top-1-approach-eda-new-models-and-stacking

August 27, 2021

1 Introduction

Hello all! In this notebook I'm going to implement what I gained on the way of learning. I'm doing this for learning purposes and share back to community what I learned. So there might be areas can be improved in future.

My main objectives on this project are:

- Applying exploratory data analysis and trying to get some insights about our dataset
- Getting data in better shape by transforming and feature engineering to help us in building better models
- Building and tuning couple models to get some stable results on predicting housing prices

In this notebook we are going to try explore the data we have and going try answer questions like:

- What are the main predictors for house pricing?
- What is more important on pricing, having big area for housing or just being in better neighborhood?
- Is quality of the house alone more important than having nice garages or basements?
- There are some features that can be modified and depends on the building but there are some other features like cannot be changed like location of the house, which group is effecting house prices?
- Can we predict the price of a house with the given traning data using machine learning techniques.
- What can our predictions achieve with different approaches?
- If we stack and blend the models, can we get more regularized results?

I hope you enjoy while reading it! And if you liked this kernel feel free to upvote and leave feedback, thanks!

[9]:
```
%pip install --upgrade scikit-learn

# Did this to use latest regressors from sklearn...
```

```
Requirement already satisfied: scikit-learn in
c:\users\pavlo\anaconda3\lib\site-packages (0.24.2)Note: you may need to restart
the kernel to use updated packages.

Requirement already satisfied: numpy>=1.13.3 in
```

```
c:\users\pavlo\anaconda3\lib\site-packages (from scikit-learn) (1.19.5)
Requirement already satisfied: scipy>=0.19.1 in
c:\users\pavlo\anaconda3\lib\site-packages (from scikit-learn) (1.6.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\users\pavlo\anaconda3\lib\site-packages (from scikit-learn) (2.1.0)
Requirement already satisfied: joblib>=0.11 in
c:\users\pavlo\anaconda3\lib\site-packages (from scikit-learn) (1.0.1)
```

[10]:
```
# Loading neccesary packages:

import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime

#

from scipy import stats
from scipy.stats import skew, boxcox_normmax, norm
from scipy.special import boxcox1p

#

import matplotlib.gridspec as gridspec
from matplotlib.ticker import MaxNLocator

#

import warnings
pd.options.display.max_columns = 250
pd.options.display.max_rows = 250
warnings.filterwarnings('ignore')
plt.style.use('fivethirtyeight')
```

2 Meeting the data

We're going to start by loading the data and taking first look on it as usual. For the column names we have great dictionary file in our dataset location so we can get familiar with them in no time.

[11]:
```
# Loading datasets.

train = pd.read_csv('house_price/input/train.csv')
test = pd.read_csv('house_price/input/test.csv')
```

[12]:
```
train.shape
```

[12]: (1460, 81)

[13]:
```
test.shape
```

[13]: (1459, 80)

[14]:
```
train.head()
```

[14]:
```
   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 LotConfig LandSlope Neighborhood Condition1  \
0         Lvl    AllPub    Inside       Gtl      CollgCr       Norm
1         Lvl    AllPub       FR2       Gtl      Veenker      Feedr
2         Lvl    AllPub    Inside       Gtl      CollgCr       Norm
3         Lvl    AllPub    Corner       Gtl      Crawfor       Norm
4         Lvl    AllPub       FR2       Gtl      NoRidge       Norm

  Condition2 BldgType HouseStyle  OverallQual  OverallCond  YearBuilt  \
0       Norm     1Fam     2Story            7            5       2003
1       Norm     1Fam     1Story            6            8       1976
2       Norm     1Fam     2Story            7            5       2001
3       Norm     1Fam     2Story            7            5       1915
4       Norm     1Fam     2Story            8            5       2000

   YearRemodAdd RoofStyle RoofMatl Exterior1st Exterior2nd MasVnrType  \
0          2003     Gable  CompShg     VinylSd     VinylSd    BrkFace
1          1976     Gable  CompShg     MetalSd     MetalSd       None
2          2002     Gable  CompShg     VinylSd     VinylSd    BrkFace
3          1970     Gable  CompShg     Wd Sdng     Wd Shng       None
4          2000     Gable  CompShg     VinylSd     VinylSd    BrkFace

   MasVnrArea ExterQual ExterCond Foundation BsmtQual BsmtCond BsmtExposure  \
0       196.0        Gd        TA      PConc       Gd       TA           No
1         0.0        TA        TA     CBlock       Gd       TA           Gd
2       162.0        Gd        TA      PConc       Gd       TA           Mn
3         0.0        TA        TA     BrkTil       TA       Gd           No
4       350.0        Gd        TA      PConc       Gd       TA           Av

  BsmtFinType1  BsmtFinSF1 BsmtFinType2  BsmtFinSF2  BsmtUnfSF  TotalBsmtSF  \
0          GLQ         706          Unf           0        150          856
1          ALQ         978          Unf           0        284         1262
2          GLQ         486          Unf           0        434          920
```

```
3          ALQ         216          Unf           0         540          756
4          GLQ         655          Unf           0         490         1145

  Heating HeatingQC CentralAir Electrical  1stFlrSF  2ndFlrSF  LowQualFinSF  \
0    GasA        Ex          Y      SBrkr       856       854             0
1    GasA        Ex          Y      SBrkr      1262         0             0
2    GasA        Ex          Y      SBrkr       920       866             0
3    GasA        Gd          Y      SBrkr       961       756             0
4    GasA        Ex          Y      SBrkr      1145      1053             0

   GrLivArea  BsmtFullBath  BsmtHalfBath  FullBath  HalfBath  BedroomAbvGr  \
0       1710             1             0         2         1             3
1       1262             0             1         2         0             3
2       1786             1             0         2         1             3
3       1717             1             0         1         0             3
4       2198             1             0         2         1             4

   KitchenAbvGr KitchenQual  TotRmsAbvGrd Functional  Fireplaces FireplaceQu  \
0             1          Gd             8        Typ           0         NaN
1             1          TA             6        Typ           1          TA
2             1          Gd             6        Typ           1          TA
3             1          Gd             7        Typ           1          Gd
4             1          Gd             9        Typ           1          TA

  GarageType  GarageYrBlt GarageFinish  GarageCars  GarageArea GarageQual  \
0     Attchd       2003.0          RFn           2         548         TA
1     Attchd       1976.0          RFn           2         460         TA
2     Attchd       2001.0          RFn           2         608         TA
3     Detchd       1998.0          Unf           3         642         TA
4     Attchd       2000.0          RFn           3         836         TA

  GarageCond PavedDrive  WoodDeckSF  OpenPorchSF  EnclosedPorch  3SsnPorch  \
0         TA          Y           0           61              0          0
1         TA          Y         298            0              0          0
2         TA          Y           0           42              0          0
3         TA          Y           0           35            272          0
4         TA          Y         192           84              0          0

   ScreenPorch  PoolArea PoolQC Fence MiscFeature  MiscVal  MoSold  YrSold  \
0            0         0    NaN   NaN         NaN        0       2    2008
1            0         0    NaN   NaN         NaN        0       5    2007
2            0         0    NaN   NaN         NaN        0       9    2008
3            0         0    NaN   NaN         NaN        0       2    2006
4            0         0    NaN   NaN         NaN        0      12    2008

  SaleType SaleCondition  SalePrice
0       WD        Normal     208500
```

```
1       WD         Normal     181500
2       WD         Normal     223500
3       WD        Abnorml     140000
4       WD         Normal     250000
```

[15]:
```
test.head()
```

[15]:
```
     Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
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

  LandContour Utilities LotConfig LandSlope Neighborhood Condition1  \
0         Lvl    AllPub    Inside       Gtl        NAmes      Feedr
1         Lvl    AllPub    Corner       Gtl        NAmes       Norm
2         Lvl    AllPub    Inside       Gtl      Gilbert       Norm
3         Lvl    AllPub    Inside       Gtl      Gilbert       Norm
4         HLS    AllPub    Inside       Gtl      StoneBr       Norm

  Condition2 BldgType HouseStyle  OverallQual  OverallCond  YearBuilt  \
0       Norm     1Fam     1Story            5            6       1961
1       Norm     1Fam     1Story            6            6       1958
2       Norm     1Fam     2Story            5            5       1997
3       Norm     1Fam     2Story            6            6       1998
4       Norm   TwnhsE     1Story            8            5       1992

   YearRemodAdd RoofStyle RoofMatl Exterior1st Exterior2nd MasVnrType  \
0          1961     Gable  CompShg     VinylSd     VinylSd       None
1          1958       Hip  CompShg     Wd Sdng     Wd Sdng    BrkFace
2          1998     Gable  CompShg     VinylSd     VinylSd       None
3          1998     Gable  CompShg     VinylSd     VinylSd    BrkFace
4          1992     Gable  CompShg     HdBoard     HdBoard       None

   MasVnrArea ExterQual ExterCond Foundation BsmtQual BsmtCond BsmtExposure  \
0         0.0        TA        TA     CBlock       TA       TA           No
1       108.0        TA        TA     CBlock       TA       TA           No
2         0.0        TA        TA      PConc       Gd       TA           No
3        20.0        TA        TA      PConc       TA       TA           No
4         0.0        Gd        TA      PConc       Gd       TA           No

  BsmtFinType1  BsmtFinSF1 BsmtFinType2  BsmtFinSF2  BsmtUnfSF  TotalBsmtSF  \
0          Rec       468.0          LwQ       144.0      270.0        882.0
1          ALQ       923.0          Unf         0.0      406.0       1329.0
2          GLQ       791.0          Unf         0.0      137.0        928.0
3          GLQ       602.0          Unf         0.0      324.0        926.0
```

```
4          ALQ       263.0          Unf         0.0     1017.0       1280.0

  Heating HeatingQC CentralAir Electrical  1stFlrSF  2ndFlrSF  LowQualFinSF  \
0    GasA        TA          Y      SBrkr       896         0             0
1    GasA        TA          Y      SBrkr      1329         0             0
2    GasA        Gd          Y      SBrkr       928       701             0
3    GasA        Ex          Y      SBrkr       926       678             0
4    GasA        Ex          Y      SBrkr      1280         0             0

   GrLivArea  BsmtFullBath  BsmtHalfBath  FullBath  HalfBath  BedroomAbvGr  \
0        896           0.0           0.0         1         0             2
1       1329           0.0           0.0         1         1             3
2       1629           0.0           0.0         2         1             3
3       1604           0.0           0.0         2         1             3
4       1280           0.0           0.0         2         0             2

   KitchenAbvGr KitchenQual  TotRmsAbvGrd Functional  Fireplaces FireplaceQu  \
0             1          TA             5        Typ           0         NaN
1             1          Gd             6        Typ           0         NaN
2             1          TA             6        Typ           1          TA
3             1          Gd             7        Typ           1          Gd
4             1          Gd             5        Typ           0         NaN

  GarageType  GarageYrBlt GarageFinish  GarageCars  GarageArea GarageQual  \
0     Attchd       1961.0          Unf         1.0       730.0         TA
1     Attchd       1958.0          Unf         1.0       312.0         TA
2     Attchd       1997.0          Fin         2.0       482.0         TA
3     Attchd       1998.0          Fin         2.0       470.0         TA
4     Attchd       1992.0          RFn         2.0       506.0         TA

  GarageCond PavedDrive  WoodDeckSF  OpenPorchSF  EnclosedPorch  3SsnPorch  \
0         TA          Y         140            0              0          0
1         TA          Y         393           36              0          0
2         TA          Y         212           34              0          0
3         TA          Y         360           36              0          0
4         TA          Y           0           82              0          0

   ScreenPorch  PoolArea PoolQC  Fence MiscFeature  MiscVal  MoSold  YrSold  \
0          120         0    NaN  MnPrv         NaN        0       6    2010
1            0         0    NaN    NaN        Gar2    12500       6    2010
2            0         0    NaN  MnPrv         NaN        0       3    2010
3            0         0    NaN    NaN         NaN        0       6    2010
4          144         0    NaN    NaN         NaN        0       1    2010

  SaleType SaleCondition
0       WD        Normal
1       WD        Normal
```

```
2         WD         Normal
3         WD         Normal
4         WD         Normal
```

[16]:
```
train.describe()
```

[16]:
```
                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

        BsmtFinSF2    BsmtUnfSF  TotalBsmtSF     1stFlrSF     2ndFlrSF  \
count  1460.000000  1460.000000  1460.000000  1460.000000  1460.000000
mean     46.549315   567.240411  1057.429452  1162.626712   346.992466
std     161.319273   441.866955   438.705324   386.587738   436.528436
min       0.000000     0.000000     0.000000   334.000000     0.000000
25%       0.000000   223.000000   795.750000   882.000000     0.000000
50%       0.000000   477.500000   991.500000  1087.000000     0.000000
75%       0.000000   808.000000  1298.250000  1391.250000   728.000000
max    1474.000000  2336.000000  6110.000000  4692.000000  2065.000000

       LowQualFinSF    GrLivArea  BsmtFullBath  BsmtHalfBath     FullBath  \
count   1460.000000  1460.000000   1460.000000   1460.000000  1460.000000
mean       5.844521  1515.463699      0.425342      0.057534     1.565068
std       48.623081   525.480383      0.518911      0.238753     0.550916
min        0.000000   334.000000      0.000000      0.000000     0.000000
25%        0.000000  1129.500000      0.000000      0.000000     1.000000
50%        0.000000  1464.000000      0.000000      0.000000     2.000000
75%        0.000000  1776.750000      1.000000      0.000000     2.000000
max      572.000000  5642.000000      3.000000      2.000000     3.000000

          HalfBath  BedroomAbvGr  KitchenAbvGr  TotRmsAbvGrd   Fireplaces  \
```

count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000
mean	0.382877	2.866438	1.046575	6.517808	0.613014
std	0.502885	0.815778	0.220338	1.625393	0.644666
min	0.000000	0.000000	0.000000	2.000000	0.000000
25%	0.000000	2.000000	1.000000	5.000000	0.000000
50%	0.000000	3.000000	1.000000	6.000000	1.000000
75%	1.000000	3.000000	1.000000	7.000000	1.000000
max	2.000000	8.000000	3.000000	14.000000	3.000000

	GarageYrBlt	GarageCars	GarageArea	WoodDeckSF	OpenPorchSF	\
count	1379.000000	1460.000000	1460.000000	1460.000000	1460.000000	
mean	1978.506164	1.767123	472.980137	94.244521	46.660274	
std	24.689725	0.747315	213.804841	125.338794	66.256028	
min	1900.000000	0.000000	0.000000	0.000000	0.000000	
25%	1961.000000	1.000000	334.500000	0.000000	0.000000	
50%	1980.000000	2.000000	480.000000	0.000000	25.000000	
75%	2002.000000	2.000000	576.000000	168.000000	68.000000	
max	2010.000000	4.000000	1418.000000	857.000000	547.000000	

	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea	MiscVal	\
count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	
mean	21.954110	3.409589	15.060959	2.758904	43.489041	
std	61.119149	29.317331	55.757415	40.177307	496.123024	
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	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	0.000000	
max	552.000000	508.000000	480.000000	738.000000	15500.000000	

	MoSold	YrSold	SalePrice
count	1460.000000	1460.000000	1460.000000
mean	6.321918	2007.815753	180921.195890
std	2.703626	1.328095	79442.502883
min	1.000000	2006.000000	34900.000000
25%	5.000000	2007.000000	129975.000000
50%	6.000000	2008.000000	163000.000000
75%	8.000000	2009.000000	214000.000000
max	12.000000	2010.000000	755000.000000

[17]:
```
test.describe()
```

[17]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	\
count	1459.000000	1459.000000	1232.000000	1459.000000	1459.000000	
mean	2190.000000	57.378341	68.580357	9819.161069	6.078821	
std	421.321334	42.746880	22.376841	4955.517327	1.436812	
min	1461.000000	20.000000	21.000000	1470.000000	1.000000	
25%	1825.500000	20.000000	58.000000	7391.000000	5.000000	

```
50%    2190.000000   50.000000   67.000000   9399.000000    6.000000
75%    2554.500000   70.000000   80.000000  11517.500000    7.000000
max    2919.000000  190.000000  200.000000  56600.000000   10.000000

       OverallCond    YearBuilt  YearRemodAdd   MasVnrArea   BsmtFinSF1  \
count  1459.000000  1459.000000   1459.000000  1444.000000  1458.000000
mean      5.553804  1971.357779   1983.662783   100.709141   439.203704
std       1.113740    30.390071     21.130467   177.625900   455.268042
min       1.000000  1879.000000   1950.000000     0.000000     0.000000
25%       5.000000  1953.000000   1963.000000     0.000000     0.000000
50%       5.000000  1973.000000   1992.000000     0.000000   350.500000
75%       6.000000  2001.000000   2004.000000   164.000000   753.500000
max       9.000000  2010.000000   2010.000000  1290.000000  4010.000000

        BsmtFinSF2    BsmtUnfSF  TotalBsmtSF     1stFlrSF     2ndFlrSF  \
count  1458.000000  1458.000000  1458.000000  1459.000000  1459.000000
mean     52.619342   554.294925  1046.117970  1156.534613   325.967786
std     176.753926   437.260486   442.898624   398.165820   420.610226
min       0.000000     0.000000     0.000000   407.000000     0.000000
25%       0.000000   219.250000   784.000000   873.500000     0.000000
50%       0.000000   460.000000   988.000000  1079.000000     0.000000
75%       0.000000   797.750000  1305.000000  1382.500000   676.000000
max    1526.000000  2140.000000  5095.000000  5095.000000  1862.000000

       LowQualFinSF    GrLivArea  BsmtFullBath  BsmtHalfBath     FullBath  \
count   1459.000000  1459.000000   1457.000000   1457.000000  1459.000000
mean       3.543523  1486.045922      0.434454      0.065202     1.570939
std       44.043251   485.566099      0.530648      0.252468     0.555190
min        0.000000   407.000000      0.000000      0.000000     0.000000
25%        0.000000  1117.500000      0.000000      0.000000     1.000000
50%        0.000000  1432.000000      0.000000      0.000000     2.000000
75%        0.000000  1721.000000      1.000000      0.000000     2.000000
max     1064.000000  5095.000000      3.000000      2.000000     4.000000

          HalfBath  BedroomAbvGr  KitchenAbvGr  TotRmsAbvGrd  Fireplaces  \
count  1459.000000   1459.000000   1459.000000   1459.000000  1459.00000
mean      0.377656      2.854010      1.042495      6.385195     0.58122
std       0.503017      0.829788      0.208472      1.508895     0.64742
min       0.000000      0.000000      0.000000      3.000000     0.00000
25%       0.000000      2.000000      1.000000      5.000000     0.00000
50%       0.000000      3.000000      1.000000      6.000000     0.00000
75%       1.000000      3.000000      1.000000      7.000000     1.00000
max       2.000000      6.000000      2.000000     15.000000     4.00000

       GarageYrBlt   GarageCars   GarageArea   WoodDeckSF  OpenPorchSF  \
count  1381.000000  1458.000000  1458.000000  1459.000000  1459.000000
mean   1977.721217     1.766118   472.768861    93.174777    48.313914
```

```
std      26.431175    0.775945   217.048611   127.744882    68.883364
min    1895.000000    0.000000     0.000000     0.000000     0.000000
25%    1959.000000    1.000000   318.000000     0.000000     0.000000
50%    1979.000000    2.000000   480.000000     0.000000    28.000000
75%    2002.000000    2.000000   576.000000   168.000000    72.000000
max    2207.000000    5.000000  1488.000000  1424.000000   742.000000

       EnclosedPorch    3SsnPorch  ScreenPorch     PoolArea       MiscVal  \
count    1459.000000  1459.000000  1459.000000  1459.000000   1459.000000
mean       24.243317     1.794380    17.064428     1.744345     58.167923
std        67.227765    20.207842    56.609763    30.491646    630.806978
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     0.000000     0.000000     0.000000      0.000000
75%         0.000000     0.000000     0.000000     0.000000      0.000000
max      1012.000000   360.000000   576.000000   800.000000  17000.000000

            MoSold       YrSold
count  1459.000000  1459.000000
mean      6.104181  2007.769705
std       2.722432     1.301740
min       1.000000  2006.000000
25%       4.000000  2007.000000
50%       6.000000  2008.000000
75%       8.000000  2009.000000
max      12.000000  2010.000000
```

- **Id column looks useless we can safely drop it from both. I'm going to save our target (SalePrice) on different variable so we can use it in future.**

[18]:
```python
# Dropping unnecessary Id column.

train.drop('Id', axis=1, inplace=True)
test.drop('Id', axis=1, inplace=True)
```

[19]:
```python
# Backing up target variables and dropping them from train data.

y = train['SalePrice'].reset_index(drop=True)
train_features = train.drop(['SalePrice'], axis=1)
test_features = test
```

3 Analysis Time!

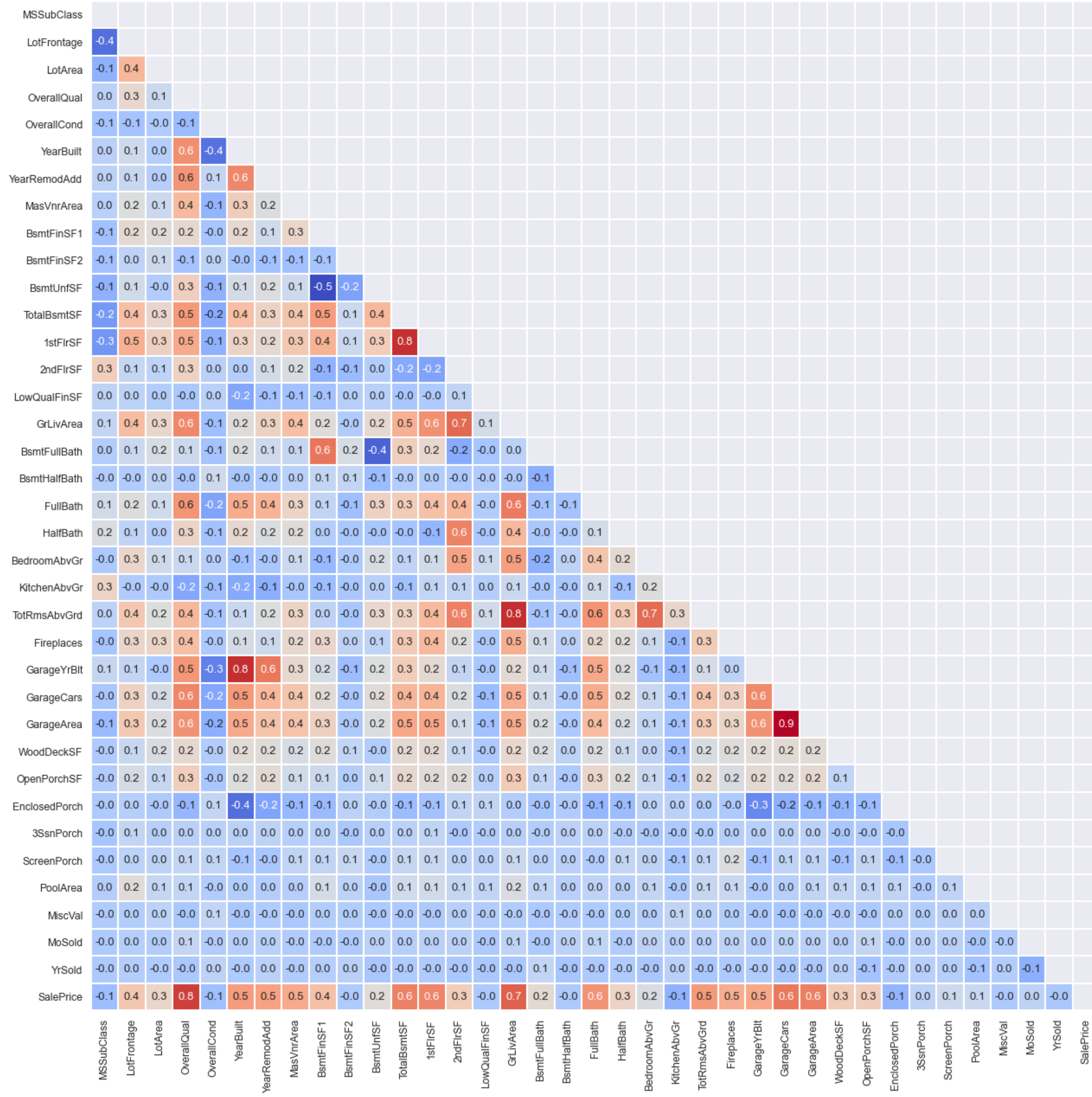
Ok the short inspection at the beginning give us some hints how should we move from here. I'm going to play with the data we have while analysing the data at the same time. With this way I hope we can get the data in better shape while digging deeper into it.

We're going to start with basic correlation table here. I dropped the top part since it's just mirror of the other part below. With this table we can understand some linear relations between different features.

Observations:

- There's strong relation between overall quality of the houses and their sale prices.
- Again above grade living area seems strong indicator for sale price.
- Garage features, number of baths and rooms, how old the building is etc. also having effect on the price on various levels too.
- There are some obvious relations we gonna pass like total square feet affecting how many rooms there are or how many cars can fit into a garage vs. garage area etc.
- Overall condition of the house seems less important on the pricing, it's interesting and worth digging.

[20]:
```
# Display numerical correlations (pearson) between features on heatmap.

sns.set(font_scale=1.1)
correlation_train = train.corr()
mask = np.triu(correlation_train.corr())
plt.figure(figsize=(20, 20))
sns.heatmap(correlation_train,
            annot=True,
            fmt='.1f',
            cmap='coolwarm',
            square=True,
            mask=mask,
            linewidths=1,
            cbar=False)

plt.show()
```



- **I'm going to merge the datasets here before we start editing it so we don't have to do these operations twice. Let's call it features since it has features only. So our data has 2919 observations and 79 features to begin with...**

[21]:
```python
# Merging train test features for engineering.

features = pd.concat([train_features, test_features]).reset_index(drop=True)
print(features.shape)
```

```
(2919, 79)
```

3.1 Missing Data

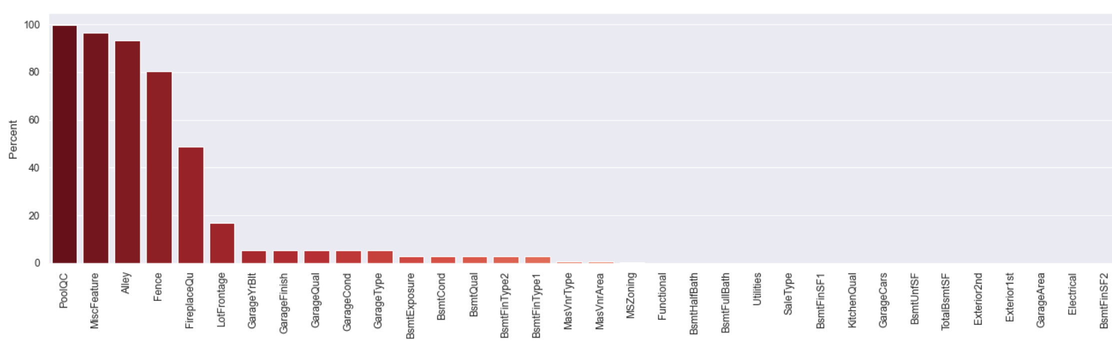
Alright, first of all we need detect missing values, then wee need to get rid of them for the next steps of our work. So let's list our missing values and visualize them:

[22]:
```python
def missing_percentage(df):

    """A function for returning missing ratios."""

    total = df.isnull().sum().sort_values(
        ascending=False)[df.isnull().sum().sort_values(ascending=False) != 0]
    percent = (df.isnull().sum().sort_values(ascending=False) / len(df) *
               100)[(df.isnull().sum().sort_values(ascending=False) / len(df) *
                     100) != 0]
    return pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
```

- **That's quite a lot! No need to panic though we got this. If you look at the data description given to us we can see that most of these missing data actually not missing, it's just means house doesn't have that specific feature, we can fix that easily...**

[23]:
```python
# Checking 'NaN' values.

missing = missing_percentage(features)

fig, ax = plt.subplots(figsize=(20, 5))
sns.barplot(x=missing.index, y='Percent', data=missing, palette='Reds_r')
plt.xticks(rotation=90)

display(missing.T.style.background_gradient(cmap='Reds', axis=1))
```

```
<pandas.io.formats.style.Styler at 0x2c4f61a2f40>
```



3.1.1 Ok this is how we gonna fix most of the missing data:

1. First we fill the NaN's in the columns where they mean 'None' so we gonna replace them with that,

2. Then we fill numerical columns where missing values indicating there is no parent feature to measure, so we replace them with 0's.
3. Even with these there are some actual missing data, by checking general trends of these features we can fill them with most frequent value(with mode).
4. MSZoning part is little bit tricky I choose to fill them with most common type of the related MSSubClass type. It's not perfect but at least we decrease randomness a little bit.
5. Again we fill the Lot Frontage with similar approach.

[24]:
```
# List of 'NaN' including columns where NaN's mean none.

none_cols = [
    'Alley', 'PoolQC', 'MiscFeature', 'Fence', 'FireplaceQu', 'GarageType',
    'GarageFinish', 'GarageQual', 'GarageCond', 'BsmtQual', 'BsmtCond',
    'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'MasVnrType'
]

# List of 'NaN' including columns where NaN's mean 0.

zero_cols = [
    'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'BsmtFullBath',
    'BsmtHalfBath', 'GarageYrBlt', 'GarageArea', 'GarageCars', 'MasVnrArea'
]

# List of 'NaN' including columns where NaN's actually missing gonna replaced␣
↪with mode.

freq_cols = [
    'Electrical', 'Exterior1st', 'Exterior2nd', 'Functional', 'KitchenQual',
    'SaleType', 'Utilities'
]

# Filling the list of columns above with appropriate values:

for col in zero_cols:
    features[col].replace(np.nan, 0, inplace=True)

for col in none_cols:
    features[col].replace(np.nan, 'None', inplace=True)

for col in freq_cols:
    features[col].replace(np.nan, features[col].mode()[0], inplace=True)
```

[25]:
```
# Filling 'MSZoning' according to MSSubClass.

features['MSZoning'] = features.groupby('MSSubClass')['MSZoning'].apply(
    lambda x: x.fillna(x.mode()[0]))
```

[26]:
```
# Filling 'MSZoning' according to Neighborhood.

features['LotFrontage'] = features.groupby(
    ['Neighborhood'])['LotFrontage'].apply(lambda x: x.fillna(x.median()))
```

[27]:
```
# Features which numerical on data but should be treated as category:

features['MSSubClass'] = features['MSSubClass'].astype(str)
features['YrSold'] = features['YrSold'].astype(str)
features['MoSold'] = features['MoSold'].astype(str)
```

4 Feature Engineering

Ok this is the part where we dig deeper into our completed dataset. There are no missing values so we're good to go! I'm going to start with grouping some values, these values are really rare and I'm thinking they do not add much, so if they appear less than 10 times in our observations they get into 'Other' group.

[28]:
```
# Transforming rare values(less than 10) into one group.

others = [
    'Condition1', 'Condition2', 'RoofMatl', 'Exterior1st', 'Exterior2nd',
    'Heating', 'Electrical', 'Functional', 'SaleType'
]

for col in others:
    mask = features[col].isin(
        features[col].value_counts()[features[col].value_counts() < 10].index)
    features[col][mask] = 'Other'
```

[29]:
```
def srt_box(y, df):

    '''A function for displaying categorical variables.'''

    fig, axes = plt.subplots(14, 3, figsize=(25, 80))
    axes = axes.flatten()

    for i, j in zip(df.select_dtypes(include=['object']).columns, axes):

        sortd = df.groupby([i])[y].median().sort_values(ascending=False)
        sns.boxplot(x=i,
                    y=y,
                    data=df,
                    palette='plasma',
                    order=sortd.index,
                    ax=j)
        j.tick_params(labelrotation=45)
```

```
        j.yaxis.set_major_locator(MaxNLocator(nbins=18))

        plt.tight_layout()
```

5 Categorical Data

We already checked some of the numerical features with correlation heatmap but what about categorical values? We want to see relations between categorical data and sale price. Boxplots seems decent way to inspect this type of relation. We're also going to sort them by the median value of that group so we can see the importances in descending order.

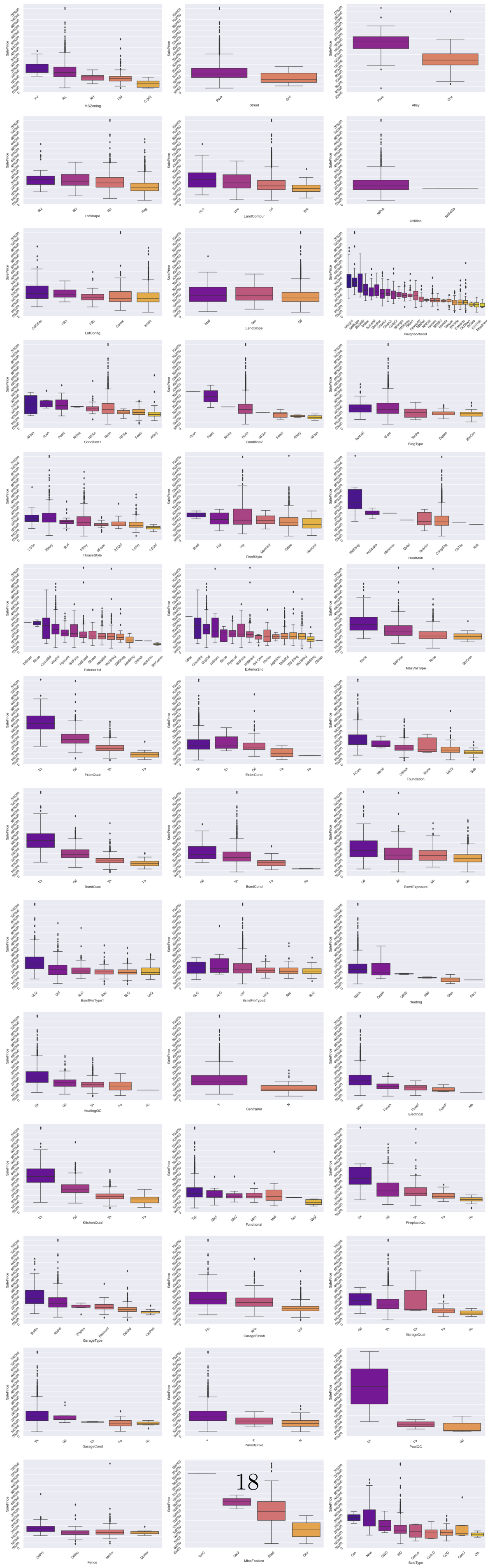
Observations:

- **MSZoning;**
- Floating village houses (I assume they are some kind of special area that retired community resides, has the highest median value.
- Residental low density houses comes second with the some outliers.
- Residental high and low seems similar meanwhile commercial is the lowest.
- **LandContour; Hillside houses seems a little bit higher expensive than the rest meanwhile banked houses are the lowest.**
- **Neighborhood;**
- Northridge Heights, Northridge and Timberland are top 3 expensive places for houses.
- Somerset, Veenker, Crawford, Clear Creek, College Creek and Bloomington Heights seems above average.
- Sawyer West has wide range for prices related to similar priced regions.
- Old Town and Edwards has some outlier prices but they generally below average.
- Briardale, Iowa DOT and Rail Road, Meadow Village are the cheapest places for houses it seems…
- **Conditions;**
- Meanwhile having wide range of values being close to North-South Railroad seems having positive effect on the price.
- Being near or adjacent to positive off-site feature (park, greenbelt, etc.) increases the price.
- These values are pretty similar but we can get some useful information from them.
- **MasVnrType;** Having stone masonry veneer seems better priced than having brick.
- **Quality Features;** There are many categorical quality values that affects the pricing on some degree, we're going to quantify them so we can create new features based on them. So we don't dive deep on them in this part.
- **CentralAir;** Having central air system has decent positive effect on sale prices.

- **GarageType;**
  - Built-In (Garage part of house - typically has room above garage) garage typed houses are the most expensive ones.
  - Attached garage types following the built-in ones.
  - Car ports are the lowest
- **Misc;** Sale type has some kind of effect on the prices but we won't get into details here. Btw… It seems having tennis court is really adding price to your house, who would have known :)

Alright, we're done with categorical data inspecting, I'm going to convert some of these categories to numerical ones, especially the ones where related to quality of the specific features.

[30]:
```
# Displaying sale prices vs. categorical values:

srt_box('SalePrice', train)
```


SalePrice
MSZoning
Street
Alley
LotShape
LandContour
Utilities
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
GarageCond
PavedDrive
PoolQC
Fence
MiscFeature
SaleType

[31]:
```
# Converting some of the categorical values to numeric ones. Choosing similar␣
↪values for closer groups to balance linear relations...

neigh_map = {
    'MeadowV': 1,
    'IDOTRR': 1,
    'BrDale': 1,
    'BrkSide': 2,
    'OldTown': 2,
    'Edwards': 2,
    'Sawyer': 3,
    'Blueste': 3,
    'SWISU': 3,
    'NPkVill': 3,
    'NAmes': 3,
    'Mitchel': 4,
    'SawyerW': 5,
    'NWAmes': 5,
    'Gilbert': 5,
    'Blmngtn': 5,
    'CollgCr': 5,
    'ClearCr': 6,
    'Crawfor': 6,
    'Veenker': 7,
    'Somerst': 7,
    'Timber': 8,
    'StoneBr': 9,
    'NridgHt': 10,
    'NoRidge': 10
}

features['Neighborhood'] = features['Neighborhood'].map(neigh_map).astype(
    'int')
ext_map = {'Po': 1, 'Fa': 2, 'TA': 3, 'Gd': 4, 'Ex': 5}
features['ExterQual'] = features['ExterQual'].map(ext_map).astype('int')
features['ExterCond'] = features['ExterCond'].map(ext_map).astype('int')
bsm_map = {'None': 0, 'Po': 1, 'Fa': 2, 'TA': 3, 'Gd': 4, 'Ex': 5}
features['BsmtQual'] = features['BsmtQual'].map(bsm_map).astype('int')
features['BsmtCond'] = features['BsmtCond'].map(bsm_map).astype('int')
bsmf_map = {
    'None': 0,
    'Unf': 1,
    'LwQ': 2,
    'Rec': 3,
```

```
    'BLQ': 4,
    'ALQ': 5,
    'GLQ': 6
}

features['BsmtFinType1'] = features['BsmtFinType1'].map(bsmf_map).astype('int')
features['BsmtFinType2'] = features['BsmtFinType2'].map(bsmf_map).astype('int')
heat_map = {'Po': 1, 'Fa': 2, 'TA': 3, 'Gd': 4, 'Ex': 5}
features['HeatingQC'] = features['HeatingQC'].map(heat_map).astype('int')
features['KitchenQual'] = features['KitchenQual'].map(heat_map).astype('int')
features['FireplaceQu'] = features['FireplaceQu'].map(bsm_map).astype('int')
features['GarageCond'] = features['GarageCond'].map(bsm_map).astype('int')
features['GarageQual'] = features['GarageQual'].map(bsm_map).astype('int')
```

6 Numeric Data

There are many numeric features the inspect, one of the best ways to see how they effect sale prices is scatter plots. We're also plotting polynomial regression lines to see general trend. With this way we can understand the numerical values and their importance on sale price, also it's really helpful to spot outliers.

Observations:

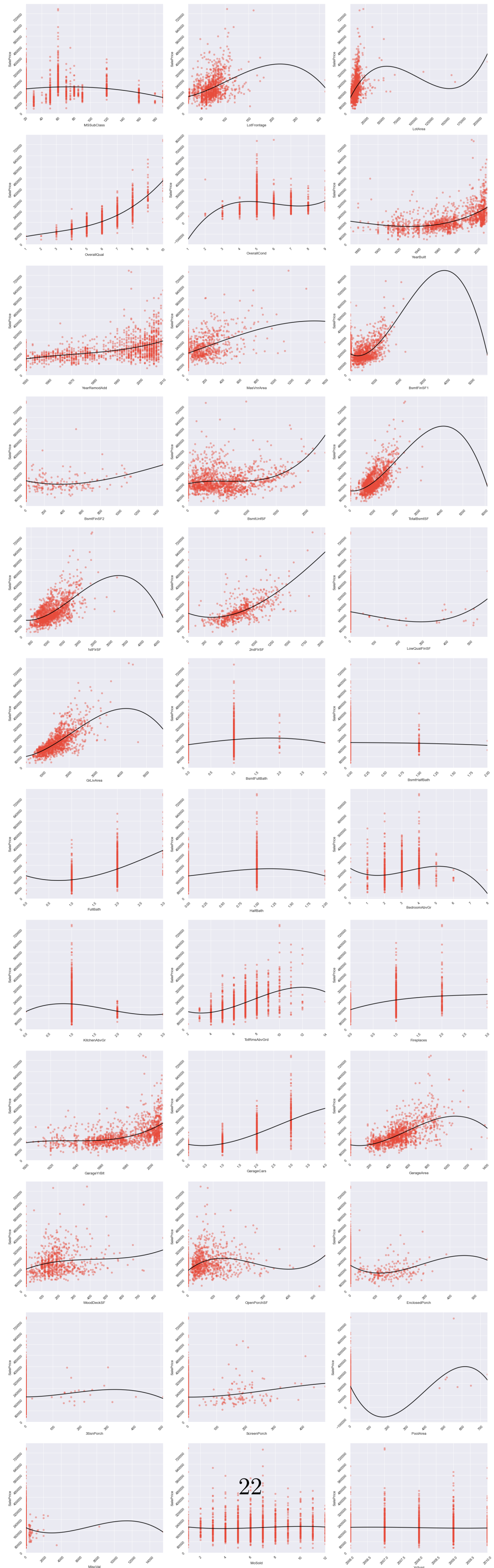
- **OverallQual;** It's clearly visible that sale price of the house increases with overall quality. This confirms the correlation in first table we did at the beginning. (Pearson corr was 0.8)
- **OverallCondition;** Looks like overall condition is left skewed where most of the houses are around 5/10 condition. But it doesn't effect the price like quality indicator…
- **YearBuilt;** Again new buildings are generally expensive than the old ones.
- **Basement;** General table shows bigger basements are increasing the price but I see some outliers there…
- **GrLivArea;** This feature is pretty linear but we can spot two outliers effecting this trend. There are some huge area houses with pretty cheap prices, there might be some reason behind it but we better drop them.
- **SaleDates;** They seem pretty unimportant on sale prices, we can drop them…

[32]:
```
# Plotting numerical features with polynomial order to detect outliers by eye.

def srt_reg(y, df):
    fig, axes = plt.subplots(12, 3, figsize=(25, 80))
    axes = axes.flatten()

    for i, j in zip(df.select_dtypes(include=['number']).columns, axes):

        sns.regplot(x=i,
                    y=y,
```

```
                  data=df,
                  ax=j,
                  order=3,
                  ci=None,
                  color='#e74c3c',
                  line_kws={'color': 'black'},
                  scatter_kws={'alpha':0.4})
    j.tick_params(labelrotation=45)
    j.yaxis.set_major_locator(MaxNLocator(nbins=10))

    plt.tight_layout()
```

[33]:

```
srt_reg('SalePrice', train)
```



6.1 Outliers

Ok here we're going to drop some outliers we detected them just above, this part is kinda subjective and you can try different approaches or you can implement some automatic outlier detection methods like isolation forests.

[34]:
```
# Dropping outliers after detecting them by eye.

features = features.join(y)
features = features.drop(features[(features['OverallQual'] < 5)
                                  & (features['SalePrice'] > 200000)].index)
features = features.drop(features[(features['GrLivArea'] > 4000)
                                  & (features['SalePrice'] < 200000)].index)
features = features.drop(features[(features['GarageArea'] > 1200)
                                  & (features['SalePrice'] < 200000)].index)
features = features.drop(features[(features['TotalBsmtSF'] > 3000)
                                  & (features['SalePrice'] > 320000)].index)
features = features.drop(features[(features['1stFlrSF'] < 3000)
                                  & (features['SalePrice'] > 600000)].index)
features = features.drop(features[(features['1stFlrSF'] > 3000)
                                  & (features['SalePrice'] < 200000)].index)

y = features['SalePrice']
y.dropna(inplace=True)
features.drop(columns='SalePrice', inplace=True)
```

6.2 Creating New Features

Ok in this part we going to create some features, these can improve our modelling. I went with basic approach by merging some important indicators and making them stronger.

[35]:
```
# Creating new features  based on previous observations. There might be some␣
↪highly correlated features now. You cab drop them if you want to...

features['TotalSF'] = (features['BsmtFinSF1'] + features['BsmtFinSF2'] +
                       features['1stFlrSF'] + features['2ndFlrSF'])
features['TotalBathrooms'] = (features['FullBath'] +
                              (0.5 * features['HalfBath']) +
                              features['BsmtFullBath'] +
                              (0.5 * features['BsmtHalfBath']))

features['TotalPorchSF'] = (features['OpenPorchSF'] + features['3SsnPorch'] +
                            features['EnclosedPorch'] +
                            features['ScreenPorch'] + features['WoodDeckSF'])
```

```
features['YearBlRm'] = (features['YearBuilt'] + features['YearRemodAdd'])

# Merging quality and conditions.

features['TotalExtQual'] = (features['ExterQual'] + features['ExterCond'])
features['TotalBsmQual'] = (features['BsmtQual'] + features['BsmtCond'] +
                            features['BsmtFinType1'] +
                            features['BsmtFinType2'])
features['TotalGrgQual'] = (features['GarageQual'] + features['GarageCond'])
features['TotalQual'] = features['OverallQual'] + features[
    'TotalExtQual'] + features['TotalBsmQual'] + features[
        'TotalGrgQual'] + features['KitchenQual'] + features['HeatingQC']

# Creating new features by using new quality indicators.

features['QualGr'] = features['TotalQual'] * features['GrLivArea']
features['QualBsm'] = features['TotalBsmQual'] * (features['BsmtFinSF1'] +
                                                  features['BsmtFinSF2'])
features['QualPorch'] = features['TotalExtQual'] * features['TotalPorchSF']
features['QualExt'] = features['TotalExtQual'] * features['MasVnrArea']
features['QualGrg'] = features['TotalGrgQual'] * features['GarageArea']
features['QlLivArea'] = (features['GrLivArea'] -
                         features['LowQualFinSF']) * (features['TotalQual'])
features['QualSFNg'] = features['QualGr'] * features['Neighborhood']
```

[36]:

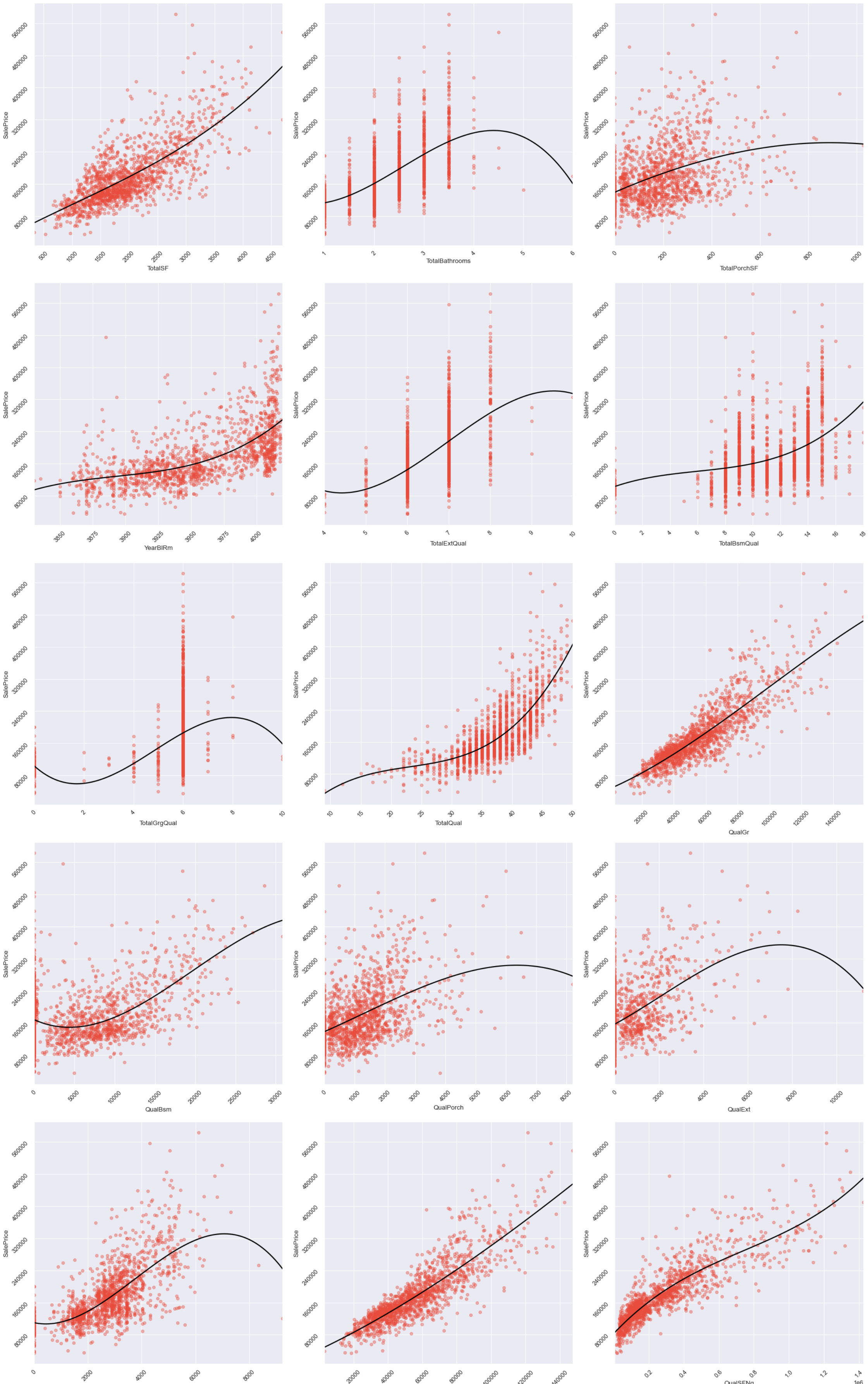
```
# Observing the effects of newly created features on sale price.

def srt_reg(feature):
    merged = features.join(y)
    fig, axes = plt.subplots(5, 3, figsize=(25, 40))
    axes = axes.flatten()

    new_features = [
        'TotalSF', 'TotalBathrooms', 'TotalPorchSF', 'YearBlRm',
        'TotalExtQual', 'TotalBsmQual', 'TotalGrgQual', 'TotalQual', 'QualGr',
        'QualBsm', 'QualPorch', 'QualExt', 'QualGrg', 'QlLivArea', 'QualSFNg'
    ]

    for i, j in zip(new_features, axes):

        sns.regplot(x=i,
                    y=feature,
                    data=merged,
                    ax=j,
                    order=3,
                    ci=None,
                    color='#e74c3c',
```

```
                    line_kws={'color': 'black'},
                    scatter_kws={'alpha':0.4})
        j.tick_params(labelrotation=45)
        j.yaxis.set_major_locator(MaxNLocator(nbins=10))

        plt.tight_layout()
```

6.3 Checking New Features

Well… They look decent enough, I hope these can help us building strong models. I also wanted to add some more basic features for having specific feature or not. This approach was widely accepted by community so I see no harm to add them.

[37]:
```
srt_reg('SalePrice')
```



SalePrice
TotalSF
TotalBathrooms
TotalPorchSF
YearBlRm
TotalExtQual
TotalBsmQual
TotalGrgQual
TotalQual
QualGr
QualBsm
QualPorch
QualExt
QualGrg
GrLivArea
QualSFNg
1e6

[38]:
```
# Creating some simple features.

features['HasPool'] = features['PoolArea'].apply(lambda x: 1 if x > 0 else 0)
features['Has2ndFloor'] = features['2ndFlrSF'].apply(lambda x: 1
                                                     if x > 0 else 0)
features['HasGarage'] = features['QualGrg'].apply(lambda x: 1 if x > 0 else 0)
features['HasBsmt'] = features['QualBsm'].apply(lambda x: 1 if x > 0 else 0)
features['HasFireplace'] = features['Fireplaces'].apply(lambda x: 1
                                                        if x > 0 else 0)
features['HasPorch'] = features['QualPorch'].apply(lambda x: 1 if x > 0 else 0)
```

6.4 Transforming the Data

Some of the continious values are not distributed evenly and not fitting on normal distribution, we can fix them by using couple transformation approaches. We're going to use boxcox here, again it's widely used by community and I want to thank them all for their great work.

We're going to list skewed features and then apply boxcox transformation with boxcox_normmax (It computes optimal boxcox transform parameter for input data, so we don't decide the lambda here)...

[39]:
```
# Numerical features we worked on which seems highly skewed but we filter again␣
↪anyways...

skewed = [
    'LotFrontage', 'LotArea', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2',
    'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GrLivArea',
    'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
    'ScreenPorch', 'PoolArea', 'LowQualFinSF', 'MiscVal'
]
```

[40]:
```
# Finding skewness of the numerical features.

skew_features = np.abs(features[skewed].apply(lambda x: skew(x)).sort_values(
    ascending=False))

# Filtering skewed features.

high_skew = skew_features[skew_features > 0.3]

# Taking indexes of high skew.

skew_index = high_skew.index

# Applying boxcox transformation to fix skewness.
```

```
for i in skew_index:
    features[i] = boxcox1p(features[i], boxcox_normmax(features[i] + 1))
```

Here we dropping some unnecessary features had their use in feature engineering or not needed at all. Obviously it's subjective but I feel they don't add much to model. Then we one hot encode the categorical data left so everything will be prepared for the modelling.

[41]:
```
# Features to drop:

to_drop = [
    'Utilities',
    'PoolQC',
    'YrSold',
    'MoSold',
    'ExterQual',
    'BsmtQual',
    'GarageQual',
    'KitchenQual',
    'HeatingQC',
]

# Dropping features.

features.drop(columns=to_drop, inplace=True)
```

[42]:
```
# Getting dummy variables for categorical data.

features = pd.get_dummies(data=features)
```

7 Double Check

- **Before we move to modelling I want to take one last look to the data we processed. Everyting seems in order, not missing datas, values are numerical etc. Our feature engineered data is present...**
- **Just want to check how transformed data correlates with sale prices before we move on and it looks decent.**
- **Again I wanted to check our target value distribution and it seems little skewed. We can fix this by applying log transformation so our models can perform better.**

[43]:
```
print(f'Number of missing values: {features.isna().sum().sum()}')
```

```
Number of missing values: 0
```

[44]:
```
features.shape
```

[44]: (2908, 226)

[45]:
```python
features.sample(5)
```

[45]:

	LotFrontage	LotArea	Neighborhood	OverallQual	OverallCond
2570	19.280169	15.877272	6	4	5
988	20.976651	14.924693	5	6	6
2706	19.626928	14.147986	3	5	6
920	19.280169	14.100684	5	6	5
1778	20.139852	14.375769	3	5	5

	YearBuilt	YearRemodAdd	MasVnrArea	ExterCond	BsmtCond	BsmtFinType1
2570	1995	1996	0.000000	3	3	1
988	1976	1976	24.084299	3	3	2
2706	1963	1963	0.000000	3	3	5
920	1994	1994	14.677027	4	4	6
1778	1953	1953	16.867723	3	0	0

	BsmtFinSF1	BsmtFinType2	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	1stFlrSF
2570	0.000000	1	0.0	127.411257	842.563625	6.836663
988	49.603648	1	0.0	75.417247	457.686867	6.491067
2706	55.245341	1	0.0	78.731747	494.618038	6.329207
920	162.815783	1	0.0	25.425830	496.050012	6.339323
1778	0.000000	0	0.0	0.000000	0.000000	6.558414

	2ndFlrSF	LowQualFinSF	GrLivArea	BsmtFullBath	BsmtHalfBath
2570	0.000000	0.0	9.454507	0.0	0.0
988	1259.287550	0.0	9.757328	0.0	0.0
2706	0.000000	0.0	8.522975	1.0	0.0
920	1075.059903	0.0	9.492889	0.0	1.0
1778	0.000000	0.0	8.937875	0.0	0.0

	FullBath	HalfBath	BedroomAbvGr	KitchenAbvGr	TotRmsAbvGrd
2570	2	0	4	1	7
988	2	1	4	1	8
2706	1	0	2	1	5
920	2	1	3	1	7
1778	1	1	2	1	7

	Fireplaces	FireplaceQu	GarageYrBlt	GarageCars	GarageArea
2570	0	0	1996.0	2.0	628.0
988	1	3	1976.0	2.0	551.0
2706	0	0	1990.0	2.0	484.0
920	0	0	1994.0	2.0	471.0
1778	0	0	1953.0	1.0	616.0

GarageCond WoodDeckSF OpenPorchSF EnclosedPorch 3SsnPorch \

2570	3	36.133149	0.000000	0.000000	0.0
988	3	0.000000	23.289458	0.000000	0.0
2706	3	43.624701	13.323536	0.000000	0.0
920	3	56.173947	14.485271	0.000000	0.0
1778	3	44.319388	0.000000	10.749179	0.0

	ScreenPorch	PoolArea	MiscVal	TotalSF	TotalBathrooms	TotalPorchSF \
2570	0.0	0.0	0.0	1680.0	2.0	152
988	0.0	0.0	0.0	2186.0	2.5	224
2706	0.0	0.0	0.0	1106.0	2.0	277
920	0.0	0.0	0.0	2535.0	3.0	387
1778	0.0	0.0	0.0	1210.0	1.5	308

	YearBlRm	TotalExtQual	TotalBsmQual	TotalGrgQual	TotalQual	QualGr \
2570	3991	6	9	6	33	55440
988	3952	6	9	6	34	69020
2706	3926	6	12	6	36	33300
920	3988	8	15	6	44	75724
1778	3906	6	0	6	23	27830

	QualBsm	QualPorch	QualExt	QualGrg	QlLivArea	QualSFNg	HasPool \
2570	0.0	912	0.0	3768.0	55440	332640	0
988	1404.0	1344	1788.0	3306.0	69020	345100	0
2706	2172.0	1662	0.0	2904.0	33300	99900	0
920	12210.0	3096	840.0	2826.0	75724	378620	0
1778	0.0	1848	840.0	3696.0	27830	83490	0

	Has2ndFloor	HasGarage	HasBsmt	HasFireplace	HasPorch	MSSubClass_120 \
2570	0	1	0	0	1	0
988	1	1	1	1	1	0
2706	0	1	1	0	1	0
920	1	1	1	0	1	0
1778	0	1	0	0	1	0

	MSSubClass_150	MSSubClass_160	MSSubClass_180	MSSubClass_190 \
2570	0	0	0	0
988	0	0	0	0
2706	0	0	0	0
920	0	0	0	0
1778	0	0	0	0

	MSSubClass_20	MSSubClass_30	MSSubClass_40	MSSubClass_45 \
2570	1	0	0	0
988	0	0	0	0
2706	1	0	0	0
920	0	0	0	0
1778	1	0	0	0

	MSSubClass_50	MSSubClass_60	MSSubClass_70	MSSubClass_75	\
2570	0	0	0	0	
988	0	1	0	0	
2706	0	0	0	0	
920	0	1	0	0	
1778	0	0	0	0	

	MSSubClass_80	MSSubClass_85	MSSubClass_90	MSZoning_C (all)	\
2570	0	0	0	0	
988	0	0	0	0	
2706	0	0	0	0	
920	0	0	0	0	
1778	0	0	0	0	

	MSZoning_FV	MSZoning_RH	MSZoning_RL	MSZoning_RM	Street_Grvl	\
2570	0	0	1	0	0	
988	0	0	1	0	0	
2706	0	0	1	0	0	
920	0	0	1	0	0	
1778	0	0	1	0	0	

	Street_Pave	Alley_Grvl	Alley_None	Alley_Pave	LotShape_IR1	\
2570	1	0	1	0	0	
988	1	0	1	0	1	
2706	1	0	1	0	1	
920	1	0	1	0	1	
1778	1	0	1	0	0	

	LotShape_IR2	LotShape_IR3	LotShape_Reg	LandContour_Bnk	\
2570	1	0	0	0	
988	0	0	0	0	
2706	0	0	0	0	
920	0	0	0	0	
1778	0	0	1	0	

	LandContour_HLS	LandContour_Low	LandContour_Lvl	LotConfig_Corner	\
2570	0	0	1	0	
988	0	0	1	0	
2706	0	0	1	0	
920	0	0	1	0	
1778	0	0	1	0	

	LotConfig_CulDSac	LotConfig_FR2	LotConfig_FR3	LotConfig_Inside	\
2570	0	0	0	1	
988	0	0	0	1	
2706	0	0	0	1	

920	0	0	0	1
1778	0	0	0	1

	LandSlope_Gtl	LandSlope_Mod	LandSlope_Sev	Condition1_Artery \
2570	1	0	0	0
988	1	0	0	0
2706	1	0	0	0
920	1	0	0	0
1778	1	0	0	0

	Condition1_Feedr	Condition1_Norm	Condition1_Other	Condition1_PosA \
2570	0	1	0	0
988	0	1	0	0
2706	0	0	0	0
920	0	1	0	0
1778	0	1	0	0

	Condition1_PosN	Condition1_RRAe	Condition1_RRAn	Condition2_Feedr \
2570	0	0	0	0
988	0	0	0	0
2706	0	1	0	0
920	0	0	0	0
1778	0	0	0	0

	Condition2_Norm	Condition2_Other	BldgType_1Fam	BldgType_2fmCon \
2570	1	0	1	0
988	1	0	1	0
2706	1	0	1	0
920	1	0	1	0
1778	1	0	1	0

	BldgType_Duplex	BldgType_Twnhs	BldgType_TwnhsE	HouseStyle_1.5Fin \
2570	0	0	0	0
988	0	0	0	0
2706	0	0	0	0
920	0	0	0	0
1778	0	0	0	0

	HouseStyle_1.5Unf	HouseStyle_1Story	HouseStyle_2.5Fin \
2570	0	1	0
988	0	0	0
2706	0	1	0
920	0	0	0
1778	0	1	0

	HouseStyle_2.5Unf	HouseStyle_2Story	HouseStyle_SFoyer \
2570	0	0	0

988	0	1	0
2706	0	0	0
920	0	1	0
1778	0	0	0

	HouseStyle_SLvl	RoofStyle_Flat	RoofStyle_Gable	RoofStyle_Gambrel \
2570	0	0	1	0
988	0	0	1	0
2706	0	0	1	0
920	0	0	1	0
1778	0	0	0	0

	RoofStyle_Hip	RoofStyle_Mansard	RoofStyle_Shed	RoofMatl_CompShg \
2570	0	0	0	1
988	0	0	0	1
2706	0	0	0	1
920	0	0	0	1
1778	1	0	0	1

	RoofMatl_Other	RoofMatl_Tar&Grv	Exterior1st_AsbShng \
2570	0	0	0
988	0	0	0
2706	0	0	0
920	0	0	0
1778	0	0	0

	Exterior1st_BrkFace	Exterior1st_CemntBd	Exterior1st_HdBoard \
2570	0	0	0
988	0	0	0
2706	0	0	1
920	0	0	1
1778	0	0	0

	Exterior1st_MetalSd	Exterior1st_Other	Exterior1st_Plywood \
2570	0	0	0
988	0	0	1
2706	0	0	0
920	0	0	0
1778	0	0	0

	Exterior1st_Stucco	Exterior1st_VinylSd	Exterior1st_Wd Sdng \
2570	0	1	0
988	0	0	0
2706	0	0	0
920	0	0	0
1778	0	0	1

```
      Exterior1st_WdShing  Exterior2nd_AsbShng  Exterior2nd_Brk Cmn  \
2570                    0                    0                    0
988                     0                    0                    0
2706                    0                    0                    0
920                     0                    0                    0
1778                    0                    0                    0

      Exterior2nd_BrkFace  Exterior2nd_CmentBd  Exterior2nd_HdBoard  \
2570                    0                    0                    0
988                     0                    0                    0
2706                    0                    0                    1
920                     0                    0                    1
1778                    0                    0                    0

      Exterior2nd_ImStucc  Exterior2nd_MetalSd  Exterior2nd_Other  \
2570                    0                    0                  0
988                     0                    0                  0
2706                    0                    0                  0
920                     0                    0                  0
1778                    0                    0                  0

      Exterior2nd_Plywood  Exterior2nd_Stucco  Exterior2nd_VinylSd  \
2570                    0                   0                    1
988                     1                   0                    0
2706                    0                   0                    0
920                     0                   0                    0
1778                    0                   0                    0

      Exterior2nd_Wd Sdng  Exterior2nd_Wd Shng  MasVnrType_BrkCmn  \
2570                    0                    0                  0
988                     0                    0                  0
2706                    0                    0                  0
920                     0                    0                  0
1778                    1                    0                  0

      MasVnrType_BrkFace  MasVnrType_None  MasVnrType_Stone  \
2570                   0                1                 0
988                    1                0                 0
2706                   0                1                 0
920                    1                0                 0
1778                   1                0                 0

      Foundation_BrkTil  Foundation_CBlock  Foundation_PConc  Foundation_Slab  \
2570                  0                  0                 1                0
988                   0                  1                 0                0
2706                  0                  0                 1                0
920                   0                  0                 1                0
```

```
1778                    0                   0                   0                   1

      Foundation_Stone  Foundation_Wood  BsmtExposure_Av  BsmtExposure_Gd  \
2570                 0                0                0                0
988                  0                0                0                0
2706                 0                0                0                0
920                  0                0                0                0
1778                 0                0                0                0

      BsmtExposure_Mn  BsmtExposure_No  BsmtExposure_None  Heating_GasA  \
2570                0                1                  0             1
988                 0                1                  0             1
2706                0                1                  0             1
920                 0                1                  0             1
1778                0                0                  1             1

      Heating_GasW  Heating_Other  CentralAir_N  CentralAir_Y  \
2570             0              0             0             1
988              0              0             0             1
2706             0              0             0             1
920              0              0             0             1
1778             0              0             0             1

      Electrical_FuseA  Electrical_FuseF  Electrical_Other  Electrical_SBrkr  \
2570                 0                 0                 0                 1
988                  0                 0                 0                 1
2706                 0                 0                 0                 1
920                  0                 0                 0                 1
1778                 1                 0                 0                 0

      Functional_Maj1  Functional_Min1  Functional_Min2  Functional_Mod  \
2570                0                0                0               0
988                 0                0                0               0
2706                0                0                0               0
920                 0                0                0               0
1778                0                0                0               0

      Functional_Other  Functional_Typ  GarageType_2Types  GarageType_Attchd  \
2570                 0               1                  0                  1
988                  0               1                  0                  1
2706                 0               1                  0                  0
920                  0               1                  0                  1
1778                 0               1                  0                  1

      GarageType_Basment  GarageType_BuiltIn  GarageType_CarPort  \
2570                   0                   0                   0
988                    0                   0                   0
```

```
2706                  0                   0                   0
920                   0                   0                   0
1778                  0                   0                   0

      GarageType_Detchd  GarageType_None  GarageFinish_Fin  GarageFinish_None  \
2570                  0                0                 0                  0
988                   0                0                 1                  0
2706                  1                0                 0                  0
920                   0                0                 0                  0
1778                  0                0                 1                  0

      GarageFinish_RFn  GarageFinish_Unf  PavedDrive_N  PavedDrive_P  \
2570                 0                 1             0             0
988                  0                 0             0             0
2706                 0                 1             0             0
920                  1                 0             0             0
1778                 0                 0             0             0

      PavedDrive_Y  Fence_GdPrv  Fence_GdWo  Fence_MnPrv  Fence_MnWw  \
2570             1            0           0            0           0
988              1            0           0            0           0
2706             1            0           0            0           0
920              1            0           0            0           0
1778             1            0           0            1           0

      Fence_None  MiscFeature_Gar2  MiscFeature_None  MiscFeature_Othr  \
2570           1                 0                 1                 0
988            1                 0                 1                 0
2706           1                 0                 1                 0
920            1                 0                 1                 0
1778           0                 0                 1                 0

      MiscFeature_Shed  MiscFeature_TenC  SaleType_COD  SaleType_CWD  \
2570                 0                 0             0             0
988                  0                 0             0             0
2706                 0                 0             0             0
920                  0                 0             0             0
1778                 0                 0             0             0

      SaleType_ConLD  SaleType_New  SaleType_Other  SaleType_WD  \
2570               0             0               0            1
988                0             0               0            1
2706               0             0               0            1
920                0             0               0            1
1778               1             0               0            0

      SaleCondition_Abnorml  SaleCondition_AdjLand  SaleCondition_Alloca  \
```

2570	0	0	0
988	0	0	0
2706	0	0	0
920	0	0	0
1778	0	0	0

	SaleCondition_Family	SaleCondition_Normal	SaleCondition_Partial
2570	0	1	0
988	0	1	0
2706	0	1	0
920	0	1	0
1778	0	1	0

[46]:
```
features.describe()
```

[46]:

	LotFrontage	LotArea	Neighborhood	OverallQual	OverallCond
count	2908.000000	2908.000000	2908.000000	2908.000000	2908.000000
mean	18.932143	14.236978	4.455640	6.081843	5.566713
std	3.698922	1.155737	2.457431	1.397639	1.114074
min	8.809473	10.151044	1.000000	1.000000	1.000000
25%	17.482488	13.809916	3.000000	5.000000	5.000000
50%	19.280169	14.351060	4.000000	6.000000	5.000000
75%	20.976651	14.815543	5.000000	7.000000	6.000000
max	48.749456	22.753416	10.000000	10.000000	9.000000

	YearBuilt	YearRemodAdd	MasVnrArea	ExterCond	BsmtCond
count	2908.000000	2908.000000	2908.000000	2908.000000	2908.000000
mean	1971.252751	1984.227992	8.183116	3.085282	2.918157
std	30.296319	20.899483	11.176757	0.372262	0.576014
min	1872.000000	1950.000000	0.000000	1.000000	0.000000
25%	1953.000000	1965.000000	0.000000	3.000000	3.000000
50%	1973.000000	1993.000000	0.000000	3.000000	3.000000
75%	2001.000000	2004.000000	18.086738	3.000000	3.000000
max	2010.000000	2010.000000	51.503144	5.000000	4.000000

	BsmtFinType1	BsmtFinSF1	BsmtFinType2	BsmtFinSF2	BsmtUnfSF
count	2908.000000	2908.000000	2908.000000	2908.000000	2908.000000
mean	3.535420	89.404294	1.273384	1.158782	60.314321
std	2.113347	79.132825	0.954352	3.241698	32.850280
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	0.000000	1.000000	0.000000	37.968111
50%	4.000000	92.161729	1.000000	0.000000	59.674861
75%	6.000000	150.796231	1.000000	0.000000	82.392450
max	6.000000	507.637657	6.000000	14.352603	154.678880

	TotalBsmtSF	1stFlrSF	2ndFlrSF	LowQualFinSF	GrLivArea
count	2908.000000	2908.000000	2908.000000	2908.000000	2908.000000

```
mean    546.849878     6.474137   460.160888      0.062741     9.193287
std     201.641654     0.272501   590.189425      0.534211     0.505437
min       0.000000     5.447959     0.000000      0.000000     7.012554
25%     430.479452     6.282596     0.000000      0.000000     8.824658
50%     524.586689     6.462370     0.000000      0.000000     9.212684
75%     669.403427     6.671500   958.401066      0.000000     9.510844
max    2267.208842     7.762519  2673.776640      5.405398    11.282379

       BsmtFullBath  BsmtHalfBath     FullBath     HalfBath  BedroomAbvGr  \
count   2908.000000   2908.000000  2908.000000  2908.000000   2908.000000
mean       0.427785      0.061210     1.565681     0.379642      2.859697
std        0.523883      0.245429     0.550340     0.502787      0.822532
min        0.000000      0.000000     0.000000     0.000000      0.000000
25%        0.000000      0.000000     1.000000     0.000000      2.000000
50%        0.000000      0.000000     2.000000     0.000000      3.000000
75%        1.000000      0.000000     2.000000     1.000000      3.000000
max        3.000000      2.000000     4.000000     2.000000      8.000000

       KitchenAbvGr  TotRmsAbvGrd   Fireplaces  FireplaceQu  GarageYrBlt  \
count   2908.000000   2908.000000  2908.000000  2908.000000  2908.000000
mean       1.044704      6.440509     0.593535     1.761692  1869.897180
std        0.214850      1.557377     0.642910     1.805455   450.471016
min        0.000000      2.000000     0.000000     0.000000     0.000000
25%        1.000000      5.000000     0.000000     0.000000  1957.000000
50%        1.000000      6.000000     1.000000     1.000000  1977.000000
75%        1.000000      7.000000     1.000000     4.000000  2001.000000
max        3.000000     15.000000     4.000000     5.000000  2207.000000

        GarageCars   GarageArea   GarageCond   WoodDeckSF  OpenPorchSF  \
count  2908.000000  2908.000000  2908.000000  2908.000000  2908.000000
mean      1.762036   471.087689     2.808116    19.577941     7.369863
std       0.760056   213.558615     0.713747    23.031009     7.657100
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       1.000000   319.750000     3.000000     0.000000     0.000000
50%       2.000000   478.000000     3.000000     0.000000     7.585893
75%       2.000000   576.000000     3.000000    38.571245    12.944100
max       5.000000  1488.000000     5.000000   152.175869    41.498257

       EnclosedPorch    3SsnPorch  ScreenPorch     PoolArea      MiscVal  \
count    2908.000000  2908.000000  2908.000000  2908.000000  2908.000000
mean        1.887984     0.080496     2.043435     0.019268     0.246631
std         4.523003     0.713728     6.736139     0.313451     1.308700
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     0.000000     0.000000     0.000000     0.000000
75%         0.000000     0.000000     0.000000     0.000000     0.000000
max        26.241996     7.826717    42.829027     5.476842    10.540601
```

	TotalSF	TotalBathrooms	TotalPorchSF	YearBlRm	TotalExtQual	\
count	2908.000000	2908.000000	2908.000000	2908.000000	2908.000000	
mean	1975.246561	2.213893	182.277166	3955.480743	6.479367	
std	720.979190	0.803749	159.503200	46.137162	0.695892	
min	334.000000	1.000000	0.000000	3830.000000	3.000000	
25%	1485.750000	1.500000	48.000000	3920.000000	6.000000	
50%	1843.000000	2.000000	164.000000	3954.000000	6.000000	
75%	2374.000000	2.500000	266.000000	4002.000000	7.000000	
max	9105.000000	7.000000	1424.000000	4020.000000	10.000000	

	TotalBsmQual	TotalGrgQual	TotalQual	QualGr	QualBsm	\
count	2908.000000	2908.000000	2908.000000	2908.000000	2908.000000	
mean	11.200825	5.608322	37.027166	56420.461486	6420.540234	
std	3.216140	1.411642	5.850825	23518.703543	6580.039000	
min	0.000000	0.000000	9.000000	3006.000000	0.000000	
25%	9.000000	6.000000	34.000000	38958.250000	0.000000	
50%	12.000000	6.000000	37.000000	52779.000000	5440.000000	
75%	14.000000	6.000000	41.000000	68484.750000	10335.500000	
max	19.000000	10.000000	53.000000	249655.000000	60150.000000	

	QualPorch	QualExt	QualGrg	QlLivArea	QualSFNg	\
count	2908.000000	2908.000000	2908.000000	2908.000000	2.908000e+03	
mean	1204.990028	682.805021	2810.068776	56269.768226	2.875559e+05	
std	1087.473080	1234.995030	1314.699819	23487.969208	2.603670e+05	
min	0.000000	0.000000	0.000000	3006.000000	6.012000e+03	
25%	308.000000	0.000000	1872.000000	38828.000000	1.038765e+05	
50%	1050.000000	0.000000	2862.000000	52696.500000	1.928385e+05	
75%	1760.000000	1045.500000	3456.000000	68270.250000	3.799160e+05	
max	9328.000000	11200.000000	9436.000000	249655.000000	1.750000e+06	

	HasPool	Has2ndFloor	HasGarage	HasBsmt	HasFireplace	\
count	2908.000000	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.003783	0.428473	0.945323	0.680536	0.512036	
std	0.061398	0.494943	0.227387	0.466349	0.499941	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	1.000000	0.000000	0.000000	
50%	0.000000	0.000000	1.000000	1.000000	1.000000	
75%	0.000000	1.000000	1.000000	1.000000	1.000000	
max	1.000000	1.000000	1.000000	1.000000	1.000000	

	HasPorch	MSSubClass_120	MSSubClass_150	MSSubClass_160	\
count	2908.00000	2908.000000	2908.000000	2908.000000	
mean	0.83425	0.062586	0.000344	0.044017	
std	0.37192	0.242258	0.018544	0.205167	
min	0.00000	0.000000	0.000000	0.000000	
25%	1.00000	0.000000	0.000000	0.000000	

```
50%        1.00000         0.000000         0.000000         0.000000
75%        1.00000         0.000000         0.000000         0.000000
max        1.00000         1.000000         1.000000         1.000000

       MSSubClass_180  MSSubClass_190  MSSubClass_20  MSSubClass_30  \
count     2908.000000     2908.000000    2908.000000    2908.000000
mean         0.005846        0.020633       0.369670       0.047455
std          0.076248        0.142176       0.482798       0.212647
min          0.000000        0.000000       0.000000       0.000000
25%          0.000000        0.000000       0.000000       0.000000
50%          0.000000        0.000000       0.000000       0.000000
75%          0.000000        0.000000       1.000000       0.000000
max          1.000000        1.000000       1.000000       1.000000

       MSSubClass_40  MSSubClass_45  MSSubClass_50  MSSubClass_60  \
count    2908.000000    2908.000000    2908.000000    2908.000000
mean        0.002063       0.006190       0.098693       0.196011
std         0.045384       0.078445       0.298301       0.397045
min         0.000000       0.000000       0.000000       0.000000
25%         0.000000       0.000000       0.000000       0.000000
50%         0.000000       0.000000       0.000000       0.000000
75%         0.000000       0.000000       0.000000       0.000000
max         1.000000       1.000000       1.000000       1.000000

       MSSubClass_70  MSSubClass_75  MSSubClass_80  MSSubClass_85  \
count    2908.000000    2908.000000    2908.000000    2908.000000
mean        0.044017       0.007909       0.040578       0.016506
std         0.205167       0.088597       0.197344       0.127434
min         0.000000       0.000000       0.000000       0.000000
25%         0.000000       0.000000       0.000000       0.000000
50%         0.000000       0.000000       0.000000       0.000000
75%         0.000000       0.000000       0.000000       0.000000
max         1.000000       1.000000       1.000000       1.000000

       MSSubClass_90  MSZoning_C (all)  MSZoning_FV  MSZoning_RH  MSZoning_RL  \
count    2908.000000       2908.000000  2908.000000  2908.000000  2908.000000
mean        0.037483          0.008253     0.047799     0.008941     0.776135
std         0.189974          0.090486     0.213378     0.094149     0.416904
min         0.000000          0.000000     0.000000     0.000000     0.000000
25%         0.000000          0.000000     0.000000     0.000000     1.000000
50%         0.000000          0.000000     0.000000     0.000000     1.000000
75%         0.000000          0.000000     0.000000     0.000000     1.000000
max         1.000000          1.000000     1.000000     1.000000     1.000000

       MSZoning_RM  Street_Grvl  Street_Pave   Alley_Grvl   Alley_None  \
count  2908.000000  2908.000000  2908.000000  2908.000000  2908.000000
mean      0.158872     0.003783     0.996217     0.041265     0.931912
```

std	0.365620	0.061398	0.061398	0.198938	0.251940
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	1.000000	0.000000	1.000000
50%	0.000000	0.000000	1.000000	0.000000	1.000000
75%	0.000000	0.000000	1.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000

	Alley_Pave	LotShape_IR1	LotShape_IR2	LotShape_IR3	LotShape_Reg \
count	2908.000000	2908.000000	2908.000000	2908.000000	2908.000000
mean	0.026823	0.330468	0.026135	0.005158	0.638239
std	0.161592	0.470462	0.159564	0.071647	0.480593
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	0.000000	0.000000	0.000000	1.000000
75%	0.000000	1.000000	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000

	LandContour_Bnk	LandContour_HLS	LandContour_Low	LandContour_Lvl \
count	2908.000000	2908.000000	2908.000000	2908.000000
mean	0.039546	0.041265	0.019601	0.899587
std	0.194924	0.198938	0.138649	0.300601
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	1.000000
50%	0.000000	0.000000	0.000000	1.000000
75%	0.000000	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000

	LotConfig_Corner	LotConfig_CulDSac	LotConfig_FR2	LotConfig_FR3 \
count	2908.000000	2908.000000	2908.000000	2908.000000
mean	0.174691	0.059835	0.029230	0.004814
std	0.379767	0.237222	0.168479	0.069230
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000

	LotConfig_Inside	LandSlope_Gtl	LandSlope_Mod	LandSlope_Sev \
count	2908.000000	2908.000000	2908.000000	2908.000000
mean	0.731431	0.952201	0.042297	0.005502
std	0.443292	0.213378	0.201301	0.073984
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	0.000000	0.000000
50%	1.000000	1.000000	0.000000	0.000000
75%	1.000000	1.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000

	Condition1_Artery	Condition1_Feedr	Condition1_Norm	Condition1_Other	\
count	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.031637	0.056052	0.860385	0.005158	
std	0.175061	0.230062	0.346647	0.071647	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	1.000000	0.000000	
50%	0.000000	0.000000	1.000000	0.000000	
75%	0.000000	0.000000	1.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

	Condition1_PosA	Condition1_PosN	Condition1_RRAe	Condition1_RRAn	\
count	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.006878	0.013067	0.009629	0.017194	
std	0.082660	0.113583	0.097669	0.130016	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

	Condition2_Feedr	Condition2_Norm	Condition2_Other	BldgType_1Fam	\
count	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.004470	0.990028	0.005502	0.830468	
std	0.066723	0.099380	0.073984	0.375286	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	1.000000	0.000000	1.000000	
50%	0.000000	1.000000	0.000000	1.000000	
75%	0.000000	1.000000	0.000000	1.000000	
max	1.000000	1.000000	1.000000	1.000000	

	BldgType_2fmCon	BldgType_Duplex	BldgType_Twnhs	BldgType_TwnhsE	\
count	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.020977	0.037483	0.033012	0.078061	
std	0.143331	0.189974	0.178700	0.268313	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

	HouseStyle_1.5Fin	HouseStyle_1.5Unf	HouseStyle_1Story	\
count	2908.000000	2908.000000	2908.000000	
mean	0.107978	0.006534	0.503783	
std	0.310406	0.080581	0.500072	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	1.000000	

75%	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000

	HouseStyle_2.5Fin	HouseStyle_2.5Unf	HouseStyle_2Story	\
count	2908.000000	2908.000000	2908.000000	
mean	0.002751	0.008253	0.298143	
std	0.052387	0.090486	0.457521	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	1.000000	
max	1.000000	1.000000	1.000000	

	HouseStyle_SFoyer	HouseStyle_SLvl	RoofStyle_Flat	RoofStyle_Gable	\
count	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.028542	0.044017	0.006534	0.793329	
std	0.166544	0.205167	0.080581	0.404987	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	1.000000	
50%	0.000000	0.000000	0.000000	1.000000	
75%	0.000000	0.000000	0.000000	1.000000	
max	1.000000	1.000000	1.000000	1.000000	

	RoofStyle_Gambrel	RoofStyle_Hip	RoofStyle_Mansard	RoofStyle_Shed	\
count	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.007565	0.187070	0.003783	0.001719	
std	0.086664	0.390035	0.061398	0.041437	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

	RoofMatl_CompShg	RoofMatl_Other	RoofMatl_Tar&Grv	\
count	2908.000000	2908.000000	2908.000000	
mean	0.986245	0.006190	0.007565	
std	0.116493	0.078445	0.086664	
min	0.000000	0.000000	0.000000	
25%	1.000000	0.000000	0.000000	
50%	1.000000	0.000000	0.000000	
75%	1.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	

	Exterior1st_AsbShng	Exterior1st_BrkFace	Exterior1st_CemntBd	\
count	2908.000000	2908.000000	2908.000000	
mean	0.015131	0.029574	0.042985	
std	0.122094	0.169437	0.202858	

min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000

	Exterior1st_HdBoard	Exterior1st_MetalSd	Exterior1st_Other \
count	2908.000000	2908.000000	2908.000000
mean	0.151307	0.154058	0.004470
std	0.358409	0.361066	0.066723
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000

	Exterior1st_Plywood	Exterior1st_Stucco	Exterior1st_VinylSd \
count	2908.000000	2908.000000	2908.000000
mean	0.075653	0.014443	0.352132
std	0.264488	0.119328	0.477717
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000

	Exterior1st_Wd Sdng	Exterior1st_WdShing	Exterior2nd_AsbShng \
count	2908.000000	2908.000000	2908.000000
mean	0.140990	0.019257	0.013067
std	0.348071	0.137451	0.113583
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000

	Exterior2nd_Brk Cmn	Exterior2nd_BrkFace	Exterior2nd_CmentBd \
count	2908.000000	2908.000000	2908.000000
mean	0.007565	0.015818	0.042985
std	0.086664	0.124794	0.202858
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000

	Exterior2nd_HdBoard	Exterior2nd_ImStucc	Exterior2nd_MetalSd \

count	2908.000000	2908.000000	2908.000000
mean	0.138927	0.004814	0.153026
std	0.345930	0.069230	0.360075
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000

	Exterior2nd_Other	Exterior2nd_Plywood	Exterior2nd_Stucco \
count	2908.000000	2908.000000	2908.000000
mean	0.004814	0.092503	0.015818
std	0.069230	0.289785	0.124794
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000

	Exterior2nd_VinylSd	Exterior2nd_Wd Sdng	Exterior2nd_Wd Shng \
count	2908.000000	2908.000000	2908.000000
mean	0.348349	0.134457	0.027854
std	0.476529	0.341201	0.164583
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	1.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000

	MasVnrType_BrkCmn	MasVnrType_BrkFace	MasVnrType_None \
count	2908.000000	2908.000000	2908.000000
mean	0.008597	0.301582	0.605915
std	0.092336	0.459024	0.488737
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	1.000000
75%	0.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000

	MasVnrType_Stone	Foundation_BrkTil	Foundation_CBlock \
count	2908.000000	2908.000000	2908.000000
mean	0.083906	0.106946	0.423659
std	0.277295	0.309098	0.494223
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	1.000000

```
max           1.000000            1.000000            1.000000

       Foundation_PConc  Foundation_Slab  Foundation_Stone  Foundation_Wood  \
count       2908.000000      2908.000000       2908.000000      2908.000000
mean           0.447043         0.016850          0.003783         0.001719
std            0.497273         0.128732          0.061398         0.041437
min            0.000000         0.000000          0.000000         0.000000
25%            0.000000         0.000000          0.000000         0.000000
50%            0.000000         0.000000          0.000000         0.000000
75%            1.000000         0.000000          0.000000         0.000000
max            1.000000         1.000000          1.000000         1.000000

       BsmtExposure_Av  BsmtExposure_Gd  BsmtExposure_Mn  BsmtExposure_No  \
count      2908.000000      2908.000000      2908.000000      2908.000000
mean          0.143054         0.092503         0.082187         0.654058
std           0.350188         0.289785         0.274697         0.475756
min           0.000000         0.000000         0.000000         0.000000
25%           0.000000         0.000000         0.000000         0.000000
50%           0.000000         0.000000         0.000000         1.000000
75%           0.000000         0.000000         0.000000         1.000000
max           1.000000         1.000000         1.000000         1.000000

       BsmtExposure_None  Heating_GasA  Heating_GasW  Heating_Other  \
count        2908.000000   2908.000000   2908.000000    2908.000000
mean            0.028198      0.984525      0.009285       0.006190
std             0.165567      0.123452      0.095925       0.078445
min             0.000000      0.000000      0.000000       0.000000
25%             0.000000      1.000000      0.000000       0.000000
50%             0.000000      1.000000      0.000000       0.000000
75%             0.000000      1.000000      0.000000       0.000000
max             1.000000      1.000000      1.000000       1.000000

       CentralAir_N  CentralAir_Y  Electrical_FuseA  Electrical_FuseF  \
count   2908.000000   2908.000000       2908.000000       2908.000000
mean       0.067400      0.932600          0.064649          0.017194
std        0.250757      0.250757          0.245948          0.130016
min        0.000000      0.000000          0.000000          0.000000
25%        0.000000      1.000000          0.000000          0.000000
50%        0.000000      1.000000          0.000000          0.000000
75%        0.000000      1.000000          0.000000          0.000000
max        1.000000      1.000000          1.000000          1.000000

       Electrical_Other  Electrical_SBrkr  Functional_Maj1  Functional_Min1  \
count       2908.000000       2908.000000      2908.000000      2908.000000
mean           0.003095          0.915062         0.006534         0.022008
std            0.055555          0.278838         0.080581         0.146735
min            0.000000          0.000000         0.000000         0.000000
```

25%	0.000000	1.000000	0.000000	0.000000
50%	0.000000	1.000000	0.000000	0.000000
75%	0.000000	1.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000

	Functional_Min2	Functional_Mod	Functional_Other	Functional_Typ \
count	2908.000000	2908.000000	2908.000000	2908.000000
mean	0.024072	0.012036	0.003783	0.931568
std	0.153298	0.109064	0.061398	0.252529
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	1.000000
50%	0.000000	0.000000	0.000000	1.000000
75%	0.000000	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000

	GarageType_2Types	GarageType_Attchd	GarageType_Basment \
count	2908.000000	2908.000000	2908.000000
mean	0.007565	0.589752	0.012380
std	0.086664	0.491963	0.110592
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	1.000000	0.000000
75%	0.000000	1.000000	0.000000
max	1.000000	1.000000	1.000000

	GarageType_BuiltIn	GarageType_CarPort	GarageType_Detchd \
count	2908.000000	2908.000000	2908.000000
mean	0.063618	0.005158	0.267538
std	0.244112	0.071647	0.442751
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000

	GarageType_None	GarageFinish_Fin	GarageFinish_None	GarageFinish_RFn \
count	2908.000000	2908.000000	2908.000000	2908.000000
mean	0.053989	0.244154	0.054677	0.278198
std	0.226035	0.429658	0.227387	0.448189
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000

	GarageFinish_Unf	PavedDrive_N	PavedDrive_P	PavedDrive_Y \
count	2908.000000	2908.000000	2908.000000	2908.000000

mean	0.422971	0.074278	0.021320	0.904402
std	0.494116	0.262268	0.144475	0.294090
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	1.000000
50%	0.000000	0.000000	0.000000	1.000000
75%	1.000000	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000

	Fence_GdPrv	Fence_GdWo	Fence_MnPrv	Fence_MnWw	Fence_None	\
count	2908.000000	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.040578	0.038514	0.112792	0.004127	0.803989	
std	0.197344	0.192468	0.316393	0.064117	0.397045	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	1.000000	
50%	0.000000	0.000000	0.000000	0.000000	1.000000	
75%	0.000000	0.000000	0.000000	0.000000	1.000000	
max	1.000000	1.000000	1.000000	1.000000	1.000000	

	MiscFeature_Gar2	MiscFeature_None	MiscFeature_Othr	MiscFeature_Shed	\
count	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.001719	0.964237	0.001376	0.032325	
std	0.041437	0.185732	0.037069	0.176891	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	1.000000	0.000000	0.000000	
50%	0.000000	1.000000	0.000000	0.000000	
75%	0.000000	1.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

	MiscFeature_TenC	SaleType_COD	SaleType_CWD	SaleType_ConLD	\
count	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.000344	0.029917	0.004127	0.008597	
std	0.018544	0.170389	0.064117	0.092336	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

	SaleType_New	SaleType_Other	SaleType_WD	SaleCondition_Abnorml	\
count	2908.000000	2908.000000	2908.000000	2908.000000	
mean	0.081155	0.009972	0.866231	0.064993	
std	0.273121	0.099380	0.340462	0.246556	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	1.000000	0.000000	
50%	0.000000	0.000000	1.000000	0.000000	
75%	0.000000	0.000000	1.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

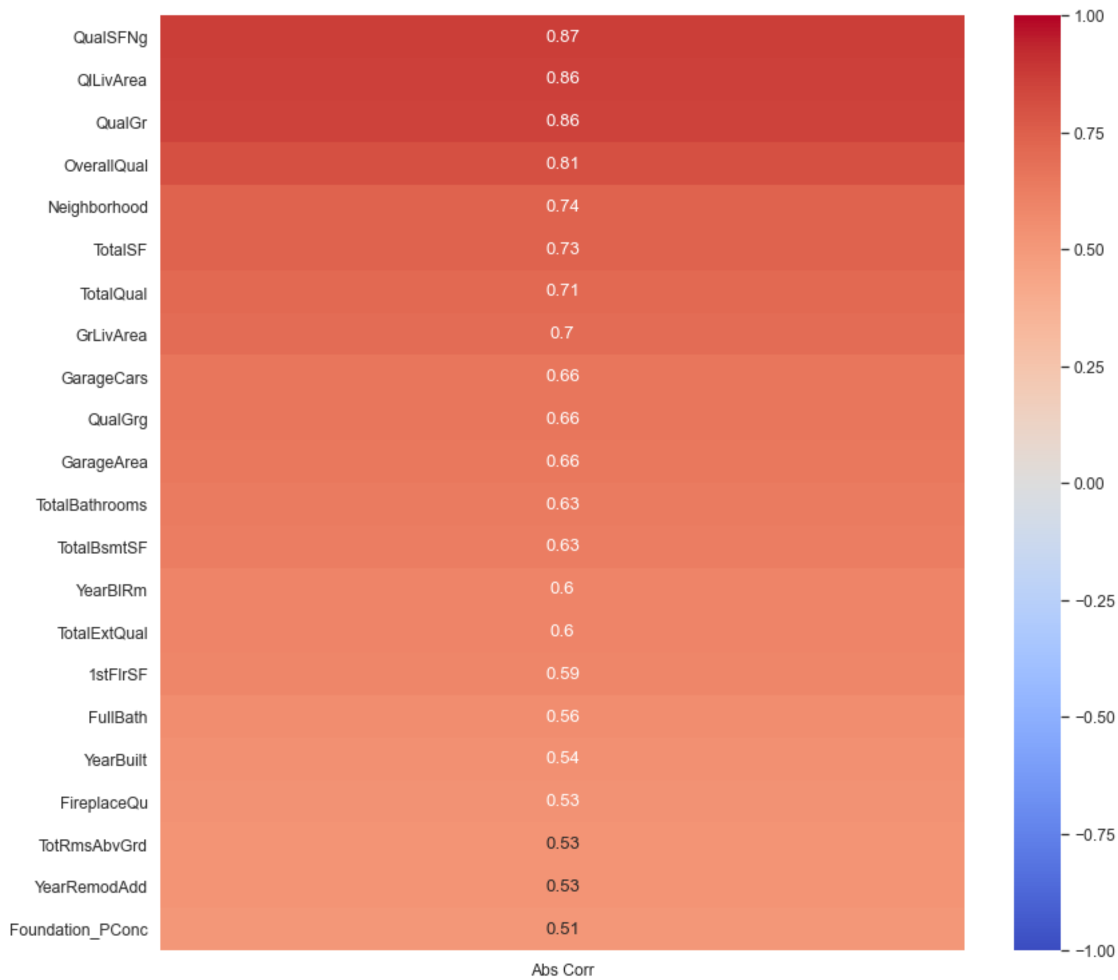
```
       SaleCondition_AdjLand  SaleCondition_Alloca  SaleCondition_Family  \
count            2908.000000           2908.000000           2908.000000
mean                0.004127              0.008253              0.015818
std                 0.064117              0.090486              0.124794
min                 0.000000              0.000000              0.000000
25%                 0.000000              0.000000              0.000000
50%                 0.000000              0.000000              0.000000
75%                 0.000000              0.000000              0.000000
max                 1.000000              1.000000              1.000000

       SaleCondition_Normal  SaleCondition_Partial
count           2908.000000            2908.000000
mean               0.823590               0.083219
std                0.381234               0.276260
min                0.000000               0.000000
25%                1.000000               0.000000
50%                1.000000               0.000000
75%                1.000000               0.000000
max                1.000000               1.000000
```

[47]:

```
# Separating train and test set.

train = features.iloc[:len(y), :]
test = features.iloc[len(train):, :]
```

[48]:

```
correlations = train.join(y).corrwith(train.join(y)['SalePrice']).iloc[:-1].
 ↪to_frame()
correlations['Abs Corr'] = correlations[0].abs()
sorted_correlations = correlations.sort_values('Abs Corr',␣
 ↪ascending=False)['Abs Corr']
fig, ax = plt.subplots(figsize=(12,12))
sns.heatmap(sorted_correlations.to_frame()[sorted_correlations>=.5],␣
 ↪cmap='coolwarm', annot=True, vmin=-1, vmax=1, ax=ax);
```



[49]:
```
def plot_dist3(df, feature, title):

    # Creating a customized chart. and giving in figsize and everything.

    fig = plt.figure(constrained_layout=True, figsize=(12, 8))

    # creating a grid of 3 cols and 3 rows.

    grid = gridspec.GridSpec(ncols=3, nrows=3, figure=fig)

    # Customizing the histogram grid.

    ax1 = fig.add_subplot(grid[0, :2])

    # Set the title.
```

```
    ax1.set_title('Histogram')

    # plot the histogram.

    sns.distplot(df.loc[:, feature],
                 hist=True,
                 kde=True,
                 fit=norm,
                 ax=ax1,
                 color='#e74c3c')
    ax1.legend(labels=['Normal', 'Actual'])

    # customizing the QQ_plot.

    ax2 = fig.add_subplot(grid[1, :2])

    # Set the title.

    ax2.set_title('Probability Plot')

    # Plotting the QQ_Plot.
    stats.probplot(df.loc[:, feature].fillna(np.mean(df.loc[:, feature])),
                   plot=ax2)
    ax2.get_lines()[0].set_markerfacecolor('#e74c3c')
    ax2.get_lines()[0].set_markersize(12.0)

    # Customizing the Box Plot:

    ax3 = fig.add_subplot(grid[:, 2])
    # Set title.

    ax3.set_title('Box Plot')

    # Plotting the box plot.

    sns.boxplot(df.loc[:, feature], orient='v', ax=ax3, color='#e74c3c')
    ax3.yaxis.set_major_locator(MaxNLocator(nbins=24))

    plt.suptitle(f'{title}', fontsize=24)
```

[50]:
```
# Checking target variable.

plot_dist3(train.join(y), 'SalePrice', 'Sale Price Before Log Transformation')
```


```
[51]: # Setting model data.

      X = train
      X_test = test
      y = np.log1p(y)
```

```
[52]: plot_dist3(train.join(y), 'SalePrice', 'Sale Price After Log Transformation')
```

8 Modelling

Well then, it's time to do some modelling! First of all I wanted to thank kaggle community for loads of examples inspired me. Especially Alex Lekov's great script and Serigne's stacked regressions approach were great guides for me!

Let's start with loading packages needed and then we set our regressors. The regressors I'm going to use here are:

- Ridge,
- Lasso,
- Elasticnet,
- Support Vector Regression
- I'm going to apply robust scaler on these before we run them because they really get effected by outliers.
- Gradient Boosting Regressor
- LightGBM Regressor
- XGBoost Regressor
- These don't need scaling in my opinion so we just go as it is
- Hist Gradient Boosting Regressor
- This is just for experimenting, it's still experimental on sklearn anyways
- Tweedie Regressor

- This regressor added in latest version of sklearn and I wanted to try it. It's generalized linear model with a Tweedie distribution. We gonna use power of 0 because we expecting normal target distribution but you can try this or other generalized models like poisson regressor or gamma regressor.

I tried to tune models by using Optuna package, that part is not added here.

[61]:
```
# Loading neccesary packages for modelling.

from sklearn.model_selection import cross_val_score, KFold, cross_validate
from sklearn.preprocessing import RobustScaler
from sklearn.linear_model import ElasticNetCV, LassoCV, RidgeCV,␣
↪TweedieRegressor
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import GradientBoostingRegressor,␣
↪HistGradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.pipeline import make_pipeline
from sklearn.metrics import mean_squared_error
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
from mlxtend.regressor import StackingCVRegressor # This is for stacking part,␣
↪works well with sklearn and others...
```

[64]:
```
# Setting kfold for future use.

kf = KFold(10)
```

[65]:
```
# Some parameters for ridge, lasso and elasticnet.

alphas_alt = [15.5, 15.6, 15.7, 15.8, 15.9, 15, 15.1, 15.2, 15.3, 15.4, 15.5]
alphas2 = [
    5e-05, 0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008
]
e_alphas = [
    0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007
]
e_l1ratio = [0.8, 0.85, 0.9, 0.95, 0.99, 1]

# ridge_cv

ridge = make_pipeline(RobustScaler(), RidgeCV(
    alphas=alphas_alt,
    cv=kf,
))

# lasso_cv:
```

```
lasso = make_pipeline(
    RobustScaler(),
    LassoCV(max_iter=1e7, alphas=alphas2, random_state=42, cv=kf))

# elasticnet_cv:

elasticnet = make_pipeline(
    RobustScaler(),
    ElasticNetCV(max_iter=1e7,
                 alphas=e_alphas,
                 cv=kf,
                 random_state=42,
                 l1_ratio=e_l1ratio))

# svr:

svr = make_pipeline(RobustScaler(),
                    SVR(C=21, epsilon=0.0099, gamma=0.00017, tol=0.000121))

# gradientboosting:

gbr = GradientBoostingRegressor(n_estimators=2900,
                                learning_rate=0.0161,
                                max_depth=4,
                                max_features='sqrt',
                                min_samples_leaf=17,
                                loss='huber',
                                random_state=42)

# lightgbm:

lightgbm = LGBMRegressor(objective='regression',
                         n_estimators=3500,
                         num_leaves=5,
                         learning_rate=0.00721,
                         max_bin=163,
                         bagging_fraction=0.35711,
                         n_jobs=-1,
                         bagging_seed=42,
                         feature_fraction_seed=42,
                         bagging_freq=7,
                         feature_fraction=0.1294,
                         min_data_in_leaf=8)

# xgboost:

xgboost = XGBRegressor(
```

```
    learning_rate=0.0139,
    n_estimators=4500,
    max_depth=4,
    min_child_weight=0,
    subsample=0.7968,
    colsample_bytree=0.4064,
    nthread=-1,
    scale_pos_weight=2,
    seed=42,
)


# hist gradient boosting regressor:

hgrd= HistGradientBoostingRegressor(    loss= 'least_squares',
    max_depth= 2,
    min_samples_leaf= 40,
    max_leaf_nodes= 29,
    learning_rate= 0.15,
    max_iter= 225,
                                    random_state=42)

# tweedie regressor:

tweed = make_pipeline(RobustScaler(),TweedieRegressor(alpha=0.005))


# stacking regressor:

stack_gen = StackingCVRegressor(regressors=(ridge, lasso, elasticnet, gbr,
                                            xgboost, lightgbm,hgrd, tweed),
                                meta_regressor=xgboost,
                                use_features_in_secondary=True)
```

9 Cross Validation

[66]:
```
def model_check(X, y, estimators, cv):

    ''' A function for testing multiple estimators.'''

    model_table = pd.DataFrame()

    row_index = 0
    for est, label in zip(estimators, labels):

        MLA_name = label
```

```
        model_table.loc[row_index, 'Model Name'] = MLA_name

        cv_results = cross_validate(est,
                                    X,
                                    y,
                                    cv=cv,
                                    scoring='neg_root_mean_squared_error',
                                    return_train_score=True,
                                    n_jobs=-1)

        model_table.loc[row_index, 'Train RMSE'] = -cv_results[
            'train_score'].mean()
        model_table.loc[row_index, 'Test RMSE'] = -cv_results[
            'test_score'].mean()
        model_table.loc[row_index, 'Test Std'] = cv_results['test_score'].std()
        model_table.loc[row_index, 'Time'] = cv_results['fit_time'].mean()

        row_index += 1

    model_table.sort_values(by=['Test RMSE'],
                            ascending=True,
                            inplace=True)

    return model_table
```

[67]:
```
# Setting list of estimators and labels for them:

estimators = [ridge, lasso, elasticnet, gbr, xgboost, lightgbm, svr, hgrd,␣
 ↪tweed]
labels = [
    'Ridge', 'Lasso', 'Elasticnet', 'GradientBoostingRegressor',
    'XGBRegressor', 'LGBMRegressor', 'SVR',␣
 ↪'HistGradientBoostingRegressor','TweedieRegressor'
]
```

10 Model Results

Allright, our results are here. Looks like our models did pretty close to each other, there might be some overfitting models and we can try to fix them by tuning but it was computationally expensive for me and since I'm going to stack and blend the models I think we can leave them as it is. We already added our models to stacking regression and set the XGBoost as meta regressor we can continue with stacking

[68]:
```
# Executing cross validation.

raw_models = model_check(X, y, estimators, kf)
display(raw_models.style.background_gradient(cmap='summer_r'))
```

```
<pandas.io.formats.style.Styler at 0x2c4f6597d30>
```

10.1 Stacking & Blending

Here we fit every single estimator we have on the train data and then blend them by assigning weights to each model and sum the results. Weights are pretty subjective and I'm pretty sure you can come up with something performs better than this if you play with it…

[69]:
```
# Fitting the models on train data.

print('=' * 20, 'START Fitting', '=' * 20)
print('=' * 55)

print(datetime.now(), 'StackingCVRegressor')
stack_gen_model = stack_gen.fit(X.values, y.values)
print(datetime.now(), 'Elasticnet')
elastic_model_full_data = elasticnet.fit(X, y)
print(datetime.now(), 'Lasso')
lasso_model_full_data = lasso.fit(X, y)
print(datetime.now(), 'Ridge')
ridge_model_full_data = ridge.fit(X, y)
print(datetime.now(), 'SVR')
svr_model_full_data = svr.fit(X, y)
print(datetime.now(), 'GradientBoosting')
gbr_model_full_data = gbr.fit(X, y)
print(datetime.now(), 'XGboost')
xgb_model_full_data = xgboost.fit(X, y)
print(datetime.now(), 'Lightgbm')
lgb_model_full_data = lightgbm.fit(X, y)
print(datetime.now(), 'Hist')
hist_full_data = hgrd.fit(X, y)
print(datetime.now(), 'Tweed')
tweed_full_data = tweed.fit(X, y)
print('=' * 20, 'FINISHED Fitting', '=' * 20)
print('=' * 58)
```

```
==================== START Fitting ====================
=======================================================
2021-08-27 16:28:49.494439 StackingCVRegressor
[LightGBM] [Warning] feature_fraction is set=0.1294, colsample_bytree=1.0 will
be ignored. Current value: feature_fraction=0.1294
[LightGBM] [Warning] min_data_in_leaf is set=8, min_child_samples=20 will be
ignored. Current value: min_data_in_leaf=8
[LightGBM] [Warning] bagging_fraction is set=0.35711, subsample=1.0 will be
ignored. Current value: bagging_fraction=0.35711
[LightGBM] [Warning] bagging_freq is set=7, subsample_freq=0 will be ignored.
Current value: bagging_freq=7
[LightGBM] [Warning] feature_fraction is set=0.1294, colsample_bytree=1.0 will
```

```
be ignored. Current value: feature_fraction=0.1294
[LightGBM] [Warning] min_data_in_leaf is set=8, min_child_samples=20 will be
ignored. Current value: min_data_in_leaf=8
[LightGBM] [Warning] bagging_fraction is set=0.35711, subsample=1.0 will be
ignored. Current value: bagging_fraction=0.35711
[LightGBM] [Warning] bagging_freq is set=7, subsample_freq=0 will be ignored.
Current value: bagging_freq=7
[LightGBM] [Warning] feature_fraction is set=0.1294, colsample_bytree=1.0 will
be ignored. Current value: feature_fraction=0.1294
[LightGBM] [Warning] min_data_in_leaf is set=8, min_child_samples=20 will be
ignored. Current value: min_data_in_leaf=8
[LightGBM] [Warning] bagging_fraction is set=0.35711, subsample=1.0 will be
ignored. Current value: bagging_fraction=0.35711
[LightGBM] [Warning] bagging_freq is set=7, subsample_freq=0 will be ignored.
Current value: bagging_freq=7
[LightGBM] [Warning] feature_fraction is set=0.1294, colsample_bytree=1.0 will
be ignored. Current value: feature_fraction=0.1294
[LightGBM] [Warning] min_data_in_leaf is set=8, min_child_samples=20 will be
ignored. Current value: min_data_in_leaf=8
[LightGBM] [Warning] bagging_fraction is set=0.35711, subsample=1.0 will be
ignored. Current value: bagging_fraction=0.35711
[LightGBM] [Warning] bagging_freq is set=7, subsample_freq=0 will be ignored.
Current value: bagging_freq=7
[LightGBM] [Warning] feature_fraction is set=0.1294, colsample_bytree=1.0 will
be ignored. Current value: feature_fraction=0.1294
[LightGBM] [Warning] min_data_in_leaf is set=8, min_child_samples=20 will be
ignored. Current value: min_data_in_leaf=8
[LightGBM] [Warning] bagging_fraction is set=0.35711, subsample=1.0 will be
ignored. Current value: bagging_fraction=0.35711
[LightGBM] [Warning] bagging_freq is set=7, subsample_freq=0 will be ignored.
Current value: bagging_freq=7
[LightGBM] [Warning] feature_fraction is set=0.1294, colsample_bytree=1.0 will
be ignored. Current value: feature_fraction=0.1294
[LightGBM] [Warning] min_data_in_leaf is set=8, min_child_samples=20 will be
ignored. Current value: min_data_in_leaf=8
[LightGBM] [Warning] bagging_fraction is set=0.35711, subsample=1.0 will be
ignored. Current value: bagging_fraction=0.35711
[LightGBM] [Warning] bagging_freq is set=7, subsample_freq=0 will be ignored.
Current value: bagging_freq=7
2021-08-27 16:32:58.751305 Elasticnet
2021-08-27 16:33:04.303518 Lasso
2021-08-27 16:33:05.732289 Ridge
2021-08-27 16:33:06.967575 SVR
2021-08-27 16:33:07.448319 GradientBoosting
2021-08-27 16:33:22.561613 XGboost
2021-08-27 16:33:39.832273 Lightgbm
2021-08-27 16:33:41.251166 Hist
2021-08-27 16:33:42.043636 Tweed
```

```
==================== FINISHED Fitting ====================
==========================================================
```

[70]:
```
# Blending models by assigning weights:

def blend_models_predict(X):
    return ((0.1 * elastic_model_full_data.predict(X)) +
            (0.1 * lasso_model_full_data.predict(X)) +
            (0.1 * ridge_model_full_data.predict(X)) +
            (0.1 * svr_model_full_data.predict(X)) +
            (0.05 * gbr_model_full_data.predict(X)) +
            (0.1 * xgb_model_full_data.predict(X)) +
            (0.05 * lgb_model_full_data.predict(X)) +
            (0.05 * hist_full_data.predict(X)) +
            (0.1 * tweed_full_data.predict(X)) +
            (0.25 * stack_gen_model.predict(X.values)))
```

10.2 Submission

Our models are tuned, stacked, fitted and blended so we are ready to predict and submit our results. One last thing that I have seen on couple examples adding weights on some quantile levels. It didn't increase my results a lot but still improved the end results a little so I decided to use it.

[72]:
```
submission = pd.read_csv('house_price/input/test.csv')
# Inversing and flooring log scaled sale price predictions
submission['SalePrice'] = np.floor(np.expm1(blend_models_predict(X_test)))
# Defining outlier quartile ranges
q1 = submission['SalePrice'].quantile(0.0050)
q2 = submission['SalePrice'].quantile(0.99)

# Applying weights to outlier ranges to smooth them
submission['SalePrice'] = submission['SalePrice'].apply(
    lambda x: x if x > q1 else x * 0.77)
submission['SalePrice'] = submission['SalePrice'].apply(lambda x: x
                                                        if x < q2 else x * 1.1)
submission = submission[['Id', 'SalePrice']]
```

[73]:
```
# Saving submission file

submission.to_csv('mysubmission.csv', index=False)
print(
    'Save submission',
    datetime.now(),
)
submission.head()
```

```
Save submission 2021-08-27 16:43:55.400722
```

[73]:

	Id	SalePrice
0	1461	118470.0
1	1462	159292.0
2	1463	188552.0
3	1464	198171.0
4	1465	187888.0