Import Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import calendar
from pandas.api.types import CategoricalDtype
from sklearn.preprocessing import StandardScaler
```

Data Loading

In [ ]:
```
train_data_path=r"train.csv"
test_data_path=r"test.csv"

df_train=pd.read_csv(train_data_path)
df_test=pd.read_csv(test_data_path)

print(df_train.shape)
print(df_test.shape)
```

```
(1460, 81)
(1459, 80)
```

Data Analysis

In [3]:
```
pd.set_option('display.max_columns',None) #to display all columns
pd.set_option('display.max_rows',None) #to display all rows
```

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

Out[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	Lan
0	1	60	RL	65.0	8450	Pave	NaN	Reg	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	

In [5]:
```
df_test.head()
```

Out[5]:

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

data integration

In [6]:
```
df=pd.concat((df_train,df_test))
temp_df=df
print(df.shape)
```

```
(2919, 81)
```

In [7]:
```
df.head()
```

Out[7]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	Lan
0	1	60	RL	65.0	8450	Pave	NaN	Reg	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	

In [8]:
```
df.tail()
```

Out[8]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShap
1454	2915	160	RM	21.0	1936	Pave	NaN	Re
1455	2916	160	RM	21.0	1894	Pave	NaN	Re
1456	2917	20	RL	160.0	20000	Pave	NaN	Re
1457	2918	85	RL	62.0	10441	Pave	NaN	Re
1458	2919	60	RL	74.0	9627	Pave	NaN	Re

Exploratory data analysis

In [9]:
```
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
```

In [10]:

```
df.describe()
```

Out[10]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	Ove
count	2919.000000	2919.000000	2433.000000	2919.000000	2919.000000	2919
mean	1460.000000	57.137718	69.305795	10168.114080	6.089072	5
std	842.787043	42.517628	23.344905	7886.996359	1.409947	
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1
25%	730.500000	20.000000	59.000000	7478.000000	5.000000	5
50%	1460.000000	50.000000	68.000000	9453.000000	6.000000	5
75%	2189.500000	70.000000	80.000000	11570.000000	7.000000	6
max	2919.000000	190.000000	313.000000	215245.000000	10.000000	9

In [11]:
```
int_feature=df.select_dtypes(include=['int64']).columns
```

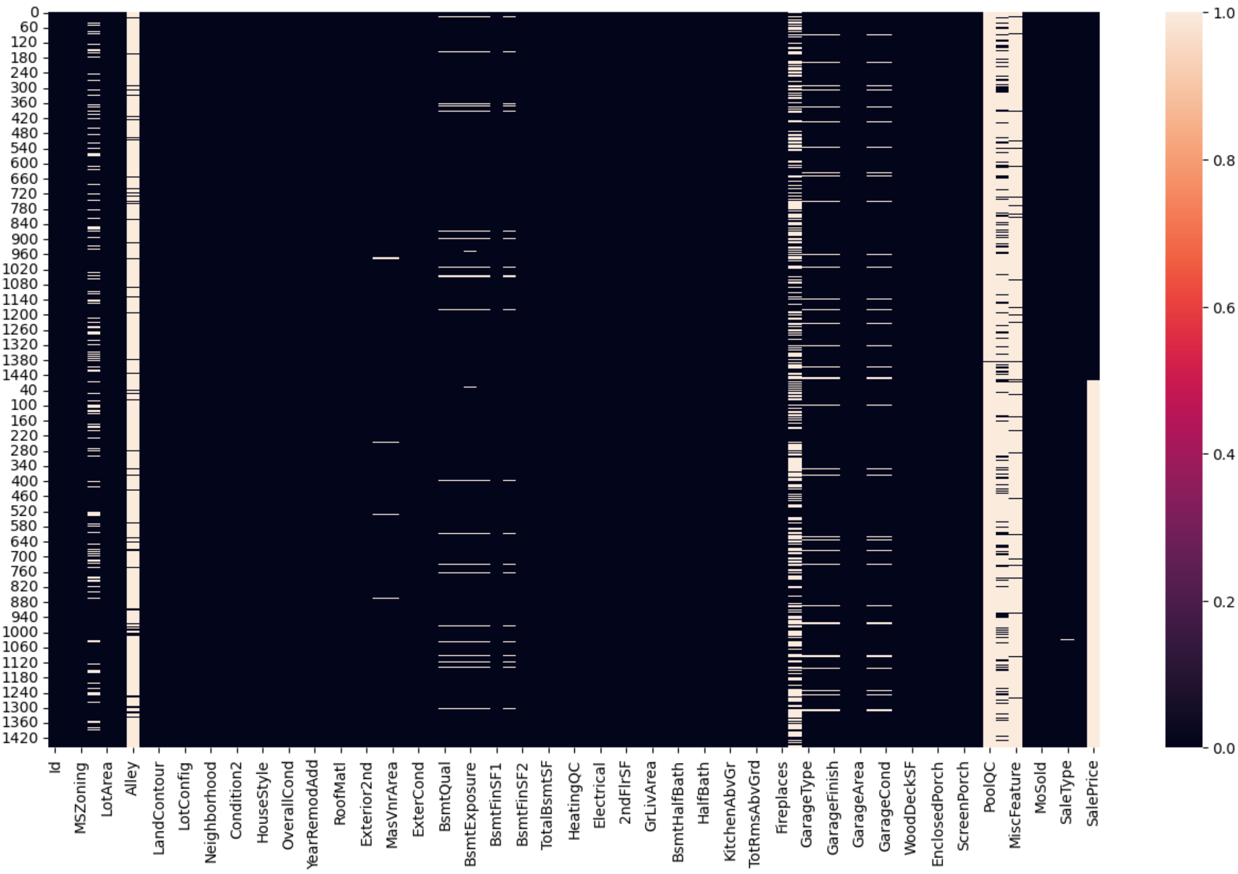
In [12]:
```
float_feature=df.select_dtypes(include=['float64']).columns
```

In [13]:
```
cat_feature=df.select_dtypes(include=['object']).columns
```

visualing missing value

In [14]:
```
plt.figure(figsize=(16,9))
sns.heatmap(df.isnull())
```

Out[14]: <AxesSubplot: >



In [15]:
```
# set index as is column
df=df.set_index("Id")
```

get the null value percentage for every feature

In [16]:
```
null_count=df.isnull().sum()
null_count
```

Out[16]:
```
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 [17]:

```python
null_percent=df.isnull().sum()/df.shape[0]*100
null_percent
```

Out[17]:

```
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
```

drop column/features

In [18]:
```
""" as per domain knowldge we will not drop this featurre rather we add s
miss_value_50_perc=null_percent[null_percent>50]
miss_value_50_perc
```

Out[18]:
```
Alley          93.216855
PoolQC         99.657417
Fence          80.438506
MiscFeature    96.402878
dtype: float64
```

In [19]:
```
""" as per domain knowldge we will not drop this featurre rather we add s
miss_value_20_50_perc=null_percent[(null_percent>20)& (null_percent<51)]
miss_value_20_50_perc
```

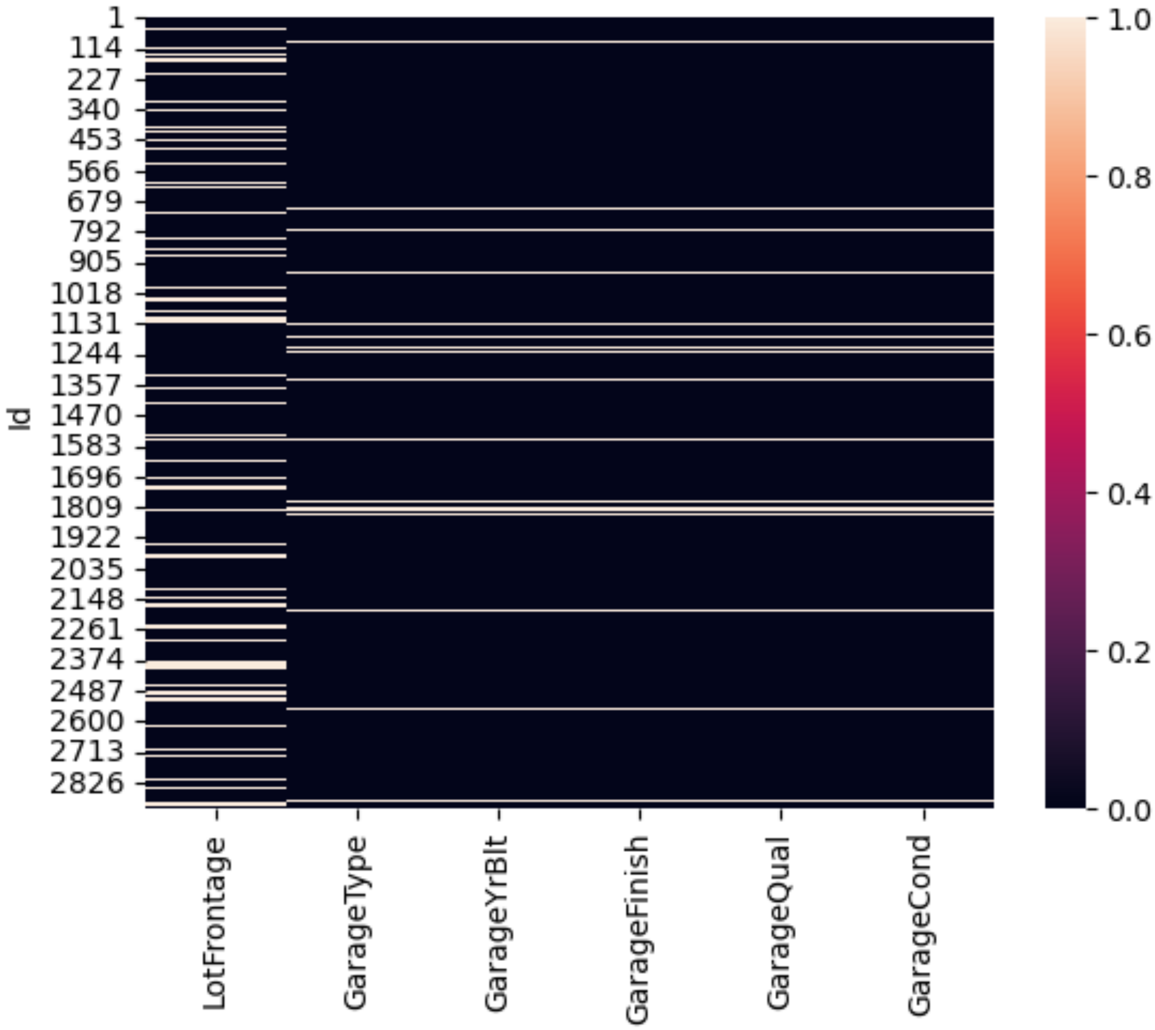
Out[19]:
```
FireplaceQu    48.646797
SalePrice      49.982871
dtype: float64
```

In [20]:
```
miss_value_5_20_perc=null_percent[(null_percent>5)& (null_percent<21)]
miss_value_5_20_perc
```

Out[20]:
```
LotFrontage     16.649538
GarageType       5.378554
GarageYrBlt      5.447071
GarageFinish     5.447071
GarageQual       5.447071
GarageCond       5.447071
dtype: float64
```

In [21]:
```
sns.heatmap(df[miss_value_5_20_perc.keys()].isnull())
```

Out[21]:
```
<AxesSubplot: ylabel='Id'>
```



In [22]:
```
## as per observation we will not drop any feature from data set
```

missing value imputation

In [23]:
```
missing_value_feat=null_percent[null_percent>0]
print("Total missing value feature=",len(missing_value_feat))
missing_value_feat
```

```
Total missing value feature= 35
```

Out[23]:
```
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 [24]:
```python
cat_na_feat=missing_value_feat[missing_value_feat.keys().isin(cat_feature
print("total number of categorical missing feature",len(cat_na_feat))
cat_na_feat
```

```
total number of categorical missing feature 23
```

Out[24]:
```
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 [25]:
```
int_na_feat=missing_value_feat[missing_value_feat.keys().isin(int_feature
print("total number of int missing feature",len(int_na_feat))
int_na_feat
```

```
total number of int missing feature 0
```

Out[25]:
```
Series([], dtype: float64)
```

In [26]:
```
float_na_feat=missing_value_feat[missing_value_feat.keys().isin(float_fea
print("total number of float missing feature",len(float_na_feat))
float_na_feat
```

```
total number of float missing feature 12
```

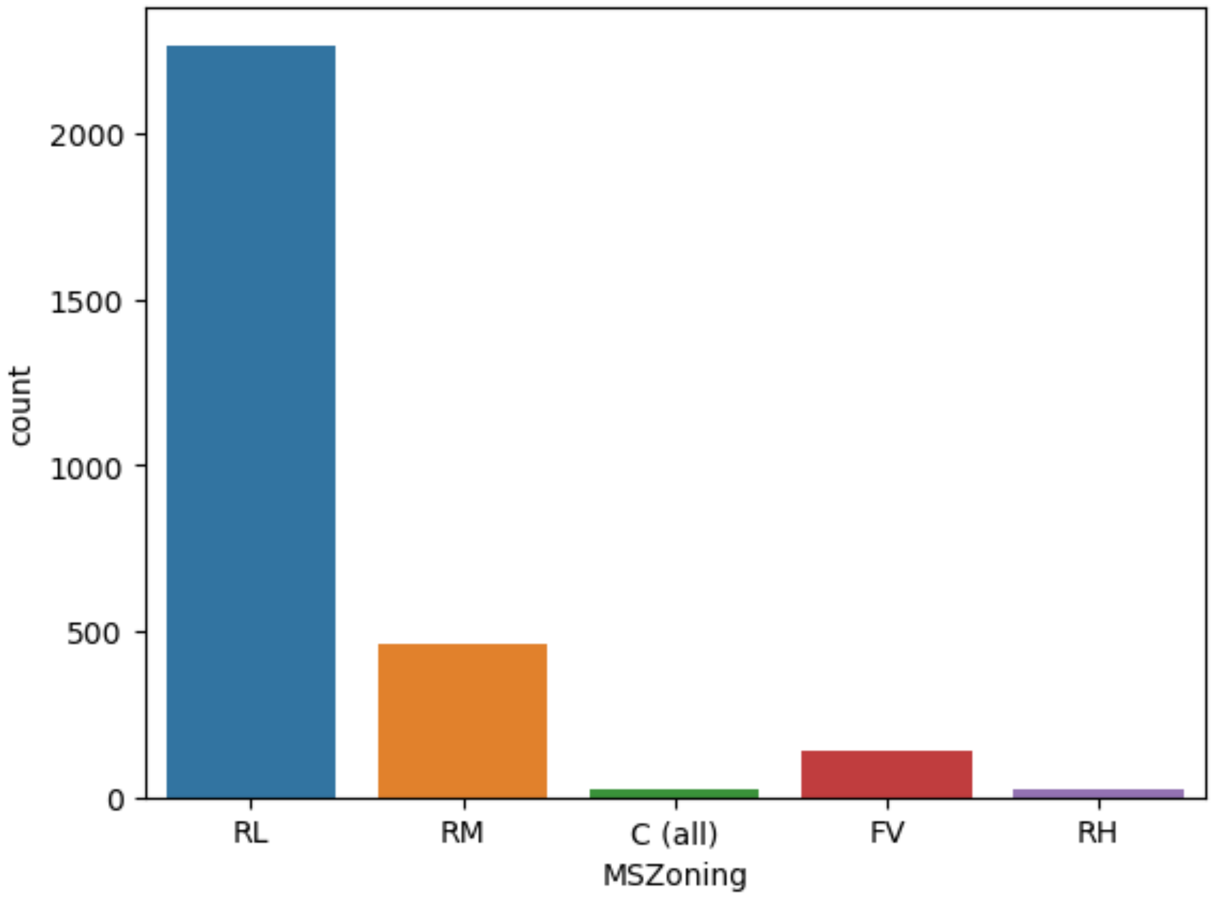
Out[26]:
```
LotFrontage      16.649538
MasVnrArea        0.787941
BsmtFinSF1        0.034258
BsmtFinSF2        0.034258
BsmtUnfSF         0.034258
TotalBsmtSF       0.034258
BsmtFullBath      0.068517
BsmtHalfBath      0.068517
GarageYrBlt       5.447071
GarageCars        0.034258
GarageArea        0.034258
SalePrice        49.982871
dtype: float64
```

In [27]:
```
## funtion to visualize data feature before and after imputation of missi
def plot_data(df, df_new, feature):
```

```
    plt.subplot(121)
    sns.countplot(x=feature, data=df)
    plt.title("Before Imputation")
    plt.subplot(122)
    sns.countplot(x=feature, data=df_new)
    plt.title("After Imputation")
    plt.show()
```

In [28]:
```
### handling MSZoning=0.137033

df["MSZoning"].value_counts()

# count plot in graph form
sns.countplot(x=df["MSZoning"])
```

Out[28]: <AxesSubplot: xlabel='MSZoning', ylabel='count'>



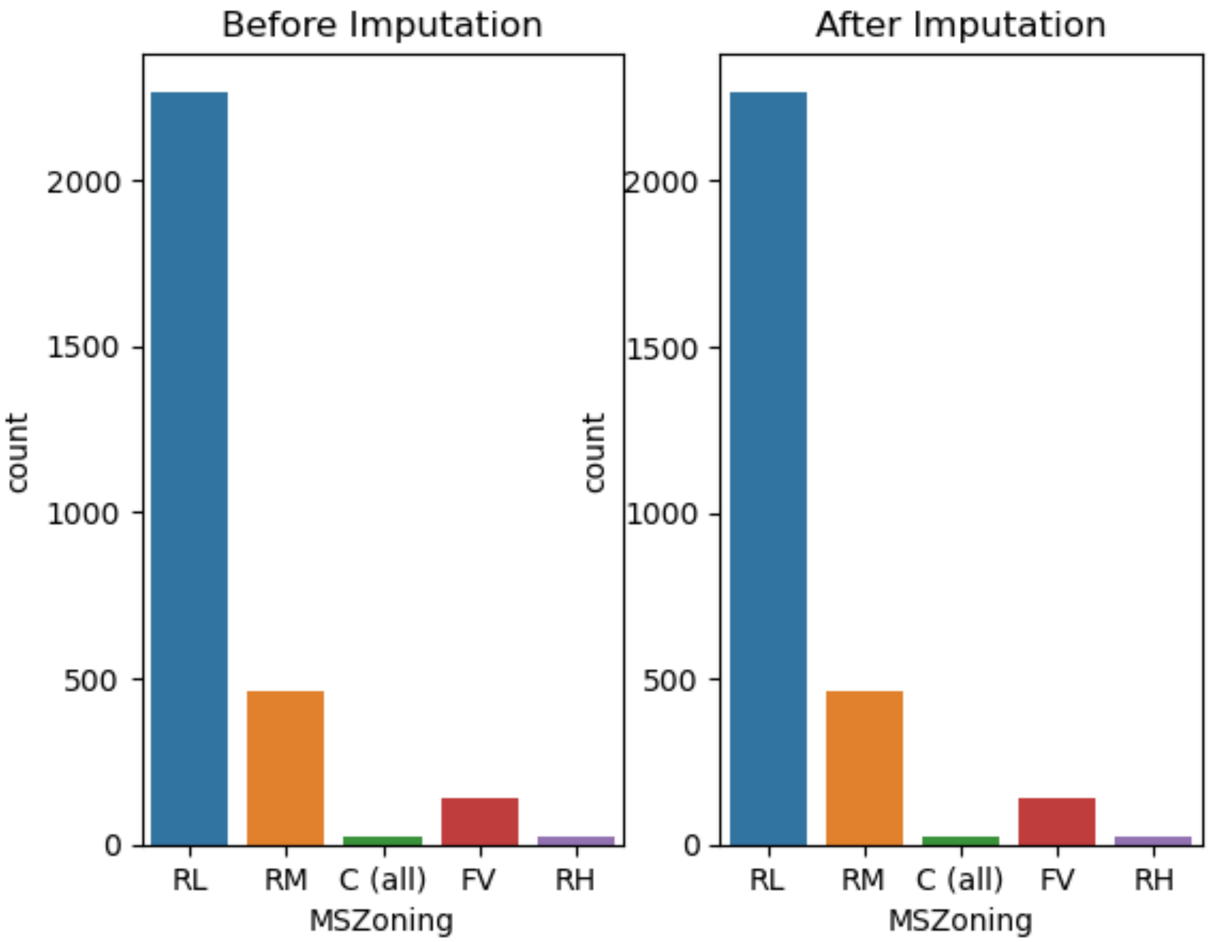
In [29]:
```
## backing up origianl data frame
df_mvi=df.copy()
```

In [30]:
```
# as we can see here RL is the mode for this feature
mszoning_mode=df["MSZoning"].mode()[0]
mszoning_mode
df_mvi["MSZoning"].replace(np.nan,mszoning_mode,inplace=True)
# now chceck do we have any missinf vLUE
df_mvi["MSZoning"].isnull().sum()
```

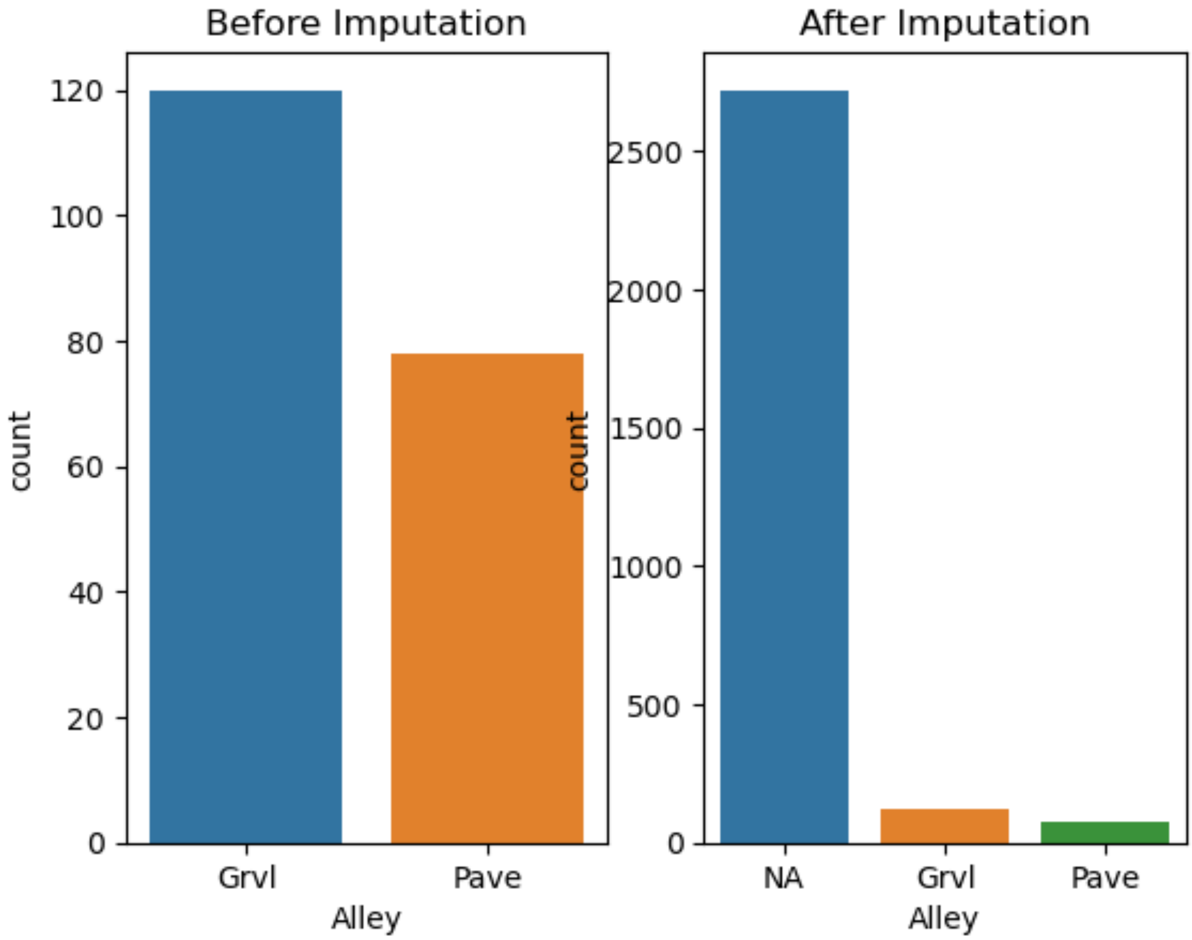
Out[30]: 0

In [ ]:

In [31]:
```
#compare before and after imputation
feature="MSZoning"
plot_data(df,df_mvi,feature)
```

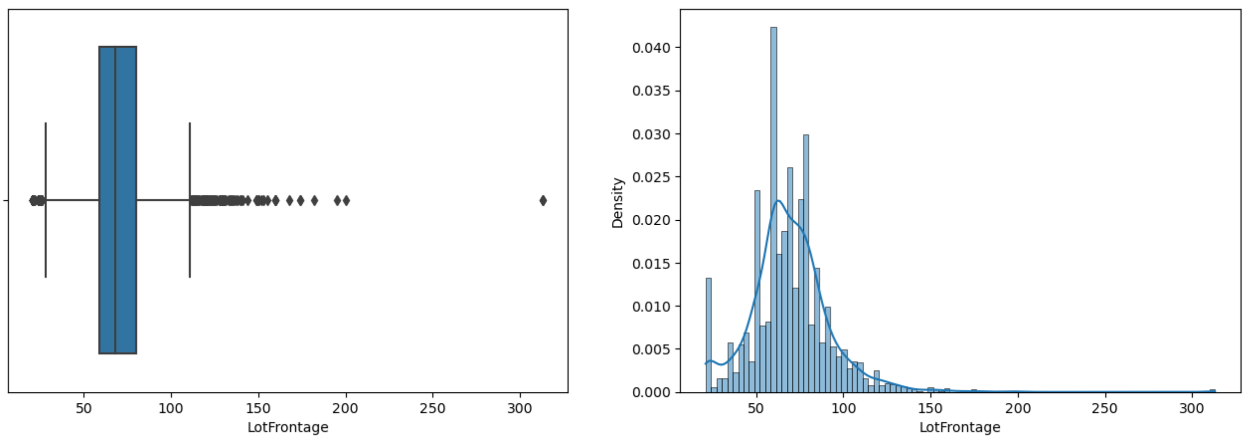


In [32]:
```
## handleing alley = 93.216855

df_mvi["Alley"].value_counts()
alley_cont="NA"
df_mvi["Alley"].replace(np.nan,alley_cont,inplace=True) # replace missing
df_mvi["Alley"].isnull().sum()

# compare before and after imputation
plot_data(df,df_mvi,"Alley")
```



In [33]:
```
#LotFrontage=16.649538
def boxHistPlot(df,feature, figsize=(16,5)):
    plt.figure(figsize=figsize)
    plt.subplot(121)
    sns.boxplot(x=feature, data=df)
    plt.subplot(122)
    sns.histplot(x=feature,data=df,stat="density", kde=True)
    plt.show()
```

In [34]:
```
boxHistPlot(df,"LotFrontage")
```

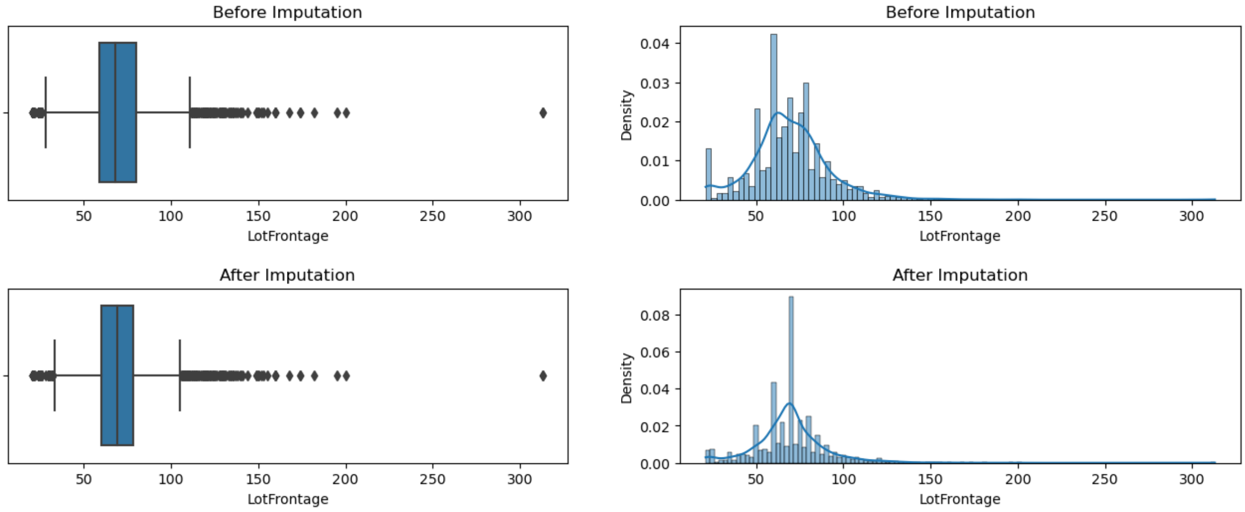


In [35]:
```
lotfrontage_mean=df["LotFrontage"].mean()
```

```
# lotfrontage_mean
df_mvi["LotFrontage"].replace(np.nan,lotfrontage_mean,inplace=True)
df_mvi["LotFrontage"].isnull().sum()
```

Out[35]: 0

In [36]:
```
# compare old and new box hist plot after imputation
def oldNewBoxHistPlot(df,df_new,feature, figsize=(16,5)):
    plt.figure(figsize=figsize)
    plt.subplot(221)
    sns.boxplot(x=feature, data=df)
    plt.title("Before Imputation")
    plt.subplot(222)
    sns.histplot(x=feature,data=df,stat="density", kde=True)
    plt.title("Before Imputation")
    plt.figure(figsize=figsize)
    plt.subplot(223)
    sns.boxplot(x=feature, data=df_new)
    plt.title("After Imputation")
    plt.subplot(224)
    sns.histplot(x=feature,data=df_new,stat="density", kde=True)
    plt.title("After Imputation")
    plt.show()

oldNewBoxHistPlot(df,df_mvi,"LotFrontage")
```



In [37]:
```
## handling utility
df["Utilities"].value_counts()
utility_const=df["Utilities"].mode()[0]
df_mvi["Utilities"].replace(np.nan,utility_const,inplace=True)
df_mvi["Utilities"].isnull().sum()
```

Out[37]: 0

In [38]:
```
# Exterior1st      0.034258
# Exterior2nd      0.034258
# both are object type
```

```
print(df["Exterior1st"].value_counts())
print("----")
print(df["Exterior2nd"].value_counts())
```

```
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
----
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 [39]:
```
exterior_1_const=df["Exterior1st"].mode()[0]
df_mvi["Exterior1st"].replace(np.nan,exterior_1_const,inplace=True)
df_mvi["Exterior1st"].isnull().sum()
```

Out[39]: 0

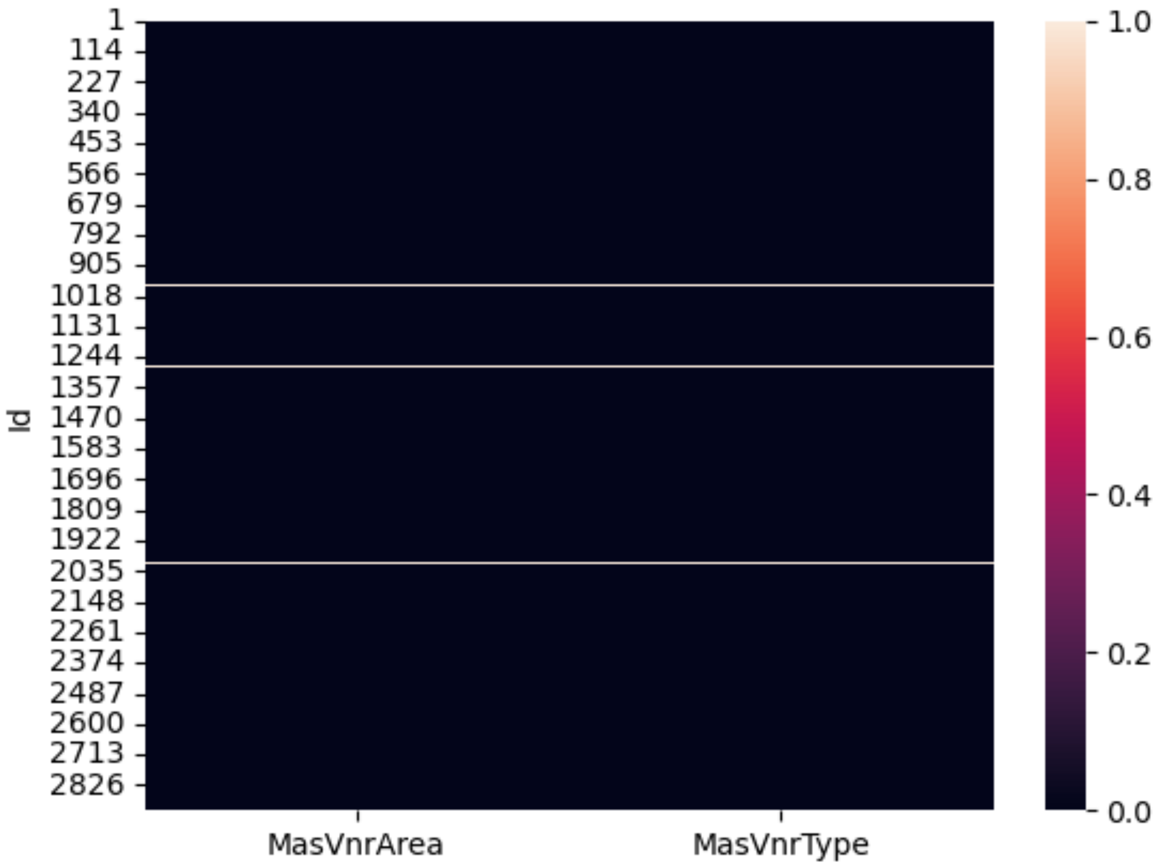
In [40]:
```
exterior_2_const=df["Exterior2nd"].mode()[0]
df_mvi["Exterior2nd"].replace(np.nan,exterior_2_const,inplace=True)
df_mvi["Exterior2nd"].isnull().sum()
```

Out[40]: 0

In [41]:
```
# MasVnrType       0.822199
# MasVnrArea       0.787941
```

```
sns.heatmap(df[["MasVnrArea", "MasVnrType"]].isnull())
```

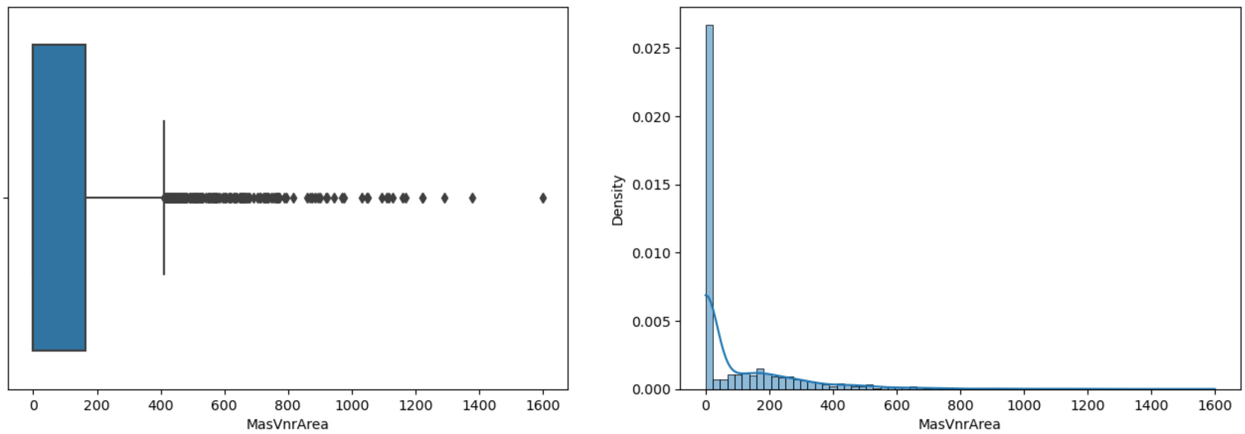
Out[41]: <AxesSubplot: ylabel='Id'>



In [42]:
```
mas_vnr_type_const=df["MasVnrType"].mode()[0]
df_mvi["MasVnrType"].replace(np.nan,mas_vnr_type_const,inplace=True)
df_mvi["MasVnrType"].isnull().sum()
```

Out[42]: 0

In [43]:
```
boxHistPlot(df,"MasVnrArea")
```



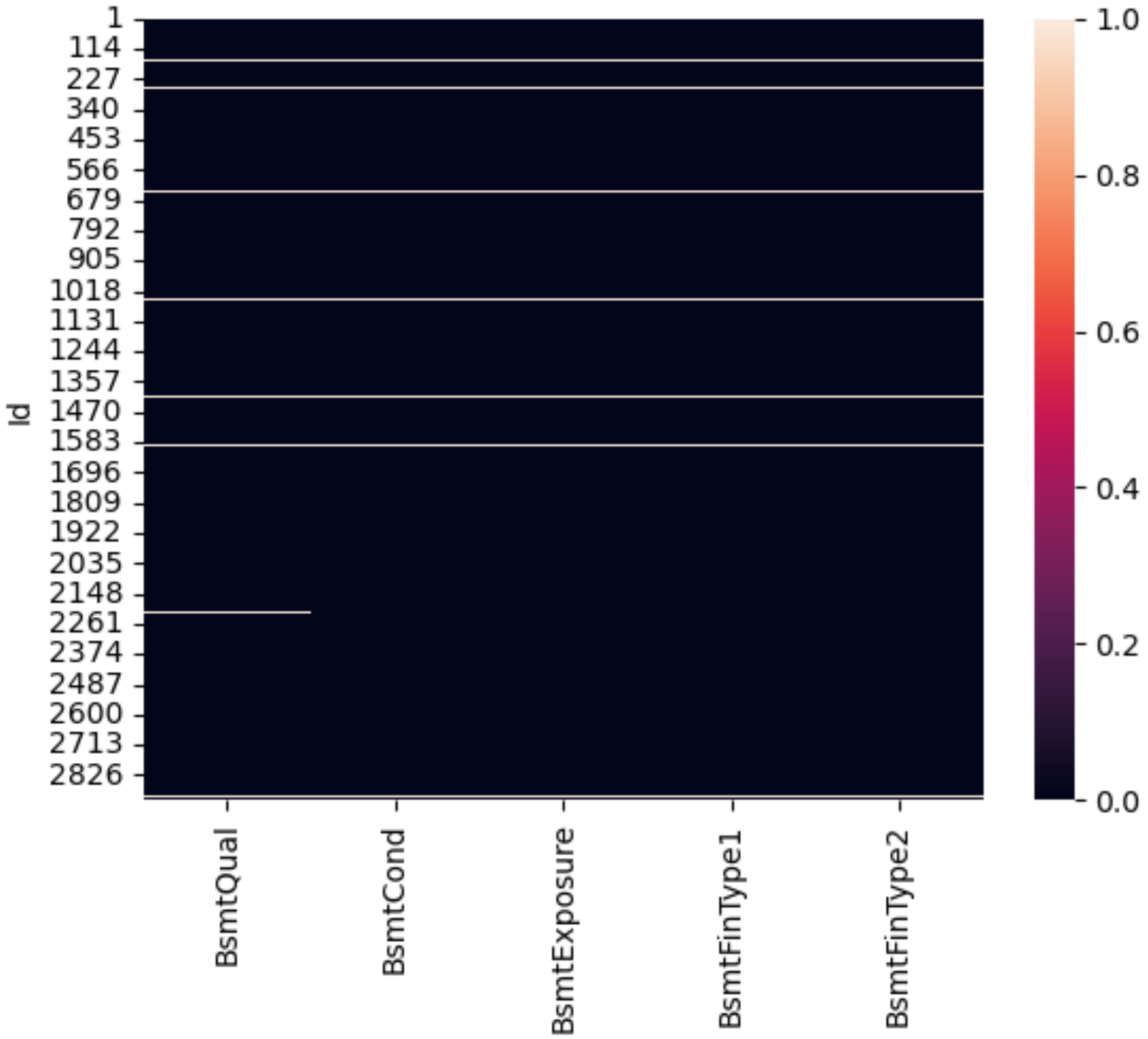
In [44]:
```
mas_vnr_area_const=0# as we can see the mode is 0 in above plots
df_mvi["MasVnrArea"].replace(np.nan,mas_vnr_area_const,inplace=True)
```

```
df_mvi["MasVnrArea"].isnull().sum()
```

Out[44]: 0

In [45]:
```
### handleing basement
## categarical
# BsmtQual         2.774923
# BsmtCond         2.809181
# BsmtExposure     2.809181
# BsmtFinType1     2.706406
# BsmtFinType2     2.740665

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


cat_bsmt_feat=["BsmtQual","BsmtCond","BsmtExposure","BsmtFinType1","BsmtF
num_bsmt_feat=["BsmtFinSF1","BsmtFinSF2","BsmtUnfSF","TotalBsmtSF","BsmtF

sns.heatmap(df[cat_bsmt_feat].isnull()) # check missing values in categor
for feat in cat_bsmt_feat:
    print(df[feat].value_counts())
    print("----")
```

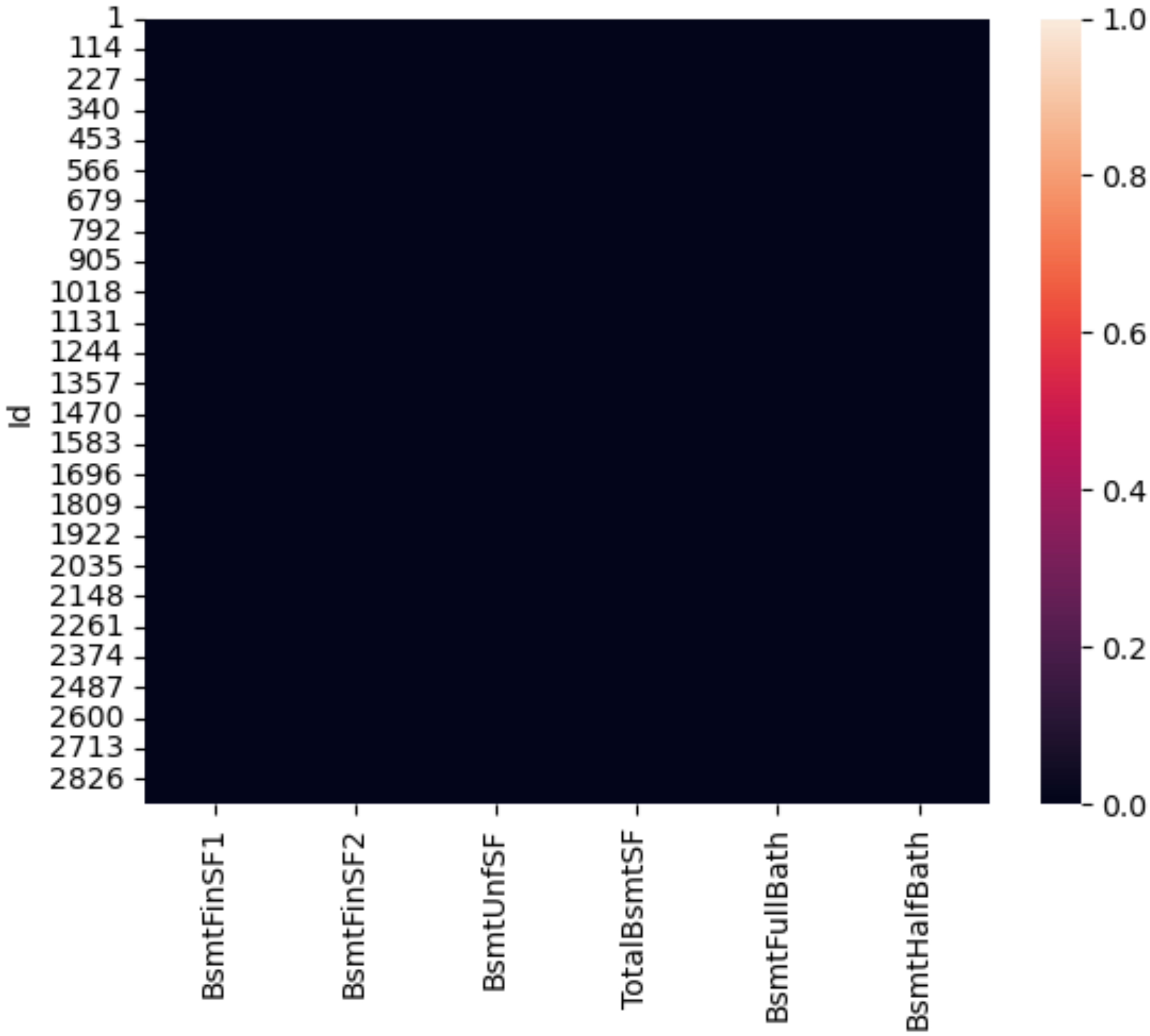
```
TA    1283
Gd    1209
Ex     258
Fa      88
Name: BsmtQual, dtype: int64
----
TA    2606
Gd     122
Fa     104
Po       5
Name: BsmtCond, dtype: int64
----
No    1904
Av     418
Gd     276
Mn     239
Name: BsmtExposure, dtype: int64
----
Unf    851
GLQ    849
ALQ    429
Rec    288
BLQ    269
LwQ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



In [46]:
```
bsmt_cont="NA"
for feat in cat_bsmt_feat:
    df_mvi[feat].replace(np.nan,bsmt_cont,inplace=True)
```

In [47]:
```
sns.heatmap(df[num_bsmt_feat].isnull()) # check missing values in numeric
```

Out[47]: <AxesSubplot: ylabel='Id'>



In [48]:
```
# analysing basement feature
df_bsmt=df[cat_bsmt_feat+num_bsmt_feat]
df_bsmt[df_bsmt.isnull().any(axis=1)]
```

Out[48]:

Id	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinType2	BsmtFir
18	NaN	NaN	NaN	NaN	NaN	
40	NaN	NaN	NaN	NaN	NaN	
91	NaN	NaN	NaN	NaN	NaN	
103	NaN	NaN	NaN	NaN	NaN	
157	NaN	NaN	NaN	NaN	NaN	
183	NaN	NaN	NaN	NaN	NaN	
260	NaN	NaN	NaN	NaN	NaN	
333	Gd	TA	No	GLQ	NaN	11
343	NaN	NaN	NaN	NaN	NaN	

363	NaN	NaN	NaN	NaN	NaN
372	NaN	NaN	NaN	NaN	NaN
393	NaN	NaN	NaN	NaN	NaN
521	NaN	NaN	NaN	NaN	NaN
533	NaN	NaN	NaN	NaN	NaN
534	NaN	NaN	NaN	NaN	NaN
554	NaN	NaN	NaN	NaN	NaN
647	NaN	NaN	NaN	NaN	NaN
706	NaN	NaN	NaN	NaN	NaN
737	NaN	NaN	NaN	NaN	NaN
750	NaN	NaN	NaN	NaN	NaN
779	NaN	NaN	NaN	NaN	NaN
869	NaN	NaN	NaN	NaN	NaN
895	NaN	NaN	NaN	NaN	NaN
898	NaN	NaN	NaN	NaN	NaN
949	Gd	TA	NaN	Unf	Unf
985	NaN	NaN	NaN	NaN	NaN
1001	NaN	NaN	NaN	NaN	NaN
1012	NaN	NaN	NaN	NaN	NaN
1036	NaN	NaN	NaN	NaN	NaN
1046	NaN	NaN	NaN	NaN	NaN
1049	NaN	NaN	NaN	NaN	NaN
1050	NaN	NaN	NaN	NaN	NaN
1091	NaN	NaN	NaN	NaN	NaN
1180	NaN	NaN	NaN	NaN	NaN
1217	NaN	NaN	NaN	NaN	NaN
1219	NaN	NaN	NaN	NaN	NaN
1233	NaN	NaN	NaN	NaN	NaN
1322	NaN	NaN	NaN	NaN	NaN
1413	NaN	NaN	NaN	NaN	NaN
1488	Gd	TA	NaN	Unf	Unf
1586	NaN	NaN	NaN	NaN	NaN

1594	NaN	NaN	NaN	NaN	NaN	
1730	NaN	NaN	NaN	NaN	NaN	
1779	NaN	NaN	NaN	NaN	NaN	
1815	NaN	NaN	NaN	NaN	NaN	
1848	NaN	NaN	NaN	NaN	NaN	
1849	NaN	NaN	NaN	NaN	NaN	
1857	NaN	NaN	NaN	NaN	NaN	
1858	NaN	NaN	NaN	NaN	NaN	
1859	NaN	NaN	NaN	NaN	NaN	
1861	NaN	NaN	NaN	NaN	NaN	
1916	NaN	NaN	NaN	NaN	NaN	
2041	Gd	NaN	Mn	GLQ	Rec	10
2051	NaN	NaN	NaN	NaN	NaN	
2067	NaN	NaN	NaN	NaN	NaN	
2069	NaN	NaN	NaN	NaN	NaN	
2121	NaN	NaN	NaN	NaN	NaN	
2123	NaN	NaN	NaN	NaN	NaN	
2186	TA	NaN	No	BLQ	Unf	10
2189	NaN	NaN	NaN	NaN	NaN	
2190	NaN	NaN	NaN	NaN	NaN	
2191	NaN	NaN	NaN	NaN	NaN	
2194	NaN	NaN	NaN	NaN	NaN	
2217	NaN	NaN	NaN	NaN	NaN	
2218	NaN	Fa	No	Unf	Unf	
2219	NaN	TA	No	Unf	Unf	
2225	NaN	NaN	NaN	NaN	NaN	
2349	Gd	TA	NaN	Unf	Unf	
2388	NaN	NaN	NaN	NaN	NaN	
2436	NaN	NaN	NaN	NaN	NaN	
2453	NaN	NaN	NaN	NaN	NaN	
2454	NaN	NaN	NaN	NaN	NaN	

2491	NaN	NaN	NaN	NaN	NaN	
2499	NaN	NaN	NaN	NaN	NaN	
2525	TA	NaN	Av	ALQ	Unf	7
2548	NaN	NaN	NaN	NaN	NaN	
2553	NaN	NaN	NaN	NaN	NaN	
2565	NaN	NaN	NaN	NaN	NaN	
2579	NaN	NaN	NaN	NaN	NaN	
2600	NaN	NaN	NaN	NaN	NaN	
2703	NaN	NaN	NaN	NaN	NaN	
2764	NaN	NaN	NaN	NaN	NaN	
2767	NaN	NaN	NaN	NaN	NaN	
2804	NaN	NaN	NaN	NaN	NaN	
2805	NaN	NaN	NaN	NaN	NaN	
2825	NaN	NaN	NaN	NaN	NaN	
2892	NaN	NaN	NaN	NaN	NaN	
2905	NaN	NaN	NaN	NaN	NaN	

In [49]:
```
bsmt_num=0
for feat in num_bsmt_feat:
    df_mvi[feat].replace(np.nan,bsmt_num,inplace=True)
```

In [50]:
```
# Electrical        0.034258 -- KitchenQual      0.034258
df["Electrical"].value_counts()
```

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

In [51]:
```
df["KitchenQual"].value_counts()
```

Out[51]:
```
TA    1492
Gd    1151
Ex     205
Fa      70
Name: KitchenQual, dtype: int64
```

In [52]:
```
df_ekk=df[["Electrical","KitchenQual","KitchenAbvGr"]]
df_ekk[df_ekk.isnull().any(axis=1)]
```

Out[52]:

Id	Electrical	KitchenQual	KitchenAbvGr
1380	NaN	Gd	1
1556	SBrkr	NaN	1

In [53]:
```
electrical_mode=df["Electrical"].mode()[0]
df_mvi["Electrical"].replace(np.nan,electrical_mode,inplace=True)
df_mvi["Electrical"].isnull().sum()
```

Out[53]: 0

In [54]:
```
kitchenqual_mode=df["KitchenQual"].mode()[0]
df_mvi["KitchenQual"].replace(np.nan,kitchenqual_mode,inplace=True)
df_mvi["KitchenQual"].isnull().sum()
```

Out[54]: 0

In [55]:
```
# Functional       0.068517 - mode
# FireplaceQu     48.646797 - NA
# PoolQC          99.657417 - NA
# Fence           80.438506 - NA
# MiscFeature     96.402878 - NA
# SaleType         0.034258 - mode
```

In [56]:
```
print(df["Functional"].value_counts())
print("----")
print(df["FireplaceQu"].value_counts())
print("----")
print(df["PoolQC"].value_counts())
print("----")
print(df["Fence"].value_counts())
print("----")
print(df["MiscFeature"].value_counts())
print("----")
print(df["SaleType"].value_counts())
print("----")
```

```
Typ     2717
Min2      70
Min1      65
Mod       35
Maj1      19
Maj2       9
Sev        2
Name: Functional, dtype: int64
----
Gd    744
TA    592
Fa     74
Po     46
Ex     43
Name: FireplaceQu, dtype: int64
----
Ex    4
Gd    4
Fa    2
Name: PoolQC, dtype: int64
----
MnPrv    329
GdPrv    118
GdWo     112
MnWw      12
Name: Fence, dtype: int64
----
Shed    95
Gar2     5
Othr     4
TenC     1
Name: MiscFeature, dtype: int64
----
WD       2525
New       239
COD        87
ConLD      26
CWD        12
ConLI       9
ConLw       8
Oth         7
Con         5
Name: SaleType, dtype: int64
----
```

In [57]:
```
functional_mode=df["Functional"].mode()[0]
df_mvi["Functional"].replace(np.nan,functional_mode,inplace=True)
df_mvi["Functional"].isnull().sum()
```

Out[57]: 0

In [58]:
```
saletype_mode=df["SaleType"].mode()[0]
df_mvi["SaleType"].replace(np.nan,saletype_mode,inplace=True)
```

```
df_mvi["SaleType"].isnull().sum()
```

Out[58]: 0

In [59]:
```
other_cat_feat=["FireplaceQu","PoolQC","Fence","MiscFeature"]

other_cat_const="NA"
for feat in other_cat_feat:
    df_mvi[feat].replace(np.nan,other_cat_const,inplace=True)

for feat in other_cat_feat:
    print(df_mvi[feat].isnull().sum())
    print("----")
```

```
0
----
0
----
0
----
0
----
```

In [60]:
```
# cat
# GarageType        5.378554 - NA
# GarageFinish      5.447071 - NA
# GarageQual        5.447071 - NA
# GarageCond        5.447071 - NA

# num
# GarageYrBlt       5.447071
# GarageCars        0.034258
# GarageArea        0.034258
```

In [61]:
```
cat_garage_feat=["GarageType","GarageFinish","GarageQual","GarageCond"]
num_garage_feat=["GarageYrBlt","GarageCars","GarageArea"]

for feat in cat_garage_feat:
    print(df[feat].value_counts())
    print("----")

for feat in num_garage_feat:
    print(df[feat].value_counts())
    print("----")
```

```
Attchd     1723
Detchd      779
BuiltIn     186
Basment      36
2Types       23
CarPort      15
Name: GarageType, dtype: int64
----
Unf    1230
```

```
RFn     811
Fin     719
Name: GarageFinish, dtype: int64
----
TA    2604
Fa     124
Gd      24
Po       5
Ex       3
Name: GarageQual, dtype: int64
----
TA    2654
Fa      74
Gd      15
Po      14
Ex       3
Name: GarageCond, dtype: int64
----
2005.0    142
2006.0    115
2007.0    115
2004.0     99
2003.0     92
1977.0     66
2008.0     61
1998.0     58
2000.0     55
1999.0     54
2002.0     53
1950.0     51
1976.0     50
1993.0     49
1968.0     48
1997.0     44
1958.0     42
1978.0     41
1956.0     41
2001.0     41
1996.0     40
1994.0     39
1966.0     39
1960.0     37
1954.0     37
1967.0     36
1959.0     36
1964.0     35
1974.0     35
1979.0     35
1995.0     35
1962.0     35
1963.0     34
1957.0     34
1965.0     34
```

1920.0	33
1969.0	32
1970.0	32
1980.0	32
1961.0	31
2009.0	29
1973.0	29
1975.0	28
1992.0	27
1972.0	27
1930.0	27
1990.0	26
1940.0	25
1955.0	24
1971.0	24
1953.0	23
1939.0	21
1988.0	20
1984.0	19
1948.0	19
1989.0	19
1987.0	18
1985.0	18
1991.0	17
1951.0	17
1952.0	16
1926.0	15
1925.0	15
1981.0	15
1949.0	14
1941.0	14
1986.0	12
1938.0	11
1983.0	11
1910.0	10
1945.0	10
1946.0	9
1982.0	9
1924.0	8
1935.0	8
1922.0	8
1936.0	7
1928.0	7
1915.0	7
1916.0	6
1937.0	6
1900.0	6
1923.0	6
1942.0	6
1921.0	5
1947.0	5
2010.0	5
1927.0	5

```
1932.0      4
1934.0      4
1931.0      4
1918.0      3
1914.0      2
1929.0      2
1917.0      2
1895.0      1
1943.0      1
2207.0      1
1908.0      1
1896.0      1
1933.0      1
1906.0      1
1919.0      1
Name: GarageYrBlt, dtype: int64
----
2.0    1594
1.0     776
3.0     374
0.0     157
4.0      16
5.0       1
Name: GarageCars, dtype: int64
----
0.0       157
576.0      97
440.0      96
240.0      69
484.0      68
528.0      65
400.0      58
480.0      54
264.0      51
288.0      50
308.0      48
280.0      30
420.0      29
336.0      29
672.0      23
462.0      23
216.0      23
384.0      21
504.0      21
506.0      21
286.0      20
312.0      19
624.0      17
525.0      17
352.0      17
495.0      17
550.0      17
360.0      16
```

180.0	16
564.0	16
300.0	16
572.0	15
460.0	14
588.0	14
660.0	14
390.0	14
540.0	14
478.0	14
520.0	13
539.0	12
297.0	12
720.0	11
252.0	11
432.0	11
472.0	11
200.0	11
470.0	11
502.0	10
294.0	10
450.0	10
461.0	10
530.0	9
482.0	9
434.0	9
578.0	9
473.0	9
542.0	9
492.0	9
529.0	9
441.0	9
490.0	9
396.0	9
474.0	9
270.0	9
552.0	8
648.0	8
527.0	8
430.0	8
431.0	8
299.0	8
451.0	8
546.0	8
676.0	8
560.0	7
512.0	7
380.0	7
616.0	7
393.0	7
392.0	7
880.0	7
610.0	7

315.0	7
410.0	7
870.0	6
840.0	6
625.0	6
388.0	6
516.0	6
786.0	6
784.0	6
500.0	6
256.0	6
642.0	6
670.0	6
486.0	6
544.0	6
575.0	6
452.0	6
320.0	6
338.0	6
600.0	6
632.0	6
250.0	6
864.0	6
541.0	5
626.0	5
498.0	5
615.0	5
438.0	5
260.0	5
437.0	5
570.0	5
463.0	5
531.0	5
820.0	5
398.0	5
429.0	5
621.0	5
319.0	5
758.0	5
275.0	5
225.0	5
650.0	5
515.0	5
612.0	5
483.0	5
517.0	5
551.0	5
521.0	5
534.0	5
433.0	5
577.0	5
608.0	5
678.0	5

```
467.0        5
532.0        5
656.0        5
850.0        5
511.0        5
583.0        5
596.0        5
442.0        5
834.0        5
402.0        5
522.0        5
379.0        4
836.0        4
496.0        4
370.0        4
477.0        4
350.0        4
418.0        4
342.0        4
730.0        4
776.0        4
281.0        4
436.0        4
746.0        4
220.0        4
888.0        4
364.0        4
580.0        4
795.0        4
792.0        4
487.0        4
598.0        4
471.0        4
499.0        4
628.0        4
586.0        4
368.0        4
416.0        4
556.0        4
305.0        4
330.0        4
253.0        4
666.0        4
276.0        4
644.0        4
210.0        4
246.0        4
627.0        4
736.0        4
810.0        4
774.0        4
397.0        4
205.0        4
```

816.0	4
630.0	4
900.0	3
228.0	3
508.0	3
874.0	3
878.0	3
668.0	3
768.0	3
686.0	3
619.0	3
574.0	3
409.0	3
444.0	3
756.0	3
788.0	3
796.0	3
614.0	3
510.0	3
554.0	3
160.0	3
403.0	3
704.0	3
326.0	3
567.0	3
779.0	3
928.0	3
466.0	3
230.0	3
234.0	3
324.0	3
871.0	3
322.0	3
844.0	3
591.0	3
215.0	3
318.0	3
868.0	3
884.0	3
649.0	3
538.0	3
658.0	3
692.0	3
524.0	3
457.0	3
195.0	3
454.0	3
856.0	3
497.0	3
750.0	3
732.0	3
453.0	3
435.0	3

273.0	3
476.0	3
366.0	3
684.0	3
852.0	3
565.0	3
826.0	3
691.0	3
468.0	3
636.0	3
932.0	3
456.0	3
505.0	3
513.0	3
569.0	3
304.0	3
846.0	3
501.0	3
514.0	3
592.0	3
754.0	3
301.0	3
566.0	3
231.0	3
751.0	2
555.0	2
925.0	2
690.0	2
533.0	2
631.0	2
365.0	2
728.0	2
310.0	2
488.0	2
724.0	2
814.0	2
725.0	2
355.0	2
663.0	2
640.0	2
738.0	2
543.0	2
371.0	2
647.0	2
523.0	2
331.0	2
224.0	2
638.0	2
597.0	2
479.0	2
579.0	2
287.0	2
343.0	2

896.0	2
812.0	2
936.0	2
828.0	2
714.0	2
712.0	2
701.0	2
372.0	2
620.0	2
561.0	2
162.0	2
357.0	2
394.0	2
622.0	2
313.0	2
762.0	2
782.0	2
518.0	2
662.0	2
351.0	2
582.0	2
722.0	2
885.0	2
920.0	2
489.0	2
944.0	2
938.0	2
545.0	2
780.0	2
404.0	2
905.0	2
912.0	2
399.0	2
1052.0	2
464.0	2
311.0	2
748.0	2
548.0	2
968.0	2
894.0	2
386.0	2
590.0	2
866.0	2
683.0	2
447.0	2
271.0	2
839.0	2
680.0	2
711.0	2
772.0	2
282.0	2
685.0	2
702.0	2

898.0	2
606.0	2
843.0	2
594.0	2
573.0	2
641.0	2
509.0	2
493.0	2
296.0	2
617.0	2
422.0	2
427.0	2
908.0	2
706.0	2
721.0	2
164.0	2
603.0	2
292.0	2
765.0	2
558.0	2
645.0	2
408.0	2
283.0	2
349.0	2
789.0	2
605.0	2
618.0	2
865.0	2
675.0	2
682.0	2
412.0	2
895.0	2
423.0	2
818.0	2
481.0	2
389.0	2
924.0	2
303.0	2
726.0	2
831.0	2
494.0	2
800.0	2
439.0	2
249.0	1
266.0	1
904.0	1
207.0	1
811.0	1
1138.0	1
316.0	1
340.0	1
226.0	1
405.0	1

```
1184.0      1
1348.0      1
740.0       1
325.0       1
869.0       1
1314.0      1
1231.0      1
687.0       1
1150.0      1
557.0       1
698.0       1
715.0       1
428.0       1
1166.0      1
295.0       1
307.0       1
401.0       1
783.0       1
851.0       1
766.0       1
469.0       1
787.0       1
267.0       1
1488.0      1
1003.0      1
613.0       1
369.0       1
599.0       1
1154.0      1
100.0       1
571.0       1
1041.0      1
963.0       1
443.0       1
773.0       1
485.0       1
1085.0      1
899.0       1
959.0       1
803.0       1
760.0       1
584.0       1
449.0       1
688.0       1
568.0       1
353.0       1
791.0       1
1008.0      1
378.0       1
258.0       1
848.0       1
317.0       1
646.0       1
```

265.0	1
609.0	1
853.0	1
890.0	1
242.0	1
806.0	1
344.0	1
356.0	1
185.0	1
892.0	1
257.0	1
729.0	1
1110.0	1
585.0	1
1040.0	1
1174.0	1
916.0	1
876.0	1
933.0	1
747.0	1
1092.0	1
859.0	1
744.0	1
1105.0	1
293.0	1
1200.0	1
184.0	1
374.0	1
217.0	1
323.0	1
332.0	1
674.0	1
667.0	1
700.0	1
907.0	1
406.0	1
832.0	1
1134.0	1
1248.0	1
1043.0	1
254.0	1
719.0	1
862.0	1
562.0	1
749.0	1
261.0	1
842.0	1
1390.0	1
306.0	1
889.0	1
830.0	1
807.0	1
358.0	1

```
186.0       1
693.0       1
426.0       1
813.0       1
995.0       1
757.0       1
1356.0      1
459.0       1
367.0       1
716.0       1
739.0       1
290.0       1
665.0       1
611.0       1
425.0       1
1220.0      1
595.0       1
857.0       1
902.0       1
1020.0      1
455.0       1
414.0       1
354.0       1
602.0       1
327.0       1
284.0       1
833.0       1
601.0       1
841.0       1
689.0       1
808.0       1
752.0       1
255.0       1
424.0       1
824.0       1
328.0       1
983.0       1
475.0       1
858.0       1
954.0       1
549.0       1
927.0       1
535.0       1
263.0       1
375.0       1
363.0       1
209.0       1
1017.0      1
671.0       1
741.0       1
581.0       1
345.0       1
1053.0      1
```

```
413.0       1
458.0       1
694.0       1
886.0       1
949.0       1
673.0       1
309.0       1
815.0       1
623.0       1
972.0       1
984.0       1
604.0       1
845.0       1
559.0       1
465.0       1
713.0       1
962.0       1
958.0       1
708.0       1
526.0       1
1014.0      1
753.0       1
1418.0      1
213.0       1
198.0       1
860.0       1
248.0       1
696.0       1
825.0       1
947.0       1
373.0       1
770.0       1
639.0       1
377.0       1
804.0       1
244.0       1
208.0       1
445.0       1
189.0       1
1069.0      1
872.0       1
923.0       1
192.0       1
1025.0      1
272.0       1
Name: GarageArea, dtype: int64
----
```

In [62]:

```python
cat_garage_cont="NA"
for feat in cat_garage_feat:
    df_mvi[feat].replace(np.nan,cat_garage_cont,inplace=True)

num_garage_val=0
for feat in num_garage_feat:
```

```
    df_mvi[feat].replace(np.nan,num_garage_val,inplace=True)
```

In [63]:
```
# df_mvi[cat_garage_feat].isnull().sum()
# df_mvi[num_garage_feat].isnull().sum()
```

Feature Transformation

Numerical to Categorical

In [64]:
```
## MSSubClass,YearBuilt,YearRemodAdd,GarageYrBlt,MoSold,YrSold
for_num_con = ["MSSubClass","YearBuilt","YearRemodAdd","GarageYrBlt","MoS
for feat in for_num_con:
    print(f"{feat}: data type = {df_mvi[feat].dtype}")
```

```
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 [65]:
```
df_mvi["MoSold"]=df_mvi["MoSold"].apply(lambda x: calendar.month_abbr[x])
```

In [66]:
```
for feat in for_num_con:
    df_mvi[feat]=df_mvi[feat].astype(str)
```

In [67]:
```
for feat in for_num_con:
    print(f"{feat}: data type = {df_mvi[feat].dtype}")
```

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

Categorial into Numerical(ordinal objects)

In [68]:
```
ordinal_end_var=[
"ExterQual",
"ExterCond",
"BsmtQual",
"BsmtCond",
"BsmtExposure",
"BsmtFinType1",
"BsmtFinType2",
"HeatingQC",
"KitchenQual",
"FireplaceQu",
"GarageQual",
```

```
"GarageCond",
"PoolQC",
"Functional",
"GarageFinish",
"PavedDrive",
"Utilities",
]
print(len(ordinal_end_var))
```

17

In [69]:
```
df_mvi["ExterQual"]=df_mvi["ExterQual"].astype(CategoricalDtype(categorie
df_mvi["ExterCond"]=df_mvi["ExterCond"].astype(CategoricalDtype(categorie
df_mvi["BsmtQual"]=df_mvi["BsmtQual"].astype(CategoricalDtype(categories=
df_mvi["BsmtCond"]=df_mvi["BsmtCond"].astype(CategoricalDtype(categories=
df_mvi["BsmtExposure"]=df_mvi["BsmtExposure"].astype(CategoricalDtype(cat
df_mvi["BsmtFinType1"]=df_mvi["BsmtFinType1"].astype(CategoricalDtype(cat
df_mvi["BsmtFinType2"]=df_mvi["BsmtFinType2"].astype(CategoricalDtype(cat
df_mvi["HeatingQC"]=df_mvi["HeatingQC"].astype(CategoricalDtype(categorie
df_mvi["KitchenQual"]=df_mvi["KitchenQual"].astype(CategoricalDtype(categ
df_mvi["FireplaceQu"]=df_mvi["FireplaceQu"].astype(CategoricalDtype(categ
df_mvi["GarageQual"]=df_mvi["GarageQual"].astype(CategoricalDtype(categor
df_mvi["GarageCond"]=df_mvi["GarageCond"].astype(CategoricalDtype(categor
df_mvi["PoolQC"]=df_mvi["PoolQC"].astype(CategoricalDtype(categories=["NA
df_mvi["Functional"]=df_mvi["Functional"].astype(CategoricalDtype(categor
df_mvi["GarageFinish"]=df_mvi["GarageFinish"].astype(CategoricalDtype(cat
df_mvi["PavedDrive"]=df_mvi["PavedDrive"].astype(CategoricalDtype(categor
df_mvi["Utilities"]=df_mvi["Utilities"].astype(CategoricalDtype(categorie
```

In [70]:
```
df_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 [71]:
```
# df_mvi["ExterQual"]
```

Categorial into Numerical(nominal objects)

In [72]:
```
df_encod=df_mvi.copy()

object_features=df_encod.select_dtypes(include="object").columns.to_list(
print(object_features)
```

```
['MSSubClass', 'MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', '
LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'Bldg
Type', 'HouseStyle', 'YearBuilt', 'YearRemodAdd', 'RoofStyle', 'RoofMatl',
'Exterior1st', 'Exterior2nd', 'MasVnrType', 'Foundation', 'Heating', 'Cent
ralAir', 'Electrical', 'GarageType', 'GarageYrBlt', 'Fence', 'MiscFeatur
e', 'MoSold', 'YrSold', 'SaleType', 'SaleCondition']
```

In [73]:
```
df_encod[object_features].head(2)
```

Out[73]:

	MSSubClass	MSZoning	Street	Alley	LotShape	LandContour	LotConfig	Lan
Id								
1	60	RL	Pave	NA	Reg	Lvl	Inside	
2	20	RL	Pave	NA	Reg	Lvl	FR2	

In [74]:
```
print("before",df_encod.shape)
df_encod=pd.get_dummies(df_encod,
                        columns=object_features,
                        prefix=object_features,
                        drop_first=True)
print("after",df_encod.shape)
```

```
before (2919, 80)
after (2919, 513)
```

In [75]:
```
df_encod.head(2)
```

Out[75]:

	LotFrontage	LotArea	Utilities	OverallQual	OverallCond	MasVnrArea	ExterQu
Id							
1	65.0	8450	3	7	5	196.0	
2	80.0	9600	3	6	8	0.0	

Split data

In [76]:
```
len_train=df_train.shape[0]
len_train
```

Out[76]: 1460

In [77]:
```
X_train=df_encod[:len_train].drop("SalePrice",axis=1)
y_train=df_encod[:len_train]["SalePrice"]
X_test=df_encod[len_train:].drop("SalePrice",axis=1)


print("Shape of X_train",X_train.shape)
print("Shape of y_train",y_train.shape)
print("Shape of X_test",X_test.shape)
```

```
Shape of X_train (1460, 512)
Shape of y_train (1460,)
Shape of X_test (1459, 512)
```

In [78]:
```
sc = StandardScaler()

sc.fit(X_train) # it will learn about mean and std variance
X_train=sc.transform(X_train)
X_test=sc.transform(X_test)
```

Cross Validation and Model Selection

In [79]:
```
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 RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor

from sklearn.neural_network import MLPRegressor

from xgboost import XGBRegressor
```

In [80]:
```
svr = SVR()
lr = LinearRegression()
sgdr = SGDRegressor()
knn = KNeighborsRegressor()
gpr = GaussianProcessRegressor()
dtr = DecisionTreeRegressor()
rfr = RandomForestRegressor()
gbr = GradientBoostingRegressor()
xgbr = XGBRegressor()

mlpr = MLPRegressor()
```

In [81]:
```
models = {"a":["LinearRegression",lr],
          "b":["SVR",svr],
          "c":["SGDRegressor",sgdr],
          "d":["KNeighborsRegressor",knn],
          "e":["GaussianProcessRegressor",gpr],
          "f":["DecisionTreeRegressor",dtr],
          "g":["GradientBoostingRegressor",gbr],
          "h":["RandomForestRegressor",rfr],
          "i":["XGBRegressor",xgbr],
          "j":["MLPRegressor",mlpr],
          }     # Create a dictionary to store the results
```

In [82]:
```
from sklearn.model_selection import cross_val_score, KFold
from sklearn.metrics import make_scorer,r2_score

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

In [83]:
```
models_score=[]
for model in models:
    print("Model Name: ",models[model][0])
    score = test_model(models[model][1], X_train, y_train)
    print("Score of Model:",score)
    print("------------------------")
    models_score.append([models[model][0],score])
```

```
Model Name:  LinearRegression
Score of Model: [-1.2722448407974715e+24]
-----------------------
Model Name:  SVR
Score of Model: [-0.052133548352104216]
-----------------------
Model Name:  SGDRegressor
Score of Model: [-6043.376589984432]
-----------------------
Model Name:  KNeighborsRegressor
Score of Model: [0.5585925623107102]
-----------------------
Model Name:  GaussianProcessRegressor
Score of Model: [-5.398916312612151]
-----------------------
Model Name:  DecisionTreeRegressor
Score of Model: [0.7008206494020515]
-----------------------
Model Name:  GradientBoostingRegressor
Score of Model: [0.8715666756167462]
-----------------------
Model Name:  RandomForestRegressor
Score of Model: [0.8457476035359204]
-----------------------
Model Name:  XGBRegressor
Score of Model: [0.8582487612757063]
-----------------------
Model Name:  MLPRegressor
```

```
c:\Users\nirde\anaconda3\lib\site-packages\sklearn\neural_network\_multila
yer_perceptron.py:684: ConvergenceWarning: Stochastic Optimizer: Maximum i
terations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
c:\Users\nirde\anaconda3\lib\site-packages\sklearn\neural_network\_multila
yer_perceptron.py:684: ConvergenceWarning: Stochastic Optimizer: Maximum i
terations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
c:\Users\nirde\anaconda3\lib\site-packages\sklearn\neural_network\_multila
yer_perceptron.py:684: ConvergenceWarning: Stochastic Optimizer: Maximum i
terations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
c:\Users\nirde\anaconda3\lib\site-packages\sklearn\neural_network\_multila
yer_perceptron.py:684: ConvergenceWarning: Stochastic Optimizer: Maximum i
terations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
c:\Users\nirde\anaconda3\lib\site-packages\sklearn\neural_network\_multila
yer_perceptron.py:684: ConvergenceWarning: Stochastic Optimizer: Maximum i
terations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
c:\Users\nirde\anaconda3\lib\site-packages\sklearn\neural_network\_multila
yer_perceptron.py:684: ConvergenceWarning: Stochastic Optimizer: Maximum i
terations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
```

```
Score of Model: [-4.852157124690591]
-----------------------
```

```
c:\Users\nirde\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:684: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
```

In [84]:
```
models_score
```

Out[84]:
```
[['LinearRegression', [-1.2722448407974715e+24]],
 ['SVR', [-0.05213354835210421 6]],
 ['SGDRegressor', [-6043.376589984432]],
 ['KNeighborsRegressor', [0.5585925623107102]],
 ['GaussianProcessRegressor', [-5.398916312612151]],
 ['DecisionTreeRegressor', [0.7008206494020515]],
 ['GradientBoostingRegressor', [0.8715666756167462]],
 ['RandomForestRegressor', [0.8457476035359204]],
 ['XGBRegressor', [0.8582487612757063]],
 ['MLPRegressor', [-4.852157124690591]]]
```

Model Training

In [85]:
```
gbr.fit(X_train,y_train)
```

Out[85]:
```
▾ GradientBoostingRegressor
GradientBoostingRegressor()
```

In [86]:
```
y_pred=gbr.predict(X_test)
```

In [87]:
```
# y_pred # is a numpy array hence we gonna convert it into dataframe
y_pred=pd.concat([df_test['Id'],pd.DataFrame(y_pred,columns=['SalePrice']
y_pred
```

Out[87]:

	Id	SalePrice
0	1461	122103.500909
1	1462	155398.306856
2	1463	175032.391922
3	1464	180116.442684
4	1465	204131.608740
5	1466	170930.824326
6	1467	157931.059686
7	1468	159943.505730
8	1469	190038.522001

9	1470	129854.392427
10	1471	200965.020774
11	1472	95461.489798
12	1473	95677.490047
13	1474	154819.496462
14	1475	137990.081928
15	1476	407464.266211
16	1477	277826.803080
17	1478	311162.185967
18	1479	286536.780138
19	1480	484967.859526
20	1481	319955.713640
21	1482	213799.485481
22	1483	165842.688843
23	1484	174316.540813
24	1485	175527.486372
25	1486	193866.191964
26	1487	350948.923099
27	1488	239107.506370
28	1489	200610.223162
29	1490	222909.030296
30	1491	190933.714005
31	1492	89257.865711
32	1493	188818.811478
33	1494	292143.047463
34	1495	301955.268491
35	1496	235919.708953
36	1497	182223.884972
37	1498	165683.328424
38	1499	167733.284069
39	1500	146715.310659
40	1501	165761.689504

41	1502	159338.955452
42	1503	290292.318318
43	1504	228485.048420
44	1505	213225.858751
45	1506	191678.968133
46	1507	228042.622860
47	1508	195479.689872
48	1509	166041.677980
49	1510	145016.837279
50	1511	148873.274558
51	1512	178365.706433
52	1513	140476.074876
53	1514	168341.029077
54	1515	178321.250756
55	1516	163465.260063
56	1517	157512.957823
57	1518	145284.852879
58	1519	223650.222259
59	1520	132093.532511
60	1521	135562.659195
61	1522	166989.911132
62	1523	107692.268599
63	1524	120870.035256
64	1525	123556.409320
65	1526	124794.305518
66	1527	105503.673378
67	1528	133699.619091
68	1529	146055.499112
69	1530	181220.595430
70	1531	119190.434746
71	1532	106179.703188

72	1533	153585.916616
73	1534	114665.845976
74	1535	149604.690622
75	1536	113075.486553
76	1537	75780.606336
77	1538	184302.282671
78	1539	221181.034718
79	1540	108956.721840
80	1541	138725.393727
81	1542	130337.588753
82	1543	214516.892439
83	1544	84190.964390
84	1545	109251.432216
85	1546	126799.918659
86	1547	139660.828687
87	1548	137303.367651
88	1549	111319.608179
89	1550	133980.685975
90	1551	123568.127279
91	1552	131290.587942
92	1553	156106.771849
93	1554	109458.356379
94	1555	166263.894242
95	1556	81448.169260
96	1557	114725.742915
97	1558	101580.818266
98	1559	95914.527711
99	1560	136645.530469
100	1561	135531.311306
101	1562	125308.470082
102	1563	123784.639426
103	1564	166086.224743

104	1565	150653.398405
105	1566	223801.202415
106	1567	78778.279881
107	1568	227519.959920
108	1569	129156.753294
109	1570	140244.037675
110	1571	123945.225373
111	1572	145447.337056
112	1573	265371.476832
113	1574	116215.902988
114	1575	239867.725367
115	1576	224554.989355
116	1577	189216.565833
117	1578	143801.558212
118	1579	139046.099765
119	1580	189748.493283
120	1581	148871.409221
121	1582	129567.053906
122	1583	341271.818674
123	1584	239930.644859
124	1585	143925.872353
125	1586	77986.234090
126	1587	104936.301678
127	1588	153852.487005
128	1589	104145.393977
129	1590	137534.074984
130	1591	99056.328371
131	1592	111662.253093
132	1593	131678.000094
133	1594	124369.012517
134	1595	118804.196066

135	1596	239929.785561
136	1597	188095.762192
137	1598	208431.230373
138	1599	204108.850972
139	1600	200544.709478
140	1601	78228.651537
141	1602	111334.949757
142	1603	105921.150194
143	1604	287596.154935
144	1605	249070.309673
145	1606	160645.497180
146	1607	178069.321487
147	1608	209344.306732
148	1609	186625.055296
149	1610	143026.142954
150	1611	141214.118444
151	1612	178362.928403
152	1613	155986.102845
153	1614	126767.172758
154	1615	89835.199781
155	1616	69680.816359
156	1617	89666.042474
157	1618	122702.361549
158	1619	137287.051012
159	1620	144900.375319
160	1621	128582.821581
161	1622	137336.085077
162	1623	301797.682282
163	1624	216966.957052
164	1625	131329.458828
165	1626	170096.812121
166	1627	191869.986549

167	1628	285184.961101
168	1629	174793.024141
169	1630	406087.401745
170	1631	238236.405258
171	1632	262256.213472
172	1633	166806.890284
173	1634	173664.161584
174	1635	182432.887194
175	1636	151499.528179
176	1637	186118.194329
177	1638	218236.337527
178	1639	180151.779664
179	1640	262064.919504
180	1641	170265.200330
181	1642	222894.088369
182	1643	224848.495045
183	1644	226359.141374
184	1645	180931.528303
185	1646	154485.850789
186	1647	168075.623506
187	1648	132775.020507
188	1649	139320.378446
189	1650	119231.007540
190	1651	121969.326572
191	1652	98405.885982
192	1653	96737.744230
193	1654	152283.522853
194	1655	160423.907191
195	1656	147972.257628
196	1657	149303.225431
197	1658	150545.561578

198	1659	145842.400109
199	1660	148731.125115
200	1661	442939.354340
201	1662	453892.305842
202	1663	383233.953574
203	1664	418047.054484
204	1665	319058.431912
205	1666	303062.146127
206	1667	356688.109284
207	1668	330779.359283
208	1669	311564.912080
209	1670	350168.225727
210	1671	243964.875191
211	1672	471676.734137
212	1673	301095.252836
213	1674	243415.019459
214	1675	169428.110458
215	1676	173664.695174
216	1677	200345.882604
217	1678	457468.890700
218	1679	402970.099861
219	1680	332540.609714
220	1681	275816.522858
221	1682	278527.701360
222	1683	177707.032387
223	1684	174145.331234
224	1685	168321.141344
225	1686	175542.377939
226	1687	166035.265398
227	1688	183565.783038
228	1689	187709.176179
229	1690	187284.226461

230	1691	180333.142832
231	1692	274860.474197
232	1693	168080.008265
233	1694	173241.081707
234	1695	163771.734289
235	1696	275793.748224
236	1697	163357.613075
237	1698	384989.254887
238	1699	318020.240324
239	1700	247430.003905
240	1701	259903.885343
241	1702	255450.832207
242	1703	244246.369865
243	1704	288459.136582
244	1705	236869.951407
245	1706	423617.999250
246	1707	223929.219935
247	1708	199502.122313
248	1709	285776.291647
249	1710	224596.802780
250	1711	253632.254146
251	1712	254681.380286
252	1713	263213.848509
253	1714	213753.108432
254	1715	203617.064648
255	1716	178797.116454
256	1717	180175.246507
257	1718	136352.401384
258	1719	201184.183914
259	1720	248686.347955
260	1721	177041.854855

261	1722	130838.164137
262	1723	151196.427545
263	1724	196430.741920
264	1725	242294.177147
265	1726	191508.387725
266	1727	160305.914473
267	1728	184017.523550
268	1729	159093.627400
269	1730	163714.147574
270	1731	124442.744657
271	1732	133220.216659
272	1733	115866.421804
273	1734	116479.922261
274	1735	139436.001715
275	1736	113158.487483
276	1737	353541.392599
277	1738	204446.241368
278	1739	277285.415934
279	1740	226413.546506
280	1741	190317.579741
281	1742	164695.504566
282	1743	175111.566533
283	1744	291933.579249
284	1745	214052.681733
285	1746	239285.083887
286	1747	231342.613021
287	1748	220995.606774
288	1749	153390.960237
289	1750	129304.459038
290	1751	244891.288802
291	1752	121880.785588
292	1753	148172.370521

293	1754	207104.826921
294	1755	166483.546078
295	1756	127570.231639
296	1757	122411.707098
297	1758	141232.580924
298	1759	160032.158525
299	1760	162892.512816
300	1761	146682.710964
301	1762	168961.833914
302	1763	179650.624443
303	1764	118311.154861
304	1765	177756.337978
305	1766	182550.835202
306	1767	207017.809727
307	1768	154586.246070
308	1769	179163.648327
309	1770	149322.521346
310	1771	137883.749309
311	1772	143361.247801
312	1773	127733.549086
313	1774	145246.785031
314	1775	140424.561024
315	1776	127157.584886
316	1777	115018.288574
317	1778	152332.936428
318	1779	116834.116153
319	1780	179763.837054
320	1781	138842.610335
321	1782	90064.673492
322	1783	128865.370079
323	1784	103461.695626

324	1785	131387.790663
325	1786	170333.270480
326	1787	171590.660090
327	1788	74216.822797
328	1789	115302.422516
329	1790	86251.919389
330	1791	187316.688099
331	1792	157712.234059
332	1793	127501.982564
333	1794	148245.546856
334	1795	133162.634375
335	1796	131894.811237
336	1797	124973.675601
337	1798	127436.775356
338	1799	116293.764391
339	1800	142172.513977
340	1801	124819.940795
341	1802	144062.076494
342	1803	156425.887437
343	1804	145692.868801
344	1805	138964.868469
345	1806	113527.961629
346	1807	142799.723292
347	1808	119987.227634
348	1809	120487.668306
349	1810	132957.503800
350	1811	97315.031698
351	1812	103778.096959
352	1813	116019.622135
353	1814	102670.910840
354	1815	55164.416741
355	1816	103538.008742

356	1817	120533.845583
357	1818	168508.058041
358	1819	126678.354317
359	1820	76456.089879
360	1821	114376.191650
361	1822	152260.066493
362	1823	57357.333595
363	1824	140058.198711
364	1825	131878.621679
365	1826	106880.519028
366	1827	99115.658764
367	1828	126987.070167
368	1829	117134.850555
369	1830	136224.373660
370	1831	175309.272626
371	1832	85960.376533
372	1833	141223.084241
373	1834	127253.577872
374	1835	134696.416163
375	1836	128392.221605
376	1837	105580.181351
377	1838	130015.355036
378	1839	87398.084519
379	1840	154724.179253
380	1841	149794.529721
381	1842	71132.488599
382	1843	134429.360726
383	1844	147009.820767
384	1845	137215.373700
385	1846	156733.893630
386	1847	157580.460485

387	1848	57635.633050
388	1849	115699.255373
389	1850	121010.441624
390	1851	162733.024038
391	1852	131393.498279
392	1853	145506.400740
393	1854	177543.751176
394	1855	142195.540922
395	1856	219528.758033
396	1857	163954.924522
397	1858	128440.277126
398	1859	123802.040823
399	1860	145836.045891
400	1861	123802.040823
401	1862	259782.418505
402	1863	255012.349554
403	1864	255012.349554
404	1865	365474.024999
405	1866	340146.425409
406	1867	243818.300590
407	1868	297942.589265
408	1869	188774.040230
409	1870	247990.436734
410	1871	261963.871600
411	1872	170336.396826
412	1873	216067.552118
413	1874	134807.258192
414	1875	199091.097570
415	1876	185851.906788
416	1877	221285.669067
417	1878	202554.172983
418	1879	139541.739493

419	1880	135062.457199
420	1881	233504.520249
421	1882	251329.122819
422	1883	193999.193628
423	1884	203346.428513
424	1885	265838.812396
425	1886	309120.175411
426	1887	203571.444720
427	1888	281405.431642
428	1889	166667.007168
429	1890	131987.880664
430	1891	129040.652971
431	1892	110667.578491
432	1893	134351.289345
433	1894	138309.098160
434	1895	145086.739672
435	1896	126323.614340
436	1897	111627.514187
437	1898	114121.564959
438	1899	164622.079824
439	1900	156753.211820
440	1901	165595.705905
441	1902	140216.042592
442	1903	214458.926413
443	1904	138997.482202
444	1905	188985.630934
445	1906	161648.792195
446	1907	208834.195593
447	1908	110307.371184
448	1909	143962.394197
449	1910	127307.999984

450	1911	209931.521106
451	1912	296340.817280
452	1913	143653.863729
453	1914	81337.348686
454	1915	305542.644206
455	1916	77598.513521
456	1917	276417.892961
457	1918	140570.988857
458	1919	162303.371537
459	1920	152745.642194
460	1921	423193.308976
461	1922	340936.197084
462	1923	204478.614938
463	1924	211665.393127
464	1925	216786.142823
465	1926	394354.768631
466	1927	139755.845147
467	1928	151855.880134
468	1929	125427.645208
469	1930	124058.993381
470	1931	137095.214957
471	1932	131073.724920
472	1933	178653.891999
473	1934	164643.920619
474	1935	169474.046008
475	1936	182195.399590
476	1937	175511.405332
477	1938	173687.255183
478	1939	259722.902574
479	1940	206279.603941
480	1941	164313.255703
481	1942	187993.869937

482	1943	223909.675379
483	1944	338175.854043
484	1945	390561.063034
485	1946	139476.402544
486	1947	325639.238005
487	1948	181078.903192
488	1949	231745.112125
489	1950	185875.324314
490	1951	290440.875116
491	1952	217235.227237
492	1953	162513.208131
493	1954	196821.808032
494	1955	124774.690558
495	1956	312855.221618
496	1957	155760.063155
497	1958	304052.706911
498	1959	137629.482481
499	1960	114722.059117
500	1961	119397.648084
501	1962	99406.399903
502	1963	102950.652157
503	1964	105381.523207
504	1965	148532.841204
505	1966	158289.513455
506	1967	286530.601051
507	1968	414771.961395
508	1969	403217.185526
509	1970	445361.466827
510	1971	453278.119322
511	1972	375608.569358
512	1973	271721.920680

513	1974	315611.831036
514	1975	442526.433178
515	1976	265024.725464
516	1977	340351.429871
517	1978	387277.120179
518	1979	350821.409379
519	1980	183394.735802
520	1981	354871.985941
521	1982	219338.326160
522	1983	212257.841225
523	1984	171649.168295
524	1985	220775.150701
525	1986	210530.217298
526	1987	183013.068567
527	1988	173864.550585
528	1989	193866.191964
529	1990	216719.450682
530	1991	219950.376888
531	1992	207033.590833
532	1993	169061.757553
533	1994	234601.087321
534	1995	181671.914359
535	1996	249298.642241
536	1997	321639.118711
537	1998	359119.653242
538	1999	267959.689277
539	2000	308133.988149
540	2001	253984.774322
541	2002	237424.176073
542	2003	252665.575101
543	2004	246325.863892
544	2005	232317.227690

545	2006	238268.788334
546	2007	246360.613172
547	2008	212364.449175
548	2009	186728.032657
549	2010	195467.443020
550	2011	143726.132232
551	2012	176611.804464
552	2013	183177.305611
553	2014	185589.972732
554	2015	205419.662176
555	2016	188744.446252
556	2017	203677.471523
557	2018	119075.235210
558	2019	132066.134520
559	2020	111896.880618
560	2021	116813.485508
561	2022	177422.780974
562	2023	138592.728767
563	2024	232178.913672
564	2025	368657.151283
565	2026	180670.989015
566	2027	146715.310659
567	2028	153972.962867
568	2029	159409.294208
569	2030	272247.794488
570	2031	259384.331604
571	2032	278103.381666
572	2033	278528.956979
573	2034	165448.209266
574	2035	197769.887762
575	2036	199002.655540

576	2037	197042.950750
577	2038	319053.060450
578	2039	236978.522408
579	2040	363696.389138
580	2041	262431.000401
581	2042	194237.573734
582	2043	169389.989789
583	2044	176009.886901
584	2045	203078.295505
585	2046	144073.222798
586	2047	144285.755433
587	2048	137584.674550
588	2049	140693.324510
589	2050	190478.072420
590	2051	108867.837301
591	2052	131172.909519
592	2053	143537.375981
593	2054	85174.542076
594	2055	166027.738391
595	2056	139425.061040
596	2057	126757.551001
597	2058	220267.672085
598	2059	130891.533921
599	2060	167435.578416
600	2061	181955.402069
601	2062	132093.532511
602	2063	117523.432419
603	2064	138627.091015
604	2065	129050.535163
605	2066	176356.843167
606	2067	122980.119095
607	2068	154986.878156

608	2069	85026.101619
609	2070	111482.933336
610	2071	91652.138387
611	2072	120307.648756
612	2073	139135.714578
613	2074	182966.351502
614	2075	145693.564720
615	2076	127107.191875
616	2077	151859.410064
617	2078	130022.651275
618	2079	135842.940189
619	2080	119140.428709
620	2081	123644.716948
621	2082	126183.760343
622	2083	140472.356152
623	2084	112360.318400
624	2085	118708.307049
625	2086	121485.942202
626	2087	117549.667116
627	2088	88977.861740
628	2089	73223.399077
629	2090	136174.193012
630	2091	118325.070328
631	2092	122738.062790
632	2093	126294.889736
633	2094	140751.139227
634	2095	148469.397012
635	2096	80460.251806
636	2097	95822.512262
637	2098	141022.176249
638	2099	66628.397051

639	2100	93973.256930
640	2101	114670.583783
641	2102	142759.266972
642	2103	100747.316613
643	2104	129127.628043
644	2105	122722.507832
645	2106	62055.756216
646	2107	222339.174646
647	2108	110122.422172
648	2109	107145.647074
649	2110	131963.393085
650	2111	164552.642971
651	2112	139325.052891
652	2113	115385.519041
653	2114	106492.305915
654	2115	156512.997569
655	2116	115871.593956
656	2117	139659.737535
657	2118	119751.244842
658	2119	106210.832819
659	2120	116017.395704
660	2121	88551.334076
661	2122	105119.432476
662	2123	84349.168161
663	2124	176189.263779
664	2125	129995.567606
665	2126	147945.055687
666	2127	154971.293467
667	2128	137297.688924
668	2129	109645.036585
669	2130	135188.779326
670	2131	149294.888841

671	2132	113290.441558
672	2133	120303.292559
673	2134	121343.437702
674	2135	97395.892385
675	2136	75348.967316
676	2137	124823.817842
677	2138	130327.937245
678	2139	155751.739445
679	2140	139258.847877
680	2141	139973.396949
681	2142	125208.369562
682	2143	149110.562337
683	2144	138265.725598
684	2145	138354.296763
685	2146	168792.806973
686	2147	134023.871112
687	2148	136411.988632
688	2149	147833.550851
689	2150	211689.142861
690	2151	133538.348972
691	2152	181489.544684
692	2153	170688.690823
693	2154	112032.809274
694	2155	137701.679766
695	2156	237444.047542
696	2157	227803.380678
697	2158	244980.173482
698	2159	218736.963228
699	2160	174253.742199
700	2161	240584.495413
701	2162	401231.789229

702	2163	355926.600134
703	2164	254668.002355
704	2165	180783.106513
705	2166	139485.885191
706	2167	218640.260143
707	2168	200772.919822
708	2169	198495.186448
709	2170	216929.969241
710	2171	145874.055831
711	2172	134713.822545
712	2173	156486.373913
713	2174	234261.165696
714	2175	323021.397890
715	2176	315392.426547
716	2177	242690.687487
717	2178	212346.863301
718	2179	143925.872353
719	2180	214626.174512
720	2181	188765.663722
721	2182	203042.149859
722	2183	185901.793039
723	2184	129323.729822
724	2185	117748.237584
725	2186	134754.000465
726	2187	153222.489302
727	2188	158339.614033
728	2189	200232.156333
729	2190	80994.853231
730	2191	85336.887276
731	2192	94881.509532
732	2193	109825.012275
733	2194	103733.712739

734	2195	121214.301947
735	2196	109639.141984
736	2197	118249.236656
737	2198	165433.969281
738	2199	170664.146195
739	2200	147865.061972
740	2201	145205.061129
741	2202	218073.798874
742	2203	151781.243721
743	2204	173232.048678
744	2205	116243.660246
745	2206	134137.968280
746	2207	209316.292543
747	2208	251667.862467
748	2209	213577.682278
749	2210	126260.835986
750	2211	118611.405501
751	2212	107088.569023
752	2213	110251.981790
753	2214	125798.147371
754	2215	106618.777440
755	2216	150784.601374
756	2217	96918.639977
757	2218	88845.496071
758	2219	73842.679324
759	2220	89865.287412
760	2221	305542.644206
761	2222	264188.037254
762	2223	280861.003538
763	2224	214163.764888
764	2225	118702.953916

765	2226	180965.952017
766	2227	192252.963365
767	2228	266600.375206
768	2229	237732.914058
769	2230	139597.289111
770	2231	215677.588836
771	2232	178948.054962
772	2233	174757.049888
773	2234	251349.281120
774	2235	223020.021502
775	2236	266610.239450
776	2237	347919.498190
777	2238	201349.947576
778	2239	116769.821213
779	2240	160002.875439
780	2241	165766.399352
781	2242	136531.685637
782	2243	131741.772719
783	2244	110497.455614
784	2245	111005.399431
785	2246	146148.951836
786	2247	105347.369317
787	2248	126543.584847
788	2249	124440.570868
789	2250	136607.707582
790	2251	123128.651092
791	2252	172413.807692
792	2253	153163.872324
793	2254	170193.177720
794	2255	182145.983390
795	2256	187052.069150
796	2257	184918.511274

797	2258	170770.896306
798	2259	165912.141216
799	2260	168278.733020
800	2261	205333.312374
801	2262	229471.535463
802	2263	344433.828510
803	2264	402397.623027
804	2265	183305.876667
805	2266	295655.492474
806	2267	386465.566748
807	2268	399613.857681
808	2269	148536.018769
809	2270	187348.104740
810	2271	228389.561599
811	2272	204054.048231
812	2273	159593.658187
813	2274	164451.304166
814	2275	164634.941954
815	2276	192604.906970
816	2277	183630.701592
817	2278	145647.324109
818	2279	129186.244500
819	2280	127320.762210
820	2281	159245.350962
821	2282	181055.871043
822	2283	106150.515094
823	2284	114278.678680
824	2285	162419.342510
825	2286	121524.722655
826	2287	361070.197185
827	2288	277871.232648

828	2289	384760.567319
829	2290	442843.282126
830	2291	333061.854161
831	2292	424449.269208
832	2293	451734.107512
833	2294	438140.673018
834	2295	431343.319019
835	2296	262123.430103
836	2297	312734.808817
837	2298	334443.950664
838	2299	406726.700812
839	2300	339724.404164
840	2301	277837.634126
841	2302	237815.284010
842	2303	246558.209712
843	2304	229583.134314
844	2305	179848.914165
845	2306	172658.767775
846	2307	169662.016863
847	2308	221672.192699
848	2309	276644.893924
849	2310	229119.720966
850	2311	196845.684085
851	2312	171718.158906
852	2313	175068.002282
853	2314	181462.233032
854	2315	178158.310860
855	2316	192821.240695
856	2317	167718.807998
857	2318	176135.718854
858	2319	179293.474756
859	2320	190512.194127

860	2321	247268.930525
861	2322	178411.703382
862	2323	182060.932374
863	2324	168764.266609
864	2325	207927.188234
865	2326	166245.141742
866	2327	214651.477439
867	2328	225749.908321
868	2329	176050.458530
869	2330	167964.386069
870	2331	345012.754048
871	2332	415866.080486
872	2333	326258.953227
873	2334	257855.248022
874	2335	260014.895930
875	2336	343413.479166
876	2337	196271.263012
877	2338	282398.812802
878	2339	207243.816232
879	2340	398499.385858
880	2341	206794.019495
881	2342	231991.822470
882	2343	212883.837676
883	2344	220133.589355
884	2345	244637.731965
885	2346	229138.523725
886	2347	197013.017863
887	2348	241255.260368
888	2349	168303.272111
889	2350	280210.703130
890	2351	238701.413857

891	2352	252288.913169
892	2353	235337.303092
893	2354	135583.396363
894	2355	162896.770382
895	2356	156235.006251
896	2357	188899.641381
897	2358	198382.936383
898	2359	127467.845344
899	2360	115346.511767
900	2361	142777.457472
901	2362	270549.309630
902	2363	141235.467838
903	2364	160112.130318
904	2365	215273.488580
905	2366	206925.451670
906	2367	225427.606248
907	2368	221754.973766
908	2369	221466.232028
909	2370	158286.944417
910	2371	168733.881859
911	2372	186058.514206
912	2373	270345.009609
913	2374	309627.395409
914	2375	262675.054184
915	2376	300414.794771
916	2377	365046.762197
917	2378	138930.115070
918	2379	188377.063404
919	2380	140173.987312
920	2381	164938.396139
921	2382	191316.143481
922	2383	193236.546027

923	2384	245536.146466
924	2385	158564.468078
925	2386	133975.880012
926	2387	136890.460685
927	2388	104015.854390
928	2389	130200.002855
929	2390	149406.187029
930	2391	142702.228061
931	2392	118872.065049
932	2393	171312.283365
933	2394	145864.201167
934	2395	191575.184802
935	2396	139898.003339
936	2397	214389.309161
937	2398	130305.572930
938	2399	65562.126004
939	2400	62168.808983
940	2401	106128.043157
941	2402	138824.110077
942	2403	144868.739635
943	2404	147098.846658
944	2405	153012.044511
945	2406	143012.823932
946	2407	129526.019413
947	2408	147605.916863
948	2409	125337.177080
949	2410	168278.306865
950	2411	120886.175020
951	2412	161272.228204
952	2413	132673.663548
953	2414	137858.760096

954	2415	139701.562058
955	2416	134625.366742
956	2417	137877.329668
957	2418	125809.806812
958	2419	118531.354738
959	2420	125934.672131
960	2421	149917.115376
961	2422	118835.631452
962	2423	134195.713920
963	2424	148885.363038
964	2425	176037.437327
965	2426	131792.012683
966	2427	145993.455873
967	2428	163786.532330
968	2429	120232.853855
969	2430	130535.907968
970	2431	110076.827680
971	2432	150748.506027
972	2433	147455.053920
973	2434	139556.082516
974	2435	146927.097619
975	2436	109921.969906
976	2437	107420.736843
977	2438	127757.154664
978	2439	114878.513104
979	2440	119629.048048
980	2441	105035.989320
981	2442	100150.380957
982	2443	114447.847153
983	2444	129222.641351
984	2445	76586.234921
985	2446	132050.948706

986	2447	193672.232733
987	2448	135786.300414
988	2449	103876.674269
989	2450	144933.996312
990	2451	118898.236635
991	2452	205848.533101
992	2453	92565.470037
993	2454	134453.807170
994	2455	111704.017453
995	2456	131222.228499
996	2457	114793.018556
997	2458	124489.029525
998	2459	113033.349716
999	2460	138847.686668
1000	2461	111041.702526
1001	2462	130623.319818
1002	2463	126210.179992
1003	2464	155679.087783
1004	2465	136974.347749
1005	2466	113371.906659
1006	2467	151141.560962
1007	2468	75080.302868
1008	2469	74211.168523
1009	2470	179129.017387
1010	2471	205192.895298
1011	2472	159680.471477
1012	2473	117578.861036
1013	2474	99540.136545
1014	2475	216010.728926
1015	2476	106392.432472
1016	2477	129266.284605

1017	2478	160500.556211
1018	2479	98974.258267
1019	2480	139039.592362
1020	2481	118044.600779
1021	2482	130633.969857
1022	2483	116223.404243
1023	2484	116407.477740
1024	2485	126051.260786
1025	2486	149653.277818
1026	2487	173940.655891
1027	2488	163192.374752
1028	2489	144874.546473
1029	2490	146817.447981
1030	2491	99724.450585
1031	2492	177012.883411
1032	2493	146128.268565
1033	2494	157847.225271
1034	2495	94721.121974
1035	2496	235494.793156
1036	2497	154099.798231
1037	2498	111761.017642
1038	2499	96624.280398
1039	2500	130033.988903
1040	2501	142573.349566
1041	2502	138256.199856
1042	2503	95125.158685
1043	2504	170586.622790
1044	2505	230855.483448
1045	2506	254823.528797
1046	2507	318565.543203
1047	2508	254823.528797
1048	2509	210512.735788

1049	2510	223123.529338
1050	2511	177648.975191
1051	2512	204170.978101
1052	2513	214184.469393
1053	2514	257231.902663
1054	2515	144933.655442
1055	2516	156912.135645
1056	2517	133397.009253
1057	2518	148948.867250
1058	2519	226321.559806
1059	2520	203565.940856
1060	2521	190103.508356
1061	2522	220457.851627
1062	2523	126886.046850
1063	2524	137924.955364
1064	2525	138557.315124
1065	2526	140494.010410
1066	2527	110251.678615
1067	2528	127889.282437
1068	2529	131907.706096
1069	2530	117353.710007
1070	2531	250370.707734
1071	2532	223200.022360
1072	2533	195564.604867
1073	2534	230948.293083
1074	2535	319306.520090
1075	2536	229991.601045
1076	2537	241048.307078
1077	2538	181100.156872
1078	2539	187972.088870
1079	2540	172740.869549

1080	2541	177278.733124
1081	2542	163865.899626
1082	2543	135278.716838
1083	2544	130657.787166
1084	2545	138416.426470
1085	2546	137400.061645
1086	2547	139508.915856
1087	2548	152280.099784
1088	2549	154676.130153
1089	2550	266389.752153
1090	2551	136905.333556
1091	2552	123315.392904
1092	2553	74097.383065
1093	2554	112471.265467
1094	2555	124765.486385
1095	2556	110139.255641
1096	2557	107825.405902
1097	2558	156816.423652
1098	2559	145872.423497
1099	2560	132821.218327
1100	2561	155313.213550
1101	2562	135643.546896
1102	2563	146381.932415
1103	2564	189871.222916
1104	2565	152156.395731
1105	2566	153383.580816
1106	2567	132337.881065
1107	2568	190134.795722
1108	2569	192292.258392
1109	2570	124216.191486
1110	2571	181203.744639
1111	2572	184755.567123

1112	2573	204549.113855
1113	2574	291784.825069
1114	2575	138986.203624
1115	2576	124520.261254
1116	2577	143361.758837
1117	2578	87958.718976
1118	2579	71108.654408
1119	2580	124721.555946
1120	2581	129337.240934
1121	2582	127090.489202
1122	2583	275938.300916
1123	2584	154481.226119
1124	2585	183698.466029
1125	2586	229634.600175
1126	2587	202292.907159
1127	2588	151382.592476
1128	2589	143774.954870
1129	2590	216360.335390
1130	2591	215314.477424
1131	2592	209685.667444
1132	2593	240546.682830
1133	2594	180447.566350
1134	2595	213047.224732
1135	2596	320307.607508
1136	2597	180958.920625
1137	2598	266347.345571
1138	2599	332552.735766
1139	2600	175009.561050
1140	2601	141774.980052
1141	2602	80927.953208
1142	2603	108506.535438

1143	2604	88633.253360
1144	2605	74657.270150
1145	2606	149794.388626
1146	2607	182914.825615
1147	2608	192764.379580
1148	2609	163169.297376
1149	2610	117247.799571
1150	2611	124683.000911
1151	2612	144511.514434
1152	2613	135569.183819
1153	2614	126890.121995
1154	2615	147533.906789
1155	2616	145586.092420
1156	2617	181208.318040
1157	2618	205649.983156
1158	2619	202377.133624
1159	2620	170985.686870
1160	2621	177733.379655
1161	2622	174846.631447
1162	2623	246272.853162
1163	2624	291920.513099
1164	2625	329032.616542
1165	2626	157431.877801
1166	2627	192693.827508
1167	2628	450318.117352
1168	2629	537455.518583
1169	2630	381013.025686
1170	2631	459604.372832
1171	2632	442807.168235
1172	2633	326530.823749
1173	2634	409334.579433
1174	2635	153765.344729

1175	2636	182300.355941
1176	2637	193919.875114
1177	2638	250824.615990
1178	2639	184440.476257
1179	2640	148639.078247
1180	2641	100048.981664
1181	2642	191956.006119
1182	2643	98946.295197
1183	2644	128908.335544
1184	2645	103612.159308
1185	2646	100304.342267
1186	2647	103214.740094
1187	2648	141689.656450
1188	2649	149762.783990
1189	2650	152604.982088
1190	2651	147188.656957
1191	2652	414505.568524
1192	2653	241148.461242
1193	2654	245781.522907
1194	2655	397730.433380
1195	2656	336298.884500
1196	2657	352261.157009
1197	2658	285387.835076
1198	2659	324092.272004
1199	2660	340731.560583
1200	2661	363039.246479
1201	2662	364617.460007
1202	2663	270309.897595
1203	2664	271874.042643
1204	2665	345356.857580
1205	2666	300917.257071

1206	2667	162923.309745
1207	2668	183565.740993
1208	2669	171137.209845
1209	2670	291596.865607
1210	2671	186028.421099
1211	2672	192745.365791
1212	2673	205717.552910
1213	2674	200784.156851
1214	2675	187875.798541
1215	2676	184933.256710
1216	2677	183104.989773
1217	2678	255164.386088
1218	2679	281723.377024
1219	2680	281814.912264
1220	2681	410922.515303
1221	2682	354091.123224
1222	2683	512078.057065
1223	2684	320905.534757
1224	2685	362277.654200
1225	2686	252648.635002
1226	2687	322041.484602
1227	2688	226504.751447
1228	2689	226579.842434
1229	2690	464276.530870
1230	2691	183783.359496
1231	2692	130880.541183
1232	2693	193493.434567
1233	2694	142820.261338
1234	2695	190275.278795
1235	2696	181356.995865
1236	2697	176989.682095
1237	2698	186567.753253

1238	2699	181302.373906
1239	2700	162984.969895
1240	2701	149997.290142
1241	2702	110770.441376
1242	2703	142751.817856
1243	2704	133262.102753
1244	2705	116655.040580
1245	2706	116372.667809
1246	2707	126933.157936
1247	2708	135912.359426
1248	2709	110916.536860
1249	2710	123556.409320
1250	2711	291597.230862
1251	2712	391690.326849
1252	2713	174262.996775
1253	2714	155516.364721
1254	2715	165554.933991
1255	2716	152740.304628
1256	2717	189694.709826
1257	2718	224680.380491
1258	2719	157246.991233
1259	2720	176008.235173
1260	2721	136938.850451
1261	2722	168281.838003
1262	2723	145575.825905
1263	2724	130455.416018
1264	2725	132839.329707
1265	2726	141879.253836
1266	2727	180093.927092
1267	2728	166319.061480
1268	2729	151944.398559

1269	2730	141852.730709
1270	2731	134019.341310
1271	2732	128565.101317
1272	2733	154547.796104
1273	2734	157030.342938
1274	2735	138607.196576
1275	2736	143803.218195
1276	2737	122772.754375
1277	2738	139701.562058
1278	2739	161585.442242
1279	2740	136999.903080
1280	2741	147771.728268
1281	2742	145094.936797
1282	2743	142484.358922
1283	2744	148637.563701
1284	2745	143022.074618
1285	2746	143197.413301
1286	2747	161416.436657
1287	2748	120763.220077
1288	2749	130114.399214
1289	2750	139151.002706
1290	2751	138521.257288
1291	2752	216267.858226
1292	2753	149711.324974
1293	2754	215269.882938
1294	2755	126947.672571
1295	2756	101024.887233
1296	2757	72059.708609
1297	2758	88090.892202
1298	2759	151376.187730
1299	2760	132069.624440
1300	2761	143749.377606

1301	2762	143363.563916
1302	2763	183224.167044
1303	2764	166975.183665
1304	2765	270562.588959
1305	2766	132180.243031
1306	2767	83095.611824
1307	2768	138769.091878
1308	2769	133478.760237
1309	2770	155493.055312
1310	2771	115858.462673
1311	2772	120952.930458
1312	2773	145645.785504
1313	2774	133218.601428
1314	2775	111860.980952
1315	2776	142109.140016
1316	2777	153323.140683
1317	2778	126296.315071
1318	2779	137382.280783
1319	2780	99079.542143
1320	2781	99218.646103
1321	2782	96188.834980
1322	2783	104622.668433
1323	2784	127638.637128
1324	2785	135010.005472
1325	2786	80413.458786
1326	2787	121722.982415
1327	2788	80076.865084
1328	2789	183072.158559
1329	2790	90446.604580
1330	2791	110782.563497
1331	2792	66260.823710

1332	2793	165545.440050
1333	2794	105800.504428
1334	2795	114848.869321
1335	2796	99469.208755
1336	2797	191732.963886
1337	2798	110059.991420
1338	2799	111004.351926
1339	2800	79822.397495
1340	2801	112135.148774
1341	2802	140088.851416
1342	2803	166436.874598
1343	2804	151140.157198
1344	2805	106458.747976
1345	2806	85852.447328
1346	2807	161618.683749
1347	2808	148263.690743
1348	2809	137698.580262
1349	2810	135934.149222
1350	2811	162192.853713
1351	2812	144075.532885
1352	2813	176411.547371
1353	2814	167900.394699
1354	2815	102941.694881
1355	2816	223355.183109
1356	2817	153279.491193
1357	2818	134346.272503
1358	2819	184879.883815
1359	2820	135108.614761
1360	2821	100769.428695
1361	2822	190735.250371
1362	2823	226249.299484
1363	2824	178863.935650

1364	2825	157607.025989
1365	2826	131424.569711
1366	2827	136489.686972
1367	2828	224031.060327
1368	2829	202130.441377
1369	2830	235795.554918
1370	2831	183598.935732
1371	2832	246968.870803
1372	2833	304737.984603
1373	2834	210989.171318
1374	2835	207695.396828
1375	2836	187495.111599
1376	2837	164006.884718
1377	2838	143821.792286
1378	2839	178160.136132
1379	2840	193334.037724
1380	2841	203646.684602
1381	2842	219963.198816
1382	2843	145502.080201
1383	2844	139243.229353
1384	2845	119891.369739
1385	2846	205282.401587
1386	2847	191817.605914
1387	2848	219330.003836
1388	2849	201628.564364
1389	2850	287453.756854
1390	2851	236153.003061
1391	2852	223822.038215
1392	2853	230806.007405
1393	2854	142859.687416
1394	2855	201349.810671

1395	2856	202114.156817
1396	2857	187495.111599
1397	2858	210576.541508
1398	2859	125912.299923
1399	2860	128566.466755
1400	2861	130563.062325
1401	2862	195874.344536
1402	2863	132624.343778
1403	2864	250350.190115
1404	2865	140279.367259
1405	2866	149483.694981
1406	2867	100005.348742
1407	2868	104469.426224
1408	2869	105381.748515
1409	2870	138689.208714
1410	2871	91474.882384
1411	2872	51279.113382
1412	2873	108749.867332
1413	2874	128949.150407
1414	2875	110661.371033
1415	2876	170860.457861
1416	2877	140183.545417
1417	2878	184586.320687
1418	2879	139457.453874
1419	2880	98317.244197
1420	2881	166197.191030
1421	2882	167619.188677
1422	2883	194880.087035
1423	2884	207346.301791
1424	2885	188415.936454
1425	2886	223820.252844
1426	2887	99800.643271

1427	2888	137083.060890
1428	2889	57468.737035
1429	2890	86359.654738
1430	2891	138192.797032
1431	2892	69975.788043
1432	2893	98985.136419
1433	2894	73002.725122
1434	2895	294782.721766
1435	2896	289467.797250
1436	2897	196104.573949
1437	2898	154903.073955
1438	2899	229527.501384
1439	2900	152508.873443
1440	2901	221113.034781
1441	2902	180440.037247
1442	2903	341872.755824
1443	2904	374666.965953
1444	2905	94257.508999
1445	2906	210456.160406
1446	2907	117604.511717
1447	2908	131626.796280
1448	2909	154793.593428
1449	2910	90602.729709
1450	2911	79959.397309
1451	2912	143212.583712
1452	2913	87766.124948
1453	2914	75954.975933
1454	2915	82324.315442
1455	2916	84298.677180
1456	2917	167620.942720
1457	2918	123952.178061

1458	2919	217032.206363

In [88]:
```
y_pred.to_csv("\House Price Prediction\ML_Model\data_set\submission.csv",
```

In [89]:
```
#storing model in pickle file
import pickle

with open('gbr.pkl','wb') as f:
    pickle.dump(gbr,f)
```