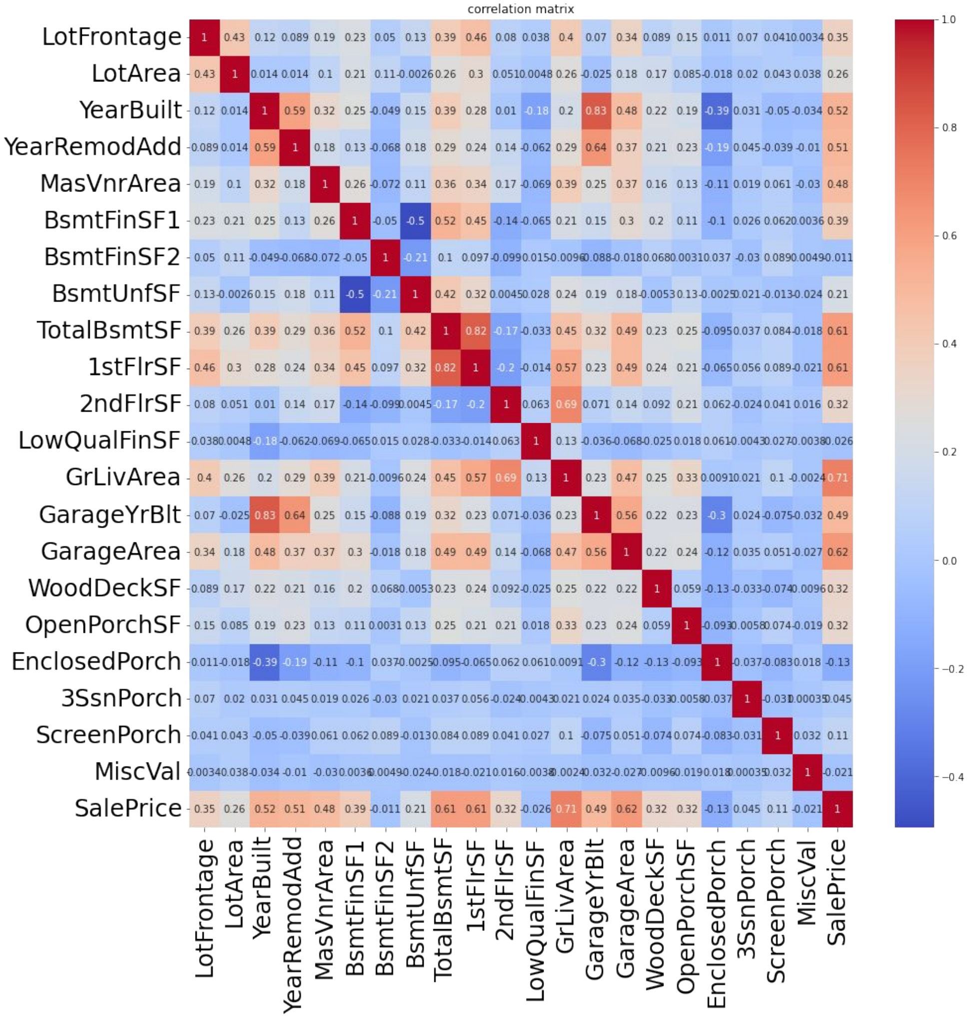
BsmtFinSF1 vs SalePrice
SalePrice
BsmtFinSF1
1stFlrSF vs SalePrice
SalePrice
1stFlrSF
BsmtFinSF2 vs SalePrice
SalePrice
BsmtFinSF2
2ndFlrSF vs SalePrice
SalePrice
2ndFlrSF
BsmtUnfSF vs SalePrice
SalePrice
BsmtUnfSF
LowQualFinSF vs SalePrice
SalePrice
LowQualFinSF
TotalBsmtSF vs SalePrice
SalePrice
TotalBsmtSF
GrLivArea vs SalePrice
SalePrice
GrLivArea


BsmtFullBath vs SalePrice
SalePrice
BsmtFullBath
BedroomAbvGr vs SalePrice
SalePrice
BedroomAbvGr
BsmtHalfBath vs SalePrice
SalePrice
BsmtHalfBath
KitchenAbvGr vs SalePrice
SalePrice
KitchenAbvGr
FullBath vs SalePrice
SalePrice
FullBath
TotRmsAbvGrd vs SalePrice
SalePrice
TotRmsAbvGrd
HalfBath vs SalePrice
SalePrice
HalfBath
Fireplaces vs SalePrice
SalePrice
Fireplaces

Collinearity

As the correlation matrix indicates, so many features are correlated.

There are three collections of columns which are correlated.



Correlation between features including year:

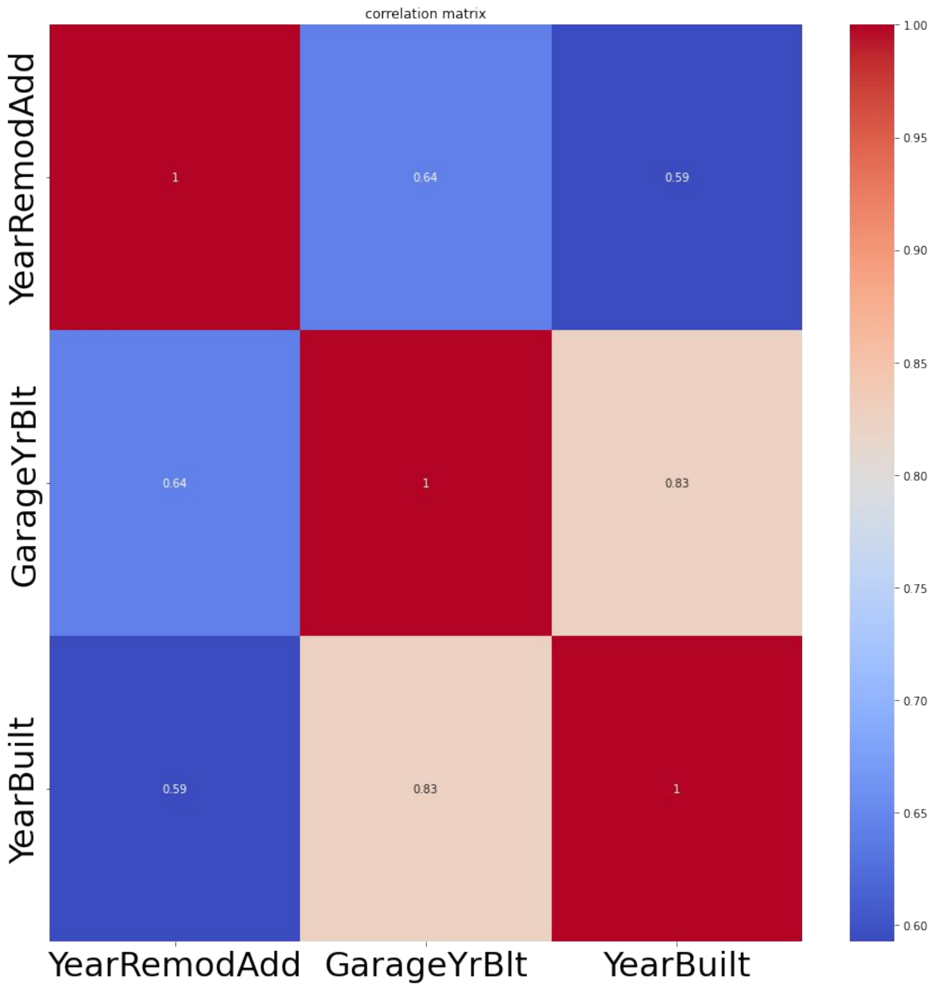
The following columns are correlated:

['YearRemodAdd', 'GarageYrBlt',

"YearBuilt"]

Decision:

Drop 'YearRemodAdd', 'GarageYrBlt'.



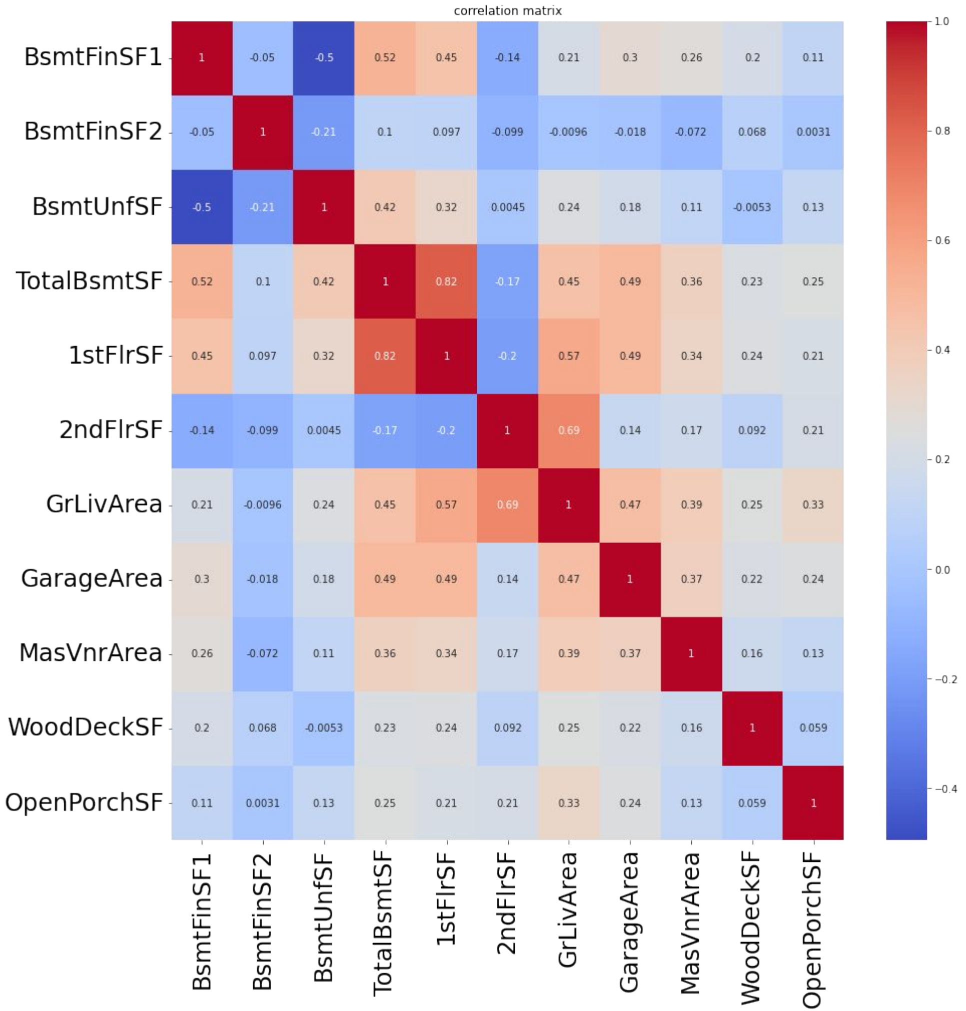
Correlation between feature of area type:

The following columns are correlated:

['BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GrLivArea', 'GarageArea', 'MasVnrArea', 'WoodDeckSF', 'OpenPorchSF']#, "LotArea", "LotFrontage"]

Decision:

Make a new feature by adding them up.



Feature engineering: define a new feature based on summation

```
# columns_of_int = ['BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GrLivArea',
'GarageArea', 'MasVnrArea', "LotArea", "LotFrontage"]

columns_of_int = col_of_int

df['Bsm'] = 0

for col in columns_of_int:

    df['Bsm'] += df[col]

df.drop(columns_of_int, axis=1, inplace=True)


print(df['Bsm'])
```

Correlation between feature of Lot type:

The following features are

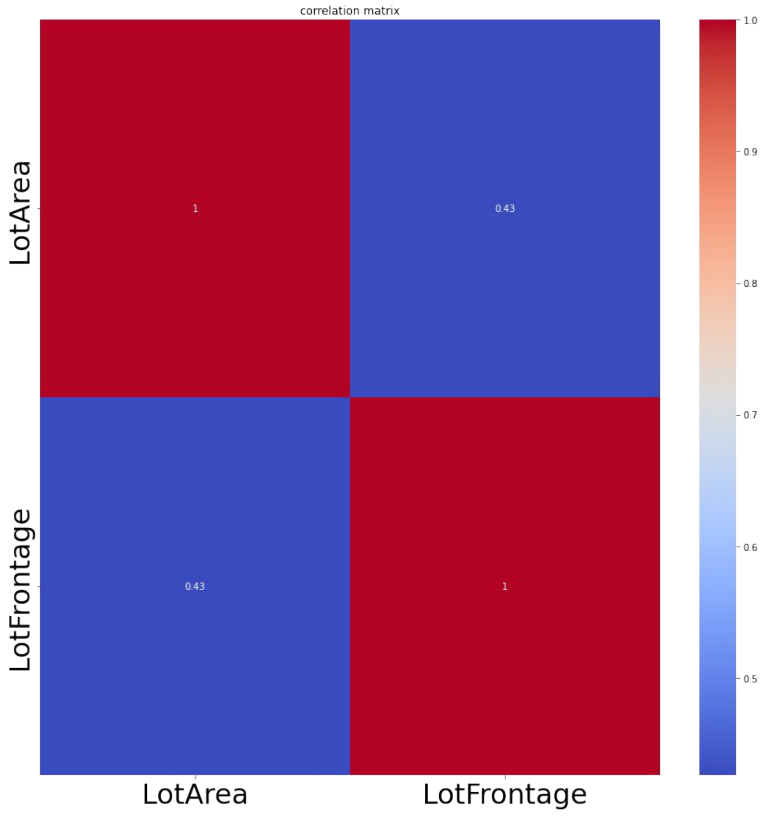
also correlated

col_of_int = ["LotArea", "LotFrontage"]

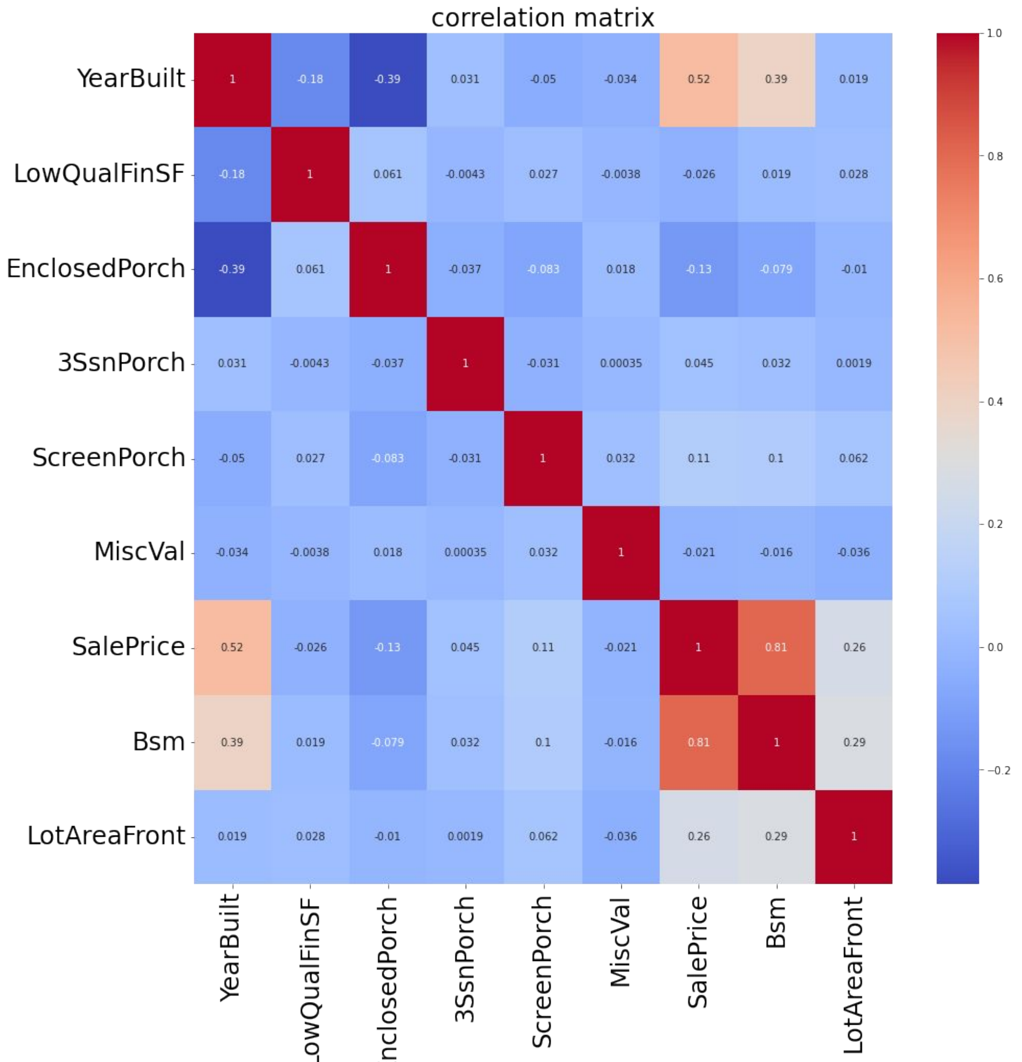
Decision:

Make a new feature by adding them up:

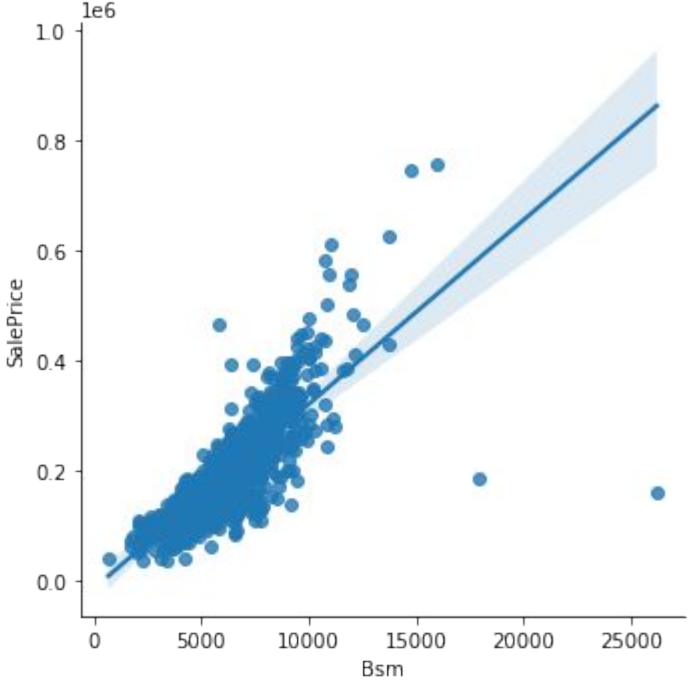
New feature: LotAreaFront

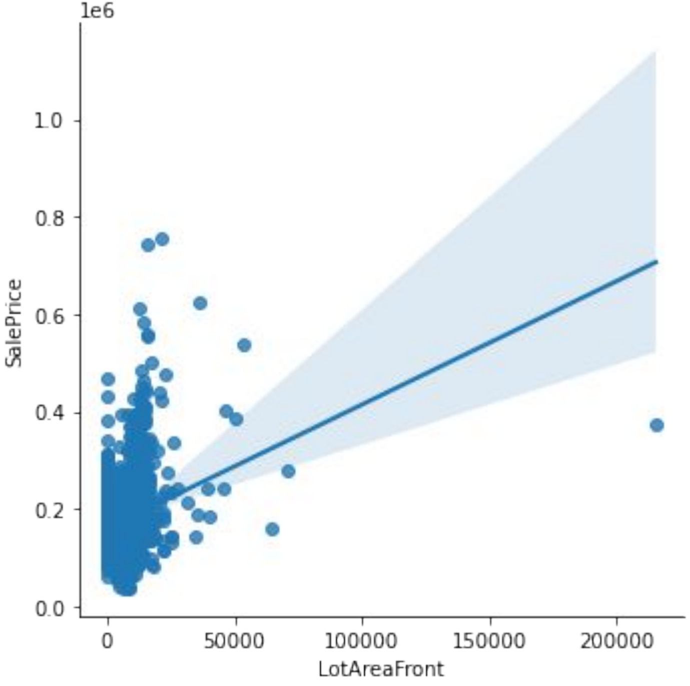


The new correlation matrix:



Examining the new features





Hypothesis Testing: Investigating the Relationship between Building Square Meters, Lot Area Frontage, and Sale Price

- **R-squared (0.718)**: Approximately 71.8% of the variability in `SalePrice` is explained by `Bsm` and `LotAreaFront`. This indicates a strong model.
- **Adjusted R-squared (0.717)**: Adjusted for the number of predictors, still indicates a good fit.

Coefficients Analysis

- *`Bsm`* :
  - **Coefficient**: 32.9027
  - **P-value**: < 0.0001
  - **Interpretation**: Every additional square meter in building's basment size increases the sale price by approximately 32.9 units, assuming `LotAreaFront` is constant. Statistically significant.
- *`LotAreaFront`* :
  - **Coefficient**: 0.7486
  - **P-value**: 0.003
  - **Interpretation**: Every additional linear foot of this parameter increases the sale price by approximately 0.75 units, assuming `Bsm` is constant. Statistically significant.

Multicollinearity Concerns

- **Condition Number (3.73e+04)**: High condition number suggests potential multicollinearity, which could affect coefficient reliability.

Dealing with outliers

In order the improve the model outliers were removed according to the following criteria:

multiplier = 2.5

Q1 = df[true_numeric_columns].quantile(0.25)

Q3 = df[true_numeric_columns].quantile(0.75)

IQR = Q3 - Q1

lower_bound = Q1 - multiplier * IQR

upper_bound = Q3 + multiplier * IQR

Dealing with columns with numeric values containing NaN

df['LotAreaFront'].fillna(0, inplace=True)

df['Bsm'].fillna(df['Bsm'].mean(), inplace=True)

Further steps for categorical columns

- Replace NaN with “N/A”
- Created dummy variables

Before applying any model:

- Data splitting
- Standard scaler

Models

Model	RMSE
Mean	62834.4
Median	62113.1
Linear Regression	10283858416.3
Decision Tree Regression	31200.8
Decision Tree Regression **Hyperparameter tuning**	28947.3
Random Forest Regression (RFR)	22059.9
RFR hyperparameter tuning	22245.3
Gradient Boosting Regression (GBR)	20177.0
GBR hyperparameter tuning	19536.5
XGBoost hyperparameter tuning	19971.9

Conclusion

Gradient Boosting Regression was the best model of all.