Project Name: House Price Prediction using Regression Techniques

EDA (Exploratory Data Analysis)

Dataset to downloaded from the below link https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data)

Lifecycle of a Data Analytics Project

1. Data Analysis-Exploratory data Analysis (EDA)
2. Feature Engineering
3. Feature Selection
4. Model Building
5. Model Deployment

1: Data Analysis Phase-EDA

In [71]:
```
## Main aim is to understand more about the data

import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.pyplot as plt
from pandas.api.types import CategoricalDtype
import calendar
from sklearn.preprocessing import OneHotEncoder


## Display all the columns of the dataframe

pd.pandas.set_option('display.max_columns',None)
```

In [72]:
```
sns.set_style('whitegrid') # plot style
plt.rcParams['figure.figsize'] = (15, 10) # plot size
```

In [73]:
```
dataset_train=pd.read_csv('train.csv')
dataset_test=pd.read_csv('test.csv')

## Display the shape of dataset with rows and columns.

print("Shape of Train Data Set: ", dataset_train.shape)
print("Shape of Test Data Set: ", dataset_test.shape) #Testdataset does not have sellprice column;thatswhy it shows 8
```

```
Shape of Train Data Set:  (1460, 81)
Shape of Test Data Set:  (1459, 80)
```

In [74]:
```
## print the top 10 records;by default it display only 5 entries
dataset_train.head(10)
```

Out[74]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Cor
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	
5	6	50	RL	85.0	14115	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Mitchel	
6	7	20	RL	75.0	10084	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Somerst	
7	8	60	RL	NaN	10382	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	NWAmes	
8	9	50	RM	51.0	6120	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	OldTown	
9	10	190	RL	50.0	7420	Pave	NaN	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	

In [75]:
```
dataset_test.head(10)
```

Out[75]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	C
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAmes	
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	NAmes	
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	Inside	Gtl	StoneBr	
5	1466	60	RL	75.0	10000	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Gilbert	
6	1467	20	RL	NaN	7980	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	
7	1468	60	RL	63.0	8402	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	
8	1469	20	RL	85.0	10176	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Gilbert	
9	1470	20	RL	70.0	8400	Pave	NaN	Reg	Lvl	AllPub	Corner	Gtl	NAmes	

Data Integration**

In [76]:
```
#Concat function concatenates data frames along rows OR columns.
```

In [77]:
```
## Concatinate train and test datasets.
```

In [78]:
```
cds=pd.concat((dataset_train, dataset_test)) # cds=concatinate data set
```

In [79]:
```
print("Shape of Concatinate/Unified Dataset: ", cds.shape)
```

```
Shape of Concatinate/Unified Dataset:  (2919, 81)
```

In [80]:
```

cds.to_csv('cds.csv', index=False)
```

In [81]:
```
cds.head()
```

Out[81]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Con
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	

In [82]:
```
cds.tail()
```

Out[82]:

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

In [83]:
```
cds.info() # It gives brief info about our concatinated dataset(cds)
```

```
<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
```

Missing/Null Values

In [84]:
```
## Check the % of nan values present in each features
## 1: To make list of features with missing values

features_with_na=[features for features in cds.columns if cds[features].isnull().sum()>1]

## 2 :To print the feature name and % of missing values.

for feature in features_with_na:
    print(feature,':', np.round(cds[feature].isnull().mean()*100, 4),  '%  of missing values.')
```

```
MSZoning : 0.137 %  of missing values.
LotFrontage : 16.6495 %  of missing values.
Alley : 93.2169 %  of missing values.
Utilities : 0.0685 %  of missing values.
MasVnrType : 0.8222 %  of missing values.
MasVnrArea : 0.7879 %  of missing values.
BsmtQual : 2.7749 %  of missing values.
BsmtCond : 2.8092 %  of missing values.
BsmtExposure : 2.8092 %  of missing values.
BsmtFinType1 : 2.7064 %  of missing values.
BsmtFinType2 : 2.7407 %  of missing values.
BsmtFullBath : 0.0685 %  of missing values.
BsmtHalfBath : 0.0685 %  of missing values.
Functional : 0.0685 %  of missing values.
FireplaceQu : 48.6468 %  of missing values.
GarageType : 5.3786 %  of missing values.
GarageYrBlt : 5.4471 %  of missing values.
GarageFinish : 5.4471 %  of missing values.
GarageQual : 5.4471 %  of missing values.
GarageCond : 5.4471 %  of missing values.
PoolQC : 99.6574 %  of missing values.
Fence : 80.4385 %  of missing values.
MiscFeature : 96.4029 %  of missing values.
SalePrice : 49.9829 %  of missing values.
```

Numerical Features of cds dataset

In [85]:
```
# list of numerical features (including integar and float)
Numerical_Features = [feature for feature in cds.columns if cds[feature].dtypes != 'O']

print('No. of numerical variables: ', len(Numerical_Features))

# Display the numerical variables with head(By default 5 entries/rows)
cds[Numerical_Features].head()
```

```
No. of numerical variables:  38
```

Out[85]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnf
0	1	60	65.0	8450	7	5	2003	2003	196.0	706.0	0.0	15
1	2	20	80.0	9600	6	8	1976	1976	0.0	978.0	0.0	28
2	3	60	68.0	11250	7	5	2001	2002	162.0	486.0	0.0	43
3	4	70	60.0	9550	7	5	1915	1970	0.0	216.0	0.0	54
4	5	60	84.0	14260	8	5	2000	2000	350.0	655.0	0.0	49

Categorical/Object Features of cds dataset

In [86]:
```
categorical_features=[feature for feature in cds.columns if cds[feature].dtypes=='O']
print('No. of categorical featuress: ', len(categorical_features))
categorical_features
```

```
No. of categorical featuress:  43
```

Out[86]:
```
['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']
```

Temporal Features.

** From the Dataset we have 4 year variables(YearBuilt, YearRemodAdd,YrSold, GarageYrBlt).

In [87]:
```
# Display the variables that contain years information
Features_Years = [feature for feature in Numerical_Features if 'Yr' in feature or 'Year' in feature]

Features_Years
```

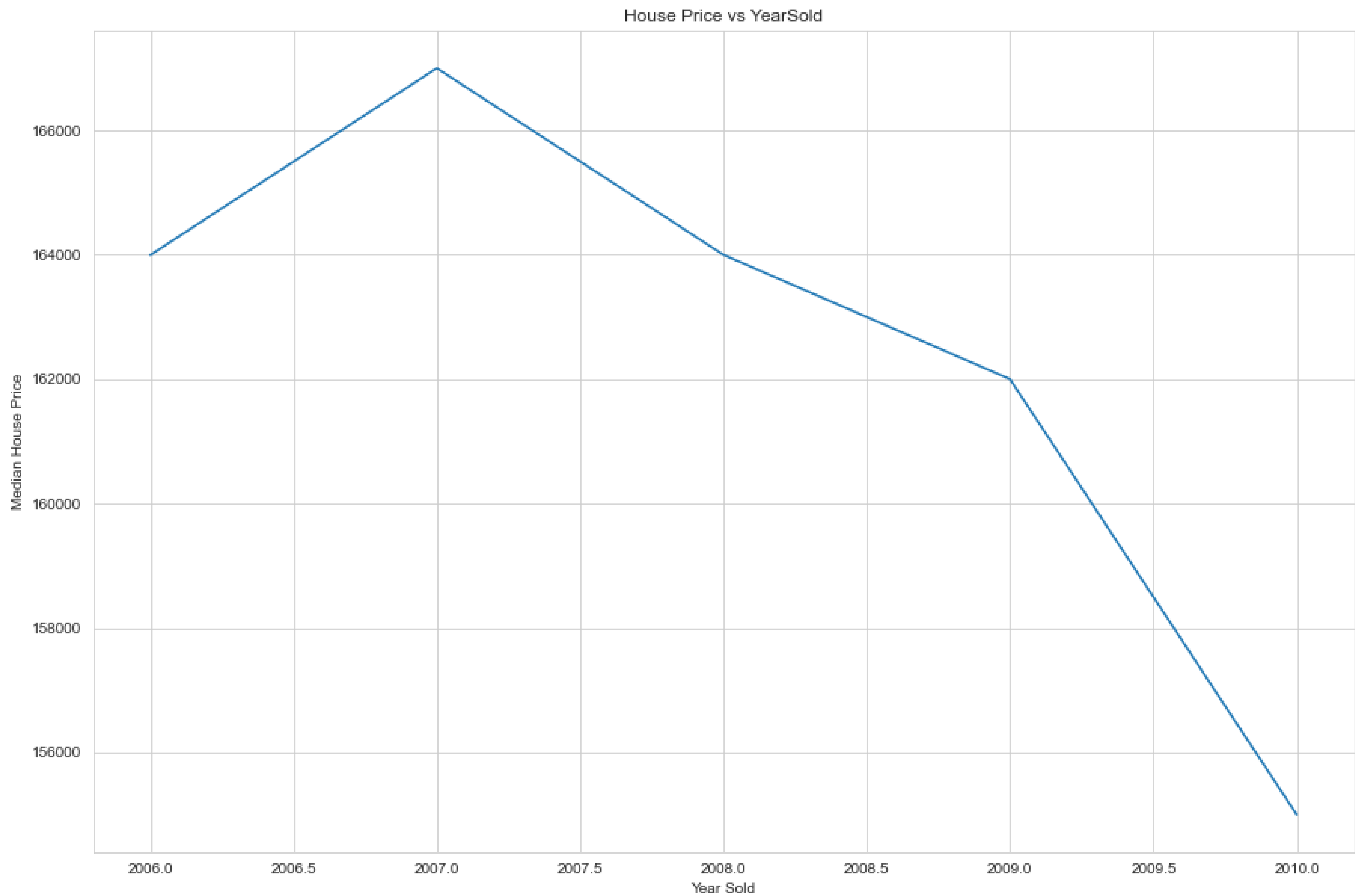
Out[87]:
```
['YearBuilt', 'YearRemodAdd', 'GarageYrBlt', 'YrSold']
```

In [88]:
```
# Explore the content years variables.
for feature in Features_Years:
    print(feature, dataset_train[feature].unique())
```

```
YearBuilt [2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 1965 2005 1962 2006
 1960 1929 1970 1967 1958 1930 2002 1968 2007 1951 1957 1927 1920 1966
 1959 1994 1954 1953 1955 1983 1975 1997 1934 1963 1981 1964 1999 1972
 1921 1945 1982 1998 1956 1948 1910 1995 1991 2009 1950 1961 1977 1985
 1979 1885 1919 1990 1969 1935 1988 1971 1952 1936 1923 1924 1984 1926
 1940 1941 1987 1986 2008 1908 1892 1916 1932 1918 1912 1947 1925 1900
 1980 1989 1992 1949 1880 1928 1978 1922 1996 2010 1946 1913 1937 1942
 1938 1974 1893 1914 1906 1890 1898 1904 1882 1875 1911 1917 1872 1905]
YearRemodAdd [2003 1976 2002 1970 2000 1995 2005 1973 1950 1965 2006 1962 2007 1960
 2001 1967 2004 2008 1997 1959 1990 1955 1983 1980 1966 1963 1987 1964
 1972 1996 1998 1989 1953 1956 1968 1981 1992 2009 1982 1961 1993 1999
 1985 1979 1977 1969 1958 1991 1971 1952 1975 2010 1984 1986 1994 1988
 1954 1957 1951 1978 1974]
GarageYrBlt [2003. 1976. 2001. 1998. 2000. 1993. 2004. 1973. 1931. 1939. 1965. 2005.
 1962. 2006. 1960. 1991. 1970. 1967. 1958. 1930. 2002. 1968. 2007. 2008.
 1957. 1920. 1966. 1959. 1995. 1954. 1953.   nan 1983. 1977. 1997. 1985.
 1963. 1981. 1964. 1999. 1935. 1990. 1945. 1987. 1989. 1915. 1956. 1948.
 1974. 2009. 1950. 1961. 1921. 1900. 1979. 1951. 1969. 1936. 1975. 1971.
 1923. 1984. 1926. 1955. 1986. 1988. 1916. 1932. 1972. 1918. 1980. 1924.
 1996. 1940. 1949. 1994. 1910. 1978. 1982. 1992. 1925. 1941. 2010. 1927.
 1947. 1937. 1942. 1938. 1952. 1928. 1922. 1934. 1906. 1914. 1946. 1908.
 1929. 1933.]
YrSold [2008 2007 2006 2009 2010]
```

In [89]:
```
# Relation between yearSold and the sales price

dataset_train.groupby('YrSold')['SalePrice'].median().plot()
plt.xlabel('Year Sold')
plt.ylabel('Median House Price')
plt.title("House Price vs YearSold")
```

Out[89]: Text(0.5, 1.0, 'House Price vs YearSold')



In [90]:
```
cds.describe()  # to get statistical information of numerical features about dataset.
                #describe() function does not display null values in calulation.
```

Out[90]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1
count	2919.000000	2919.000000	2433.000000	2919.000000	2919.000000	2919.000000	2919.000000	2919.000000	2896.000000	2918.000000
mean	1460.000000	57.137718	69.305795	10168.114080	6.089072	5.564577	1971.312778	1984.264474	102.201312	441.423235
std	842.787043	42.517628	23.344905	7886.996359	1.409947	1.113131	30.291442	20.894344	179.334253	455.610826
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0.000000
25%	730.500000	20.000000	59.000000	7478.000000	5.000000	5.000000	1953.500000	1965.000000	0.000000	0.000000
50%	1460.000000	50.000000	68.000000	9453.000000	6.000000	5.000000	1973.000000	1993.000000	0.000000	368.500000
75%	2189.500000	70.000000	80.000000	11570.000000	7.000000	6.000000	2001.000000	2004.000000	164.000000	733.000000
max	2919.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000

In [91]:
```
cds.describe().shape  # total 38 columns(26 int and 12 float)
```

Out[91]: (8, 38)

Handle Missing Values in Dataset

In [92]:
```
cds.corr() # Pearson corealtion between variables
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_40264\3582175081.py:1: FutureWarning: The default value of numeric_only
in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specif
y the value of numeric_only to silence this warning.
  cds.corr() # Pearson corealtion between variables
```

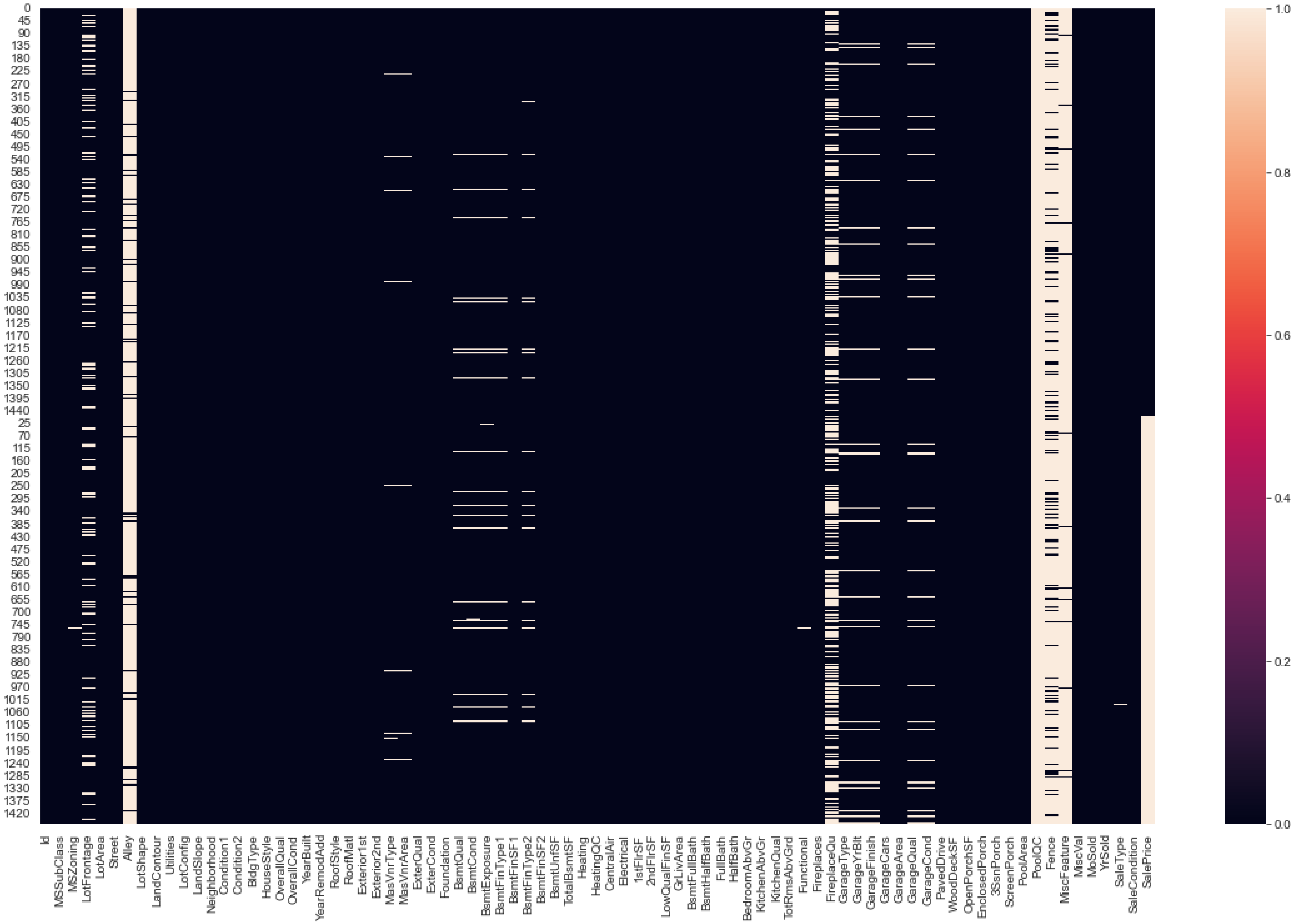
Out[92]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1
Id	1.000000	0.008931	-0.027549	-0.040746	-0.029771	-0.002839	-0.016581	-0.050438	-0.025219	-0.016947
MSSubClass	0.008931	1.000000	-0.417359	-0.201730	0.033638	-0.065625	0.034409	0.043315	0.005433	-0.064311
LotFrontage	-0.027549	-0.417359	1.000000	0.489896	0.217645	-0.075508	0.122811	0.091557	0.221079	0.219408
LotArea	-0.040746	-0.201730	0.489896	1.000000	0.100541	-0.035617	0.024128	0.021612	0.125596	0.194031
OverallQual	-0.029771	0.033638	0.217645	0.100541	1.000000	-0.093847	0.597554	0.571532	0.432947	0.281810
OverallCond	-0.002839	-0.065625	-0.075508	-0.035617	-0.093847	1.000000	-0.368477	0.047654	-0.136007	-0.050418
YearBuilt	-0.016581	0.034409	0.122811	0.024128	0.597554	-0.368477	1.000000	0.612235	0.314051	0.279581
YearRemodAdd	-0.050438	0.043315	0.091557	0.021612	0.571532	0.047654	0.612235	1.000000	0.196875	0.152126
MasVnrArea	-0.025219	0.005433	0.221079	0.125596	0.432947	-0.136007	0.314051	0.196875	1.000000	0.303490
BsmtFinSF1	-0.016947	-0.064311	0.219408	0.194031	0.281810	-0.050418	0.279581	0.152126	0.303490	1.000000
BsmtFinSF2	0.018251	-0.072530	0.047431	0.084059	-0.042771	0.041501	-0.027595	-0.062153	-0.015645	-0.055045
BsmtUnfSF	-0.014453	-0.125994	0.113714	0.021362	0.275175	-0.138202	0.130473	0.165175	0.090163	-0.477404
TotalBsmtSF	-0.024924	-0.219965	0.354822	0.254138	0.549294	-0.174002	0.408515	0.298107	0.397240	0.536467
1stFlrSF	-0.008678	-0.248641	0.458247	0.332460	0.479152	-0.157418	0.310814	0.242245	0.395834	0.458092
2ndFlrSF	-0.022252	0.309309	0.026545	0.031515	0.245596	0.005494	0.017588	0.158985	0.121014	-0.162301
LowQualFinSF	-0.037816	0.026482	0.004894	0.000554	-0.048393	0.009048	-0.144191	-0.060371	-0.057912	-0.066028
GrLivArea	-0.029046	0.071677	0.382462	0.284519	0.575126	-0.116569	0.242666	0.316972	0.402994	0.211669
BsmtFullBath	0.000145	0.009950	0.113245	0.128349	0.164543	-0.042133	0.211580	0.134947	0.141593	0.638847
BsmtHalfBath	0.010387	-0.001878	-0.025629	0.026292	-0.040732	0.084181	-0.030282	-0.046285	0.015006	0.078361
FullBath	-0.009946	0.139140	0.181668	0.125826	0.528483	-0.215504	0.471169	0.457980	0.259777	0.081525
HalfBath	-0.015358	0.178750	0.039452	0.034244	0.272668	-0.088577	0.269743	0.211430	0.191950	-0.007311
BedroomAbvGr	0.003074	-0.008796	0.234892	0.132801	0.073075	-0.008477	-0.053101	-0.021912	0.078126	-0.113547
KitchenAbvGr	-0.011702	0.260155	0.004676	-0.020854	-0.159325	-0.086700	-0.137614	-0.142431	-0.051389	-0.086354
TotRmsAbvGrd	-0.029368	0.040509	0.349513	0.213802	0.389761	-0.092027	0.114280	0.198250	0.278228	0.052141
Fireplaces	-0.035236	-0.055151	0.261970	0.261185	0.390753	-0.030999	0.170680	0.134157	0.275195	0.293089
GarageYrBlt	-0.026666	0.087898	0.076673	-0.008628	0.571803	-0.325849	0.834812	0.652365	0.255112	0.194270
GarageCars	-0.010208	-0.046597	0.310587	0.180434	0.600744	-0.181787	0.538074	0.426022	0.361190	0.255482

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1
GarageArea	-0.008865	-0.103394	0.359786	0.213251	0.565122	-0.154149	0.480735	0.376765	0.374061	0.310449
WoodDeckSF	-0.007056	-0.017654	0.122070	0.158045	0.255317	0.020123	0.229426	0.218513	0.166200	0.223492
OpenPorchSF	0.009960	-0.015923	0.164896	0.104797	0.298084	-0.068978	0.198554	0.242182	0.144650	0.124163
EnclosedPorch	0.021609	-0.020867	0.011509	0.020974	-0.139256	0.071044	-0.374073	-0.220456	-0.111499	-0.099712
3SsnPorch	-0.046538	-0.037529	0.028289	0.015995	0.018715	0.043739	0.015958	0.037433	0.013612	0.050908
ScreenPorch	0.022208	-0.049181	0.075858	0.054375	0.042910	0.043713	-0.041046	-0.046878	0.065209	0.096823
PoolArea	0.014332	-0.003080	0.174119	0.093708	0.030740	-0.016876	0.002304	-0.011407	0.004512	0.084462
MiscVal	0.008244	-0.028867	0.044272	0.069029	0.005562	0.033956	-0.010886	-0.003124	0.044811	0.093295
MoSold	0.006448	-0.001231	0.011254	0.004156	0.030405	-0.006256	0.013938	0.017693	-0.000117	-0.000942
YrSold	-0.256050	-0.015028	-0.007917	-0.024234	-0.019614	0.030102	-0.012344	0.033203	-0.018510	0.022556
SalePrice	-0.021917	-0.084284	0.351799	0.263843	0.790982	-0.077856	0.522897	0.507101	0.477493	0.386420

In [93]:
```
plt.figure(figsize=(20,12))
sns.heatmap(cds.isnull())
# It contains valuees in 0 and 1 format;Black cplour represent 0(not null) and full white colour represent 1,
#means where null values are present.


### plt.savefig("C:\Users/heatmap_of_CDS_null_values.png")***Not able to save..Must chk later
```

Out[93]:
```
<AxesSubplot:>
```



0
45
90
135
180
225
270
315
360
405
450
495
540
585
630
675
720
765
810
855
900
945
990
1035
1080
1125
1170
1215
1260
1305
1350
1395
1440
25
70
115
160
205
250
295
340
385
430
475
520
565
610
655
700
745
790
835
880
925
970
1015
1060
1105
1150
1195
1240
1285
1330
1375
1420
Id
MSSubClass
MSZoning
LotFrontage
LotArea
Street
Alley
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
BldgType
HouseStyle
OverallQual
OverallCond
YearBuilt
YearRemodAdd
RoofStyle
RoofMatl
Exterior1st
Exterior2nd
MasVnrType
MasVnrArea
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinSF1
BsmtFinType2
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
Heating
HeatingQC
CentralAir
Electrical
1stFlrSF
2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual
TotRmsAbvGrd
Functional
Fireplaces
FireplaceQu
GarageType
GarageYrBlt
GarageFinish
GarageCars
GarageArea
GarageQual
GarageCond
PavedDrive
WoodDeckSF
OpenPorchSF
EnclosedPorch
3SsnPorch
ScreenPorch
PoolArea
PoolQC
Fence
MiscFeature
MiscVal
MoSold
YrSold
SaleType
SaleCondition
SalePrice
1.0
0.8
0.6
0.4
0.2
0.0

In [94]:
```
# As we can see in above heatmap, most null value features are:
#Alley, FireplaceQu,PoolIQC, Fence,MiscFeature
```

In [95]:
```
pd.set_option("display.max_columns", None) # This function used to display all rows and columns
pd.set_option("display.max_rows", None)
```

In [96]:
```
#set ID column as Index
cds=cds.set_index("Id")


nullvalues_count=cds.isnull().sum() # Count null values presesnt in each faetures
nullvalues_count
```

Out[96]:
```
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 [97]:
```
percentage_nullvalues = cds.isnull().sum()/cds.shape[0]*100
percentage_nullvalues    # Display % of null values presesnt in each faetures.
```

Out[97]:

```
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 [98]:
```
# Here we do not have any threshold value to conclude which feature to remove and which not to.
```

2: Fetature Engineering:

Drop Variables/Columns:

In [99]:
```
miss_values_20_percent=percentage_nullvalues[percentage_nullvalues>20]
miss_values_20_percent
```

Out[99]:
```
Alley          93.216855
FireplaceQu    48.646797
PoolQC         99.657417
Fence          80.438506
MiscFeature    96.402878
SalePrice      49.982871
dtype: float64
```

In [100]:
```
cds["Alley"].value_counts() # Here,other than Grvl and Pave, there is None values present.
```

Out[100]:
```
Grvl    120
Pave     78
Name: Alley, dtype: int64
```

In [101]:
```
# As per domain given description and knowledge, I will not drop Alley features,
# instead None values, I will add  a constant value"NA"
```

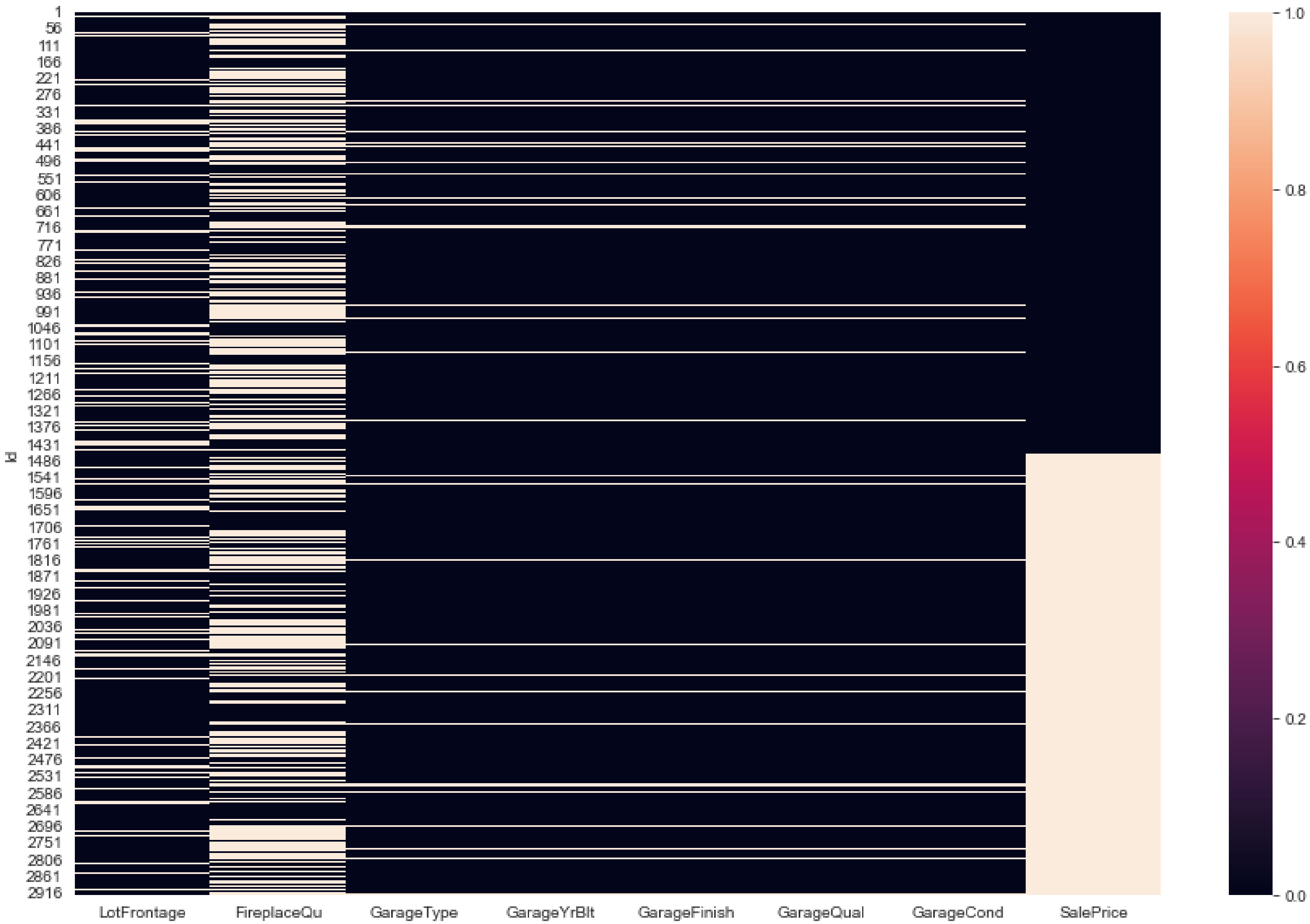
In [102]:
```
miss_values_5_50_percent=percentage_nullvalues[(percentage_nullvalues>5) & (percentage_nullvalues<51)]
miss_values_5_50_percent
```

Out[102]:
```
LotFrontage     16.649538
FireplaceQu     48.646797
GarageType       5.378554
GarageYrBlt      5.447071
GarageFinish     5.447071
GarageQual       5.447071
GarageCond       5.447071
SalePrice       49.982871
dtype: float64
```

In [103]:
```
#Here, If I check FireplaceQu in data description, NA means no fireplace.
#As per domain given description and knowledge, I will not drop FireplaceQu,
# instead None values, I will add  a constant value"NA"
```

In [104]:
```
sns.heatmap(cds[miss_values_5_50_percent.keys()].isnull()) # isnull means , no missing values, there is NA value.
```

Out[104]: <AxesSubplot:ylabel='Id'>



In [105]:
```
# If there is no garage, there will not be GarageYrBlt,GrgQuality, GarageCondition. So as per domain knowledge,
#we can not drop these features.
```

In [ ]:

Missing Values Imputation/Replace.

In [106]:
```
missingvalues_features=percentage_nullvalues[percentage_nullvalues>0]
print("Total of missingValues features:",len(missingvalues_features))
```

```
Total of missingValues features: 35
```

In [107]:
```
missingvalues_features
```

Out[107]:
```
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 [108]:
```
Cat_Na_Feature = missingvalues_features[missingvalues_features.keys().isin(categorical_features)]
print("Total of Categorical NA Features:",len(Cat_Na_Feature))
Cat_Na_Feature
```

```
Total of Categorical NA Features: 23
```

Out[108]:
```
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 [109]:
```
Numerical_Na_Feature = missingvalues_features[missingvalues_features.keys().isin(Numerical_Features)]
print("Total of Numerical NA Features:",len(Numerical_Na_Feature))
Numerical_Na_Feature
```

```
Total of Numerical NA Features: 12
```

Out[109]:
```
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=0.137033

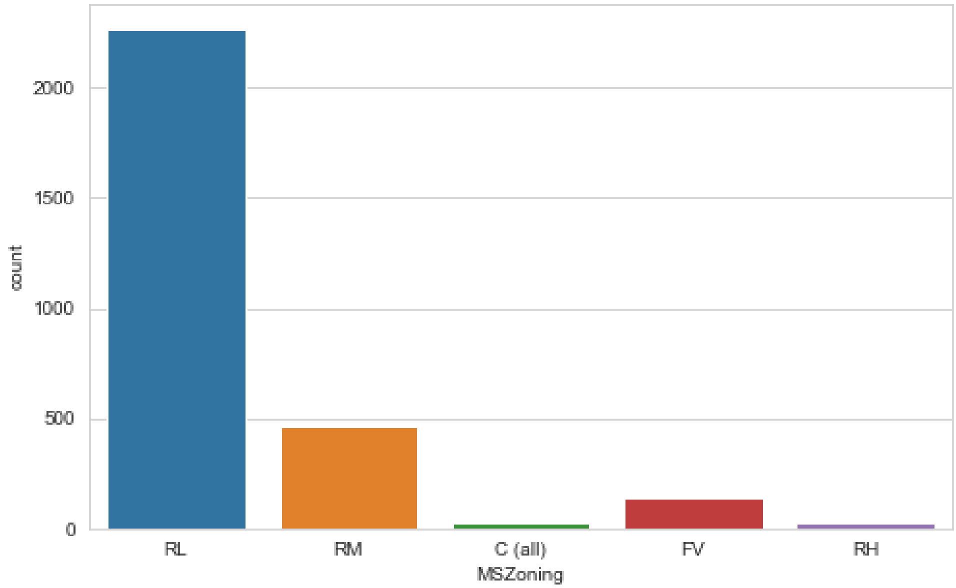
In [110]:
```
###Backup of Orizinal data###
cds_mvi=cds.copy()
cds_mvi.shape

#Here in output we have only 80 features instead of 81, Because we set "Id" as a Index feature earlier.
```

Out[110]:
```
(2919, 80)
```

In [111]:
```
cds["MSZoning"].value_counts() # Display MSZoning categories.
```

Out[111]:
```
RL         2265
RM          460
FV          139
RH           26
C (all)      25
Name: MSZoning, dtype: int64
```

In [112]:
```
#sns.countplot(cds["MSZoning"])
#plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='MSZoning',data=cds)
plt.show()
```



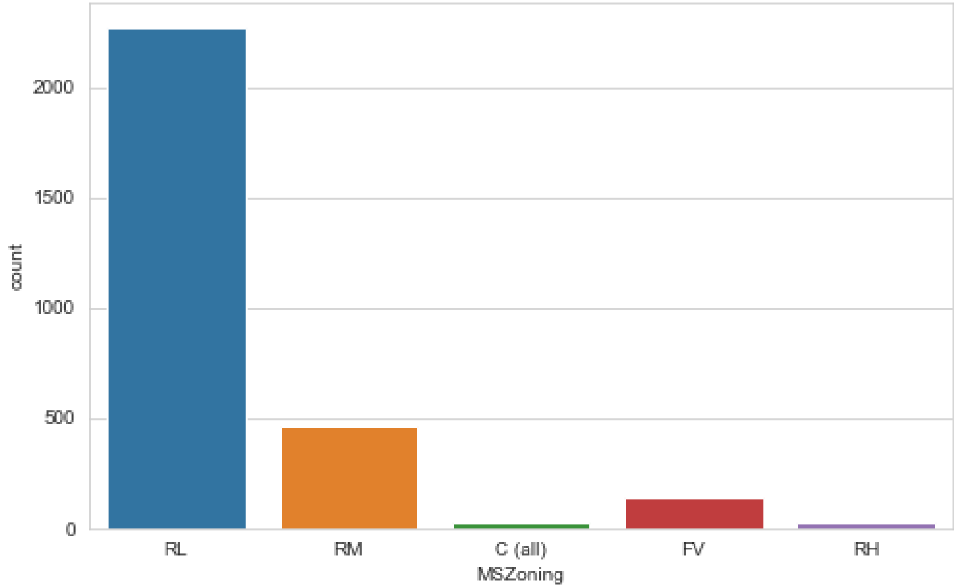
In [113]:
```
msz_mode=cds_mvi["MSZoning"].mode()[0]
msz_mode
```

Out[113]: 'RL'

In [114]:
```
msz_mode=cds_mvi["MSZoning"].mode()[0]
cds_mvi["MSZoning"].replace(np.nan,msz_mode,inplace=True)
cds_mvi["MSZoning"].isnull().sum()
```

Out[114]: 0

In [115]:
```
plt.figure(figsize=(8,5))
sns.countplot(x ='MSZoning', data = cds_mvi)
plt.show()
```



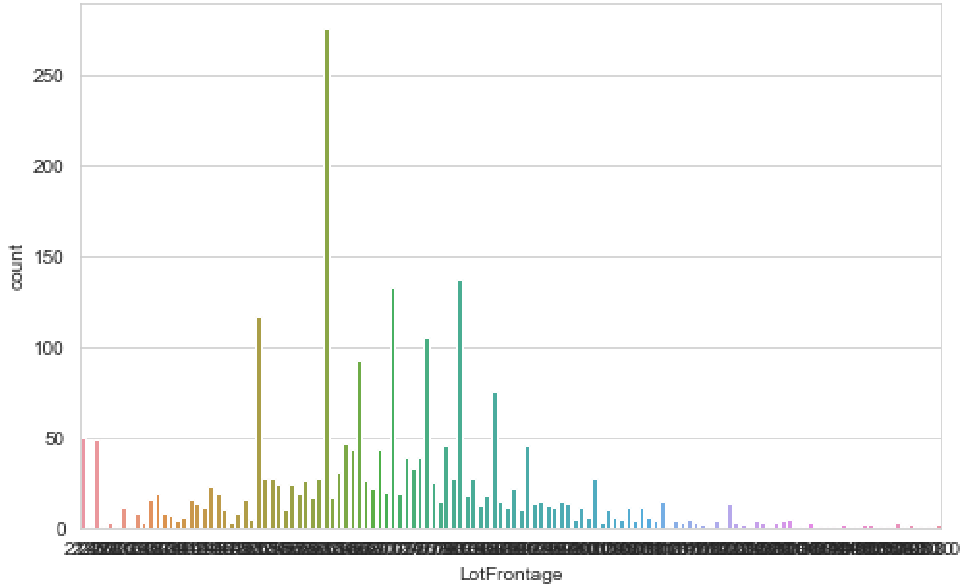
In [116]:
```
nullvalues_count=cds_mvi.isnull().sum() # Count null values presesnt in each faetures
nullvalues_count
```

Out[116]:
```
MSSubClass         0
MSZoning           0
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
```

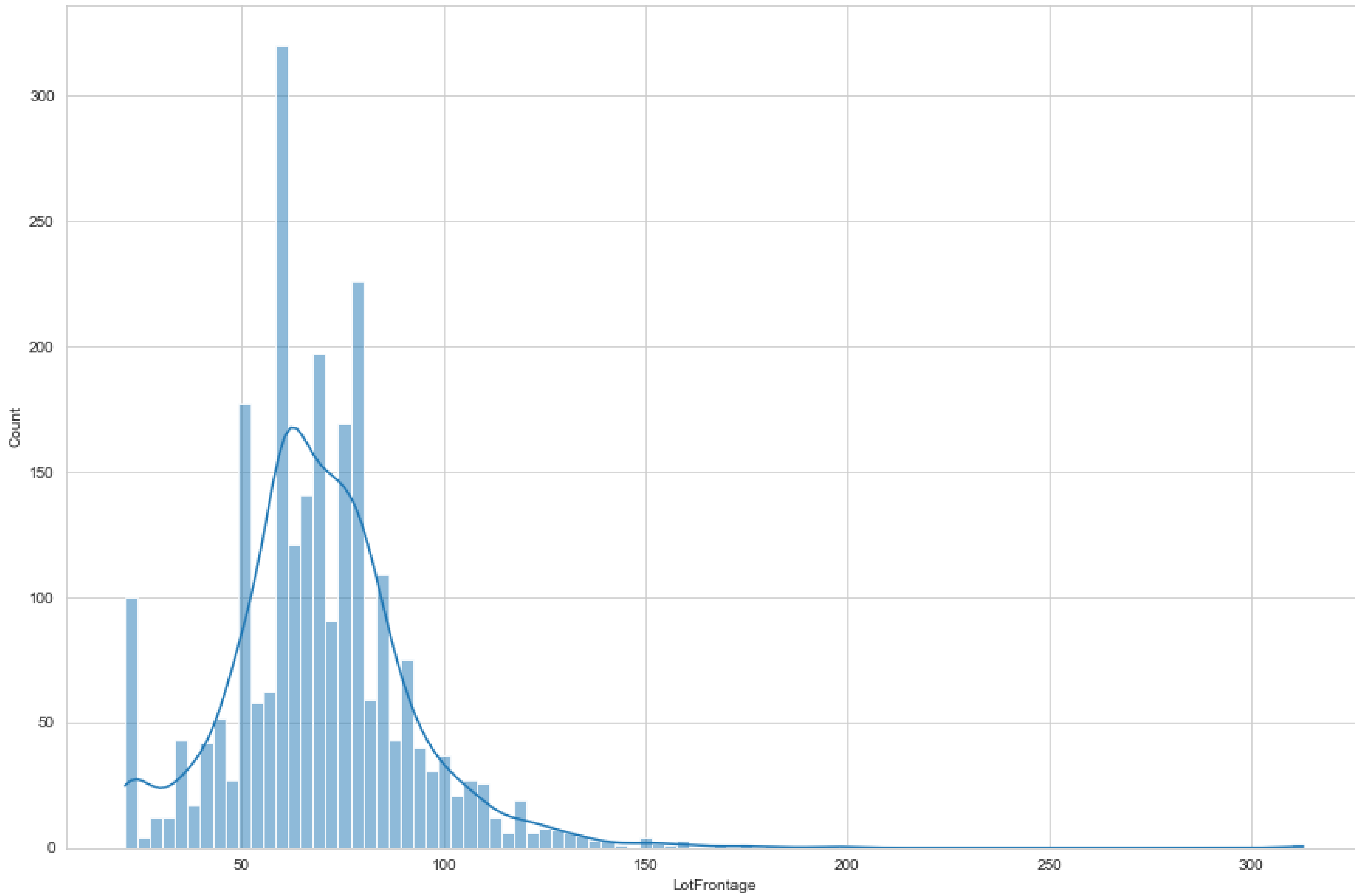
```
LowQualFinSF        0
GrLivArea           0
BsmtFullBath        2
BsmtHalfBath        2
FullBath            0
HalfBath            0
BedroomAbvGr        0
KitchenAbvGr        0
KitchenQual         1
TotRmsAbvGrd        0
Functional          2
Fireplaces          0
FireplaceQu      1420
GarageType        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
```

Handling LotFrontage= 486

In [117]:
```
plt.figure(figsize=(8,5))
sns.countplot(x='LotFrontage',data=cds)
plt.show()
sns.histplot(x='LotFrontage',data=cds, kde=True)# Kernel density estimate to smooth the histogram.
```

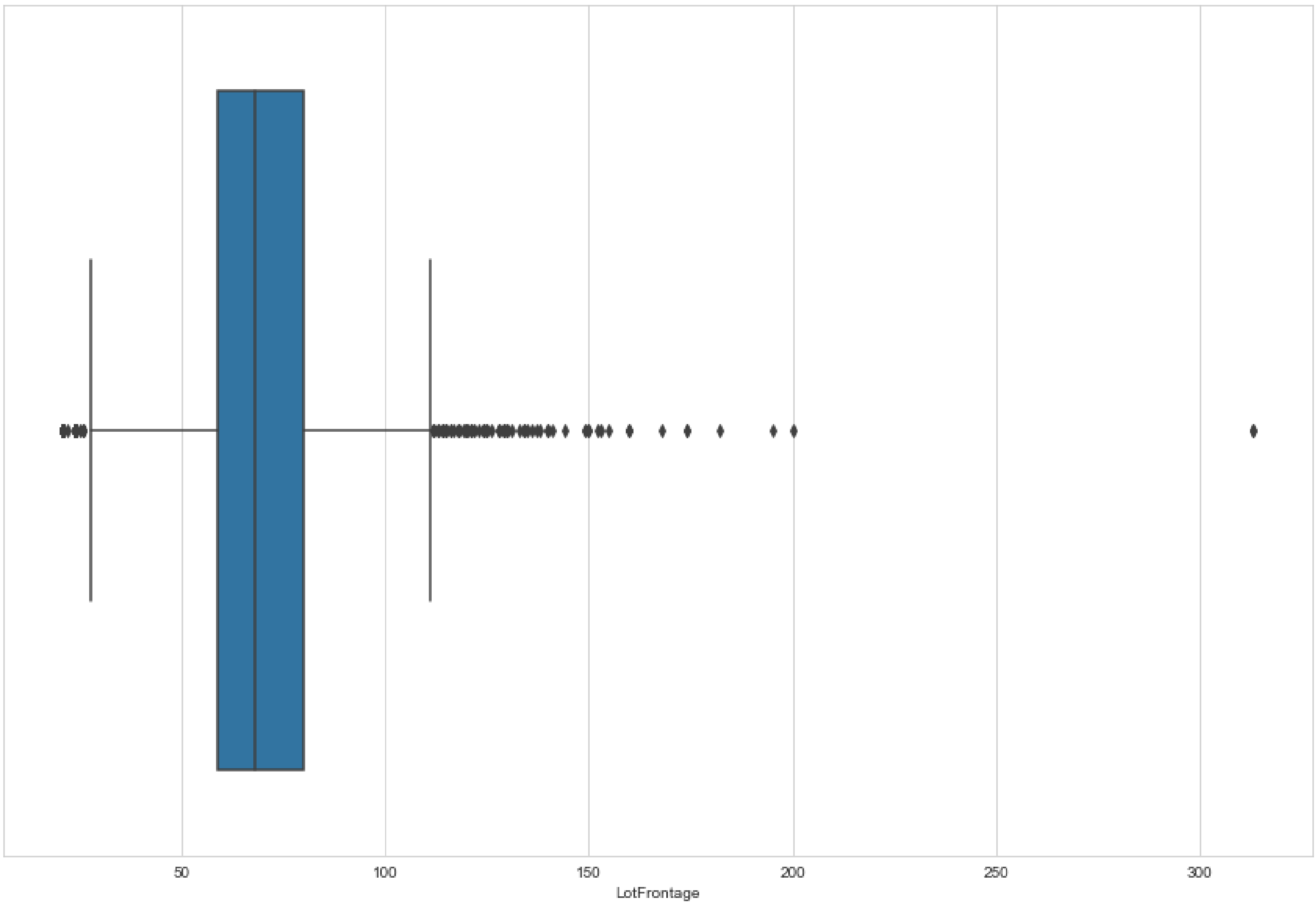


Out[117]:
```
<AxesSubplot:xlabel='LotFrontage', ylabel='Count'>
```


300
250
200
150
100
50
0
Count
50
100
150
200
250
300
LotFrontage

In [118]:
```
sns.boxplot(x='LotFrontage',data=cds)
```

Out[118]: <AxesSubplot:xlabel='LotFrontage'>



In [119]:
```
### From Above graphs, we can see that our data is right skiewed, not in proper bell shape.
## If we have left/right skiewd data, we have to take median value(as it is less senstive to outliers),
#not mean for data imputation.
```

In [120]:
```
LotFrontage_Median=cds["LotFrontage"].median()
LotFrontage_Median
```

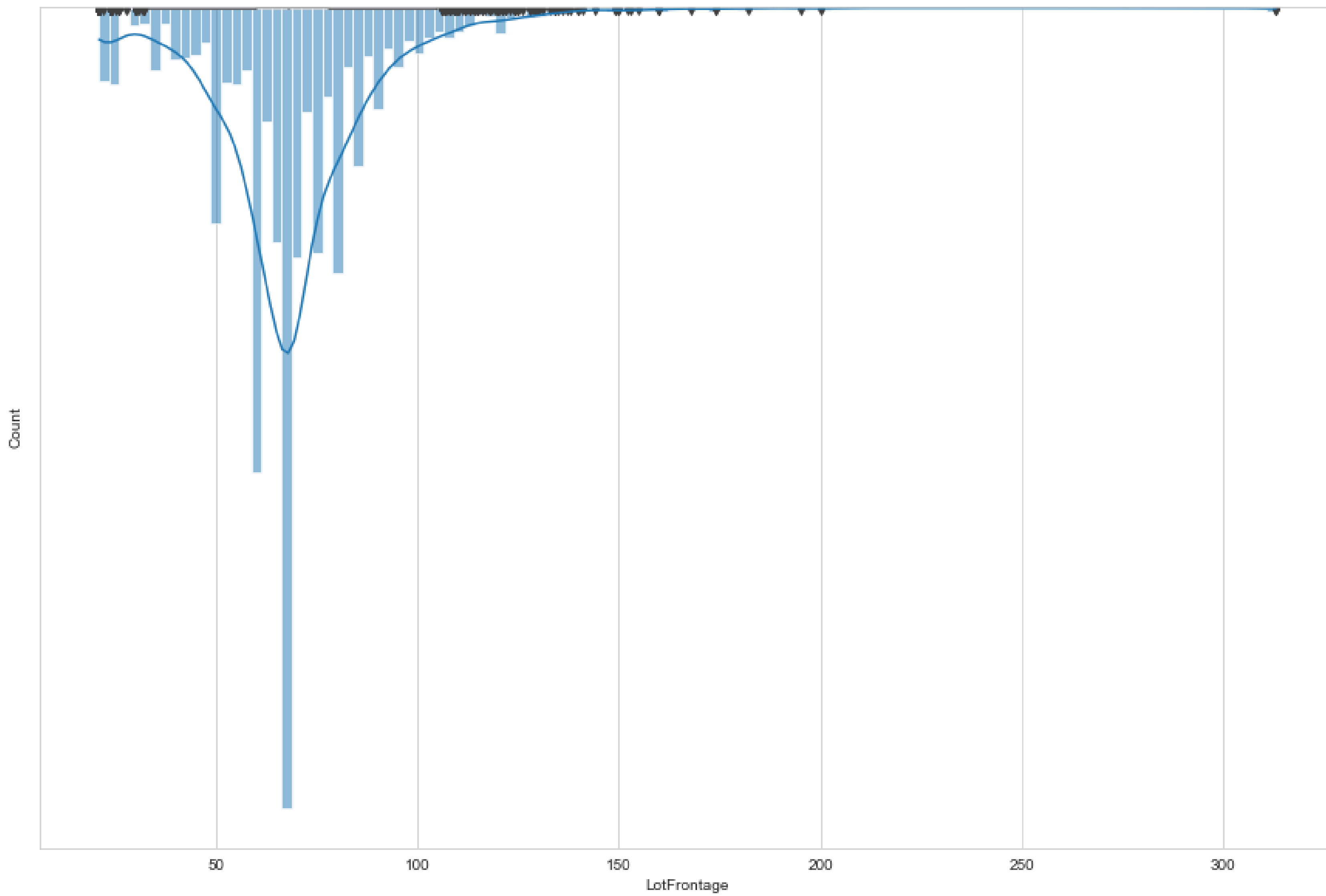
Out[120]: 68.0

In [121]:
```
LotFrontage_Median=cds["LotFrontage"].median()
cds_mvi["LotFrontage"].replace(np.nan,LotFrontage_Median,inplace=True)
cds_mvi["LotFrontage"].isnull().sum()
```

Out[121]: 0

In [122]:
```
sns.boxplot(x='LotFrontage',data=cds_mvi)
sns.histplot(x='LotFrontage',data=cds_mvi, kde=True)# Kernel density estimate to smooth the histogram.
```

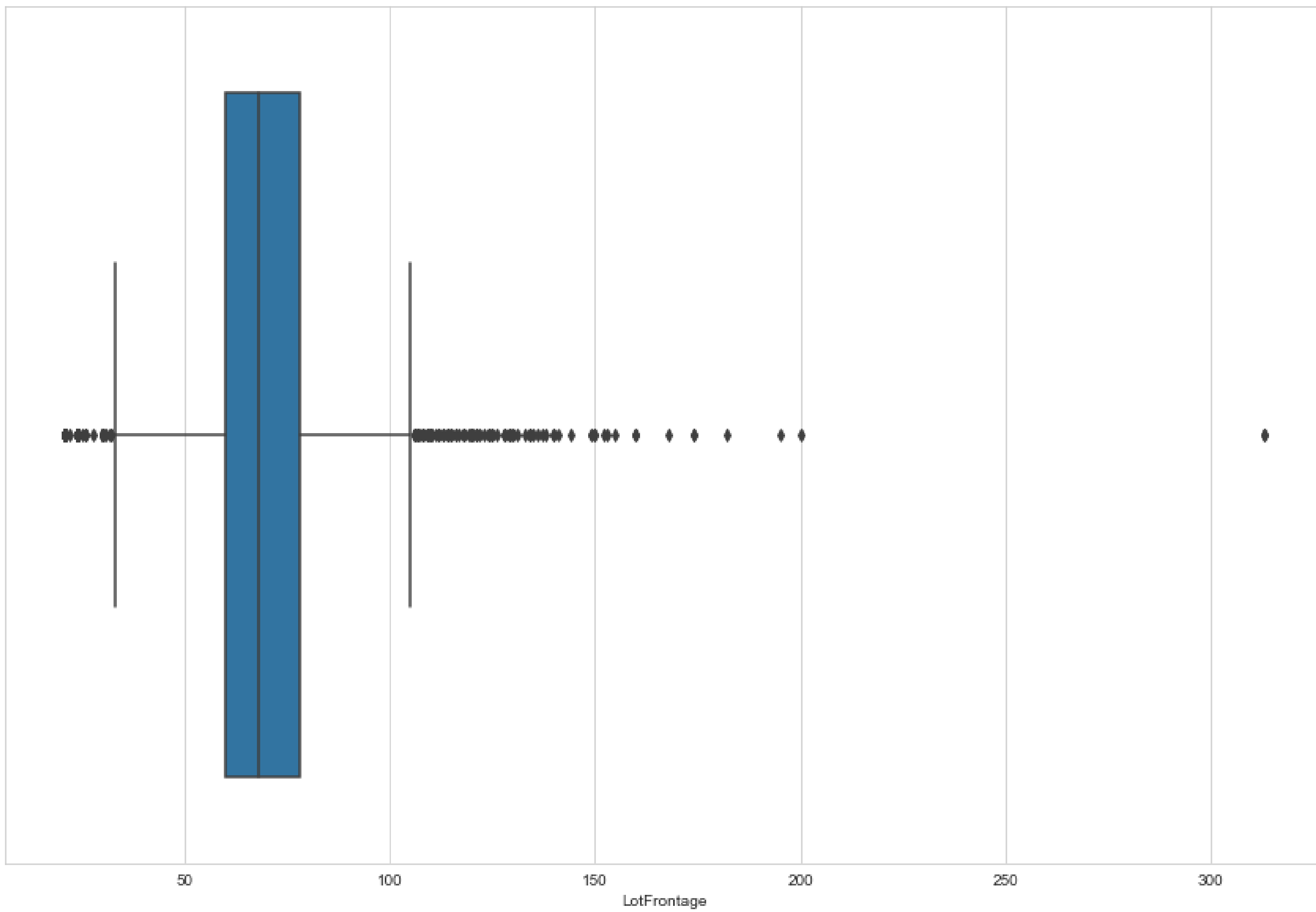
Out[122]: <AxesSubplot:xlabel='LotFrontage', ylabel='Count'>



In [123]:

```
sns.boxplot(x='LotFrontage',data=cds_mvi)
```

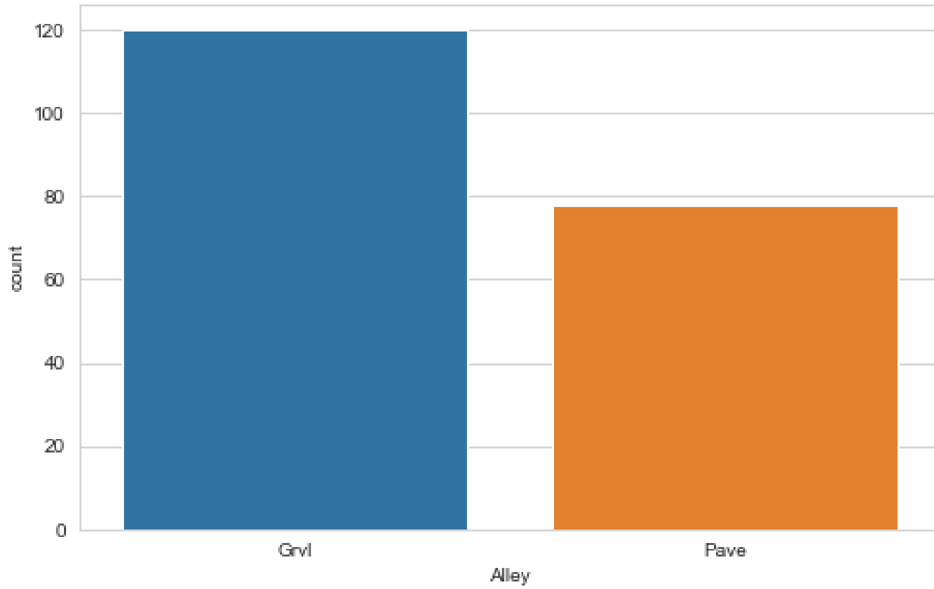
Out[123]: <AxesSubplot:xlabel='LotFrontage'>



In [124]:
```
#As we can see after imputation, our data is not so much disturbed, only left side outliers increased.
```

Handling Alley=93.216855

In [125]:
```
plt.figure(figsize=(8,5))
sns.countplot(x='Alley',data=cds)
plt.show()
```



In [126]:
```
cds["Alley"].value_counts() # Display Alley categories.
```

Out[126]:
```
Grvl    120
Pave     78
Name: Alley, dtype: int64
```

In [127]:
```
Alley_Count="NA"    ## I added "NA" constant
cds_mvi["Alley"].replace(np.nan,Alley_Count,inplace=True)
cds_mvi["Alley"].isnull().sum()
```

Out[127]:
```
0
```

Handling Utilities=0.06851

In [128]:
```
cds["Utilities"].value_counts()
```

Out[128]:
```
AllPub    2916
NoSeWa       1
Name: Utilities, dtype: int64
```

In [129]:
```
utilities_Mode=cds["Utilities"].mode()[0]
cds_mvi["Utilities"].replace(np.nan,utilities_Mode,inplace=True)
cds_mvi["Utilities"].isnull().sum()
```

Out[129]:
```
0
```

Exterior1st=0.034 and Exterior2nd=0.34258

In [130]:
```
cds["Exterior1st"].value_counts()
```

Out[130]:
```
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 [131]:
```
cds["Exterior2nd"].value_counts()# Both of thaem almost similar values, I will impute mode vaue here.
```

Out[131]:
```
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 [132]:
```
exterior1st_Mode=cds["Exterior1st"].mode()[0]
exterior3nd_Mode=cds["Exterior2nd"].mode()[0]

cds_mvi["Exterior1st"].replace(np.nan,exterior1st_Mode,inplace=True)
cds_mvi["Exterior2nd"].replace(np.nan,exterior3nd_Mode,inplace=True)
print("Ext1s is null:",cds_mvi["Exterior1st"].isnull().sum())
print("Ext2nd is null:",cds_mvi["Exterior2nd"].isnull().sum())
```

```
Ext1s is null: 0
Ext2nd is null: 0
```

Handling MasVnrType=0.822199(Categorical) and MasVnrArea=0.787941(Numerical)

In [133]:
```
cds["MasVnrType"].value_counts()# After run, we can see there is already None category available, So I ll compute Mod
```

Out[133]:
```
None       1742
BrkFace     879
Stone       249
BrkCmn       25
Name: MasVnrType, dtype: int64
```

In [134]:
```
masvantype_Mode=cds["MasVnrType"].mode()[0]
cds_mvi["MasVnrType"].replace(np.nan,masvantype_Mode,inplace=True)
print("masvantype_Mode is null:",cds_mvi["Exterior1st"].isnull().sum())
```

```
masvantype_Mode is null: 0
```

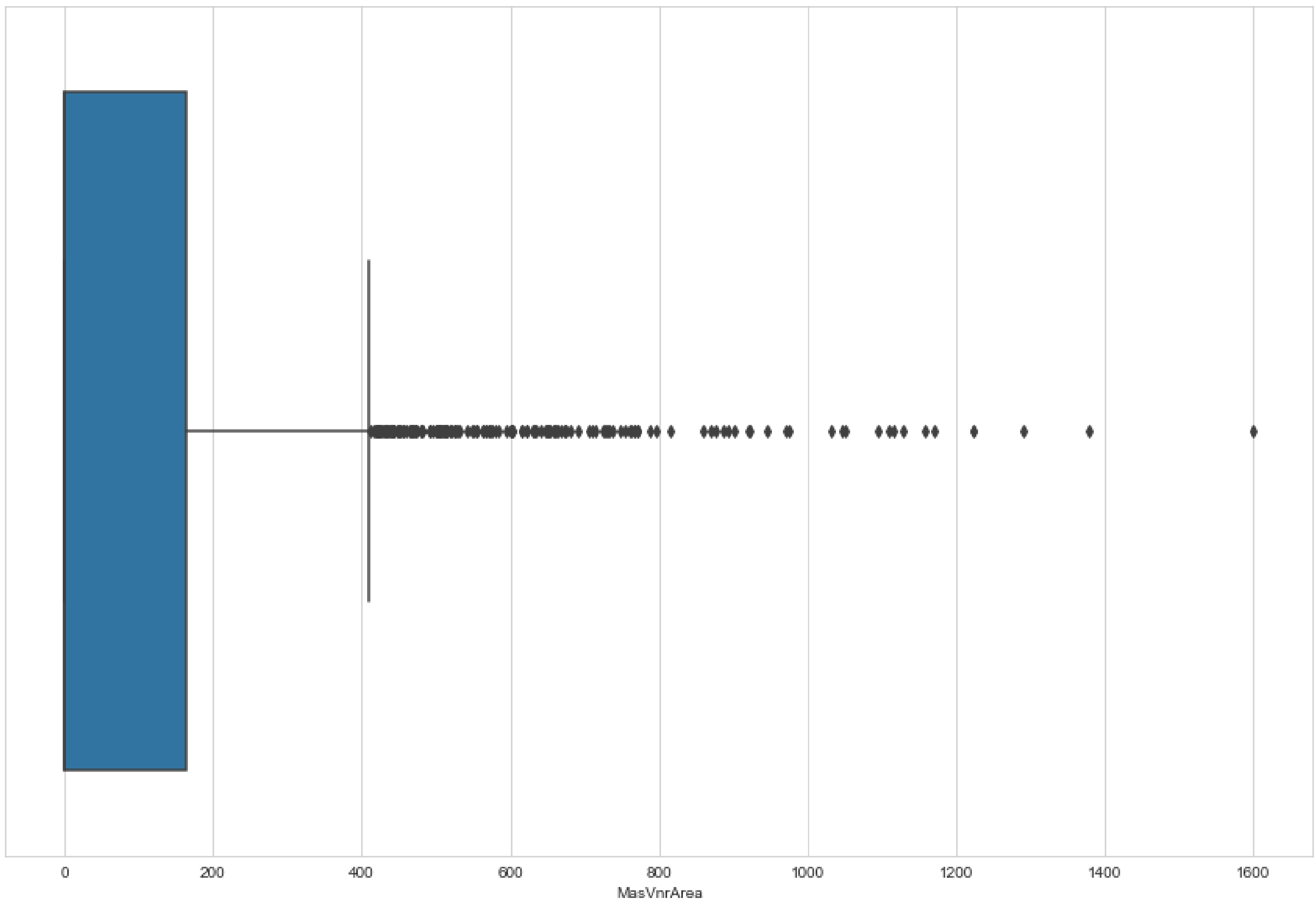
In [135]:
```
cds["MasVnrArea"].value_counts()
```

```
150.0         5
172.0         5
82.0          5
182.0         5
100.0         5
206.0         5
162.0         5
272.0         5
194.0         5
136.0         5
68.0          5
88.0          5
226.0         4
450.0         4
14.0          4
480.0         4
94.0          4
336.0         4
166.0         4
248.0         4
```

In [136]:

```
sns.boxplot(x='MasVnrArea',data=cds)

# Here in boxplot, we can see, there is so many outliers.So its better to impute zero value rather than mode and medi
```

Out[136]: <AxesSubplot:xlabel='MasVnrArea'>



In [137]:
```
masvnrarea_constant=0
cds_mvi["MasVnrArea"].replace(np.nan,masvnrarea_constant,inplace=True)
print("masvantype_Mode is null:",cds_mvi["MasVnrArea"].isnull().sum())
```

```
masvantype_Mode is null: 0
```

Handling Basement Features:-

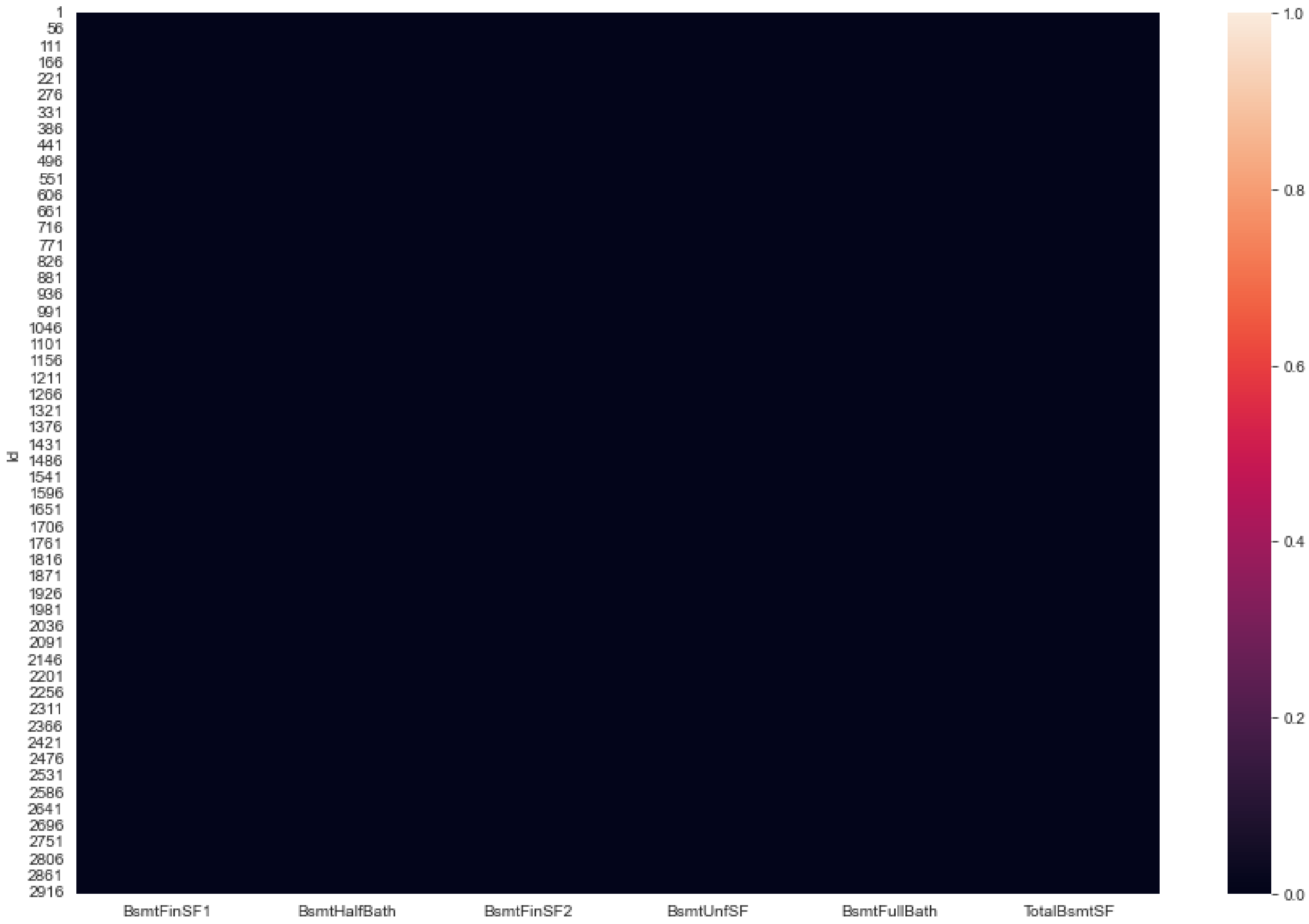
In [138]:
```

#Numerical_bsmt_feature
BsmtFinSF1       0.034258
BsmtHalfBath     0.068517
BsmtFinSF2       0.034258
BsmtUnfSF        0.034258
BsmtFullBath     0.068517
TotalBsmtSF      0.034258

Catagorical_bsmt_feature
BsmtQual         2.774923
BsmtExposure     2.809181
BsmtFinType1     2.706406
BsmtFinType2     2.740665
BsmtCond         2.809181

#####################Just for Refrences##############
```

```
  Input In [138]
    BsmtFinSF1       0.034258
                     ^
SyntaxError: invalid syntax
```

In [139]:
```
Numerical_bsmt_feature=["BsmtFinSF1",
"BsmtHalfBath",
"BsmtFinSF2",
"BsmtUnfSF",
"BsmtFullBath",
"TotalBsmtSF"]

Catagorical_bsmt_feature=["BsmtQual","BsmtExposure","BsmtFinType1","BsmtFinType2","BsmtCond"]
```

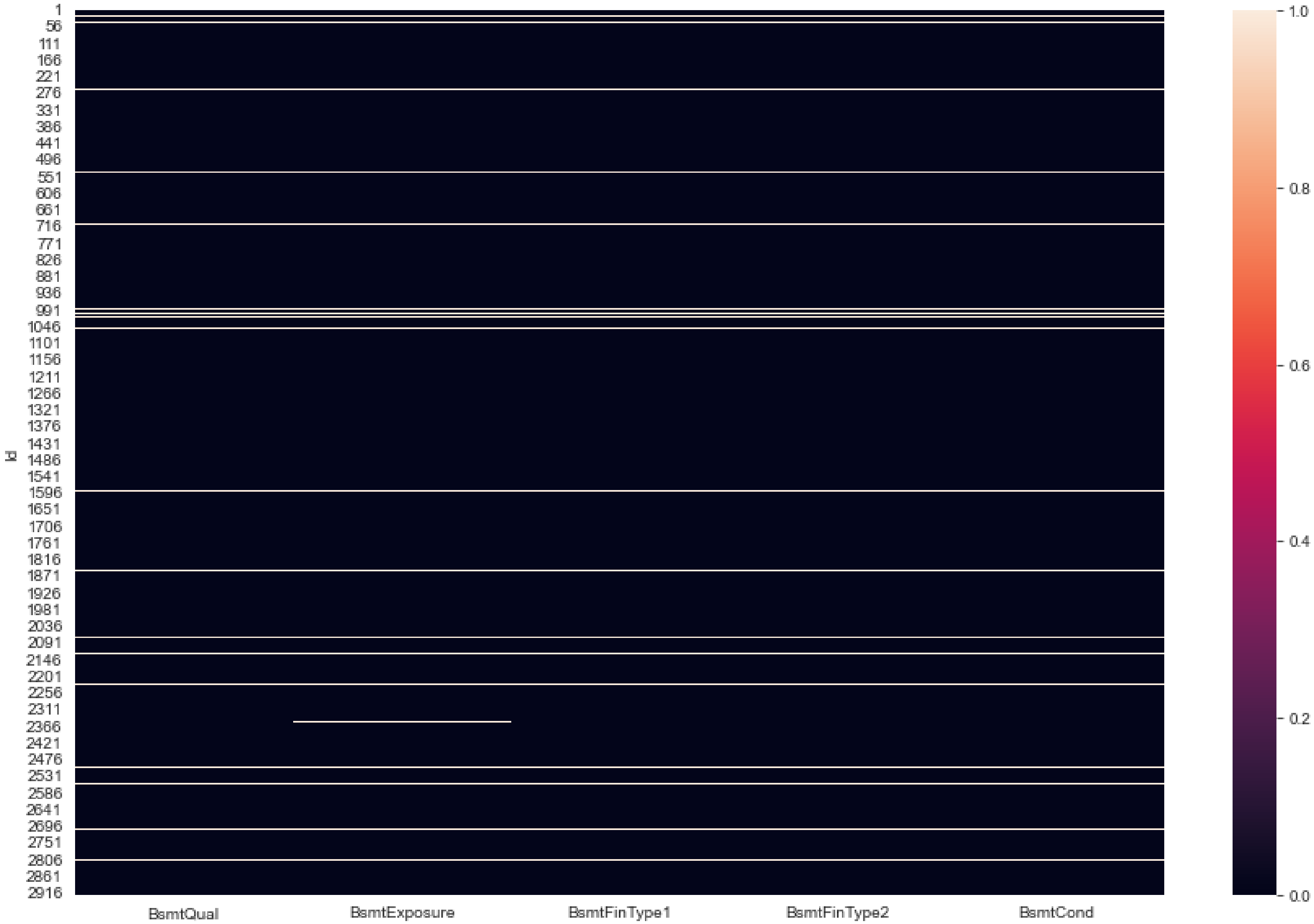
In [140]:
```
sns.heatmap(cds[Numerical_bsmt_feature].isnull())
```

Out[140]: <AxesSubplot:ylabel='Id'>



In [141]:
```
sns.heatmap(cds[Catagorical_bsmt_feature].isnull())
```

Out[141]: <AxesSubplot:ylabel='Id'>



In [144]:
```
for feature in Catagorical_bsmt_feature:
    print(f"value count of {feature}:{cds[feature].value_counts()}")
```

```
value count of BsmtQual:TA    1283
Gd    1209
Ex     258
Fa      88
Name: BsmtQual, dtype: int64
value count of BsmtExposure:No    1904
Av     418
Gd     276
Mn     239
Name: BsmtExposure, dtype: int64
value count of BsmtFinType1:Unf    851
GLQ    849
ALQ    429
Rec    288
BLQ    269
LwQ    154
Name: BsmtFinType1, dtype: int64
value count of BsmtFinType2:Unf    2493
Rec     105
LwQ      87
BLQ      68
ALQ      52
GLQ      34
Name: BsmtFinType2, dtype: int64
value count of BsmtCond:TA    2606
Gd     122
Fa     104
Po       5
Name: BsmtCond, dtype: int64
```

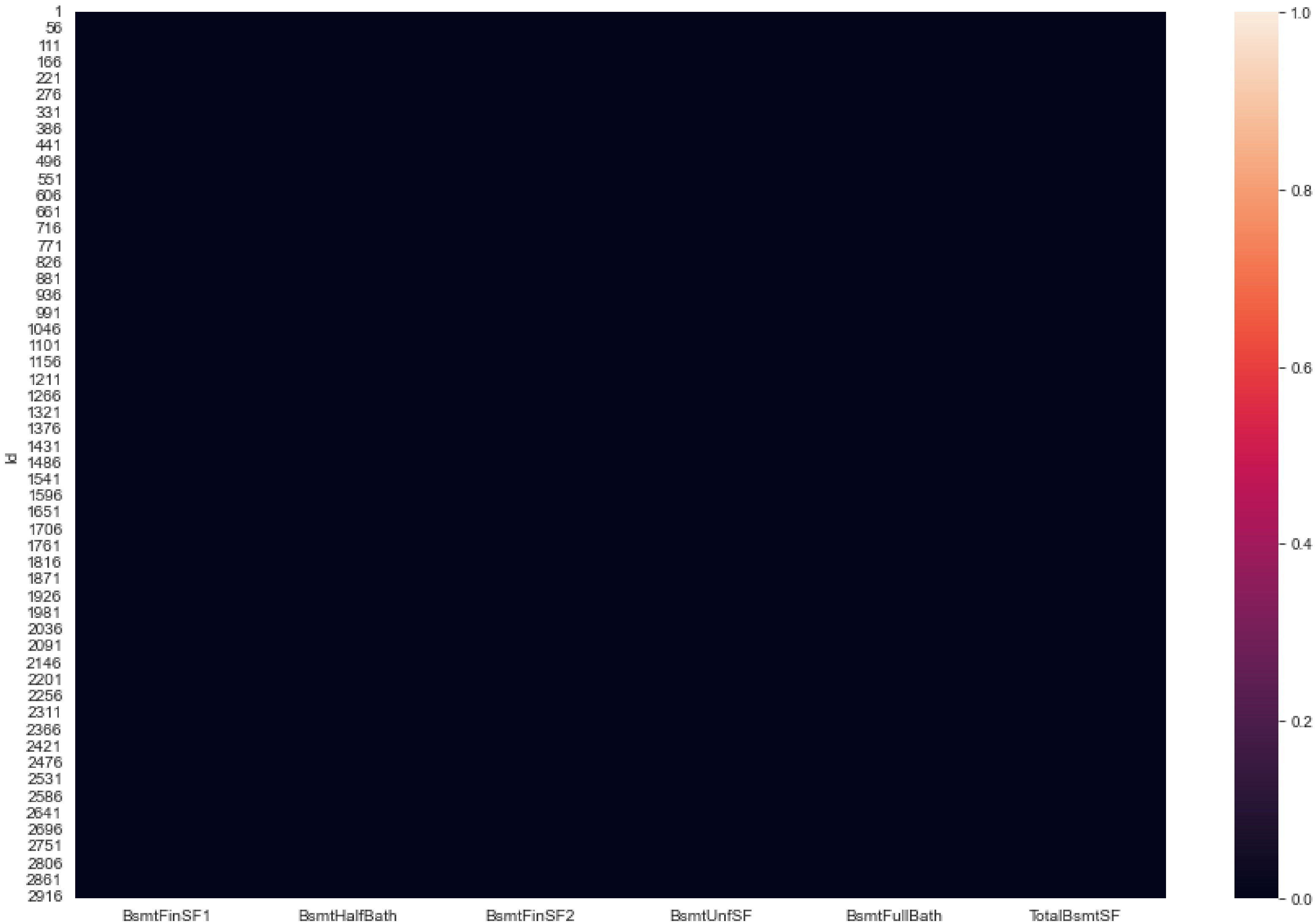
In [145]:
```
bsmt_constant="NA"
for feature in Catagorical_bsmt_feature:
    cds_mvi[feature].replace(np.nan,bsmt_constant,inplace=True)
```

In [146]:
```
cds_mvi[Catagorical_bsmt_feature].isnull().sum()
```

Out[146]:
```
BsmtQual        0
BsmtExposure    0
BsmtFinType1    0
BsmtFinType2    0
BsmtCond        0
dtype: int64
```

In [147]:
```
sns.heatmap(cds[Numerical_bsmt_feature].isnull()) # Heatmap of Numerical Features.
#Here we are not able to see any missing value because % age of missing values is very less.
#BsmtFinSF1       0.034258
#BsmtHalfBath     0.068517
#BsmtFinSF2       0.034258
#BsmtUnfSF        0.034258
#BsmtFullBath     0.068517
#TotalBsmtSF      0.034258
```

Out[147]:
```
<AxesSubplot:ylabel='Id'>
```



In [148]:

```
bsmt_num_constant=0
for feature in Numerical_bsmt_feature:
    cds_mvi[feature].replace(np.nan,bsmt_num_constant,inplace=True)
```

In [149]:
```
cds_mvi[Numerical_bsmt_feature].isnull().sum()# To Check , is there any missing value available-Crosscheck
```

Out[149]:
```
BsmtFinSF1      0
BsmtHalfBath    0
BsmtFinSF2      0
BsmtUnfSF       0
BsmtFullBath    0
TotalBsmtSF     0
dtype: int64
```

Handling Electrical =0.034258 and KitchenQual=0.034258

In [150]:
```
cds["Electrical"].value_counts()
```

Out[150]:
```
SBrkr    2671
FuseA     188
FuseF      50
FuseP       8
Mix         1
Name: Electrical, dtype: int64
```

In [151]:
```
cds["KitchenQual"].value_counts()
```

Out[151]:
```
TA    1492
Gd    1151
Ex     205
Fa      70
Name: KitchenQual, dtype: int64
```

In [152]:
```
elt_Mode=cds["Electrical"].mode()[0]
cds_mvi["Electrical"].replace(np.nan,elt_Mode,inplace=True)
print("Elt is null:",cds_mvi["Electrical"].isnull().sum())

KitchenQ_Mode=cds["KitchenQual"].mode()[0]
cds_mvi["KitchenQual"].replace(np.nan,KitchenQ_Mode,inplace=True)
print("KitchenQ is null:",cds_mvi["KitchenQual"].isnull().sum())
```

```
Elt is null: 0
KitchenQ is null: 0
```

Handling Remaining Categorical Features:

In [153]:
```
#Functional       0.068517-Mode
#FireplaceQu     48.646797-NA
#PoolQC          99.657417-NA
#Fence           80.438506-NA
#MiscFeature     96.402878-NA
#SaleType        0.034258-Mode


## I will impute mode values in Functional and saleType Cat-Feature and in remaining, will put NA values.
```

In [154]:
```
cds["Functional"].value_counts()
```

Out[154]:
```
Typ     2717
Min2      70
Min1      65
Mod       35
Maj1      19
Maj2       9
Sev        2
Name: Functional, dtype: int64
```

In [155]:
```
cds["SaleType"].value_counts()
```

Out[155]:
```
WD       2525
New       239
COD        87
ConLD      26
CWD        12
ConLI       9
ConLw       8
Oth         7
Con         5
Name: SaleType, dtype: int64
```

In [156]:
```
function_Mode=cds["Functional"].mode()[0]
cds_mvi["Functional"].replace(np.nan,function_Mode,inplace=True)
print("Function is null:",cds_mvi["Functional"].isnull().sum())

saletype_Mode=cds["SaleType"].mode()[0]
cds_mvi["SaleType"].replace(np.nan,saletype_Mode,inplace=True)
print("SaleType is null:",cds_mvi["SaleType"].isnull().sum())
```

```
Function is null: 0
SaleType is null: 0
```

In [157]:
```
Others_cat_Feat=["FireplaceQu",
"PoolQC",
"Fence",
"MiscFeature"]
```

In [158]:
```
for fa in Others_cat_Feat:
    print(f"value count of {fa}:{cds[feature].value_counts()}") ## I will impute NA""
```

```
1004.0     4
951.0      4
923.0      4
1107.0     4
900.0      4
1228.0     4
1058.0     4
1128.0     4
1122.0     4
916.0      4
1568.0     4
1204.0     4
1838.0     4
1051.0     4
707.0      4
528.0      4
1088.0     4
1350.0     4
1686.0     4
1055.0     4
```

In [159]:
```
FireQ="NA"   ## I added "NA" constant
cds_mvi["FireplaceQu"].replace(np.nan,FireQ,inplace=True)
cds_mvi["FireplaceQu"].isnull().sum()
```

Out[159]: 0

In [160]:
```
PoolQ="NA"   ## I added "NA" constant
cds_mvi["PoolQC"].replace(np.nan,PoolQ,inplace=True)
cds_mvi["PoolQC"].isnull().sum()
```

Out[160]: 0

In [161]:
```
fence="NA"   ## I added "NA" constant
cds_mvi["Fence"].replace(np.nan,fence,inplace=True)
cds_mvi["Fence"].isnull().sum()
```

Out[161]: 0

In [162]:
```
miscF="NA"   ## I added "NA" constant
cds_mvi["MiscFeature"].replace(np.nan,miscF,inplace=True)
cds_mvi["MiscFeature"].isnull().sum()
```

Out[162]: 0

Handling Garage Features:

In [163]:
```
########Numerical_garage_feature  ###############
GarageYrBlt      5.447071--0
GarageCars       0.034258--0
GarageArea       0.034258--0

#Catagorical_garage_feature
GarageType       5.378554 NA
GarageFinish     5.447071 NA
GarageQual       5.447071 NA
GarageCond       5.447071 NA

############ Just for Refrence######
```

```
  Input In [163]
    GarageYrBlt      5.447071--0
                     ^
SyntaxError: invalid syntax
```

In [164]:
```
Numerical_garage_feature=["GarageYrBlt","GarageCars","GarageArea"]
Catagorical_garage_feature=["GarageType","GarageFinish","GarageQual","GarageCond"]
```

In [165]:
```
garage_num_constant=0
for feature in Numerical_garage_feature:
    cds_mvi[feature].replace(np.nan,garage_num_constant,inplace=True)
```

In [166]:
```
cds_mvi[Numerical_garage_feature].isnull().sum()# To Check , is there any missing value available-Crosscheck
```

Out[166]:
```
GarageYrBlt    0
GarageCars     0
GarageArea     0
dtype: int64
```

In [167]:
```
garage_constant="NA"
for feature in Catagorical_garage_feature:
    cds_mvi[feature].replace(np.nan,garage_constant,inplace=True)
```

In [168]:
```
cds_mvi[Catagorical_garage_feature].isnull().sum()# To Check , is there any missing value available-Crosscheck
```

Out[168]:
```
GarageType      0
GarageFinish    0
GarageQual      0
GarageCond      0
dtype: int64
```

In [169]:
```
#Now I have computed all the miising values:
#Lets cross validate, if any missing value left in dataset(cds_mvi)
```

In [170]:
```
cds_mvi.isnull().any(axis=1).sum()
```

Out[170]:
```
1459
```

In [171]:
```
#This 1459 is the Sales Price , Target variable not missing values.
```

In [172]:
```
cds.head(20) # Display Top 20 entries before value imputation on conatinated dataaset(cds)
```

Out[172]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condit
Id													
1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	N
2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker	F
3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	N
4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	N
5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	N
6	50	RL	85.0	14115	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Mitchel	N
7	20	RL	75.0	10084	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Somerst	N
8	60	RL	NaN	10382	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	NWAmes	F
9	50	RM	51.0	6120	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	OldTown	A
10	190	RL	50.0	7420	Pave	NaN	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	A
11	20	RL	70.0	11200	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Sawyer	N
12	60	RL	85.0	11924	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	NridgHt	N
13	20	RL	NaN	12968	Pave	NaN	IR2	Lvl	AllPub	Inside	Gtl	Sawyer	N
14	20	RL	91.0	10652	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	N
15	20	RL	NaN	10920	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	NAmes	N
16	45	RM	51.0	6120	Pave	NaN	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	N
17	20	RL	NaN	11241	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	NAmes	N
18	90	RL	72.0	10791	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Sawyer	N
19	20	RL	66.0	13695	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	SawyerW	R
20	20	RL	70.0	7560	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAmes	N

In [173]:

```
cds_mvi.head(20)
# Display Top 20 entries after value imputation, Here I converted some  categorical feature NaN to NA and some numeri
#I imputed mode/mean/constant("0") values after analyzing dataset description, based on my best knowledge.
```

Out[173]:

Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condit
1	60	RL	65.0	8450	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	N
2	20	RL	80.0	9600	Pave	NA	Reg	Lvl	AllPub	FR2	Gtl	Veenker	F
3	60	RL	68.0	11250	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	N
4	70	RL	60.0	9550	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	N
5	60	RL	84.0	14260	Pave	NA	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	N
6	50	RL	85.0	14115	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	Mitchel	N
7	20	RL	75.0	10084	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Somerst	N
8	60	RL	68.0	10382	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	NWAmes	F
9	50	RM	51.0	6120	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	OldTown	A
10	190	RL	50.0	7420	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	A
11	20	RL	70.0	11200	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Sawyer	N
12	60	RL	85.0	11924	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	NridgHt	N
13	20	RL	68.0	12968	Pave	NA	IR2	Lvl	AllPub	Inside	Gtl	Sawyer	N
14	20	RL	91.0	10652	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	N
15	20	RL	68.0	10920	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	NAmes	N
16	45	RM	51.0	6120	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	N
17	20	RL	68.0	11241	Pave	NA	IR1	Lvl	AllPub	CulDSac	Gtl	NAmes	N
18	90	RL	72.0	10791	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Sawyer	N
19	20	RL	66.0	13695	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	SawyerW	R
20	20	RL	70.0	7560	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	NAmes	N

Features Conversion/Transformation

In [174]:
```
cds.columns # Display all the features/Columns;Here cds is our concatinated dataset of (Test and Train)
```

Out[174]:
```
Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street', 'Alley',
       'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
       'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
       'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle',
       'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea',
       'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC',
       'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType', 'GarageYrBlt',
       'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual', 'GarageCond',
       'PavedDrive', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVal',
       'MoSold', 'YrSold', 'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')
```

In [175]:
```
# After carefully understanding of Datadesscrption file of dataset, I understand that there are
#some variable/feature that are present as numerical but after further reading, those are categorical in nature.
# So next I am going to select those variables and convert it into string.
```

In [176]:
```
features_dtype_convert = ["MSSubClass","YearBuilt","YearRemodAdd","GarageYrBlt","YrSold","MoSold"]# Case senstive
for feature in features_dtype_convert:
    print(f"{feature}:Data Type : {cds_mvi[feature].dtype}")
```

```
MSSubClass:Data Type : int64
YearBuilt:Data Type : int64
YearRemodAdd:Data Type : int64
GarageYrBlt:Data Type : float64
YrSold:Data Type : int64
MoSold:Data Type : int64
```

In [177]:
```
cds_mvi["MoSold"].unique() # Here, MoSold=Month of Sold: It was innumerical but nature wise it is categorical.
```

Out[177]:
```
array([ 2,  5,  9, 12, 10,  8, 11,  4,  1,  7,  3,  6], dtype=int64)
```

In [178]:
```
cds_mvi[features_dtype_convert].head()  # Display top categories,
#I want to concentrate now on MoSold, which is Temporal variable.
```

Out[178]:

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

In [180]:
```
for feature in features_dtype_convert:
    cds_mvi[feature]=cds_mvi[feature].astype(str)
```

In [181]:
```
for feature in features_dtype_convert:
    print(f"{feature}:Data Type : {cds_mvi[feature].dtype}")
```

```
MSSubClass:Data Type : object
YearBuilt:Data Type : object
YearRemodAdd:Data Type : object
GarageYrBlt:Data Type : object
YrSold:Data Type : object
MoSold:Data Type : object
```

In [182]:
```
#As we can see above, I have converted all the features into object.
```

Categorical Feature to Numerical Feature Transformation:

In [183]:
```
#As we know, our model only train onto numerica; features, so Iwe need to convert categorical features into numerical
```

In [184]:
```
# There is two popular techniques are an 1: Ordinal Encoding
# 2: One-Hot Encoding for categorical data trasformation.
```

Ordinal encoding for categorical data:

In [185]:
```
# Our Categorical data contain diffrent label values rather than numeric values.# I will choose Ordinal encoding
#Techniques in which I will do ranking of diffrent values. example: for poor: 1 and Exceelent:5...
```

In [186]:
```
Ord_enco_feat=["ExterQual","ExterCond","Functional","GarageCond","GarageQual","GarageFinish","PoolQC",
               "BsmtFinType1","BsmtFinType2","BsmtExposure","BsmtQual","BsmtCond",
               "FireplaceQu","KitchenQual","HeatingQC","PavedDrive","Utilities"]
print("Total No. of features to convert Ordinal_Numerical format:", len(Ord_enco_feat))
```

```
Total No. of features to convert Ordinal_Numerical format: 17
```

Convert every Categorical Features into Numerical one-by-one

In [187]:
```
cds_mvi["ExterQual"].value_counts()
```

Out[187]:
```
TA    1798
Gd     979
Ex     107
Fa      35
Name: ExterQual, dtype: int64
```

In [188]:
```
cds_mvi["ExterQual"].unique()
```

Out[188]:
```
array(['Gd', 'TA', 'Ex', 'Fa'], dtype=object)
```

In [189]:
```
cds_mvi["ExterQual"]=cds_mvi["ExterQual"].astype(CategoricalDtype(categories=
                                                                   ["Po", "Fa", "TA", "Gd", "Ex"], ordered = True)).cat
```

In [190]:
```
cds_mvi["ExterQual"].value_counts() # values has been updated from cat features to numerical features.
```

Out[190]:
```
2    1798
3     979
4     107
1      35
Name: ExterQual, dtype: int64
```

In [191]:
```
cds_mvi["ExterCond"].value_counts()
```

Out[191]:
```
TA    2538
Gd     299
Fa      67
Ex      12
Po       3
Name: ExterCond, dtype: int64
```

In [192]:
```
cds_mvi["ExterCond"].unique()
```

Out[192]:
```
array(['TA', 'Gd', 'Fa', 'Po', 'Ex'], dtype=object)
```

In [193]:
```
cds_mvi["ExterCond"]=cds_mvi["ExterCond"].astype(CategoricalDtype(categories=
                                                                  ["TA", "Gd", "Fa", "Po", "Ex"], ordered = True)).cat
```

In [194]:
```
cds_mvi["ExterCond"].value_counts()
```

Out[194]:
```
0    2538
1     299
2      67
4      12
3       3
Name: ExterCond, dtype: int64
```

In [195]:
```
cds_mvi["Functional"].value_counts()
```

Out[195]:
```
Typ     2719
Min2      70
Min1      65
Mod       35
Maj1      19
Maj2       9
Sev        2
Name: Functional, dtype: int64
```

In [196]:
```
cds_mvi["Functional"].unique()
```

Out[196]:
```
array(['Typ', 'Min1', 'Maj1', 'Min2', 'Mod', 'Maj2', 'Sev'], dtype=object)
```

In [197]:
```
cds_mvi["Functional"]=cds_mvi["Functional"].astype(CategoricalDtype(categories=
                                                                  ['Typ', 'Min1', 'Maj1', 'Min2', 'Mod', 'Maj2', 'Sev']
```

In [198]:
```
cds_mvi["Functional"].value_counts()
```

Out[198]:
```
0    2719
3      70
1      65
4      35
2      19
5       9
6       2
Name: Functional, dtype: int64
```

In [199]:
```
# Lets compute remaining codes alltogether.
```

In [200]:
```
#cds_mvi["Utilities"].unique()
#cds_mvi["BsmtFinSF1"].value_counts()
#,"PavedDrive","Utilities"
```

In [201]:
```
cds_mvi["GarageCond"]=cds_mvi["GarageCond"].astype(CategoricalDtype(categories=
                                    ['TA', 'Fa', 'NA', 'Gd', 'Po', 'Ex'], ordered = True)).cat.codes
cds_mvi["GarageQual"]=cds_mvi["GarageQual"].astype(CategoricalDtype(categories=
                                    ['TA', 'Fa', 'Gd', 'NA', 'Ex', 'Po'], ordered = True)).cat.codes
cds_mvi["GarageFinish"]=cds_mvi["GarageFinish"].astype(CategoricalDtype(categories=
                                    ['RFn', 'Unf', 'Fin', 'NA'], ordered = True)).cat.codes
cds_mvi["PoolQC"]=cds_mvi["PoolQC"].astype(CategoricalDtype(categories=
                                    ['NA', 'Ex', 'Fa', 'Gd'], ordered = True)).cat.codes

cds_mvi["BsmtFinType1"]=cds_mvi["BsmtFinType1"].astype(CategoricalDtype(categories=
                                    ['NA', 'Unf', 'LwQ', 'Rec', 'BLQ','ALQ', 'GLQ'], ordered = True)
cds_mvi["BsmtFinType2"]=cds_mvi["BsmtFinType2"].astype(CategoricalDtype(categories=
                                    ['NA', 'Unf', 'LwQ', 'Rec', 'BLQ','ALQ', 'GLQ'], ordered = True)
cds_mvi["BsmtExposure"]=cds_mvi["BsmtExposure"].astype(CategoricalDtype(categories=
                                    ['NA', 'No', 'Mn', 'Av','Gd'], ordered = True)).cat.codes
cds_mvi["BsmtQual"]=cds_mvi["BsmtQual"].astype(CategoricalDtype(categories=
                                    ['NA', 'Po', 'Fa', 'TA', 'Gd', 'Ex'], ordered = True)).cat.codes

cds_mvi["BsmtCond"]=cds_mvi["BsmtCond"].astype(CategoricalDtype(categories=
                                    ['NA', 'Po', 'Fa', 'TA', 'Gd', 'Ex'], ordered = True)).cat.codes
cds_mvi["FireplaceQu"]=cds_mvi["FireplaceQu"].astype(CategoricalDtype(categories=
                                    ['NA', 'Po', 'Fa', 'TA', 'Gd', 'Ex'], ordered = True)).cat.codes
cds_mvi["KitchenQual"]=cds_mvi["KitchenQual"].astype(CategoricalDtype(categories=
                                    ['Po', 'Fa', 'TA', 'Gd', 'Ex'], ordered = True)).cat.codes
cds_mvi["HeatingQC"]=cds_mvi["HeatingQC"].astype(CategoricalDtype(categories=
                                    ['Po', 'Fa', 'TA', 'Gd', 'Ex'], ordered = True)).cat.codes
cds_mvi["PavedDrive"]=cds_mvi["PavedDrive"].astype(CategoricalDtype(categories=
                                    ['N', 'P', 'Y'], ordered = True)).cat.codes
cds_mvi["Utilities"]=cds_mvi["Utilities"].astype(CategoricalDtype(categories=
                                    ['ELO', 'NASeWa', 'NASeWr', 'AllPub'], ordered = True)).cat.code
```

In [202]:
```
cds_mvi.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2919 entries, 1 to 2919
Data columns (total 80 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   MSSubClass     2919 non-null   object 
 1   MSZoning       2919 non-null   object 
 2   LotFrontage    2919 non-null   float64
 3   LotArea        2919 non-null   int64  
 4   Street         2919 non-null   object 
 5   Alley          2919 non-null   object 
 6   LotShape       2919 non-null   object 
 7   LandContour    2919 non-null   object 
 8   Utilities      2919 non-null   int8   
 9   LotConfig      2919 non-null   object 
 10  LandSlope      2919 non-null   object 
 11  Neighborhood   2919 non-null   object 
 12  Condition1     2919 non-null   object 
 13  Condition2     2919 non-null   object 
 14  BldgType       2919 non-null   object 
 15  HouseStyle     2919 non-null   object 
 16  OverallQual    2919 non-null   int64  
 17  OverallCond    2919 non-null   int64  
 18  YearBuilt      2919 non-null   object 
 19  YearRemodAdd   2919 non-null   object 
 20  RoofStyle      2919 non-null   object 
 21  RoofMatl       2919 non-null   object 
 22  Exterior1st    2919 non-null   object 
 23  Exterior2nd    2919 non-null   object 
 24  MasVnrType     2919 non-null   object 
 25  MasVnrArea     2919 non-null   float64
 26  ExterQual      2919 non-null   int8   
 27  ExterCond      2919 non-null   int8   
 28  Foundation     2919 non-null   object 
 29  BsmtQual       2919 non-null   int8   
 30  BsmtCond       2919 non-null   int8   
 31  BsmtExposure   2919 non-null   int8   
 32  BsmtFinType1   2919 non-null   int8   
 33  BsmtFinSF1     2919 non-null   float64
 34  BsmtFinType2   2919 non-null   int8   
 35  BsmtFinSF2     2919 non-null   float64
 36  BsmtUnfSF      2919 non-null   float64
 37  TotalBsmtSF    2919 non-null   float64
 38  Heating        2919 non-null   object 
```

```
 39  HeatingQC      2919 non-null   int8
 40  CentralAir     2919 non-null   object
 41  Electrical     2919 non-null   object
 42  1stFlrSF       2919 non-null   int64
 43  2ndFlrSF       2919 non-null   int64
 44  LowQualFinSF   2919 non-null   int64
 45  GrLivArea      2919 non-null   int64
 46  BsmtFullBath   2919 non-null   float64
 47  BsmtHalfBath   2919 non-null   float64
 48  FullBath       2919 non-null   int64
 49  HalfBath       2919 non-null   int64
 50  BedroomAbvGr   2919 non-null   int64
 51  KitchenAbvGr   2919 non-null   int64
 52  KitchenQual    2919 non-null   int8
 53  TotRmsAbvGrd   2919 non-null   int64
 54  Functional     2919 non-null   int8
 55  Fireplaces     2919 non-null   int64
 56  FireplaceQu    2919 non-null   int8
 57  GarageType     2919 non-null   object
 58  GarageYrBlt    2919 non-null   object
 59  GarageFinish   2919 non-null   int8
 60  GarageCars     2919 non-null   float64
 61  GarageArea     2919 non-null   float64
 62  GarageQual     2919 non-null   int8
 63  GarageCond     2919 non-null   int8
 64  PavedDrive     2919 non-null   int8
 65  WoodDeckSF     2919 non-null   int64
 66  OpenPorchSF    2919 non-null   int64
 67  EnclosedPorch  2919 non-null   int64
 68  3SsnPorch      2919 non-null   int64
 69  ScreenPorch    2919 non-null   int64
 70  PoolArea       2919 non-null   int64
 71  PoolQC         2919 non-null   int8
 72  Fence          2919 non-null   object
 73  MiscFeature    2919 non-null   object
 74  MiscVal        2919 non-null   int64
 75  MoSold         2919 non-null   object
 76  YrSold         2919 non-null   object
 77  SaleType       2919 non-null   object
 78  SaleCondition  2919 non-null   object
 79  SalePrice      1460 non-null   float64
dtypes: float64(11), int64(20), int8(17), object(32)
memory usage: 1.5+ MB
```

In [203]:
```
cds_mvi["SaleCondition"].value_counts() #Example 1: Here sale condition is integer type but info() still showing it 
# so we need to convert these kind to numerical-one-by-one.
```

Out[203]:
```
Normal     2402
Partial     245
Abnorml     190
Family       46
Alloca       24
AdjLand      12
Name: SaleCondition, dtype: int64
```

In [204]:
```
cds_mvi["MoSold"].value_counts()
#Example 2: Here MpSold is integer type but info() still showing it is object
# so we need to convert these kind to numerical-one-by-one.
```

Out[204]:
```
6     503
7     446
5     394
4     279
8     233
3     232
10    173
9     158
11    142
2     133
1     122
12    104
Name: MoSold, dtype: int64
```

In [205]:
```
# Categorical features contains label order rather than  the numerical values.
# Now we have left with nearly 30 varibles which has to be converted into numerical data and all of them are nominal
# For that purpose, technique best suited is One-Hot Encoding which creates dummy variables and it is
#used where where order does not matter, will not give to priority to categories, qill be consider only 0 and 1..
```

One-Hot Encoding for categorical data:

In [206]:
```
cds_encode=cds_mvi.copy() # Make copy of cds_mvi data set and assign it to new varible"cds_encode"
```

In [207]:
```
object_encod_feature=cds_encode.select_dtypes(include="object").columns.tolist()
# tolist()convert pandas dataFrame(cds_encode) columns to list.

print("Total Objectdatatypes features:-",len(object_encod_feature))
print("List of Features:\n",object_encod_feature)
```

```
Total Objectdatatypes features:- 32
List of Features:
 ['MSSubClass', 'MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'LotConfig', 'LandSlope', 'Neighborhood',
'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'YearBuilt', 'YearRemodAdd', 'RoofStyle', 'RoofMatl', 'Exteri
or1st', 'Exterior2nd', 'MasVnrType', 'Foundation', 'Heating', 'CentralAir', 'Electrical', 'GarageType', 'GarageYrBl
t', 'Fence', 'MiscFeature', 'MoSold', 'YrSold', 'SaleType', 'SaleCondition']
```

In [209]:
```
#*********************Working***************************
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: