In [223]:
```
import pandas as pd
import numpy as np
import seaborn as sns
sns.set(style="ticks", color_codes=True)
import matplotlib.pyplot as plt
%matplotlib inline
import tensorflow as tf
from plotly import __version__
from plotly.offline import download_plotlyjs,init_notebook_mode,iplot,plot
import cufflinks as cf
init_notebook_mode(connected=True)
cf.go_offline()
```

In [224]:
```
pd.options.display.float_format = '{:.2f}'.format
```

In [225]:
```
df_train = pd.read_csv('train.csv')
df_test = pd.read_csv('test.csv')
```

In [226]:
```
df_train.head(2)
```

Out[226]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	Sal
0	1	60	RL	65.00	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	
1	2	20	RL	80.00	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	

2 rows × 81 columns

In [227]:
```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
Id               1460 non-null int64
MSSubClass       1460 non-null int64
MSZoning         1460 non-null object
LotFrontage      1201 non-null float64
LotArea          1460 non-null int64
Street           1460 non-null object
Alley            91 non-null object
LotShape         1460 non-null object
LandContour      1460 non-null object
Utilities        1460 non-null object
LotConfig        1460 non-null object
LandSlope        1460 non-null object
Neighborhood     1460 non-null object
Condition1       1460 non-null object
Condition2       1460 non-null object
BldgType         1460 non-null object
HouseStyle       1460 non-null object
OverallQual      1460 non-null int64
OverallCond      1460 non-null int64
YearBuilt        1460 non-null int64
YearRemodAdd     1460 non-null int64
RoofStyle        1460 non-null object
RoofMatl         1460 non-null object
Exterior1st      1460 non-null object
Exterior2nd      1460 non-null object
MasVnrType       1452 non-null object
MasVnrArea       1452 non-null float64
ExterQual        1460 non-null object
ExterCond        1460 non-null object
Foundation       1460 non-null object
BsmtQual         1423 non-null object
BsmtCond         1423 non-null object
BsmtExposure     1422 non-null object
BsmtFinType1     1423 non-null object
BsmtFinSF1       1460 non-null int64
BsmtFinType2     1422 non-null object
BsmtFinSF2       1460 non-null int64
BsmtUnfSF        1460 non-null int64
TotalBsmtSF      1460 non-null int64
Heating          1460 non-null object
HeatingQC        1460 non-null object
CentralAir       1460 non-null object
Electrical       1459 non-null object
1stFlrSF         1460 non-null int64
2ndFlrSF         1460 non-null int64
LowQualFinSF     1460 non-null int64
GrLivArea        1460 non-null int64
BsmtFullBath     1460 non-null int64
BsmtHalfBath     1460 non-null int64
FullBath         1460 non-null int64
HalfBath         1460 non-null int64
BedroomAbvGr     1460 non-null int64
KitchenAbvGr     1460 non-null int64
KitchenQual      1460 non-null object
TotRmsAbvGrd     1460 non-null int64
Functional       1460 non-null object
Fireplaces       1460 non-null int64
FireplaceQu      770 non-null object
GarageType       1379 non-null object
GarageYrBlt      1379 non-null float64
GarageFinish     1379 non-null object
GarageCars       1460 non-null int64
GarageArea       1460 non-null int64
GarageQual       1379 non-null object
GarageCond       1379 non-null object
PavedDrive       1460 non-null object
WoodDeckSF       1460 non-null int64
OpenPorchSF      1460 non-null int64
EnclosedPorch    1460 non-null int64
3SsnPorch        1460 non-null int64
ScreenPorch      1460 non-null int64
PoolArea         1460 non-null int64
PoolQC           7 non-null object
Fence            281 non-null object
MiscFeature      54 non-null object
MiscVal          1460 non-null int64
MoSold           1460 non-null int64
YrSold           1460 non-null int64
SaleType         1460 non-null object
SaleCondition    1460 non-null object
SalePrice        1460 non-null int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

Function to find missing value

In [228]:
```
def findMissingValue(df):
    data = pd.DataFrame(columns=['Columns','Number_Of_MValue','Percentage_Of_MValue'])

    data_col = []
    data_mValue = []
    data_perMValu = []

    for fn in df.columns:
        targetNum = len(df)
        x= df[fn].describe()[0]
        if x !=targetNum:
            missingValue = targetNum-x
            percentOfMV = round(float((missingValue/targetNum)*100),2)
            percentOfMV= str(percentOfMV) + ' '+ '%'

            data_col.append(fn)
            data_mValue.append(missingValue)
            data_perMValu.append(percentOfMV)

    data['Columns']=data_col
    data['Number_Of_MValue']=data_mValue
    data['Percentage_Of_MValue']=data_perMValu

    return data.sort_values('Number_Of_MValue',ascending=False).reset_index(drop=True)
```

In [229]:
```
findMissingValue(df_train)
```

Out[229]:

	Columns	Number_Of_MValue	Percentage_Of_MValue
0	PoolQC	1453.00	99.52 %
1	MiscFeature	1406.00	96.3 %
2	Alley	1369.00	93.77 %
3	Fence	1179.00	80.75 %
4	FireplaceQu	690.00	47.26 %
5	LotFrontage	259.00	17.74 %
6	GarageType	81.00	5.55 %
7	GarageYrBlt	81.00	5.55 %
8	GarageFinish	81.00	5.55 %
9	GarageQual	81.00	5.55 %
10	GarageCond	81.00	5.55 %
11	BsmtExposure	38.00	2.6 %
12	BsmtFinType2	38.00	2.6 %
13	BsmtFinType1	37.00	2.53 %
14	BsmtCond	37.00	2.53 %
15	BsmtQual	37.00	2.53 %
16	MasVnrArea	8.00	0.55 %
17	MasVnrType	8.00	0.55 %
18	Electrical	1.00	0.07 %

In [230]:
```
findMissingValue(df_test)
```

Out[230]:

	Columns	Number_Of_MValue	Percentage_Of_MValue
0	PoolQC	1456.00	99.79 %
1	MiscFeature	1408.00	96.5 %
2	Alley	1352.00	92.67 %
3	Fence	1169.00	80.12 %
4	FireplaceQu	730.00	50.03 %
5	LotFrontage	227.00	15.56 %
6	GarageCond	78.00	5.35 %
7	GarageYrBlt	78.00	5.35 %
8	GarageQual	78.00	5.35 %
9	GarageFinish	78.00	5.35 %
10	GarageType	76.00	5.21 %
11	BsmtCond	45.00	3.08 %
12	BsmtExposure	44.00	3.02 %
13	BsmtQual	44.00	3.02 %
14	BsmtFinType1	42.00	2.88 %
15	BsmtFinType2	42.00	2.88 %
16	MasVnrType	16.00	1.1 %
17	MasVnrArea	15.00	1.03 %
18	MSZoning	4.00	0.27 %
19	BsmtFullBath	2.00	0.14 %
20	BsmtHalfBath	2.00	0.14 %
21	Functional	2.00	0.14 %
22	Utilities	2.00	0.14 %
23	GarageCars	1.00	0.07 %
24	GarageArea	1.00	0.07 %
25	TotalBsmtSF	1.00	0.07 %
26	KitchenQual	1.00	0.07 %
27	BsmtUnfSF	1.00	0.07 %
28	BsmtFinSF2	1.00	0.07 %
29	BsmtFinSF1	1.00	0.07 %
30	Exterior2nd	1.00	0.07 %
31	Exterior1st	1.00	0.07 %
32	SaleType	1.00	0.07 %

Dealing With Missing Values:

Droping all the columns that have more than 80% missing values

In [231]:
```
df_train.drop(columns=['Alley','PoolQC','Fence','MiscFeature'],axis=1,inplace=True)
df_test.drop(columns=['Alley','PoolQC','Fence','MiscFeature'],axis=1,inplace=True)
```

Filling up FireplaceQu

In [232]:
```
for i in df_train.columns:
    if 'qu' in i.lower():
        print(i)
```

```
OverallQual
ExterQual
BsmtQual
LowQualFinSF
KitchenQual
FireplaceQu
GarageQual
```

Creating a function to convert all the rating into number from 0 to 6 for all the quality criteria

In [233]:
```
def convert_qual_to_num(df):
    for i in df[['FireplaceQu','OverallQual','ExterQual','BsmtQual','LowQualFinSF','KitchenQual','GarageQual']]:
        df[i].mask(df[i] == 'Ex', 5, inplace=True)
        df[i].mask(df[i] == 'Gd', 4, inplace=True)
        df[i].mask(df[i] == 'TA', 3, inplace=True)
        df[i].mask(df[i] == 'Fa', 2, inplace=True)
        df[i].mask(df[i] == 'Po', 1, inplace=True)
        df[i].mask(df[i] == 'NA', 0, inplace=True)
```

In [234]:
```
convert_qual_to_num(df_train)
convert_qual_to_num(df_test)
```

```
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\core\ops\__init__.py:1115: FutureWarning:

elementwise comparison failed; returning scalar instead, but in the future will perform elementwise comparison
```

Tried to find the mean for each rating point based on all the quality criteria

In [235]:
```
for i in range(5):
    mean_value = df_train[df_train['FireplaceQu']==(i+1)][['OverallQual','ExterQual','BsmtQual','KitchenQual','GarageQual']].sum(axis=1).unique().mean()
    print('The mean value for rating '+str(i+1)+' is - '+str(round(mean_value,2)))
```

```
The mean value for rating 1 is - 15.14
The mean value for rating 2 is - 20.0
The mean value for rating 3 is - 20.36
The mean value for rating 4 is - 19.5
The mean value for rating 5 is - 23.0
```

Created a function to fillup the missing value based on mean value of all other quality criteria

Here I considered an assumption based on the mean value I got above. However, there are some discrepancies but I tried to put my best guess here.

In [236]:
```
def fillup_mValue_fireplace(df):

    mask = (df['FireplaceQu'].isna()) & df[['OverallQual','ExterQual','BsmtQual','KitchenQual','GarageQual']].sum(axis=1)>=23
    df['FireplaceQu'][mask] = 5
    mask = (df['FireplaceQu'].isna()) & df[['OverallQual','ExterQual','BsmtQual','KitchenQual','GarageQual']].sum(axis=1)>=20
    df['FireplaceQu'][mask] = 4
    mask = (df['FireplaceQu'].isna()) & df[['OverallQual','ExterQual','BsmtQual','KitchenQual','GarageQual']].sum(axis=1)>=17
    df['FireplaceQu'][mask] = 3
    mask = (df['FireplaceQu'].isna()) & df[['OverallQual','ExterQual','BsmtQual','KitchenQual','GarageQual']].sum(axis=1)>=15
    df['FireplaceQu'][mask] = 2
    mask = (df['FireplaceQu'].isna()) & df[['OverallQual','ExterQual','BsmtQual','KitchenQual','GarageQual']].sum(axis=1)>=5
    df['FireplaceQu'][mask] = 1
    mask = (df['FireplaceQu'].isna()) & df[['OverallQual','ExterQual','BsmtQual','KitchenQual','GarageQual']].sum(axis=1)<5
    df['FireplaceQu'][mask] = 0
```

In [237]:
```
fillup_mValue_fireplace(df_train)
fillup_mValue_fireplace(df_test)
```

Filling up LotFrontage

Here I am going to use mean value to fillup the missing values

In [238]:
```
df_train[df_train['LotFrontage'].notna()]['LotFrontage'].sum()/len(df_train[df_train['LotFrontage'].notna()])
```

Out[238]: 70.04995836802665

In [239]:
```
df_test[df_test['LotFrontage'].notna()]['LotFrontage'].sum()/len(df_test[df_test['LotFrontage'].notna()])
```

Out[239]: 68.58035714285714

In [240]:
```
df_train['LotFrontage'].fillna(value=70,axis=0,inplace=True)
df_test['LotFrontage'].fillna(value=70,axis=0,inplace=True)
```

In [241]:
```
findMissingValue(df_train)
```

Out[241]:

	Columns	Number_Of_MValue	Percentage_Of_MValue
0	GarageType	81.00	5.55 %
1	GarageYrBlt	81.00	5.55 %
2	GarageFinish	81.00	5.55 %
3	GarageQual	81.00	5.55 %
4	GarageCond	81.00	5.55 %
5	BsmtExposure	38.00	2.6 %
6	BsmtFinType2	38.00	2.6 %
7	BsmtQual	37.00	2.53 %
8	BsmtCond	37.00	2.53 %
9	BsmtFinType1	37.00	2.53 %
10	MasVnrType	8.00	0.55 %
11	MasVnrArea	8.00	0.55 %
12	Electrical	1.00	0.07 %

Filling up GarageYrBlt

Since YrBlt and GarageYrBlt same for all, I am considering YrBlt as GarageYrBlt

In [242]:
```
df_train['GarageYrBlt'].fillna(value=df_train['YearBuilt'],axis=0,inplace=True)
df_test['GarageYrBlt'].fillna(value=df_test['YearBuilt'],axis=0,inplace=True)
```

In [243]:
```
findMissingValue(df_train)
```

Out[243]:

	Columns	Number_Of_MValue	Percentage_Of_MValue
0	GarageType	81.00	5.55 %
1	GarageFinish	81.00	5.55 %
2	GarageQual	81.00	5.55 %
3	GarageCond	81.00	5.55 %
4	BsmtExposure	38.00	2.6 %
5	BsmtFinType2	38.00	2.6 %
6	BsmtQual	37.00	2.53 %
7	BsmtCond	37.00	2.53 %
8	BsmtFinType1	37.00	2.53 %
9	MasVnrType	8.00	0.55 %
10	MasVnrArea	8.00	0.55 %
11	Electrical	1.00	0.07 %

Filling up all other missing values

In [138]:
```
df_train['Electrical'].unique()
```

Out[138]:
```
array(['SBrkr', 'FuseF', 'FuseA', 'FuseP', 'Mix', nan], dtype=object)
```

In [244]:
```
df_train['GarageType'].fillna(value='No',axis=0,inplace=True)
df_test['GarageType'].fillna(value='No',axis=0,inplace=True)
df_train['GarageFinish'].fillna(value='NoGarage',axis=0,inplace=True)
df_test['GarageFinish'].fillna(value='NoGarage',axis=0,inplace=True)
df_train['GarageQual'].fillna(value=0,axis=0,inplace=True)
df_test['GarageQual'].fillna(value=0,axis=0,inplace=True)
df_train['GarageCond'].fillna(value='Po',axis=0,inplace=True)
df_test['GarageCond'].fillna(value='Po',axis=0,inplace=True)
df_train['BsmtExposure'].fillna(value='No',axis=0,inplace=True)
df_test['BsmtExposure'].fillna(value='No',axis=0,inplace=True)
df_train['BsmtFinType2'].fillna(value='No',axis=0,inplace=True)
df_test['BsmtFinType2'].fillna(value='No',axis=0,inplace=True)
df_train['BsmtQual'].fillna(value=0,axis=0,inplace=True)
df_test['BsmtQual'].fillna(value=0,axis=0,inplace=True)
df_train['BsmtCond'].fillna(value='Po',axis=0,inplace=True)
df_test['BsmtCond'].fillna(value='Po',axis=0,inplace=True)
df_train['BsmtFinType1'].fillna(value='No',axis=0,inplace=True)
df_test['BsmtFinType1'].fillna(value='No',axis=0,inplace=True)
df_train['MasVnrType'].fillna(value='No',axis=0,inplace=True)
df_test['MasVnrType'].fillna(value='No',axis=0,inplace=True)
df_train['MasVnrArea'].fillna(value=0,axis=0,inplace=True)
df_test['MasVnrArea'].fillna(value=0,axis=0,inplace=True)
df_train['Electrical'].fillna(value='No',axis=0,inplace=True)
df_test['Electrical'].fillna(value='No',axis=0,inplace=True)
```

In [245]:
```
findMissingValue(df_train)
```

Out[245]:

Columns	Number_Of_MValue	Percentage_Of_MValue

In [246]:
```
findMissingValue(df_test)
```

Out[246]:

	Columns	Number_Of_MValue	Percentage_Of_MValue
0	MSZoning	4.00	0.27 %
1	Utilities	2.00	0.14 %
2	BsmtFullBath	2.00	0.14 %
3	BsmtHalfBath	2.00	0.14 %
4	Functional	2.00	0.14 %
5	Exterior1st	1.00	0.07 %
6	Exterior2nd	1.00	0.07 %
7	BsmtFinSF1	1.00	0.07 %
8	BsmtFinSF2	1.00	0.07 %
9	BsmtUnfSF	1.00	0.07 %
10	TotalBsmtSF	1.00	0.07 %
11	KitchenQual	1.00	0.07 %
12	GarageCars	1.00	0.07 %
13	GarageArea	1.00	0.07 %
14	SaleType	1.00	0.07 %

In [255]:
```
df_test['KitchenQual'].unique()
```

Out[255]:
```
array([3, 4, 5, 2, nan], dtype=object)
```

In [256]:
```
df_test['MSZoning'].fillna(value='No',axis=0,inplace=True)
df_test['Utilities'].fillna(value='No',axis=0,inplace=True)
df_test['BsmtFullBath'].fillna(value=0,axis=0,inplace=True)
df_test['BsmtHalfBath'].fillna(value=0,axis=0,inplace=True)
df_test['Functional'].fillna(value='No',axis=0,inplace=True)
df_test['Exterior1st'].fillna(value='No',axis=0,inplace=True)
df_test['Exterior2nd'].fillna(value='No',axis=0,inplace=True)
df_test['BsmtFinSF1'].fillna(value=0,axis=0,inplace=True)
df_test['BsmtFinSF2'].fillna(value=0,axis=0,inplace=True)
df_test['BsmtUnfSF'].fillna(value=0,axis=0,inplace=True)
df_test['TotalBsmtSF'].fillna(value=0,axis=0,inplace=True)
df_test['KitchenQual'].fillna(value=0,axis=0,inplace=True)
df_test['GarageCars'].fillna(value=0,axis=0,inplace=True)
df_test['GarageArea'].fillna(value=0,axis=0,inplace=True)
df_test['SaleType'].fillna(value='No',axis=0,inplace=True)
```

In [257]:
```
findMissingValue(df_test)
```

Out[257]:

Columns	Number_Of_MValue	Percentage_Of_MValue

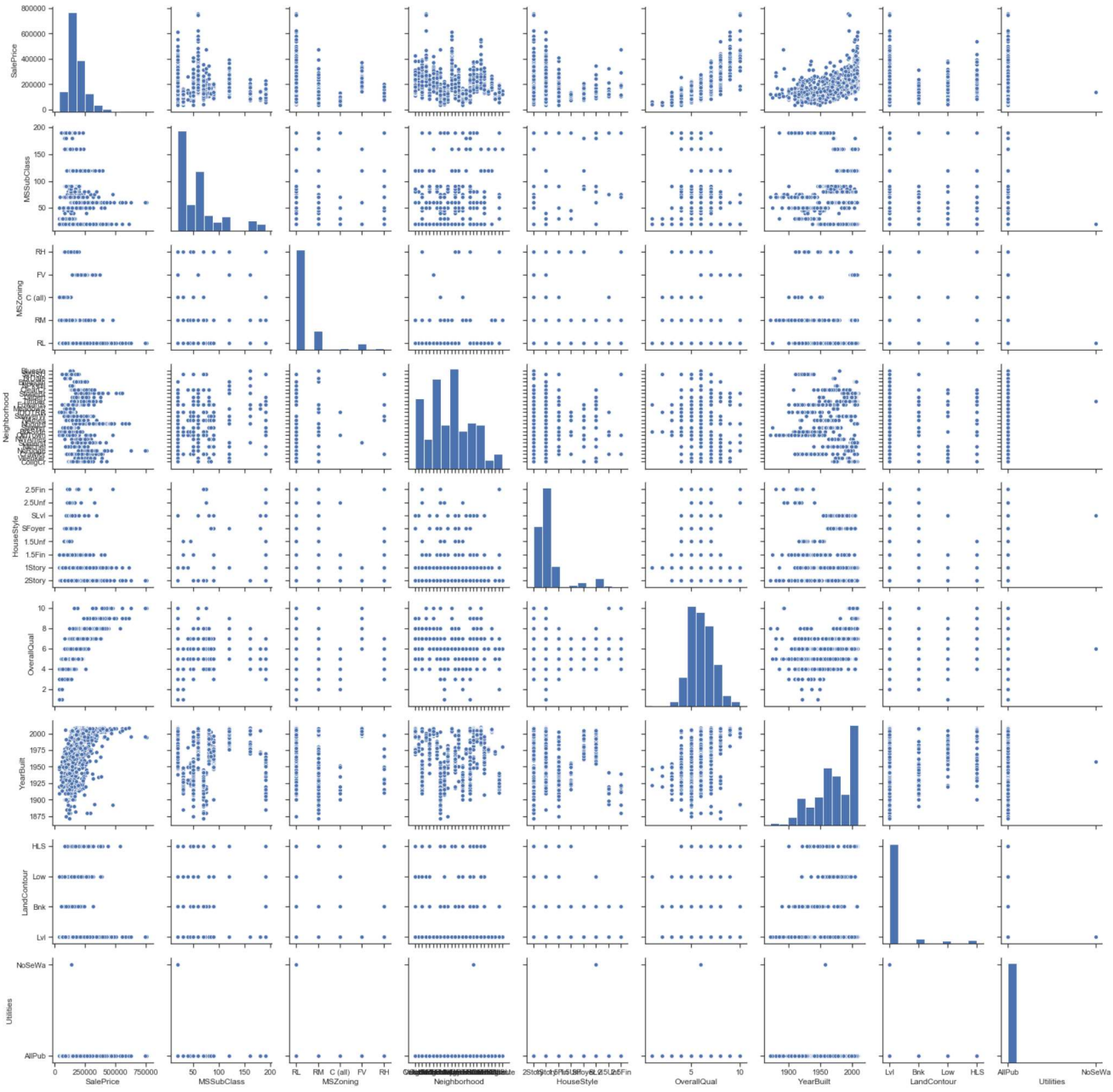
In [258]:
```
len(df_test)
```

Out[258]:
```
1459
```

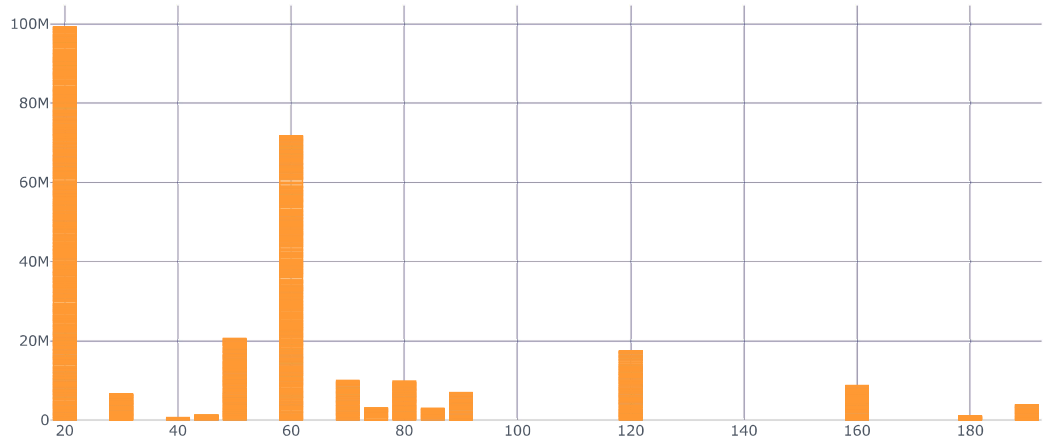
In [259]:
```
df_t = df_test.copy()
```

Exploratory Data Analysis:

In [169]:
```
g = sns.pairplot(df_train, vars=['SalePrice','MSSubClass','MSZoning','Neighborhood','HouseStyle','OverallQual','YearBuilt','LandContour','Utilities'])
```

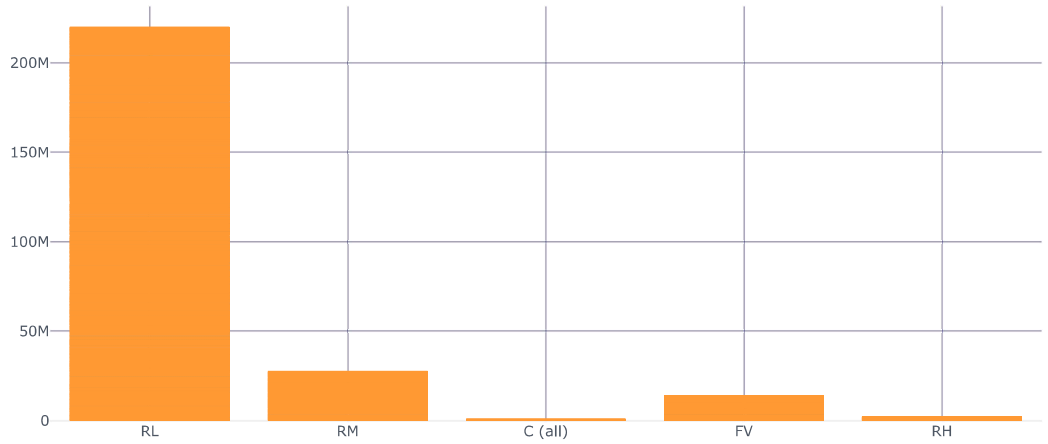


In [296]:
```
df_train.iplot(kind='bar',x='MSSubClass',y='SalePrice',gridcolor='blue',bins=30)
```



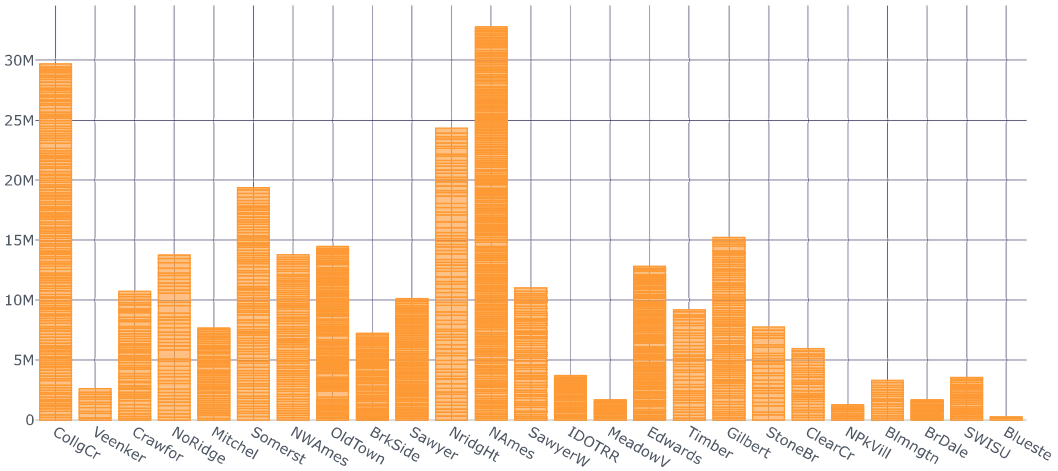
Export to plot.ly »

In [171]:
```
df_train.iplot(kind='bar',x='MSZoning',y='SalePrice',gridcolor='blue',bins=30)
```



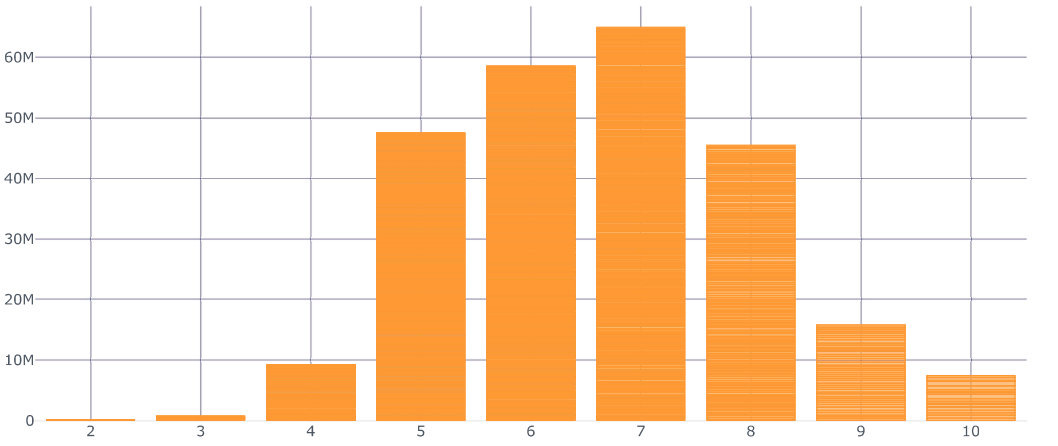
Export to plot.ly »

In [172]:
```
df_train.iplot(kind='bar',x='Neighborhood',y='SalePrice',gridcolor='blue',bins=30)
```



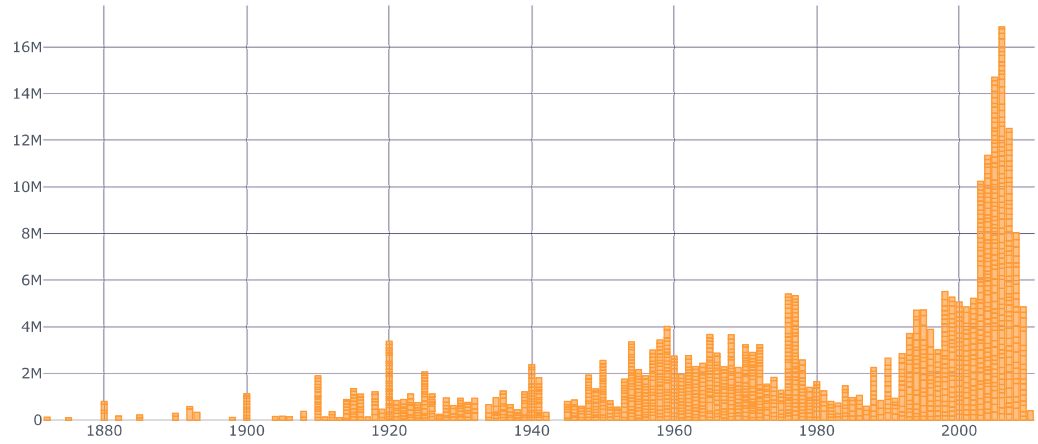
Export to plot.ly »

In [30]:
```
df_train.iplot(kind='bar',x='OverallQual',y='SalePrice',gridcolor='blue',bins=30)
```



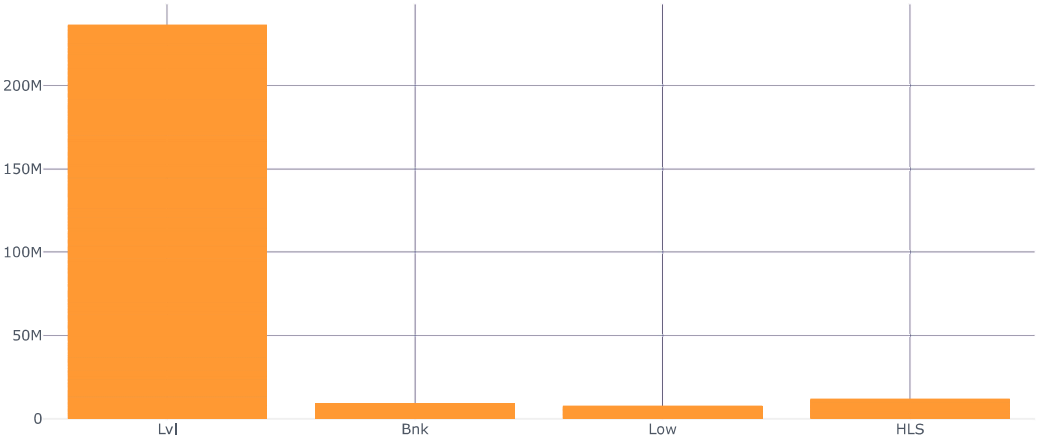
Export to plot.ly »

In [173]:
```
df_train.iplot(kind='bar',x='YearBuilt',y='SalePrice',gridcolor='blue',bins=30)
```



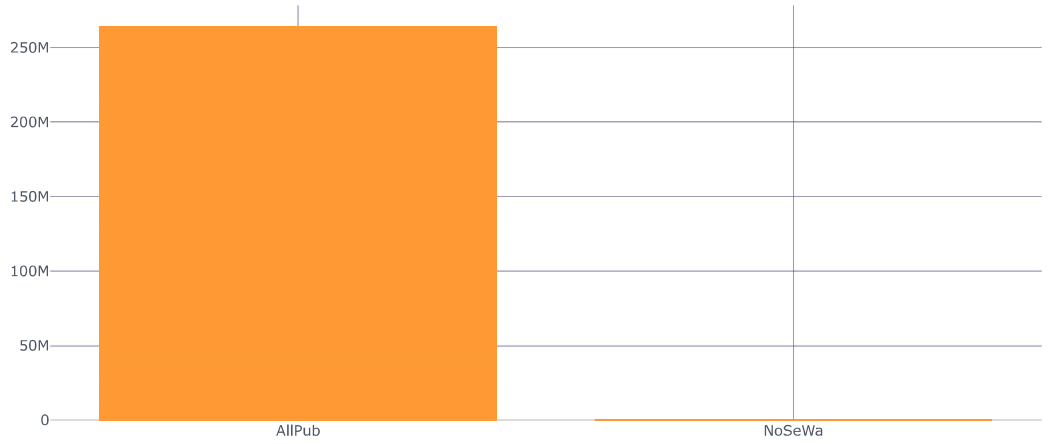
Export to plot.ly »

In [174]:
```
df_train.iplot(kind='bar',x='LandContour',y='SalePrice',gridcolor='blue',bins=30)
```



Export to plot.ly »

In [175]:
```
df_train.iplot(kind='bar',x='Utilities',y='SalePrice',gridcolor='blue',bins=30)
```



Export to plot.ly »

Predictive Analysis:

In [260]:
```
df_train_copy = df_train.copy()
df_test_copy = df_test.copy()
```

In [261]:
```
df_train.drop(columns=['Id'],axis=1,inplace=True)
df_test.drop(columns=['Id'],axis=1,inplace=True)
```

In [262]:
```
df_train.columns.nunique()
```

Out[262]: 76

In [263]:
```
df_test.columns.nunique()
```

Out[263]: 75

In [264]:
```
df_train = pd.get_dummies(df_train,drop_first=True)
```

In [265]:
```
df_train.columns.nunique()
```

Out[265]: 234

In [266]:
```
df_test= pd.get_dummies(df_test,drop_first=True)
```

In [267]:
```
df_test.columns.nunique()
```

Out[267]: 221

In [268]:
```
df_train.columns.difference(df_test.columns)
```

Out[268]:
```
Index(['Condition2_RRAe', 'Condition2_RRAn', 'Condition2_RRNn',
       'Electrical_Mix', 'Electrical_No', 'Exterior1st_ImStucc',
       'Exterior1st_Stone', 'Exterior2nd_Other', 'Heating_GasA',
       'Heating_OthW', 'HouseStyle_2.5Fin', 'KitchenQual_3', 'KitchenQual_4',
       'KitchenQual_5', 'RoofMatl_CompShg', 'RoofMatl_Membran',
       'RoofMatl_Metal', 'RoofMatl_Roll', 'SalePrice', 'Utilities_NoSeWa'],
      dtype='object')
```

In [269]:
```
df_train.drop(columns=['Condition2_RRAe', 'Condition2_RRAn', 'Condition2_RRNn',
       'Electrical_Mix', 'Electrical_No', 'Exterior1st_ImStucc',
       'Exterior1st_Stone', 'Exterior2nd_Other', 'Heating_GasA',
       'Heating_OthW', 'HouseStyle_2.5Fin', 'KitchenQual_3', 'KitchenQual_4',
       'KitchenQual_5', 'RoofMatl_CompShg', 'RoofMatl_Membran',
       'RoofMatl_Metal', 'RoofMatl_Roll', 'Utilities_NoSeWa'],axis=1,inplace=True)
```

In [270]:
```
df_train.columns.nunique()
```

Out[270]: 215

In [271]:
```
df_test.columns.difference(df_train.columns)
```

Out[271]:
```
Index(['Exterior1st_No', 'Exterior2nd_No', 'Functional_No', 'KitchenQual',
       'MSZoning_No', 'SaleType_No', 'Utilities_No'],
      dtype='object')
```

In [272]:
```
df_test.drop(columns=['Exterior1st_No', 'Exterior2nd_No', 'Functional_No', 'KitchenQual',
       'MSZoning_No', 'SaleType_No', 'Utilities_No'],axis=1,inplace=True)
```

In [273]:
```
df_test.columns.nunique()
```

Out[273]: 214

Building Machine Learning Algorithm

In [274]:
```
X= df_train.drop(['SalePrice'],axis=1)
y= df_train['SalePrice']
```

In [275]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
```

In [276]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=101)
```

In [277]:
```
from imblearn.ensemble import BalancedBaggingClassifier
from sklearn.linear_model import LogisticRegression

#Create an object of the classifier.
bbc_lr = BalancedBaggingClassifier(base_estimator=LogisticRegression(),
                                   sampling_strategy='auto',
                                   replacement=False,
                                   random_state=1)

y_train = df_train['SalePrice']
X_train = df_train.drop(['SalePrice'], axis=1, inplace=False)

#Train the classifier.
bbc_lr.fit(X_train, y_train)


prediction = bbc_lr.predict(X_test)

bbc_lr.score(X_test,y_test)
print('The Logistic Regression Accuracy is {:.2f} %'.format(bbc_lr.score(X_test,y_test)*100))

print('\n')

print(classification_report(y_test,prediction))
print('\n')
print(confusion_matrix(y_test,prediction))
```

```
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:
Liblinear failed to converge, increase the number of iterations.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:
Liblinear failed to converge, increase the number of iterations.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:
Liblinear failed to converge, increase the number of iterations.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:
Liblinear failed to converge, increase the number of iterations.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:
Liblinear failed to converge, increase the number of iterations.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:
Liblinear failed to converge, increase the number of iterations.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:
Liblinear failed to converge, increase the number of iterations.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:
Liblinear failed to converge, increase the number of iterations.
```

```
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:
Liblinear failed to converge, increase the number of iterations.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning:
Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
The Logistic Regression Accuracy is 65.41 %
```

	precision	recall	f1-score	support
34900	1.00	1.00	1.00	1
35311	1.00	1.00	1.00	1
39300	1.00	1.00	1.00	1
40000	1.00	1.00	1.00	1
52500	1.00	1.00	1.00	1
55000	1.00	1.00	1.00	1
58500	1.00	1.00	1.00	1
60000	0.00	0.00	0.00	0
67000	0.00	0.00	0.00	0
75000	0.00	0.00	0.00	0
76000	0.50	1.00	0.67	1
79000	1.00	1.00	1.00	1
80000	0.50	1.00	0.67	1
80500	1.00	1.00	1.00	1
83000	1.00	1.00	1.00	1
85000	0.00	0.00	0.00	0
86000	0.00	0.00	0.00	1
87000	1.00	1.00	1.00	1
87500	1.00	1.00	1.00	1
89500	1.00	1.00	1.00	1
90000	0.00	0.00	0.00	1
91000	1.00	0.50	0.67	2
92900	0.00	0.00	0.00	0
93000	1.00	1.00	1.00	2
94750	0.00	0.00	0.00	0
95000	1.00	1.00	1.00	2
97000	1.00	1.00	1.00	1
100000	1.00	0.25	0.40	4
101000	0.00	0.00	0.00	0
102000	0.50	1.00	0.67	1
102776	0.00	0.00	0.00	0
104000	1.00	1.00	1.00	1
104900	0.00	0.00	0.00	0
105900	1.00	1.00	1.00	1
107500	1.00	1.00	1.00	1
108000	0.00	0.00	0.00	2
109008	0.00	0.00	0.00	0
109900	1.00	1.00	1.00	1
110000	0.00	0.00	0.00	3
110500	1.00	1.00	1.00	1
111000	0.00	0.00	0.00	0
111250	1.00	1.00	1.00	1
112000	1.00	0.33	0.50	3
112500	0.00	0.00	0.00	1
113000	1.00	0.50	0.67	2
114500	0.20	1.00	0.33	1
114504	1.00	1.00	1.00	1
116000	1.00	1.00	1.00	1
117000	0.00	0.00	0.00	1
118000	0.00	0.00	0.00	0
118400	0.00	0.00	0.00	0
119000	1.00	1.00	1.00	1
119200	1.00	1.00	1.00	1
120000	1.00	1.00	1.00	1
121000	0.00	0.00	0.00	0
121600	1.00	1.00	1.00	1
122500	1.00	1.00	1.00	1
124000	1.00	0.50	0.67	2
125000	1.00	1.00	1.00	1
125500	1.00	1.00	1.00	1
126000	1.00	1.00	1.00	1
126175	1.00	1.00	1.00	1
127000	1.00	0.33	0.50	3
128000	0.00	0.00	0.00	1
128500	0.50	0.50	0.50	2
128950	1.00	1.00	1.00	1
129000	1.00	0.50	0.67	2
129500	0.00	0.00	0.00	0
129900	1.00	1.00	1.00	1
130000	0.00	0.00	0.00	2
130250	0.00	0.00	0.00	0
130500	1.00	1.00	1.00	1
131500	0.00	0.00	0.00	1
132000	0.00	0.00	0.00	2
132250	1.00	1.00	1.00	1
132500	1.00	1.00	1.00	1
133000	1.00	1.00	1.00	1
134450	0.00	0.00	0.00	0
134500	1.00	1.00	1.00	1
134800	1.00	1.00	1.00	1
135000	1.00	0.25	0.40	4
135900	1.00	1.00	1.00	1
135960	0.00	0.00	0.00	0
136000	1.00	1.00	1.00	1
136900	0.00	0.00	0.00	0
137000	1.00	0.33	0.50	3
137500	0.00	0.00	0.00	1
139000	1.00	0.50	0.67	4
139950	0.00	0.00	0.00	0
140000	1.00	0.20	0.33	5

141000	1.00	0.50	0.67	2
142000	1.00	0.50	0.67	2
142500	1.00	1.00	1.00	1
142953	1.00	1.00	1.00	1
143000	0.00	0.00	0.00	1
143500	0.00	0.00	0.00	0
144000	0.00	0.00	0.00	2
144152	1.00	1.00	1.00	1
144900	0.00	0.00	0.00	0
145000	1.00	0.29	0.44	7
145250	0.50	1.00	0.67	1
146800	1.00	1.00	1.00	1
147400	1.00	1.00	1.00	1
148000	0.00	0.00	0.00	1
148500	0.50	1.00	0.67	1
149350	1.00	1.00	1.00	1
149500	0.00	0.00	0.00	1
149900	0.00	0.00	0.00	0
150500	0.50	1.00	0.67	1
150750	0.50	1.00	0.67	1
152000	0.00	0.00	0.00	1
153337	0.00	0.00	0.00	0
153500	1.00	1.00	1.00	1
153575	0.00	0.00	0.00	0
153900	1.00	1.00	1.00	1
154000	0.00	0.00	0.00	0
154900	0.00	0.00	0.00	0
155000	1.00	0.33	0.50	3
155900	0.50	1.00	0.67	1
156000	1.00	1.00	1.00	1
157000	1.00	1.00	1.00	1
157500	1.00	1.00	1.00	1
158000	0.00	0.00	0.00	2
158500	1.00	1.00	1.00	1
159000	0.00	0.00	0.00	1
159500	1.00	0.50	0.67	2
160000	1.00	0.25	0.40	4
162500	0.00	0.00	0.00	0
162900	1.00	1.00	1.00	1
163000	0.00	0.00	0.00	1
163900	0.00	0.00	0.00	0
164500	0.50	1.00	0.67	1
164900	1.00	1.00	1.00	1
165000	1.00	1.00	1.00	1
165150	1.00	1.00	1.00	1
165400	1.00	1.00	1.00	1
165500	1.00	1.00	1.00	1
166000	1.00	1.00	1.00	1
167500	0.00	0.00	0.00	0
168000	1.00	1.00	1.00	1
168500	1.00	1.00	1.00	1
169000	1.00	1.00	1.00	1
169500	1.00	1.00	1.00	1
171750	1.00	1.00	1.00	1
172785	1.00	1.00	1.00	1
173000	0.50	0.50	0.50	2
173900	0.00	0.00	0.00	0
174000	0.00	0.00	0.00	2
174500	0.00	0.00	0.00	0
174900	0.00	0.00	0.00	0
175000	0.00	0.00	0.00	1
175900	0.00	0.00	0.00	1
176000	1.00	0.50	0.67	2
176500	0.00	0.00	0.00	0
177500	0.00	0.00	0.00	1
178000	1.00	0.50	0.67	2
178900	0.00	0.00	0.00	0
179000	0.00	0.00	0.00	2
179200	1.00	1.00	1.00	1
179500	0.00	0.00	0.00	0
179540	1.00	1.00	1.00	1
179600	0.50	1.00	0.67	1
179900	1.00	1.00	1.00	2
180000	0.00	0.00	0.00	2
180500	1.00	1.00	1.00	2
182900	0.00	0.00	0.00	0
183500	1.00	1.00	1.00	1
184000	1.00	0.50	0.67	2
184900	0.00	0.00	0.00	0
185000	0.00	0.00	0.00	2
185500	0.33	1.00	0.50	1
187000	0.33	1.00	0.50	1
187500	1.00	1.00	1.00	2
189000	1.00	1.00	1.00	1
189950	1.00	1.00	1.00	1
190000	0.00	0.00	0.00	3
191000	1.00	0.50	0.67	2
192000	1.00	1.00	1.00	1
194201	1.00	1.00	1.00	1
194500	1.00	1.00	1.00	1
196000	1.00	1.00	1.00	1
196500	1.00	1.00	1.00	1
197000	0.00	0.00	0.00	1
197500	0.00	0.00	0.00	1
197900	1.00	1.00	1.00	1
200000	0.00	0.00	0.00	2

200500	0.00	0.00	0.00	0
203000	0.00	0.00	0.00	0
204000	0.50	1.00	0.67	1
204900	0.00	0.00	0.00	0
205000	0.00	0.00	0.00	0
206000	0.00	0.00	0.00	0
206300	1.00	1.00	1.00	1
207000	1.00	1.00	1.00	1
207500	0.00	0.00	0.00	1
208500	1.00	1.00	1.00	1
211000	1.00	1.00	1.00	1
212000	1.00	1.00	1.00	1
212900	0.00	0.00	0.00	0
213000	1.00	0.50	0.67	2
213490	1.00	1.00	1.00	1
214000	0.00	0.00	0.00	2
214900	1.00	1.00	1.00	1
216837	1.00	1.00	1.00	1
219500	1.00	1.00	1.00	2
220000	1.00	0.33	0.50	3
221000	0.00	0.00	0.00	0
222000	0.00	0.00	0.00	1
222500	0.00	0.00	0.00	1
223000	0.00	0.00	0.00	0
223500	1.00	1.00	1.00	1
224900	0.00	0.00	0.00	0
225000	0.00	0.00	0.00	2
226000	0.00	0.00	0.00	0
226700	0.00	0.00	0.00	0
227875	0.00	0.00	0.00	0
229456	0.00	0.00	0.00	0
230000	0.00	0.00	0.00	2
230500	0.00	0.00	0.00	0
232000	0.00	0.00	0.00	1
234000	1.00	1.00	1.00	1
236000	0.50	1.00	0.67	1
236500	1.00	1.00	1.00	1
237000	0.00	0.00	0.00	0
239000	1.00	0.50	0.67	2
239900	0.00	0.00	0.00	0
243000	0.00	0.00	0.00	0
244000	1.00	1.00	1.00	1
246578	1.00	1.00	1.00	1
248000	1.00	1.00	1.00	1
249700	0.00	0.00	0.00	0
250000	1.00	0.50	0.67	2
252000	1.00	1.00	1.00	1
252678	1.00	1.00	1.00	1
255000	0.50	1.00	0.67	1
255500	1.00	1.00	1.00	1
256000	1.00	1.00	1.00	1
256300	1.00	1.00	1.00	1
257000	0.50	1.00	0.67	1
259500	1.00	1.00	1.00	1
260000	1.00	1.00	1.00	1
263435	1.00	1.00	1.00	1
264561	1.00	1.00	1.00	1
265979	1.00	1.00	1.00	1
268000	1.00	1.00	1.00	1
269500	1.00	1.00	1.00	1
271000	0.00	0.00	0.00	0
274000	1.00	1.00	1.00	1
281213	0.00	0.00	0.00	0
285000	1.00	1.00	1.00	1
286000	1.00	1.00	1.00	1
287000	1.00	1.00	1.00	1
290000	1.00	0.50	0.67	2
295000	1.00	1.00	1.00	1
301500	1.00	1.00	1.00	1
302000	1.00	1.00	1.00	1
303477	1.00	1.00	1.00	1
311500	1.00	1.00	1.00	1
313000	0.00	0.00	0.00	0
314813	1.00	1.00	1.00	1
315500	1.00	1.00	1.00	1
315750	1.00	1.00	1.00	1
318000	1.00	1.00	1.00	1
319900	1.00	1.00	1.00	1
325300	1.00	1.00	1.00	1
328000	1.00	1.00	1.00	1
335000	1.00	1.00	1.00	1
359100	1.00	1.00	1.00	1
372500	1.00	1.00	1.00	1
377426	1.00	1.00	1.00	1
377500	1.00	1.00	1.00	1
386250	1.00	1.00	1.00	1
392000	1.00	1.00	1.00	1
392500	1.00	1.00	1.00	1
402861	1.00	1.00	1.00	1
412500	1.00	1.00	1.00	1
424870	1.00	1.00	1.00	1
475000	1.00	1.00	1.00	1
582933	1.00	1.00	1.00	1
accuracy			0.65	292
macro avg	0.62	0.60	0.59	292

```
weighted avg       0.78      0.65      0.67       292


[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning:

Liblinear failed to converge, increase the number of iterations.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\metrics\classification.py:1437: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\metrics\classification.py:1439: UndefinedMetricWarning:

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

In [278]:
```
from sklearn.tree import DecisionTreeClassifier

#Create an object of the classifier.
bbc_dt = BalancedBaggingClassifier(base_estimator=DecisionTreeClassifier(),
                                   sampling_strategy='auto',
                                   replacement=False,
                                   random_state=0)

y_train = df_train['SalePrice']
X_train = df_train.drop(['SalePrice'], axis=1, inplace=False)

#Train the classifier.
bbc_dt.fit(X_train, y_train)

prediction = bbc_dt.predict(X_test)

bbc_dt.score(X_test,y_test)
print('The Decision Tree Accuracy is {:.2f} %'.format(bbc_dt.score(X_test,y_test)*100))

print('\n')

print(classification_report(y_test,prediction))
print('\n')
print(confusion_matrix(y_test,prediction))
```

```
The Decision Tree Accuracy is 67.47 %
```

	precision	recall	f1-score	support
34900	1.00	1.00	1.00	1
35311	0.25	1.00	0.40	1
37900	0.00	0.00	0.00	0
39300	1.00	1.00	1.00	1
40000	0.50	1.00	0.67	1
52500	1.00	1.00	1.00	1
55000	1.00	1.00	1.00	1
58500	1.00	1.00	1.00	1
61000	0.00	0.00	0.00	0
62383	0.00	0.00	0.00	0
66500	0.00	0.00	0.00	0
68400	0.00	0.00	0.00	0
75000	0.00	0.00	0.00	0
75500	0.00	0.00	0.00	0
76000	0.50	1.00	0.67	1
76500	0.00	0.00	0.00	0
79000	0.33	1.00	0.50	1
79500	0.00	0.00	0.00	0
80000	0.00	0.00	0.00	1
80500	1.00	1.00	1.00	1
82500	0.00	0.00	0.00	0
83000	0.50	1.00	0.67	1
83500	0.00	0.00	0.00	0
84000	0.00	0.00	0.00	0
85400	0.00	0.00	0.00	0
86000	0.00	0.00	0.00	1
87000	1.00	1.00	1.00	1
87500	1.00	1.00	1.00	1
89000	0.00	0.00	0.00	0
89500	1.00	1.00	1.00	1
90000	1.00	1.00	1.00	1
90350	0.00	0.00	0.00	0
91000	0.67	1.00	0.80	2
93000	1.00	0.50	0.67	2
93500	0.00	0.00	0.00	0
95000	1.00	1.00	1.00	2
96500	0.00	0.00	0.00	0
97000	1.00	1.00	1.00	1
98300	0.00	0.00	0.00	0
98600	0.00	0.00	0.00	0
99500	0.00	0.00	0.00	0
100000	0.33	0.25	0.29	4
102000	1.00	1.00	1.00	1
102776	0.00	0.00	0.00	0
104000	1.00	1.00	1.00	1
105900	1.00	1.00	1.00	1
107500	1.00	1.00	1.00	1
107900	0.00	0.00	0.00	0
108000	1.00	1.00	1.00	2
108500	0.00	0.00	0.00	0
109900	0.00	0.00	0.00	1
110000	1.00	0.33	0.50	3
110500	0.50	1.00	0.67	1
111000	0.00	0.00	0.00	0
111250	0.33	1.00	0.50	1
112000	0.00	0.00	0.00	3
112500	1.00	1.00	1.00	1
113000	1.00	1.00	1.00	2
114500	1.00	1.00	1.00	1
114504	1.00	1.00	1.00	1
116000	1.00	1.00	1.00	1
117000	0.50	1.00	0.67	1
118858	0.00	0.00	0.00	0
119000	0.00	0.00	0.00	1
119200	1.00	1.00	1.00	1
120000	0.00	0.00	0.00	1
121500	0.00	0.00	0.00	0
121600	1.00	1.00	1.00	1
122500	0.50	1.00	0.67	1
124000	1.00	0.50	0.67	2
125000	0.00	0.00	0.00	1
125500	0.00	0.00	0.00	1
126000	0.50	1.00	0.67	1
126175	1.00	1.00	1.00	1
127000	0.50	0.33	0.40	3
127500	0.00	0.00	0.00	0
128000	0.00	0.00	0.00	1
128500	0.50	0.50	0.50	2
128950	0.50	1.00	0.67	1
129000	1.00	1.00	1.00	2
129900	1.00	1.00	1.00	1
130000	0.00	0.00	0.00	2
130500	1.00	1.00	1.00	1
131400	0.00	0.00	0.00	0
131500	0.00	0.00	0.00	1
132000	1.00	0.50	0.67	2
132250	1.00	1.00	1.00	1
132500	0.00	0.00	0.00	1
133000	0.00	0.00	0.00	1
134500	1.00	1.00	1.00	1

134800	1.00	1.00	1.00	1
135000	0.00	0.00	0.00	4
135900	1.00	1.00	1.00	1
136000	1.00	1.00	1.00	1
137000	1.00	0.33	0.50	3
137500	0.00	0.00	0.00	1
138000	0.00	0.00	0.00	0
138887	0.00	0.00	0.00	0
139000	0.00	0.00	0.00	4
140000	1.00	0.40	0.57	5
141000	1.00	0.50	0.67	2
142000	0.50	0.50	0.50	2
142500	1.00	1.00	1.00	1
142600	0.00	0.00	0.00	0
142953	1.00	1.00	1.00	1
143000	0.00	0.00	0.00	1
143500	0.00	0.00	0.00	0
144000	1.00	0.50	0.67	2
144152	1.00	1.00	1.00	1
144900	0.00	0.00	0.00	0
145000	0.00	0.00	0.00	7
145250	1.00	1.00	1.00	1
146800	1.00	1.00	1.00	1
147400	1.00	1.00	1.00	1
148000	1.00	1.00	1.00	1
148500	0.50	1.00	0.67	1
149300	0.00	0.00	0.00	0
149350	1.00	1.00	1.00	1
149500	1.00	1.00	1.00	1
150500	0.50	1.00	0.67	1
150750	0.50	1.00	0.67	1
151000	0.00	0.00	0.00	0
152000	1.00	1.00	1.00	1
153500	1.00	1.00	1.00	1
153900	1.00	1.00	1.00	1
155000	1.00	0.33	0.50	3
155835	0.00	0.00	0.00	0
155900	0.50	1.00	0.67	1
156000	1.00	1.00	1.00	1
156500	0.00	0.00	0.00	0
157000	1.00	1.00	1.00	1
157500	1.00	1.00	1.00	1
158000	1.00	0.50	0.67	2
158500	1.00	1.00	1.00	1
158900	0.00	0.00	0.00	0
159000	0.00	0.00	0.00	1
159500	1.00	1.00	1.00	2
159950	0.00	0.00	0.00	0
160000	0.00	0.00	0.00	4
162000	0.00	0.00	0.00	0
162900	1.00	1.00	1.00	1
163000	0.00	0.00	0.00	1
164000	0.00	0.00	0.00	0
164500	1.00	1.00	1.00	1
164900	1.00	1.00	1.00	1
165000	0.00	0.00	0.00	1
165150	1.00	1.00	1.00	1
165400	1.00	1.00	1.00	1
165500	0.00	0.00	0.00	1
165600	0.00	0.00	0.00	0
166000	1.00	1.00	1.00	1
168000	1.00	1.00	1.00	1
168500	0.50	1.00	0.67	1
169000	1.00	1.00	1.00	1
169500	1.00	1.00	1.00	1
169900	0.00	0.00	0.00	0
171750	1.00	1.00	1.00	1
172785	1.00	1.00	1.00	1
173000	1.00	1.00	1.00	2
174000	0.00	0.00	0.00	2
175000	0.00	0.00	0.00	1
175900	1.00	1.00	1.00	1
176000	1.00	0.50	0.67	2
177500	1.00	1.00	1.00	1
178000	0.00	0.00	0.00	2
179000	1.00	1.00	1.00	2
179200	0.50	1.00	0.67	1
179540	1.00	1.00	1.00	1
179600	1.00	1.00	1.00	1
179900	1.00	1.00	1.00	2
180000	1.00	0.50	0.67	2
180500	1.00	0.50	0.67	2
181000	0.00	0.00	0.00	0
183500	0.33	1.00	0.50	1
184000	1.00	1.00	1.00	2
184750	0.00	0.00	0.00	0
185000	0.00	0.00	0.00	2
185500	1.00	1.00	1.00	1
186000	0.00	0.00	0.00	0
187000	1.00	1.00	1.00	1
187100	0.00	0.00	0.00	0
187500	1.00	0.50	0.67	2
187750	0.00	0.00	0.00	0
189000	1.00	1.00	1.00	1
189950	1.00	1.00	1.00	1
190000	1.00	0.33	0.50	3

191000	1.00	0.50	0.67	2
192000	0.00	0.00	0.00	1
192500	0.00	0.00	0.00	0
194201	1.00	1.00	1.00	1
194500	0.00	0.00	0.00	1
196000	0.00	0.00	0.00	1
196500	1.00	1.00	1.00	1
197000	0.00	0.00	0.00	1
197500	1.00	1.00	1.00	1
197900	1.00	1.00	1.00	1
200000	0.00	0.00	0.00	2
201000	0.00	0.00	0.00	0
204000	1.00	1.00	1.00	1
206300	1.00	1.00	1.00	1
207000	1.00	1.00	1.00	1
207500	1.00	1.00	1.00	1
208500	1.00	1.00	1.00	1
211000	1.00	1.00	1.00	1
212000	1.00	1.00	1.00	1
213000	1.00	1.00	1.00	2
213490	0.50	1.00	0.67	1
214000	1.00	1.00	1.00	2
214900	1.00	1.00	1.00	1
216837	1.00	1.00	1.00	1
217500	0.00	0.00	0.00	0
219500	1.00	1.00	1.00	2
220000	0.00	0.00	0.00	3
221000	0.00	0.00	0.00	0
222000	1.00	1.00	1.00	1
222500	1.00	1.00	1.00	1
223500	1.00	1.00	1.00	1
225000	0.00	0.00	0.00	2
230000	0.00	0.00	0.00	2
232000	1.00	1.00	1.00	1
234000	0.00	0.00	0.00	1
236000	1.00	1.00	1.00	1
236500	1.00	1.00	1.00	1
239000	0.00	0.00	0.00	2
240000	0.00	0.00	0.00	0
243000	0.00	0.00	0.00	0
244000	1.00	1.00	1.00	1
246578	1.00	1.00	1.00	1
248000	1.00	1.00	1.00	1
250000	1.00	0.50	0.67	2
252000	1.00	1.00	1.00	1
252678	1.00	1.00	1.00	1
255000	1.00	1.00	1.00	1
255500	1.00	1.00	1.00	1
256000	1.00	1.00	1.00	1
256300	1.00	1.00	1.00	1
257000	1.00	1.00	1.00	1
259500	1.00	1.00	1.00	1
260000	1.00	1.00	1.00	1
263435	1.00	1.00	1.00	1
264561	1.00	1.00	1.00	1
265979	1.00	1.00	1.00	1
268000	1.00	1.00	1.00	1
269500	1.00	1.00	1.00	1
274000	1.00	1.00	1.00	1
285000	1.00	1.00	1.00	1
286000	1.00	1.00	1.00	1
287000	1.00	1.00	1.00	1
287090	0.00	0.00	0.00	0
290000	1.00	0.50	0.67	2
295000	1.00	1.00	1.00	1
301500	1.00	1.00	1.00	1
302000	1.00	1.00	1.00	1
303477	1.00	1.00	1.00	1
311500	1.00	1.00	1.00	1
314813	1.00	1.00	1.00	1
315500	1.00	1.00	1.00	1
315750	1.00	1.00	1.00	1
318000	1.00	1.00	1.00	1
319900	1.00	1.00	1.00	1
325300	1.00	1.00	1.00	1
328000	1.00	1.00	1.00	1
335000	1.00	1.00	1.00	1
359100	1.00	1.00	1.00	1
372500	1.00	1.00	1.00	1
377426	1.00	1.00	1.00	1
377500	1.00	1.00	1.00	1
386250	1.00	1.00	1.00	1
392000	1.00	1.00	1.00	1
392500	1.00	1.00	1.00	1
402861	1.00	1.00	1.00	1
412500	1.00	1.00	1.00	1
423000	0.00	0.00	0.00	0
424870	1.00	1.00	1.00	1
475000	1.00	1.00	1.00	1
582933	1.00	1.00	1.00	1
accuracy			0.67	292
macro avg	0.62	0.62	0.60	292
weighted avg	0.72	0.67	0.67	292

```
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\metrics\classification.py:1437: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\metrics\classification.py:1439: UndefinedMetricWarning:

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

In [279]:
```
from sklearn.ensemble import RandomForestClassifier
#Create an object of the classifier.
bbc_rf = BalancedBaggingClassifier(base_estimator=RandomForestClassifier(),
                                   sampling_strategy='auto',
                                   replacement=False,
                                   random_state=0)

y_train = df_train['SalePrice']
X_train = df_train.drop(['SalePrice'], axis=1, inplace=False)

#Train the classifier.
bbc_rf.fit(X_train, y_train)

prediction = bbc_rf.predict(X_test)

bbc_rf.score(X_test,y_test)
print('The Random Forest Accuracy is {:.2f} %'.format(bbc_rf.score(X_test,y_test)*100))

print('\n')

print(classification_report(y_test,prediction))
print('\n')
print(confusion_matrix(y_test,prediction))
```

```
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning:
The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
```

```
The Random Forest Accuracy is 78.77 %
```

	precision	recall	f1-score	support
34900	1.00	1.00	1.00	1
35311	1.00	1.00	1.00	1
39300	1.00	1.00	1.00	1
40000	1.00	1.00	1.00	1
52500	1.00	1.00	1.00	1
55000	1.00	1.00	1.00	1
58500	1.00	1.00	1.00	1
76000	1.00	1.00	1.00	1
78000	0.00	0.00	0.00	0
79000	1.00	1.00	1.00	1
79500	0.00	0.00	0.00	0
80000	0.00	0.00	0.00	1
80500	1.00	1.00	1.00	1
83000	1.00	1.00	1.00	1
85400	0.00	0.00	0.00	0
86000	0.00	0.00	0.00	1
87000	1.00	1.00	1.00	1
87500	1.00	1.00	1.00	1
89500	1.00	1.00	1.00	1
90000	0.00	0.00	0.00	1
91000	1.00	1.00	1.00	2
92000	0.00	0.00	0.00	0
92900	0.00	0.00	0.00	0
93000	1.00	0.50	0.67	2
94500	0.00	0.00	0.00	0
95000	1.00	1.00	1.00	2
97000	0.50	1.00	0.67	1
99500	0.00	0.00	0.00	0
100000	1.00	0.25	0.40	4
102000	0.50	1.00	0.67	1
104000	1.00	1.00	1.00	1
105000	0.00	0.00	0.00	0
105900	0.50	1.00	0.67	1
107500	1.00	1.00	1.00	1
107900	0.00	0.00	0.00	0
108000	1.00	1.00	1.00	2
109900	1.00	1.00	1.00	1
110000	1.00	0.33	0.50	3
110500	1.00	1.00	1.00	1
111250	1.00	1.00	1.00	1
112000	1.00	0.33	0.50	3
112500	1.00	1.00	1.00	1
113000	1.00	1.00	1.00	2
114500	1.00	1.00	1.00	1
114504	1.00	1.00	1.00	1
116000	1.00	1.00	1.00	1
117000	1.00	1.00	1.00	1
118858	0.00	0.00	0.00	0
119000	0.00	0.00	0.00	1
119200	1.00	1.00	1.00	1
120000	0.50	1.00	0.67	1
121600	1.00	1.00	1.00	1
122500	1.00	1.00	1.00	1
123600	0.00	0.00	0.00	0
124000	1.00	0.50	0.67	2
124900	0.00	0.00	0.00	0
125000	1.00	1.00	1.00	1
125500	1.00	1.00	1.00	1
126000	1.00	1.00	1.00	1
126175	1.00	1.00	1.00	1
127000	1.00	0.33	0.50	3
128000	0.00	0.00	0.00	1
128500	1.00	0.50	0.67	2
128950	1.00	1.00	1.00	1
129000	1.00	1.00	1.00	2
129900	0.00	0.00	0.00	1
130000	1.00	1.00	1.00	2
130500	1.00	1.00	1.00	1
131500	1.00	1.00	1.00	1
132000	0.67	1.00	0.80	2
132250	1.00	1.00	1.00	1
132500	0.00	0.00	0.00	1
133000	0.00	0.00	0.00	1
134000	0.00	0.00	0.00	0
134500	0.50	1.00	0.67	1
134800	1.00	1.00	1.00	1
134900	0.00	0.00	0.00	0
135000	0.00	0.00	0.00	4
135750	0.00	0.00	0.00	0
135900	1.00	1.00	1.00	1
135960	0.00	0.00	0.00	0
136000	1.00	1.00	1.00	1
136905	0.00	0.00	0.00	0
137000	1.00	1.00	1.00	3
137500	1.00	1.00	1.00	1
137900	0.00	0.00	0.00	0
139000	1.00	0.25	0.40	4
139600	0.00	0.00	0.00	0
140000	1.00	0.60	0.75	5
141000	1.00	0.50	0.67	2

142000	1.00	1.00	1.00	2
142500	0.50	1.00	0.67	1
142600	0.00	0.00	0.00	0
142953	0.50	1.00	0.67	1
143000	0.00	0.00	0.00	1
143500	0.00	0.00	0.00	0
144000	1.00	0.50	0.67	2
144152	1.00	1.00	1.00	1
144900	0.00	0.00	0.00	0
145000	0.80	0.57	0.67	7
145250	1.00	1.00	1.00	1
146800	1.00	1.00	1.00	1
147400	1.00	1.00	1.00	1
148000	1.00	1.00	1.00	1
148500	0.50	1.00	0.67	1
149350	1.00	1.00	1.00	1
149500	1.00	1.00	1.00	1
150500	0.50	1.00	0.67	1
150750	0.50	1.00	0.67	1
152000	1.00	1.00	1.00	1
153500	1.00	1.00	1.00	1
153900	1.00	1.00	1.00	1
155000	0.67	0.67	0.67	3
155900	1.00	1.00	1.00	1
156000	1.00	1.00	1.00	1
156500	0.00	0.00	0.00	0
157000	1.00	1.00	1.00	1
157500	1.00	1.00	1.00	1
158000	1.00	1.00	1.00	2
158500	1.00	1.00	1.00	1
159000	0.00	0.00	0.00	1
159500	1.00	1.00	1.00	2
160000	1.00	0.50	0.67	4
162000	0.00	0.00	0.00	0
162900	1.00	1.00	1.00	1
163000	1.00	1.00	1.00	1
163900	0.00	0.00	0.00	0
164000	0.00	0.00	0.00	0
164500	1.00	1.00	1.00	1
164900	1.00	1.00	1.00	1
165000	0.00	0.00	0.00	1
165150	1.00	1.00	1.00	1
165400	1.00	1.00	1.00	1
165500	0.00	0.00	0.00	1
166000	1.00	1.00	1.00	1
167900	0.00	0.00	0.00	0
168000	1.00	1.00	1.00	1
168500	1.00	1.00	1.00	1
169000	1.00	1.00	1.00	1
169500	1.00	1.00	1.00	1
171750	1.00	1.00	1.00	1
172785	1.00	1.00	1.00	1
173000	1.00	0.50	0.67	2
174000	0.50	0.50	0.50	2
174500	0.00	0.00	0.00	0
174900	0.00	0.00	0.00	0
175000	1.00	1.00	1.00	1
175500	0.00	0.00	0.00	0
175900	1.00	1.00	1.00	1
176000	1.00	0.50	0.67	2
177500	1.00	1.00	1.00	1
178000	1.00	0.50	0.67	2
179000	1.00	1.00	1.00	2
179200	1.00	1.00	1.00	1
179500	0.00	0.00	0.00	0
179540	1.00	1.00	1.00	1
179600	1.00	1.00	1.00	1
179900	1.00	1.00	1.00	2
180000	1.00	0.50	0.67	2
180500	1.00	1.00	1.00	2
183200	0.00	0.00	0.00	0
183500	1.00	1.00	1.00	1
184000	1.00	1.00	1.00	2
185000	1.00	1.00	1.00	2
185500	1.00	1.00	1.00	1
185850	0.00	0.00	0.00	0
186500	0.00	0.00	0.00	0
187000	1.00	1.00	1.00	1
187500	1.00	0.50	0.67	2
188000	0.00	0.00	0.00	0
189000	1.00	1.00	1.00	1
189950	1.00	1.00	1.00	1
190000	1.00	0.67	0.80	3
191000	1.00	1.00	1.00	2
192000	0.00	0.00	0.00	1
194201	1.00	1.00	1.00	1
194500	0.00	0.00	0.00	1
196000	0.00	0.00	0.00	1
196500	1.00	1.00	1.00	1
197000	0.00	0.00	0.00	1
197500	1.00	1.00	1.00	1
197900	1.00	1.00	1.00	1
200000	1.00	0.50	0.67	2
204000	1.00	1.00	1.00	1
204900	0.00	0.00	0.00	0
206000	0.00	0.00	0.00	0

206300	1.00	1.00	1.00	1
207000	1.00	1.00	1.00	1
207500	1.00	1.00	1.00	1
208500	1.00	1.00	1.00	1
211000	1.00	1.00	1.00	1
212000	1.00	1.00	1.00	1
213000	1.00	1.00	1.00	2
213490	1.00	1.00	1.00	1
214000	1.00	1.00	1.00	2
214900	1.00	1.00	1.00	1
216837	1.00	1.00	1.00	1
217500	0.00	0.00	0.00	0
219500	1.00	1.00	1.00	2
220000	0.50	0.33	0.40	3
222000	1.00	1.00	1.00	1
222500	1.00	1.00	1.00	1
223500	1.00	1.00	1.00	1
225000	0.00	0.00	0.00	2
230000	1.00	0.50	0.67	2
232000	1.00	1.00	1.00	1
234000	1.00	1.00	1.00	1
236000	1.00	1.00	1.00	1
236500	1.00	1.00	1.00	1
239000	1.00	0.50	0.67	2
244000	1.00	1.00	1.00	1
246578	1.00	1.00	1.00	1
248000	1.00	1.00	1.00	1
250000	0.00	0.00	0.00	2
252000	0.50	1.00	0.67	1
252678	1.00	1.00	1.00	1
255000	1.00	1.00	1.00	1
255500	1.00	1.00	1.00	1
256000	1.00	1.00	1.00	1
256300	1.00	1.00	1.00	1
257000	1.00	1.00	1.00	1
259500	1.00	1.00	1.00	1
260000	1.00	1.00	1.00	1
263435	1.00	1.00	1.00	1
264561	1.00	1.00	1.00	1
265979	1.00	1.00	1.00	1
268000	1.00	1.00	1.00	1
269500	1.00	1.00	1.00	1
274000	1.00	1.00	1.00	1
285000	1.00	1.00	1.00	1
286000	1.00	1.00	1.00	1
287000	1.00	1.00	1.00	1
287090	0.00	0.00	0.00	0
290000	1.00	0.50	0.67	2
295000	1.00	1.00	1.00	1
301000	0.00	0.00	0.00	0
301500	1.00	1.00	1.00	1
302000	1.00	1.00	1.00	1
303477	1.00	1.00	1.00	1
307000	0.00	0.00	0.00	0
311500	1.00	1.00	1.00	1
314813	1.00	1.00	1.00	1
315500	1.00	1.00	1.00	1
315750	1.00	1.00	1.00	1
318000	1.00	1.00	1.00	1
319900	1.00	1.00	1.00	1
325300	1.00	1.00	1.00	1
328000	1.00	1.00	1.00	1
335000	1.00	1.00	1.00	1
359100	1.00	1.00	1.00	1
372500	1.00	1.00	1.00	1
377426	1.00	1.00	1.00	1
377500	1.00	1.00	1.00	1
386250	1.00	1.00	1.00	1
392000	1.00	1.00	1.00	1
392500	1.00	1.00	1.00	1
402861	1.00	1.00	1.00	1
412500	1.00	1.00	1.00	1
424870	1.00	1.00	1.00	1
475000	1.00	1.00	1.00	1
582933	1.00	1.00	1.00	1
accuracy			0.79	292
macro avg	0.74	0.72	0.72	292
weighted avg	0.88	0.79	0.81	292

```
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
```

```
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\metrics\classification.py:1437: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\metrics\classification.py:1439: UndefinedMetricWarning:

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

In [280]:
```
from sklearn.svm import SVC
#Create an object of the classifier.
bbc_sv = BalancedBaggingClassifier(base_estimator=SVC(random_state=1),
                                   sampling_strategy='auto',
                                   replacement=False,
                                   random_state=1)

y_train = df_train['SalePrice']
X_train = df_train.drop(['SalePrice'], axis=1, inplace=False)

#Train the classifier.
bbc_sv.fit(X_train, y_train)

prediction = bbc_sv.predict(X_test)

bbc_sv.score(X_test,y_test)
print('The Support Vector Accuracy is {:.2f} %'.format(bbc_sv.score(X_test,y_test)*100))

print('\n')

print(classification_report(y_test,prediction))
print('\n')
print(confusion_matrix(y_test,prediction))
```

```
C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning:

The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly t
o 'auto' or 'scale' to avoid this warning.
```

```
The Support Vector Accuracy is 40.75 %
```

	precision	recall	f1-score	support
34900	1.00	1.00	1.00	1
35311	1.00	1.00	1.00	1
39300	1.00	1.00	1.00	1
40000	1.00	1.00	1.00	1
52500	0.00	0.00	0.00	1
55000	1.00	1.00	1.00	1
58500	1.00	1.00	1.00	1
67000	0.00	0.00	0.00	0
76000	1.00	1.00	1.00	1
79000	0.00	0.00	0.00	1
80000	0.50	1.00	0.67	1
80500	1.00	1.00	1.00	1
83000	1.00	1.00	1.00	1
86000	0.00	0.00	0.00	1
87000	0.00	0.00	0.00	1
87500	1.00	1.00	1.00	1
88000	0.00	0.00	0.00	0
89500	1.00	1.00	1.00	1
90000	0.00	0.00	0.00	1
91000	1.00	0.50	0.67	2
93000	1.00	0.50	0.67	2
95000	1.00	0.50	0.67	2
97000	0.00	0.00	0.00	1
99900	0.00	0.00	0.00	0
100000	0.00	0.00	0.00	4
102000	0.00	0.00	0.00	1
104000	1.00	1.00	1.00	1
105900	1.00	1.00	1.00	1
106500	0.00	0.00	0.00	0
107000	0.00	0.00	0.00	0
107500	0.00	0.00	0.00	1
108000	0.00	0.00	0.00	2
108500	0.00	0.00	0.00	0
109900	0.50	1.00	0.67	1
110000	0.00	0.00	0.00	3
110500	1.00	1.00	1.00	1
111250	1.00	1.00	1.00	1
112000	0.00	0.00	0.00	3
112500	0.00	0.00	0.00	1
113000	0.00	0.00	0.00	2
114500	0.50	1.00	0.67	1
114504	0.00	0.00	0.00	1
116000	1.00	1.00	1.00	1
117000	0.00	0.00	0.00	1
118400	0.00	0.00	0.00	0
119000	0.00	0.00	0.00	1
119200	1.00	1.00	1.00	1
119750	0.00	0.00	0.00	0
120000	0.00	0.00	0.00	1
120500	0.00	0.00	0.00	0
121600	1.00	1.00	1.00	1
122000	0.00	0.00	0.00	0
122500	0.00	0.00	0.00	1
124000	0.00	0.00	0.00	2
125000	0.00	0.00	0.00	1
125500	0.00	0.00	0.00	1
126000	0.00	0.00	0.00	1
126175	1.00	1.00	1.00	1
127000	0.00	0.00	0.00	3
128000	0.00	0.00	0.00	1
128500	0.33	0.50	0.40	2
128900	0.00	0.00	0.00	0
128950	1.00	1.00	1.00	1
129000	0.00	0.00	0.00	2
129900	0.00	0.00	0.00	1
130000	0.00	0.00	0.00	2
130500	0.00	0.00	0.00	1
131500	0.00	0.00	0.00	1
132000	0.00	0.00	0.00	2
132250	1.00	1.00	1.00	1
132500	0.00	0.00	0.00	1
133000	0.00	0.00	0.00	1
134500	0.00	0.00	0.00	1
134800	1.00	1.00	1.00	1
135000	0.00	0.00	0.00	4
135900	1.00	1.00	1.00	1
135960	0.00	0.00	0.00	0
136000	0.00	0.00	0.00	1
137000	0.00	0.00	0.00	3
137500	0.00	0.00	0.00	1
139000	0.00	0.00	0.00	4
140000	0.00	0.00	0.00	5
141000	0.00	0.00	0.00	2
142000	0.00	0.00	0.00	2
142500	0.00	0.00	0.00	1
142953	1.00	1.00	1.00	1
143000	0.00	0.00	0.00	1
144000	0.00	0.00	0.00	2
144152	1.00	1.00	1.00	1
145000	0.00	0.00	0.00	7

145250	1.00	1.00	1.00	1
146800	0.50	1.00	0.67	1
147400	1.00	1.00	1.00	1
148000	0.00	0.00	0.00	1
148500	0.50	1.00	0.67	1
149350	0.50	1.00	0.67	1
149500	0.00	0.00	0.00	1
150500	1.00	1.00	1.00	1
150750	0.50	1.00	0.67	1
152000	0.00	0.00	0.00	1
153337	0.00	0.00	0.00	0
153500	1.00	1.00	1.00	1
153900	0.00	0.00	0.00	1
155000	0.00	0.00	0.00	3
155900	0.50	1.00	0.67	1
156000	0.00	0.00	0.00	1
157000	0.00	0.00	0.00	1
157500	1.00	1.00	1.00	1
158000	0.00	0.00	0.00	2
158500	1.00	1.00	1.00	1
159000	0.00	0.00	0.00	1
159500	1.00	0.50	0.67	2
160000	1.00	0.25	0.40	4
162000	0.00	0.00	0.00	0
162900	0.50	1.00	0.67	1
163000	0.00	0.00	0.00	1
164500	0.00	0.00	0.00	1
164900	1.00	1.00	1.00	1
165000	0.00	0.00	0.00	1
165150	0.00	0.00	0.00	1
165400	1.00	1.00	1.00	1
165500	1.00	1.00	1.00	1
166000	1.00	1.00	1.00	1
168000	0.00	0.00	0.00	1
168500	0.00	0.00	0.00	1
169000	0.00	0.00	0.00	1
169500	0.00	0.00	0.00	1
171750	1.00	1.00	1.00	1
171900	0.00	0.00	0.00	0
172400	0.00	0.00	0.00	0
172785	1.00	1.00	1.00	1
173000	0.00	0.00	0.00	2
173900	0.00	0.00	0.00	0
174000	0.00	0.00	0.00	2
175000	0.00	0.00	0.00	1
175900	0.00	0.00	0.00	1
176000	0.00	0.00	0.00	2
177500	0.00	0.00	0.00	1
178000	0.00	0.00	0.00	2
178740	0.00	0.00	0.00	0
179000	0.00	0.00	0.00	2
179200	1.00	1.00	1.00	1
179400	0.00	0.00	0.00	0
179540	0.50	1.00	0.67	1
179600	0.50	1.00	0.67	1
179900	0.00	0.00	0.00	2
180000	0.00	0.00	0.00	2
180500	0.00	0.00	0.00	2
183500	0.50	1.00	0.67	1
184000	1.00	0.50	0.67	2
185000	0.00	0.00	0.00	2
185500	1.00	1.00	1.00	1
187000	1.00	1.00	1.00	1
187500	0.00	0.00	0.00	2
188500	0.00	0.00	0.00	0
189000	0.00	0.00	0.00	1
189950	1.00	1.00	1.00	1
190000	0.00	0.00	0.00	3
191000	1.00	0.50	0.67	2
192000	1.00	1.00	1.00	1
192500	0.00	0.00	0.00	0
194201	1.00	1.00	1.00	1
194500	0.00	0.00	0.00	1
196000	0.00	0.00	0.00	1
196500	0.00	0.00	0.00	1
197000	1.00	1.00	1.00	1
197500	0.00	0.00	0.00	1
197900	1.00	1.00	1.00	1
200000	0.00	0.00	0.00	2
200141	0.00	0.00	0.00	0
200500	0.00	0.00	0.00	0
201000	0.00	0.00	0.00	0
204000	1.00	1.00	1.00	1
206300	1.00	1.00	1.00	1
207000	1.00	1.00	1.00	1
207500	0.00	0.00	0.00	1
208500	1.00	1.00	1.00	1
211000	0.00	0.00	0.00	1
212000	1.00	1.00	1.00	1
212900	0.00	0.00	0.00	0
213000	1.00	0.50	0.67	2
213490	0.00	0.00	0.00	1
214000	0.00	0.00	0.00	2
214900	0.00	0.00	0.00	1
216837	1.00	1.00	1.00	1
219500	1.00	1.00	1.00	2

```
      220000       1.00      0.33      0.50         3
      222000       0.00      0.00      0.00         1
      222500       0.00      0.00      0.00         1
      223500       1.00      1.00      1.00         1
      225000       0.00      0.00      0.00         2
      230000       0.00      0.00      0.00         2
      232000       0.00      0.00      0.00         1
      234000       0.00      0.00      0.00         1
      236000       1.00      1.00      1.00         1
      236500       1.00      1.00      1.00         1
      239000       1.00      0.50      0.67         2
      244000       0.00      0.00      0.00         1
      246578       0.50      1.00      0.67         1
      248000       1.00      1.00      1.00         1
      250000       0.00      0.00      0.00         2
      252000       0.50      1.00      0.67         1
      252678       1.00      1.00      1.00         1
      255000       1.00      1.00      1.00         1
      255500       1.00      1.00      1.00         1
      256000       0.00      0.00      0.00         1
      256300       0.00      0.00      0.00         1
      257000       1.00      1.00      1.00         1
      259500       1.00      1.00      1.00         1
      260000       0.00      0.00      0.00         1
      263000       0.00      0.00      0.00         0
      263435       1.00      1.00      1.00         1
      264561       0.00      0.00      0.00         1
      265979       1.00      1.00      1.00         1
      268000       0.00      0.00      0.00         1
      269500       1.00      1.00      1.00         1
      274000       1.00      1.00      1.00         1
      285000       1.00      1.00      1.00         1
      286000       1.00      1.00      1.00         1
      287000       1.00      1.00      1.00         1
      290000       1.00      0.50      0.67         2
      295000       1.00      1.00      1.00         1
      301500       1.00      1.00      1.00         1
      302000       1.00      1.00      1.00         1
      303477       1.00      1.00      1.00         1
      306000       0.00      0.00      0.00         0
      307000       0.00      0.00      0.00         0
      311500       1.00      1.00      1.00         1
      314813       1.00      1.00      1.00         1
      315500       1.00      1.00      1.00         1
      315750       1.00      1.00      1.00         1
      318000       0.00      0.00      0.00         1
      319900       1.00      1.00      1.00         1
      325300       1.00      1.00      1.00         1
      328000       1.00      1.00      1.00         1
      335000       1.00      1.00      1.00         1
      359100       1.00      1.00      1.00         1
      372500       1.00      1.00      1.00         1
      377426       1.00      1.00      1.00         1
      377500       1.00      1.00      1.00         1
      386250       1.00      1.00      1.00         1
      392000       1.00      1.00      1.00         1
      392500       1.00      1.00      1.00         1
      402861       1.00      1.00      1.00         1
      412500       1.00      1.00      1.00         1
      424870       1.00      1.00      1.00         1
      475000       1.00      1.00      1.00         1
      582933       1.00      1.00      1.00         1
      745000       0.00      0.00      0.00         0
      755000       0.00      0.00      0.00         0

    accuracy                           0.41       292
   macro avg       0.44      0.45      0.43       292
weighted avg       0.43      0.41      0.40       292


[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\metrics\classification.py:1437: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.

C:\Users\piash\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\metrics\classification.py:1439: UndefinedMetricWarning:

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

Saving the Model

Since Random Forest has the highest accuracy rate, I am saving this model

In [281]:
```
# Saving Model
import pickle
saved_model = pickle.dumps(bbc_rf)
```

In [282]:
```
# Load the Pickled model
bbc_rf_from_pickle = pickle.loads(saved_model)
```

In [283]:
```
# Using the loaded pickle model to make predictions
df_test['SalePrice']= bbc_rf_from_pickle.predict(df_test)
```

In [284]:
```
df_test_copy['SalePrice']=df_test['SalePrice']
```

In [220]:
```
df_test['SalePrice']=df_test['SalePrice']
```

In [297]:
```
df_test.head(5)
```

Out[297]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtQual	BsmtFinSF1	...	SaleType_ConLw	SaleType_New	SaleType
0	20	80.00	11622	5	6	1961	1961	0.00	3	468.00	...	0	0	
1	20	81.00	14267	6	6	1958	1958	108.00	3	923.00	...	0	0	
2	60	74.00	13830	5	5	1997	1998	0.00	4	791.00	...	0	0	
3	60	78.00	9978	6	6	1998	1998	20.00	3	602.00	...	0	0	
4	120	43.00	5005	8	5	1992	1992	0.00	4	263.00	...	0	0	

5 rows × 215 columns

In [286]:
```
df_test_copy[['Id','SalePrice']]
```

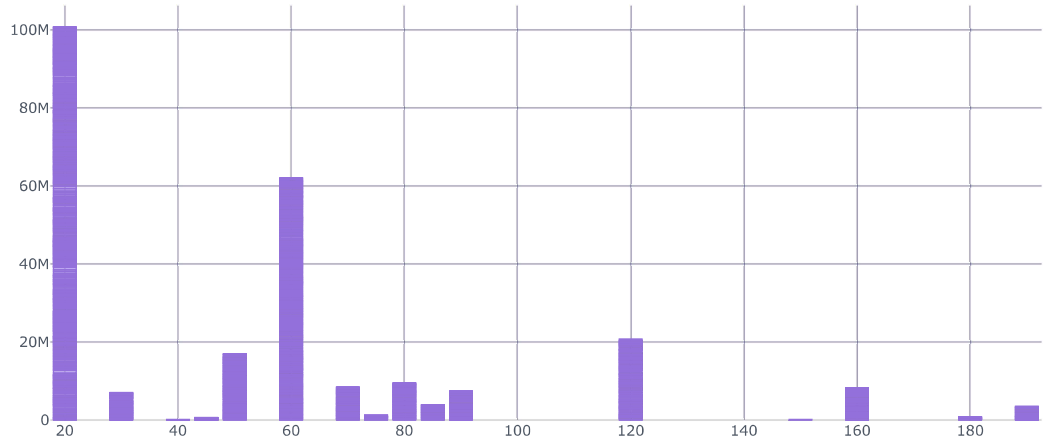
Out[286]:

	Id	SalePrice
0	1461	161500
1	1462	151500
2	1463	188500
3	1464	186500
4	1465	236500
...	...	...
1454	2915	81000
1455	2916	75000
1456	2917	147000
1457	2918	93500
1458	2919	165600

1459 rows × 2 columns

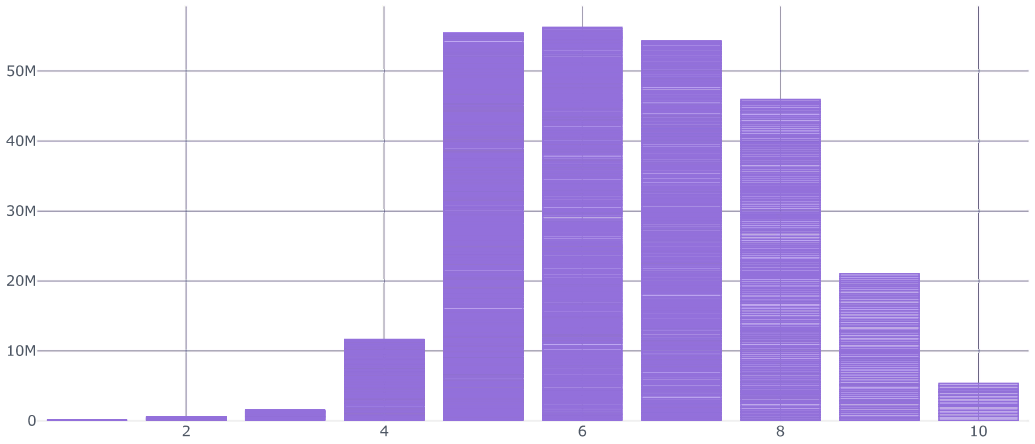
Exploratory Data Analysis With Test Dataset

In [298]:
```
df_test.iplot(kind='bar',x='MSSubClass',y='SalePrice',gridcolor='blue',bins=30, colors='mediumpurple')
```



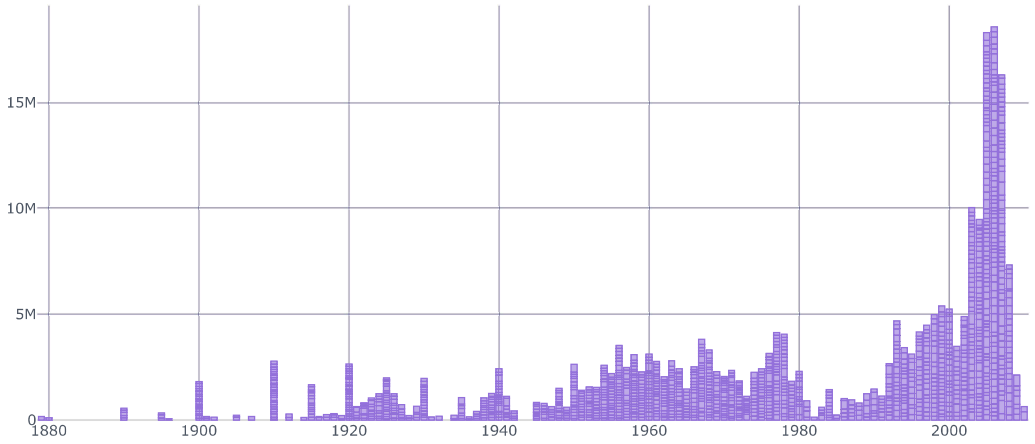
Export to plot.ly »

In [299]:
```
df_test.iplot(kind='bar',x='OverallQual',y='SalePrice',gridcolor='blue',bins=30, colors='mediumpurple')
```



Export to plot.ly »

In [300]:
```
df_test.iplot(kind='bar',x='YearBuilt',y='SalePrice',gridcolor='blue',bins=30, colors='mediumpurple')
```



Export to plot.ly »