In [32]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [33]:

```
df=pd.read_csv("house.csv")
df.head()
```

Out[33]:

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

5 rows × 81 columns

In [34]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             1460 non-null   int64  
 1   MSSubClass     1460 non-null   int64  
 2   MSZoning       1460 non-null   object 
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64  
 5   Street         1460 non-null   object 
 6   Alley          91 non-null     object 
 7   LotShape       1460 non-null   object 
 8   LandContour    1460 non-null   object 
 9   Utilities      1460 non-null   object 
 10  LotConfig      1460 non-null   object 
 11  LandSlope      1460 non-null   object 
 12  Neighborhood   1460 non-null   object 
 13  Condition1     1460 non-null   object 
 14  Condition2     1460 non-null   object 
 15  BldgType       1460 non-null   object 
 16  HouseStyle     1460 non-null   object 
 17  OverallQual    1460 non-null   int64  
 18  OverallCond    1460 non-null   int64  
 19  YearBuilt      1460 non-null   int64  
 20  YearRemodAdd   1460 non-null   int64  
 21  RoofStyle      1460 non-null   object 
 22  RoofMatl       1460 non-null   object 
 23  Exterior1st    1460 non-null   object 
 24  Exterior2nd    1460 non-null   object 
 25  MasVnrType     1452 non-null   object 
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object 
 28  ExterCond      1460 non-null   object 
 29  Foundation     1460 non-null   object 
 30  BsmtQual       1423 non-null   object 
 31  BsmtCond       1423 non-null   object 
 32  BsmtExposure   1422 non-null   object 
 33  BsmtFinType1   1423 non-null   object 
 34  BsmtFinSF1     1460 non-null   int64  
 35  BsmtFinType2   1422 non-null   object 
 36  BsmtFinSF2     1460 non-null   int64  
 37  BsmtUnfSF      1460 non-null   int64  
 38  TotalBsmtSF    1460 non-null   int64  
 39  Heating        1460 non-null   object 
 40  HeatingQC      1460 non-null   object 
 41  CentralAir     1460 non-null   object 
 42  Electrical     1459 non-null   object 
 43  1stFlrSF       1460 non-null   int64  
 44  2ndFlrSF       1460 non-null   int64  
 45  LowQualFinSF   1460 non-null   int64  
 46  GrLivArea      1460 non-null   int64  
 47  BsmtFullBath   1460 non-null   int64  
 48  BsmtHalfBath   1460 non-null   int64  
 49  FullBath       1460 non-null   int64  
 50  HalfBath       1460 non-null   int64  
 51  BedroomAbvGr   1460 non-null   int64  
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

In [35]:

```
df.describe()
```

Out[35]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	Ye
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.

8 rows × 38 columns

In [36]:

```
df.isna().sum()
```

Out[36]:

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

In [37]:

```
df=df[["Id","MSSubClass","MSZoning","LotFrontage","LotArea","LotShape","LandContour","Utili
```

In [38]:

```
df
```

Out[38]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	LotShape	LandContour	Utilities	P
0	1	60	RL	65.0	8450	Reg	Lvl	AllPub	
1	2	20	RL	80.0	9600	Reg	Lvl	AllPub	
2	3	60	RL	68.0	11250	IR1	Lvl	AllPub	
3	4	70	RL	60.0	9550	IR1	Lvl	AllPub	
4	5	60	RL	84.0	14260	IR1	Lvl	AllPub	
...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Reg	Lvl	AllPub	
1456	1457	20	RL	85.0	13175	Reg	Lvl	AllPub	
1457	1458	70	RL	66.0	9042	Reg	Lvl	AllPub	
1458	1459	20	RL	68.0	9717	Reg	Lvl	AllPub	
1459	1460	20	RL	75.0	9937	Reg	Lvl	AllPub	

1460 rows × 16 columns

In [ ]:

In [39]:

```
df.isna().sum()
```

Out[39]:

```
Id                0
MSSubClass        0
MSZoning          0
LotFrontage     259
LotArea           0
LotShape          0
LandContour       0
Utilities         0
PoolArea          0
MoSold            0
YrSold            0
SaleCondition     0
GarageArea        0
Street            0
HouseStyle        0
SalePrice         0
dtype: int64
```

In [40]:

```
amean=df["LotFrontage"].value_counts().mean()
amean
```

Out[40]:

```
10.918181818181818
```

In [41]:

```
df["LotFrontage"].fillna(amean,inplace=True)
```

In [42]:

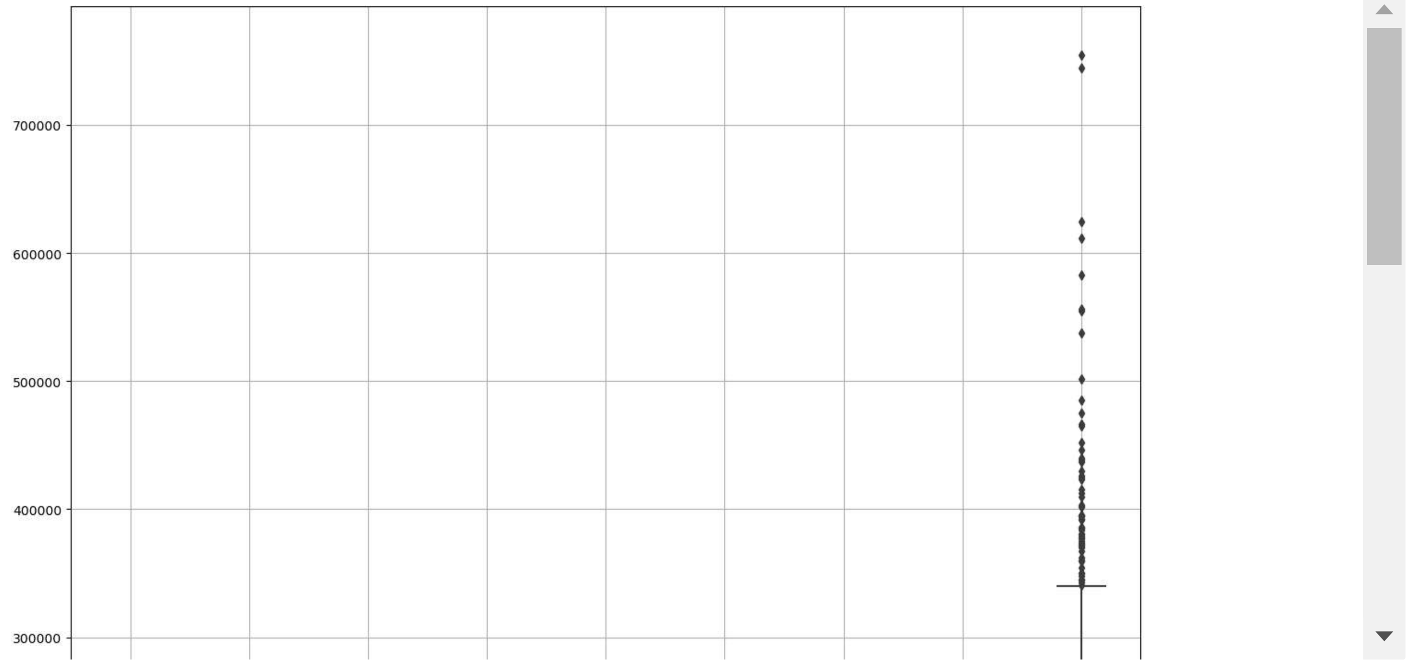
```
df["LotFrontage"].isna().sum()
```

Out[42]:

```
0
```

In [43]:

```
plt.figure(figsize=(15,15))
sns.boxplot(data=df)
plt.grid()
```



In [ ]:

In [ ]:

In [44]:

```
features=df.iloc[:,1:]
features.drop("PoolArea",axis=1,inplace=True)
features
```

Out[44]:

	MSSubClass	MSZoning	LotFrontage	LotArea	LotShape	LandContour	Utilities	MoSold
0	60	RL	65.0	8450	Reg	Lvl	AllPub	2
1	20	RL	80.0	9600	Reg	Lvl	AllPub	5
2	60	RL	68.0	11250	IR1	Lvl	AllPub	9
3	70	RL	60.0	9550	IR1	Lvl	AllPub	2
4	60	RL	84.0	14260	IR1	Lvl	AllPub	12
...	...	...	...	...	...	...	...	...
1455	60	RL	62.0	7917	Reg	Lvl	AllPub	8
1456	20	RL	85.0	13175	Reg	Lvl	AllPub	2
1457	70	RL	66.0	9042	Reg	Lvl	AllPub	5
1458	20	RL	68.0	9717	Reg	Lvl	AllPub	4
1459	20	RL	75.0	9937	Reg	Lvl	AllPub	6

1460 rows × 14 columns

In [ ]:

In [45]:

```
target=df.iloc[:,-1]
target
```

Out[45]:

```
0       208500
1       181500
2       223500
3       140000
4       250000
         ...
1455    175000
1456    210000
1457    266500
1458    142125
1459    147500
Name: SalePrice, Length: 1460, dtype: int64
```

In [46]:

```
from scipy.stats import skew
```

In [47]:

```
colname=features.select_dtypes(["int64","float64"]).columns
colname
```

Out[47]:

```
Index(['MSSubClass', 'LotFrontage', 'LotArea', 'MoSold', 'YrSold',
       'GarageArea', 'SalePrice'],
      dtype='object')
```

In [ ]:

In [48]:

```
features["MSSubClass"]=np.log(features["MSSubClass"])
```

In [49]:

```
features["LotFrontage"]=np.log(features["LotFrontage"])
```

In [ ]:

In [ ]:

In [50]:

```
features["MoSold"]=np.log(features["MoSold"])
```

In [51]:

```
features["YrSold"]=np.log(features["YrSold"])
```

In [ ]:

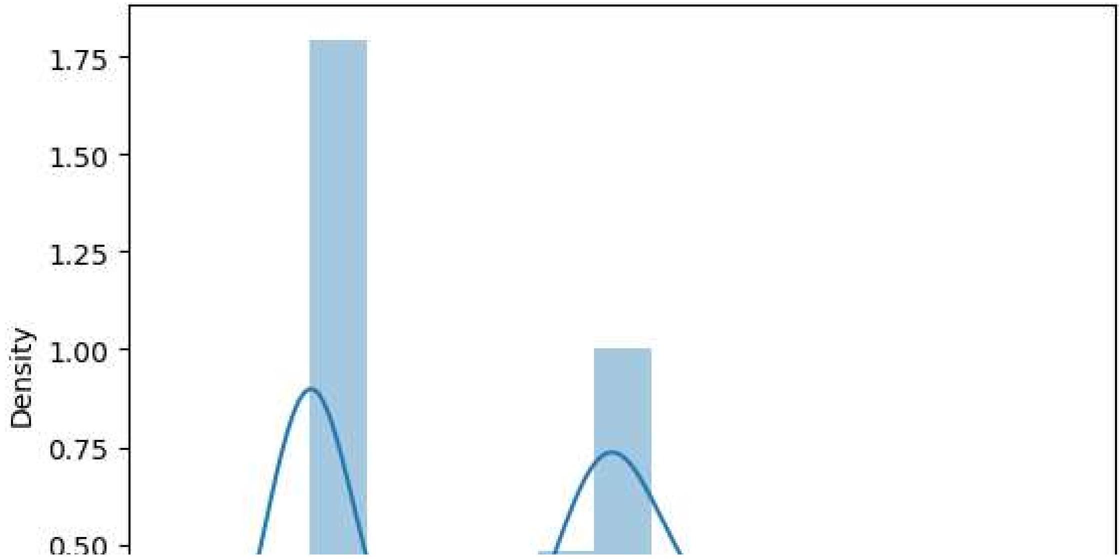
In [52]:

```
features["SalePrice"]=np.log(features["SalePrice"])
```

In [53]:

```
for i in features[colname]:
    print(i)
    print(skew(features[i]))
    plt.figure()
    sns.distplot(features[i])
    plt.show()
```

```
MSSubClass
0.23227194974210955
```



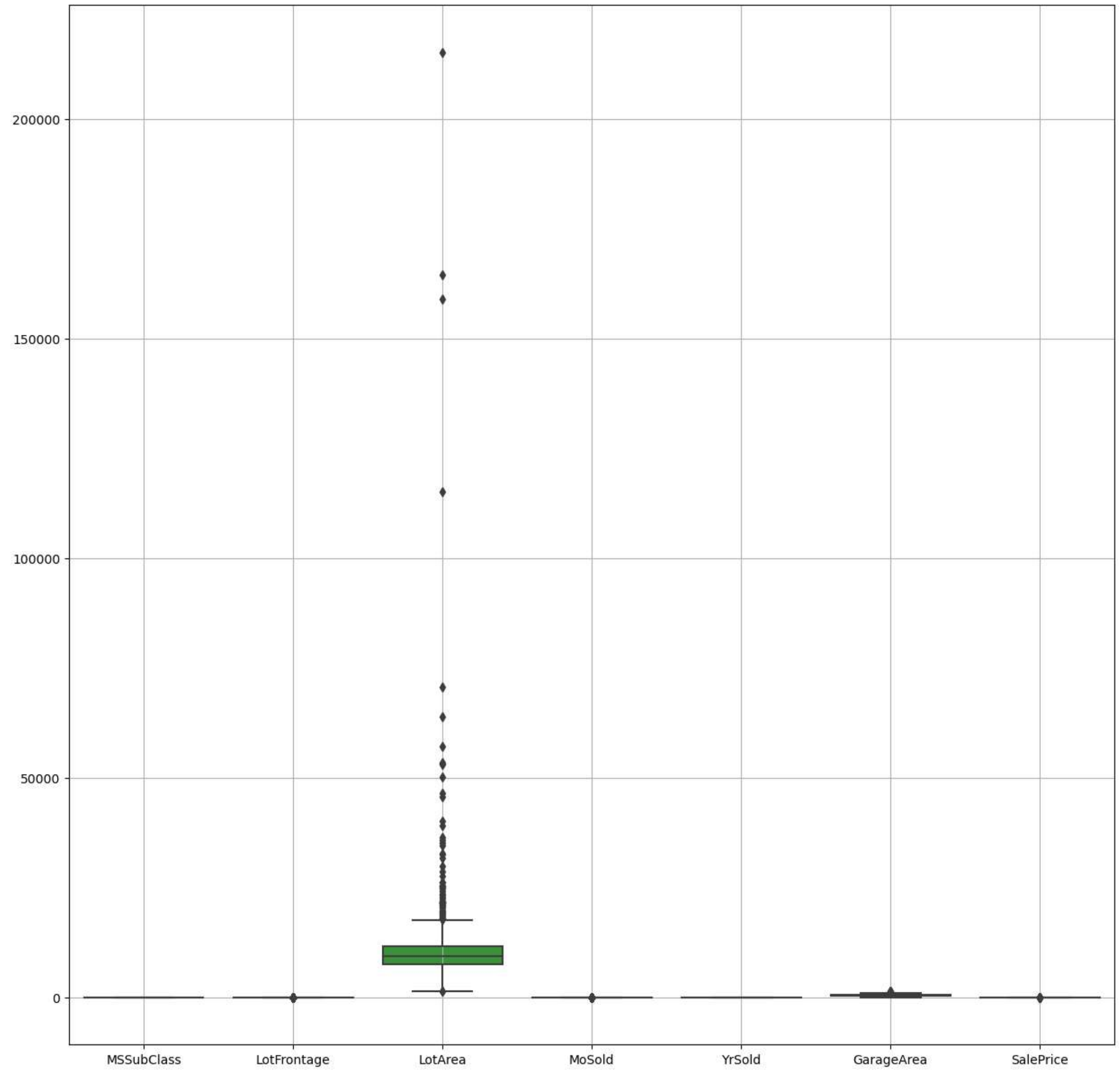
In [54]:

```
features.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   MSSubClass     1460 non-null   float64
 1   MSZoning       1460 non-null   object 
 2   LotFrontage    1460 non-null   float64
 3   LotArea        1460 non-null   int64  
 4   LotShape       1460 non-null   object 
 5   LandContour    1460 non-null   object 
 6   Utilities      1460 non-null   object 
 7   MoSold         1460 non-null   float64
 8   YrSold         1460 non-null   float64
 9   SaleCondition  1460 non-null   object 
 10  GarageArea     1460 non-null   int64  
 11  Street         1460 non-null   object 
 12  HouseStyle     1460 non-null   object 
 13  SalePrice      1460 non-null   float64
dtypes: float64(5), int64(2), object(7)
memory usage: 159.8+ KB
```

In [55]:

```
plt.figure(figsize=(15,15))
sns.boxplot(data=features)
plt.grid()
```



In [56]:

```
ca=features.select_dtypes("object").columns
ca
```

Out[56]:

```
Index(['MSZoning', 'LotShape', 'LandContour', 'Utilities', 'SaleCondition',
       'Street', 'HouseStyle'],
      dtype='object')
```

In [57]:

```
from sklearn.preprocessing import OrdinalEncoder
three=OrdinalEncoder()
features[ca]=three.fit_transform(features[ca])
```

In [58]:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(features,target,test_size=0.3,random_state=1)
```

In [59]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(xtrain,ytrain)
ypred=lr.predict(xtest)
ypred
```

Out[59]:

```
array([ 2.44323165e+05,  1.95455790e+05,  1.19214568e+05,  4.33749746e+04,
        1.46778715e+05,  3.06441844e+05,  2.85058254e+05,  1.43429170e+05,
        2.04744130e+05,  2.69211469e+05,  1.94643924e+05, -1.75466568e+04,
        1.88441445e+05,  3.45106265e+05,  2.43053968e+05,  6.40651834e+04,
        1.24992201e+05,  7.70616026e+04,  2.67106134e+05,  1.51434252e+05,
        1.22711857e+05,  1.27122488e+05,  2.50076222e+05,  3.19459075e+05,
        1.97284970e+04,  2.00441668e+05,  1.31359803e+05,  1.90915701e+05,
        3.87765340e+05,  1.19171347e+05,  9.51932597e+04,  1.13366377e+05,
        1.26872379e+05,  3.92450470e+04,  1.62067357e+05,  3.27766426e+05,
        1.35264730e+05,  4.83234367e+04,  2.68798907e+05,  1.01834220e+05,
        1.44249444e+05,  1.46295304e+05,  5.60381427e+04,  1.40950789e+05,
        1.87843159e+05,  1.74688592e+05,  1.12291975e+05,  1.73941312e+05,
        2.52535533e+05,  2.82674497e+05,  7.84047291e+04,  3.00517869e+05,
        1.29071295e+05,  2.48903932e+05,  2.11346687e+05,  1.14627487e+05,
        1.13425146e+05,  1.87668807e+05,  1.13551822e+05,  1.92184690e+05,
        1.87920968e+05,  2.71446091e+05,  3.40073312e+04,  1.36565030e+05,
        1.85122830e+05,  1.32350331e+05,  1.47255991e+05,  2.61091869e+05,
        1.75277134e+05,  1.54201146e+05,  2.25594861e+05,  1.05881389e+05,
```

In [60]:

```
train=lr.score(xtrain,ytrain)
test=lr.score(xtest,ytest)
```

In [61]:

```
train
```

Out[61]:

```
0.9156788946576503
```

In [62]:

```
test
```

Out[62]:

```
0.8699138713467953
```

In [ ]:

In [ ]: