Housing_Sale_Regression_model

August 13, 2021

0.1 Regression Models for Housing Sales Dataset

1 Dataset Description

For this project I am using the Housing Price dataset that can be obtained from https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data. If we ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence. With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa this dataset allows the prediction of housing prices accurately.

[38]:
```
import pandas as pd # Import Pandas library for exploratory data analysis
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from pylab import rcParams

%matplotlib inline
sns.set(style='whitegrid', palette='muted', font_scale=1.5)

rcParams['figure.figsize'] = 14, 8
```

[39]:
```
data = pd.read_csv("train.csv")
data.head()
```

[39]:
```
   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
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

  LandContour Utilities  … PoolArea PoolQC Fence MiscFeature MiscVal MoSold  \
0         Lvl    AllPub  …        0    NaN   NaN         NaN       0      2
1         Lvl    AllPub  …        0    NaN   NaN         NaN       0      5
2         Lvl    AllPub  …        0    NaN   NaN         NaN       0      9
```

```
3          Lvl    AllPub  …        0    NaN   NaN          NaN       0      2
4          Lvl    AllPub  …        0    NaN   NaN          NaN       0     12

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500
3   2006        WD        Abnorml     140000
4   2008        WD         Normal     250000

[5 rows x 81 columns]
```

[40]:
```python
data.shape
```

[40]: (1460, 81)

There are 81 columns and 1460 rows in this dataset. The columns refer to the attributes such as LotArea, LotShape, GarageArea, GrLiveArea, etc. As this is a large dataset most of the columns are provided as abrevations the features and their subnotations are provided below for reference:

MSSubClass: Identifies the type of dwelling involved in the sale.

```
 20   1-STORY 1946 & NEWER ALL STYLES
 30   1-STORY 1945 & OLDER
 40   1-STORY W/FINISHED ATTIC ALL AGES
 45   1-1/2 STORY - UNFINISHED ALL AGES
 50   1-1/2 STORY FINISHED ALL AGES
 60   2-STORY 1946 & NEWER
 70   2-STORY 1945 & OLDER
 75   2-1/2 STORY ALL AGES
 80   SPLIT OR MULTI-LEVEL
 85   SPLIT FOYER
 90   DUPLEX - ALL STYLES AND AGES
120   1-STORY PUD (Planned Unit Development) - 1946 & NEWER
150   1-1/2 STORY PUD - ALL AGES
160   2-STORY PUD - 1946 & NEWER
180   PUD - MULTILEVEL - INCL SPLIT LEV/FOYER
190   2 FAMILY CONVERSION - ALL STYLES AND AGES
```

MSZoning: Identifies the general zoning classification of the sale.

```
A    Agriculture
C    Commercial
FV   Floating Village Residential
I    Industrial
RH   Residential High Density
RL   Residential Low Density
RP   Residential Low Density Park
RM   Residential Medium Density
```

LotFrontage: Linear feet of street connected to property

LotArea: Lot size in square feet

Street: Type of road access to property

```
Grvl Gravel
Pave Paved
```

Alley: Type of alley access to property

```
Grvl Gravel
Pave Paved
NA   No alley access
```

LotShape: General shape of property

```
Reg  Regular
IR1  Slightly irregular
IR2  Moderately Irregular
IR3  Irregular
```

LandContour: Flatness of the property

```
Lvl  Near Flat/Level
Bnk  Banked - Quick and significant rise from street grade to building
HLS  Hillside - Significant slope from side to side
Low  Depression
```

Utilities: Type of utilities available

```
AllPub   All public Utilities (E,G,W,& S)
NoSewr   Electricity, Gas, and Water (Septic Tank)
NoSeWa   Electricity and Gas Only
ELO  Electricity only
```

LotConfig: Lot configuration

```
Inside   Inside lot
Corner   Corner lot
CulDSac  Cul-de-sac
FR2  Frontage on 2 sides of property
FR3  Frontage on 3 sides of property
```

LandSlope: Slope of property

```
Gtl  Gentle slope
Mod  Moderate Slope
Sev  Severe Slope
```

Neighborhood: Physical locations within Ames city limits

```
Blmngtn  Bloomington Heights
Blueste  Bluestem
BrDale   Briardale
BrkSide  Brookside
ClearCr  Clear Creek
CollgCr  College Creek
Crawfor  Crawford
Edwards  Edwards
Gilbert  Gilbert
IDOTRR   Iowa DOT and Rail Road
MeadowV  Meadow Village
Mitchel  Mitchell
Names    North Ames
NoRidge  Northridge
NPkVill  Northpark Villa
NridgHt  Northridge Heights
NWAmes   Northwest Ames
OldTown  Old Town
SWISU    South & West of Iowa State University
Sawyer   Sawyer
SawyerW  Sawyer West
Somerst  Somerset
StoneBr  Stone Brook
Timber   Timberland
Veenker  Veenker
```

Condition1: Proximity to various conditions

```
Artery   Adjacent to arterial street
Feedr    Adjacent to feeder street
Norm Normal
RRNn Within 200' of North-South Railroad
RRAn Adjacent to North-South Railroad
PosN Near positive off-site feature--park, greenbelt, etc.
PosA Adjacent to postive off-site feature
RRNe Within 200' of East-West Railroad
RRAe Adjacent to East-West Railroad
```

Condition2: Proximity to various conditions (if more than one is present)

```
Artery   Adjacent to arterial street
Feedr    Adjacent to feeder street
Norm Normal
RRNn Within 200' of North-South Railroad
RRAn Adjacent to North-South Railroad
PosN Near positive off-site feature--park, greenbelt, etc.
PosA Adjacent to postive off-site feature
RRNe Within 200' of East-West Railroad
RRAe Adjacent to East-West Railroad
```

BldgType: Type of dwelling

```
1Fam Single-family Detached
2FmCon   Two-family Conversion; originally built as one-family dwelling
Duplx    Duplex
TwnhsE   Townhouse End Unit
TwnhsI   Townhouse Inside Unit
```

HouseStyle: Style of dwelling

```
1Story   One story
1.5Fin   One and one-half story: 2nd level finished
1.5Unf   One and one-half story: 2nd level unfinished
2Story   Two story
2.5Fin   Two and one-half story: 2nd level finished
2.5Unf   Two and one-half story: 2nd level unfinished
SFoyer   Split Foyer
SLvl Split Level
```

OverallQual: Rates the overall material and finish of the house

```
10   Very Excellent
9    Excellent
8    Very Good
7    Good
6    Above Average
5    Average
4    Below Average
3    Fair
2    Poor
1    Very Poor
```

OverallCond: Rates the overall condition of the house

```
10   Very Excellent
9    Excellent
8    Very Good
7    Good
6    Above Average
5    Average
4    Below Average
3    Fair
2    Poor
1    Very Poor
```

YearBuilt: Original construction date

YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)

RoofStyle: Type of roof

```
Flat Flat
```

```
Gable    Gable
Gambrel  Gabrel (Barn)
Hip  Hip
Mansard  Mansard
Shed Shed
```

RoofMatl: Roof material

```
ClyTile  Clay or Tile
CompShg  Standard (Composite) Shingle
Membran  Membrane
Metal    Metal
Roll Roll
Tar&Grv  Gravel & Tar
WdShake  Wood Shakes
WdShngl  Wood Shingles
```

Exterior1st: Exterior covering on house

```
AsbShng  Asbestos Shingles
AsphShn  Asphalt Shingles
BrkComm  Brick Common
BrkFace  Brick Face
CBlock   Cinder Block
CemntBd  Cement Board
HdBoard  Hard Board
ImStucc  Imitation Stucco
MetalSd  Metal Siding
Other    Other
Plywood  Plywood
PreCast  PreCast
Stone    Stone
Stucco   Stucco
VinylSd  Vinyl Siding
Wd Sdng  Wood Siding
WdShing  Wood Shingles
```

Exterior2nd: Exterior covering on house (if more than one material)

```
AsbShng  Asbestos Shingles
AsphShn  Asphalt Shingles
BrkComm  Brick Common
BrkFace  Brick Face
CBlock   Cinder Block
CemntBd  Cement Board
HdBoard  Hard Board
ImStucc  Imitation Stucco
MetalSd  Metal Siding
Other    Other
```

```
Plywood  Plywood
PreCast  PreCast
Stone    Stone
Stucco   Stucco
VinylSd  Vinyl Siding
Wd Sdng  Wood Siding
WdShing  Wood Shingles
```

MasVnrType: Masonry veneer type

```
BrkCmn   Brick Common
BrkFace  Brick Face
CBlock   Cinder Block
None None
Stone    Stone
```

MasVnrArea: Masonry veneer area in square feet

ExterQual: Evaluates the quality of the material on the exterior

```
Ex   Excellent
Gd   Good
TA   Average/Typical
Fa   Fair
Po   Poor
```

ExterCond: Evaluates the present condition of the material on the exterior

```
Ex   Excellent
Gd   Good
TA   Average/Typical
Fa   Fair
Po   Poor
```

Foundation: Type of foundation

```
BrkTil   Brick & Tile
CBlock   Cinder Block
PConc    Poured Contrete
Slab Slab
Stone    Stone
Wood Wood
```

BsmtQual: Evaluates the height of the basement

```
Ex   Excellent (100+ inches)
Gd   Good (90-99 inches)
TA   Typical (80-89 inches)
Fa   Fair (70-79 inches)
Po   Poor (<70 inches
```

```
NA   No Basement
```

BsmtCond: Evaluates the general condition of the basement

```
Ex   Excellent
Gd   Good
TA   Typical - slight dampness allowed
Fa   Fair - dampness or some cracking or settling
Po   Poor - Severe cracking, settling, or wetness
NA   No Basement
```

BsmtExposure: Refers to walkout or garden level walls

```
Gd   Good Exposure
Av   Average Exposure (split levels or foyers typically score average or above)
Mn   Mimimum Exposure
No   No Exposure
NA   No Basement
```

BsmtFinType1: Rating of basement finished area

```
GLQ  Good Living Quarters
ALQ  Average Living Quarters
BLQ  Below Average Living Quarters
Rec  Average Rec Room
LwQ  Low Quality
Unf  Unfinshed
NA   No Basement
```

BsmtFinSF1: Type 1 finished square feet

BsmtFinType2: Rating of basement finished area (if multiple types)

```
GLQ  Good Living Quarters
ALQ  Average Living Quarters
BLQ  Below Average Living Quarters
Rec  Average Rec Room
LwQ  Low Quality
Unf  Unfinshed
NA   No Basement
```

BsmtFinSF2: Type 2 finished square feet

BsmtUnfSF: Unfinished square feet of basement area

TotalBsmtSF: Total square feet of basement area

Heating: Type of heating

```
Floor    Floor Furnace
GasA Gas forced warm air furnace
GasW Gas hot water or steam heat
Grav Gravity furnace
```

```
OthW Hot water or steam heat other than gas
Wall Wall furnace
```

HeatingQC: Heating quality and condition

```
Ex   Excellent
Gd   Good
TA   Average/Typical
Fa   Fair
Po   Poor
```

CentralAir: Central air conditioning

```
N    No
Y    Yes
```

Electrical: Electrical system

```
SBrkr    Standard Circuit Breakers & Romex
FuseA    Fuse Box over 60 AMP and all Romex wiring (Average)
FuseF    60 AMP Fuse Box and mostly Romex wiring (Fair)
FuseP    60 AMP Fuse Box and mostly knob & tube wiring (poor)
Mix  Mixed
```

1stFlrSF: First Floor square feet

2ndFlrSF: Second floor square feet

LowQualFinSF: Low quality finished square feet (all floors)

GrLivArea: Above grade (ground) living area square feet

BsmtFullBath: Basement full bathrooms

BsmtHalfBath: Basement half bathrooms

FullBath: Full bathrooms above grade

HalfBath: Half baths above grade

Bedroom: Bedrooms above grade (does NOT include basement bedrooms)

Kitchen: Kitchens above grade

KitchenQual: Kitchen quality

```
Ex   Excellent
Gd   Good
TA   Typical/Average
Fa   Fair
Po   Poor
```

TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)

Functional: Home functionality (Assume typical unless deductions are warranted)

```
Typ  Typical Functionality
Min1 Minor Deductions 1
Min2 Minor Deductions 2
Mod  Moderate Deductions
Maj1 Major Deductions 1
Maj2 Major Deductions 2
Sev  Severely Damaged
Sal  Salvage only
```

Fireplaces: Number of fireplaces

FireplaceQu: Fireplace quality

```
Ex   Excellent - Exceptional Masonry Fireplace
Gd   Good - Masonry Fireplace in main level
TA   Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement
Fa   Fair - Prefabricated Fireplace in basement
Po   Poor - Ben Franklin Stove
NA   No Fireplace
```

GarageType: Garage location

```
2Types   More than one type of garage
Attchd   Attached to home
Basment  Basement Garage
BuiltIn  Built-In (Garage part of house - typically has room above garage)
CarPort  Car Port
Detchd   Detached from home
NA   No Garage
```

GarageYrBlt: Year garage was built

GarageFinish: Interior finish of the garage

```
Fin  Finished
RFn  Rough Finished
Unf  Unfinished
NA   No Garage
```

GarageCars: Size of garage in car capacity

GarageArea: Size of garage in square feet

GarageQual: Garage quality

```
Ex   Excellent
Gd   Good
```

```
TA   Typical/Average
Fa   Fair
Po   Poor
NA   No Garage
```

GarageCond: Garage condition

```
Ex   Excellent
Gd   Good
TA   Typical/Average
Fa   Fair
Po   Poor
NA   No Garage
```

PavedDrive: Paved driveway

```
Y    Paved
P    Partial Pavement
N    Dirt/Gravel
```

WoodDeckSF: Wood deck area in square feet

OpenPorchSF: Open porch area in square feet

EnclosedPorch: Enclosed porch area in square feet

3SsnPorch: Three season porch area in square feet

ScreenPorch: Screen porch area in square feet

PoolArea: Pool area in square feet

PoolQC: Pool quality

```
Ex   Excellent
Gd   Good
TA   Average/Typical
Fa   Fair
NA   No Pool
```

Fence: Fence quality

```
GdPrv    Good Privacy
MnPrv    Minimum Privacy
GdWo Good Wood
MnWw Minimum Wood/Wire
NA   No Fence
```

MiscFeature: Miscellaneous feature not covered in other categories

```
Elev Elevator
Gar2 2nd Garage (if not described in garage section)
```

```
Othr Other
Shed Shed (over 100 SF)
TenC Tennis Court
NA   None
```

MiscVal: $Value of miscellaneous feature

MoSold: Month Sold (MM)

YrSold: Year Sold (YYYY)

SaleType: Type of sale

```
WD   Warranty Deed - Conventional
CWD  Warranty Deed - Cash
VWD  Warranty Deed - VA Loan
New  Home just constructed and sold
COD  Court Officer Deed/Estate
Con  Contract 15% Down payment regular terms
ConLw    Contract Low Down payment and low interest
ConLI    Contract Low Interest
ConLD    Contract Low Down
Oth  Other
```

SaleCondition: Condition of sale

```
Normal   Normal Sale
Abnorml  Abnormal Sale -  trade, foreclosure, short sale
AdjLand  Adjoining Land Purchase
Alloca   Allocation - two linked properties with separate deeds, typically condo with a gara
Family   Sale between family members
Partial  Home was not completed when last assessed (associated with New Homes)
```

2 Exploring the data

An investigation of the dataset will determine the following: - total missing values - create a heatmap for correlation matrix of the data - density plot for SalePrice - scatterplot for SalePrice and GrLiveArea relation - scatterplot for SalePrice and TotalBsmtSF relation - boxplot for OverallQuality and SalePrice relation - pairplot for the above features

[41]:
```
data.describe()
```

[41]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	\
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	
std	421.610009	42.300571	24.284752	9981.264932	1.382997	
min	1.000000	20.000000	21.000000	1300.000000	1.000000	
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	

```
75%    1095.250000   70.000000   80.000000  11601.500000     7.000000
max    1460.000000  190.000000  313.000000 215245.000000    10.000000

       OverallCond    YearBuilt  YearRemodAdd   MasVnrArea   BsmtFinSF1  …  \
count  1460.000000  1460.000000   1460.000000  1452.000000  1460.000000  …
mean      5.575342  1971.267808   1984.865753   103.685262   443.639726  …
std       1.112799    30.202904     20.645407   181.066207   456.098091  …
min       1.000000  1872.000000   1950.000000     0.000000     0.000000  …
25%       5.000000  1954.000000   1967.000000     0.000000     0.000000  …
50%       5.000000  1973.000000   1994.000000     0.000000   383.500000  …
75%       6.000000  2000.000000   2004.000000   166.000000   712.250000  …
max       9.000000  2010.000000   2010.000000  1600.000000  5644.000000  …

        WoodDeckSF  OpenPorchSF  EnclosedPorch    3SsnPorch  ScreenPorch  \
count  1460.000000  1460.000000    1460.000000  1460.000000  1460.000000
mean     94.244521    46.660274      21.954110     3.409589    15.060959
std     125.338794    66.256028      61.119149    29.317331    55.757415
min       0.000000     0.000000       0.000000     0.000000     0.000000
25%       0.000000     0.000000       0.000000     0.000000     0.000000
50%       0.000000    25.000000       0.000000     0.000000     0.000000
75%     168.000000    68.000000       0.000000     0.000000     0.000000
max     857.000000   547.000000     552.000000   508.000000   480.000000

          PoolArea       MiscVal       MoSold       YrSold      SalePrice
count  1460.000000   1460.000000  1460.000000  1460.000000    1460.000000
mean      2.758904     43.489041     6.321918  2007.815753  180921.195890
std      40.177307    496.123024     2.703626     1.328095   79442.502883
min       0.000000      0.000000     1.000000  2006.000000   34900.000000
25%       0.000000      0.000000     5.000000  2007.000000  129975.000000
50%       0.000000      0.000000     6.000000  2008.000000  163000.000000
75%       0.000000      0.000000     8.000000  2009.000000  214000.000000
max     738.000000  15500.000000    12.000000  2010.000000  755000.000000

[8 rows x 38 columns]
```

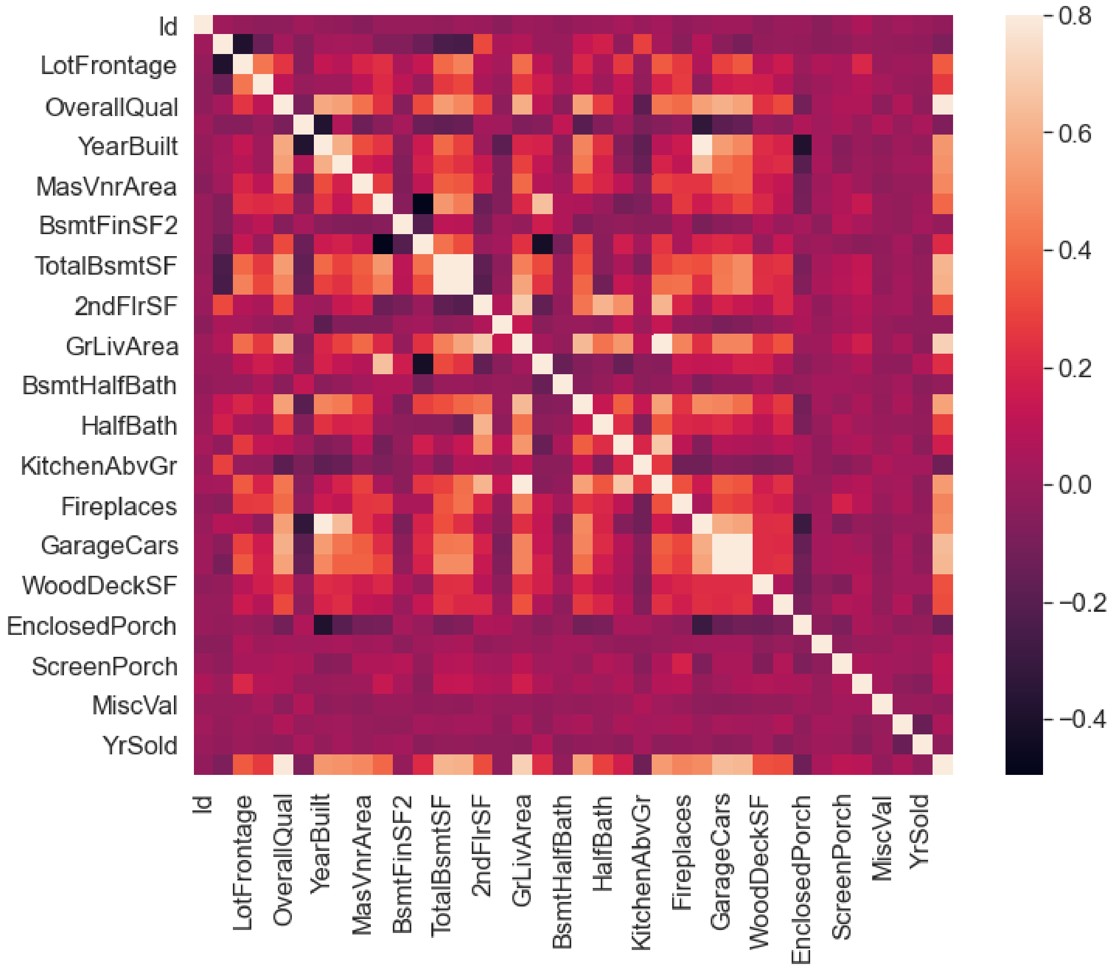
[42]:
```python
total = data.isnull().sum().sort_values(ascending=False)
percent = (data.isnull().sum()/data.isnull().count()).
 ↪sort_values(ascending=False)
missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
missing_data.head(20) #missing values
```

[42]:
```
              Total   Percent
PoolQC         1453  0.995205
MiscFeature    1406  0.963014
Alley          1369  0.937671
Fence          1179  0.807534
FireplaceQu     690  0.472603
```

```
LotFrontage     259  0.177397
GarageYrBlt      81  0.055479
GarageCond       81  0.055479
GarageType       81  0.055479
GarageFinish     81  0.055479
GarageQual       81  0.055479
BsmtFinType2     38  0.026027
BsmtExposure     38  0.026027
BsmtQual         37  0.025342
BsmtCond         37  0.025342
BsmtFinType1     37  0.025342
MasVnrArea        8  0.005479
MasVnrType        8  0.005479
Electrical        1  0.000685
Id                0  0.000000
```

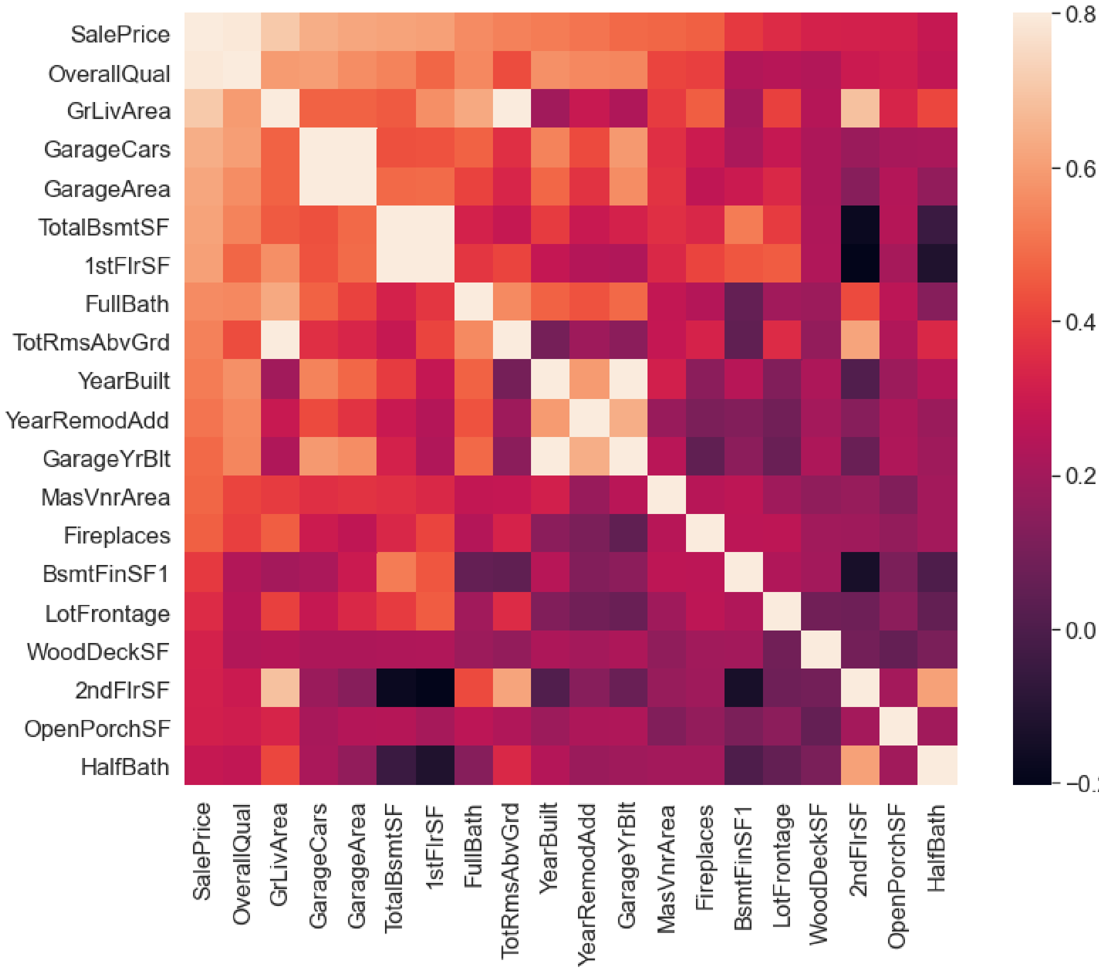
[43]:

```
corrmat = data.corr()
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True); #heatmap for correlation matrix of␣
 ↪the data
```



[44]:
```
k = 20 #number of variables for heatmap
cols = corrmat.nlargest(k, 'SalePrice')['SalePrice'].index #creates an array of␣
↪the top 20 correlation value columns
```

[45]:
```
f, ax = plt.subplots(figsize=(14, 10))
sns.heatmap(data[cols].corr(), vmax=.8, square=True);# heatmap of top 20
```

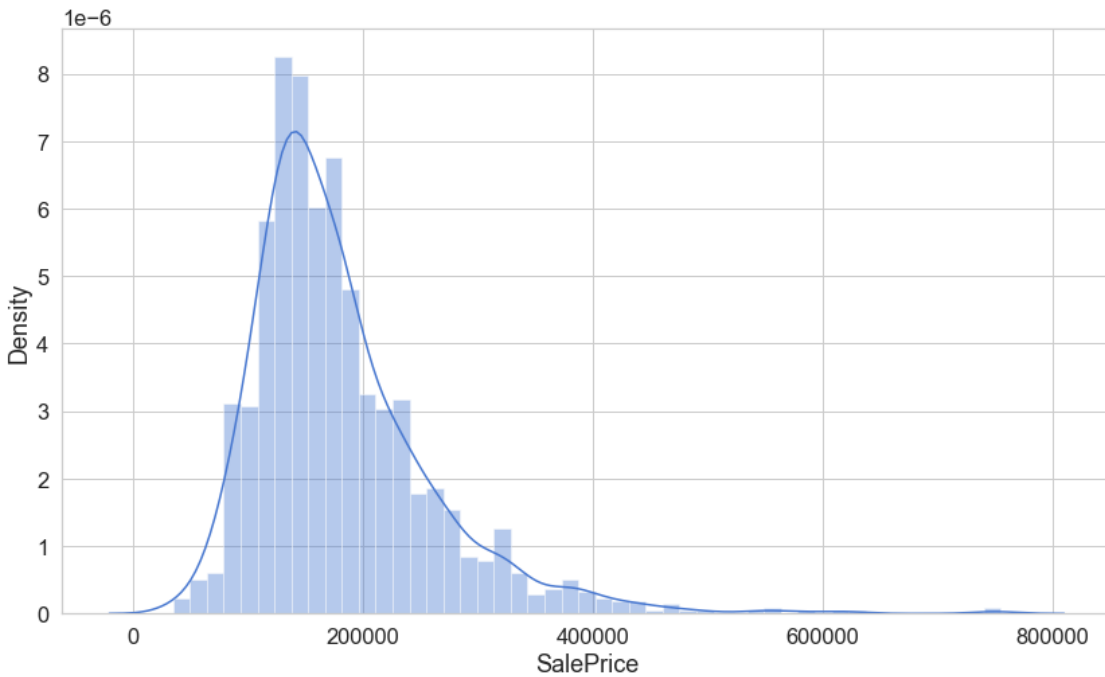


Density plot for SalePrice

[46]:
```
sns.distplot(data['SalePrice'])
```

```
C:\Users\insan\anaconda\lib\site-packages\seaborn\distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[46]:
```
<AxesSubplot:xlabel='SalePrice', ylabel='Density'>
```



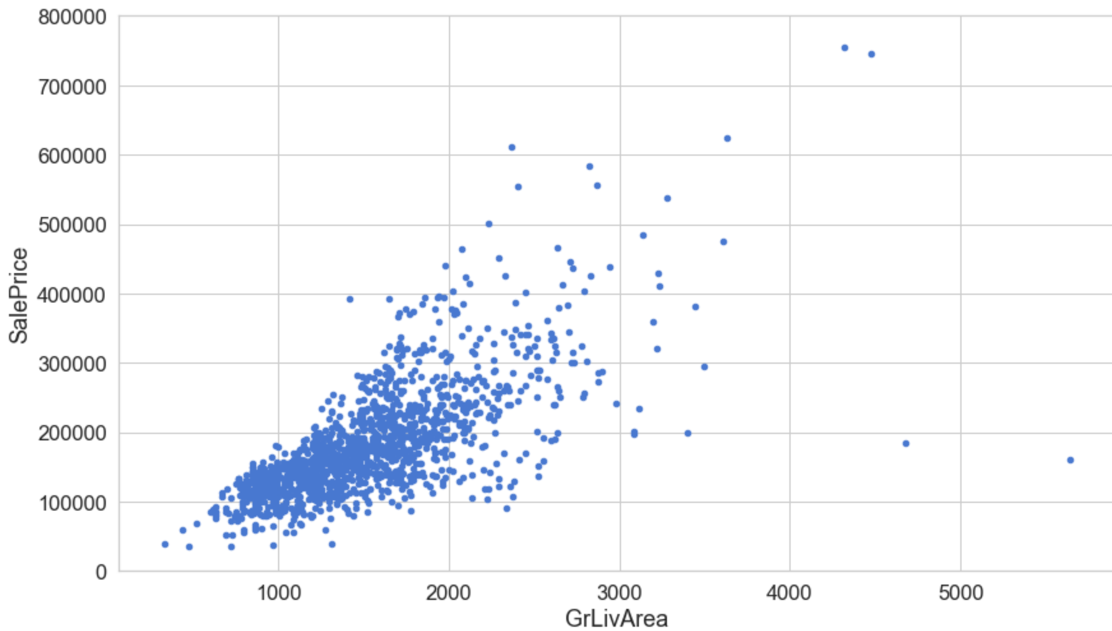
Here we can observe that the plot is skewed to one side. We shall correct this in Feature engineering section

Scatterplot for SalePrice and GrLiveArea relation

[47]:
```
df = pd.concat([data['SalePrice'], data["GrLivArea"]], axis=1)
df.plot.scatter(x="GrLivArea", y='SalePrice', ylim=(0,800000))
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
*x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with
a single row if you intend to specify the same RGB or RGBA value for all points.
```

[47]:
```
<AxesSubplot:xlabel='GrLivArea', ylabel='SalePrice'>
```

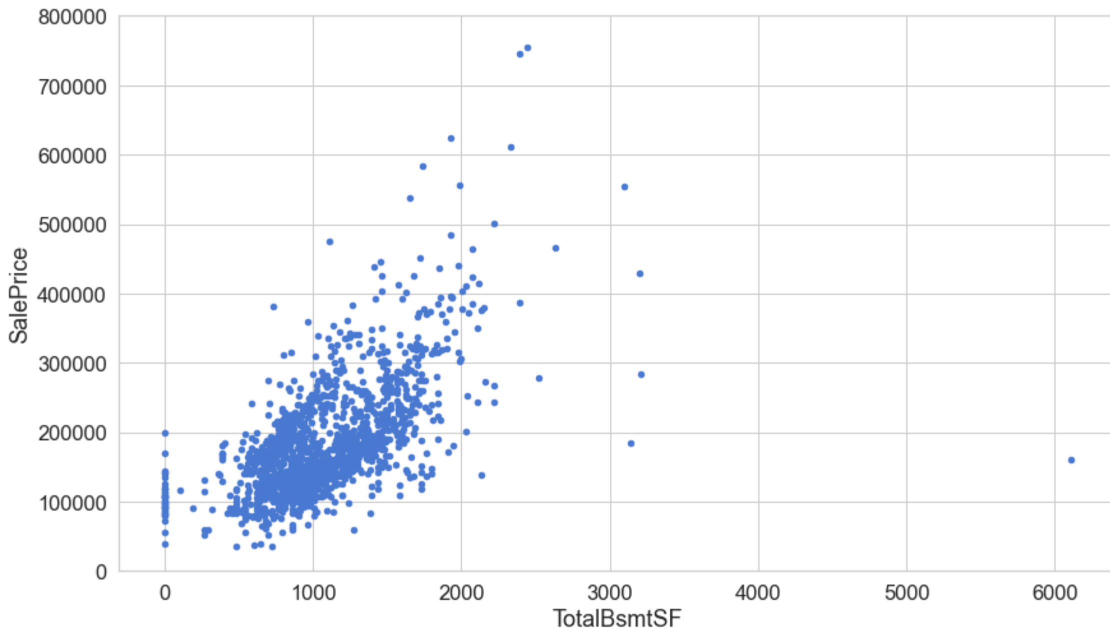


A linear relation can be observed. A few outliers are visible in the far right but it was observed for this data removing the outliers dosent help in the model.

Scatterplot for SalePrice and TotalBsmtSF relation

[48]:
```
var = 'TotalBsmtSF'
df = pd.concat([data['SalePrice'], data[var]], axis=1)
df.plot.scatter(x=var, y='SalePrice', ylim=(0,800000));
```

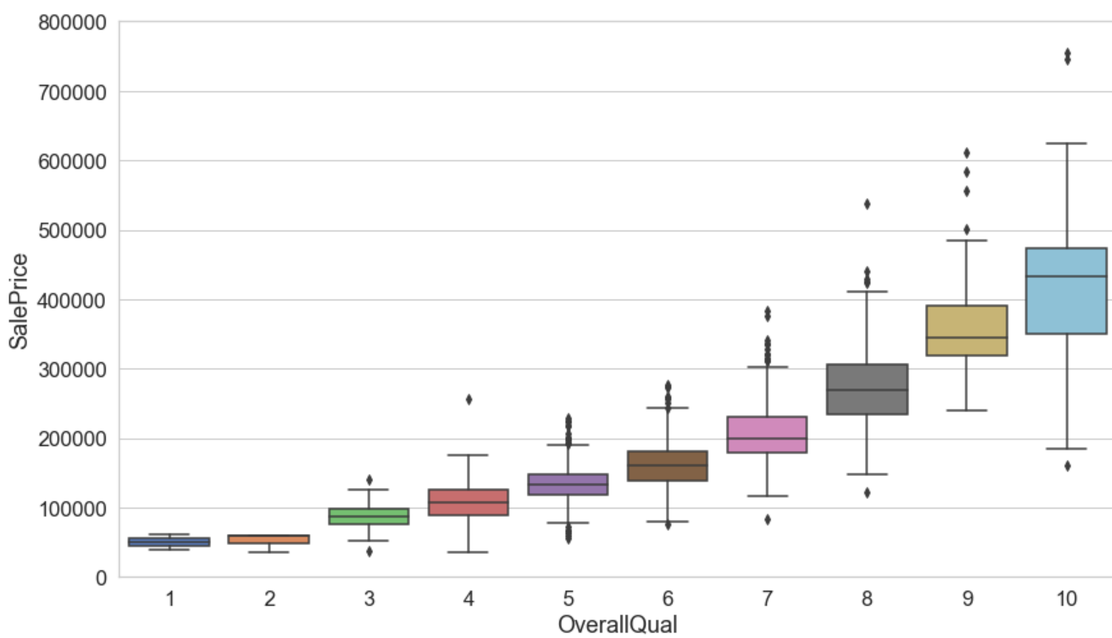
```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
*x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with
a single row if you intend to specify the same RGB or RGBA value for all points.
```



Here again a linear relation can be observed between SalePrice and TotalBsmtSF

Boxplot for OverallQuality and SalePrice relation

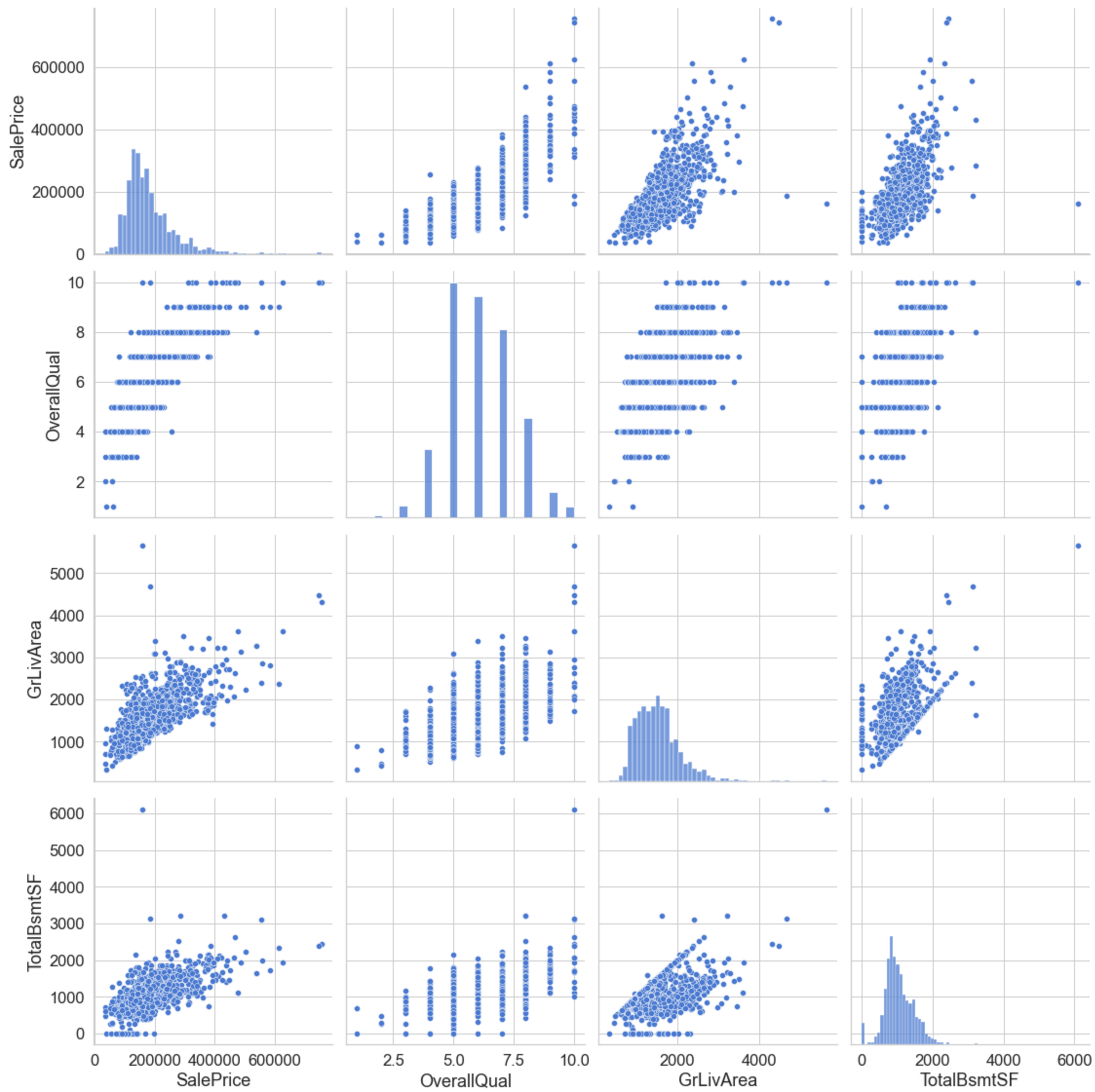
[49]:
```
var = 'OverallQual'
df = pd.concat([data['SalePrice'], data[var]], axis=1)
f, ax = plt.subplots(figsize=(14, 8))
fig = sns.boxplot(x=var, y="SalePrice", data=df)
fig.axis(ymin=0, ymax=800000);
```



It can be observed that there is a clear increase in Saleprice with respect to the Overall Quality Value.

Pairplot for the above features

[50]:
```
cols = ['SalePrice', 'OverallQual', 'GrLivArea', 'TotalBsmtSF']
sns.pairplot(data[cols], height = 4);
```



3 Featureset engineering and preparing the data

- creating a smaller dataframe for model training with non null values

- correcting right skew of 'SalePrice' column
- create new feature

Smaller dataframe for model training with non null values

[51]:
```
df= data[['OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea',
         'TotalBsmtSF', '1stFlrSF', 'FullBath', 'TotRmsAbvGrd', 'YearBuilt',
         'YearRemodAdd','Fireplaces', 'BsmtFinSF1',
         'WoodDeckSF', '2ndFlrSF', 'OpenPorchSF', 'HalfBath','SalePrice']]
df
```

[51]:
```
      OverallQual  GrLivArea  GarageCars  GarageArea  TotalBsmtSF  1stFlrSF  \
0               7       1710           2         548          856       856
1               6       1262           2         460         1262      1262
2               7       1786           2         608          920       920
3               7       1717           3         642          756       961
4               8       2198           3         836         1145      1145
…             …          …           …          …           …         …
1455            6       1647           2         460          953       953
1456            6       2073           2         500         1542      2073
1457            7       2340           1         252         1152      1188
1458            5       1078           1         240         1078      1078
1459            5       1256           1         276         1256      1256

      FullBath  TotRmsAbvGrd  YearBuilt  YearRemodAdd  Fireplaces  BsmtFinSF1  \
0            2             8       2003          2003           0         706
1            2             6       1976          1976           1         978
2            2             6       2001          2002           1         486
3            1             7       1915          1970           1         216
4            2             9       2000          2000           1         655
…          …            …          …           …            …          …
1455         2             7       1999          2000           1           0
1456         2             7       1978          1988           2         790
1457         2             9       1941          2006           2         275
1458         1             5       1950          1996           0          49
1459         1             6       1965          1965           0         830

      WoodDeckSF  2ndFlrSF  OpenPorchSF  HalfBath  SalePrice
0              0       854           61         1     208500
1            298         0            0         0     181500
2              0       866           42         1     223500
3              0       756           35         0     140000
4            192      1053           84         1     250000
…            …         …           …           …        …
1455           0       694           40         1     175000
1456         349         0            0         0     210000
1457           0      1152           60         0     266500
1458         366         0            0         0     142125
```

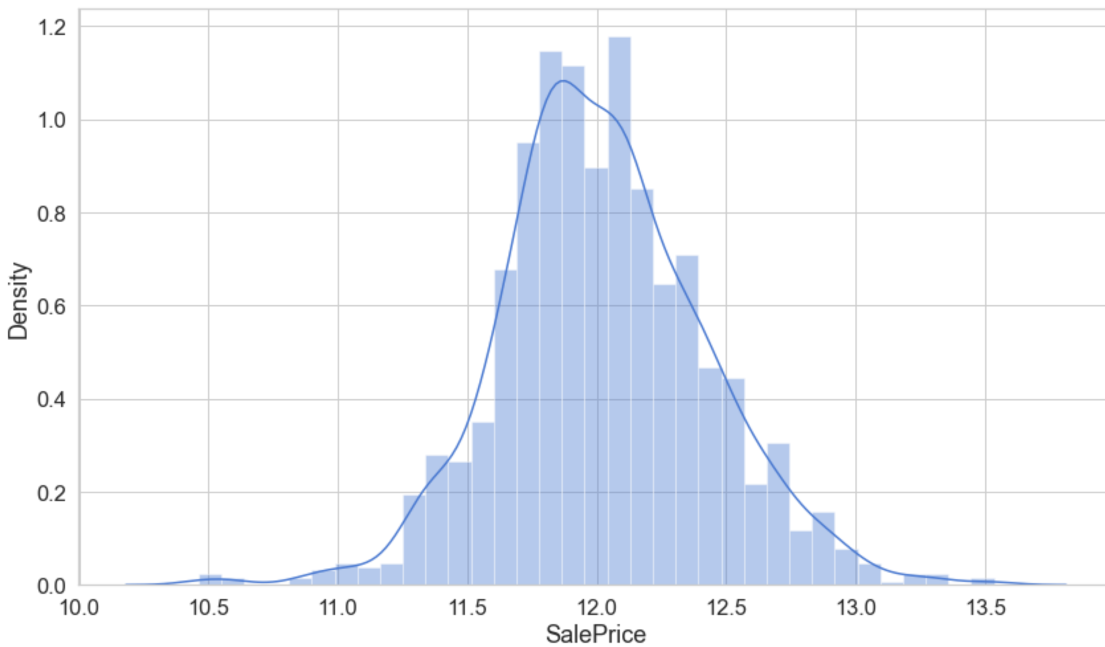
```
1459          736           0            68           1      147500

[1460 rows x 17 columns]
```

Checking and correcting the skewness of 'SalePrice' column

[52]:
```python
log_sale_price = np.log(df['SalePrice'])
sns.distplot(log_sale_price)
```

```
C:\Users\insan\anaconda\lib\site-packages\seaborn\distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

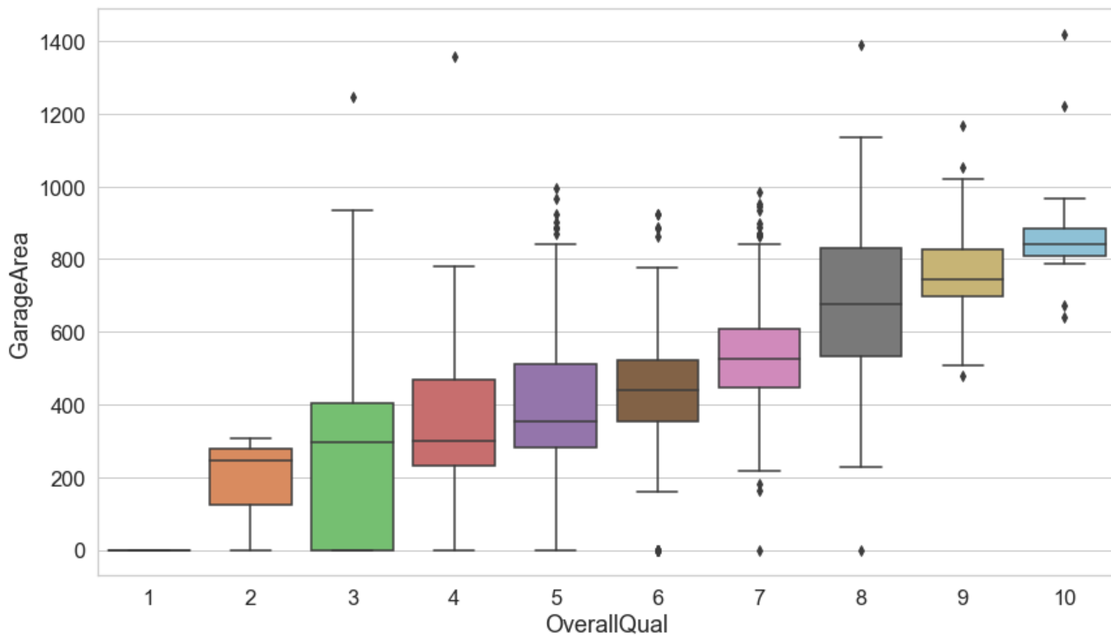
[52]: <AxesSubplot:xlabel='SalePrice', ylabel='Density'>



Here we can observe that the skewness is corrected hence we can update the column

Ploting the relation between Overall Quality value and GarageArea

[53]:
```python
var = 'OverallQual'
df2 = pd.concat([df['GarageArea'], df[var]], axis=1)
f, ax = plt.subplots(figsize=(14, 8))
fig = sns.boxplot(x=var, y="GarageArea", data=df2)
```



Creating a new feature which gives the garage area per car

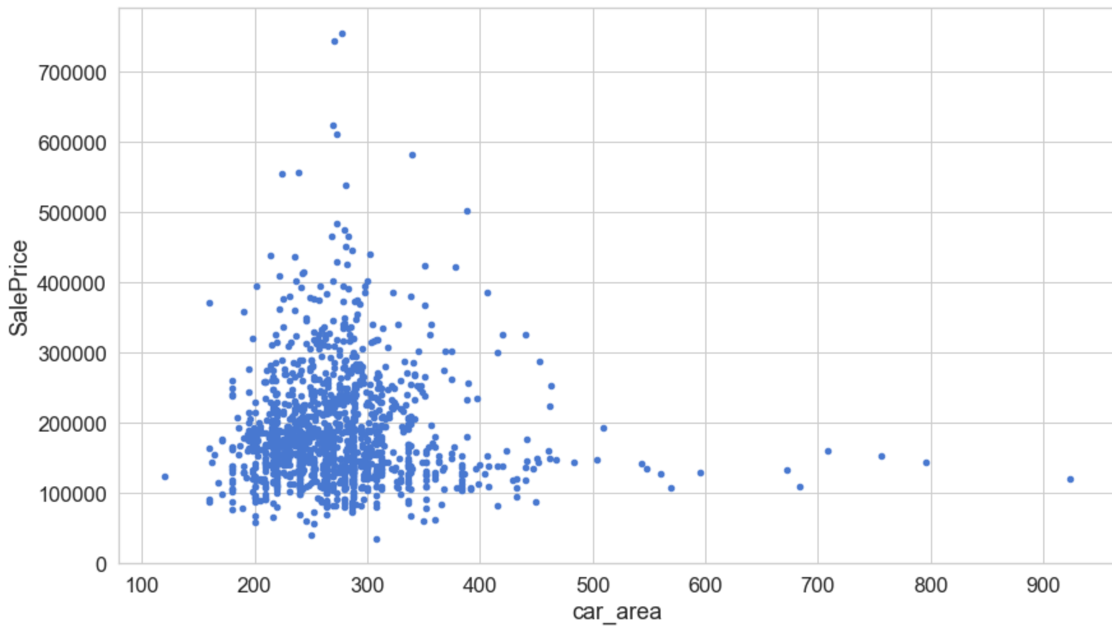
[54]:
```
df['car_area']=df['GarageArea']/df['GarageCars']
```

[55]:
```
df['car_area']
```

[55]:
```
0       274.000000
1       230.000000
2       304.000000
3       214.000000
4       278.666667
           …
1455    230.000000
1456    250.000000
1457    252.000000
1458    240.000000
1459    276.000000
Name: car_area, Length: 1460, dtype: float64
```

[56]:
```
var = 'car_area'
df2 = pd.concat([df['SalePrice'], df[var]], axis=1)
df2.plot.scatter(x=var, y='SalePrice');
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
*x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with
a single row if you intend to specify the same RGB or RGBA value for all points.
```



4 One hot encoding

in this section we will one hot encode the catogarical type features(columns)

[57]:
```
# Get a Pd.Series consisting of all the string categoricals
one_hot_encode_cols = data.dtypes[data.dtypes == object]  # filtering by string␣
↪categoricals
one_hot_encode_cols = one_hot_encode_cols.index.tolist()  # list of categorical␣
↪fields

# Here we see another way of one-hot-encoding:
# Encode these columns as categoricals so one hot encoding works on split data␣
↪(if desired)
for col in one_hot_encode_cols:
    data[col] = pd.Categorical(data[col])

# Do the one hot encoding
data = pd.get_dummies(data, columns=one_hot_encode_cols)
```

[58]:
```
data.shape
```

[58]: (1460, 290)

[59]:
```
total = data.isnull().sum().sort_values(ascending=False)
percent = (data.isnull().sum()/data.isnull().count()).
↪sort_values(ascending=False)
```

```
missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
missing_data.head(20)
```

[59]:

	Total	Percent
LotFrontage	259	0.177397
GarageYrBlt	81	0.055479
MasVnrArea	8	0.005479
Id	0	0.000000
BsmtExposure_Av	0	0.000000
BsmtFinType1_GLQ	0	0.000000
BsmtFinType1_BLQ	0	0.000000
BsmtFinType1_ALQ	0	0.000000
BsmtExposure_No	0	0.000000
BsmtExposure_Mn	0	0.000000
BsmtExposure_Gd	0	0.000000
BsmtCond_TA	0	0.000000
BsmtFinType1_Rec	0	0.000000
BsmtCond_Po	0	0.000000
BsmtCond_Gd	0	0.000000
BsmtCond_Fa	0	0.000000
BsmtQual_TA	0	0.000000
BsmtQual_Gd	0	0.000000
BsmtQual_Fa	0	0.000000
BsmtFinType1_LwQ	0	0.000000

[60]:
```
data["LotFrontage"].fillna(data["LotFrontage"].mean(), inplace = True)
data = data.dropna(axis = 0)
```

Next, split the data in train and test data sets.

[61]:
```
from sklearn.model_selection import train_test_split

train, test = train_test_split(data, test_size=0.3, random_state=42)
```

There are a number of columns that have skewed features–a log transformation can be applied to them. Note that this includes the `SalePrice`, our predictor. However, let's keep that one as is.

[62]:
```
# Create a list of float colums to check for skewing
mask = data.dtypes == float
float_cols = data.columns[mask]
```

[63]:
```
skew_limit = 0.75
skew_vals = train[float_cols].skew()

skew_cols = (skew_vals
             .sort_values(ascending=False)
             .to_frame()
             .rename(columns={0:'Skew'})
```

```
            .query('abs(Skew) > {0}'.format(skew_limit)))

skew_cols
```

[63]:

	Skew
LotFrontage	3.025243
MasVnrArea	2.573758

[64]:
```
# Mute the setting wtih a copy warnings
pd.options.mode.chained_assignment = None

for col in skew_cols.index.tolist():
    if col == "SalePrice":
        continue
    train[col] = np.log1p(train[col])
    test[col]  = test[col].apply(np.log1p)  # same thing
```

[65]:
```
feature_cols = [x for x in train.columns if x != 'SalePrice']
X_train = train[feature_cols]
y_train = train['SalePrice']

X_test  = test[feature_cols]
y_test  = test['SalePrice']
```

5 Model testing

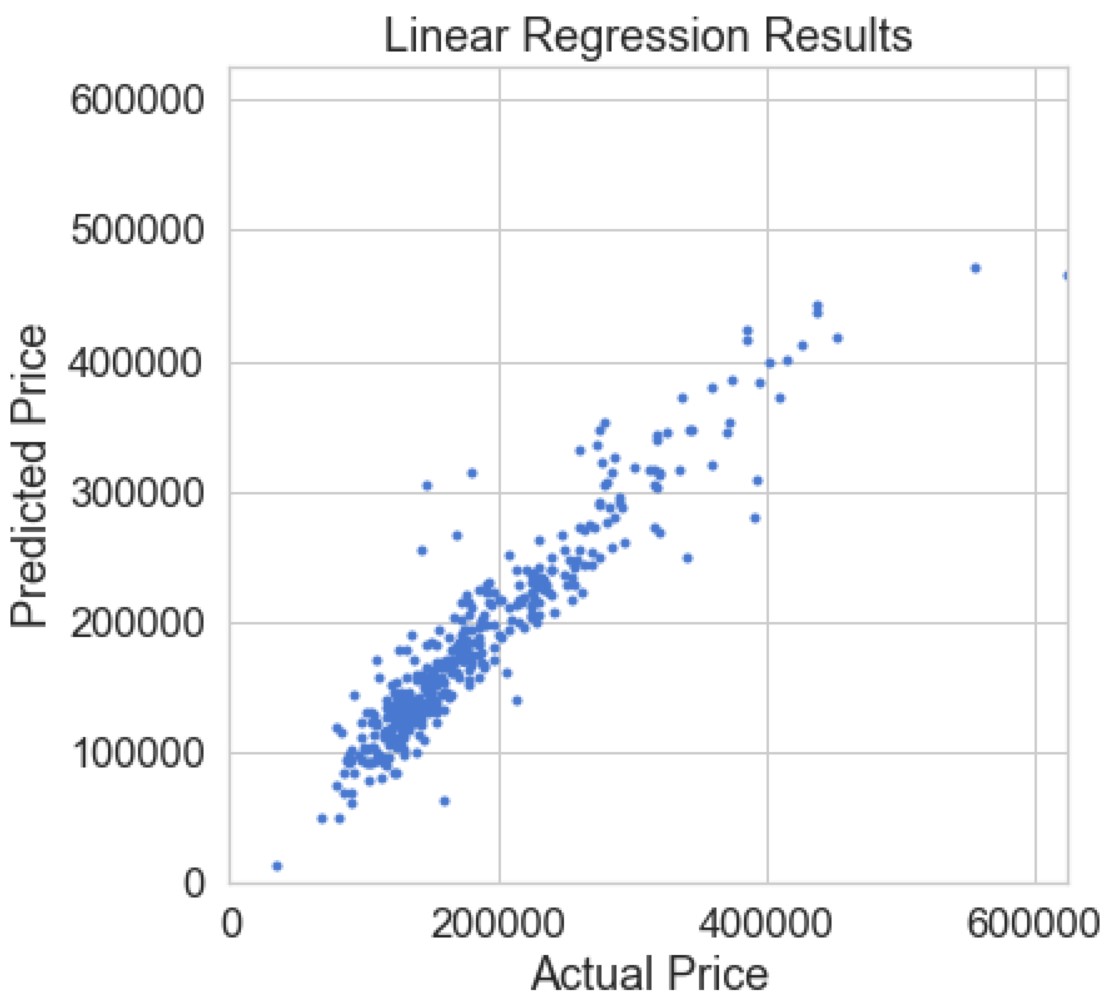
[66]:
```
from sklearn.metrics import mean_squared_error


def rmse(ytrue, ypredicted):
    return np.sqrt(mean_squared_error(ytrue, ypredicted))
```

- Fit a basic linear regression model
- print the root-mean-squared error for this model
- plot the predicted vs actual sale price based on the model.

[67]:
```
from sklearn.linear_model import LinearRegression

linearRegression = LinearRegression().fit(X_train, y_train)

linearRegression_rmse = rmse(y_test, linearRegression.predict(X_test))

print(linearRegression_rmse)
```

```
65126.24245777501
```

[68]:
```
f = plt.figure(figsize=(6,6))
ax = plt.axes()

ax.plot(y_test, linearRegression.predict(X_test),
         marker='o', ls='', ms=3.0)

lim = (0, y_test.max())

ax.set(xlabel='Actual Price',
       ylabel='Predicted Price',
       xlim=lim,
       ylim=lim,
       title='Linear Regression Results');
```



Ridge regression uses L2 normalization to reduce the magnitude of the coefficients. This can be helpful in situations where there is high variance. The regularization functions in Scikit-learn each contain versions that have cross-validation built in.

Fit a regular (non-cross validated) Ridge model to a range of values and plot the RMSE using the cross validated error function you created above.

Then repeat the fitting of the Ridge models using the range of values from the prior section. Compare the results. Now for the RidgeCV method. It's not possible to get the alpha values for the models that weren't selected, unfortunately. The resulting error values and values are very similar to those obtained above

[69]:
```
from sklearn.linear_model import RidgeCV

alphas = np.geomspace(1e1, 1e3, num=10)

ridgeCV = RidgeCV(alphas=alphas,
                  cv=4).fit(X_train, y_train)

ridgeCV_rmse = rmse(y_test, ridgeCV.predict(X_test))

print(ridgeCV.alpha_, ridgeCV_rmse)
```

```
10.0 28403.778242654047
```

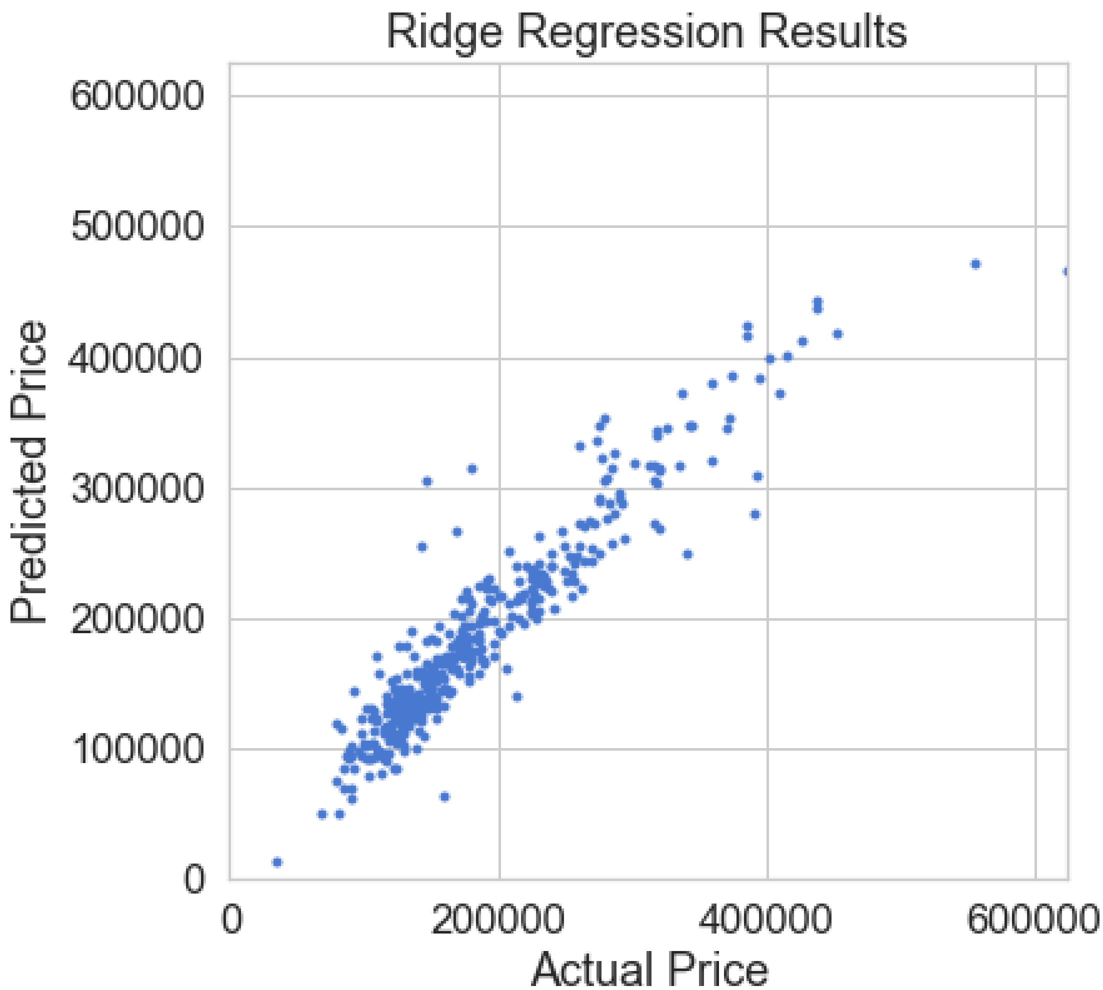
[70]:
```
rmse_vals = [linearRegression_rmse, ridgeCV_rmse]

labels = ['Linear', 'Ridge']

rmse_df = pd.Series(rmse_vals, index=labels).to_frame()
rmse_df.rename(columns={0: 'RMSE'}, inplace=1)
rmse_df
```

[70]:

	RMSE
Linear	65126.242458
Ridge	28403.778243

We can also make a plot of actual vs predicted housing prices as before.

[71]:
```
f = plt.figure(figsize=(6,6))
ax = plt.axes()

ax.plot(y_test, linearRegression.predict(X_test),
         marker='o', ls='', ms=3.0)

lim = (0, y_test.max())

ax.set(xlabel='Actual Price',
       ylabel='Predicted Price',
       xlim=lim,
       ylim=lim,
       title='Ridge Regression Results');
```



6 Next Steps

- One of the most important inputs to housing sale price prediction is the average value of a purticular location i.e average price per area. Such features could improve the model prediction while using this dataset for training.
- Gathering data on the nearby aminities such as distance to a hospital, school, bank etc. can help better determine SalePrice.
- Fitting and Testing LASSO and Elastic Net models to look for improved performance