import pandas as pd

df = df = pd.read\_csv("/train.csv")

pd.set\_option('display.max\_columns', None)

df

df.info()

df.describe()

df.isnull()

df.columns[round((df.isnull().sum()/len(df.index))\*100,1) > 0]

missing\_columns = df.columns[100\*(df.isnull().sum()/len(df.index)) > 80]

print(missing\_columns)

df.drop(missing\_columns)

df.describe()

import pandas as pd

import numpy as np

import seaborn as sns

from matplotlib import pyplot as plt

sns.distplot(df['SalePrice'])

plt.show()

import seaborn as sns

%matplotlib inline

sns.pairplot(df, x\_vars="WoodDeckSF", y\_vars="SalePrice", height= 5, aspect =0.7, kind='scatter' )

%matplotlib inline

sns.pairplot(df, x\_vars="WoodDeckSF", y\_vars="LotFrontage", height= 5, aspect =0.7, kind='scatter' )

%matplotlib inline

sns.pairplot(df, x\_vars="WoodDeckSF", y\_vars="YearBuilt", height= 5, aspect =0.7, kind='scatter' )

%matplotlib inline

sns.pairplot(df, x\_vars="WoodDeckSF", y\_vars="BsmtUnfSF", height= 5, aspect =0.7, kind='scatter' )

df.drop(["Id"], axis=1)

df["HouseAge"] = df["YrSold"] - df["YearBuilt"]

df

df["Remodel"] = df.apply(lambda x: 0 if (x['YearRemodAdd'] - x['YearBuilt'] == 0) else 1, axis=1)

df

df["GarageAge"] = df["YrSold"] - df["GarageYrBlt"]

df

df.drop(["YrSold", "GarageYrBlt", "YearBuilt", "YearRemodAdd"], axis =1, inplace = True)

df

print(df["GarageQual"].value\_counts())

print(df["GarageQual"].value\_counts())

df["GarageQual"] = np.where(df.groupby('GarageQual')['GarageQual'].transform('size') <=100 , 'Other', df['GarageQual'])

print(df["GarageQual"].value\_counts())

print(df["OverallCond"].value\_counts())

print(df["OverallQual"].value\_counts())

df["OverallCond"] = df["OverallCond"].map(lambda i: "Avg" if i <=5 else ("Good" if i >5 and i <=7 else "Excellent"))

df["OverallQual"] = df["OverallQual"].map(lambda i: "Avg" if i <=5 else "Good" if i >5 and i <=7 else "Excellent")

print(df["OverallCond"].value\_counts())

print(df["OverallQual"].value\_counts())

df\_categorical = df.select\_dtypes(include=['object'])

categorical\_col = df\_categorical.columns

print(categorical\_col)

df\_numeric = df.select\_dtypes(include=['int64', 'float64'])

numeric\_col = df\_numeric.columns

print(numeric\_col)

sns.pairplot(df\_numeric)

plt.show()