House Prices : Advanced Regression Techniques

Aim: Predict the sale price of a house

Features (80) :

MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Condition2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMatl, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, ExterCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, Heating, HeatingQC, CentralAir, Electrical, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, Bedroom, Kitchen, KitchenQual, TotRmsAbvGrd, Functional, Fireplaces, FireplaceQu, GarageType, GarageYrBlt, GarageFinish, GarageCars, GarageArea, GarageQual, GarageCond, PavedDrive, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, PoolQC, Fence, MiscFeature, MiscVal, MoSold, YrSold, SaleType, SaleCondition

Kaggle dataset: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

In [ ]:
```
# import necessary libraries

import pandas as pd
import sys
import numpy as np
import seaborn as sns
from math import sqrt
from pylab import rcParams

from sklearn import metrics
from sklearn.metrics import mean_squared_error
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV

import sklearn
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import ElasticNet, Lasso
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.kernel_ridge import KernelRidge

from sklearn.ensemble import StackingRegressor

from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import RobustScaler
from sklearn.model_selection import KFold, cross_val_score, train_test_split
```

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler

%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
```

---------- 1. LOADING & LOOKING AT THE DATA ----------

- The housing dataset is available on Kaggle under "House Prices: Advanced Regression Techniques". The "train.csv" file contains the training data and "test.csv" contains the testing data. The training data contains data for 1460 rows which corresponds to 1460 house's data and 80 columns which correspond to the feature of those houses. Similarly, the testing data contains data of 1461 houses and their 79 attributes.

In [ ]:
```
# load dataset
csv_path = "train.csv"
df_train = pd.read_csv(csv_path, sep = ',')

csv_path = "test.csv"
df_test = pd.read_csv(csv_path, sep = ',')
```

In [ ]:
```
# check shape
print(df_train.shape)
print(df_test.shape)
```

```
(1460, 81)
(1459, 80)
```

In [ ]:
```
# look a first 10 rows of training data
df_train.head(10)
```

Out[ ]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandCor
0	1	60	RL	65.0	8450	Pave	NaN	Reg	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	
5	6	50	RL	85.0	14115	Pave	NaN	IR1	
6	7	20	RL	75.0	10084	Pave	NaN	Reg	
7	8	60	RL	NaN	10382	Pave	NaN	IR1	
8	9	50	RM	51.0	6120	Pave	NaN	Reg	
9	10	190	RL	50.0	7420	Pave	NaN	Reg	

10 rows × 81 columns

In [ ]:
```
# look a first 10 rows of testing data
df_test.head(10)
```

Out[ ]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandC
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	
5	1466	60	RL	75.0	10000	Pave	NaN	IR1	
6	1467	20	RL	NaN	7980	Pave	NaN	IR1	
7	1468	60	RL	63.0	8402	Pave	NaN	IR1	
8	1469	20	RL	85.0	10176	Pave	NaN	Reg	
9	1470	20	RL	70.0	8400	Pave	NaN	Reg	

10 rows × 80 columns

In [ ]:
```
# see all the column names
df_train.columns
```

Out[ ]:
```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             1460 non-null   int64  
 1   MSSubClass     1460 non-null   int64  
 2   MSZoning       1460 non-null   object 
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64  
 5   Street         1460 non-null   object 
 6   Alley          91 non-null     object 
 7   LotShape       1460 non-null   object 
 8   LandContour    1460 non-null   object 
 9   Utilities      1460 non-null   object 
 10  LotConfig      1460 non-null   object 
 11  LandSlope      1460 non-null   object 
 12  Neighborhood   1460 non-null   object 
 13  Condition1     1460 non-null   object 
 14  Condition2     1460 non-null   object 
 15  BldgType       1460 non-null   object 
 16  HouseStyle     1460 non-null   object 
 17  OverallQual    1460 non-null   int64  
 18  OverallCond    1460 non-null   int64  
 19  YearBuilt      1460 non-null   int64  
 20  YearRemodAdd   1460 non-null   int64  
 21  RoofStyle      1460 non-null   object 
 22  RoofMatl       1460 non-null   object 
 23  Exterior1st    1460 non-null   object 
 24  Exterior2nd    1460 non-null   object 
 25  MasVnrType     1452 non-null   object 
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object 
 28  ExterCond      1460 non-null   object 
 29  Foundation     1460 non-null   object 
 30  BsmtQual       1423 non-null   object 
 31  BsmtCond       1423 non-null   object 
 32  BsmtExposure   1422 non-null   object 
 33  BsmtFinType1   1423 non-null   object 
 34  BsmtFinSF1     1460 non-null   int64  
 35  BsmtFinType2   1422 non-null   object 
 36  BsmtFinSF2     1460 non-null   int64  
 37  BsmtUnfSF      1460 non-null   int64  
 38  TotalBsmtSF    1460 non-null   int64  
 39  Heating        1460 non-null   object 
 40  HeatingQC      1460 non-null   object 
 41  CentralAir     1460 non-null   object 
 42  Electrical     1459 non-null   object 
 43  1stFlrSF       1460 non-null   int64  
 44  2ndFlrSF       1460 non-null   int64  
 45  LowQualFinSF   1460 non-null   int64  
 46  GrLivArea      1460 non-null   int64  
 47  BsmtFullBath   1460 non-null   int64  
 48  BsmtHalfBath   1460 non-null   int64  
 49  FullBath       1460 non-null   int64  
 50  HalfBath       1460 non-null   int64  
 51  BedroomAbvGr   1460 non-null   int64  
 52  KitchenAbvGr   1460 non-null   int64  
 53  KitchenQual    1460 non-null   object 
 54  TotRmsAbvGrd   1460 non-null   int64  
```

```
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

- There are 1460 rows and 81 columns
- There are columns with large number of null entries like PoolQC, MiscFeature
- The columns have Three types of datatypes: float64(3), int64(35), object(43)

In [ ]:
```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 80 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             1459 non-null   int64  
 1   MSSubClass     1459 non-null   int64  
 2   MSZoning       1455 non-null   object 
 3   LotFrontage    1232 non-null   float64
 4   LotArea        1459 non-null   int64  
 5   Street         1459 non-null   object 
 6   Alley          107 non-null    object 
 7   LotShape       1459 non-null   object 
 8   LandContour    1459 non-null   object 
 9   Utilities      1457 non-null   object 
 10  LotConfig      1459 non-null   object 
 11  LandSlope      1459 non-null   object 
 12  Neighborhood   1459 non-null   object 
 13  Condition1     1459 non-null   object 
 14  Condition2     1459 non-null   object 
 15  BldgType       1459 non-null   object 
 16  HouseStyle     1459 non-null   object 
 17  OverallQual    1459 non-null   int64  
 18  OverallCond    1459 non-null   int64  
 19  YearBuilt      1459 non-null   int64  
 20  YearRemodAdd   1459 non-null   int64  
 21  RoofStyle      1459 non-null   object 
 22  RoofMatl       1459 non-null   object 
 23  Exterior1st    1458 non-null   object 
 24  Exterior2nd    1458 non-null   object 
 25  MasVnrType     1443 non-null   object 
 26  MasVnrArea     1444 non-null   float64
 27  ExterQual      1459 non-null   object 
 28  ExterCond      1459 non-null   object 
 29  Foundation     1459 non-null   object 
 30  BsmtQual       1415 non-null   object 
 31  BsmtCond       1414 non-null   object 
 32  BsmtExposure   1415 non-null   object 
 33  BsmtFinType1   1417 non-null   object 
 34  BsmtFinSF1     1458 non-null   float64
 35  BsmtFinType2   1417 non-null   object 
 36  BsmtFinSF2     1458 non-null   float64
 37  BsmtUnfSF      1458 non-null   float64
 38  TotalBsmtSF    1458 non-null   float64
 39  Heating        1459 non-null   object 
 40  HeatingQC      1459 non-null   object 
 41  CentralAir     1459 non-null   object 
 42  Electrical     1459 non-null   object 
 43  1stFlrSF       1459 non-null   int64  
 44  2ndFlrSF       1459 non-null   int64  
 45  LowQualFinSF   1459 non-null   int64  
 46  GrLivArea      1459 non-null   int64  
 47  BsmtFullBath   1457 non-null   float64
 48  BsmtHalfBath   1457 non-null   float64
 49  FullBath       1459 non-null   int64  
 50  HalfBath       1459 non-null   int64  
 51  BedroomAbvGr   1459 non-null   int64  
 52  KitchenAbvGr   1459 non-null   int64  
 53  KitchenQual    1458 non-null   object 
 54  TotRmsAbvGrd   1459 non-null   int64  
```

```
 55  Functional     1457 non-null   object
 56  Fireplaces     1459 non-null   int64
 57  FireplaceQu    729 non-null    object
 58  GarageType     1383 non-null   object
 59  GarageYrBlt    1381 non-null   float64
 60  GarageFinish   1381 non-null   object
 61  GarageCars     1458 non-null   float64
 62  GarageArea     1458 non-null   float64
 63  GarageQual     1381 non-null   object
 64  GarageCond     1381 non-null   object
 65  PavedDrive     1459 non-null   object
 66  WoodDeckSF     1459 non-null   int64
 67  OpenPorchSF    1459 non-null   int64
 68  EnclosedPorch  1459 non-null   int64
 69  3SsnPorch      1459 non-null   int64
 70  ScreenPorch    1459 non-null   int64
 71  PoolArea       1459 non-null   int64
 72  PoolQC         3 non-null      object
 73  Fence          290 non-null    object
 74  MiscFeature    51 non-null     object
 75  MiscVal        1459 non-null   int64
 76  MoSold         1459 non-null   int64
 77  YrSold         1459 non-null   int64
 78  SaleType       1458 non-null   object
 79  SaleCondition  1459 non-null   object
dtypes: float64(11), int64(26), object(43)
memory usage: 912.0+ KB
```

- There are 1459 rows and 80 columns
- There are columns with large number of null entries like PoolQC, MiscFeature etc
- The columns have Three types of datatypes: float64(11), int64(26), object(43)

Looking at the label to predict

In [ ]:
```
df_train['SalePrice'].describe()
```

Out[ ]:
```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

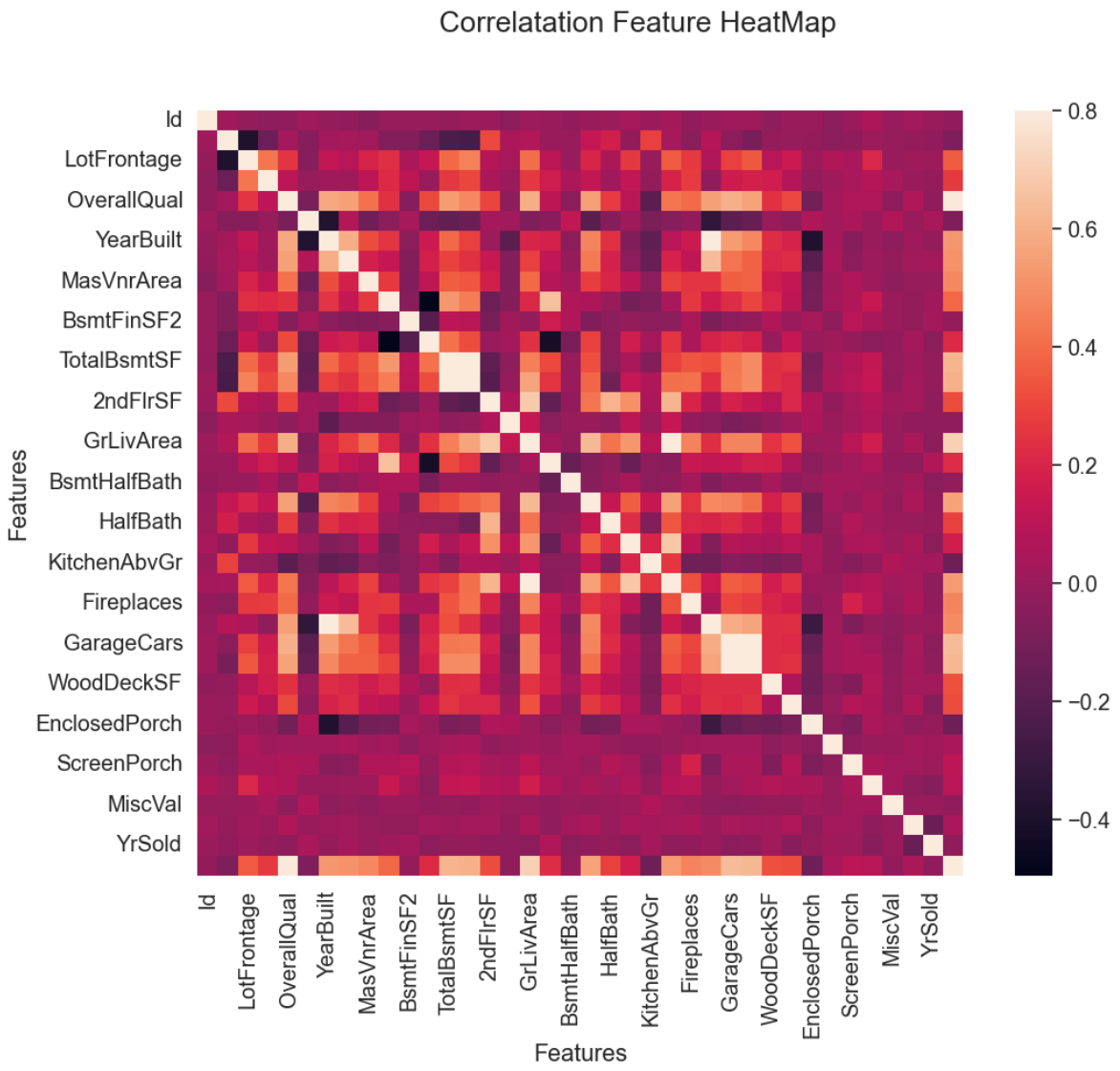
- The average SalePrice of a house is 180,921
- The Maximum SalePrice of a house is 755,000 and Minimum 34,900

In [ ]:
```
#correlation matrix
corr_mat = df_train.corr()
f, ax = plt.subplots(figsize=(12, 9))

sns.heatmap(corr_mat, vmax=.8,square=True)

plt.suptitle("Correlatation Feature HeatMap")
```

```
plt.xlabel("Features")
plt.ylabel("Features")
```

```
C:\Users\singh\AppData\Local\Temp\ipykernel_39868\1795499774.py:2: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
rsion, it will default to False. Select only valid columns or specify the value o
f numeric_only to silence this warning.
  corr_mat = df_train.corr()
```

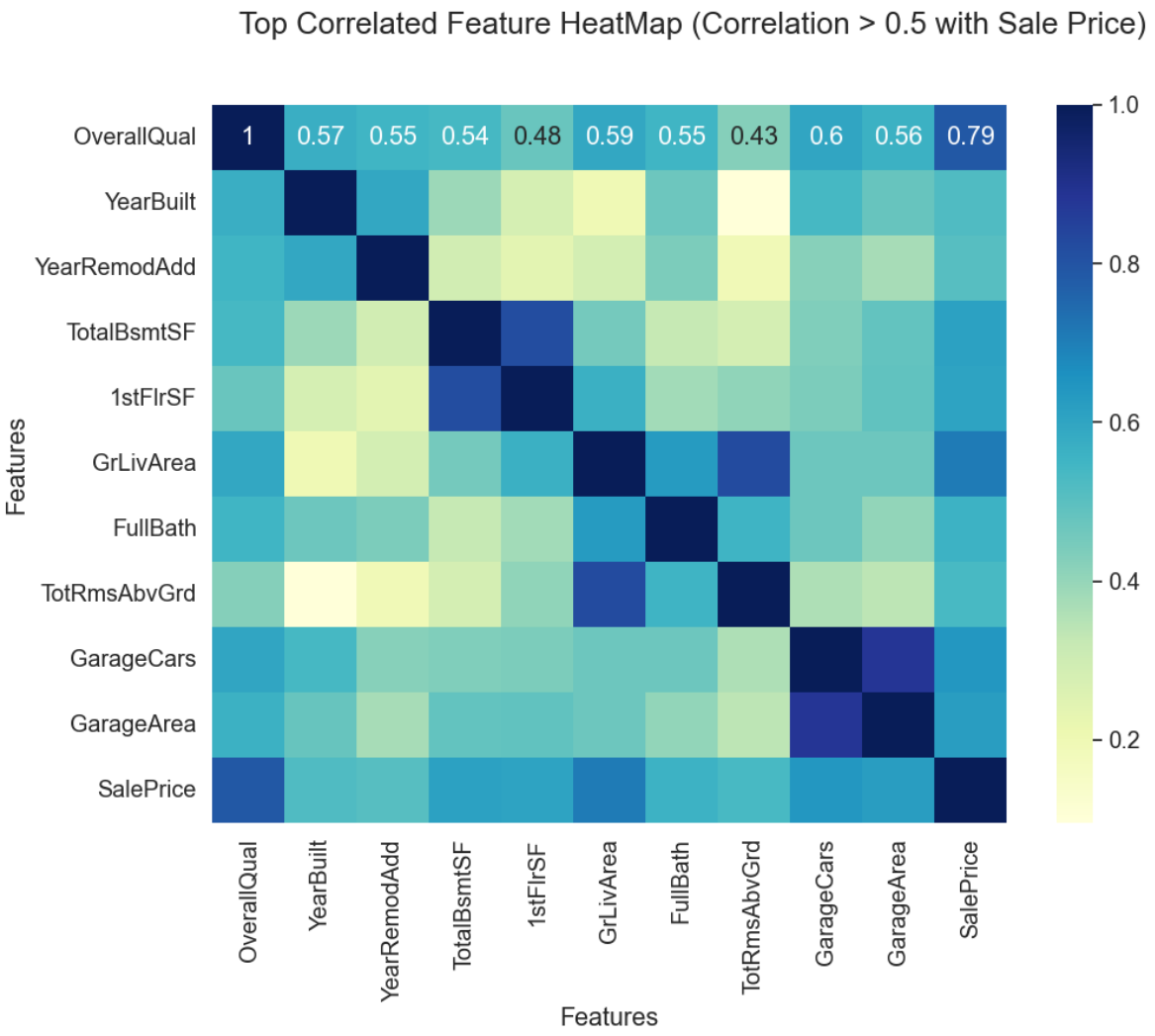
Out[ ]: Text(163.2500000000001, 0.5, 'Features')



In [ ]:
```
# most correlated features
corr_mat = df_train.corr()

sns.set(font_scale = 1.3)
plt.figure(figsize = (11,8))

top_corr = corr_mat.index[abs(corr_mat["SalePrice"])>0.5]
g = sns.heatmap(df_train[top_corr].corr(),annot=True,cmap="YlGnBu")
plt.suptitle("Top Correlated Feature HeatMap (Correlation > 0.5 with Sale Price)
plt.xlabel("Features")
plt.ylabel("Features")
```

```
C:\Users\singh\AppData\Local\Temp\ipykernel_39868\1260948267.py:2: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
rsion, it will default to False. Select only valid columns or specify the value o
f numeric_only to silence this warning.
  corr_mat = df_train.corr()
```

Out[ ]: Text(99.74999999999999, 0.5, 'Features')



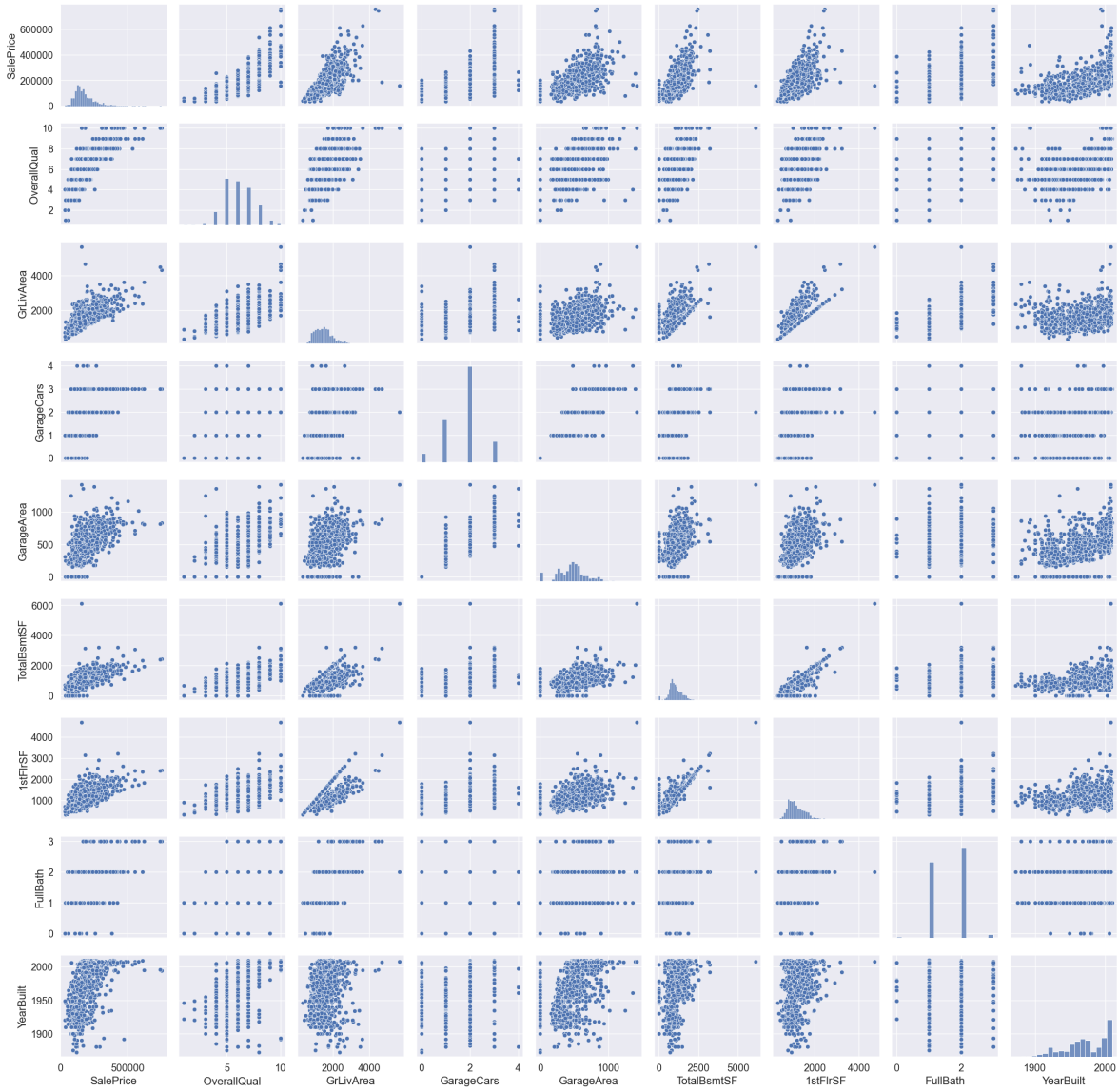
- OverallQual and GrLivArea seem to be the most correlated to SalePrice

In [ ]:
```
print("Correlation Values")

corr = df_train.corr().drop('SalePrice')
corr.sort_values(["SalePrice"], ascending = False, inplace = True)
print(corr.SalePrice)
```

```
Correlation Values
OverallQual      0.790982
GrLivArea        0.708624
GarageCars       0.640409
GarageArea       0.623431
TotalBsmtSF      0.613581
1stFlrSF         0.605852
FullBath         0.560664
TotRmsAbvGrd     0.533723
YearBuilt        0.522897
YearRemodAdd     0.507101
GarageYrBlt      0.486362
MasVnrArea       0.477493
Fireplaces       0.466929
BsmtFinSF1       0.386420
LotFrontage      0.351799
WoodDeckSF       0.324413
2ndFlrSF         0.319334
OpenPorchSF      0.315856
HalfBath         0.284108
LotArea          0.263843
BsmtFullBath     0.227122
BsmtUnfSF        0.214479
BedroomAbvGr     0.168213
ScreenPorch      0.111447
PoolArea         0.092404
MoSold           0.046432
3SsnPorch        0.044584
BsmtFinSF2      -0.011378
BsmtHalfBath    -0.016844
MiscVal         -0.021190
Id              -0.021917
LowQualFinSF    -0.025606
YrSold          -0.028923
OverallCond     -0.077856
MSSubClass      -0.084284
EnclosedPorch   -0.128578
KitchenAbvGr    -0.135907
Name: SalePrice, dtype: float64
C:\Users\singh\AppData\Local\Temp\ipykernel_39868\3756221311.py:3: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
rsion, it will default to False. Select only valid columns or specify the value o
f numeric_only to silence this warning.
  corr = df_train.corr().drop('SalePrice')
```

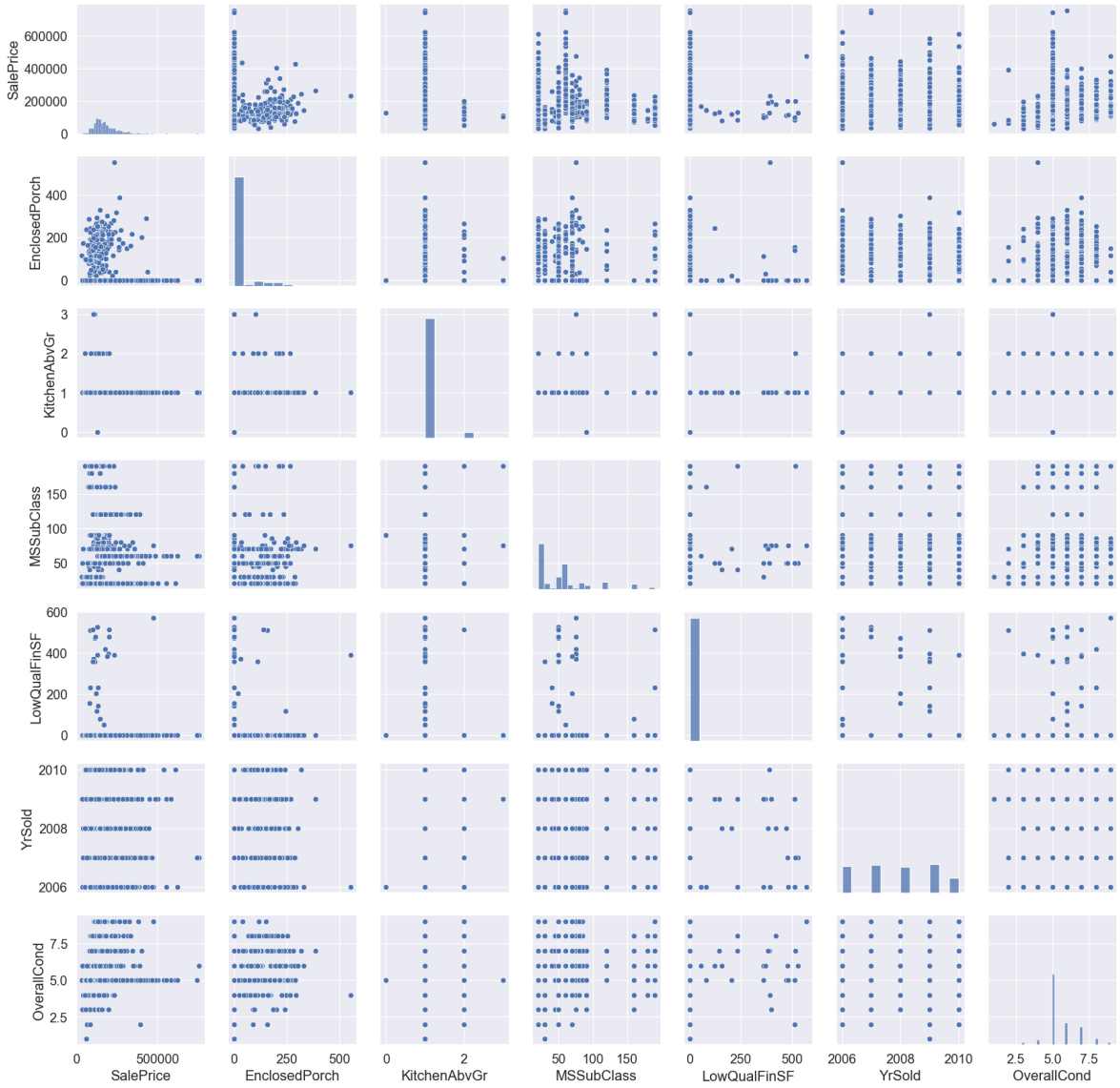
In [ ]:
```
rcParams['figure.figsize'] = 5,5
cols = ['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars','GarageArea', 'Tot
sns_plot = sns.pairplot(df_train[cols])

plt.suptitle('Scatter plots between top 9 most corr features', y=1.04, size=25)
plt.tight_layout()
plt.show()
```

Scatter plots between top 9 most corr features



In [ ]:
```
rcParams['figure.figsize'] = 5,5
cols = ['SalePrice','EnclosedPorch', 'KitchenAbvGr', 'MSSubClass', 'LowQualFinSF
sns_plot = sns.pairplot(df_train[cols])

plt.suptitle('Scatter plots between least 6  corr features', y=1.04, size=20)
plt.tight_layout()
plt.show()
```

Scatter plots between least 6 corr features



---------- 2. HANDLING DATA ----------

Drop Id Column

In [ ]:
```
#drop id as it is not required for training or prediction
train_ID = df_train['Id']
test_ID = df_test['Id']

df_train.drop(['Id'], axis=1, inplace=True)
df_test.drop(['Id'], axis=1, inplace=True)

df_train.shape, df_test.shape
```

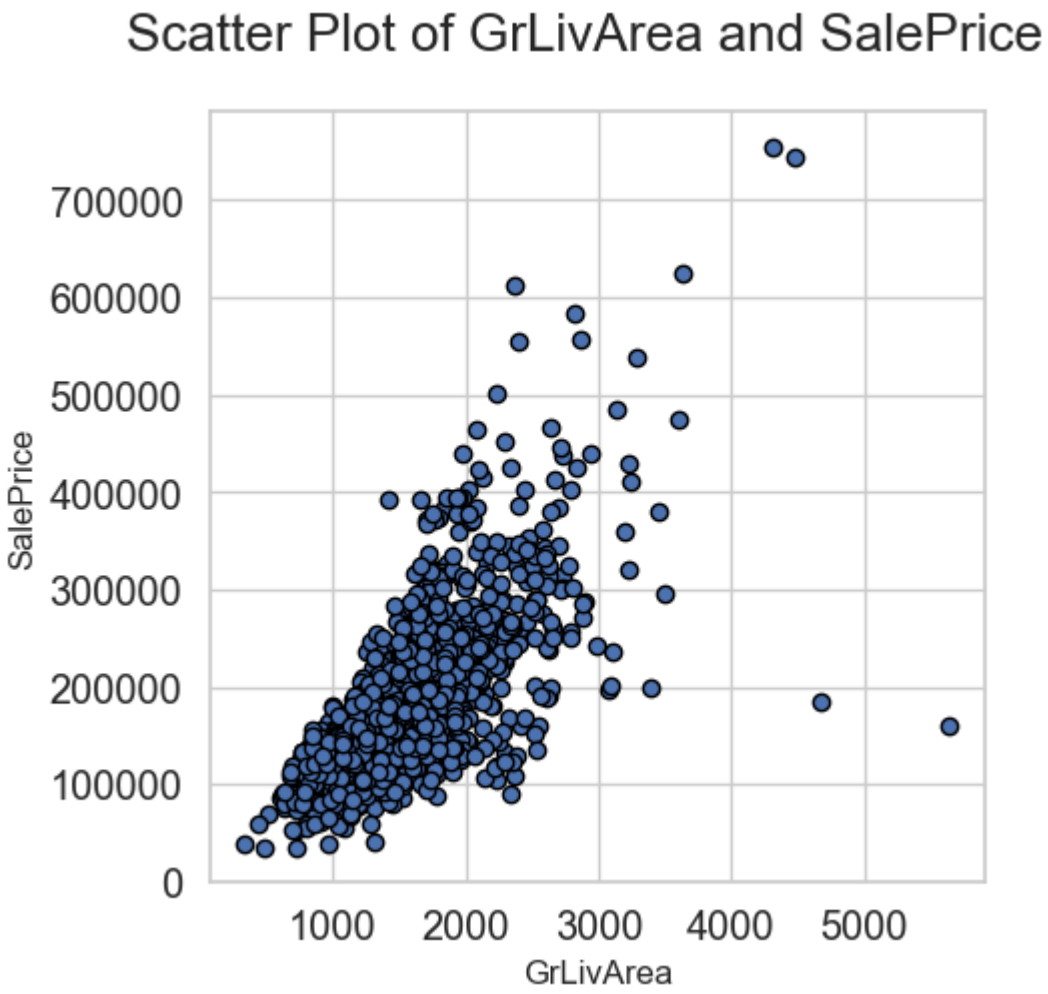
Out[ ]:
```
((1460, 80), (1459, 79))
```

Checking for Outliers

In [ ]:
```
sns.set_style('whitegrid')
edgecolor = 'black'

fig = plt.figure(figsize=(12,12))

#function to plot scatter plot between a feature and the Sale Price
def scatter_plot(a):
    fig, ax = plt.subplots()
    ax.scatter(x = df_train[a], y = df_train['SalePrice'], edgecolor=edgecolor)
    plt.ylabel('SalePrice', fontsize=12)
    plt.xlabel(a, fontsize=12)
    plt.suptitle("Scatter Plot of "+ a + " and SalePrice")
    plt.show()
```

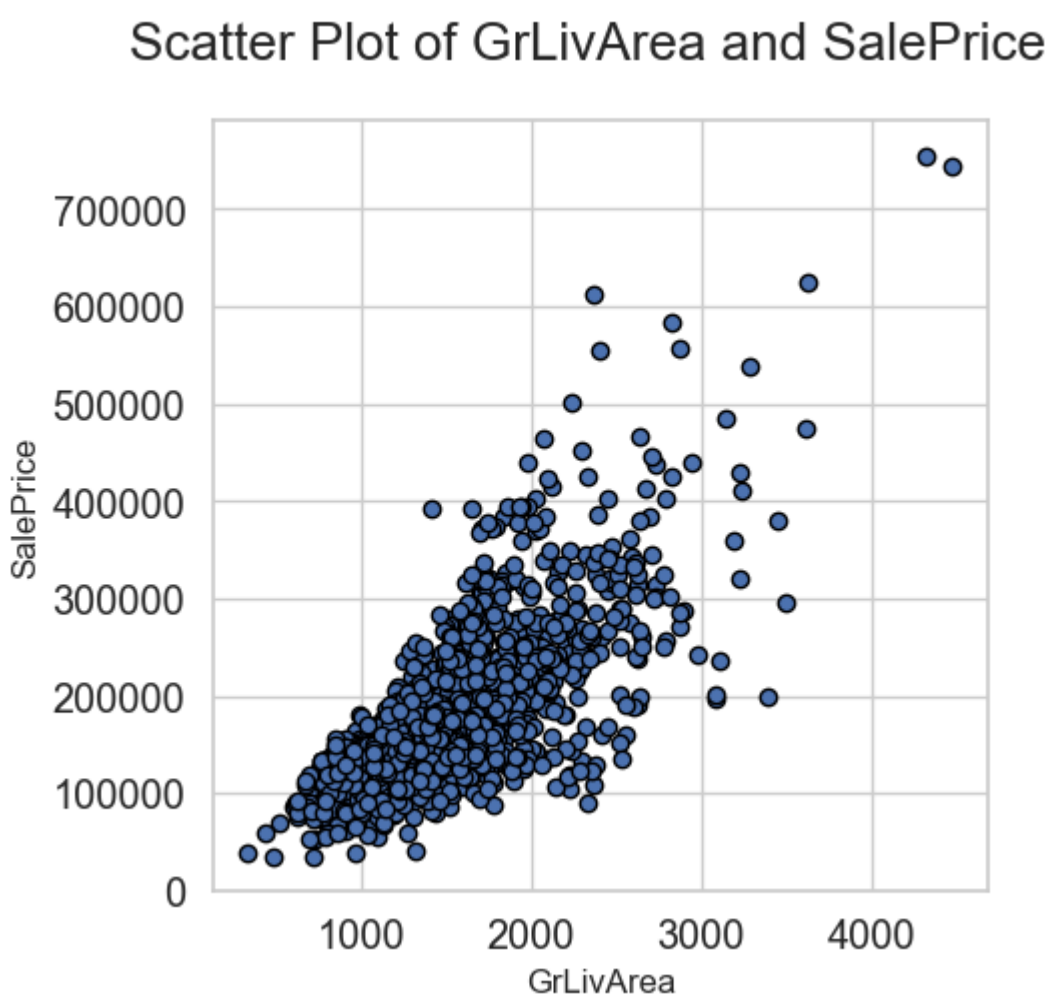
```
<Figure size 1200x1200 with 0 Axes>
```

In [ ]:
```
scatter_plot('GrLivArea')
```

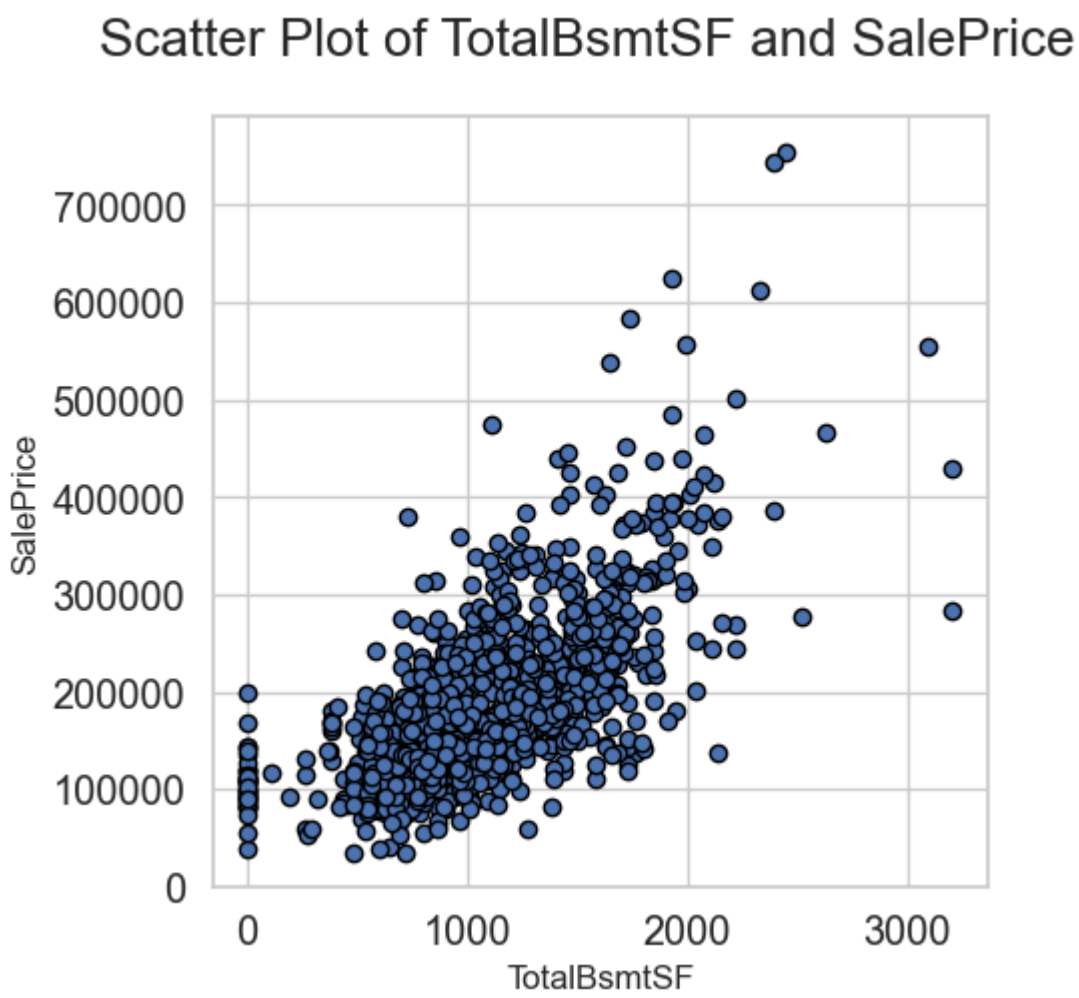


- It can be observed that there are large outliers which can negatively affect the prediction of sale price highly
- So the outliers need to be deleted

In [ ]:
```
#Deleting outliers
df_train =  df_train.drop( df_train[( df_train['GrLivArea'] > 4000) & ( df_train

#Check the graphic again
scatter_plot('GrLivArea')
```

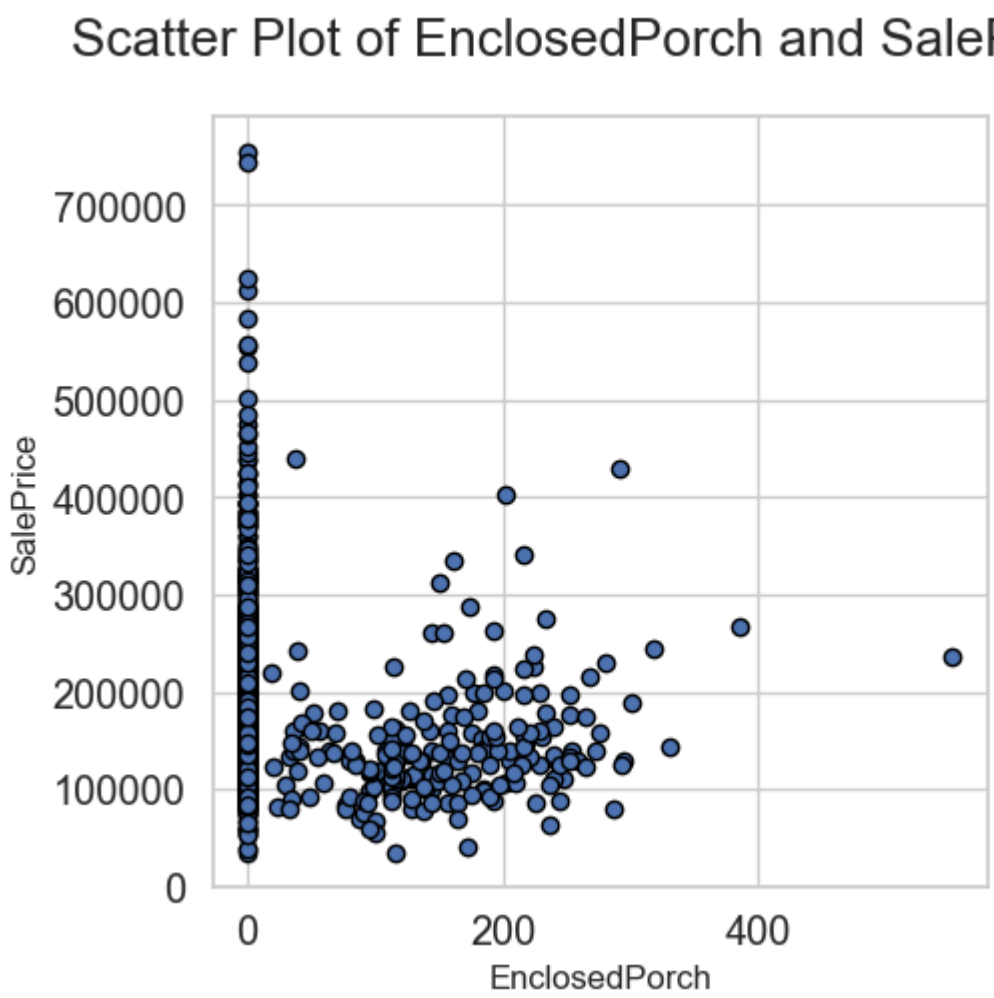


In [ ]:
```
scatter_plot('TotalBsmtSF')
```



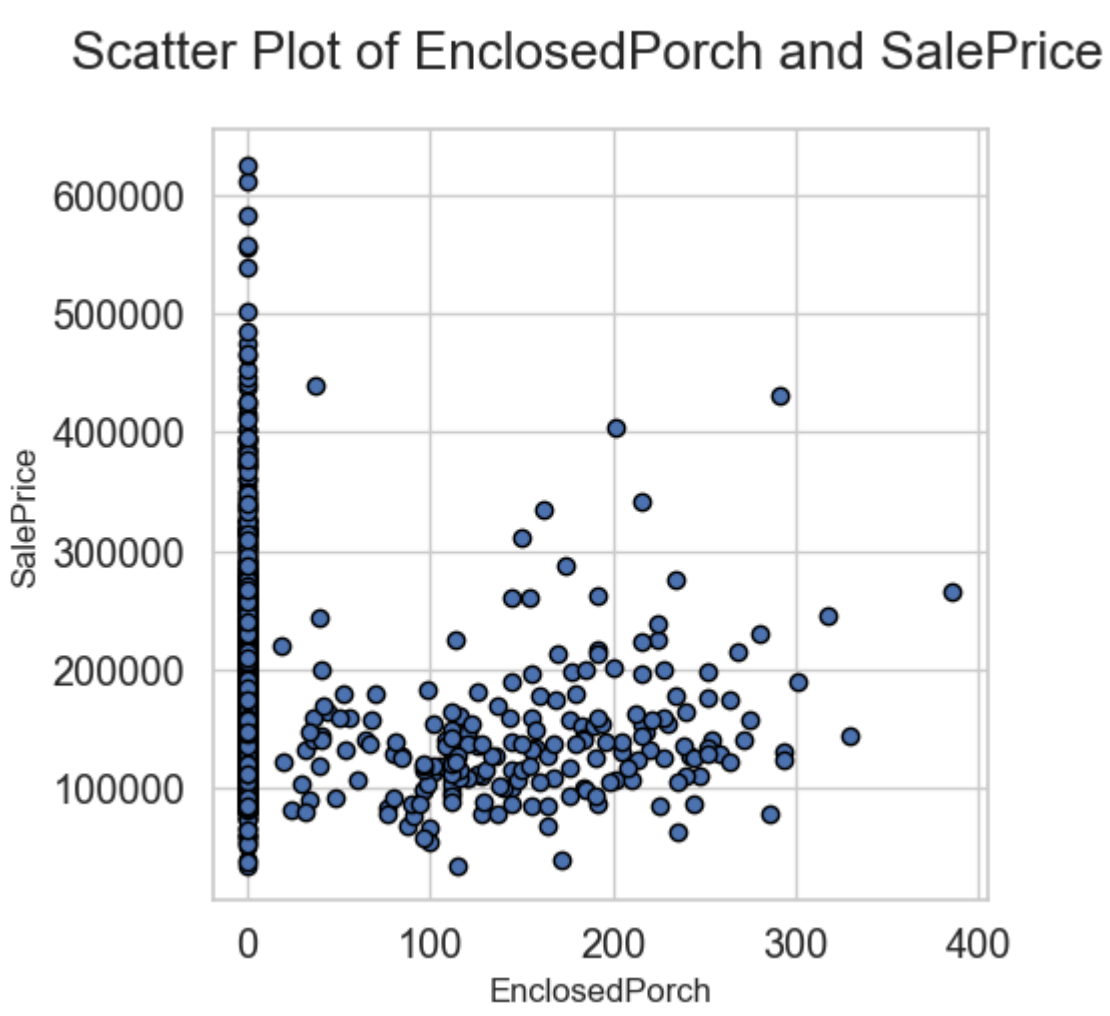
- There arent too large outliers, we do not need to delete any points

In [ ]:
```
scatter_plot('EnclosedPorch')
```



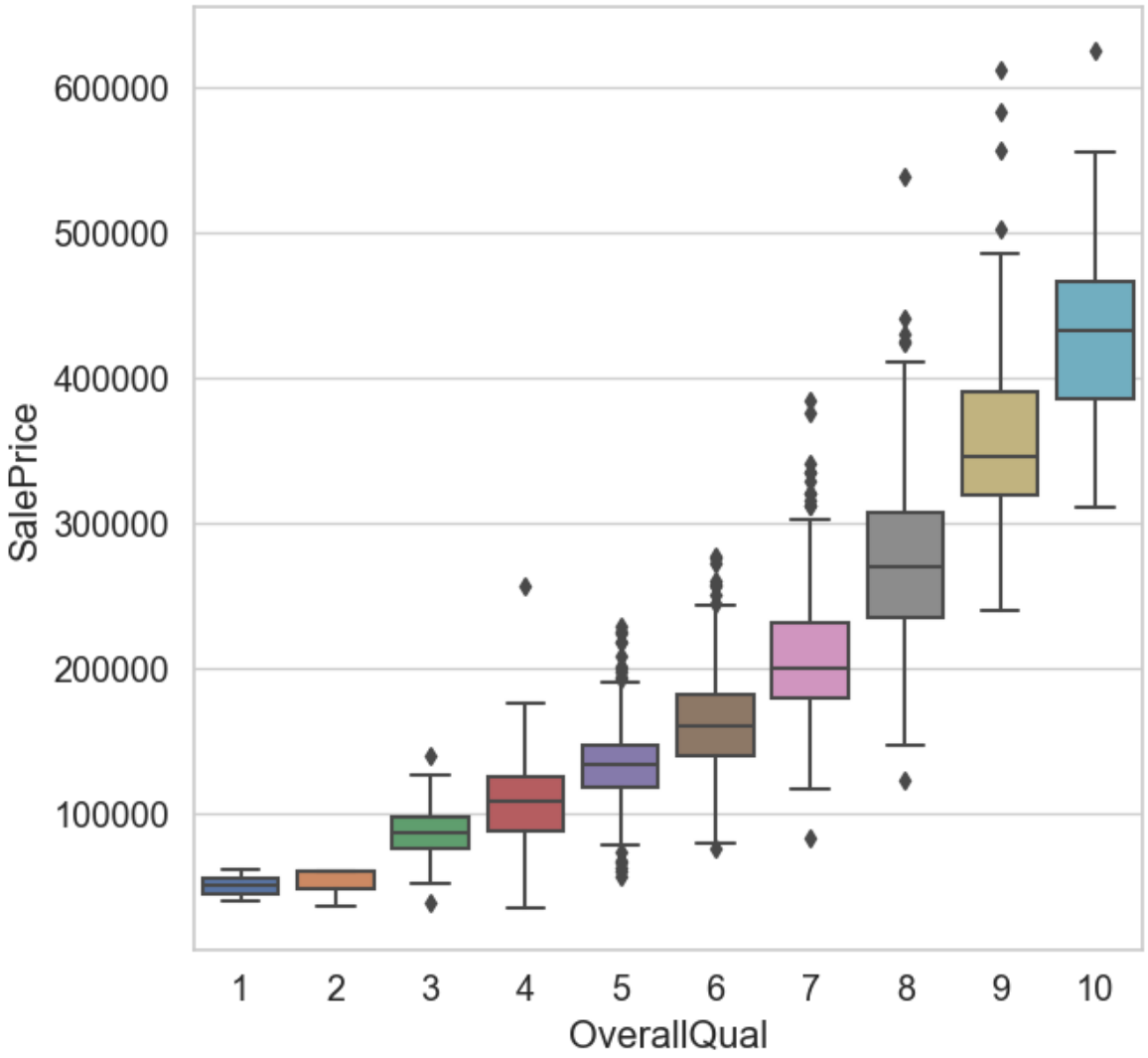
- There is are some outliers that should be deleted so that it doesnt affect our predictions much

In [ ]:
```
#Deleting outliers
df_train =  df_train.drop( df_train[( df_train['EnclosedPorch']>400)].index)

#Deleting outliers
df_train =  df_train.drop( df_train[( df_train['SalePrice']>700000)].index)

#check plot again
scatter_plot('EnclosedPorch')
```



In [ ]:
```
# plot a box plot for categorical feature : Overall Quality

fig = plt.figure(figsize=(7,7))
data = pd.concat([df_train['SalePrice'], df_train['OverallQual']], axis=1)
sns.boxplot(x = df_train['OverallQual'], y="SalePrice", data = data)
```

Out[ ]:
```
<Axes: xlabel='OverallQual', ylabel='SalePrice'>
```

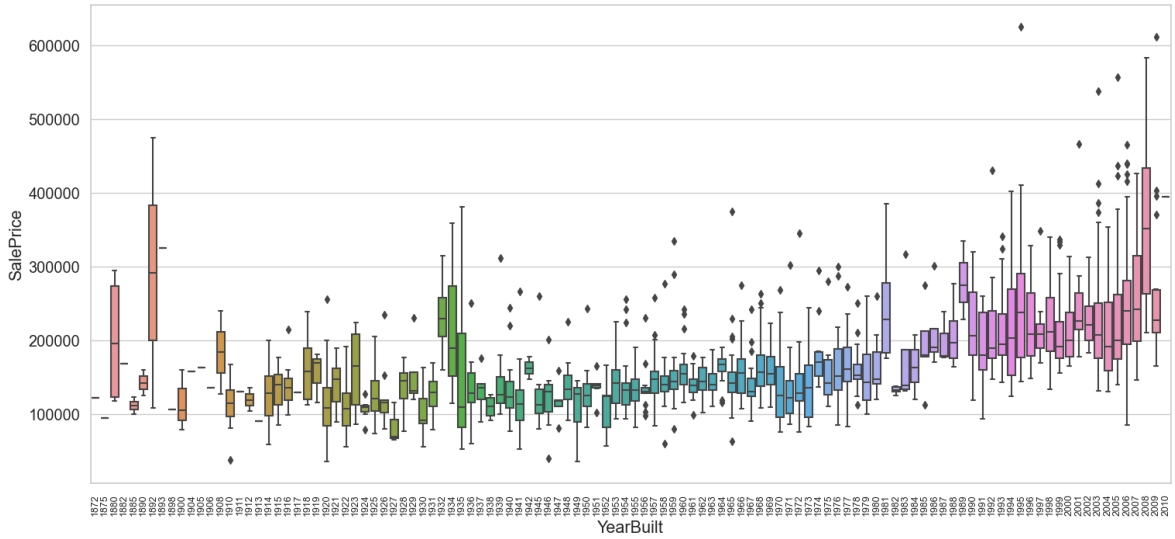


In [ ]:
```
# plot a box plot for categorical feature : Year Built
fig = plt.figure(figsize=(18,8))

data = pd.concat([df_train['SalePrice'], df_train['YearBuilt']], axis=1)
sns.boxplot(x= df_train['YearBuilt'], y="SalePrice", data=data)
plt.xticks(rotation=90,fontsize= 9)
```

Out[ ]:
```
(array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
         13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
         26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
         39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
         52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
         65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
         78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
         91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
        104, 105, 106, 107, 108, 109, 110, 111]),
 [Text(0, 0, '1872'),
  Text(1, 0, '1875'),
  Text(2, 0, '1880'),
  Text(3, 0, '1882'),
  Text(4, 0, '1885'),
  Text(5, 0, '1890'),
  Text(6, 0, '1892'),
  Text(7, 0, '1893'),
  Text(8, 0, '1898'),
  Text(9, 0, '1900'),
  Text(10, 0, '1904'),
  Text(11, 0, '1905'),
  Text(12, 0, '1906'),
  Text(13, 0, '1908'),
  Text(14, 0, '1910'),
  Text(15, 0, '1911'),
  Text(16, 0, '1912'),
  Text(17, 0, '1913'),
  Text(18, 0, '1914'),
  Text(19, 0, '1915'),
  Text(20, 0, '1916'),
  Text(21, 0, '1917'),
  Text(22, 0, '1918'),
  Text(23, 0, '1919'),
  Text(24, 0, '1920'),
  Text(25, 0, '1921'),
  Text(26, 0, '1922'),
  Text(27, 0, '1923'),
  Text(28, 0, '1924'),
  Text(29, 0, '1925'),
  Text(30, 0, '1926'),
  Text(31, 0, '1927'),
  Text(32, 0, '1928'),
  Text(33, 0, '1929'),
  Text(34, 0, '1930'),
  Text(35, 0, '1931'),
  Text(36, 0, '1932'),
  Text(37, 0, '1934'),
  Text(38, 0, '1935'),
  Text(39, 0, '1936'),
  Text(40, 0, '1937'),
  Text(41, 0, '1938'),
  Text(42, 0, '1939'),
  Text(43, 0, '1940'),
  Text(44, 0, '1941'),
  Text(45, 0, '1942'),
  Text(46, 0, '1945'),
  Text(47, 0, '1946'),
  Text(48, 0, '1947'),
  Text(49, 0, '1948'),
  Text(50, 0, '1949'),
```

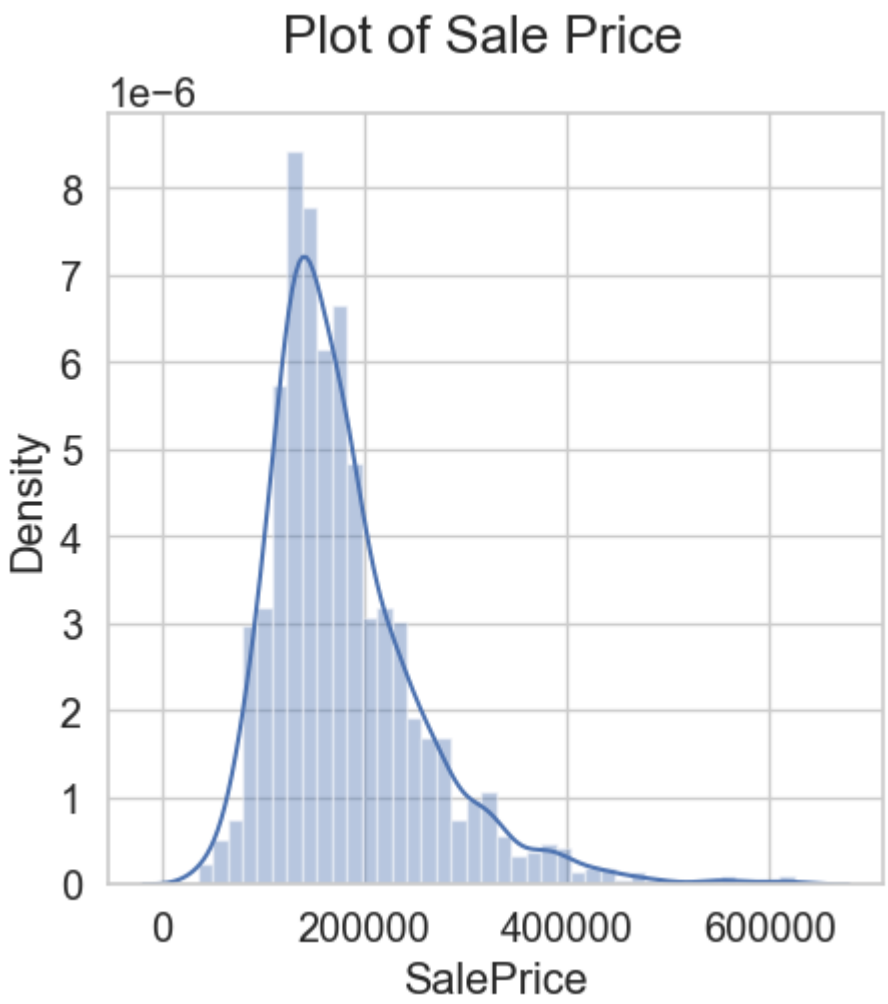
```
Text(51, 0, '1950'),
Text(52, 0, '1951'),
Text(53, 0, '1952'),
Text(54, 0, '1953'),
Text(55, 0, '1954'),
Text(56, 0, '1955'),
Text(57, 0, '1956'),
Text(58, 0, '1957'),
Text(59, 0, '1958'),
Text(60, 0, '1959'),
Text(61, 0, '1960'),
Text(62, 0, '1961'),
Text(63, 0, '1962'),
Text(64, 0, '1963'),
Text(65, 0, '1964'),
Text(66, 0, '1965'),
Text(67, 0, '1966'),
Text(68, 0, '1967'),
Text(69, 0, '1968'),
Text(70, 0, '1969'),
Text(71, 0, '1970'),
Text(72, 0, '1971'),
Text(73, 0, '1972'),
Text(74, 0, '1973'),
Text(75, 0, '1974'),
Text(76, 0, '1975'),
Text(77, 0, '1976'),
Text(78, 0, '1977'),
Text(79, 0, '1978'),
Text(80, 0, '1979'),
Text(81, 0, '1980'),
Text(82, 0, '1981'),
Text(83, 0, '1982'),
Text(84, 0, '1983'),
Text(85, 0, '1984'),
Text(86, 0, '1985'),
Text(87, 0, '1986'),
Text(88, 0, '1987'),
Text(89, 0, '1988'),
Text(90, 0, '1989'),
Text(91, 0, '1990'),
Text(92, 0, '1991'),
Text(93, 0, '1992'),
Text(94, 0, '1993'),
Text(95, 0, '1994'),
Text(96, 0, '1995'),
Text(97, 0, '1996'),
Text(98, 0, '1997'),
Text(99, 0, '1998'),
Text(100, 0, '1999'),
Text(101, 0, '2000'),
Text(102, 0, '2001'),
Text(103, 0, '2002'),
Text(104, 0, '2003'),
Text(105, 0, '2004'),
Text(106, 0, '2005'),
Text(107, 0, '2006'),
Text(108, 0, '2007'),
Text(109, 0, '2008'),
```

```
Text(110, 0, '2009'),
Text(111, 0, '2010')])
```



In [ ]:
```
sns.distplot(df_train['SalePrice'])

plt.suptitle( "Plot of Sale Price")

print("Skewness: %f" % df_train['SalePrice'].skew())
print("Kurtosis: %f" % df_train['SalePrice'].kurt())
```

```
C:\Users\singh\AppData\Local\Temp\ipykernel_39868\497540977.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_train['SalePrice'])
Skewness: 1.567473
Kurtosis: 3.888317
```



In [ ]:
```
# applying log transformation to correct the positive skewness in the data
# taking logs means that errors in predicting expensive and cheap houses will af

df_train['SalePrice'] = np.log(df_train['SalePrice'])
plt.suptitle("Plot of Sale Price after log transformation")
sns.distplot(df_train['SalePrice'])
plt.show()
```

```
C:\Users\singh\AppData\Local\Temp\ipykernel_39868\2944919078.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_train['SalePrice'])
```

In [ ]:
```
df_train['SalePrice'].describe()
```

Out[ ]:
```
count    1455.000000
mean       12.021706
std         0.396112
min        10.460242
25%        11.774520
50%        12.001505
75%        12.272562
max        13.345507
Name: SalePrice, dtype: float64
```

In [ ]:
```
df_train['SalePrice']
```

Out[ ]:
```
0       12.247694
1       12.109011
2       12.317167
3       11.849398
4       12.429216
          ...
1455    12.072541
1456    12.254863
1457    12.493130
1458    11.864462
1459    11.901583
Name: SalePrice, Length: 1455, dtype: float64
```

In [ ]:
```
df_train.shape
```

Out[ ]: (1455, 80)

Handling missing data

In [ ]:
```
#function to see the missing data in a dataframe
def missing_data(df,n):
    total = df.isnull().sum().sort_values(ascending=False)          # Total No o
    percentage = (df.isnull().sum() / df.isnull().count()).sort_values(ascending
    No_unique_val = df.nunique()                                    # No of uniqu
    missing_data = pd.concat([total, percentage, No_unique_val], axis=1,
                             keys=['Total No of missing val', '% of Missing val'

    print(missing_data.head(n))
```

In [ ]:
```
#training data
missing_data(df_train,20)
```

	Total No of missing val	% of Missing val	No of unique val
PoolQC	1451	99.725086	2
MiscFeature	1401	96.288660	4
Alley	1364	93.745704	2
Fence	1176	80.824742	4
FireplaceQu	690	47.422680	5
LotFrontage	259	17.800687	109
GarageYrBlt	81	5.567010	97
GarageCond	81	5.567010	5
GarageType	81	5.567010	6
GarageFinish	81	5.567010	3
GarageQual	81	5.567010	5
BsmtExposure	38	2.611684	4
BsmtFinType2	38	2.611684	6
BsmtCond	37	2.542955	4
BsmtQual	37	2.542955	4
BsmtFinType1	37	2.542955	6
MasVnrArea	8	0.549828	324
MasVnrType	8	0.549828	4
Electrical	1	0.068729	5
MSSubClass	0	0.000000	15

In [ ]:
```
df_train['PoolQC'].unique()
```

Out[ ]: array([nan, 'Fa', 'Gd'], dtype=object)

- PoolQC,Alley have only two unique values
- PoolQC has 99.7% of missing data, which means most of the values are NA: No Pool ie most of the houses do not have a pool
- PoolQC,Alley,MiscFeature will be dropped due to large number of missing values

In [ ]:
```
#test data
missing_data(df_test,34)
```

	Total No of missing val	% of Missing val	No of unique val
PoolQC	1456	99.794380	2
MiscFeature	1408	96.504455	3
Alley	1352	92.666210	2
Fence	1169	80.123372	4
FireplaceQu	730	50.034270	5
LotFrontage	227	15.558602	115
GarageYrBlt	78	5.346127	97
GarageFinish	78	5.346127	3
GarageQual	78	5.346127	4
GarageCond	78	5.346127	5
GarageType	76	5.209047	6
BsmtCond	45	3.084304	4
BsmtExposure	44	3.015764	4
BsmtQual	44	3.015764	4
BsmtFinType2	42	2.878684	6
BsmtFinType1	42	2.878684	6
MasVnrType	16	1.096642	4
MasVnrArea	15	1.028101	303
MSZoning	4	0.274160	5
Functional	2	0.137080	7
BsmtHalfBath	2	0.137080	3
BsmtFullBath	2	0.137080	4
Utilities	2	0.137080	1
KitchenQual	1	0.068540	4
SaleType	1	0.068540	9
BsmtFinSF1	1	0.068540	669
GarageCars	1	0.068540	6
BsmtUnfSF	1	0.068540	793
TotalBsmtSF	1	0.068540	736
Exterior2nd	1	0.068540	15
Exterior1st	1	0.068540	13
GarageArea	1	0.068540	459
BsmtFinSF2	1	0.068540	161
TotRmsAbvGrd	0	0.000000	12

In [ ]:
```
df_test['Utilities'].unique()
```

Out[ ]:
```
array(['AllPub', nan], dtype=object)
```

- all records mostly "AllPub" for Utilities

- PoolQC,Alley,MiscFeature will be dropped due to large number of missing values
- Utilities has only 1 unique value
- Utility will also be dropped

In [ ]:
```
# calculate total number of null values in training data
null_train = df_train.isnull().sum().sum()
print(null_train)

# calculate total number of null values in test data
null_test = df_test.isnull().sum().sum()
print(null_test)
```

```
6950
7000
```

In [ ]:
```
# save the 'SalePrice'column as train_label
train_label = df_train['SalePrice'].reset_index(drop=True)

# # drop 'SalePrice' column from df_train
df_train = df_train.drop(['SalePrice'], axis=1)
# # now df_train contains all training features
```

In [ ]:
```
# function to HANDLE the missing data in a dataframe
def missing (df):

    # drop theses columns due to large null values or many same values
    df = df.drop(['Utilities','PoolQC','MiscFeature','Alley'], axis=1)

    # Null value likely means No Fence so fill as "None"
    df["Fence"] = df["Fence"].fillna("None")

    # Null value likely means No Fireplace so fill as "None"
    df["FireplaceQu"] = df["FireplaceQu"].fillna("None")

    # Lot frontage is the feet of street connected to property, which is likely
    df["LotFrontage"] = df["LotFrontage"].fillna(df["LotFrontage"].median())

    # Null value likely means  typical(Typ)
    df["Functional"] = df["Functional"].fillna("Typ")

    # Only one null value so fill as the most frequent value(mode)
    df['KitchenQual'] = df['KitchenQual'].fillna(df['KitchenQual'].mode()[0])

    # Only one null value so fill as the most frequent value(mode)
    df['Electrical'] = df['Electrical'].fillna(df['Electrical'].mode()[0])

    # Very few null value so fill with the most frequent value(mode)
    df['SaleType'] = df['SaleType'].fillna(df['SaleType'].mode()[0])

    # Null value likely means no masonry veneer
    df["MasVnrType"] = df["MasVnrType"].fillna("None") #so fill as "None" (since
    df["MasVnrArea"] = df["MasVnrArea"].fillna(0)      #so fill as o

    # Only one null value so fill as the most frequent value(mode)
    df['Exterior1st'] = df['Exterior1st'].fillna(df['Exterior1st'].mode()[0])
    df['Exterior2nd'] = df['Exterior2nd'].fillna(df['Exterior2nd'].mode()[0])

    #MSZoning is general zoning classification,Very few null value so fill with
    df['MSZoning'] = df['MSZoning'].fillna(df['MSZoning'].mode()[0])

    #Null value likely means no Identified type of dwelling so fill as "None"
    df['MSSubClass'] = df['MSSubClass'].fillna("None")

    # Null value likely means No Garage, so fill as "None" (since these are cate
    for col in ('GarageType', 'GarageFinish', 'GarageQual', 'GarageCond'):
        df[col] = df[col].fillna('None')

    # Null value likely means No Garage and no cars in garage, so fill as 0
    for col in ('GarageYrBlt', 'GarageArea', 'GarageCars'):
        df[col] = df[col].fillna(0)

    # Null value likely means No Basement, so fill as 0
    for col in ('BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF','TotalBsmtSF', 'BsmtFull
        df[col] = df[col].fillna(0)
```

```
    # Null value likely means No Basement, so fill as "None" (since these are ca
    for col in ('BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFin
        df[col] = df[col].fillna('None')

    return df
```

In [ ]:
```
df_train = missing(df_train)
df_test = missing(df_test)
```

In [ ]:
```
# calculate total number of null values in training data
null_train = df_train.isnull().sum().sum()
print(null_train)

# calculate total number of null values in test data
null_test = df_test.isnull().sum().sum()
print(null_test)
```

```
0
0
```

In [ ]:
```
df_train.shape,df_test.shape
```

Out[ ]:
```
((1455, 75), (1459, 75))
```

In [ ]:
```
def add_new_cols(df):

    df['Total_SF'] = df['TotalBsmtSF'] + df['1stFlrSF'] + df['2ndFlrSF']

    df['Total_Bathrooms'] = (df['FullBath'] + (0.5 * df['HalfBath']) + df['BsmtF
                             + (0.5 * df['BsmtHalfBath']))

    df['Total_Porch_SF'] = (df['OpenPorchSF'] + df['3SsnPorch'] + df['EnclosedPo
                            df['ScreenPorch'] + df['WoodDeckSF'])

    df['Total_Square_Feet'] = (df['BsmtFinSF1'] + df['BsmtFinSF2'] + df['1stFlrS

    df['Total_Quality'] = df['OverallQual'] + df['OverallCond']

    return df
```

In [ ]:
```
# add the new columns
df_train = add_new_cols(df_train)
df_test = add_new_cols(df_test)
```

In [ ]:
```
df_train.shape,df_test.shape
```

Out[ ]:
```
((1455, 80), (1459, 80))
```

Check data types

In [ ]:
```
#training data
g1 = df_train.columns.to_series().groupby(df_train.dtypes).groups
```

In [ ]:
```
{k.name: v for k, v in g1.items()}
```

Out[ ]:
```
{'int64': Index(['MSSubClass', 'LotArea', 'OverallQual', 'OverallCond', 'YearBu
ilt',
        'YearRemodAdd', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF',
        '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath',
        'BsmtHalfBath', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr',
        'TotRmsAbvGrd', 'Fireplaces', 'GarageCars', 'GarageArea', 'WoodDeckSF',
        'OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea',
        'MiscVal', 'MoSold', 'YrSold', 'Total_SF', 'Total_Porch_SF',
        'Total_Square_Feet', 'Total_Quality'],
       dtype='object'),
 'float64': Index(['LotFrontage', 'MasVnrArea', 'GarageYrBlt', 'Total_Bathroom
s'], dtype='object'),
 'object': Index(['MSZoning', 'Street', 'LotShape', 'LandContour', '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', 'GarageCon
d',
       'PavedDrive', 'Fence', 'SaleType', 'SaleCondition'],
      dtype='object')}
```

In [ ]:
```
#testing data
g2 = df_test.columns.to_series().groupby(df_test.dtypes).groups
```

In [ ]:
```
{k.name: v for k, v in g2.items()}
```

Out[ ]:
```
{'int64': Index(['MSSubClass', 'LotArea', 'OverallQual', 'OverallCond', 'YearBu
ilt',
        'YearRemodAdd', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea',
        'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd',
        'Fireplaces', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorc
h',
        'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'YrSold',
        'Total_Porch_SF', 'Total_Quality'],
       dtype='object'),
 'float64': Index(['LotFrontage', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'Bs
mtUnfSF',
        'TotalBsmtSF', 'BsmtFullBath', 'BsmtHalfBath', 'GarageYrBlt',
        'GarageCars', 'GarageArea', 'Total_SF', 'Total_Bathrooms',
        'Total_Square_Feet'],
       dtype='object'),
 'object': Index(['MSZoning', 'Street', 'LotShape', 'LandContour', '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', 'GarageCon
d',
       'PavedDrive', 'Fence', 'SaleType', 'SaleCondition'],
      dtype='object')}
```

In [ ]:
```
#get dummy values for categorical data
df_train = pd.get_dummies(df_train)
df_test = pd.get_dummies(df_test)
```

```
print(df_train.shape)
print(df_test.shape)
```

```
(1455, 292)
(1459, 278)
```

In [ ]:
```
#align the training and testing data
df_train, df_test = df_train.align(df_test, join = 'inner', axis=1)
```

In [ ]:
```
print(df_train.shape)
print(df_test.shape)
```

```
(1455, 278)
(1459, 278)
```

In [ ]:
```
# calculate total number of null values in training data
null_train = df_train.isnull().sum().sum()
print(null_train)

# calculate total number of null values in test data
null_test = df_test.isnull().sum().sum()
print(null_test)
```

```
0
0
```

In [ ]:
```
df_train.head(5)
```

Out[ ]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodA
0	60	65.0	8450	7	5	2003	2
1	20	80.0	9600	6	8	1976	1
2	60	68.0	11250	7	5	2001	2
3	70	60.0	9550	7	5	1915	1
4	60	84.0	14260	8	5	2000	2

5 rows × 278 columns

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

Out[ ]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodA
0	20	80.0	11622	5	6	1961	1
1	20	81.0	14267	6	6	1958	1
2	60	74.0	13830	5	5	1997	1
3	60	78.0	9978	6	6	1998	1
4	120	43.0	5005	8	5	1992	1

5 rows × 278 columns

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1455 entries, 0 to 1459
Columns: 278 entries, MSSubClass to SaleCondition_Partial
dtypes: float64(4), int64(37), uint8(237)
memory usage: 814.2 KB
```

In [ ]:
```
X_test = df_test           # testing features
```

In [ ]:
```
df_train["SalePrice"] = train_label
```

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

Out[ ]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodA
0	60	65.0	8450	7	5	2003	2
1	20	80.0	9600	6	8	1976	1
2	60	68.0	11250	7	5	2001	2
3	70	60.0	9550	7	5	1915	1
4	60	84.0	14260	8	5	2000	2

5 rows × 279 columns

In [ ]:
```
train_set, valid_set = train_test_split(df_train,train_size= 0.7, shuffle=False)

X_train = train_set.drop(["SalePrice"], axis=1)  # training features
y_train = train_set["SalePrice"].copy()             # training label

X_valid = valid_set.drop(["SalePrice"], axis=1)  # testing features
y_valid = valid_set["SalePrice"].copy()               # testing label
```

In [ ]:
```
print("X_train shape: {}".format(X_train.shape))
print("y_train shape: {}".format(y_train.shape))
print()
print("X_valid shape: {}".format(X_valid.shape))
print("y_valid shape: {}".format(y_valid.shape))
print()
print("X_test shape: {}".format(X_test.shape))
```

```
X_train shape: (1018, 278)
y_train shape: (1018,)

X_valid shape: (437, 278)
y_valid shape: (437,)

X_test shape: (1459, 278)
```

Check data type and null values

In [ ]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1018 entries, 0 to 1020
Columns: 278 entries, MSSubClass to SaleCondition_Partial
dtypes: float64(4), int64(37), uint8(237)
memory usage: 569.6 KB
```

In [ ]:
```
X_valid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 437 entries, 1021 to 1459
Columns: 278 entries, MSSubClass to SaleCondition_Partial
dtypes: float64(4), int64(37), uint8(237)
memory usage: 244.5 KB
```

In [ ]:
```
y_train
```

Out[ ]:
```
0       12.247694
1       12.109011
2       12.317167
3       11.849398
4       12.429216
          ...
1016    12.271345
1017    12.078239
1018    12.175613
1019    11.373663
1020    12.160029
Name: SalePrice, Length: 1018, dtype: float64
```

In [ ]:
```
y_valid
```

Out[ ]:
```
1021    12.567237
1022    11.630709
1023    12.028739
1024    12.588191
1025    11.561716
          ...
1455          NaN
1456          NaN
1457          NaN
1458          NaN
1459          NaN
Name: SalePrice, Length: 437, dtype: float64
```

In [ ]:
```
null_t_x = X_train.isnull().sum().sum()
print(null_t_x)

null_t_y = y_train.isnull().sum().sum()
print(null_t_y)
```

```
0
0
```

In [ ]:
```
null_v_x = X_valid.isnull().sum().sum()
print(null_v_x)

null_v_y = y_valid.isnull().sum().sum()
print(null_v_y)
```

```
0
5
```

- No null values in X_valid

- There are 5 null values in y_valid

In [ ]:
```
np.where(np.isnan(y_valid))
```

Out[ ]:
```
(array([432, 433, 434, 435, 436], dtype=int64),)
```

In [ ]:
```
# replace null values by mean value of y_valid column
mean = np.nanmean(y_valid)
y_valid = np.nan_to_num(y_valid,nan = mean)
```

In [ ]:
```
#check again
np.where(np.isnan(y_valid))
```

Out[ ]:
```
(array([], dtype=int64),)
```

In [ ]:
```
y_valid.dtype
```

Out[ ]:
```
dtype('float64')
```

In [ ]:
```
print("Valid data shape:")
print(X_valid.shape, y_valid.shape)
print()
```

```
Valid data shape:
(437, 278) (437,)
```

---------- 3. SET CROSS VALIDATION AND RMSE ----------

Cross Validation

- done to avoid underfitting/overfitting of data and to get a better understanging of how good our models are performing
- split data into k subsets, and train on k-1 of those subset,leaving one for testing
- performing 10-fold cross validation for each model#

In [ ]:
```
# calculating cross validation score with scoring set to negative mean absolute
def cross_validation(model):

    scores = np.sqrt(-cross_val_score(model, X_train, y_train, cv = 12, scoring
    mean = np.mean(scores)
    print("Mean CV score: ",mean)
```

RMSE

In [ ]:
```
# function to calculate Root mean square error (RMSE)
def rmse(y_pred, y_train):

    rmse_ = np.sqrt(metrics.mean_squared_error(y_pred,y_train))
    print("rmse: ", rmse_)
```

Plot Label

In [ ]:
```
# function to plot actual vs predicited label
def actual_vs_pred_plot(y_train,y_pred):

    fig = plt.figure(figsize=(12,12))
    fig, ax = plt.subplots()

    ax.scatter(y_train, y_pred,color = "teal",edgecolor = 'lightblue')
    ax.plot([y_train.min(),y_train.max()], [y_train.min(), y_train.max()], 'k--'
    ax.set_xlabel('Actual')
    ax.set_ylabel('Predicted')
    plt.suptitle("Actual vs Predicted Scatter Plot",size=14)
    plt.show()
```

---------- 4. DATA MODELLING ----------

MODELS

1. LINEAR REGRESSION MODEL

- Linear Regression is the first model used. In this model, the target value is expected to be a linear combination of the features. The coefficients are set to minimize the residual sum of squares between the target predicted and the observed features

In [ ]:
```
reg = linear_model.LinearRegression()
```

In [ ]:
```
cross_validation(reg)
```

```
Mean CV score:  0.4752199075412438
```

In [ ]:
```
#fit on training
model_reg = reg.fit(X_train, y_train)

#predict value of sale price on the training set
y1_pred = reg.predict(X_train)

#caculate root mean square error
rmse(y1_pred,y_train)
```

```
rmse:  0.3442710335666392
```

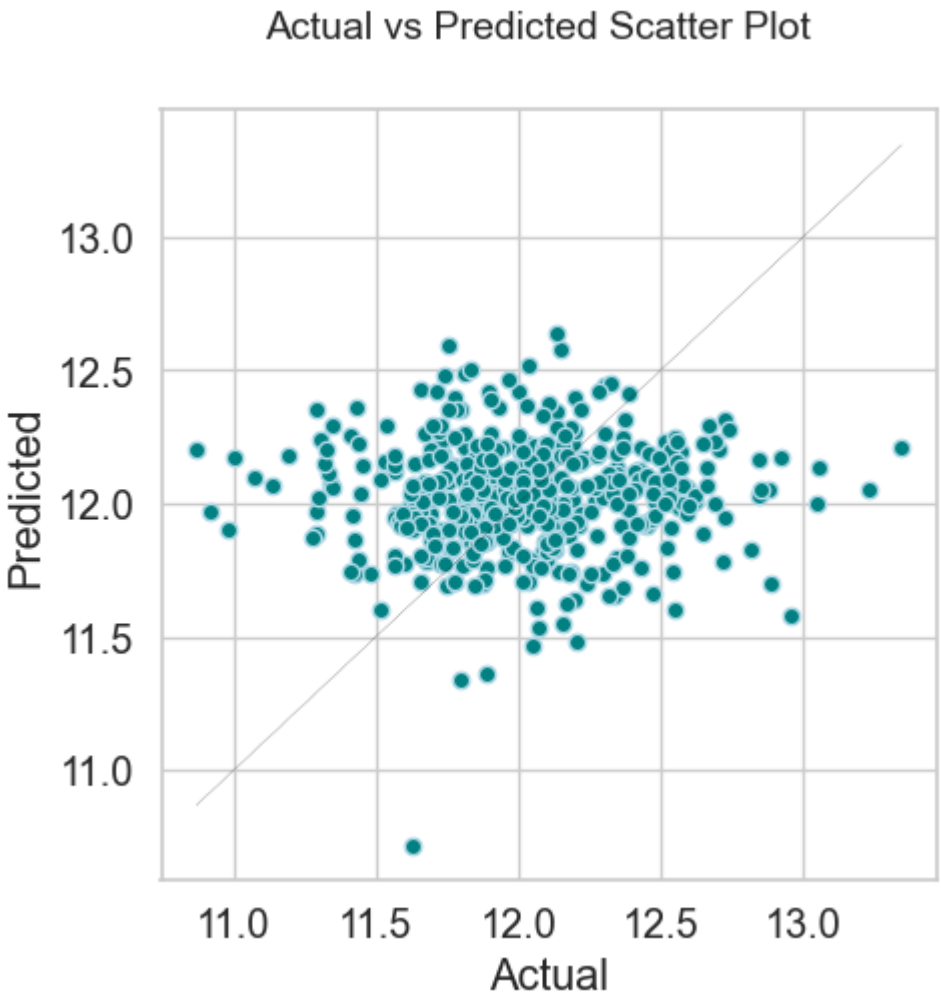
In [ ]:
```
#predict value of sale price on the validation set
y1_pred_v = reg.predict(X_valid)
```

```
#caculate root mean square error
rmse(y1_pred_v, y_valid)
```

```
rmse:  0.4279348039715779
 0.4279348039715779
```

In [ ]:
```
#plot
actual_vs_pred_plot(y_valid,y1_pred_v)
```

```
<Figure size 1200x1200 with 0 Axes>
```



2. RIDGE MODEL

- The second model used is Ridge Regression. Ridge Regression is a regularized version of linear regression. The parameter alpha is used to regularize the model. For alpha equal to zero, ridge regression is just a linear regression. RidgeCV model is used to implement ridge regression as it has a built-in cross validation of the alpha parameter. Sixteen different values of alpha between 7e-4 and 20 were used with a 10-fold cross validation. A pipeline using min-max scaler was built to apply to training, validation and testing data.

In [ ]:
```
# to find the best value of alphas from this list, i will use RidgeCV
alphas_ = [ 7e-4, 5e-4, 3e-4, 1e-4, 1e-3, 5e-2, 1e-2, 0.1, 0.3, 1, 3, 5, 10, 15,

# use robust scaler as unlike other scalers, the centering and scaling of ro bus
#is based on percentiles and are therefore is not influenced by a few number of
```

```
ridge = make_pipeline(MinMaxScaler(), linear_model.RidgeCV(alphas = alphas_, cv
```

In [ ]:
```
cross_validation(ridge)
```

```
Mean CV score:  0.41672707496259215
```

In [ ]:
```
#fit
model_ridge = ridge.fit(X_train, y_train)

#predict value of sale price on the training set
y2_pred = ridge.predict(X_train)

#caculate root mean square error
rmse(y2_pred,y_train)
```

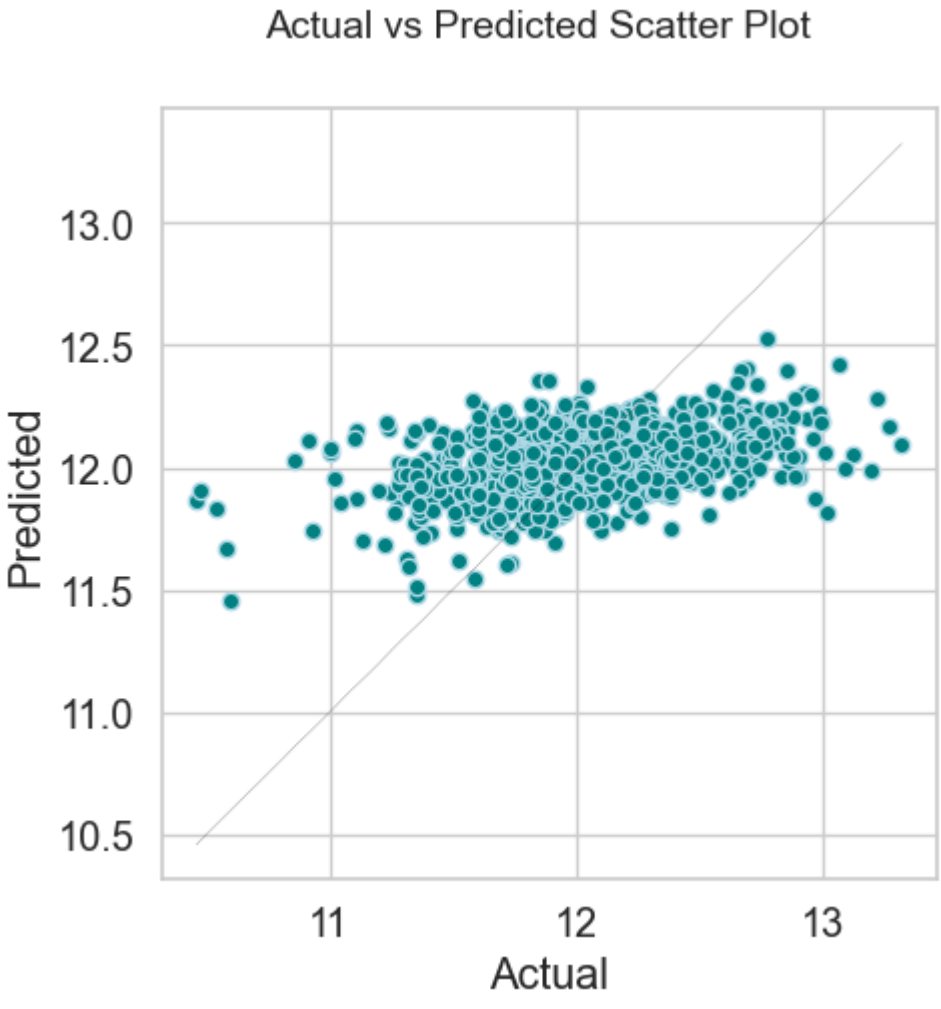
```
rmse:  0.36727237018186476
```

In [ ]:
```
#predict value of sale price on the valid set
y2_pred_v = ridge.predict(X_valid)

#caculate root mean square error
rmse(y2_pred_v, y_valid)
```

```
rmse:  0.39578861674332816
```

In [ ]:
```
#plot
actual_vs_pred_plot(y_train,y2_pred)
```

```
<Figure size 1200x1200 with 0 Axes>
```



3. LASSO MODEL

- Lasso regression is also a regularized version of linear regression. Lasso regression automatically performs feature selection and can estimates sparse coefficients. LassoCV model was used to implement lasso regression as it has a built-in cross validation of the alpha parameter. Different values of alpha were set with a 10-fold cross validation. Robust scaler was used in a pipeline to scale the training, validation and testing data.

In [ ]:
```
# to find the best value of alphas from this list, i will use LassoCV
alpha2 = [0.0001, 0.0002, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008]

#use robust scaler so that predictions are not influenced by a few number of ver

lasso = make_pipeline(RobustScaler(), linear_model.LassoCV(alphas = alpha2, rand
```

In [ ]:
```
cross_validation(lasso)
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 15.47362
575406759, tolerance: 0.014062005916392917
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.64140
9756856184, tolerance: 0.01388682924653013
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 36.42589
46802201, tolerance: 0.014074783611391048
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.11443
7265382335, tolerance: 0.013641632636133466
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.91644
307846436, tolerance: 0.013867601255958597
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.31494
4270845615, tolerance: 0.01398203379828378
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.38928
574541305, tolerance: 0.01343266438665632
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 42.47522
906055463, tolerance: 0.014123457216722002
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 45.94631
772296238, tolerance: 0.014245077315842255
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 45.02256
2313928404, tolerance: 0.013871240498036671
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 42.85347
1985128174, tolerance: 0.014003757971204503
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 31.40342
7843163186, tolerance: 0.014144090071961648
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 38.21087
8598659704, tolerance: 0.014083396081368514
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 10.47688
0688472889, tolerance: 0.01419767179067312
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 17.20120
7655884502, tolerance: 0.013648506564403386
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.898535
5308236365, tolerance: 0.014338722637059974
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 24.21897
1121193253, tolerance: 0.014460714071915126
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 39.01046
9615529786, tolerance: 0.01408716910659155
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 27.15044
2633575324, tolerance: 0.014219998851852694
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.18592
6057028034, tolerance: 0.013869513751085373
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 38.13408
8114225165, tolerance: 0.014271391363214017
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 35.18288
758001171, tolerance: 0.01399915959633298
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.01476
377996009, tolerance: 0.013707584223022018
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.20481
402162133, tolerance: 0.013932102904575652
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.015492
744569257866, tolerance: 0.013525788400155336
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 18.26308
5217201173, tolerance: 0.013525788400155336
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.23461
667897132, tolerance: 0.014045010447412219
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.60797
856625055, tolerance: 0.01349762647587234
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.25798
230614353, tolerance: 0.014182811081658054
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.26826
776894666, tolerance: 0.014308044538793639
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.86880
5525155224, tolerance: 0.013937045611147662
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.30260
921760263, tolerance: 0.014072599528942138
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.018270
676380211626, tolerance: 0.01384242420987892
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.800066
7819013074, tolerance: 0.01384242420987892
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 29.14628
9589708964, tolerance: 0.013876446474780958
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 11.40570
0121518507, tolerance: 0.013442425862961384
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 26.41108
288538284, tolerance: 0.01425087479092477
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 27.64308
1831713573, tolerance: 0.013882685255525742
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.018650
169176780196, tolerance: 0.014021245126162873
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.00985
842004008, tolerance: 0.013722652355119453
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 39.13251
4132391634, tolerance: 0.014119977466120207
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.72252
979088061, tolerance: 0.013768178404593573
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 35.79422
132451256, tolerance: 0.01396029363376195
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.063031
90897018851, tolerance: 0.013634669073809572
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.27019
021683749, tolerance: 0.013634669073809572
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 18.93205
5975948934, tolerance: 0.013378198504056454
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 42.50951
117646173, tolerance: 0.013900198260009773
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.037584
99760377276, tolerance: 0.013350060591992716
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.80272
694345861, tolerance: 0.013350060591992716
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 36.28535
50372172, tolerance: 0.014043023497178923
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.45810
333759769, tolerance: 0.014163246977531574
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.072635
44170528746, tolerance: 0.01378831205192525
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.27580
455085245, tolerance: 0.01378831205192525
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.24724
1637356396, tolerance: 0.013919654587606599
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 35.92449
767225314, tolerance: 0.013532427064856805
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 29.32982
981729166, tolerance: 0.013929599102321001
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 34.01224
1538785716, tolerance: 0.013578284929557842
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.017825
346616575644, tolerance: 0.013770288377981974
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 19.39096
835738046, tolerance: 0.013770288377981974
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 33.44510
084688431, tolerance: 0.013417906932012455
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 34.93877
447362543, tolerance: 0.013404245477335385
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 33.88350
523476731, tolerance: 0.01371004187244328
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 35.06806
415817156, tolerance: 0.013159811613991528
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 33.47016
243809196, tolerance: 0.01385303605712451
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.52882
729809801, tolerance: 0.013973091275160382
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.81413
265803653, tolerance: 0.013598024005352321
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 33.73029
6500735115, tolerance: 0.013729224937300737
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 16.03309
1963993314, tolerance: 0.0136777199070858
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 30.32030
1002624092, tolerance: 0.013717816460828832
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.03811
444737591, tolerance: 0.013703819487876268
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 33.11990
53148095, tolerance: 0.01330551574665131
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 32.47832
9539852865, tolerance: 0.0141171776496551
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 33.88481
5541922855, tolerance: 0.013744408273487455
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 13.66916
424196883, tolerance: 0.013878067872404868
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.78325
822108888, tolerance: 0.013646251128345386
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 37.67864
3734062256, tolerance: 0.01404492579209056
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.53611
3593968096, tolerance: 0.013688851772421717
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 39.39576
388795449, tolerance: 0.013881952501850588
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.22749
683902228, tolerance: 0.013529836086047357
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 4.289372
26357732, tolerance: 0.013670641764035776
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.25752
5246229484, tolerance: 0.013670641764035776
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 32.96130
6416094686, tolerance: 0.013274768016506627
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 39.18480
852595509, tolerance: 0.013554451504562406
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.45695
75708033, tolerance: 0.013964525295261461
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.13691
355059203, tolerance: 0.014086232782743643
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.58901
2605071815, tolerance: 0.013712464652421971
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.75440
511402076, tolerance: 0.013845055618829401
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 13.37020
1219869543, tolerance: 0.013685067387109945
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 42.38252
661707282, tolerance: 0.013685067387109945
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 12.49880
3466190402, tolerance: 0.014087626694964838
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 30.25129
6098451384, tolerance: 0.014087626694964838
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 6.818608
14838609, tolerance: 0.013719247345967347
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 37.00743
639077855, tolerance: 0.013719247345967347
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 11.82445
0977160893, tolerance: 0.013915184806156035
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 28.12342
858469788, tolerance: 0.013915184806156035
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 11.17550
8465758064, tolerance: 0.013563753555977423
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 35.44304
3327689274, tolerance: 0.013563753555977423
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 12.18760
6145476131, tolerance: 0.013715204815778544
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.14493
116384349, tolerance: 0.013715204815778544
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.220114
1636781756, tolerance: 0.013830893331015535
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 34.91191
223564776, tolerance: 0.013830893331015535
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.37005
6207550235, tolerance: 0.01372516724905493
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.94371
708411756, tolerance: 0.014123290092369482
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 11.38466
4879780388, tolerance: 0.013752880544695674
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.83728
966376388, tolerance: 0.013752880544695674
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 12.84726
782588907, tolerance: 0.013889065078739877
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 36.33302
2194141186, tolerance: 0.013889065078739877
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.99126
947598644, tolerance: 0.014158251392731108
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 34.02311
171490523, tolerance: 0.01455725816168879
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 42.53157
6295302635, tolerance: 0.014200132137281211
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 31.32371
4164605327, tolerance: 0.014393475383925358
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 38.04948
8042364445, tolerance: 0.014041417544163829
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.49650
352189429, tolerance: 0.01418313333209035
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 3.087513
5695320495, tolerance: 0.013786578962971075
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 35.33360
805204438, tolerance: 0.014306489435414193
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.20781
243925859, tolerance: 0.013904361486610521
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.42632
912084454, tolerance: 0.014418480448371478
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.85390
566107205, tolerance: 0.014224601355564914
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 41.37069
8800213106, tolerance: 0.014357499542533613
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.262290
4312899834, tolerance: 0.013859093480151831
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.190549
719415557, tolerance: 0.013859093480151831
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.952080
6581997618, tolerance: 0.013859093480151831
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 45.35370
6016371454, tolerance: 0.013859093480151831
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.135726
097670812, tolerance: 0.014258058927726155
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.074256
3003300347, tolerance: 0.014258058927726155
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.824710
0059115837, tolerance: 0.014258058927726155
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.62957
6004413465, tolerance: 0.014258058927726155
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.809260
6270377303, tolerance: 0.01390107252164853
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.85682
6773502334, tolerance: 0.01390107252164853
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.32584
970466459, tolerance: 0.014094381320551649
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.851909
4377221705, tolerance: 0.01374231536369735
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.851596
0495385428, tolerance: 0.01374231536369735
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.608547
5588693612, tolerance: 0.01374231536369735
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 0.024286
259354767026, tolerance: 0.01374231536369735
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.525992
2900396106, tolerance: 0.01374231536369735
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 38.33248
0094252126, tolerance: 0.01374231536369735
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.124925
508780592, tolerance: 0.013883915191350321
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.125440
259695452, tolerance: 0.013883915191350321
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.134324
6444952797, tolerance: 0.013883915191350321
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.109982
440133223, tolerance: 0.013883915191350321
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.062180
9044211545, tolerance: 0.013883915191350321
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.806689
1376019782, tolerance: 0.013883915191350321
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.71132
3694882104, tolerance: 0.013883915191350321
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.710486
4807187994, tolerance: 0.0134874457430947
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 28.57937
864449119, tolerance: 0.0134874457430947
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.53795
4392071, tolerance: 0.014007361027747377
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.147505
8965951916, tolerance: 0.013605211982889606
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.124917
242027209, tolerance: 0.013605211982889606
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.101883
0034741995, tolerance: 0.013605211982889606
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.780172
6421501627, tolerance: 0.013605211982889606
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.33096
236407451, tolerance: 0.013605211982889606
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.788762
5098813373, tolerance: 0.014046231127915153
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 43.45859
4347180515, tolerance: 0.014046231127915153
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.770658
2401631152, tolerance: 0.01429885076424951
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.59049
018782781, tolerance: 0.01429885076424951
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 1.940593
6152705294, tolerance: 0.014058286995831615
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.73611
6956500815, tolerance: 0.014058286995831615
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.48485
361753502, tolerance: 0.014024783417259264
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 35.55647
2880866785, tolerance: 0.014420250216884206
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 42.49566
7903360605, tolerance: 0.014074346425843166
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 29.22409
2748985964, tolerance: 0.014265100364641873
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.626254
222472184, tolerance: 0.01391241732338039
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 39.61533
780613203, tolerance: 0.01391241732338039
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.968885
838741656, tolerance: 0.014044429363342392
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.83286
4441200456, tolerance: 0.014044429363342392
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 14.11736
4185881002, tolerance: 0.013655130098633712
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 33.76796
834413761, tolerance: 0.014175426472818496
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 2.948632
8724427864, tolerance: 0.013771588629217556
  model = cd_fast.enet_coordinate_descent_gram(
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Duality gap: 40.426622671990216, tolerance: 0.013771588629217556
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Duality gap: 37.616350284330196, tolerance: 0.01421446978773399
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Duality gap: 43.13431739529945, tolerance: 0.014463830254971615
  model = cd_fast.enet_coordinate_descent_gram(
Mean CV score:  0.4297643272515321
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Duality gap: 44.28310696257647, tolerance: 0.014059197139630457
  model = cd_fast.enet_coordinate_descent_gram(
```

In [ ]:

```python
#fit
model_lasso = lasso.fit(X_train, y_train)

#predict value of quality on the training set
y3_pred = lasso.predict(X_train)

#caculate root mean square error
rmse(y3_pred,y_train)
```

```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 48.69179
9741508845, tolerance: 0.015151270252779393
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 26.92416
2875655867, tolerance: 0.015366965177081287
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 48.11917
5819414, tolerance: 0.015214986226185617
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 15.45774
7448094082, tolerance: 0.015069259741237932
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 47.04851
073276161, tolerance: 0.015069259741237932
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 40.17393
4335878045, tolerance: 0.01487908645188081
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 25.55853
678857588, tolerance: 0.015023604319544939
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 48.04128
456574552, tolerance: 0.014992439021043876
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 14.62400
2068564238, tolerance: 0.015030438481461334
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 44.24415
52789656, tolerance: 0.015030438481461334
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 46.05350
579278047, tolerance: 0.015504346785604396
  model = cd_fast.enet_coordinate_descent_gram(
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\l
inear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not con
verge. You might want to increase the number of iterations. Duality gap: 13.49344
8788226942, tolerance: 0.01520520059432298
  model = cd_fast.enet_coordinate_descent_gram(
```

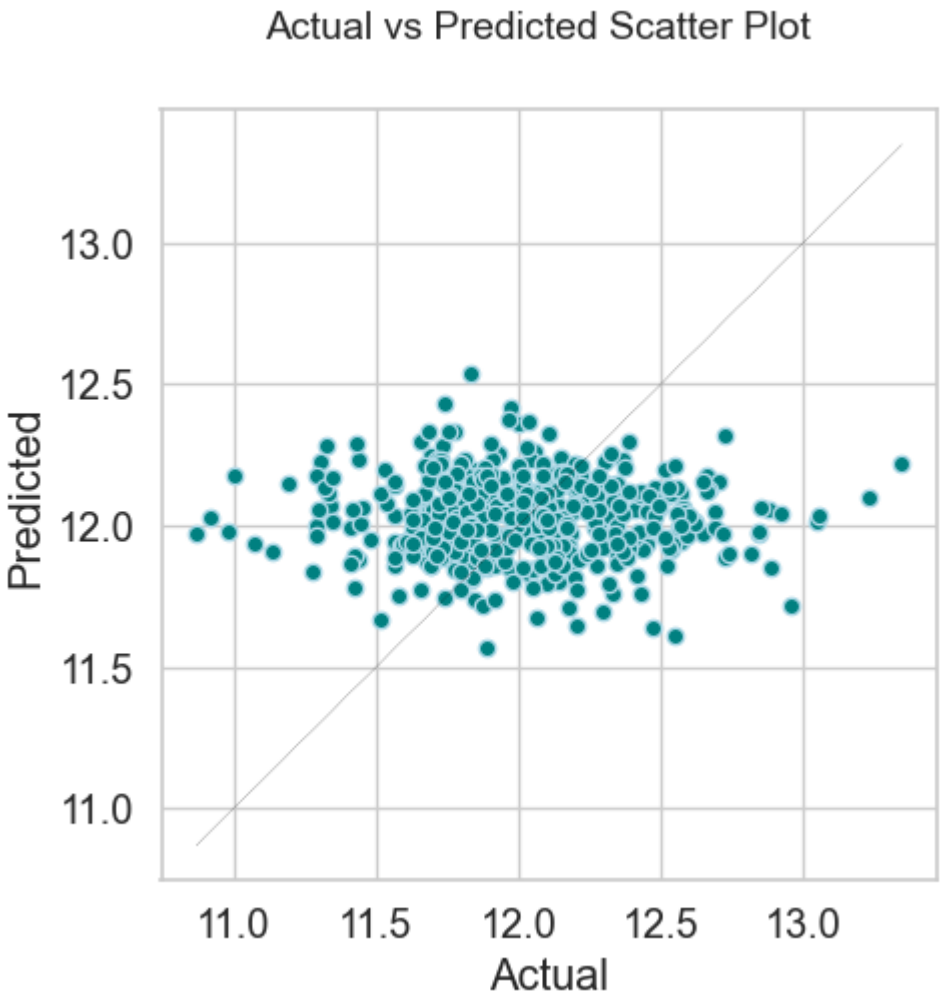
```
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Duality gap: 49.36538171974647, tolerance: 0.01520520059432298
  model = cd_fast.enet_coordinate_descent_gram(
rmse:  0.36267996691815335
c:\Users\singh\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:617: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Duality gap: 46.031063421148986, tolerance: 0.015372083714337646
  model = cd_fast.enet_coordinate_descent_gram(
```

In [ ]:
```
#predict value of sale price on the validation set
y3_pred_v = lasso.predict(X_valid)

#caculate root mean square error
rmse(y3_pred_v, y_valid)
```

```
rmse:  0.4059493256188701
```

In [ ]:
```
actual_vs_pred_plot(y_valid,y3_pred_v)
```

```
<Figure size 1200x1200 with 0 Axes>
```



4. K-NEAREST NEIGHBOUR REGRESSION MODEL

- K -nearest neighbour regressor is another popular model for regression tasks. It is a simple supervised machine learning model. The numbers of neighbours were set to

three different values and the performance of this model was noted. Weights were set to uniform to assign equal weights to all points in each neighbourhood. The algorithm used was set to auto so that the best performing algorithm on the values was used. The leaf size was set to 25.

In [ ]:
```
from sklearn.neighbors import KNeighborsRegressor

# N = 5 #
neigh = KNeighborsRegressor(n_neighbors = 5,
                            weights = 'uniform',
                            algorithm = 'auto',
                            leaf_size=25)
neigh.fit(X_train,y_train)

#predict value of sale price on the training set
y4_pred = neigh.predict(X_train)

#caculate root mean square error
rmse(y4_pred,y_train)
```

```
rmse:  0.34885424380933583
```

In [ ]:
```
# N = 7 #
neigh1 = KNeighborsRegressor(n_neighbors = 7,
                             weights = 'uniform',
                             leaf_size=25)
neigh1.fit(X_train,y_train)

#predict value of quality on the training set
y_pred = neigh1.predict(X_train)

#caculate root mean square error
rmse(y_pred,y_train)
```

```
rmse:  0.3665712393534244
```

In [ ]:
```
# N = 9 #
neigh2 = KNeighborsRegressor(n_neighbors = 9,
                             weights = 'uniform',
                             leaf_size=25)
neigh2.fit(X_train,y_train)

#predict value of quality on the training set
y_pred = neigh2.predict(X_train)

#caculate root mean square error
rmse(y_pred,y_train)
```

```
rmse:  0.37262338937265044
```

In [ ]:
```
# N=5 performs best
```

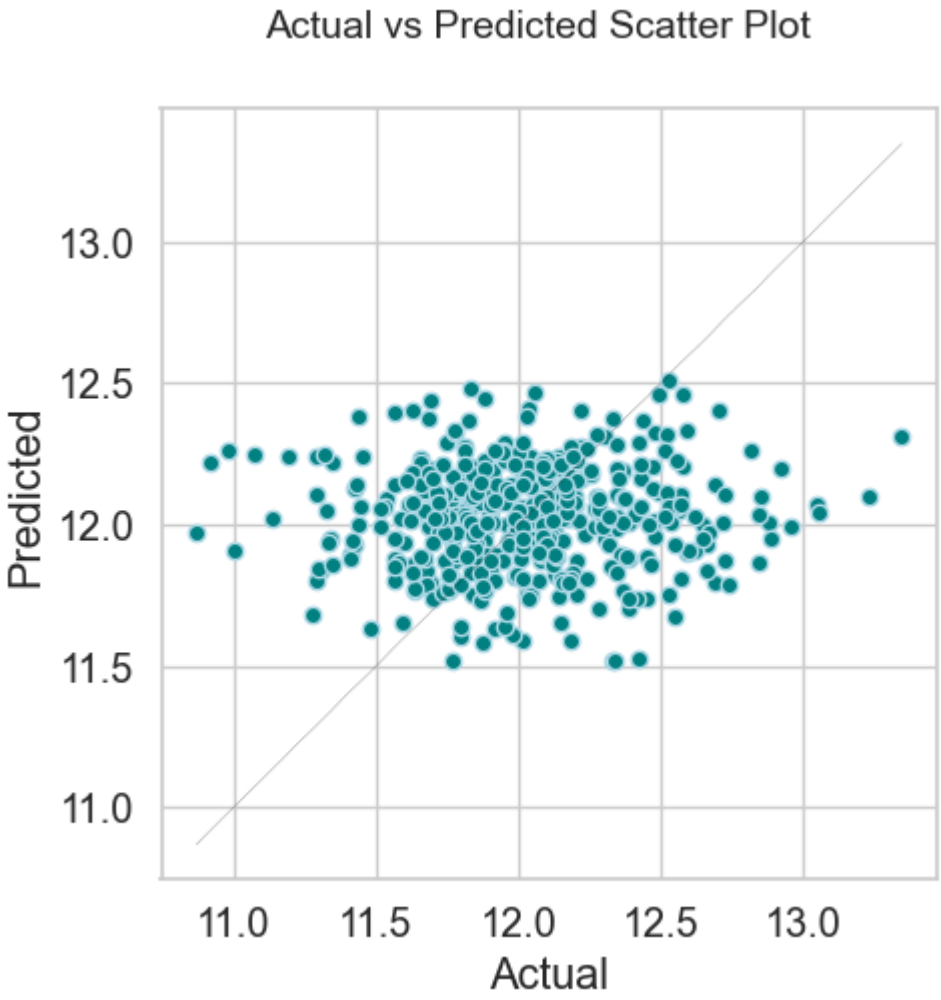
In [ ]:
```
#predict value of sale price on the validation set
y4_pred_v = neigh.predict(X_valid)

#caculate root mean square error
rmse(y4_pred_v, y_valid)
```

```
rmse:  0.41351487769327555
```

Note: rmse increases when values of k(no. of neighbours) increase

In [ ]:
```
actual_vs_pred_plot(y_valid,y4_pred_v)
```

```
<Figure size 1200x1200 with 0 Axes>
```



5. DECISION TREE MODEL

- Decision tree model is also used to fit this data as it does not require much data cleaning and is not influenced by outliers. Decision trees can, unlike linear models, fit linearly inseparable datasets. The values of minimum leaves were set between 1 to 9 because a very small number of minimum leaves can cause overfitting whereas a large number of minimum leaves will prevent the tree from learning. Maximum depth of 7 and 9 were used to fit the data for predictions.

In [ ]:
```
from sklearn import tree
```

In [ ]:
```
# set max depth to 5
tree_regr1 = tree.DecisionTreeRegressor(max_depth = 7, min_samples_leaf=5,random

# set max depth to 9
tree_regr2 = tree.DecisionTreeRegressor(max_depth = 9,min_samples_leaf=9,random_

#fit the traning data to a decision tree model
tree_regr11 = tree_regr1.fit(X_train,y_train)
```

```
tree_regr12 = tree_regr2.fit(X_train,y_train)

#predict value of sale price on the training set
y1 = tree_regr1.predict(X_train)
y2 = tree_regr2.predict(X_train)
```

In [ ]:
```
cross_validation(tree_regr1)
cross_validation(tree_regr2)
```

```
Mean CV score:  0.4440722344760503
Mean CV score:  0.45825272349446555
```

In [ ]:
```
#caculate root mean square error
rmse(y1,y_train)
```

```
rmse:  0.3238501847516405
```

In [ ]:
```
rmse(y2,y_train)
```

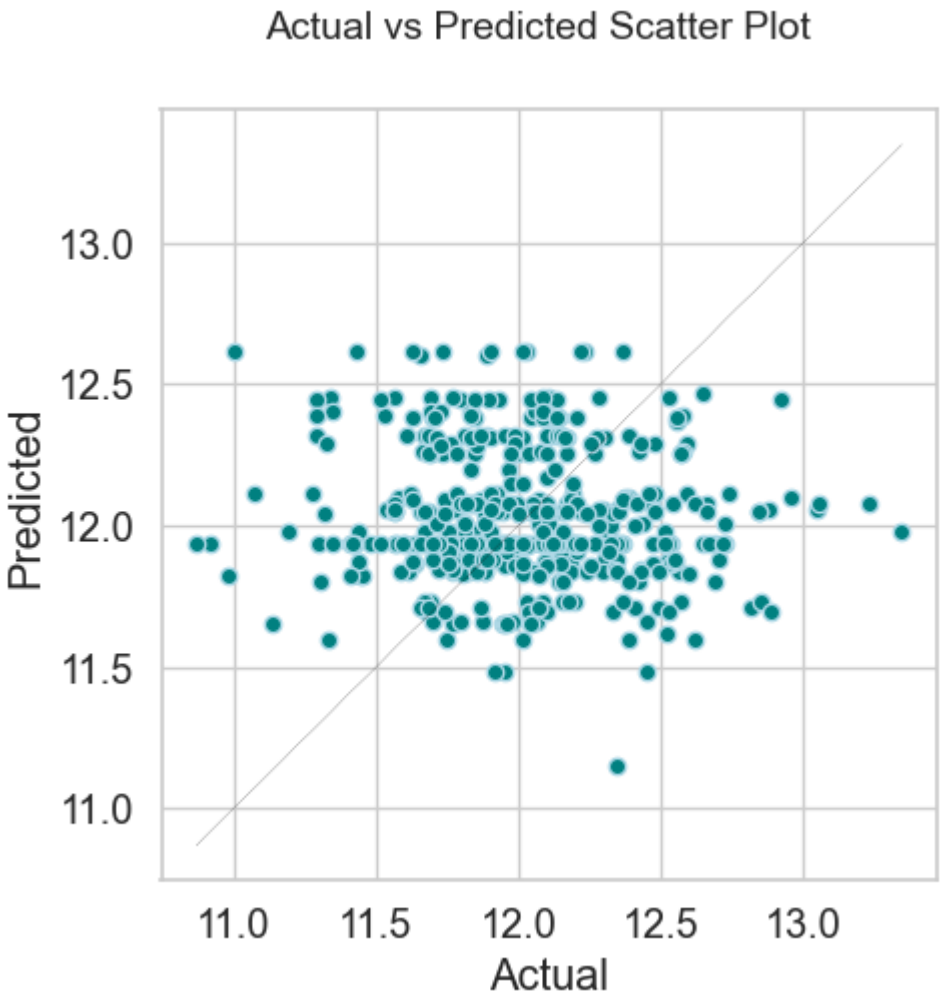
```
rmse:  0.319434991726199
```

In [ ]:
```
#predict value of sale price on the validation set
y5_pred_v = tree_regr2.predict(X_valid)

#caculate root mean square error
rmse(y5_pred_v, y_valid)
```

```
rmse:  0.4583579345988703
```

In [ ]:
```
#plot
actual_vs_pred_plot(y_valid,y5_pred_v)
```

```
<Figure size 1200x1200 with 0 Axes>
```



6. Random Forest MODEL

- Random forest model is an ensemble method based on randomized decision trees. Grid search was used to select the best parameters with a 5-fold cross validation. The number of trees in the forest was set to 200 with a maximum depth of 5 and 3 minimum leaves.

In [ ]:
```
rforest = RandomForestRegressor(n_estimators=200,max_depth=13,random_state=42)
```

In [ ]:
```
# grid search to find best value of C, gamma and epsilon
param_grid  = {'n_estimators': [100,150,200,250,300,350,400],
               'max_depth': [5,7,9,11,13,15,17],
               'min_samples_leaf': [3,5,7,9,11,13,15]}

# set cross validation to 5
clf = GridSearchCV(rforest, param_grid, cv = 5, n_jobs = -2)
clf.fit(X_train,y_train)
```

Out[ ]:
```
GridSearchCV
estimator: RandomForestRegressor
RandomForestRegressor
```

In [ ]:
```
clf.best_params_
```

Out[ ]:
```
{'max_depth': 5, 'min_samples_leaf': 3, 'n_estimators': 200}
```

In [ ]:
```
rforest = RandomForestRegressor(n_estimators=100, max_depth=5, min_samples_leaf=
```

In [ ]:
```
cross_validation(rforest)
```

```
Mean CV score:  0.4033235895902428
```

In [ ]:
```
#fit
model_rforest = rforest.fit(X_train, y_train)

#predict value of sale price on the training set
y6_pred = rforest.predict(X_train)

#caculate root mean square error
rmse(y6_pred,y_train)
```

```
rmse:  0.34988313302053653
```

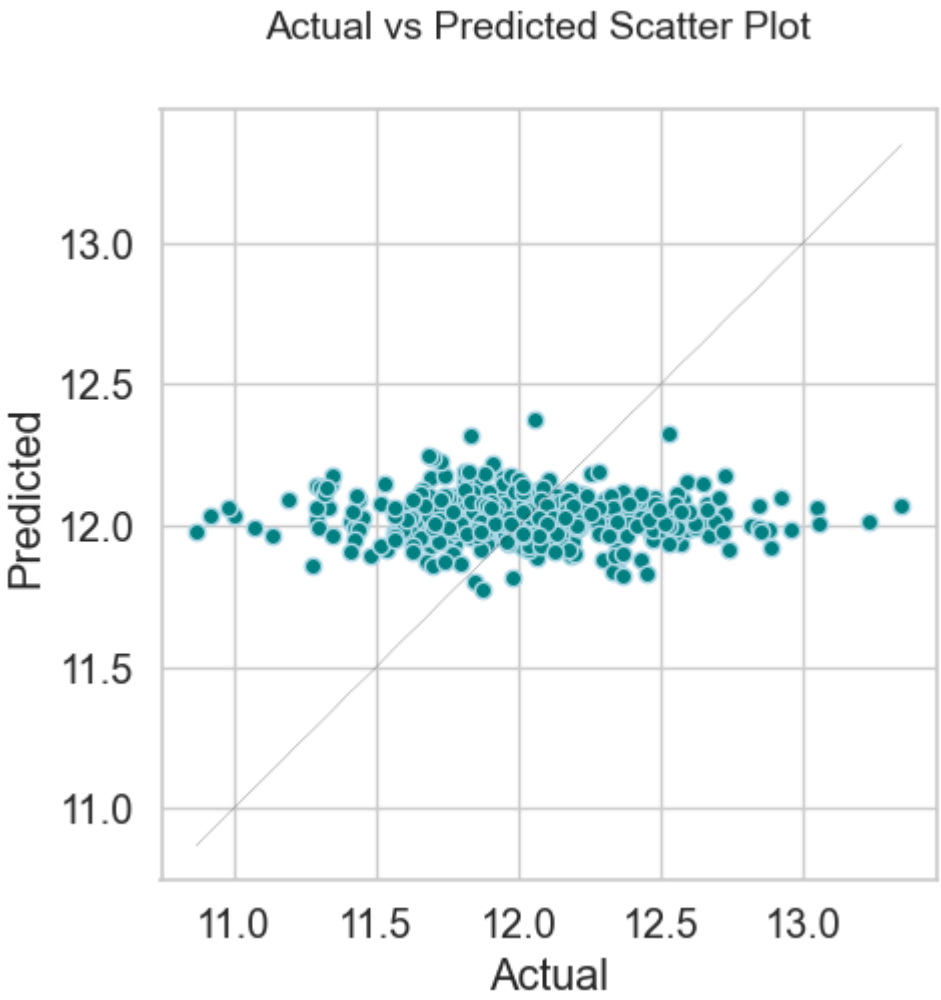
In [ ]:
```
#predict value of sale price on the validation set
y6_pred_v = rforest.predict(X_valid)

#caculate root mean square error
rmse(y6_pred_v, y_valid)
```

```
rmse:  0.386302468562908
```

In [ ]:
```
#0: 0.38852359192540425
#1: 0.38616747296757176
```

In [ ]:
```
#plot
actual_vs_pred_plot(y_valid, y6_pred_v)
```

```
<Figure size 1200x1200 with 0 Axes>
```



7. Support Vector Regressor MODEL

- Support vector regressor is another powerful model. It is memory efficient and offers different kernels to choose from. Grid search was used to find the best value of the hyperparameters C, gamma and epsilon. The sigmoid kernel was used along with the default value of epsilon.

In [ ]:
```
svr_basic = SVR(C = 10, gamma = 0.001)
```

In [ ]:
```
# grid search to find best value of C, gamma and epsilon and default kernel 'rbf
param_grid  = {'C': [5,7,10,15,20,30],'gamma': [0.001, 0.0001, 0.0011, 0.00011],

# set cross validation to 5
clf = GridSearchCV(svr_basic, param_grid, cv = 10, n_jobs = -2)
clf.fit(X_train,y_train)
```

Out[ ]:
```
▸ GridSearchCV
▸ estimator: SVR
    ▸ SVR
```

In [ ]:
```
clf.best_params_
```

Out[ ]:
```
{'C': 5, 'epsilon': 0.1, 'gamma': 0.0011}
```

In [ ]:
```
#make final SVR model with best parameters found from grid search
svr = make_pipeline(MinMaxScaler(), SVR(C= 5, epsilon= 0.1, gamma=0.0011, kernel
```

In [ ]:
```
cross_validation(svr)
```

```
Mean CV score:  0.40963206887105597
```

In [ ]:
```
#fit
model_svr = svr.fit(X_train, y_train)

#predict value of sale price on the training set
y7_pred = svr.predict(X_train)

#caculate root mean square error
rmse(y7_pred,y_train)
```

```
rmse:  0.3824587851531543
```

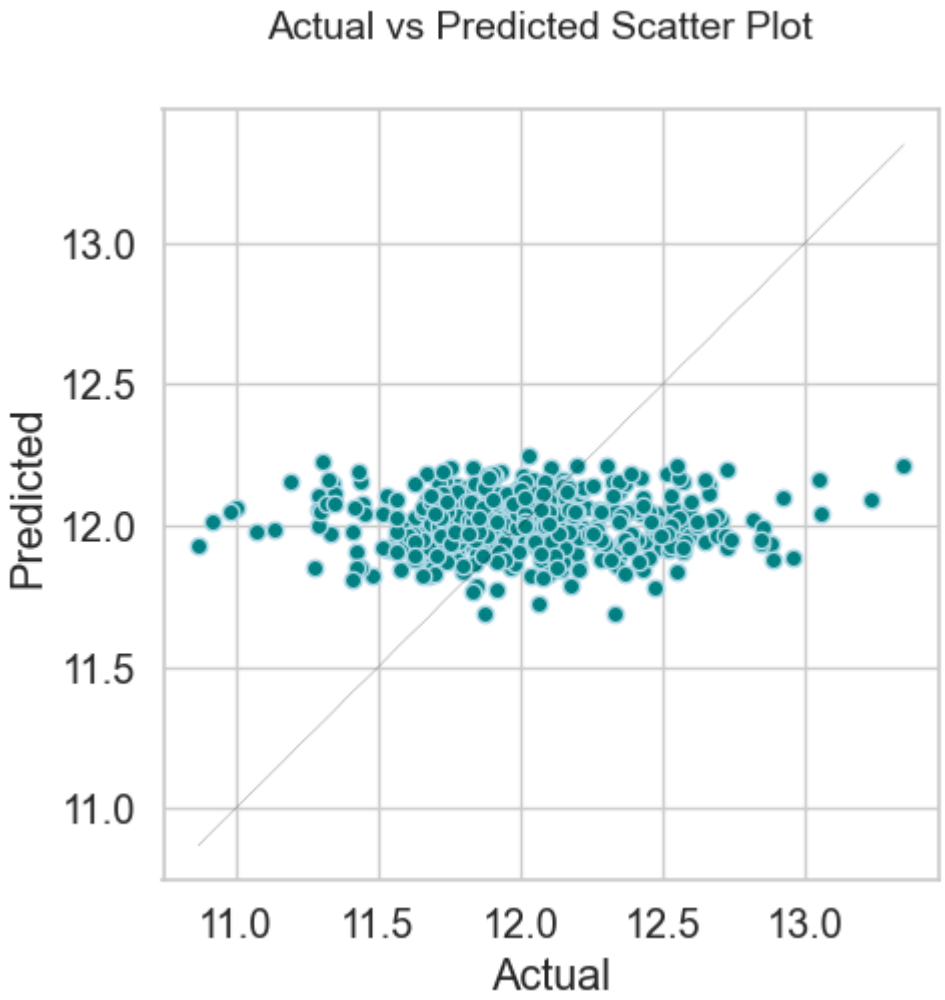
In [ ]:
```
#predict value of sale price on the validation set
y7_pred_v = svr.predict(X_valid)

#caculate root mean square error
rmse(y7_pred_v, y_valid)
```

```
rmse:  0.3900469727418301
```

In [ ]:
```
# Linear - 0.4338387095039476
# Sigmoid - 0.3900469727418305
# With sigmoid as default kernel - 0.39670545624904924
# rbf - 0.39420253052849114
```

In [ ]:
```
actual_vs_pred_plot(y_valid, y7_pred_v)
```

```
<Figure size 1200x1200 with 0 Axes>
```



8. Gradient Boosting Regressor MODEL

- Gradient boosting regression is an ensemble of weak prediction models. Two gradient boosting models with different depths were evaluated. The loss was set to 'huber' which is a combination of least square regression and a highly robust loss function.

In [ ]:
```
# set max depth to 4, min_samples_leaf to 15
gbr1 = GradientBoostingRegressor(n_estimators=200, learning_rate=0.05, max_depth
                                 min_samples_leaf=7, loss='huber', random_state =
```

In [ ]:
```
# set max depth to 7, min_samples_leaf to 10
gbr2 = GradientBoostingRegressor(n_estimators=200, learning_rate=0.05, max_depth
                                 min_samples_leaf=10, loss='huber', random_state
```

In [ ]:
```
cross_validation(gbr1)
cross_validation(gbr2)
```

```
Mean CV score:  0.4287974195276836
Mean CV score:  0.4292489907640939
```

In [ ]:
```
#fit
model_gbr1 = gbr1.fit(X_train, y_train)
model_gbr2 = gbr2.fit(X_train, y_train)

#predict value of sale price on the training set
```

```
y_g1_pred = gbr1.predict(X_train)
y_g2_pred = gbr2.predict(X_train)

#caculate root mean square error
rmse(y_g1_pred,y_train)
rmse(y_g2_pred,y_train)
```

```
rmse:  0.15045439854847656
rmse:  0.13917493901563793
```

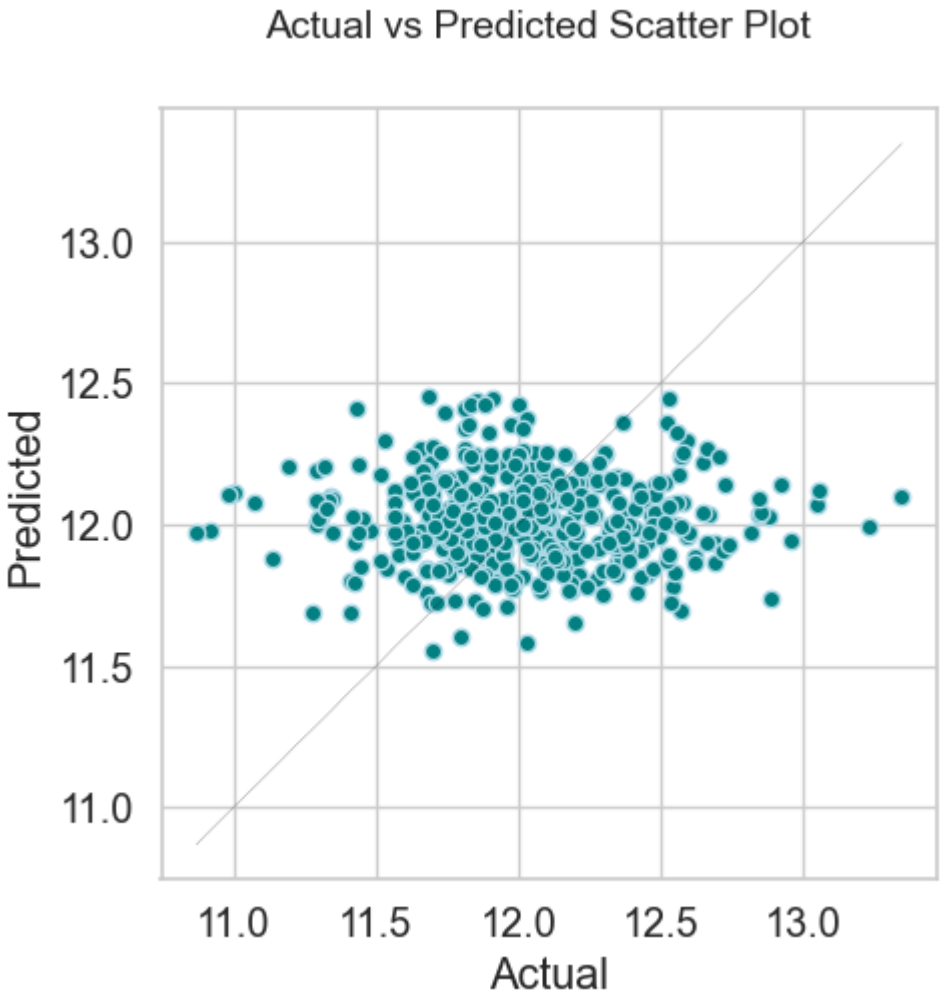
- model gbr2 performs best

In [ ]:
```
#predict value of sale price on the validation set
y8_pred_v = gbr2.predict(X_valid)

#caculate root mean square error
rmse(y8_pred_v, y_valid)
```

```
rmse:  0.4118219430457788
```

In [ ]:
```
# plot for gbr2
actual_vs_pred_plot(y_valid, y8_pred_v)
```

```
<Figure size 1200x1200 with 0 Axes>
```



9. STACKED REGRESSOR MODEL

- The final model used is the stacked regressor model. Stacking allows the power of each individual estimator to be used by using their output as a final estimator input. Random forest, Support vector regressor, K -nearest neighbour regressor and ridge regressor were stacked with random forest as the final estimator.

In [ ]:
```
# using Random Forest,Support Vector Regressor and Gradient Boosting to build a
estimators = [('Random Forest', rforest),
              ("Support Vector Regressor",svr),
              ("K",neigh),
              ("Ridge",ridge)
              ]
```

In [ ]:
```
stacked = StackingRegressor(estimators = estimators, final_estimator = rforest,
```

In [ ]:
```
cross_validation(stacked)
```

```
Mean CV score:  0.4096861892199762
```

In [ ]:
```
#fit
model_stack = stacked.fit(X_train, y_train)

#predict value of sale price on the training set
y9_pred = stacked.predict(X_train)

#caculate root mean square error
rmse(y9_pred,y_train)
```

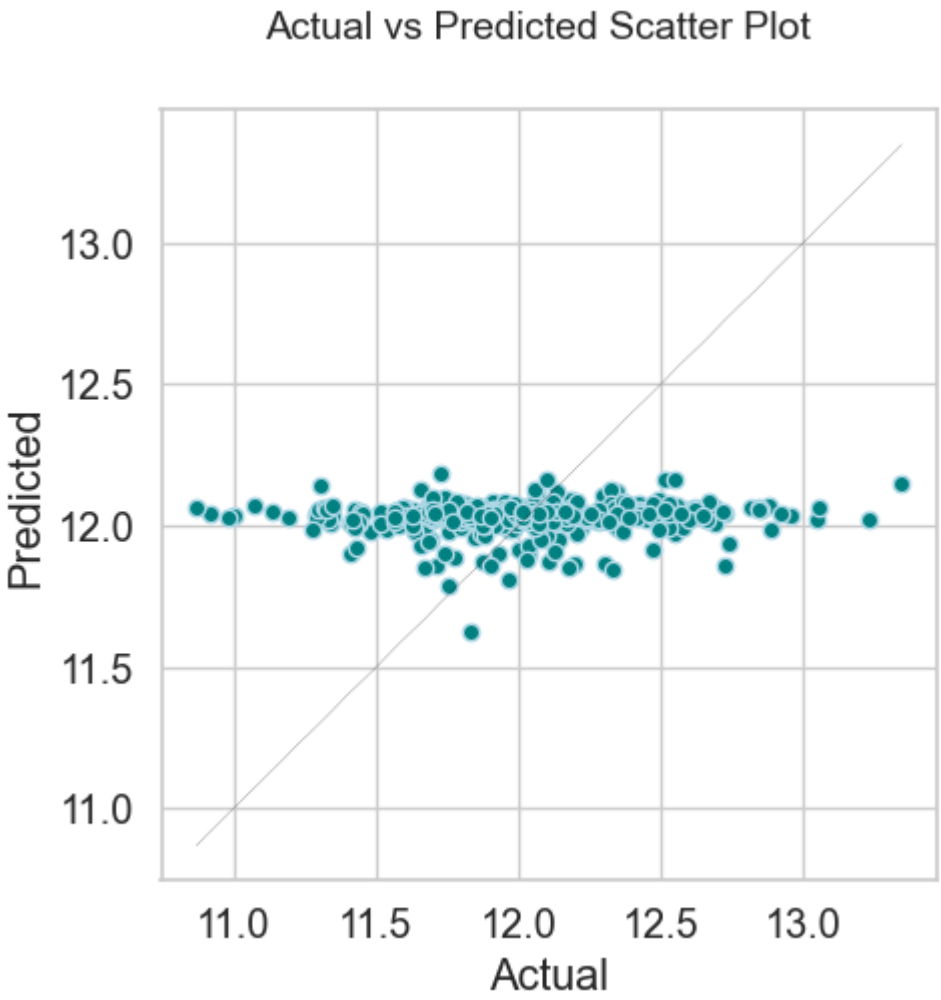
```
rmse:  0.40597813360670537
```

In [ ]:
```
#predict value of sale price on the validation set
y9_pred_v = stacked.predict(X_valid)

#caculate root mean square error
rmse(y9_pred_v, y_valid)
```

```
rmse:  0.37656322732768965
```

In [ ]:
```
# plot
actual_vs_pred_plot(y_valid,y9_pred_v)
```

```
<Figure size 1200x1200 with 0 Axes>
```



Observations

RMSE:

- linear reg : 0.42793480397157035
- ridge : 0.3957886167433282
- lasso : 0.4059493256188701
- k-nearest neighbour(k=5) : 0.41351487769327555
- decision tree(maxdepth=9) : 0.4583579345988703
- random forest : 0.38616747296757176
- Support Vector Regressor : 0.3900469727418305
- Gradient Boosting Regressor : 0.4118219430457788
- Stacked Regressor model : 0.3769718491202983

How errors compare:

- The lowest error is of : Stacked Regressor model
- The largest error is of : decision tree(maxdepth=9)
- Therefore Stacked Regressor model will be applied to the test data as it is the best performing model

---------- 5. TEST DATA PREDICTION ----------

In [ ]:
```
csv_path = "submission.csv"
df_sub = pd.read_csv(csv_path, sep = ',')
```

In [ ]:
```
df_sub.shape
```

Out[ ]:
```
(1459, 2)
```

In [ ]:
```
df_sub.head()
```

Out[ ]:

	Id	SalePrice
0	1461	129628.764474
1	1462	161001.627427
2	1463	174812.536981
3	1464	191413.661678
4	1465	182125.044985

In [ ]:
```
X_test.shape
```

Out[ ]:
```
(1459, 278)
```

In [ ]:
```
#predict value of sale price on the training set
y_final_pred = stacked.predict(X_test)

y_final_pred
```

Out[ ]:
```
array([11.99237181, 12.03894025, 12.04841174, ..., 12.06365339,
       11.88679711, 12.04042399])
```

In [ ]:
```
#undo the log tranformation to get predictions in terms of original label
predictions = np.expm1(y_final_pred)
print(predictions)
```

```
[161516.99092359 169216.5167971  170826.87379261 ... 173450.51504068
 145334.05952624 169467.77833468]
```

In [ ]:
```
submit = pd.DataFrame()
submit['Id'] = test_ID
submit['SalePrice'] = predictions
submit.to_csv('submission.csv',index=False)
```

In [ ]:
```
submit
```

Out[ ]:

	Id	SalePrice
0	1461	161516.990924
1	1462	169216.516797
2	1463	170826.873793
3	1464	162559.869981
4	1465	168201.618066
...	...	...
1454	2915	170147.397414
1455	2916	159996.655430
1456	2917	173450.515041
1457	2918	145334.059526
1458	2919	169467.778335

1459 rows × 2 columns