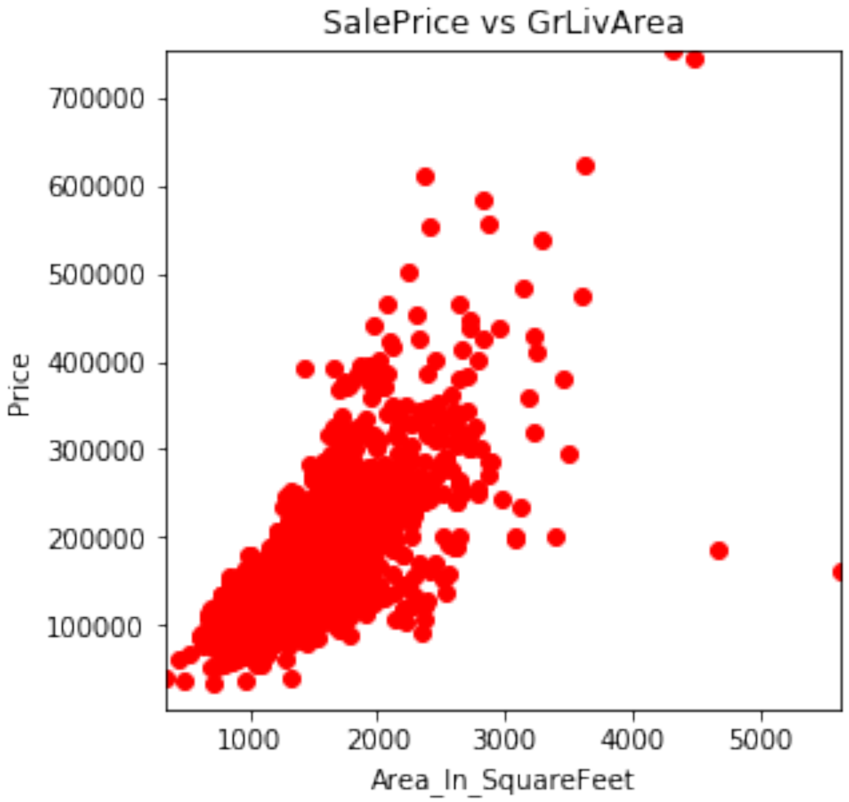

There is a possitive correlation between SalePrice and GrLivArea.

```
In [28]: print("Correlation coefficient:", np.corrcoef(SalePrice,GrLivArea))
```

```
Correlation coefficient: [[1.         0.70862448]
 [0.70862448 1.        ]]
```

7 IV. Identify additional relevant feature

```
In [29]: #find the correlation among the columns in the dataframe
         housedata[housedata.columns[1:]].corr()['SalePrice'][:-1]
```

```
Out[29]: MSSubClass      -0.084284
         LotFrontage      0.351799
         LotArea          0.263843
         OverallQual      0.790982
         OverallCond     -0.077856
         YearBuilt        0.522897
         YearRemodAdd     0.507101
         MasVnrArea       0.477493
         BsmtFinSF1       0.386420
```

```
BsmtFinSF2      -0.011378
BsmtUnfSF        0.214479
TotalBsmtSF      0.613581
1stFlrSF         0.605852
2ndFlrSF         0.319334
LowQualFinSF    -0.025606
GrLivArea        0.708624
BsmtFullBath     0.227122
BsmtHalfBath    -0.016844
FullBath         0.560664
HalfBath         0.284108
BedroomAbvGr     0.168213
KitchenAbvGr    -0.135907
TotRmsAbvGrd     0.533723
Fireplaces       0.466929
GarageYrBlt      0.486362
GarageCars       0.640409
GarageArea       0.623431
WoodDeckSF       0.324413
OpenPorchSF      0.315856
EnclosedPorch   -0.128578
3SsnPorch        0.044584
ScreenPorch      0.111447
PoolArea         0.092404
MiscVal         -0.021190
MoSold           0.046432
YrSold          -0.028923
Name: SalePrice, dtype: float64
```

Find features with high correlation with salePrice: I will pick 1stFlrSF: First Floor square feet.

```
In [30]: housedata["total_area"]= housedata["GrLivArea"]+housedata["TotalBsmtSF"]
         housedata["AreaPerRoom"]= housedata["GrLivArea"]/ housedata["TotRmsAbvGrd"]
         housedata.head()
```

```
Out[30]:    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     ...      Fence MiscFeature MiscVal MoSold YrSold  \
         0         Lvl    AllPub     ...        NaN         NaN       0      2   2008
         1         Lvl    AllPub     ...        NaN         NaN       0      5   2007
         2         Lvl    AllPub     ...        NaN         NaN       0      9   2008
         3         Lvl    AllPub     ...        NaN         NaN       0      2   2006
         4         Lvl    AllPub     ...        NaN         NaN       0     12   2008
```

```
  SaleType SaleCondition  SalePrice  total_area  AreaPerRoom
0       WD        Normal     208500        2566   213.750000
1       WD        Normal     181500        2524   210.333333
2       WD        Normal     223500        2706   297.666667
3       WD       Abnorml     140000        2473   245.285714
4       WD        Normal     250000        3343   244.222222

[5 rows x 83 columns]
```

8 V. Prepare data for k-Nearest-Neighbor method.

```
In [31]: housedatakNN = housedata.filter(['OverallQual','YearBuilt','TotalBsmtSF','GrLivArea',
                                           '1stFlrSF','total_area','AreaPerRoom','SalePrice'], a
         housedatakNN.head()
```

```
Out[31]:    OverallQual  YearBuilt  TotalBsmtSF  GrLivArea  1stFlrSF  total_area  \
         0            7       2003          856       1710       856        2566
         1            6       1976         1262       1262      1262        2524
         2            7       2001          920       1786       920        2706
         3            7       1915          756       1717       961        2473
         4            8       2000         1145       2198      1145        3343

            AreaPerRoom  SalePrice
         0   213.750000     208500
         1   210.333333     181500
         2   297.666667     223500
         3   245.285714     140000
         4   244.222222     250000
```

new data frame with SalePrice and the 7 selected features

```
In [32]: #No Missing Values
         pd.isnull(housedatakNN).sum()
```

```
Out[32]: OverallQual    0
         YearBuilt      0
         TotalBsmtSF    0
         GrLivArea      0
         1stFlrSF       0
         total_area     0
         AreaPerRoom    0
         SalePrice      0
         dtype: int64
```

```
In [52]: #mean value
         housedatakNN.mean(axis = 0)
```

```
Out[52]: OverallQual          6.099315
         YearBuilt         1971.267808
         TotalBsmtSF       1057.429452
         GrLivArea         1515.463699
         1stFlrSF          1162.626712
         total_area        2572.893151
         AreaPerRoom        230.905362
         SalePrice       180921.195890
         dtype: float64
```

```
In [34]: #standard deviation
         housedatakNN.std(axis = 0)
```

```
Out[34]: OverallQual         1.382997
         YearBuilt          30.202904
         TotalBsmtSF       438.705324
         GrLivArea         525.480383
         1stFlrSF          386.587738
         total_area        823.598492
         AreaPerRoom        44.740397
         SalePrice       79442.502883
         dtype: float64
```

9 Feature normalization

```
In [54]: housedatakNN_Normalization = (housedatakNN - housedatakNN.mean())/housedatakNN.std()
         housedatakNN_Normalization.head()
```

```
Out[54]:    OverallQual  YearBuilt  TotalBsmtSF  GrLivArea  1stFlrSF  total_area  \
         0     0.651256   1.050634    -0.459145   0.370207 -0.793162   -0.008370
         1    -0.071812   0.156680     0.466305  -0.482347  0.257052   -0.059365
         2     0.651256   0.984415    -0.313261   0.514836 -0.627611    0.161616
         3     0.651256  -1.862993    -0.687089   0.383528 -0.521555   -0.121289
         4     1.374324   0.951306     0.199611   1.298881 -0.045596    0.935051

            AreaPerRoom  SalePrice
         0    -0.383442   0.347154
         1    -0.459809   0.007286
         2     1.492193   0.535970
         3     0.321418  -0.515105
         4     0.297647   0.869545
```

10 VI. Apply the kNN (k=5) method to predict sale price of the first instance from the test set.

```
In [36]: # Extract files
         import zipfile
```

```
        # if not os.path.exists("Data"):
        #     os.mkdir("Data")
        with zipfile.ZipFile("Data/test.csv.zip", "r") as file:
            file.printdir()
            file.extractall("Data/house-prices")
```

```
File Name                                             Modified             Size
test.csv                                       2018-11-28 21:31:58       451405
__MACOSX/                                      2019-11-03 21:24:04            0
__MACOSX/._test.csv                            2018-11-28 21:31:58          212
```

```
In [37]: Testset = pd.read_csv("Data/house-prices/test.csv", delimiter=",")

In [38]: print("Feature names:", ", ".join(Testset.columns))
```

```
Feature names: Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandCo
```

```
In [39]: Testset.head()
```

```
Out[39]:      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      ...       ScreenPorch PoolArea PoolQC  Fence  \
         0         Lvl    AllPub      ...               120        0    NaN  MnPrv
         1         Lvl    AllPub      ...                 0        0    NaN    NaN
         2         Lvl    AllPub      ...                 0        0    NaN  MnPrv
         3         Lvl    AllPub      ...                 0        0    NaN    NaN
         4         HLS    AllPub      ...               144        0    NaN    NaN

           MiscFeature MiscVal MoSold  YrSold  SaleType  SaleCondition
         0         NaN       0      6    2010        WD         Normal
         1        Gar2   12500      6    2010        WD         Normal
         2         NaN       0      3    2010        WD         Normal
         3         NaN       0      6    2010        WD         Normal
         4         NaN       0      1    2010        WD         Normal

         [5 rows x 80 columns]
```

```
In [40]: #First we need to add The new previous Column to the Test Set Frame.
         Testset["total_area"]= Testset["GrLivArea"]+housedata["TotalBsmtSF"]
         Testset["AreaPerRoom"]= Testset["GrLivArea"]/ housedata["TotRmsAbvGrd"]
         Testset.head()
```

```
Out[40]:      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     ...       PoolQC  Fence MiscFeature MiscVal MoSold  \
         0         Lvl    AllPub     ...          NaN  MnPrv         NaN       0      6
         1         Lvl    AllPub     ...          NaN    NaN        Gar2   12500      6
         2         Lvl    AllPub     ...          NaN  MnPrv         NaN       0      3
         3         Lvl    AllPub     ...          NaN    NaN         NaN       0      6
         4         HLS    AllPub     ...          NaN    NaN         NaN       0      1

           YrSold SaleType  SaleCondition  total_area  AreaPerRoom
         0   2010       WD         Normal      1752.0   112.000000
         1   2010       WD         Normal      2591.0   221.500000
         2   2010       WD         Normal      2549.0   271.500000
         3   2010       WD         Normal      2360.0   229.142857
         4   2010       WD         Normal      2425.0   142.222222

         [5 rows x 82 columns]
```

```
In [41]: #Selecting the Specific 7 featured From Test set
         TestsetkNN = Testset.filter(['OverallQual','YearBuilt','TotalBsmtSF','GrLivArea',
                                      '1stFlrSF','total_area','AreaPerRoom'], axis=1)
         TestsetkNN.head()
```

```
Out[41]:    OverallQual  YearBuilt  TotalBsmtSF  GrLivArea  1stFlrSF  total_area  \
         0            5       1961        882.0        896       896      1752.0
         1            6       1958       1329.0       1329      1329      2591.0
         2            5       1997        928.0       1629       928      2549.0
         3            6       1998        926.0       1604       926      2360.0
         4            8       1992       1280.0       1280      1280      2425.0

            AreaPerRoom
         0   112.000000
         1   221.500000
         2   271.500000
         3   229.142857
         4   142.222222
```

```
In [79]: #Feature normalization
         TestsetkNN_Normalization = (TestsetkNN - housedatakNN.mean())/housedatakNN.std()
         TestsetkNN_Normalization.head()
```

```
Out[79]:    1stFlrSF  AreaPerRoom  GrLivArea  OverallQual  SalePrice  TotalBsmtSF  \
         0 -0.689693    -2.657673  -1.178852    -0.794879        NaN    -0.399880
         1  0.430364    -0.210221  -0.354844    -0.071812        NaN     0.619027
```

```
2 -0.606917     0.907337   0.216062    -0.794879        NaN    -0.295026
3 -0.612091    -0.039394   0.168486    -0.071812        NaN    -0.299585
4  0.303614    -1.982172  -0.448092     1.374324        NaN     0.507335

   YearBuilt  total_area
0  -0.339961   -0.996715
1  -0.439289    0.021985
2   0.851977   -0.029011
3   0.885087   -0.258491
4   0.686430   -0.179569
```

```
In [80]: #first instance Values After normalized
         Y = TestsetkNN_Normalization.iloc[0:1]
         Y.head()
```

```
Out[80]:    1stFlrSF  AreaPerRoom  GrLivArea  OverallQual  SalePrice  TotalBsmtSF  \
         0 -0.689693    -2.657673  -1.178852    -0.794879        NaN     -0.39988

            YearBuilt  total_area
         0  -0.339961   -0.996715
```

```
In [82]: #training data X
         X = housedatakNN
         X.head()
```

```
Out[82]:    OverallQual  YearBuilt  TotalBsmtSF  GrLivArea  1stFlrSF  total_area  \
         0            7       2003          856       1710       856        2566
         1            6       1976         1262       1262      1262        2524
         2            7       2001          920       1786       920        2706
         3            7       1915          756       1717       961        2473
         4            8       2000         1145       2198      1145        3343

            AreaPerRoom  SalePrice
         0   213.750000     208500
         1   210.333333     181500
         2   297.666667     223500
         3   245.285714     140000
         4   244.222222     250000
```