House_pricing

About the data set

💡 House prices is regression data set about: Predict sales prices and practice feature engineering, RFs, and gradient boosting, of training set is 81 column and test set is 79 column .

Cleaning and Preparing the data

- Feature Selection
- Dealing with missing data
- Dealing with outliers
- Feature Encoding
- Model Building and Enhancing
  - Linear Regression
  - KNN Regressor

Information about DataFrame

```
# reading the dataset
houses = pd.read_csv("DataSet/train.csv")
display(f"shape of the dataset {houses.shape}", houses.head(3))
```

[2] Python

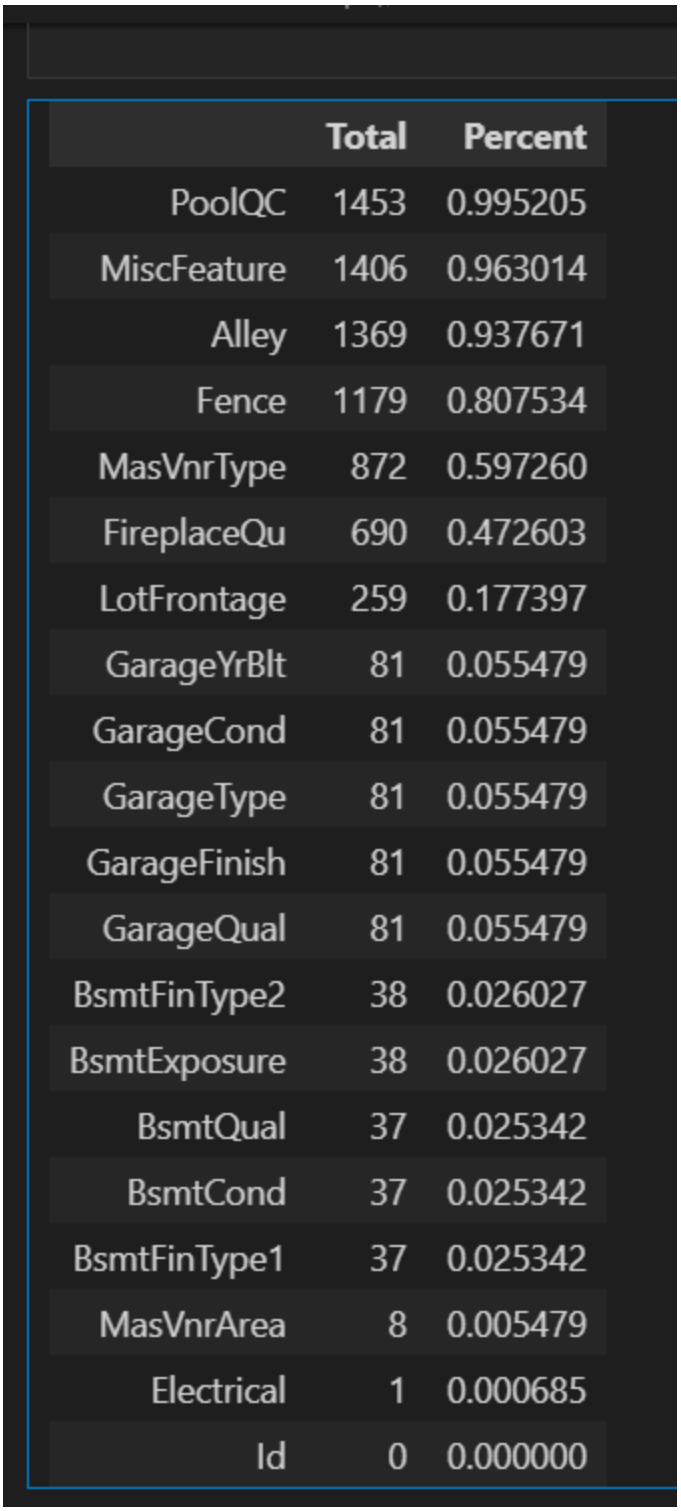
'shape of the dataset (1460, 81)'

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN

3 rows × 81 columns

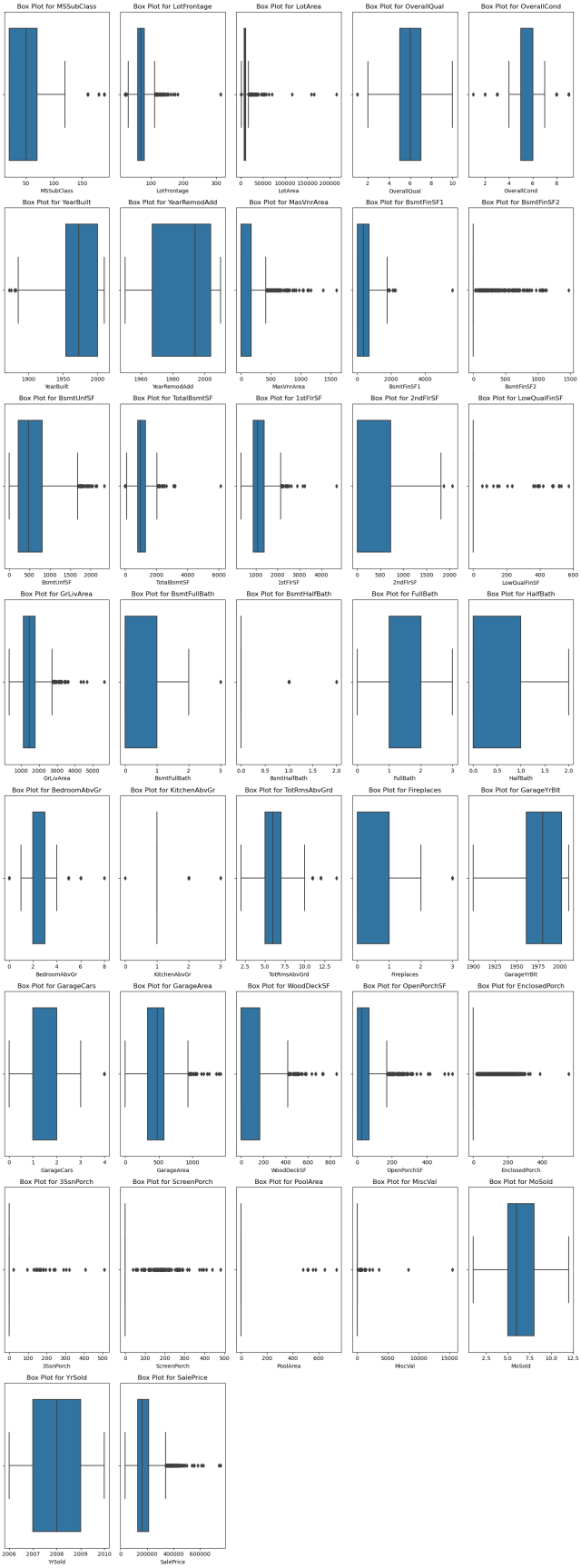
Detecting missing data

There are 19 classes that contain Missing value



	Total	Percent
PoolQC	1453	0.995205
MiscFeature	1406	0.963014
Alley	1369	0.937671
Fence	1179	0.807534
MasVnrType	872	0.597260
FireplaceQu	690	0.472603
LotFrontage	259	0.177397
GarageYrBlt	81	0.055479
GarageCond	81	0.055479
GarageType	81	0.055479
GarageFinish	81	0.055479
GarageQual	81	0.055479
BsmtFinType2	38	0.026027
BsmtExposure	38	0.026027
BsmtQual	37	0.025342
BsmtCond	37	0.025342
BsmtFinType1	37	0.025342
MasVnrArea	8	0.005479
Electrical	1	0.000685
Id	0	0.000000

Detecting outliers


Box Plot for MSSubClass
MSSubClass
Box Plot for LotFrontage
LotFrontage
Box Plot for LotArea
LotArea
Box Plot for OverallQual
OverallQual
Box Plot for OverallCond
OverallCond
Box Plot for YearBuilt
YearBuilt
Box Plot for YearRemodAdd
YearRemodAdd
Box Plot for MasVnrArea
MasVnrArea
Box Plot for BsmtFinSF1
BsmtFinSF1
Box Plot for BsmtFinSF2
BsmtFinSF2
Box Plot for BsmtUnfSF
BsmtUnfSF
Box Plot for TotalBsmtSF
TotalBsmtSF
Box Plot for 1stFlrSF
1stFlrSF
Box Plot for 2ndFlrSF
2ndFlrSF
Box Plot for LowQualFinSF
LowQualFinSF
Box Plot for GrLivArea
GrLivArea
Box Plot for BsmtFullBath
BsmtFullBath
Box Plot for BsmtHalfBath
BsmtHalfBath
Box Plot for FullBath
FullBath
Box Plot for HalfBath
HalfBath
Box Plot for BedroomAbvGr
BedroomAbvGr
Box Plot for KitchenAbvGr
KitchenAbvGr
Box Plot for TotRmsAbvGrd
TotRmsAbvGrd
Box Plot for Fireplaces
Fireplaces
Box Plot for GarageYrBlt
GarageYrBlt
Box Plot for GarageCars
GarageCars
Box Plot for GarageArea
GarageArea
Box Plot for WoodDeckSF
WoodDeckSF
Box Plot for OpenPorchSF
OpenPorchSF
Box Plot for EnclosedPorch
EnclosedPorch
Box Plot for 3SsnPorch
3SsnPorch
Box Plot for ScreenPorch
ScreenPorch
Box Plot for PoolArea
PoolArea
Box Plot for MiscVal
MiscVal
Box Plot for MoSold
MoSold
Box Plot for YrSold
YrSold
Box Plot for SalePrice
SalePrice

Dealing with outliers

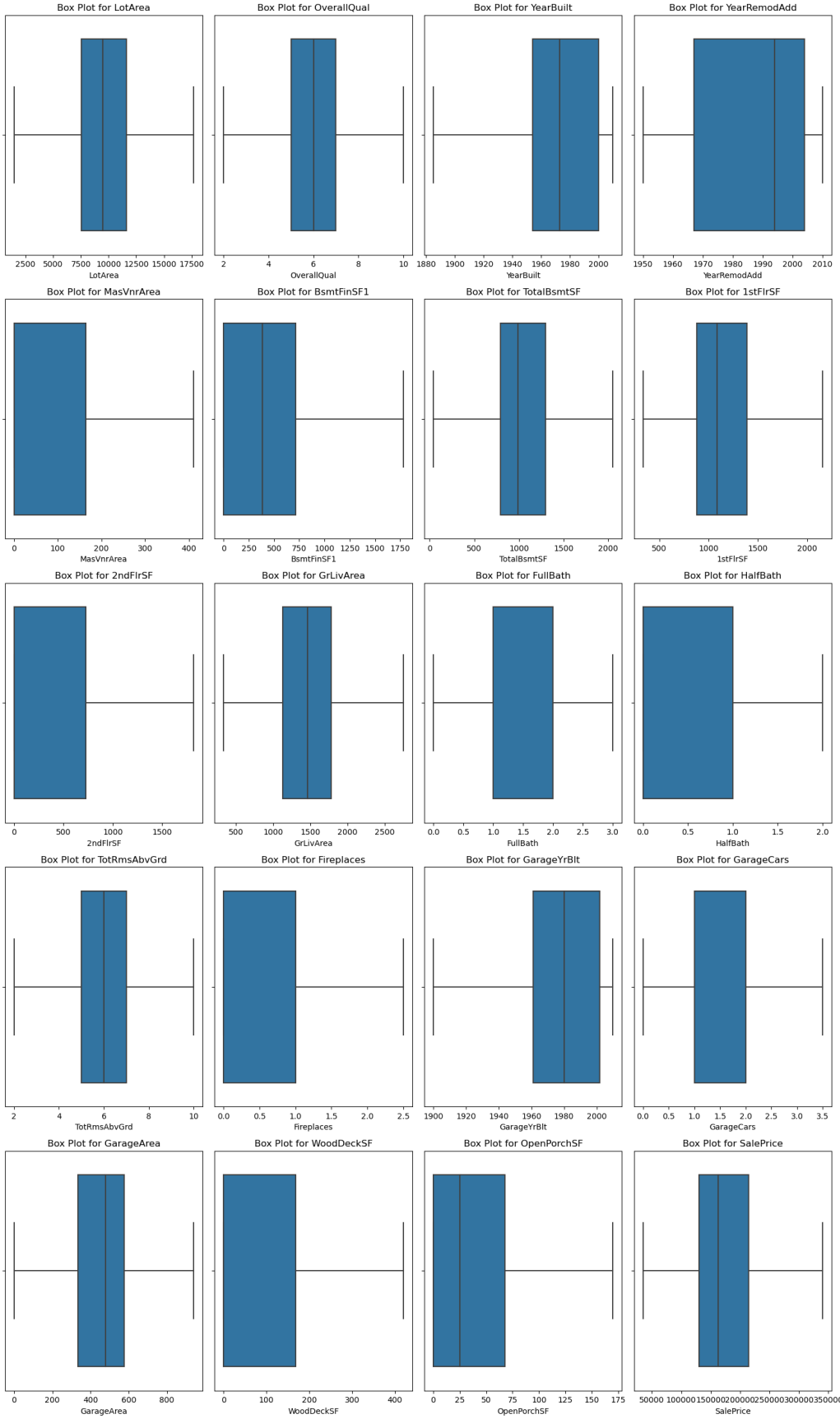
Handle outliers using (IQR) :

IQR=Q3-Q1

lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR

then replace outliers with lower and upper

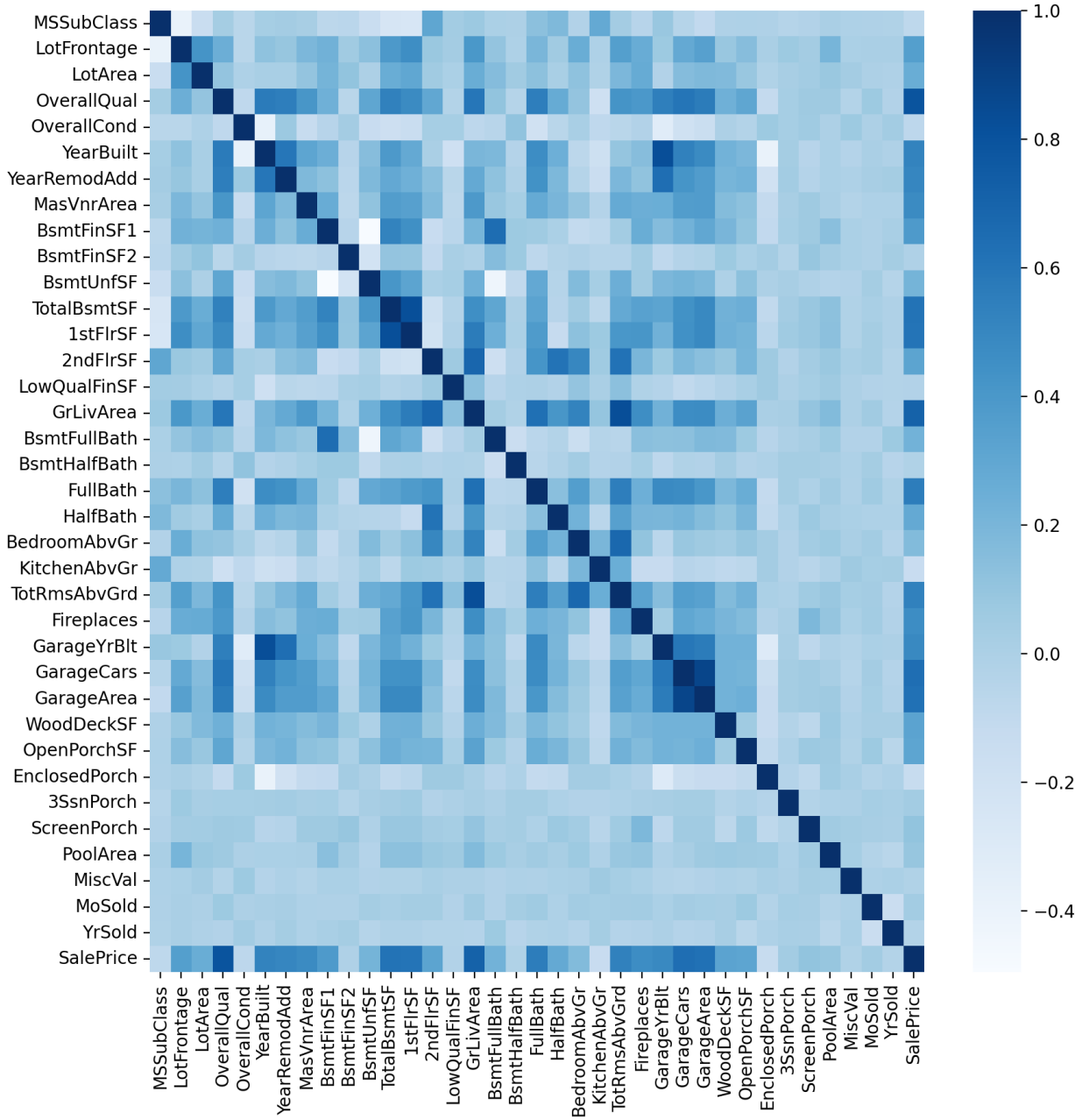
visualization outliers


Box Plot for LotArea
LotArea
Box Plot for OverallQual
OverallQual
Box Plot for YearBuilt
YearBuilt
Box Plot for YearRemodAdd
YearRemodAdd
Box Plot for MasVnrArea
MasVnrArea
Box Plot for BsmtFinSF1
BsmtFinSF1
Box Plot for TotalBsmtSF
TotalBsmtSF
Box Plot for 1stFlrSF
1stFlrSF
Box Plot for 2ndFlrSF
2ndFlrSF
Box Plot for GrLivArea
GrLivArea
Box Plot for FullBath
FullBath
Box Plot for HalfBath
HalfBath
Box Plot for TotRmsAbvGrd
TotRmsAbvGrd
Box Plot for Fireplaces
Fireplaces
Box Plot for GarageYrBlt
GarageYrBlt
Box Plot for GarageCars
GarageCars
Box Plot for GarageArea
GarageArea
Box Plot for WoodDeckSF
WoodDeckSF
Box Plot for OpenPorchSF
OpenPorchSF
Box Plot for SalePrice
SalePrice

Correlation analysis

first drop column id and use datatype numerical only and visualize it using ” heatmap”

output is:number of column that have text ⇒ (37, 1)



Cleaning and Preparing the data

Feature Selection

First:drop data and delete weak correlation ,and select column that contain null value greater than 81 and drop it (remove 7 feature)

Output: **before delete weak correlation ⇒ (1459, 74)**
after delete weak correlation ⇒ (1459, 58)

Fillna in each numerical cloumn

first select numerical value and drop id column ,then fillna with "random_sample"

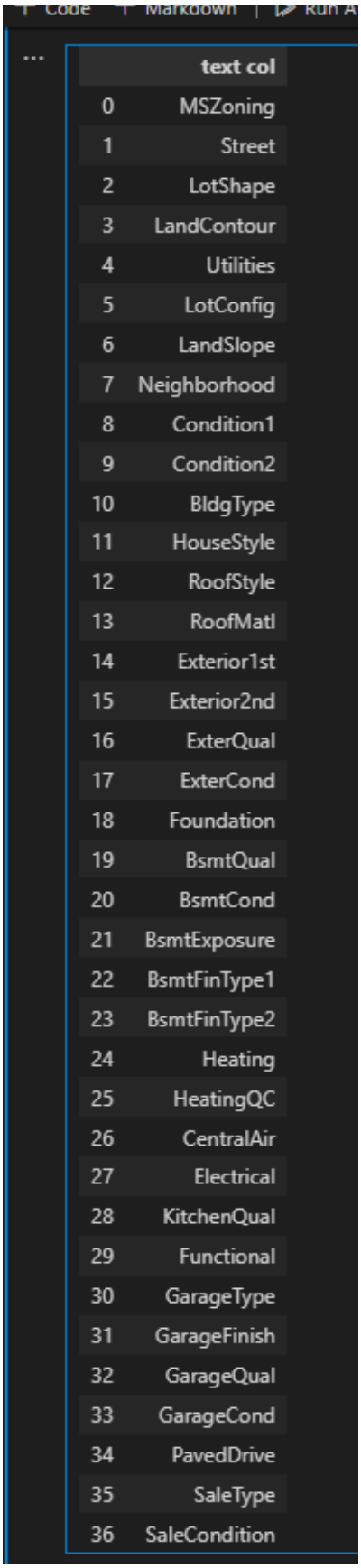
second select object value and replace it with "mode":

```
(mode_for_coll = houses[col].mode()[0]
 houses[col].fillna(mode_for_coll, inplace=True)
)
```

Feature Encoding

First select object and count it

Output:number of column that have text ⇒ (37, 1)



	text col
0	MSZoning
1	Street
2	LotShape
3	LandContour
4	Utilities
5	LotConfig
6	LandSlope
7	Neighborhood
8	Condition1
9	Condition2
10	BldgType
11	HouseStyle
12	RoofStyle
13	RoofMatl
14	Exterior1st
15	Exterior2nd
16	ExterQual
17	ExterCond
18	Foundation
19	BsmtQual
20	BsmtCond
21	BsmtExposure
22	BsmtFinType1
23	BsmtFinType2
24	Heating
25	HeatingQC
26	CentralAir
27	Electrical
28	KitchenQual
29	Functional
30	GarageType
31	GarageFinish
32	GarageQual
33	GarageCond
34	PavedDrive
35	SaleType
36	SaleCondition

Transform object to number using “labelenconder”

Column Non-Null Count Dtype

Column	Non-Null Count	Dtype
Id	1459 non-null	int64
MSZoning	1459 non-null	int32
LotArea	1459 non-null	int64
Street	1459 non-null	int32
LotShape	1459 non-null	int32
LandContour	1459 non-null	int32
Utilities	1459 non-null	int32
LotConfig	1459 non-null	int32
LandSlope	1459 non-null	int32
Neighborhood	1459 non-null	int32
Condition1	1459 non-null	int32
Condition2	1459 non-null	int32
BldgType	1459 non-null	int32
HouseStyle	1459 non-null	int32
OverallQual	1459 non-null	int64
YearBuilt	1459 non-null	int64
YearRemodAdd	1459 non-null	float64
RoofStyle	1459 non-null	int32
RoofMatl	1459 non-null	int32
Exterior1st	1459 non-null	int32
Exterior2nd	1459 non-null	int32
MasVnrArea	1459 non-null	float64
ExterQual	1459 non-null	int32
ExterCond	1459 non-null	int32
Foundation	1459 non-null	int32
BsmtQual	1459 non-null	int32

BsmtCond	1459 non-null	int32
BsmtExposure	1459 non-null	int32
BsmtFinType1	1459 non-null	int32
BsmtFinSF1	1459 non-null	float64
BsmtFinType2	1459 non-null	int32
TotalBsmtSF	1459 non-null	float64
Heating	1459 non-null	int32
HeatingQC	1459 non-null	int32
CentralAir	1459 non-null	int32
Electrical	1459 non-null	int32
1stFlrSF	1459 non-null	float64
2ndFlrSF	1459 non-null	int64
GrLivArea	1459 non-null	float64
FullBath	1459 non-null	float64
HalfBath	1459 non-null	float64
KitchenQual	1459 non-null	int32
TotRmsAbvGrd	1459 non-null	int64
Functional	1459 non-null	int32
Fireplaces	1459 non-null	float64
GarageType	1459 non-null	int32
GarageYrBlt	1459 non-null	float64
GarageFinish	1459 non-null	int32
GarageCars	1459 non-null	float64
GarageArea	1459 non-null	float64
GarageQual	1459 non-null	int32
GarageCond	1459 non-null	int32
PavedDrive	1459 non-null	int32
WoodDeckSF	1459 non-null	int64
OpenPorchSF	1459 non-null	int64

SaleType	1459 non-null	int32
SaleCondition	1459 non-null	int32
SalePrice	1459 non-null	int64
dtypes	float64(12), int32(37), int64(9)	memory usage: 461.6 KB

Model Building and Enhancing

```
# split data to features and target
X = houses.iloc[:, :-1].values
y = houses.iloc[:, -1:].values
# split data to train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size =0
```

standardscaler

using it to removes the mean and scales each feature/variable to unit variance

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler(copy=True,with_std=True)
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Support vector Regressor

```
from sklearn.svm import SVR

svr = SVR(kernel="linear", C=100)
svr.fit(X_train, y_train)
```

```
y_pred = svr.predict(X_test)
```

score

```
# score for train
print(f"train score: {svr.score(X_train, y_train)}")
# score for test
print(f"test score: {svr.score(X_test, y_test)}")
```

output:

train score: 0.8702118477654037
test score: 0.901781188013509

Errors:

```
# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)  # RMSE
print(f"Mean Absolute Error: {mae}")
print(f"Mean Squared Error: {mse}")
print(f"Root Mean Squared Error: {rmse}")

r2 = r2_score(y_test, y_pred)
print(f"Coefficient of Determination (R^2): {r2}")
```

Final accuracy

after handling error and use linear regression model

```
r2 = r2_score(y_test, y_pred)
print(f"Coefficient of Determination (R^2): {r2}")
```

Mean Absolute Error: 16513.75805104947
Mean Squared Error:
484870712.0657572
Root Mean Squared Error:
22019.780018559613
Coefficient of Determination (R^2)
: 0.901781188013509