Printout

07 February 2023 17:11

In [1]:
```
#import librararies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Goal of the Project

```
Predict the price of a house by its features. If you are a buyer or seller of the house but
you don't know the exact price of the house, so supervised machine learning regression
algorithms can help you to predict the price of the house just providing features of the
target house.
```

Load The Dataset

In [3]:
```
test=pd.read_csv(r"test.csv")
train=pd.read_csv(r"train.csv")
```

In [4]:
```
test.head()
```

Out[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	S
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	...	
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	...	
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	...	
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	...	
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	...	

5 rows × 80 columns

In [5]:
```
train.tail()
```

Out[5]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	..
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	..
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	..
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	..
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	..
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	..

5 rows × 81 columns

In [6]:
```
test.isnull().any().sum()
```

Out[6]:
```
33
```

In [7]:
```
1 train.isnull().sum()
```

Out[7]:
```
Id                0
MSSubClass        0
MSZoning          0
LotFrontage     259
LotArea           0
                 ...
MoSold            0
YrSold            0
SaleType          0
SaleCondition     0
SalePrice         0
Length: 81, dtype: int64
```

In [8]:
```
1 pd.set_option("display.max_columns",None)
2 pd.set_option("display.max_rows",None)
```

In [9]:
```
1 train.isnull().sum()
```

Out[9]:
```
Id                0
MSSubClass        0
MSZoning          0
LotFrontage     259
LotArea           0
Street            0
Alley          1369
LotShape          0
LandContour       0
Utilities         0
LotConfig         0
LandSlope         0
Neighborhood      0
Condition1        0
Condition2        0
BldgType          0
HouseStyle        0
OverallQual       0
OverallCond       0
YearBuilt         0
YearRemodAdd      0
RoofStyle         0
RoofMatl          0
Exterior1st       0
Exterior2nd       0
MasVnrType        8
MasVnrArea        8
ExterQual         0
ExterCond         0
Foundation        0
BsmtQual         37
BsmtCond         37
BsmtExposure     38
BsmtFinType1     37
BsmtFinSF1        0
BsmtFinType2     38
BsmtFinSF2        0
BsmtUnfSF         0
TotalBsmtSF       0
Heating           0
HeatingQC         0
CentralAir        0
Electrical        1
1stFlrSF          0
2ndFlrSF          0
LowQualFinSF      0
GrLivArea         0
BsmtFullBath      0
BsmtHalfBath      0
FullBath          0
HalfBath          0
BedroomAbvGr      0
KitchenAbvGr      0
KitchenQual       0
TotRmsAbvGrd      0
Functional        0
Fireplaces        0
FireplaceQu     690
GarageType       81
GarageYrBlt      81
GarageFinish     81
GarageCars        0
GarageArea        0
GarageQual       81
GarageCond       81
PavedDrive        0
WoodDeckSF        0
OpenPorchSF       0
EnclosedPorch     0
3SsnPorch         0
ScreenPorch       0
PoolArea          0
PoolQC         1453
```

```
Fence           1179
MiscFeature     1406
MiscVal            0
MoSold             0
YrSold             0
SaleType           0
SaleCondition      0
SalePrice          0
dtype: int64
```

In [10]:
```
train
```

Out[10]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilitie
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPu
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPu
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPu
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPu
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPu
5	6	50	RL	85.0	14115	Pave	NaN	IR1	Lvl	AllPu
6	7	20	RL	75.0	10084	Pave	NaN	Reg	Lvl	AllPu
7	8	60	RL	NaN	10382	Pave	NaN	IR1	Lvl	AllPu
8	9	50	RM	51.0	6120	Pave	NaN	Reg	Lvl	AllPu
9	10	190	RL	50.0	7420	Pave	NaN	Reg	Lvl	AllPu

In [11]:
```
test.head()
```

Out[11]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotC
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	C
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	

In [12]:
```
test["MSSubClass"].isnull().sum()
```

Out[12]: 0

In [13]:
```
train["MSZoning"].isnull().sum()
```

Out[13]: 0

In [14]:
```
train["Fence"].notnull().sum()
```

Out[14]: 281

In [15]:
```
train["Fence"].isnull().sum()
```

Out[15]: 1179

In [16]:
```
test.MSSubClass.isnull().any()
```

Out[16]: False

In [17]:
```
df=pd.concat([train,test])
print(df.shape)
```

```
(2919, 81)
```

In [18]:
```
pd.set_option("display.max_columns",None)
pd.set_option("display.max_rows",None)
```

In [19]:
```
df.head()
```

Out[19]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotCon
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Ins
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	F
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Ins
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Cor
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	F

In [20]:
```
df.tail()
```

Out[20]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	L
1454	2915	160	RM	21.0	1936	Pave	NaN	Reg	Lvl	AllPub	
1455	2916	160	RM	21.0	1894	Pave	NaN	Reg	Lvl	AllPub	
1456	2917	20	RL	160.0	20000	Pave	NaN	Reg	Lvl	AllPub	
1457	2918	85	RL	62.0	10441	Pave	NaN	Reg	Lvl	AllPub	
1458	2919	60	RL	74.0	9627	Pave	NaN	Reg	Lvl	AllPub	

In [21]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2919 entries, 0 to 1458
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             2919 non-null   int64
 1   MSSubClass     2919 non-null   int64
 2   MSZoning       2915 non-null   object
 3   LotFrontage    2433 non-null   float64
 4   LotArea        2919 non-null   int64
 5   Street         2919 non-null   object
 6   Alley          198 non-null    object
 7   LotShape       2919 non-null   object
 8   LandContour    2919 non-null   object
 9   Utilities      2917 non-null   object
 10  LotConfig      2919 non-null   object
 11  LandSlope      2919 non-null   object
 12  Neighborhood   2919 non-null   object
 13  Condition1     2919 non-null   object
 14  Condition2     2919 non-null   object
 15  BldgType       2919 non-null   object
 16  HouseStyle     2919 non-null   object
 17  OverallQual    2919 non-null   int64
 18  OverallCond    2919 non-null   int64
 19  YearBuilt      2919 non-null   int64
 20  YearRemodAdd   2919 non-null   int64
 21  RoofStyle      2919 non-null   object
 22  RoofMatl       2919 non-null   object
 23  Exterior1st    2918 non-null   object
 24  Exterior2nd    2918 non-null   object
 25  MasVnrType     2895 non-null   object
 26  MasVnrArea     2896 non-null   float64
 27  ExterQual      2919 non-null   object
 28  ExterCond      2919 non-null   object
 29  Foundation     2919 non-null   object
 30  BsmtQual       2838 non-null   object
 31  BsmtCond       2837 non-null   object
 32  BsmtExposure   2837 non-null   object
 33  BsmtFinType1   2840 non-null   object
 34  BsmtFinSF1     2918 non-null   float64
 35  BsmtFinType2   2839 non-null   object
 36  BsmtFinSF2     2918 non-null   float64
 37  BsmtUnfSF      2918 non-null   float64
 38  TotalBsmtSF    2918 non-null   float64
 39  Heating        2919 non-null   object
 40  HeatingQC      2919 non-null   object
 41  CentralAir     2919 non-null   object
 42  Electrical     2918 non-null   object
 43  1stFlrSF       2919 non-null   int64
 44  2ndFlrSF       2919 non-null   int64
 45  LowQualFinSF   2919 non-null   int64
 46  GrLivArea      2919 non-null   int64
 47  BsmtFullBath   2917 non-null   float64
 48  BsmtHalfBath   2917 non-null   float64
 49  FullBath       2919 non-null   int64
 50  HalfBath       2919 non-null   int64
 51  BedroomAbvGr   2919 non-null   int64
 52  KitchenAbvGr   2919 non-null   int64
 53  KitchenQual    2918 non-null   object
 54  TotRmsAbvGrd   2919 non-null   int64
 55  Functional     2917 non-null   object
 56  Fireplaces     2919 non-null   int64
 57  FireplaceQu    1499 non-null   object
 58  GarageType     2762 non-null   object
 59  GarageYrBlt    2760 non-null   float64
 60  GarageFinish   2760 non-null   object
 61  GarageCars     2918 non-null   float64
 62  GarageArea     2918 non-null   float64
 63  GarageQual     2760 non-null   object
 64  GarageCond     2760 non-null   object
 65  PavedDrive     2919 non-null   object
 66  WoodDeckSF     2919 non-null   int64
 67  OpenPorchSF    2919 non-null   int64
```

```
 68  EnclosedPorch  2919 non-null   int64
 69  3SsnPorch      2919 non-null   int64
 70  ScreenPorch    2919 non-null   int64
 71  PoolArea       2919 non-null   int64
 72  PoolQC         10 non-null     object
 73  Fence          571 non-null    object
 74  MiscFeature    105 non-null    object
 75  MiscVal        2919 non-null   int64
 76  MoSold         2919 non-null   int64
 77  YrSold         2919 non-null   int64
 78  SaleType       2918 non-null   object
 79  SaleCondition  2919 non-null   object
 80  SalePrice      1460 non-null   float64
dtypes: float64(12), int64(26), object(43)
memory usage: 1.8+ MB
```

Most Null Features

In [ ]:
```
1
```

```
#### Most Null Features
1)Alley          198 non-null    object
2)FireplaceQu    1499 non-null   object
3)PoolQC         10 non-null     object
4)Fence          571 non-null    object
5)MiscFeature    105 non-null    object
```

In [22]:
```
df.head()
```

Out[22]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotCon
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Ins
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	F
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Ins
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Cor
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	F

In [23]:
```
df.set_index("Id")

```

Out[23]:

Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	Lot
1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	
2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	
3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	
4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	
5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	
6	50	RL	85.0	14115	Pave	NaN	IR1	Lvl	AllPub	
7	20	RL	75.0	10084	Pave	NaN	Reg	Lvl	AllPub	
8	60	RL	NaN	10382	Pave	NaN	IR1	Lvl	AllPub	
9	50	RM	51.0	6120	Pave	NaN	Reg	Lvl	AllPub	

INTGER FEATURES

In [24]:
```
int_features=df.select_dtypes(include=["int64"]).columns
print(len(int_features))
print(int_features)
```

```
26
Index(['Id', 'MSSubClass', 'LotArea', 'OverallQual', 'OverallCond',
       'YearBuilt', 'YearRemodAdd', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr',
       'TotRmsAbvGrd', 'Fireplaces', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal',
       'MoSold', 'YrSold'],
      dtype='object')
```

FLOAT FEATURES

In [25]:
```
float_features=df.select_dtypes(include=["float64"]).keys()
print(len(float_features))
print(float_features)
```

```
12
Index(['LotFrontage', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF',
       'TotalBsmtSF', 'BsmtFullBath', 'BsmtHalfBath', 'GarageYrBlt',
       'GarageCars', 'GarageArea', 'SalePrice'],
      dtype='object')
```

In [26]:
```
float_features=df.select_dtypes(include=["float"]).keys()
len(float_features)
```

Out[26]: 12

CATGORICAL FEATURES

In [27]:
```
cat_features=df.select_dtypes(include=['object']).columns
print(len(cat_features))
print(cat_features)
```

```
43
Index(['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities',
       'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2',
       'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st',
       'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation',
       'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2',
       'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual',
       'Functional', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual',
       'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature',
       'SaleType', 'SaleCondition'],
      dtype='object')
```

In [28]:
```
1 df.LotFrontage
```

Out[28]:
```
0       65.0
1       80.0
2       68.0
3       60.0
4       84.0
5       85.0
6       75.0
7        NaN
8       51.0
9       50.0
10      70.0
11      85.0
12       NaN
13      91.0
14       NaN
15      51.0
16       NaN
17      72.0
18      66.0
```

In [29]:
```
1 print(int_features)
```

```
Index(['Id', 'MSSubClass', 'LotArea', 'OverallQual', 'OverallCond',
       'YearBuilt', 'YearRemodAdd', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr',
       'TotRmsAbvGrd', 'Fireplaces', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal',
       'MoSold', 'YrSold'],
      dtype='object')
```

NULL COUNT

In [30]:
```
1 null_count=df.isnull().sum()
2 null_count
```

Out[30]:
```
Id                0
MSSubClass        0
MSZoning          4
LotFrontage     486
LotArea           0
Street            0
Alley          2721
LotShape          0
LandContour       0
Utilities         2
LotConfig         0
LandSlope         0
Neighborhood      0
Condition1        0
Condition2        0
BldgType          0
HouseStyle        0
OverallQual       0
OverallCond       0
YearBuilt         0
YearRemodAdd      0
RoofStyle         0
RoofMatl          0
Exterior1st       1
Exterior2nd       1
MasVnrType       24
MasVnrArea       23
ExterQual         0
ExterCond         0
Foundation        0
BsmtQual         81
BsmtCond         82
BsmtExposure     82
BsmtFinType1     79
BsmtFinSF1        1
BsmtFinType2     80
BsmtFinSF2        1
BsmtUnfSF         1
TotalBsmtSF       1
Heating           0
HeatingQC         0
CentralAir        0
Electrical        1
1stFlrSF          0
2ndFlrSF          0
LowQualFinSF      0
GrLivArea         0
BsmtFullBath      2
BsmtHalfBath      2
FullBath          0
HalfBath          0
BedroomAbvGr      0
KitchenAbvGr      0
KitchenQual       1
TotRmsAbvGrd      0
Functional        2
Fireplaces        0
FireplaceQu    1420
GarageType      157
GarageYrBlt     159
GarageFinish    159
GarageCars        1
GarageArea        1
GarageQual      159
GarageCond      159
PavedDrive        0
WoodDeckSF        0
OpenPorchSF       0
EnclosedPorch     0
3SsnPorch         0
ScreenPorch       0
PoolArea          0
```

```
PoolQC           2909
Fence            2348
MiscFeature      2814
MiscVal             0
MoSold              0
YrSold              0
SaleType            1
SaleCondition       0
SalePrice        1459
dtype: int64
```

In [31]:
```
null_value_perc=(df.isnull().sum()/df.shape[0])*100
null_value_perc
```

Out[31]:

Id	0.000000
MSSubClass	0.000000
MSZoning	0.137033
LotFrontage	16.649538
LotArea	0.000000
Street	0.000000
Alley	93.216855
LotShape	0.000000
LandContour	0.000000
Utilities	0.068517
LotConfig	0.000000
LandSlope	0.000000
Neighborhood	0.000000
Condition1	0.000000
Condition2	0.000000
BldgType	0.000000
HouseStyle	0.000000
OverallQual	0.000000
OverallCond	0.000000
YearBuilt	0.000000
YearRemodAdd	0.000000
RoofStyle	0.000000
RoofMatl	0.000000
Exterior1st	0.034258
Exterior2nd	0.034258
MasVnrType	0.822199
MasVnrArea	0.787941
ExterQual	0.000000
ExterCond	0.000000
Foundation	0.000000
BsmtQual	2.774923
BsmtCond	2.809181
BsmtExposure	2.809181
BsmtFinType1	2.706406
BsmtFinSF1	0.034258
BsmtFinType2	2.740665
BsmtFinSF2	0.034258
BsmtUnfSF	0.034258
TotalBsmtSF	0.034258
Heating	0.000000
HeatingQC	0.000000
CentralAir	0.000000
Electrical	0.034258
1stFlrSF	0.000000
2ndFlrSF	0.000000
LowQualFinSF	0.000000
GrLivArea	0.000000
BsmtFullBath	0.068517
BsmtHalfBath	0.068517
FullBath	0.000000
HalfBath	0.000000
BedroomAbvGr	0.000000
KitchenAbvGr	0.000000
KitchenQual	0.034258
TotRmsAbvGrd	0.000000
Functional	0.068517
Fireplaces	0.000000
FireplaceQu	48.646797
GarageType	5.378554
GarageYrBlt	5.447071
GarageFinish	5.447071
GarageCars	0.034258
GarageArea	0.034258
GarageQual	5.447071
GarageCond	5.447071
PavedDrive	0.000000
WoodDeckSF	0.000000
OpenPorchSF	0.000000
EnclosedPorch	0.000000
3SsnPorch	0.000000
ScreenPorch	0.000000
PoolArea	0.000000

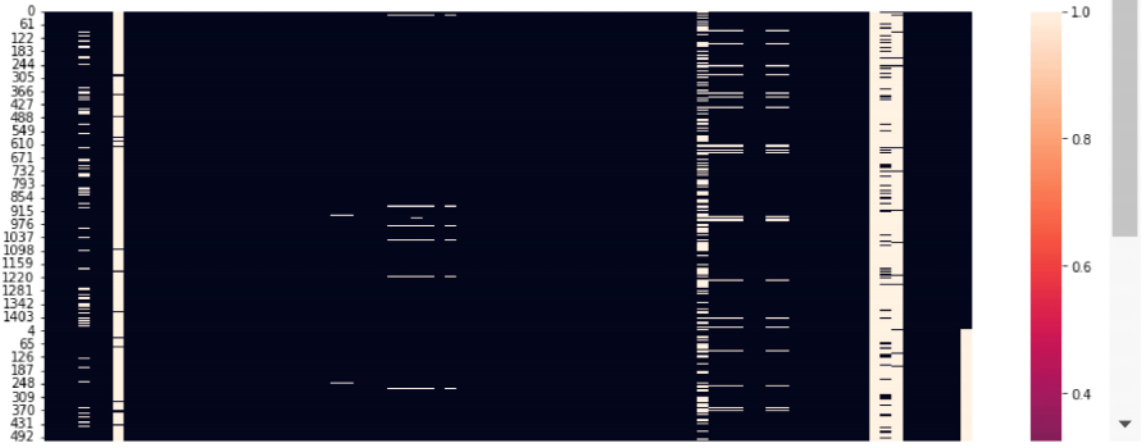
```
PoolQC           99.657417
Fence            80.438506
MiscFeature      96.402878
MiscVal           0.000000
MoSold            0.000000
YrSold            0.000000
SaleType          0.034258
SaleCondition     0.000000
SalePrice        49.982871
dtype: float64
```

In [32]:
```
float_features
```

Out[32]:
```
Index(['LotFrontage', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF',
       'TotalBsmtSF', 'BsmtFullBath', 'BsmtHalfBath', 'GarageYrBlt',
       'GarageCars', 'GarageArea', 'SalePrice'],
      dtype='object')
```

In [33]:
```
plt.figure(figsize=(16,9))
sns.heatmap(df.isnull())
```

Out[33]: <AxesSubplot:>



Drop Columns/Features

In [34]:
```
#### as per domain knowledge we will not drop those feature insted none value we will add consta
missing_50_perc_value=null_value_perc[null_value_perc>50]
missing_50_perc_value
```

Out[34]:
```
Alley          93.216855
PoolQC         99.657417
Fence          80.438506
MiscFeature    96.402878
dtype: float64
```

In [35]:
```
df.Alley.value_counts()
```

Out[35]:
```
Grvl    120
Pave     78
Name: Alley, dtype: int64
```

In [36]:
```
#### as per domain knowledge we will not drop those feature insted none value we will add consta
missing_20_50_perc_value=null_value_perc[(null_value_perc>20)&(null_value_perc<50)]
missing_20_50_perc_value
```

Out[36]:
```
FireplaceQu    48.646797
SalePrice      49.982871
dtype: float64
```

In [40]:
```
1 miss_5_20_perc_value=null_value_perc[(null_value_perc>5)&(null_value_perc<20)]
2 miss_5_20_perc_value
```

Out[40]:
```
LotFrontage     16.649538
GarageType       5.378554
GarageYrBlt      5.447071
GarageFinish     5.447071
GarageQual       5.447071
GarageCond       5.447071
dtype: float64
```

In [41]:
```
1 df.LotFrontage
```

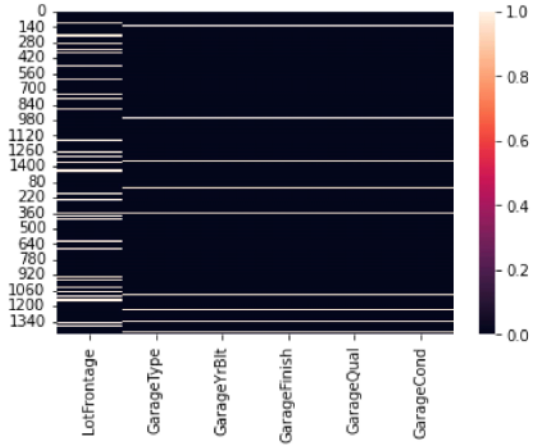
Out[41]:
```
0        65.0
1        80.0
2        68.0
3        60.0
4        84.0
5        85.0
6        75.0
7         NaN
8        51.0
9        50.0
10       70.0
11       85.0
12        NaN
13       91.0
14        NaN
15       51.0
16        NaN
17       72.0
18       66.0
```

In [42]:
```
1 df.LotFrontage.value_counts().head()
```

Out[42]:
```
60.0    276
80.0    137
70.0    133
50.0    117
75.0    105
Name: LotFrontage, dtype: int64
```

In [43]:
```
1 sns.heatmap(df[miss_5_20_perc_value.keys()].isnull())
```

Out[43]: <AxesSubplot:>



Mising Value Imputation

In [44]:
```
missing_value_feartures=null_value_perc[null_value_perc>0]
len(missing_value_feartures)
missing_value_feartures
```

Out[44]:
```
MSZoning         0.137033
LotFrontage     16.649538
Alley           93.216855
Utilities        0.068517
Exterior1st      0.034258
Exterior2nd      0.034258
MasVnrType       0.822199
MasVnrArea       0.787941
BsmtQual         2.774923
BsmtCond         2.809181
BsmtExposure     2.809181
BsmtFinType1     2.706406
BsmtFinSF1       0.034258
BsmtFinType2     2.740665
BsmtFinSF2       0.034258
BsmtUnfSF        0.034258
TotalBsmtSF      0.034258
Electrical       0.034258
BsmtFullBath     0.068517
BsmtHalfBath     0.068517
KitchenQual      0.034258
Functional       0.068517
FireplaceQu     48.646797
GarageType       5.378554
GarageYrBlt      5.447071
GarageFinish     5.447071
GarageCars       0.034258
GarageArea       0.034258
GarageQual       5.447071
GarageCond       5.447071
PoolQC          99.657417
Fence           80.438506
MiscFeature     96.402878
SaleType         0.034258
SalePrice       49.982871
dtype: float64
```

CATEGORICAL NA FEATURES

In [45]:
```
cat_na_feat=missing_value_feartures[missing_value_feartures.keys()]
cat_na_feat
```

Out[45]:
```
MSZoning         0.137033
LotFrontage     16.649538
Alley           93.216855
Utilities        0.068517
Exterior1st      0.034258
Exterior2nd      0.034258
MasVnrType       0.822199
MasVnrArea       0.787941
BsmtQual         2.774923
BsmtCond         2.809181
BsmtExposure     2.809181
BsmtFinType1     2.706406
BsmtFinSF1       0.034258
BsmtFinType2     2.740665
BsmtFinSF2       0.034258
BsmtUnfSF        0.034258
TotalBsmtSF      0.034258
Electrical       0.034258
BsmtFullBath     0.068517
BsmtHalfBath     0.068517
KitchenQual      0.034258
Functional       0.068517
FireplaceQu     48.646797
GarageType       5.378554
GarageYrBlt      5.447071
GarageFinish     5.447071
GarageCars       0.034258
GarageArea       0.034258
GarageQual       5.447071
GarageCond       5.447071
PoolQC          99.657417
Fence           80.438506
MiscFeature     96.402878
SaleType         0.034258
SalePrice       49.982871
dtype: float64
```

In [46]:
```
cat_na_feat=missing_value_feartures[missing_value_feartures.keys().isin(cat_features)]
print("Total Number Of Missing Categorical Features :",len(cat_na_feat))
cat_na_feat
```

```
Total Number Of Missing Categorical Features : 23
```

Out[46]:
```
MSZoning         0.137033
Alley           93.216855
Utilities        0.068517
Exterior1st      0.034258
Exterior2nd      0.034258
MasVnrType       0.822199
BsmtQual         2.774923
BsmtCond         2.809181
BsmtExposure     2.809181
BsmtFinType1     2.706406
BsmtFinType2     2.740665
Electrical       0.034258
KitchenQual      0.034258
Functional       0.068517
FireplaceQu     48.646797
GarageType       5.378554
GarageFinish     5.447071
GarageQual       5.447071
GarageCond       5.447071
PoolQC          99.657417
Fence           80.438506
MiscFeature     96.402878
SaleType         0.034258
dtype: float64
```

In [ ]:
```
1
```

INTGER NA FEATURES

In [47]:
```
1 int_na_feat=missing_value_feartures[missing_value_feartures.keys().isin(int_features)]
2 print(f"Total Number Of Missing Numerical Features :{len(int_na_feat)}")
```

```
Total Number Of Missing Numerical Features :0
```

In [48]:
```
1 int_na_feat
```

Out[48]:
```
Series([], dtype: float64)
```

FLOAT NA FEATURES

In [49]:
```
1 float_na_feat=missing_value_feartures[missing_value_feartures.keys().isin(float_features)]
2 print(f"Total Number Of Missing Floating Features :{len(float_na_feat)}")
```

```
Total Number Of Missing Floating Features :12
```

In [50]:
```
1 float_na_feat
```

Out[50]:
```
LotFrontage     16.649538
MasVnrArea       0.787941
BsmtFinSF1       0.034258
BsmtFinSF2       0.034258
BsmtUnfSF        0.034258
TotalBsmtSF      0.034258
BsmtFullBath     0.068517
BsmtHalfBath     0.068517
GarageYrBlt      5.447071
GarageCars       0.034258
GarageArea       0.034258
SalePrice       49.982871
dtype: float64
```

Handling MSZONING Features

In [51]:
```
1 df.MSZoning.value_counts()
```

Out[51]:
```
RL         2265
RM          460
FV          139
RH           26
C (all)      25
Name: MSZoning, dtype: int64
```

In [52]:
```
1 # df.MSZoning
```

In [53]:
```
1 df_mvi=df.copy()
2 df_mvi.shape
```

Out[53]:
```
(2919, 81)
```

In [54]:
```
1 df_mvi=df.copy()
2 df_mvi.shape
```

Out[54]:
```
(2919, 81)
```

HERE WE HAVE REPACE NA VALUE BY MODE DUE TO THE DOMAIN KNOWLEDGE RA more time occurs VALUE IN THIS MSZONING FEATURES

In [55]:
```
mszoning_mode=df.MSZoning.mode()[0]
df_mvi.MSZoning.replace(np.nan,mszoning_mode,inplace=True)
df_mvi.MSZoning.isnull().sum().sum()
```

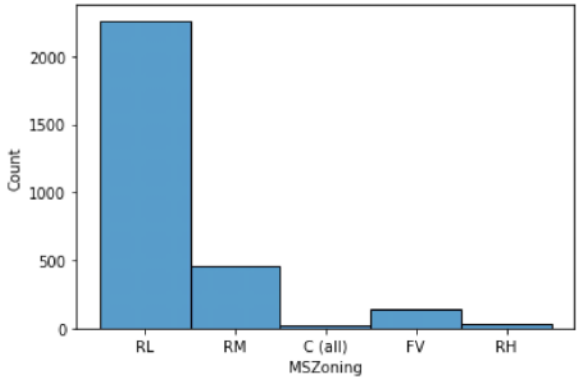
Out[55]: 0

In [56]:
```
df.shape
```

Out[56]: (2919, 81)

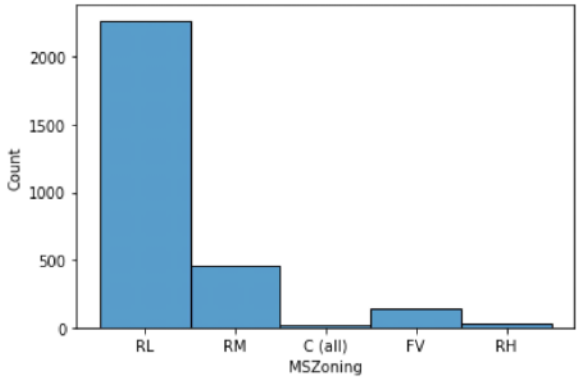
In [57]:
```
sns.histplot(df["MSZoning"])
```

Out[57]: <AxesSubplot:xlabel='MSZoning', ylabel='Count'>



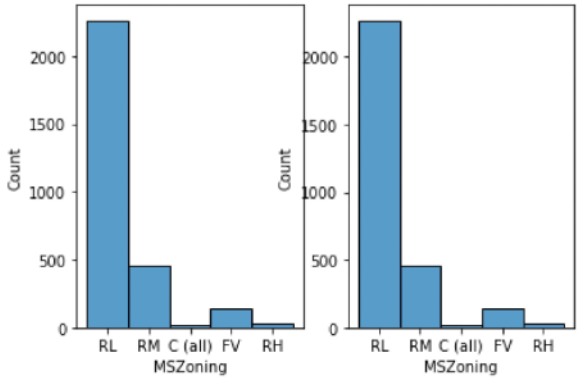
In [58]:
```
sns.histplot(df_mvi["MSZoning"])
```

Out[58]: <AxesSubplot:xlabel='MSZoning', ylabel='Count'>



In [59]:
```
def new_old_hist(df,df_new,features):
    plt.subplot(121)
    sns.histplot(df[features])
    plt.subplot(122)
    bins=len(df_mvi[features].unique())
    sns.histplot(df_mvi[features],cbar=True,bins=True)
```

In [60]:
```
1 new_old_hist(df,df_mvi,'MSZoning')
```



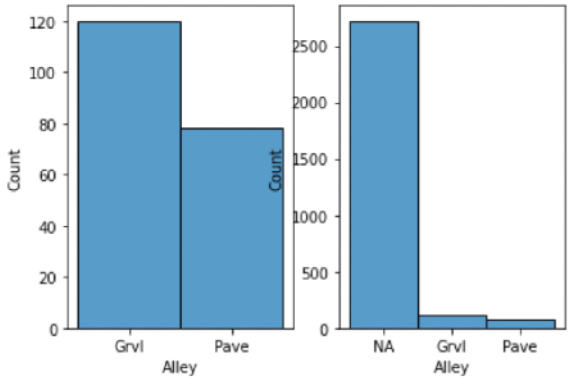
Handalling Alley Features

WHY WE ARE REPLACE THE NAN VALUE BY NA BECUASE THE ALLLEY IS MOST IMPORTANT FEATUREST AS PER OUR DOMAIN KONWLEGE

In [61]:
```
1 alley_count="NA"
2 df_mvi.Alley.replace(np.nan,alley_count,inplace=True)
3 df_mvi.Alley.isnull().sum()
```

Out[61]: 0

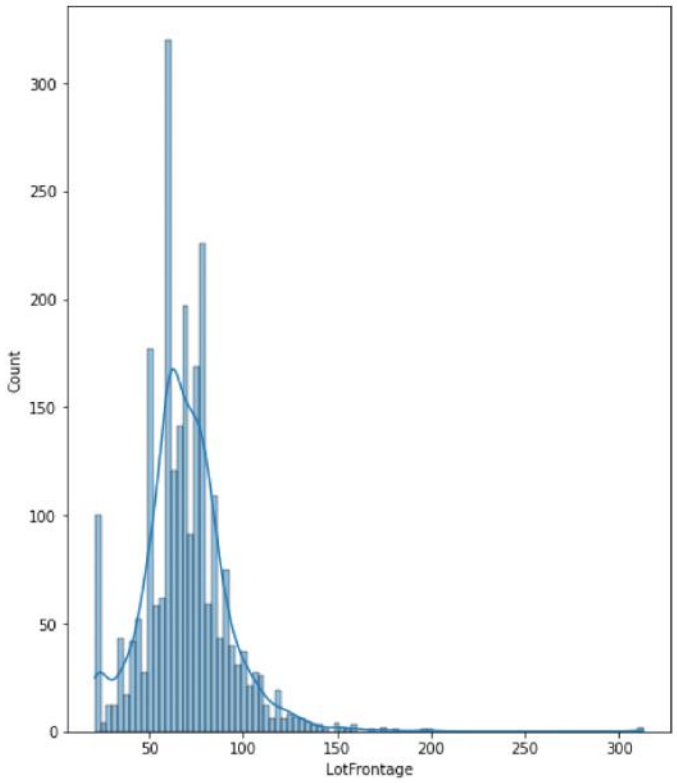
In [62]:
```
1 new_old_hist(df,df_mvi,'Alley')
```



In [63]:
```
1 def box_hist_plot(df):
2     plt.figure(figsize=(16,9))
3     plt.subplot(121)
4     sns.histplot(df,kde=True)
```

Handling LotFrontage Features

In [64]:
```
box_hist_plot(df["LotFrontage"])
#here we see the data is right skewed hence we fill the median of data on the plase of nan value
```

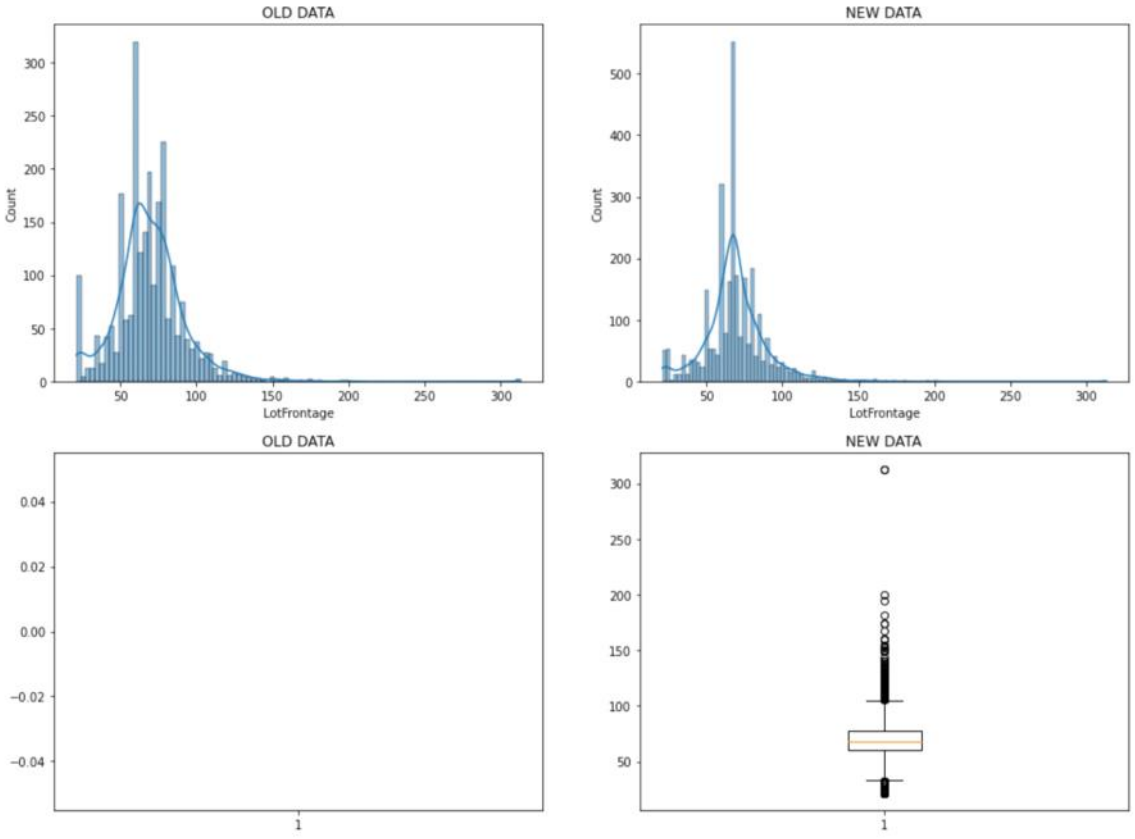


In [65]:
```
lotfrnt_median=df.LotFrontage.median()
df_mvi["LotFrontage"].replace(np.nan,lotfrnt_median,inplace=True)
df_mvi["LotFrontage"].isnull().sum()
```

Out[65]: 0

In [66]:
```
def box_cat_plot(df,df_new,features):
    plt.figure(figsize=(16,12))
    plt.subplot(221)
    plt.title("OLD DATA")
    sns.histplot(df[features],kde=True)
    plt.subplot(222)
    plt.title("NEW DATA")
    sns.histplot(df_new[features],kde=True)
    plt.subplot(223)
    plt.title("OLD DATA")
    plt.boxplot(df[features])
    plt.subplot(224)
    plt.title("NEW DATA")
    plt.boxplot(df_new[features])
```

In [67]:
```
1 box_cat_plot(df,df_mvi,"LotFrontage")
```



Handling Utilities Features

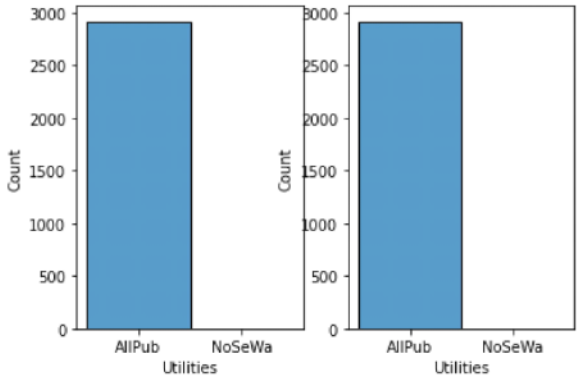
In [68]:
```
1 df.Utilities.value_counts()
```

Out[68]:
```
AllPub    2916
NoSeWa       1
Name: Utilities, dtype: int64
```

In [69]:
```
1 utilities_mode=df.Utilities.mode()[0]
2 df_mvi.Utilities.replace(np.nan,utilities_mode,inplace=True)
3 df_mvi.Utilities.isnull().sum().sum()
```

Out[69]:
```
0
```

In [70]:
```
1 new_old_hist(df,df_mvi,'Utilities')
```



In [71]:
```
1 df_mvi.Utilities
```

Out[71]:
```
0       AllPub
1       AllPub
2       AllPub
3       AllPub
4       AllPub
5       AllPub
6       AllPub
7       AllPub
8       AllPub
9       AllPub
10      AllPub
11      AllPub
12      AllPub
13      AllPub
14      AllPub
15      AllPub
16      AllPub
17      AllPub
18      AllPub
```

Handling Exterior1st and Exterior2nd Features

In [72]:
```
1 df.Exterior1st.value_counts()
```

Out[72]:
```
VinylSd    1025
MetalSd     450
HdBoard     442
Wd Sdng     411
Plywood     221
CemntBd     126
BrkFace      87
WdShing      56
AsbShng      44
Stucco       43
BrkComm       6
AsphShn       2
Stone         2
CBlock        2
ImStucc       1
Name: Exterior1st, dtype: int64
```

In [73]:
```
1 df.Exterior2nd.value_counts()
```

Out[73]:
```
VinylSd    1014
MetalSd     447
HdBoard     406
Wd Sdng     391
Plywood     270
CmentBd     126
Wd Shng      81
BrkFace      47
Stucco       47
AsbShng      38
Brk Cmn      22
ImStucc      15
Stone         6
AsphShn       4
CBlock        3
Other         1
Name: Exterior2nd, dtype: int64
```

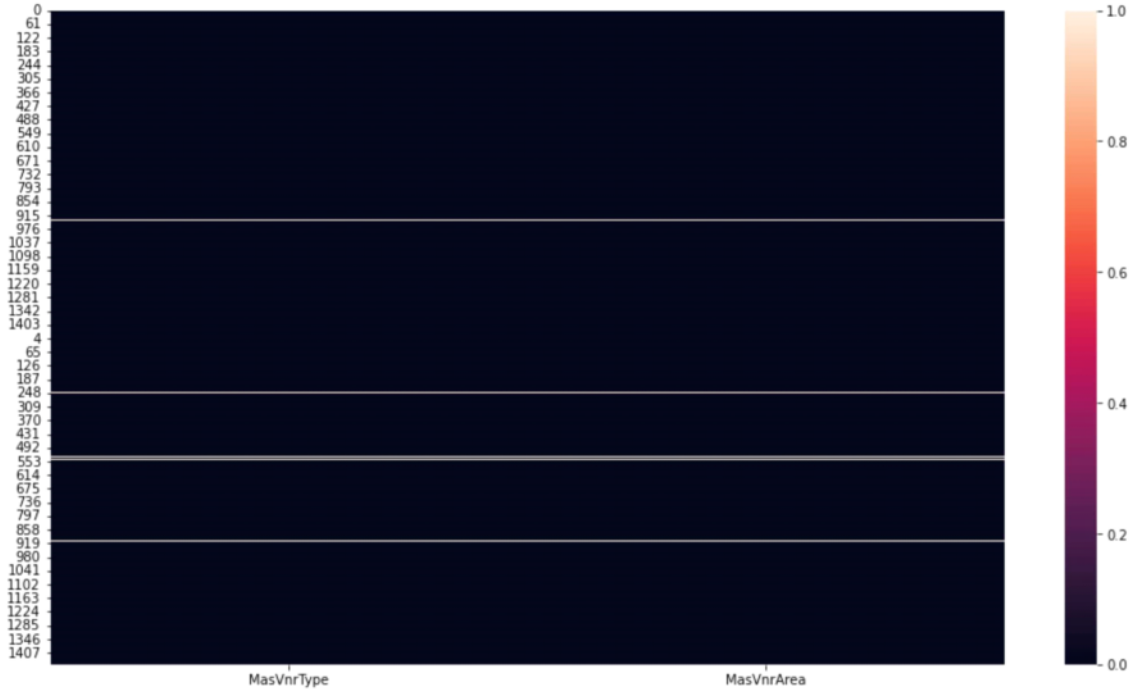
In [74]:
```
1 Exterior1st_mode=df.Exterior1st.mode()[0]
2 df_mvi.Exterior1st.replace(np.nan,Exterior1st_mode,inplace=True)
3 print("Exterior1st Is Cleared",df_mvi.Exterior1st.isnull().sum().sum())
4
5 Exterior2nd_mode=df.Exterior2nd.mode()[0]
6 df_mvi.Exterior2nd.replace(np.nan,Exterior2nd_mode,inplace=True)
7 print("Exterior2nd Is Cleared",df_mvi.Exterior2nd.isnull().sum().sum())
```

```
Exterior1st Is Cleared 0
Exterior2nd Is Cleared 0
```

Handling Missing value MasVnrType (cat) AND MasVnrArea (num)

In [75]:
```
1 plt.figure(figsize=(16,9))
2 sns.heatmap(df[[ "MasVnrType","MasVnrArea"]].isnull())
```

Out[75]:
```
<AxesSubplot:>
```



In [76]:
```
df["MasVnrType"].value_counts() #categorical Variable
```

Out[76]:
```
None       1742
BrkFace     879
Stone       249
BrkCmn       25
Name: MasVnrType, dtype: int64
```

In [77]:
```
df.MasVnrArea.value_counts()  #Numrical Variabel
```

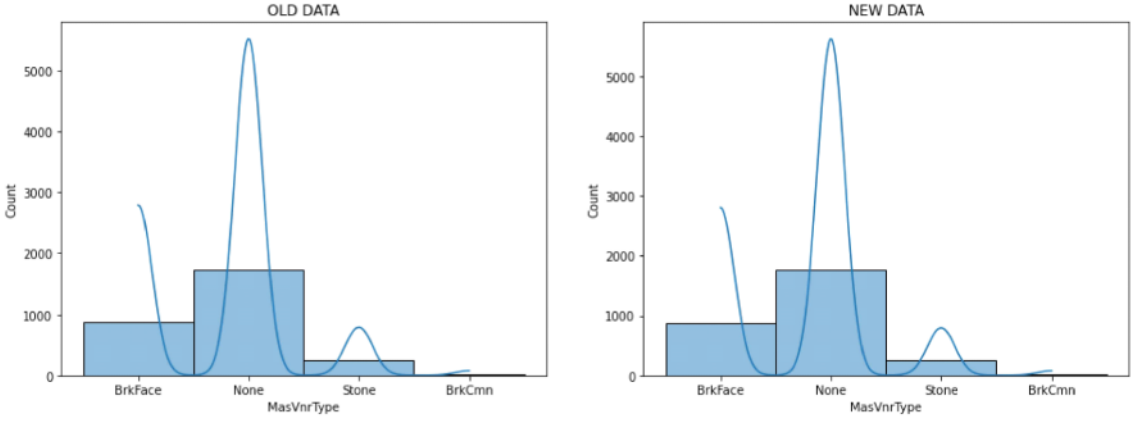
Out[77]:
```
0.0       1738
120.0       15
176.0       13
200.0       13
216.0       12
180.0       12
72.0        11
16.0        11
144.0       11
108.0       11
340.0       10
196.0        9
210.0        9
128.0        9
80.0         9
170.0        8
132.0        8
302.0        8
40.0         8
```

In [78]:
```
MasVnrType_mode=df.MasVnrType.mode()[0]
df_mvi.MasVnrType.replace(np.nan,MasVnrType_mode,inplace=True)
print("MasVnrType Is Cleared",df_mvi.MasVnrType.isnull().sum().sum())
```

```
MasVnrType Is Cleared 0
```

In [79]:
```
def box_cat_plot1(df,df_new,features):
    plt.figure(figsize=(16,12))
    plt.subplot(221)
    plt.title("OLD DATA")
    sns.histplot(df[features],kde=True)
    plt.subplot(222)
    plt.title("NEW DATA")
    sns.histplot(df_new[features],kde=True)
```

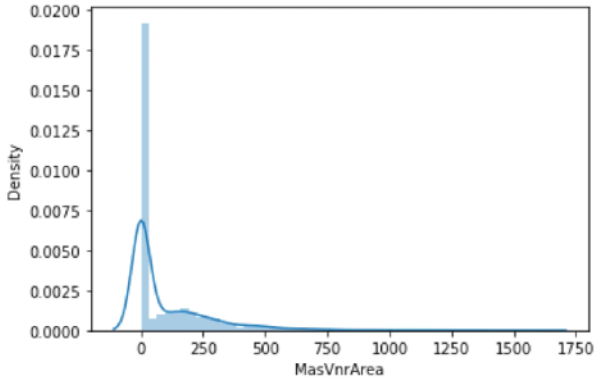
In [80]:
```
box_cat_plot1(df,df_mvi,"MasVnrType")
```



In [81]:
```
1 sns.distplot(df.MasVnrArea,kde=True)
```

```
C:\Users\Sagar\AppData\Local\Temp\ipykernel_2548\2124892109.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df.MasVnrArea,kde=True)
```

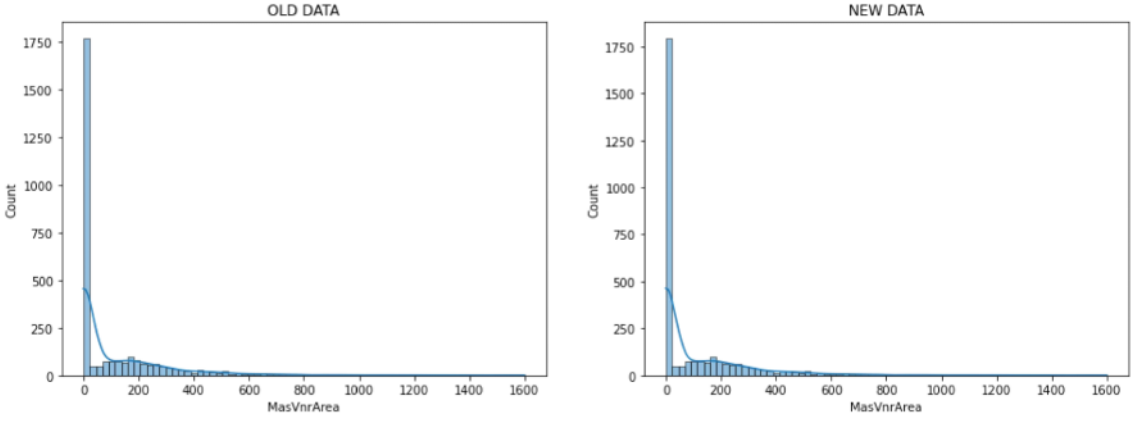
Out[81]:
```
<AxesSubplot:xlabel='MasVnrArea', ylabel='Density'>
```



In [82]:
```
1 #put zero as per our domain Knowlege
2 MasVnrArea_Zero=0
3 df_mvi.MasVnrArea.replace(np.nan,MasVnrArea_Zero,inplace=True)
4 print("MasVnrArea Is Cleared",df_mvi.MasVnrArea.isnull().sum().sum())
```

```
MasVnrArea Is Cleared 0
```

In [83]:
```
1 box_cat_plot1(df,df_mvi,"MasVnrArea")
```



Handling Missing value (cat) featrures and (num) featrures of bsmt

In [ ]:
```
1 #categorical featrures
```

```
BsmtQual          2.774923
BsmtCond          2.809181
BsmtExposure      2.809181
BsmtFinType1      2.706406
BsmtFinType2      2.740665

#numerical featrures
BsmtFinSF1        0.034258
BsmtFinSF2        0.034258
TotalBsmtSF       0.034258
BsmtUnfSF         0.034258
BsmtFullBath      0.068517
BsmtHalfBath      0.068517
```

In [84]:
```
cat_bsmt_features=["BsmtQual","BsmtCond","BsmtExposure","BsmtFinType1","BsmtFinType2"]
```

In [85]:
```
num_bsmt_features=["BsmtFinSF1","BsmtFinSF2","TotalBsmtSF","BsmtUnfSF","BsmtFullBath","BsmtHalfB
```

In [86]:
```
cat_bsmt_features
```

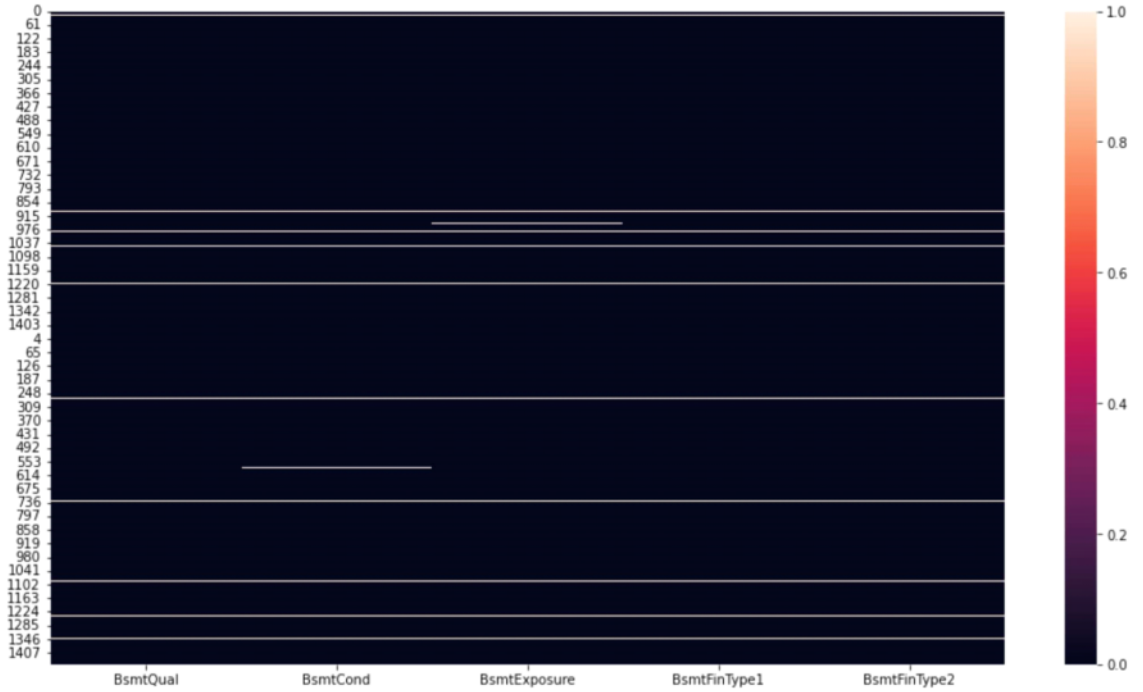
Out[86]:
```
['BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2']
```

In [87]:
```
print(num_bsmt_features,end=" ")
```

```
['BsmtFinSF1', 'BsmtFinSF2', 'TotalBsmtSF', 'BsmtUnfSF', 'BsmtFullBath', 'BsmtHalfBath']
```

In [88]:
```
plt.figure(figsize=(16,9))
sns.heatmap(df[cat_bsmt_features].isnull())
```

Out[88]:
```
<AxesSubplot:>
```



In [89]:
```
for var in cat_bsmt_features:
    print(f"\n*************Value Count Of{var}************* \n",df[var].value_counts())
```

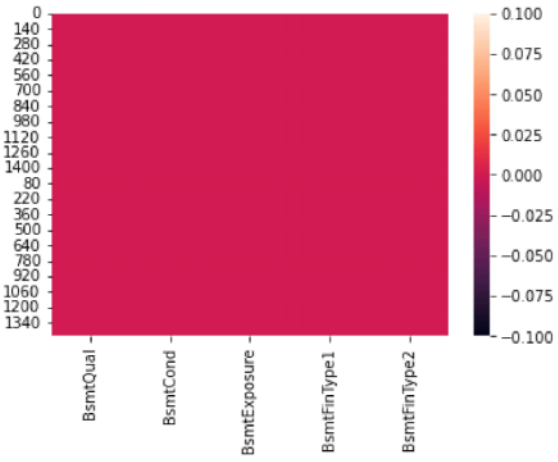
```
*************Value Count OfBsmtQual*************
 TA    1283
Gd    1209
Ex     258
Fa      88
Name: BsmtQual, dtype: int64

*************Value Count OfBsmtCond*************
 TA    2606
Gd     122
Fa     104
Po       5
Name: BsmtCond, dtype: int64

*************Value Count OfBsmtExposure*************
 No    1904
Av     418
Gd     276
Mn     239
Name: BsmtExposure, dtype: int64

*************Value Count OfBsmtFinType1*************
 Unf    851
GLQ    849
ALQ    429
Rec    288
BLQ    269
LwQ    154
Name: BsmtFinType1, dtype: int64

*************Value Count OfBsmtFinType2*************
 Unf    2493
Rec     105
LwQ      87
BLQ      68
ALQ      52
GLQ      34
Name: BsmtFinType2, dtype: int64
```

In [90]:
```
# fillna
bsmt_const="NA"
for var in cat_bsmt_features:
    df_mvi[var].replace(np.nan,bsmt_const,inplace=True)
    print(f"{var} Is Cleared",df_mvi[var].isnull().sum().sum())
```

```
BsmtQual Is Cleared 0
BsmtCond Is Cleared 0
BsmtExposure Is Cleared 0
BsmtFinType1 Is Cleared 0
BsmtFinType2 Is Cleared 0
```

In [91]:
```
1 sns.heatmap(df_mvi[cat_bsmt_features].isnull())
```

Out[91]: <AxesSubplot:>



In [92]:
```
1 df_mvi[num_bsmt_features].isnull().sum()
```

Out[92]:
```
BsmtFinSF1      1
BsmtFinSF2      1
TotalBsmtSF     1
BsmtUnfSF       1
BsmtFullBath    2
BsmtHalfBath    2
dtype: int64
```

In [93]:
```
1 df_bsmt=df[cat_bsmt_features+num_bsmt_features]
2 df_bsmt[df_bsmt.isnull().any(axis=1)]
```

Out[93]:

	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinType2	BsmtFinSF1	BsmtFinSF2	Tota
17	NaN	NaN	NaN	NaN	NaN	0.0	0.0	
39	NaN	NaN	NaN	NaN	NaN	0.0	0.0	
90	NaN	NaN	NaN	NaN	NaN	0.0	0.0	
102	NaN	NaN	NaN	NaN	NaN	0.0	0.0	
156	NaN	NaN	NaN	NaN	NaN	0.0	0.0	
182	NaN	NaN	NaN	NaN	NaN	0.0	0.0	
259	NaN	NaN	NaN	NaN	NaN	0.0	0.0	
332	Gd	TA	No	GLQ	NaN	1124.0	479.0	
342	NaN	NaN	NaN	NaN	NaN	0.0	0.0	
362	NaN	NaN	NaN	NaN	NaN	0.0	0.0	

In [94]:
```
bsmt_num=0
for var in num_bsmt_features:
    df_mvi[var].replace(np.nan,bsmt_num,inplace=True)
    print(f"{var} Is Cleared : ",df_mvi[var].isnull().sum())
```

```
BsmtFinSF1 Is Cleared :  0
BsmtFinSF2 Is Cleared :  0
TotalBsmtSF Is Cleared :  0
BsmtUnfSF Is Cleared :  0
BsmtFullBath Is Cleared :  0
BsmtHalfBath Is Cleared :  0
```

Handling the features 'Electrical', 'KitchenQual' ,Using 'KitchenAbvGr'

In [95]:
```
df_elc_kit=df[['Electrical', 'KitchenQual' ,'KitchenAbvGr']]
df_elc_kit[df_elc_kit.isnull().any(axis=1)]
```

Out[95]:

	Electrical	KitchenQual	KitchenAbvGr
1379	NaN	Gd	1
95	SBrkr	NaN	1

In [96]:
```
elec_mode=df.Electrical.mode()[0]
df_mvi.Electrical.replace(np.nan,elec_mode,inplace=True)
df_mvi.Electrical.isnull().sum()
```

Out[96]: 0

In [97]:
```
KitchenQual_mode=df.KitchenQual.mode()[0]
df_mvi.KitchenQual.replace(np.nan,KitchenQual_mode,inplace=True)
df_mvi.KitchenQual.isnull().sum()
```

Out[97]: 0

Handling FEATURES & Functional & FireplaceQu & PoolQC & Fence & MiscFeature & SaleType

```
FEATURES         MISSING%  FILLING MISSSING VALUE MEOTHED
Functional       0.068517  #MODE
FireplaceQu      48.646797  #NA
PoolQC          99.657417  #NA
Fence           80.438506  #NA
MiscFeature     96.402878  #NA
SaleType         0.034258  #MODE
```

Function for repace the none value by mode

In [98]:
```
#fuction for repace the none value by mode
def fun(df,df_new,features):
    mode=df[features].mode()[0]
    df_mvi[features].replace(np.nan,mode,inplace=True)
    print(f"{features} is cleared : ",df_mvi[features].isnull().sum())
```

Handling MODE fillng features

In [99]:
```
print("Functional value count",df.Functional.value_counts())
print("..................................................")
print("SaleType value count",df.SaleType.value_counts())
print("..................................................")
```

```
Functional value count Typ     2717
Min2      70
Min1      65
Mod       35
Maj1      19
Maj2       9
Sev        2
Name: Functional, dtype: int64
..................................................
SaleType value count WD       2525
New       239
COD        87
ConLD      26
CWD        12
ConLI       9
ConLw       8
Oth         7
Con         5
Name: SaleType, dtype: int64
..................................................
```

In [100]:
```
fun(df,df_mvi,"Functional")
fun(df,df_mvi,"SaleType")
```

```
Functional is cleared :  0
SaleType is cleared :  0
```

Handling NA fillng Other cat features

In [101]:
```
other_cat_features=["FireplaceQu","PoolQC","Fence","MiscFeature"]
```

In [102]:
```
for var in other_cat_features:
    print(f"{var} value count :",df[var].value_counts())
```

```
FireplaceQu value count : Gd    744
TA    592
Fa     74
Po     46
Ex     43
Name: FireplaceQu, dtype: int64
PoolQC value count : Ex    4
Gd    4
Fa    2
Name: PoolQC, dtype: int64
Fence value count : MnPrv    329
GdPrv    118
GdWo     112
MnWw      12
Name: Fence, dtype: int64
MiscFeature value count : Shed    95
Gar2     5
Othr     4
TenC     1
Name: MiscFeature, dtype: int64
```

Function for repace the none value by Constant Value NA

In [103]:
```
#fuction for repace the none value by Constant Value NA
def fun1(df,df_new,features):
    cost_val="NA"
    df_mvi[features].replace(np.nan,cost_val,inplace=True)
    print(f"{features} is cleared : ",df_mvi[features].isnull().sum())
```

In [104]:
```
#for FireplaceQu
fun1(df,df_mvi,"FireplaceQu")
fun1(df,df_mvi,"PoolQC")
fun1(df,df_mvi,"Fence")
fun1(df,df_mvi,"MiscFeature")
```

```
FireplaceQu is cleared :  0
PoolQC is cleared :  0
Fence is cleared :  0
MiscFeature is cleared :  0
```

Handling garge type categorical as well as numerical variable

Categorical variable in Garage Sectoer

```
GarageType       5.378554
GarageFinish     5.447071
GarageQual       5.447071
GarageCond       5.447071
```

In [105]:
```
cat_grage_features=["GarageType","GarageFinish","GarageQual" ,"GarageCond"]
cat_cont=0
for var in cat_grage_features:
    df_mvi[var].replace(np.nan,cat_cont,inplace=True)
    print(f"{var} is cleared : ",df_mvi[var].isnull().sum())
```

```
GarageType is cleared :  0
GarageFinish is cleared :  0
GarageQual is cleared :  0
GarageCond is cleared :  0
```

Numerical variable in Garage Sectoer

```
GarageYrBlt      5.447071
GarageCars       0.034258
GarageArea       0.034258
```

In [106]:
```
num_grage_features=["GarageYrBlt","GarageCars","GarageArea" ]
num_cont=0
for var in num_grage_features:
    df_mvi[var].replace(np.nan,num_cont,inplace=True)
    print(f"{var} is cleared : ",df_mvi[var].isnull().sum())
```

```
GarageYrBlt is cleared :  0
GarageCars is cleared :  0
GarageArea is cleared :  0
```

In [107]:
```
df_mvi.isnull().sum().sum()
```

Out[107]: 1459

Features Transformation

CONVERT THE NUMERICAL VARIBLE IN CATEGORICAL VARIBLE

In [108]:
```
conversion_in_cat=["MSSubClass","YearBuilt","YearRemodAdd","GarageYrBlt","MoSold","YrSold"]
for var in conversion_in_cat:
    print(f"{var} data type",df_mvi[var].dtypes)
```

```
MSSubClass data type int64
YearBuilt data type int64
YearRemodAdd data type int64
GarageYrBlt data type float64
MoSold data type int64
YrSold data type int64
```

In [109]:
```
df_mvi[conversion_in_cat].head()
```

Out[109]:

	MSSubClass	YearBuilt	YearRemodAdd	GarageYrBlt	MoSold	YrSold
0	60	2003	2003	2003.0	2	2008
1	20	1976	1976	1976.0	5	2007
2	60	2001	2002	2001.0	9	2008
3	70	1915	1970	1998.0	2	2006
4	60	2000	2000	2000.0	12	2008

In [110]:
```
import calendar as cl #this labarary convert integer  month int string value
# EX: 1)MAR=3,2)FEB=2 OR 1=JAN,6=JUN
```

In [111]:
```
cl.month_abbr[12]
```

Out[111]: 'Dec'

In [112]:
```
df_mvi["MoSold"].unique()
```

Out[112]: array([2, 5, 9, 12, 10, 8, 11, 4, 1, 7, 3, 6], dtype=int64)

In [113]:
```
df_mvi["MoSold"]=df_mvi["MoSold"].apply(lambda var :cl.month_abbr[var])
```

In [114]:
```
df_mvi["MoSold"].unique()
```

Out[114]:
```
array(['Feb', 'May', 'Sep', 'Dec', 'Oct', 'Aug', 'Nov', 'Apr', 'Jan',
       'Jul', 'Mar', 'Jun'], dtype=object)
```

In [115]:
```
for var in conversion_in_cat:
    df_mvi[var]=df_mvi[var].astype(str)
    print(f"{var} data type",df_mvi[var].dtypes)
```

```
MSSubClass data type object
YearBuilt data type object
YearRemodAdd data type object
GarageYrBlt data type object
MoSold data type object
YrSold data type object
```

CONVERT THE CATEGORICAL VARIBLE IN NUMERICAL VARIBLE

In [116]:
```
ordinal_enc_var=['ExterQual', 'ExterCond', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1
                 'HeatingQC', 'Kitchen00Qual', 'Functional', 'GarageType', 'GarageFinish', 'Gara
                 'GarageCond', 'PavedDrive', 'PoolQCUtilities', 'FireplaceQu']

```

In [117]:
```
len(ordinal_enc_var)
```

Out[117]: 17

In [118]:
```
from pandas.api.types import CategoricalDtype
```

In [119]:
```
df_mvi['ExterQual'].unique()
```

Out[119]: array(['Gd', 'TA', 'Ex', 'Fa'], dtype=object)

In [120]:
```
df_mvi['ExterQual']=df_mvi['ExterQual'].astype(CategoricalDtype(categories=['Gd', 'TA', 'Ex', 'F
```

In [121]:
```
df_mvi['ExterQual'].value_counts()
```

Out[121]:
```
1    1798
0     979
2     107
3      35
Name: ExterQual, dtype: int64
```

In [122]:
```
df_mvi['ExterCond'].unique()
```

Out[122]: array(['TA', 'Gd', 'Fa', 'Po', 'Ex'], dtype=object)

In [123]:
```
df_mvi['ExterCond']=df_mvi['ExterCond'].astype(CategoricalDtype(categories=['TA', 'Gd', 'Fa', 'P
```

In [124]:
```
def fun1(var):
    print(f"Uinque Value of {var} : ",df_mvi[var].unique())
```

In [125]:
```
fun1("FireplaceQu")
# ['ExterQual', 'ExterCond', 'BsmtQual', 'BsmtCond',
#  'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'HeatingQC', 'KitchenQual',
#  'Functional', 'GarageType', 'GarageFinish',
#  'GarageQual',GarageCond', 'PavedDrive',
#  'PoolQC,Utilities', 'FireplaceQu']
```

```
Uinque Value of FireplaceQu :  ['NA' 'TA' 'Gd' 'Fa' 'Ex' 'Po']
```

In [126]:

```
df_mvi.BsmtQual=df_mvi.BsmtQual.astype(CategoricalDtype(categories=['Gd' 'TA' 'Ex' 'NA' 'Fa'],or

df_mvi.BsmtCond=df_mvi.BsmtCond.astype(CategoricalDtype(categories=['TA' 'Gd' 'NA' 'Fa' 'Po'],or

df_mvi.BsmtExposure=df_mvi.BsmtExposure.astype(CategoricalDtype(categories=['No' 'Gd' 'Mn' 'Av'

df_mvi.BsmtFinType1=df_mvi.BsmtFinType1.astype(CategoricalDtype(categories=['GLQ' 'ALQ' 'Unf' 'R

df_mvi.BsmtFinType2=df_mvi.BsmtFinType2.astype(CategoricalDtype(categories=['Unf', 'BLQ', 'NA',

df_mvi.HeatingQC=df_mvi.HeatingQC.astype(CategoricalDtype(categories=['Ex' 'Gd' 'TA' 'Fa' 'Po'],

df_mvi.KitchenQual=df_mvi.KitchenQual.astype(CategoricalDtype(categories=['Gd' 'TA' 'Ex' 'Fa'],o

df_mvi.Functional=df_mvi.Functional.astype(CategoricalDtype(categories=['Typ' 'Min1' 'Maj1' 'Min

df_mvi.GarageType=df_mvi.GarageType.astype(CategoricalDtype(categories=['Attchd' 'Detchd' 'Built

df_mvi.GarageFinish=df_mvi.GarageFinish.astype(CategoricalDtype(categories= ['RFn' 'Unf' 'Fin',]

df_mvi.GarageQual=df_mvi.GarageQual.astype(CategoricalDtype(categories=['TA' 'Fa' 'Gd' 'Ex' 'Po'

df_mvi.GarageCond=df_mvi.GarageCond.astype(CategoricalDtype(categories=['TA' 'Fa'  'Gd' 'Po' 'Ex

df_mvi.PavedDrive=df_mvi.PavedDrive.astype(CategoricalDtype(categories= ['Y' 'N' 'P'],ordered=Tr

df_mvi.PoolQC=df_mvi.PoolQC.astype(CategoricalDtype(categories= ['NA' 'Ex' 'Fa' 'Gd'],ordered=Tr

df_mvi.Utilities=df_mvi.Utilities.astype(CategoricalDtype(categories=['AllPub' 'NoSeWa'],ordered

df_mvi.FireplaceQu=df_mvi.FireplaceQu.astype(CategoricalDtype(categories=['NA' 'TA' 'Gd' 'Fa' 'E
```

In [127]:
```
1 df_mvi.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2919 entries, 0 to 1458
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             2919 non-null   int64
 1   MSSubClass     2919 non-null   object
 2   MSZoning       2919 non-null   object
 3   LotFrontage    2919 non-null   float64
 4   LotArea        2919 non-null   int64
 5   Street         2919 non-null   object
 6   Alley          2919 non-null   object
 7   LotShape       2919 non-null   object
 8   LandContour    2919 non-null   object
 9   Utilities      2919 non-null   int8
 10  LotConfig      2919 non-null   object
 11  LandSlope      2919 non-null   object
 12  Neighborhood   2919 non-null   object
 13  Condition1     2919 non-null   object
 14  Condition2     2919 non-null   object
 15  BldgType       2919 non-null   object
 16  HouseStyle     2919 non-null   object
 17  OverallQual    2919 non-null   int64
 18  OverallCond    2919 non-null   int64
 19  YearBuilt      2919 non-null   object
 20  YearRemodAdd   2919 non-null   object
 21  RoofStyle      2919 non-null   object
 22  RoofMatl       2919 non-null   object
 23  Exterior1st    2919 non-null   object
 24  Exterior2nd    2919 non-null   object
 25  MasVnrType     2919 non-null   object
 26  MasVnrArea     2919 non-null   float64
 27  ExterQual      2919 non-null   int8
 28  ExterCond      2919 non-null   int8
 29  Foundation     2919 non-null   object
 30  BsmtQual       2919 non-null   int8
 31  BsmtCond       2919 non-null   int8
 32  BsmtExposure   2919 non-null   int8
 33  BsmtFinType1   2919 non-null   int8
 34  BsmtFinSF1     2919 non-null   float64
 35  BsmtFinType2   2919 non-null   int8
 36  BsmtFinSF2     2919 non-null   float64
 37  BsmtUnfSF      2919 non-null   float64
 38  TotalBsmtSF    2919 non-null   float64
 39  Heating        2919 non-null   object
 40  HeatingQC      2919 non-null   int8
 41  CentralAir     2919 non-null   object
 42  Electrical     2919 non-null   object
 43  1stFlrSF       2919 non-null   int64
 44  2ndFlrSF       2919 non-null   int64
 45  LowQualFinSF   2919 non-null   int64
 46  GrLivArea      2919 non-null   int64
 47  BsmtFullBath   2919 non-null   float64
 48  BsmtHalfBath   2919 non-null   float64
 49  FullBath       2919 non-null   int64
 50  HalfBath       2919 non-null   int64
 51  BedroomAbvGr   2919 non-null   int64
 52  KitchenAbvGr   2919 non-null   int64
 53  KitchenQual    2919 non-null   int8
 54  TotRmsAbvGrd   2919 non-null   int64
 55  Functional     2919 non-null   int8
 56  Fireplaces     2919 non-null   int64
 57  FireplaceQu    2919 non-null   int8
 58  GarageType     2919 non-null   int8
 59  GarageYrBlt    2919 non-null   object
 60  GarageFinish   2919 non-null   int8
 61  GarageCars     2919 non-null   float64
 62  GarageArea     2919 non-null   float64
 63  GarageQual     2919 non-null   int8
 64  GarageCond     2919 non-null   int8
 65  PavedDrive     2919 non-null   int8
 66  WoodDeckSF     2919 non-null   int64
 67  OpenPorchSF    2919 non-null   int64
```

```
 68  EnclosedPorch  2919 non-null   int64
 69  3SsnPorch      2919 non-null   int64
 70  ScreenPorch    2919 non-null   int64
 71  PoolArea       2919 non-null   int64
 72  PoolQC         2919 non-null   int8
 73  Fence          2919 non-null   object
 74  MiscFeature    2919 non-null   object
 75  MiscVal        2919 non-null   int64
 76  MoSold         2919 non-null   object
 77  YrSold         2919 non-null   object
 78  SaleType       2919 non-null   object
 79  SaleCondition  2919 non-null   object
 80  SalePrice      1460 non-null   float64
dtypes: float64(11), int64(21), int8(18), object(31)
memory usage: 1.5+ MB
```

ONE HOT CODING FOR NOMINAL CATEGORIGAL DATA

In [128]:
```
df_encod=df_mvi.copy()
object_features_1=df_encod.select_dtypes(include="object").columns.tolist()
object_features_1
```

Out[128]:
```
['MSSubClass',
 'MSZoning',
 'Street',
 'Alley',
 'LotShape',
 'LandContour',
 'LotConfig',
 'LandSlope',
 'Neighborhood',
 'Condition1',
 'Condition2',
 'BldgType',
 'HouseStyle',
 'YearBuilt',
 'YearRemodAdd',
 'RoofStyle',
 'RoofMatl',
 'Exterior1st',
 'Exterior2nd',
 'MasVnrType',
 'Foundation',
 'Heating',
 'CentralAir',
 'Electrical',
 'GarageYrBlt',
 'Fence',
 'MiscFeature',
 'MoSold',
 'YrSold',
 'SaleType',
 'SaleCondition']
```

In [129]:
```
df_encod[object_features_1].head()
```

Out[129]:

	MSSubClass	MSZoning	Street	Alley	LotShape	LandContour	LotConfig	LandSlope	Neighborhood	Cor
0	60	RL	Pave	NA	Reg	Lvl	Inside	Gtl	CollgCr	
1	20	RL	Pave	NA	Reg	Lvl	FR2	Gtl	Veenker	
2	60	RL	Pave	NA	IR1	Lvl	Inside	Gtl	CollgCr	
3	70	RL	Pave	NA	IR1	Lvl	Corner	Gtl	Crawfor	
4	60	RL	Pave	NA	IR1	Lvl	FR2	Gtl	NoRidge	

In [130]:
```
print("shape before encoding : ",df_encod.shape)
df_encod=pd.get_dummies(df_encod,
                        columns=object_features_1
                        ,prefix=object_features_1,
                        drop_first=True)
print("shape afther encoding : ",df_encod.shape)
```

```
shape before encoding :  (2919, 81)
shape afther encoding :  (2919, 509)
```

In [131]:
```
df_encod.columns
```

Out[131]:
```
Index(['Id', 'LotFrontage', 'LotArea', 'Utilities', 'OverallQual',
       'OverallCond', 'MasVnrArea', 'ExterQual', 'ExterCond', 'BsmtQual',
       ...
       'SaleType_ConLI', 'SaleType_ConLw', 'SaleType_New', 'SaleType_Oth',
       'SaleType_WD', 'SaleCondition_AdjLand', 'SaleCondition_Alloca',
       'SaleCondition_Family', 'SaleCondition_Normal',
       'SaleCondition_Partial'],
      dtype='object', length=509)
```

In [132]:
```
df_encod.head()
```

Out[132]:

	Id	LotFrontage	LotArea	Utilities	OverallQual	OverallCond	MasVnrArea	ExterQual	ExterCond	BsmtQu
0	1	65.0	8450	-1	7	5	196.0	0	0	-
1	2	80.0	9600	-1	6	8	0.0	1	0	-
2	3	68.0	11250	-1	7	5	162.0	0	0	-
3	4	60.0	9550	-1	7	5	0.0	1	0	-
4	5	84.0	14260	-1	8	5	350.0	0	0	-

In [133]:
```
df_encod.select_dtypes(include="object").columns.tolist()
```

Out[133]:
```
[]
```

Data Testing And Spliting

In [134]:
```
df_encod.shape
```

Out[134]:
```
(2919, 509)
```

In [135]:
```
len_train=train.shape[0]
len_train
```

Out[135]:
```
1460
```

In [136]:
```
x_train=df_encod[:len_train].drop("SalePrice",axis=1)
x_train.shape
```

Out[136]:
```
(1460, 508)
```

In [137]:
```
y_train=df_encod["SalePrice"][:len_train]
y_train.shape
```

Out[137]:
```
(1460,)
```

In [138]:
```
x_test=df_encod[len_train:].drop("SalePrice",axis=1)
```

In [139]:
```
1 x_test.shape
```

Out[139]: (1459, 508)

In [140]:
```
1 from sklearn.preprocessing import StandardScaler
2 sc=StandardScaler()
3 sc.fit(x_train)
4 x_train=sc.transform(x_train)
5 x_test=sc.transform(x_test)
```

In [141]:
```
1 x_train
```

Out[141]:
```
array([[-1.73086488, -0.21271975, -0.20714171, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [-1.7284922 ,  0.46815755, -0.09188637, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [-1.72611953, -0.07654429,  0.07347998, ..., -0.11785113,
         0.4676514 , -0.30599503],
       ...,
       [ 1.72611953, -0.16732793, -0.14781027, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [ 1.7284922 , -0.07654429, -0.08016039, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [ 1.73086488,  0.24119845, -0.05811155, ..., -0.11785113,
         0.4676514 , -0.30599503]])
```

In [142]:
```
1 x_test
```

Out[142]:
```
array([[ 1.73323755,  0.46815755,  0.11076257, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [ 1.73561022,  0.51354937,  0.37584985, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [ 1.7379829 ,  0.19580663,  0.33205282, ..., -0.11785113,
         0.4676514 , -0.30599503],
       ...,
       [ 5.18784929,  4.09950312,  0.95042275, ..., -0.11785113,
        -2.13834494, -0.30599503],
       [ 5.19022196, -0.34889521, -0.00759964, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [ 5.19259463,  0.19580663, -0.08918038, ..., -0.11785113,
         0.4676514 , -0.30599503]])
```

In [143]:
```
1 sc.mean_
```

Out[143]:
```
array([ 7.30500000e+02,  6.96863014e+01,  1.05168281e+04, -1.00000000e+00,
        6.09931507e+00,  5.57534247e+00,  1.03117123e+02,  7.20547945e-01,
        1.48630137e-01, -1.00000000e+00, -1.00000000e+00, -1.00000000e+00,
       -1.00000000e+00,  4.43639726e+02,  4.76712329e-01,  4.65493151e+01,
        5.67240411e+02,  1.05742945e+03, -1.00000000e+00,  1.16262671e+03,
        3.46992466e+02,  5.84452055e+00,  1.51546370e+03,  4.25342466e-01,
        5.75342466e-02,  1.56506849e+00,  3.82876712e-01,  2.86643836e+00,
        1.04657534e+00, -1.00000000e+00,  6.51780822e+00, -1.00000000e+00,
        6.13013699e-01, -1.00000000e+00, -1.00000000e+00, -1.00000000e+00,
        1.76712329e+00,  4.72980137e+02, -1.00000000e+00, -1.00000000e+00,
       -1.00000000e+00,  9.42445205e+01,  4.66602740e+01,  2.19541096e+01,
        3.40958904e+00,  1.50609589e+01,  2.75890411e+00, -1.00000000e+00,
        4.34890411e+01,  0.00000000e+00,  4.31506849e-02,  6.84931507e-03,
        2.05479452e-02,  3.67123288e-01,  4.72602740e-02,  2.73972603e-03,
        8.21917808e-03,  9.86301370e-02,  2.04794521e-01,  4.10958904e-02,
        1.09589041e-02,  3.97260274e-02,  1.36986301e-02,  3.56164384e-02,
        4.45205479e-02,  1.09589041e-02,  7.88356164e-01,  1.49315068e-01,
        9.95890411e-01,  9.37671233e-01,  2.80821918e-02,  2.80821918e-02,
        6.84931507e-03,  6.33561644e-01,  3.42465753e-02,  2.46575342e-02,
```

In [144]:
```
1 sc.n_features_in_
```

Out[144]: 508

In [145]:
```
sc.feature_names_in_
```

Out[145]:
```
array(['Id', 'LotFrontage', 'LotArea', 'Utilities', 'OverallQual',
       'OverallCond', 'MasVnrArea', 'ExterQual', 'ExterCond', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF',
       'HeatingQC', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea',
       'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'MiscVal', 'MSSubClass_150', 'MSSubClass_160', 'MSSubClass_180',
       'MSSubClass_190', 'MSSubClass_20', 'MSSubClass_30',
       'MSSubClass_40', 'MSSubClass_45', 'MSSubClass_50', 'MSSubClass_60',
       'MSSubClass_70', 'MSSubClass_75', 'MSSubClass_80', 'MSSubClass_85',
       'MSSubClass_90', 'MSZoning_FV', 'MSZoning_RH', 'MSZoning_RL',
       'MSZoning_RM', 'Street_Pave', 'Alley_NA', 'Alley_Pave',
       'LotShape_IR2', 'LotShape_IR3', 'LotShape_Reg', 'LandContour_HLS',
       'LandContour_Low', 'LandContour_Lvl', 'LotConfig_CulDSac',
```

In [146]:
```
sc.with_mean
```

Out[146]: True

In [147]:
```
sc.with_std
```

Out[147]: True

Train Model

In [148]:
```
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import SGDRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.neural_network import MLPRegressor
from xgboost import XGBRegressor
from sklearn.isotonic import IsotonicRegression
```

In [149]:
```
svr=SVR()
lr=LinearRegression()
sgdr=SGDRegressor()
knr=KNeighborsRegressor()
gpr=GaussianProcessRegressor()
dtr=DecisionTreeRegressor()
gbr=GradientBoostingRegressor()
rfr=RandomForestRegressor()
mlpr=MLPRegressor()
xgbr=XGBRegressor()
ir=IsotonicRegression()
```

In [150]:
```
models={"a":["LinearRegression",lr],
        "b":["SGDRegressor",sgdr],
        "c":["SVR",svr],
        "d":["GaussianProcessRegressor",gpr],
        "e":["KNeighborsRegressor",knr],
        "f":["DecisionTreeRegressor",dtr],
        "g":["GradientBoostingRegressor",gbr],
        "h":["RandomForestRegressor",rfr],
        "i":["MLPRegressor",mlpr],
        "j":["XGBRegressor",xgbr],
       }
```

In [ ]:

```python

```

In [151]:

```python
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.metrics import r2_score,make_scorer

def test_model(model,x_train=x_train,y_train=y_train):
    cv=KFold(n_splits=7,shuffle=True, random_state=45)
    r2=make_scorer(r2_score)
    r2_val_score=cross_val_score(model,x_train,y_train,cv=cv,scoring=r2)
    score=[r2_val_score.mean()]
    return score
```

In [152]:

```python
model_score=[]
for var in models:
    print("Traing Model : ",models[var][0])
    score=test_model(models[var][1],x_train,y_train)
    print(f"score of the {models[var][0]}",score )
    model_score.append(models[var][0])
```

```
Traing Model :  LinearRegression
score of the LinearRegression [-3.9459732823025646e+26]
Traing Model :  SGDRegressor
score of the SGDRegressor [-4945.624891385911]
Traing Model :  SVR
score of the SVR [-0.052182279403310404]
Traing Model :  GaussianProcessRegressor
score of the GaussianProcessRegressor [-5.400748068567466]
Traing Model :  KNeighborsRegressor
score of the KNeighborsRegressor [0.5417681139066205]
Traing Model :  DecisionTreeRegressor
score of the DecisionTreeRegressor [0.6777017267659083]
Traing Model :  GradientBoostingRegressor
score of the GradientBoostingRegressor [0.8627464608275279]
Traing Model :  RandomForestRegressor
score of the RandomForestRegressor [0.8417152468877314]
Traing Model :  MLPRegressor

C:\Users\Sagar\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:702: Co
nvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization has
n't converged yet.
  warnings.warn(
C:\Users\Sagar\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:702: Co
nvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization has
n't converged yet.
  warnings.warn(
C:\Users\Sagar\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:702: Co
nvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization has
n't converged yet.
  warnings.warn(
C:\Users\Sagar\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:702: Co
nvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization has
n't converged yet.
  warnings.warn(
C:\Users\Sagar\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:702: Co
nvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization has
n't converged yet.
  warnings.warn(
C:\Users\Sagar\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:702: Co
nvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization has
n't converged yet.
  warnings.warn(
C:\Users\Sagar\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:702: Co
nvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization has
n't converged yet.
  warnings.warn(

score of the MLPRegressor [-4.8871508446241325]
Traing Model :  XGBRegressor
score of the XGBRegressor [0.8325862547476647]
```

In [ ]:
```
1
```