In [91]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
# to visualise al the columns in the dataframe
pd.pandas.set_option('display.max_columns', None)
```

In [92]:

```
dataset=pd.read_csv('train.csv')
dataset.head()
```

Out[92]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighbor
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Co
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Ve
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Co
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Cr
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoI

Missing Values

In [93]:

```
## Let us capture all the nan values
## First lets handle Categorical features which are missing
features_nan=[feature for feature in dataset.columns if dataset[feature].isnull().sum()>1 and datas
et[feature].dtypes=='O']

for feature in features_nan:
    print("{}: {}% missing values".format(feature,np.round(dataset[feature].isnull().mean(),4)))
```

```
Alley: 0.9377% missing values
MasVnrType: 0.0055% missing values
BsmtQual: 0.0253% missing values
BsmtCond: 0.0253% missing values
BsmtExposure: 0.026% missing values
BsmtFinType1: 0.0253% missing values
BsmtFinType2: 0.026% missing values
FireplaceQu: 0.4726% missing values
GarageType: 0.0555% missing values
GarageFinish: 0.0555% missing values
GarageQual: 0.0555% missing values
GarageCond: 0.0555% missing values
PoolQC: 0.9952% missing values
Fence: 0.8075% missing values
MiscFeature: 0.963% missing values
```

In [94]:

```
## Replace missing value with a new label
def replace_cat_feature(dataset,features_nan):
    data=dataset.copy()
    data[features_nan]=data[features_nan].fillna('Missing')
    return data

dataset=replace_cat_feature(dataset,features_nan)

dataset[features_nan].isnull().sum()
```

Out[94]:

```
Alley           0
MasVnrType      0
BsmtQual        0
BsmtCond        0
BsmtExposure    0
BsmtFinType1    0
BsmtFinType2    0
FireplaceQu     0
GarageType      0
GarageFinish    0
GarageQual      0
GarageCond      0
PoolQC          0
Fence           0
MiscFeature     0
dtype: int64
```

In [95]:

```
dataset.head()
```

Out[95]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighb
0	1	60	RL	65.0	8450	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
1	2	20	RL	80.0	9600	Pave	Missing	Reg	Lvl	AllPub	FR2	Gtl	\
2	3	60	RL	68.0	11250	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
3	4	70	RL	60.0	9550	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	'
4	5	60	RL	84.0	14260	Pave	Missing	IR1	Lvl	AllPub	FR2	Gtl	N

In [96]:

```
## Now lets check for numerical variables the contains missing values
numerical_with_nan=[feature for feature in dataset.columns if dataset[feature].isnull().sum()>1 and
dataset[feature].dtypes!='O']

## We will print the numerical nan variables and percentage of missing values

for feature in numerical_with_nan:
    print("{}: {}% missing value".format(feature,np.around(dataset[feature].isnull().mean(),4)))
```

```
LotFrontage: 0.1774% missing value
MasVnrArea: 0.0055% missing value
GarageYrBlt: 0.0555% missing value
```

In [97]:

```
## Replacing the numerical Missing Values

for feature in numerical_with_nan:
    ## We will replace by using median since there are outliers
    median_value=dataset[feature].median()

    ## create a new feature to capture nan values
    dataset[feature+'nan']=np.where(dataset[feature].isnull(),1,0)
    dataset[feature].fillna(median_value,inplace=True)

dataset[numerical_with_nan].isnull().sum()
```

Out[97]:

```
LotFrontage    0
MasVnrArea     0
GarageYrBlt    0
dtype: int64
```

In [98]:

```
dataset.head(50)
```

Out[98]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neigh
0	1	60	RL	65.0	8450	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
1	2	20	RL	80.0	9600	Pave	Missing	Reg	Lvl	AllPub	FR2	Gtl	
2	3	60	RL	68.0	11250	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
3	4	70	RL	60.0	9550	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	
4	5	60	RL	84.0	14260	Pave	Missing	IR1	Lvl	AllPub	FR2	Gtl	
5	6	50	RL	85.0	14115	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
6	7	20	RL	75.0	10084	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
7	8	60	RL	69.0	10382	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	I
8	9	50	RM	51.0	6120	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
9	10	190	RL	50.0	7420	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
10	11	20	RL	70.0	11200	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
11	12	60	RL	85.0	11924	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
12	13	20	RL	69.0	12968	Pave	Missing	IR2	Lvl	AllPub	Inside	Gtl	
13	14	20	RL	91.0	10652	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
14	15	20	RL	69.0	10920	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	
15	16	45	RM	51.0	6120	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
16	17	20	RL	69.0	11241	Pave	Missing	IR1	Lvl	AllPub	CulDSac	Gtl	
17	18	90	RL	72.0	10791	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
18	19	20	RL	66.0	13695	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	S
19	20	20	RL	70.0	7560	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
20	21	60	RL	101.0	14215	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	
21	22	45	RM	57.0	7449	Pave	Grvl	Reg	Bnk	AllPub	Inside	Gtl	
22	23	20	RL	75.0	9742	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
23	24	120	RM	44.0	4224	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	N
24	25	20	RL	69.0	8246	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
25	26	20	RL	110.0	14230	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
26	27	20	RL	60.0	7200	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
27	28	20	RL	98.0	11478	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
28	29	20	RL	47.0	16321	Pave	Missing	IR1	Lvl	AllPub	CulDSac	Gtl	
29	30	30	RM	60.0	6324	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
30	31	70	C (all)	50.0	8500	Pave	Pave	Reg	Lvl	AllPub	Inside	Gtl	
31	32	20	RL	69.0	8544	Pave	Missing	IR1	Lvl	AllPub	CulDSac	Gtl	
32	33	20	RL	85.0	11049	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
33	34	20	RL	70.0	10552	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
34	35	120	RL	60.0	7313	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
35	36	60	RL	108.0	13418	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
36	37	20	RL	112.0	10859	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
37	38	20	RL	74.0	8532	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
38	39	20	RL	68.0	7922	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
39	40	90	RL	65.0	6040	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
40	41	20	RL	84.0	8658	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
41	42	20	RL	115.0	16905	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
42	43	85	RL	69.0	9180	Pave	Missing	IR1	Lvl	AllPub	CulDSac	Gtl	S
43	44	20	RL	69.0	9200	Pave	Missing	IR1	Lvl	AllPub	CulDSac	Gtl	
44	45	20	RL	70.0	7945	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neigh
45	46	120	RL	[illegible]	7658	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
46	47	50	RL	48.0	12822	Pave	Missing	IR1	Lvl	AllPub	CulDSac	Gtl	
47	48	20	FV	84.0	11096	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
48	49	190	RM	33.0	4456	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
49	50	20	RL	66.0	7742	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	

In [99]:

```
dataset[['YearBuilt','YearRemodAdd','GarageYrBlt']].head()
```

Out[99]:

	YearBuilt	YearRemodAdd	GarageYrBlt
0	2003	2003	2003.0
1	1976	1976	1976.0
2	2001	2002	2001.0
3	1915	1970	1998.0
4	2000	2000	2000.0

Numerical Variables

Since the numerical variables are skewed we will perform log normal distribution

In [100]:

```
dataset.head()
```

Out[100]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighb
0	1	60	RL	65.0	8450	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
1	2	20	RL	80.0	9600	Pave	Missing	Reg	Lvl	AllPub	FR2	Gtl	\
2	3	60	RL	68.0	11250	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
3	4	70	RL	60.0	9550	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	'
4	5	60	RL	84.0	14260	Pave	Missing	IR1	Lvl	AllPub	FR2	Gtl	N

In [101]:

```
import numpy as np
num_features=['LotFrontage', 'LotArea', '1stFlrSF', 'GrLivArea', 'SalePrice']

for feature in num_features:
    dataset[feature]=np.log(dataset[feature])
```

In [102]:

```
dataset.head()
```

Out[102]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighl
0	1	60	RL	4.174387	9.041922	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
1	2	20	RL	4.382027	9.169518	Pave	Missing	Reg	Lvl	AllPub	FR2	Gtl	
2	3	60	RL	4.219508	9.328123	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	

3	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighl
4	5	60	RL	4.430817	9.565214	Pave	Missing	IR1	Lvl	AllPub	FR2	Gtl	

Handling Rare Categorical Feature

We will remove categorical variables that are present less than 1% of the observations

In [103]:

```
categorical_features=[feature for feature in dataset.columns if dataset[feature].dtype=='O']
```

In [104]:

```
categorical_features
```

Out[104]:

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

In [105]:

```
for feature in categorical_features:
    temp=dataset.groupby(feature)['SalePrice'].count()/len(dataset)
    temp_df=temp[temp>0.01].index
    dataset[feature]=np.where(dataset[feature].isin(temp_df),dataset[feature],'Rare_var')
```

In [106]:

```
dataset.head(100)
```

Out[106]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Nei
0	1	60	RL	4.174387	9.041922	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
1	2	20	RL	4.382027	9.169518	Pave	Missing	Reg	Lvl	AllPub	FR2	Gtl	
2	3	60	RL	4.219508	9.328123	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
3	4	70	RL	4.094345	9.164296	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	
4	5	60	RL	4.430817	9.565214	Pave	Missing	IR1	Lvl	AllPub	FR2	Gtl	
5	6	50	RL	4.442651	9.554993	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
6	7	20	RL	4.317488	9.218705	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
7	8	60	RL	4.234107	9.247829	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	
8	9	50	RM	3.931826	8.719317	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
9	10	190	RL	3.912023	8.911934	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
10	11	20	RL	4.248495	9.323669	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
11	12	60	RL	4.442651	9.386308	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
12	13	20	RL	4.234107	9.470240	Pave	Missing	IR2	Lvl	AllPub	Inside	Gtl	
13	14	20	RL	4.510860	9.273503	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
14	15	20	RL	4.234107	9.298351	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	
15	16	45	RM	3.931826	8.719317	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
16	17	20	RL	4.234107	9.327323	Pave	Missing	IR1	Lvl	AllPub	CulDSac	Gtl	
17	18	90	RL	4.276666	9.286468	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
18	19	20	RL	4.189655	9.524786	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
19	20	20	RL	4.248495	8.930626	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
20	21	60	RL	4.615121	9.562053	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	
21	22	45	RM	4.043051	8.915835	Pave	Grvl	Reg	Bnk	AllPub	Inside	Gtl	
22	23	20	RL	4.317488	9.184202	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
23	24	120	RM	3.784190	8.348538	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
24	25	20	RL	4.234107	9.017484	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
25	26	20	RL	4.700480	9.563108	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
26	27	20	RL	4.094345	8.881836	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
27	28	20	RL	4.584967	9.348187	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
28	29	20	RL	3.850148	9.700208	Pave	Missing	IR1	Lvl	AllPub	CulDSac	Gtl	
29	30	30	RM	4.094345	8.752107	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
...	...	...	...	...	...	...	...	...	...	...	...	...	
70	71	20	RL	4.553877	9.521568	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
71	72	20	RL	4.234107	8.935772	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
72	73	60	RL	4.304065	9.224342	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	
73	74	20	RL	4.442651	9.230143	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
74	75	50	RM	4.094345	8.663888	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
75	76	180	RM	3.044522	7.375256	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
76	77	20	RL	4.234107	9.044876	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
77	78	50	RM	3.912023	9.063579	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
78	79	90	RL	4.276666	9.285262	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
79	80	50	RM	4.094345	9.253400	Pave	Grvl	Reg	Lvl	AllPub	Corner	Gtl	
80	81	60	RL	4.605170	9.472705	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
81	82	120	RM	3.465736	8.411833	Pave	Missing	Reg	Lvl	AllPub	FR2	Gtl	
82	83	20	RL	4.356709	9.230731	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
83	84	20	RL	4.382027	9.092907	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
84	85	80	RL	4.234107	9.051345	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neig
84	85	60	RL	4.234107	9.051345	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
85	86	60	RL	4.795791	9.684025	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
86	87	60	RL	4.804021	9.385218	Pave	Missing	IR2	Lvl	AllPub	Inside	Gtl	
87	88	160	FV	3.688879	8.281724	Pave	Pave	Reg	Lvl	AllPub	Corner	Gtl	
88	89	50	Rare_var	4.653960	9.044286	Pave	Missing	IR1	Lvl	AllPub	Corner	Gtl	
89	90	20	RL	4.094345	8.995909	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
90	91	20	RL	4.094345	8.881836	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
91	92	20	RL	4.442651	9.047821	Pave	Missing	Reg	Lvl	AllPub	Inside	Gtl	
92	93	30	RL	4.382027	9.500020	Pave	Grvl	IR1	HLS	AllPub	Inside	Gtl	
93	94	190	Rare_var	4.094345	8.881836	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
94	95	60	RL	4.234107	9.141740	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
95	96	60	RL	4.234107	9.186560	Pave	Missing	IR2	Lvl	AllPub	Corner	Gtl	
96	97	20	RL	4.356709	9.236398	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	
97	98	20	RL	4.290459	9.298443	Pave	Missing	Reg	HLS	AllPub	Inside	Gtl	
98	99	30	RL	4.442651	9.270965	Pave	Missing	Reg	Lvl	AllPub	Corner	Gtl	
99	100	20	RL	4.343805	9.139918	Pave	Missing	IR1	Lvl	AllPub	Inside	Gtl	

100 rows × 84 columns

In [107]:

```
for feature in categorical_features:
    labels_ordered=dataset.groupby([feature])['SalePrice'].mean().sort_values().index
    labels_ordered={k:i for i,k in enumerate(labels_ordered,0)}
    dataset[feature]=dataset[feature].map(labels_ordered)
```

In [108]:

```
dataset.head(10)
```

Out[108]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighbo
0	1	60	3	4.174387	9.041922	1	2	0	1	1	0	0	
1	2	20	3	4.382027	9.169518	1	2	0	1	1	2	0	
2	3	60	3	4.219508	9.328123	1	2	1	1	1	0	0	
3	4	70	3	4.094345	9.164296	1	2	1	1	1	1	0	
4	5	60	3	4.430817	9.565214	1	2	1	1	1	2	0	
5	6	50	3	4.442651	9.554993	1	2	1	1	1	0	0	
6	7	20	3	4.317488	9.218705	1	2	0	1	1	0	0	
7	8	60	3	4.234107	9.247829	1	2	1	1	1	1	0	
8	9	50	1	3.931826	8.719317	1	2	0	1	1	0	0	
9	10	190	3	3.912023	8.911934	1	2	0	1	1	1	0	

Feature Scaling

In [109]:

```
feature_scale=[feature for feature in dataset.columns if feature not in ['Id','SalePrice']]

from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(dataset[feature_scale])
```

Out[109]:

```
MinMaxScaler(copy=True, feature_range=(0, 1))
```

In [110]:

```
scaler.transform(dataset[feature_scale])
```

Out[110]:

```
array([[0.23529412, 0.75      , 0.41820812, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.75      , 0.49506375, ..., 0.        , 0.        ,
        0.        ],
       [0.23529412, 0.75      , 0.434909  , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.29411765, 0.75      , 0.42385922, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.75      , 0.434909  , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.75      , 0.47117546, ..., 0.        , 0.        ,
        0.        ]])
```

In [111]:

```
# transform the train and test set, and add on the Id and SalePrice variables
data = pd.concat([dataset[['Id', 'SalePrice']].reset_index(drop=True),
                    pd.DataFrame(scaler.transform(dataset[feature_scale]), columns=feature_scale)],
                    axis=1)
```

In [112]:

```
data.head()
```

Out[112]:

	Id	SalePrice	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlop
0	1	12.247694	0.235294	0.75	0.418208	0.366344	1.0	1.0	0.000000	0.333333	1.0	0.00	0.
1	2	12.109011	0.000000	0.75	0.495064	0.391317	1.0	1.0	0.000000	0.333333	1.0	0.50	0.
2	3	12.317167	0.235294	0.75	0.434909	0.422359	1.0	1.0	0.333333	0.333333	1.0	0.00	0.
3	4	11.849398	0.294118	0.75	0.388581	0.390295	1.0	1.0	0.333333	0.333333	1.0	0.25	0.
4	5	12.429216	0.235294	0.75	0.513123	0.468761	1.0	1.0	0.333333	0.333333	1.0	0.50	0.

In [113]:

```
data.shape
```

Out[113]:

```
(1460, 84)
```

In [114]:

```
data.to_csv('train_data.csv',index=False)
```

In [115]:

```
data.head()
```

Out[115]:

	Id	SalePrice	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlop
0	1	12.247694	0.235294	0.75	0.418208	0.366344	1.0	1.0	0.000000	0.333333	1.0	0.00	0.
1	2	12.109011	0.000000	0.75	0.495064	0.391317	1.0	1.0	0.000000	0.333333	1.0	0.50	0.
2	3	12.317167	0.235294	0.75	0.434909	0.422359	1.0	1.0	0.333333	0.333333	1.0	0.00	0.
3	4	11.849398	0.294118	0.75	0.388581	0.390295	1.0	1.0	0.333333	0.333333	1.0	0.25	0.

	Id	SalePrice	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlop
4	5	12.429216	0.235294	0.75	0.513123	0.468761	1.0	1.0	0.333333	0.333333	1.0	0.50	0.

In [116]:

```
## Capture the dependent feature
y_train=data[['SalePrice']]
```

In [117]:

```
## drop dependent feature from dataset
X_train=data.drop(['Id','SalePrice'],axis=1)
```

In [118]:

```
from sklearn.model_selection import train_test_split
# Use train_test_split from sci-kit learn to segment our data into train and a local testset
X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.2)
```

In [119]:

```
X_train.shape
```

Out[119]:

```
(1168, 82)
```

In [120]:

```
#Train the model
from sklearn import linear_model
model = linear_model.LinearRegression()
```

In [121]:

```
#Fit the model
model.fit(X_train, y_train)
```

Out[121]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [122]:

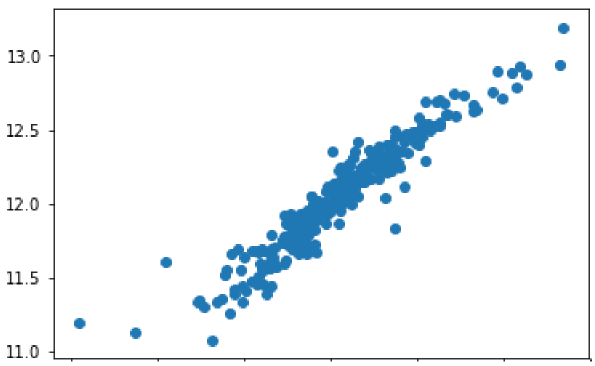
```
y_pred = model.predict(X_test)
```

In [123]:

```
plt.scatter(y_test, y_pred)
```

Out[123]:

```
<matplotlib.collections.PathCollection at 0x1f64757bcc0>
```

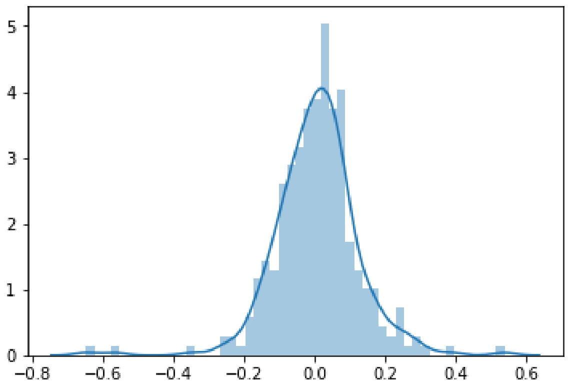


In [124]:

```
sns.distplot((y_test - y_pred), bins=50)
```

Out[124]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f645036fd0>
```



In [125]:

```
# Import Sci-Kit Learn
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import Normalizer
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor, GradientBoostingRegressor,
BaggingRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.model_selection import RandomizedSearchCV, cross_val_score, StratifiedKFold,
learning_curve, KFold

# Ensemble Models
from xgboost import XGBRegressor
```

In [126]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.2, random_state=0
)
```

In [127]:

```
def rmse(y_test, y_pred):
    return np.sqrt(mean_squared_error(np.log(y_test), np.log(y_pred)))
```

Linear Regression

In [128]:

```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression

lin_regressor=LinearRegression()
#Fit the model
lin_regressor.fit(X_train, y_train)
mse=cross_val_score(lin_regressor,X_train,y_train,scoring='neg_mean_squared_error',cv=5)

mean_mse=np.mean(mse)
print(mean_mse)
```

-1.1950389389358715e+23

In [129]:

```
prediction_linear=lin_regressor.predict(X_test)
```

In [130]:

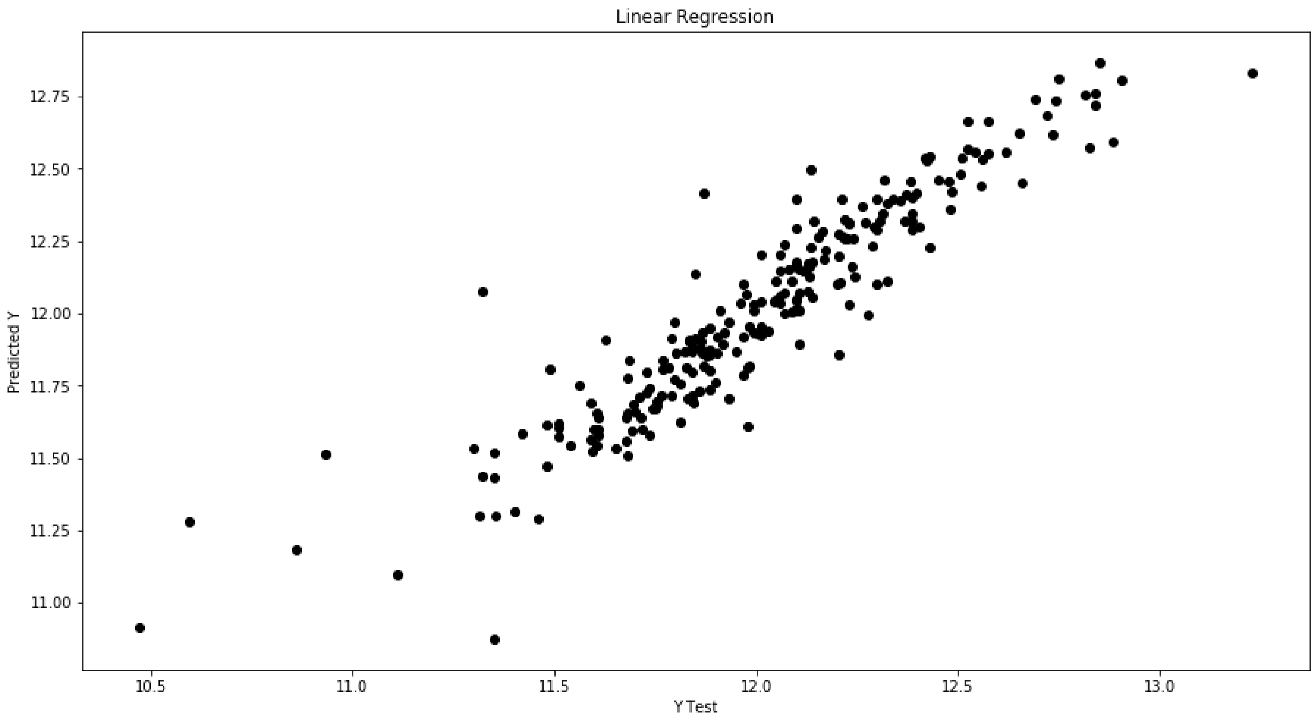
```
rmse(y_test, prediction_linear)
```

Out[130]:

0.012680858287713277

In [131]:

```
plt.figure(figsize=(15,8))
plt.scatter(y_test,prediction_linear, c= 'black')
plt.title("Linear Regression")
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
plt.show()
```



Ridge Regression

In [132]:

```
from sklearn.linear_model import Ridge
from sklearn.model_selection import GridSearchCV

ridge=Ridge()
parameters={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-2,1,5,10,20,30,35,40,45,50,55,100]}
ridge_regressor=GridSearchCV(ridge,parameters,scoring='neg_mean_squared_error',cv=5)
ridge_regressor.fit(X_train,y_train)
```

```
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\ridge.py:147: LinAlgWarning: Ill-conditioned matrix (rcond=2.80827e-18): result may not be accurate.
  overwrite_a=True).T
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\ridge.py:147: LinAlgWarning: Ill-conditioned matrix (rcond=6.93441e-19): result may not be accurate.
  overwrite_a=True).T
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\ridge.py:147: LinAlgWarning: Ill-
```

```
conditioned matrix (rcond=2.09046e-18): result may not be accurate.
  overwrite_a=True).T
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\ridge.py:147: LinAlgWarning: Ill-
conditioned matrix (rcond=3.55387e-19): result may not be accurate.
  overwrite_a=True).T
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\ridge.py:147: LinAlgWarning: Ill-
conditioned matrix (rcond=8.30857e-19): result may not be accurate.
  overwrite_a=True).T
```

Out[132]:

```
GridSearchCV(cv=5, error_score='raise-deprecating',
             estimator=Ridge(alpha=1.0, copy_X=True, fit_intercept=True,
                             max_iter=None, normalize=False, random_state=None,
                             solver='auto', tol=0.001),
             iid='warn', n_jobs=None,
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.001, 0.01, 1, 5, 10,
                                   20, 30, 35, 40, 45, 50, 55, 100]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring='neg_mean_squared_error', verbose=0)
```

In [133]:

```
print(ridge_regressor.best_params_)
print(ridge_regressor.best_score_)
```

```
{'alpha': 5}
-0.019425374387872268
```

In [134]:

```
prediction_ridge=ridge_regressor.predict(X_test)
```

In [135]:

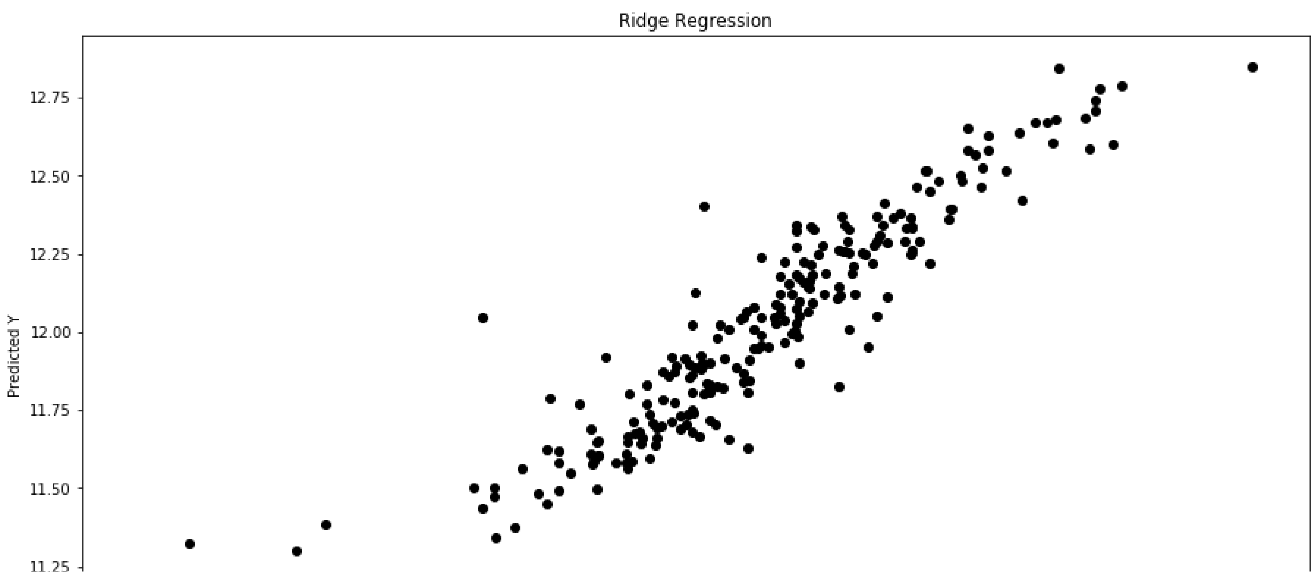
```
rmse(y_test, prediction_ridge)
```

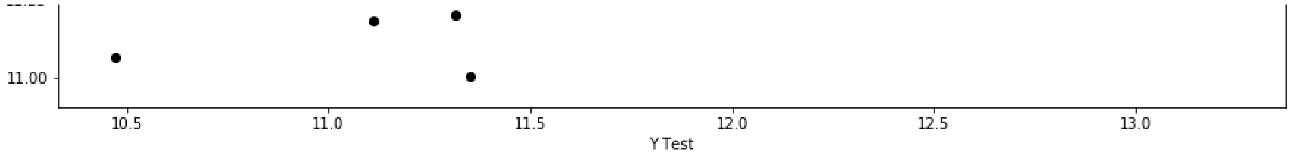
Out[135]:

```
0.012715320940289439
```

In [136]:

```
plt.figure(figsize=(15,8))
plt.scatter(y_test,prediction_ridge, c= 'black')
plt.title("Ridge Regression")
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
plt.show()
```





Lasso Regression

In [137]:

```
from sklearn.linear_model import Lasso
from sklearn.model_selection import GridSearchCV
lasso=Lasso()
parameters={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-2,1,5,10,20,30,35,40,45,50,55,100]}
lasso_regressor=GridSearchCV(lasso,parameters,scoring='neg_mean_squared_error',cv=5)

lasso_regressor.fit(X_train,y_train)
print(lasso_regressor.best_params_)
print(lasso_regressor.best_score_)
```

```
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 3.825267438677306, tolerance: 0.011599733802408117
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 4.348709008552711, tolerance: 0.01223592930751219
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 5.0109512403186045, tolerance: 0.012156869581577657
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 4.835254512518043, tolerance: 0.01166143140395687
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 4.039308544691328, tolerance: 0.011865383041266359
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 4.015931897939252, tolerance: 0.01223592930751219
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 4.371718451307496, tolerance: 0.012156869581577657
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 4.14447886107545, tolerance: 0.01166143140395687
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 4.036638041351252, tolerance: 0.011865383041266359
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 0.16943950479299552, tolerance: 0.01223592930751219
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 1.3788142512888957, tolerance: 0.012156869581577657
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 1.3872264500798632, tolerance: 0.01166143140395687
  positive)
C:\Users\Mahesh\Anaconda3\lib\site-packages\sklearn\linear_model\coordinate_descent.py:475:
ConvergenceWarning: Objective did not converge. You might want to increase the number of
iterations. Duality gap: 3.784770447081856, tolerance: 0.011865383041266359
  positive)
```

```
{'alpha': 0.001}
-0.0176239606389574
```

In [138]:

```python
prediction_lasso=lasso_regressor.predict(X_test)
```

In [139]:

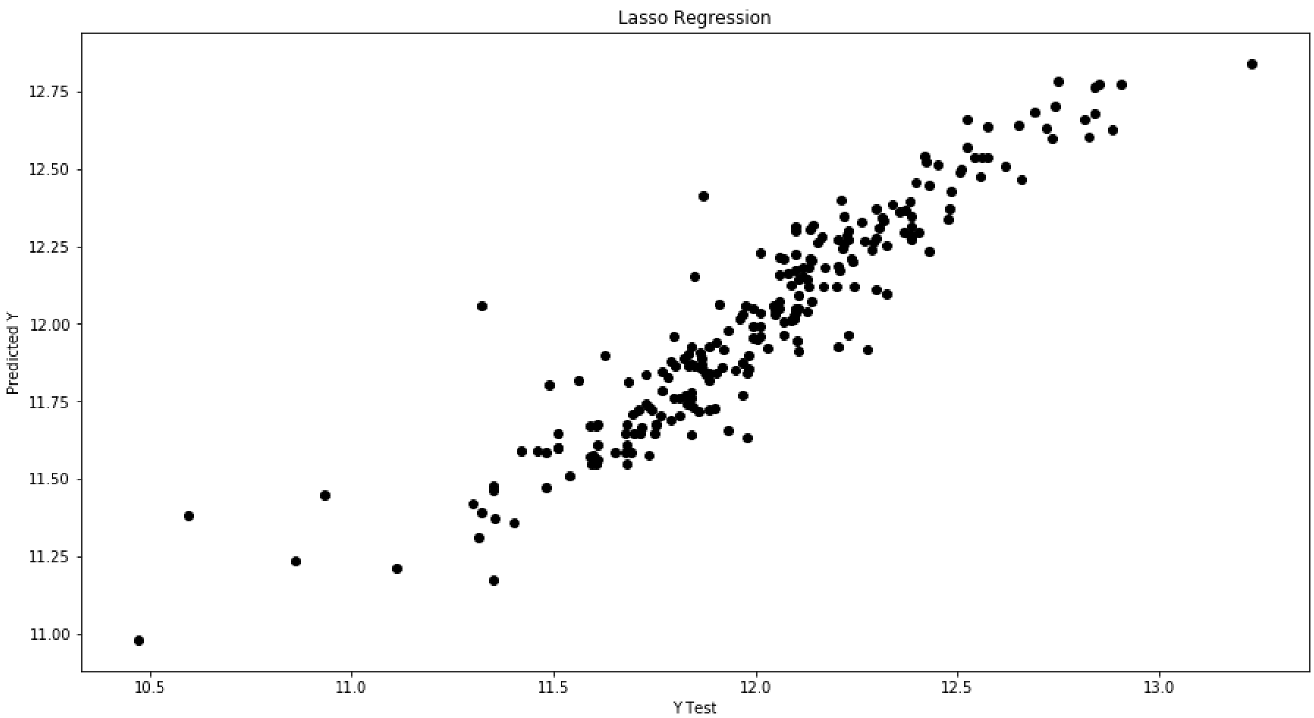
```python
rmse(y_test, prediction_lasso)
```

Out[139]:

```
0.012434517730068081
```

In [140]:

```python
plt.figure(figsize=(15,8))
plt.scatter(y_test,prediction_lasso, c= 'black')
plt.title("Lasso Regression")
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
plt.show()
```



In [ ]: