In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
pd.pandas.set_option('display.max_columns', None)
```

In [2]:
```
house_data = pd.read_csv('train.csv')
house_data.head()
```

Out[2]:

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

In [3]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(house_data, house_data['SalePrice'], test_size=0.1, random_state=
X_train.shape, X_test.shape
```

Out[3]:
```
((1314, 81), (146, 81))
```

In [4]:
```
cat_features_with_nan = [feature for feature in house_data.columns if house_data[feature].isnull().sum() > 1 and hous
for feature in cat_features_with_nan:
    print("{}: {}% missing values".format(feature, np.round(house_data[feature].isnull().mean(), 4)))
```

```
Alley: 0.9377% missing values
MasVnrType: 0.5973% 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 [5]:
```
def fill_missing_categorical(dataframe, features):
    df_copy = dataframe.copy()
    df_copy[features] = df_copy[features].fillna('Missing')
    return df_copy
```

In [6]:
```
house_data = fill_missing_categorical(house_data, cat_features_with_nan)
house_data[cat_features_with_nan].isnull().sum()
house_data.head()
```

Out[6]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
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

In [8]:
```
num_features_with_nan = [feature for feature in house_data.columns if house_data[feature].isnull().sum() > 1 and hous
for feature in num_features_with_nan:
    print("{}: {}% missing value".format(feature, np.around(house_data[feature].isnull().mean(), 4)))
```

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

In [9]:
```
for feature in num_features_with_nan:
    median_val = house_data[feature].median()
    house_data[feature + '_nan'] = np.where(house_data[feature].isnull(), 1, 0)
    house_data.loc[:, feature] = house_data[feature].fillna(median_val)

print(house_data[num_features_with_nan].isnull().sum())
house_data.head(10)
```

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

Out[9]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
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
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

In [10]:
```
for feature in ['YearBuilt', 'YearRemodAdd', 'GarageYrBlt']:
    house_data[feature] = house_data['YrSold'] - house_data[feature]
```

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

Out[10]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
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

In [11]:

```
log_transform_features = ['LotFrontage', 'LotArea', '1stFlrSF', 'GrLivArea', 'SalePrice']
for feature in log_transform_features:
    house_data[feature] = np.log(house_data[feature])

house_data.head()
```

Out[11]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
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

In [12]:

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

```
house_data.head(10)
```

Out[12]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
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

In [13]:

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

house_data.head(10)
```

Out[13]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	N
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	

In [14]:

```
features_to_scale = [feature for feature in house_data.columns if feature not in ['Id', 'SalePrice']]
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(house_data[features_to_scale])
scaled_data = pd.concat([house_data[['Id', 'SalePrice']].reset_index(drop=True),
                         pd.DataFrame(scaler.transform(house_data[features_to_scale]), columns=features_to_scale)],
                        axis=1)
scaled_data.to_csv('X_train.csv', index=False)
```