HOUSING: PRICE PREDICTION

Submitted by:

P.MANIVANNAN

INTRODUCTION

Problem Statement:

The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:

- Which variables are important to predict the price of variable?
- How do these variables describe the price of the house?

- Conceptual Background of the Domain Problem

Houses are one of the necessary needs of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.

- Review of Literature

The Research is done on Housing Market and how each variable play a role in Price of the house with Data Science as tool the research is done. First Brief Exploratory Data Analysis is done which includes Handling Null values, Skewness Removal, Outlier Treatment, Visualizing the Data to Understand Correlation with House Price and understanding Relation between various input features. Then Data is Fed to Machine Learning Algorithms. after Evaluating and Testing Different Algorithms Best

Algorithm is Selected. Based on Evaluation Metrics such as Training Score, R2 Score, Cross Validation Score etc.

- Motivation for the Problem Undertaken

I always wondered how the housing brokers fix the price of property. During my first house purchase I was confused how they decide weather the price of house is overvalued or undervalued. This kept me motivated during this research to find what are important factors contribute to the Price of the house.

Analytical Problem Framing

- Data Sources and their formats

The data contains 1168 rows and 81 columns. The columns are

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

The data is in Csv format.

- Data Pre-processing Done

a. The data has null values in both categorical and Numerical Data. For handling the null values, I used Simple Imputer class

for numerical data I used mean method in simple imputer and for categorical data null values I used most_frequent method in simple imputer.

b. We had categorical data which needs to be encoded. for this approach I used pd.get_dummies method of pandas library as categorical data are not ordinal in nature.
c. Then Skewness Is checked for the data and Skewness is removed using power_transform function. Although skewness of some features is not entirely removed. Even after applying various methods like log, sqrt, cbrt methods i just left that features with skewness.
d. Then outliers is checked using Z Score method. Although when i tried to remove data above Z Score above 3 I found that Data loss is more than 30%. So, I just removed data with Z Score above 4 so data loss is minimal.
e. The data are then standardized using Standard Scaler.
f. As the columns count increased after encoding with dummies method. I decided to used Principal Component Analysis to reduce the No of columns. To decide the No of columns I used pca.explained_variance_ratio_ method to find how much each feature matters (maximum variance) .

- Data Inputs- Logic- Output Relationships

Using Correlation function corr(). The correlation of features is analysed a nd how it affects our target variable Sale Price. I found that OverallQual(Overall Quality) Feature has highest positive correlation with Target Varia ble ,then features such as GrLivArea , GarageCars, GarageArea, TotalBs mtSF, 1stFlrSF, FullBath, TotRmsAbvGrd has good positive correlation to target variable . On the Flip side KitchenAbvGr, EnclosedPorch features h ave negative correlation with Target Variable.

- State the set of assumptions (if any) related to the problem under consideration

a. For Handling the Null values Mean values are replaced in the place of null values
b. For handling the categorical data dummies method is used assuming the categorical data do not have ordinal relationship with each other. as more insight is need to conclude which category is ordinal or not its safe to use dummies method.
c. Skewness is removed assuming the prediction is not affected by skewness removal.
d. Outliers are removed, assuming that those data are really outliers and do not contribute to dataset
e. Principal component analysis technique is used to reduce the column count. But the variance of features are retained by using pca.explained_variance_ratio_ method to find the variance exhibited by different features.

- Hardware and Software Requirements and Tools Used

The Hardware requirements are 16 Gb RAM,500 GB SSD, at least 6 Cores processor

The software requirements are Windows, Anaconda Framework and libraries used are Pandas, NumPy, Sklearn, SciPy, Matplotlib, Seaborn

We use Pandas for reading the Dataset, Creating Dataframe, much more to handle data in dataset

Certain functions of NumPy are used like log , cbrt ,sqrt skewness transformation . absolute (abs) function used along with Zscore.

SKlearn Library is used for Machine Learning Algorithms like SVC, Random Forest, Linear Regression etc and Metrics for analysing them.

Scipy is scientific python library used for Zscore method etc.

Matplotlib and Seaborn are used for visualizations plotting graphs charts, etc

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Since the Target Variable is Continuous in nature I should Use Regression Approach to Solve this Problem and build a Regression Model

- Testing of Identified Approaches (Algorithms)
  a. Linear Regression
  b. Ridge Regression
  c. Lasso Regression
  d. Elastic Net Regression
  e. Decision Tree Regressor
  f. Random Forest Regressor
  g. Ada Boost Regressor
  h. KNeighbours Regressor

- Run and Evaluate selected models

  I have custom written a Code Block for Running all Algorithms and Storing them in DataFrame and Ranking them based on their Accuracy Score. The best part is the code finds the best random state for each algorithm and prints them

Code:

```
result = pd.DataFrame(columns=["Model Name","Train Score","Test Score","Cross Val Score"])
lin=[]
dec=[]
rid=[]
las=[]
kne=[]
rfr=[]
ada=[]
ela=[]
licol=[lin,dec,rid,las,kne,rfr,ada,ela]

algo = [LinearRegression(),DecisionTreeRegressor(),Ridge(),Lasso(),KNeighborsRegressor(),
        RandomForestRegressor(),AdaBoostRegressor(),ElasticNet()]
oo= 0
for v in algo:
    r = 0
    acc = 0
    for i in range(0,25):
        al = v
        train_x,test_x,train_y,test_y = train_test_split(x_final,y,test_size=.20,random_state=i)
        al.fit(train_x,train_y)
        score = al.score(train_x,train_y)
        if score>acc:
            acc = score
            r = i

    print(f'the best random state is {r} for {v}')


    train_x,test_x,train_y,test_y = train_test_split(x_final,y,test_size=.20,random_state=r)
    al.fit(train_x,train_y)
    trs = al.score(train_x,train_y)
    tss = al.score(test_x,test_y)
    cvs = cross_val_score(al,x_final,y,cv=KFold(5)).mean()
    licol[oo].insert(0,v)
    licol[oo].insert(1,trs)
    licol[oo].insert(2,tss)
    licol[oo].insert(3,cvs)
    result.loc[oo] = licol[oo]
    oo+=1

final_result = result.sort_values(by=["Cross Val Score","Test Score"],ascending=False)
```

If you see the code I have created a list of algorithm which I need to run. And here is the output:

Output1:

```
the best random state is 19 for LinearRegression()
the best random state is 0 for DecisionTreeRegressor()
the best random state is 19 for Ridge()
the best random state is 19 for Lasso()
the best random state is 15 for KNeighborsRegressor()
the best random state is 18 for RandomForestRegressor()
the best random state is 18 for AdaBoostRegressor()
the best random state is 19 for ElasticNet()
```

Output2:

	Model Name	Train Score	Test Score	Cross Val Score
2	Ridge()	0.922478	0.808978	0.868960
3	Lasso()	0.922539	0.808799	0.867689
0	LinearRegression()	0.922540	0.808722	0.867603
7	ElasticNet()	0.865149	0.765167	0.833501
5	(DecisionTreeRegressor(max_features='auto', ra...	0.979563	0.744732	0.818541
4	KNeighborsRegressor()	0.887029	0.713255	0.809422
6	(DecisionTreeRegressor(max_depth=3, random_sta...	0.893004	0.692484	0.771113
1	DecisionTreeRegressor()	1.000000	0.747593	0.633597

```
final_result.loc[2]["Model Name"]
```

```
Ridge()
```

Here you can see the Model Name, Training Score, Testing Score. Based on this best model is selected and further proceeded for hyperparameter tuning. Here Ridge Regression performed best compared to others.

Code:

```
rf = Ridge()
parameters = {'alpha': [0.0001,0.001,0.01,0.1,1,1.5,2],
 'fit_intercept': [True,False],
 'solver': ['auto', 'svd', 'cholesky', 'lsqr', 'sparse_cg','sag', 'saga', 'lbfgs'],
 'max_iter': [None,1000,1500,15000,2000]}
train_x,test_x,train_y,test_y = train_test_split(x_final,y,test_size=.20,random_state=19)
gsv = GridSearchCV(rf,parameters)
gsv.fit(train_x,train_y)
gsv.best_params_
```

Output:

```
{'alpha': 2, 'fit_intercept': True, 'max_iter': 1000, 'solver': 'saga'}
```

The Best Parameters are listed now we need to run the model with the best parameters

Code:

```
rf = Ridge(alpha=2,fit_intercept=True,max_iter=1000,solver='saga')
train_x,test_x,train_y,test_y = train_test_split(x_final,y,test_size=.15,random_state=19)
rf.fit(train_x,train_y)
trs = rf.score(train_x,train_y)
tss = rf.score(test_x,test_y)
pred = rf.predict(test_x)
cvs = cross_val_score(rf,x_final,y).mean()
print(f'the training score is {trs} the testing score is {tss} the cross val score is {cvs}')
print("Mean Squared Error",mean_squared_error(test_y,pred))
print("Mean Absolute Error", mean_absolute_error(test_y,pred))
print("Root Mean Squared Error", np.sqrt(mean_squared_error(test_y,pred)))
print("R2 Score", r2_score(test_y,pred))
```

Output:

```
the training score is 0.9024978429263666 the testing score is 0.8770356615937597 the cross val score is 0.8700964003980884
Mean Squared Error 699498777.5392836
Mean Absolute Error 18402.15741443228
Root Mean Squared Error 26448.039200275012
R2 Score 0.8770356615937597
```

The Best Training Score for Ridge Regression After Hyper parameter tuning for Training is 90% , Testing is 87% and Cross Validation Score is 87%

- Key Metrics for success in solving problem under consideration

  Since it's a Regression problem The key metrics used are Score method in Regression for training and testing, R2 Score, Mean Squared error, Mean Absolute Error, Root Mean Squared Error.

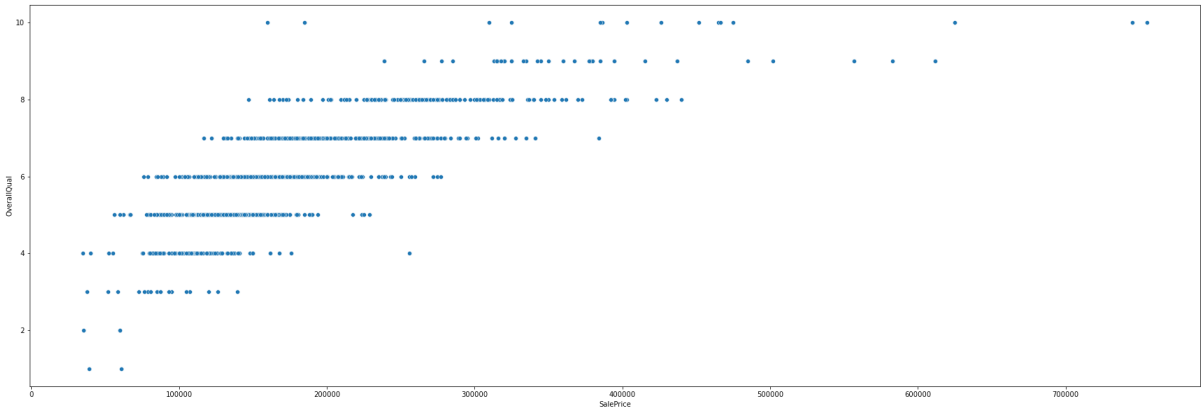
Visualizations

For visualizations the Matplotlib & Seaborn Library are used.

Code:

```
plt.figure(figsize=(30,10))
sns.scatterplot(df['SalePrice'],df['OverallQual'])
```

Output:

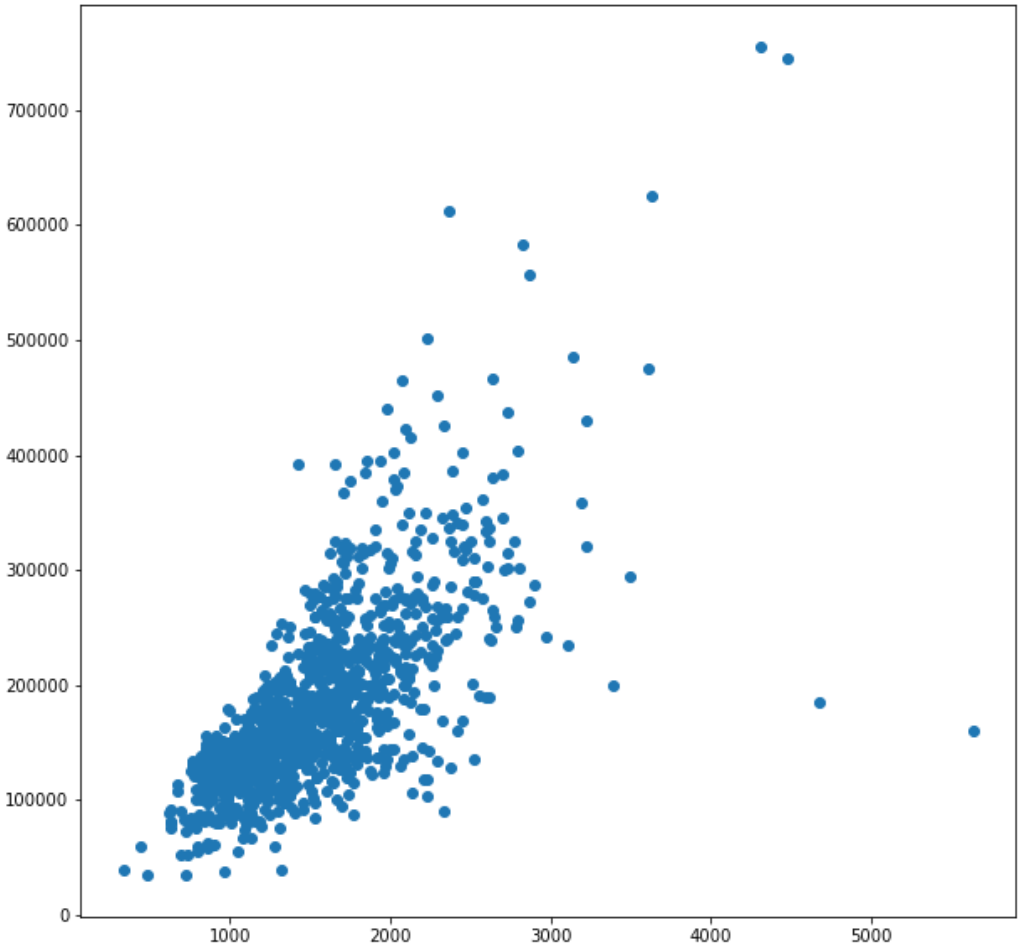


Observation:

OverallQual is Positively Correlated to Sale Price. with increase in Overall Quality the Sale Price also increases

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['GrLivArea'],df['SalePrice'])
```



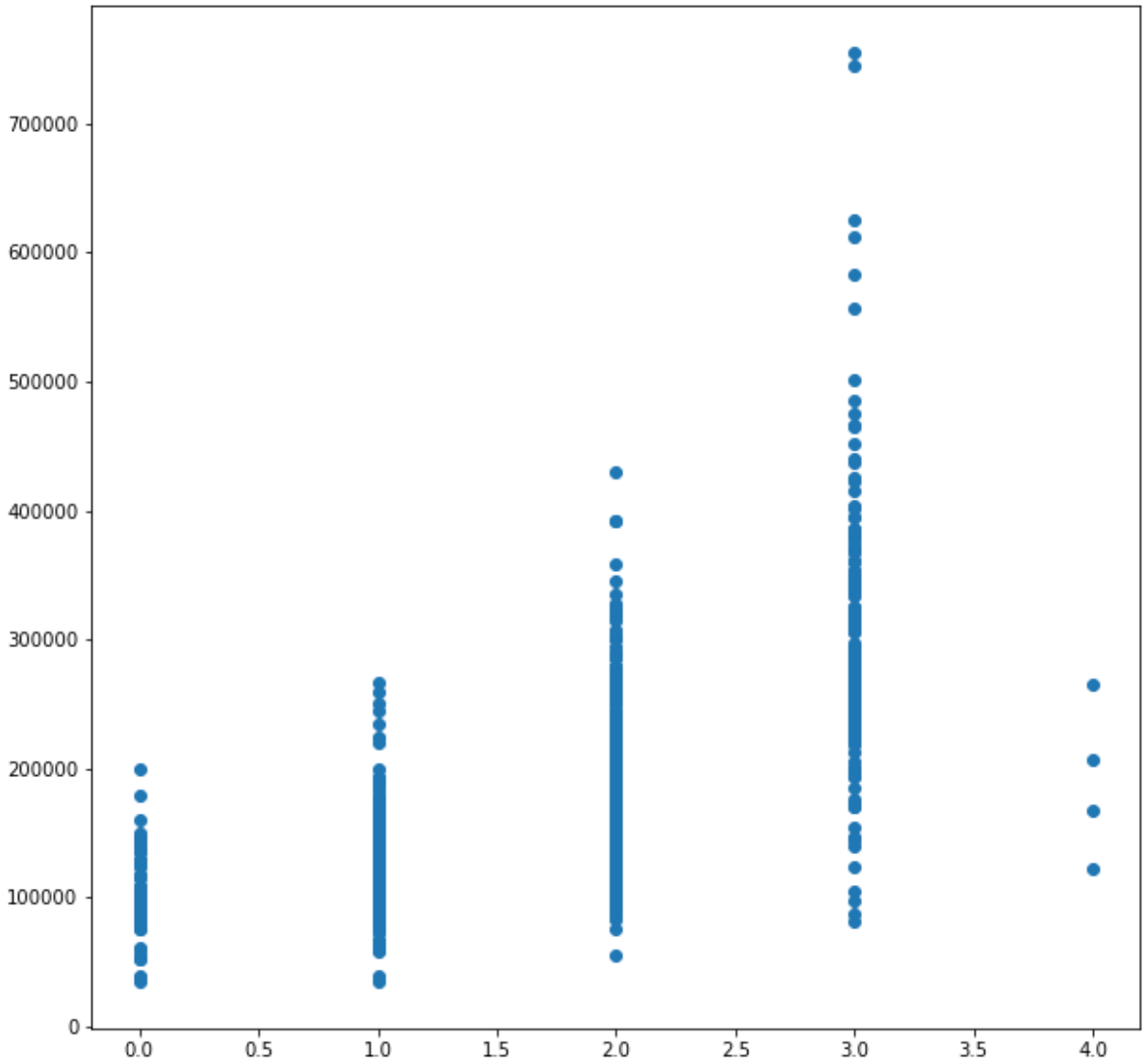
Observation:

(ground) living area square feet increases the Sale Price also Increases. Strong positive correlation

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['GarageCars'],df['SalePrice'])
```

Output:



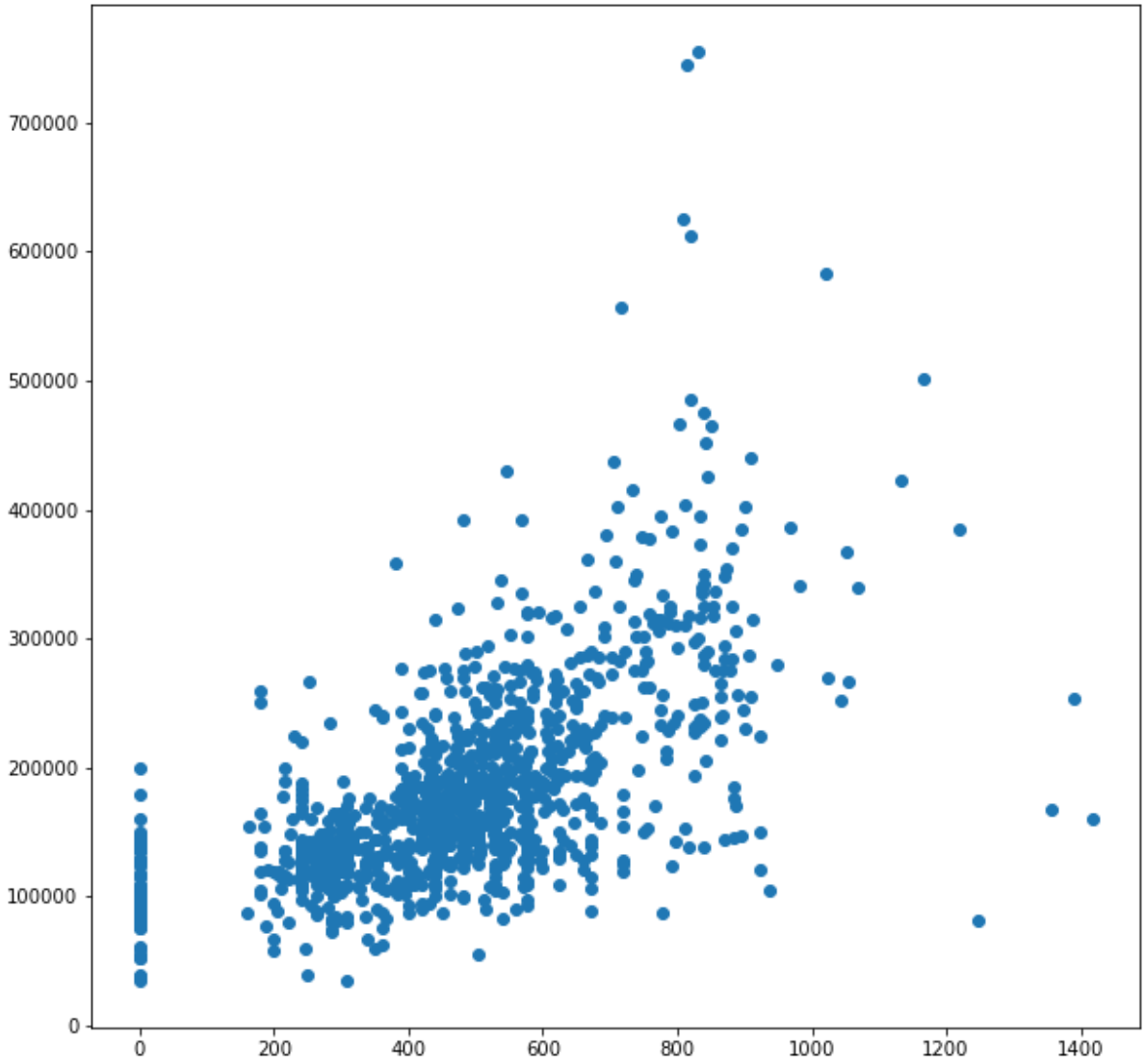
Observation:

Garage Car Capacity in terms of cars is Highly Positively Correlated to House price.

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['GarageArea'],df['SalePrice'])
```

Output:



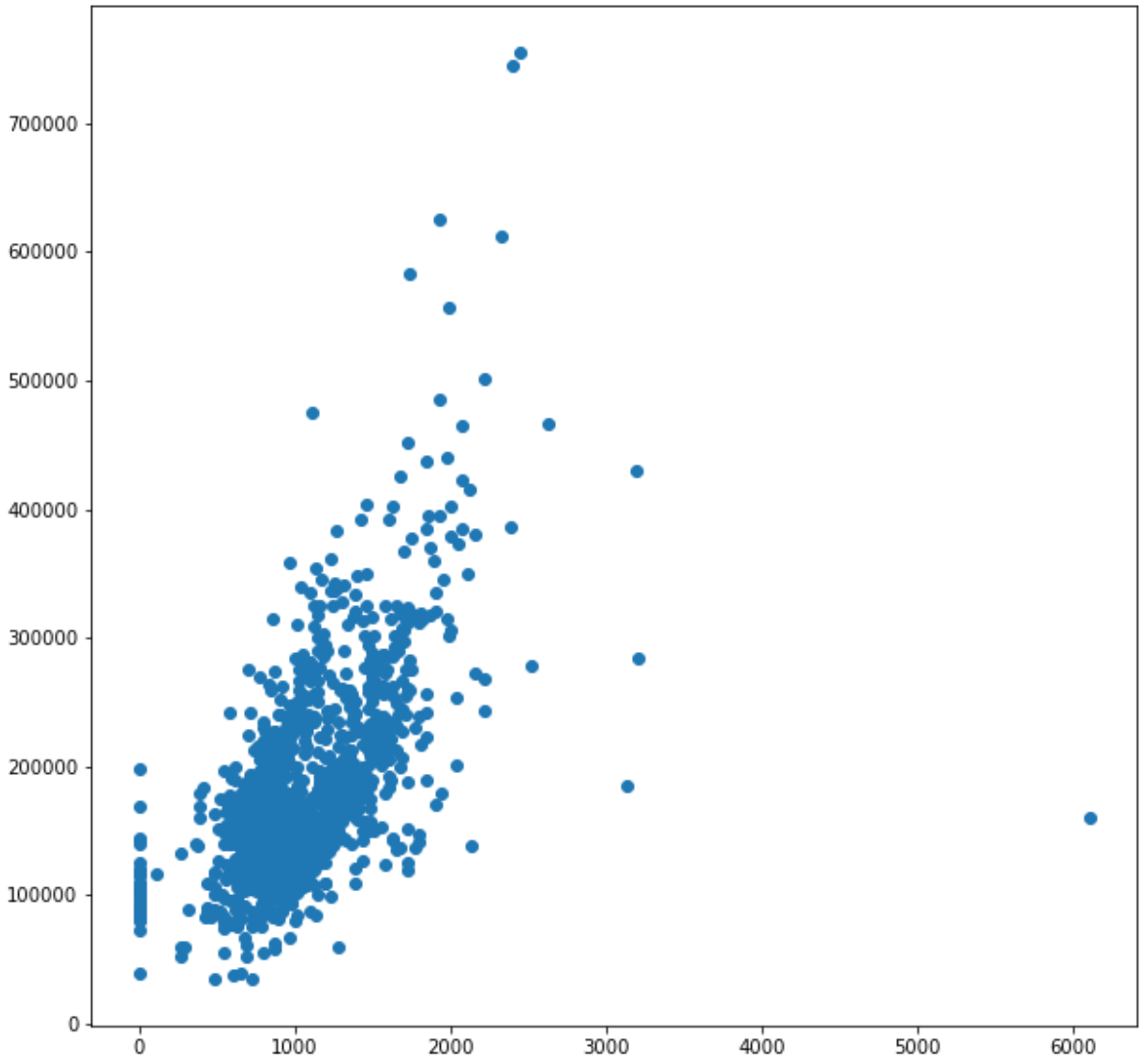
Observation:

Garage Area is also Highly Positively correlated to House price. it seems people prefer more garage space

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['TotalBsmtSF'],df['SalePrice'])
```

Output:



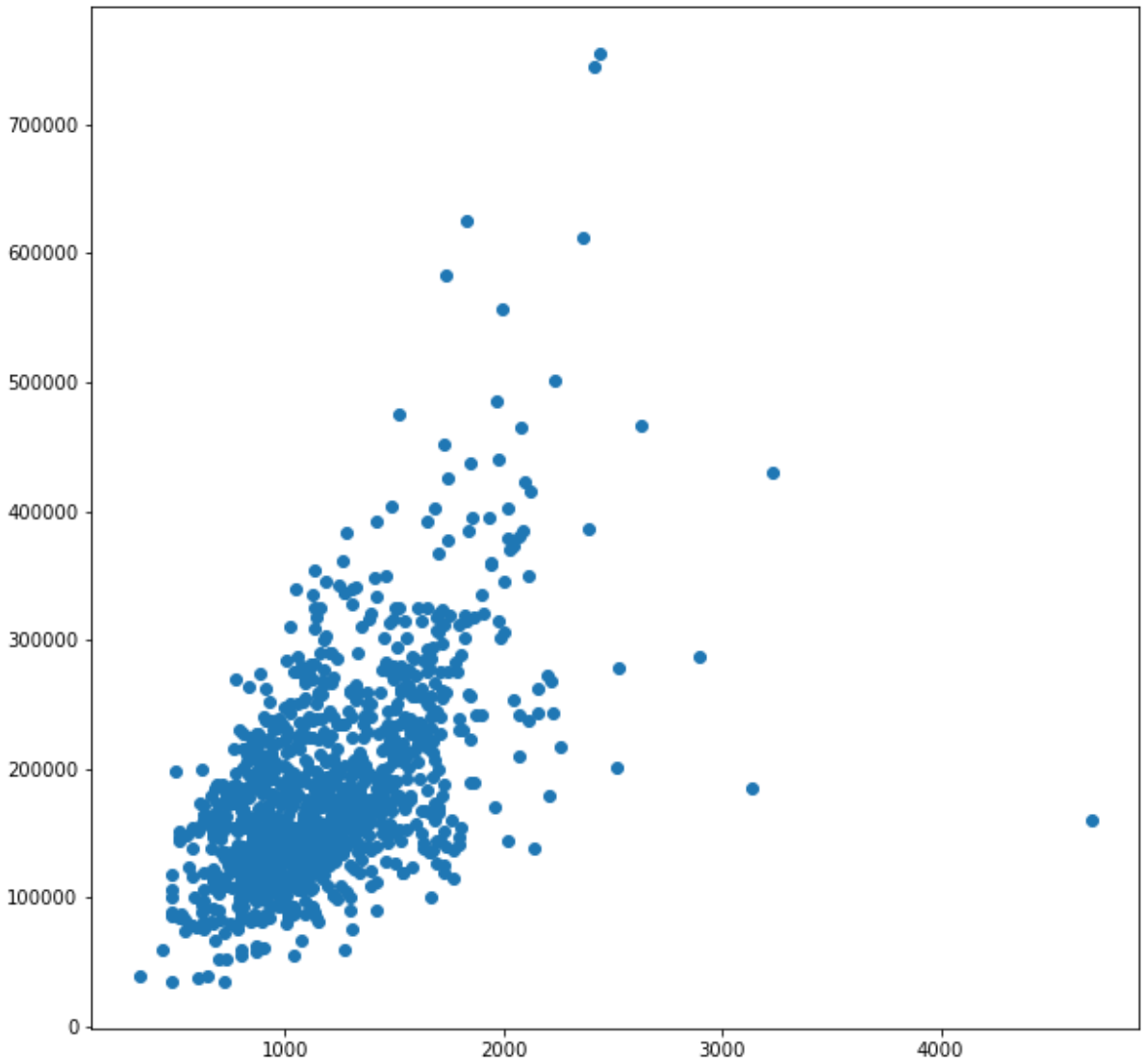
Observation:

Basement Area is also Positively Correlated to Sale Price . People prefer extra spacious basement

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['1stFlrSF'],df['SalePrice'])
```

Output:



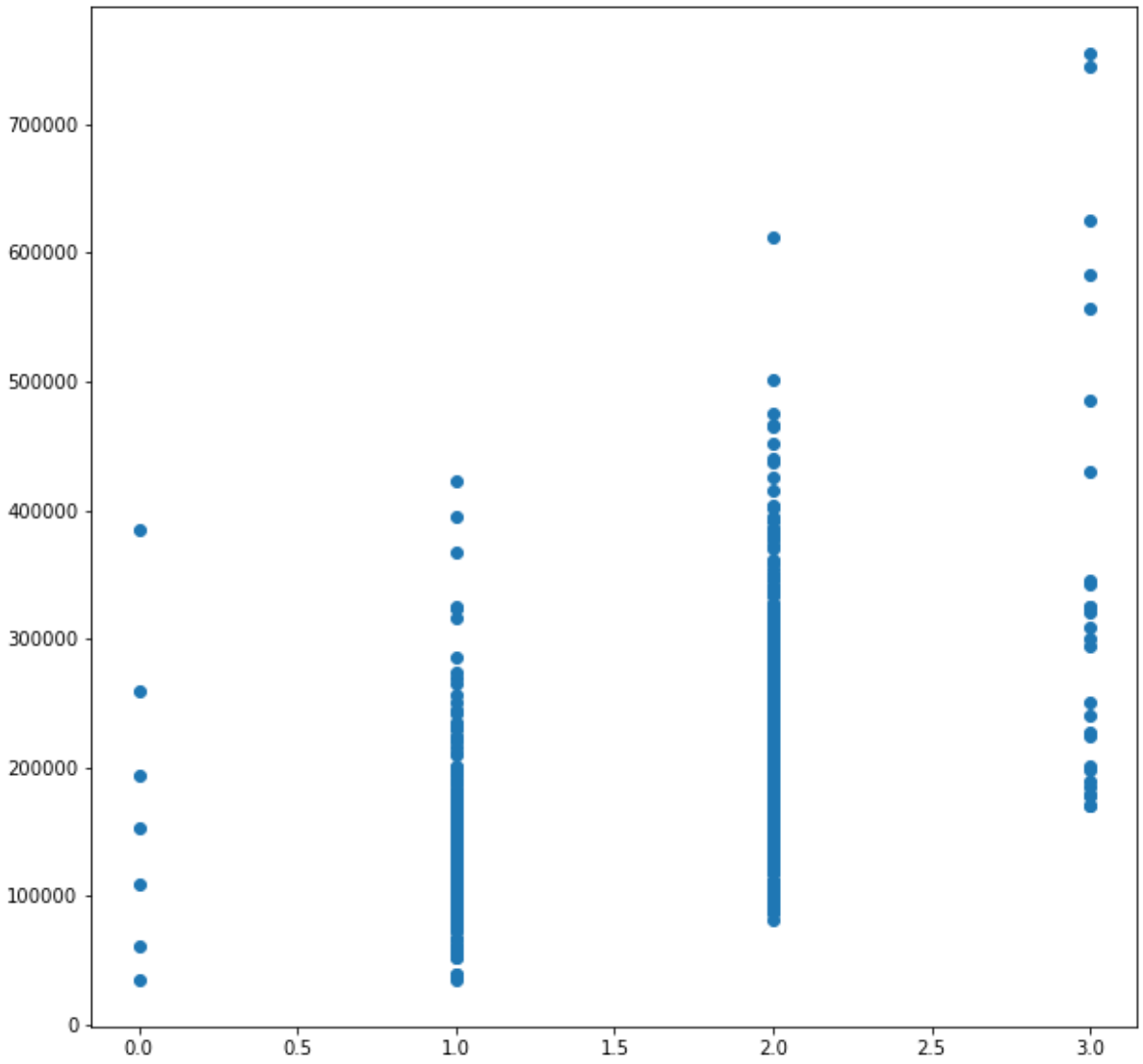
Observation:

First Floor in Square feet is also Positively Correlated to Sale Price. higher Square feet of first floor more space for the house

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['FullBath'],df['SalePrice'])
```

Output:



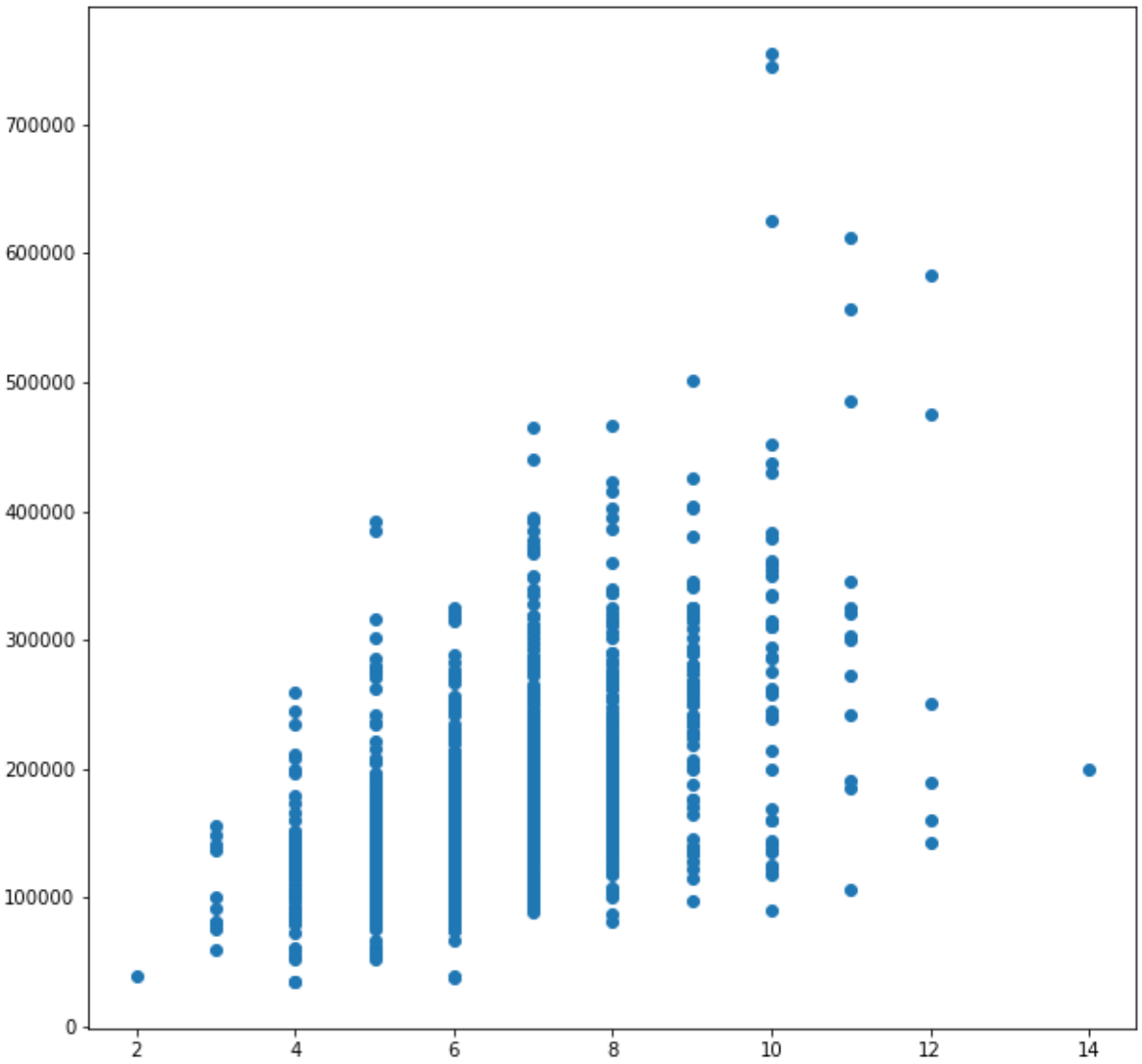
Observation:

As the Full bathroom above ground increases the Price of house also increases its strong positive correlation

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['TotRmsAbvGrd'],df['SalePrice'])
```

Output:



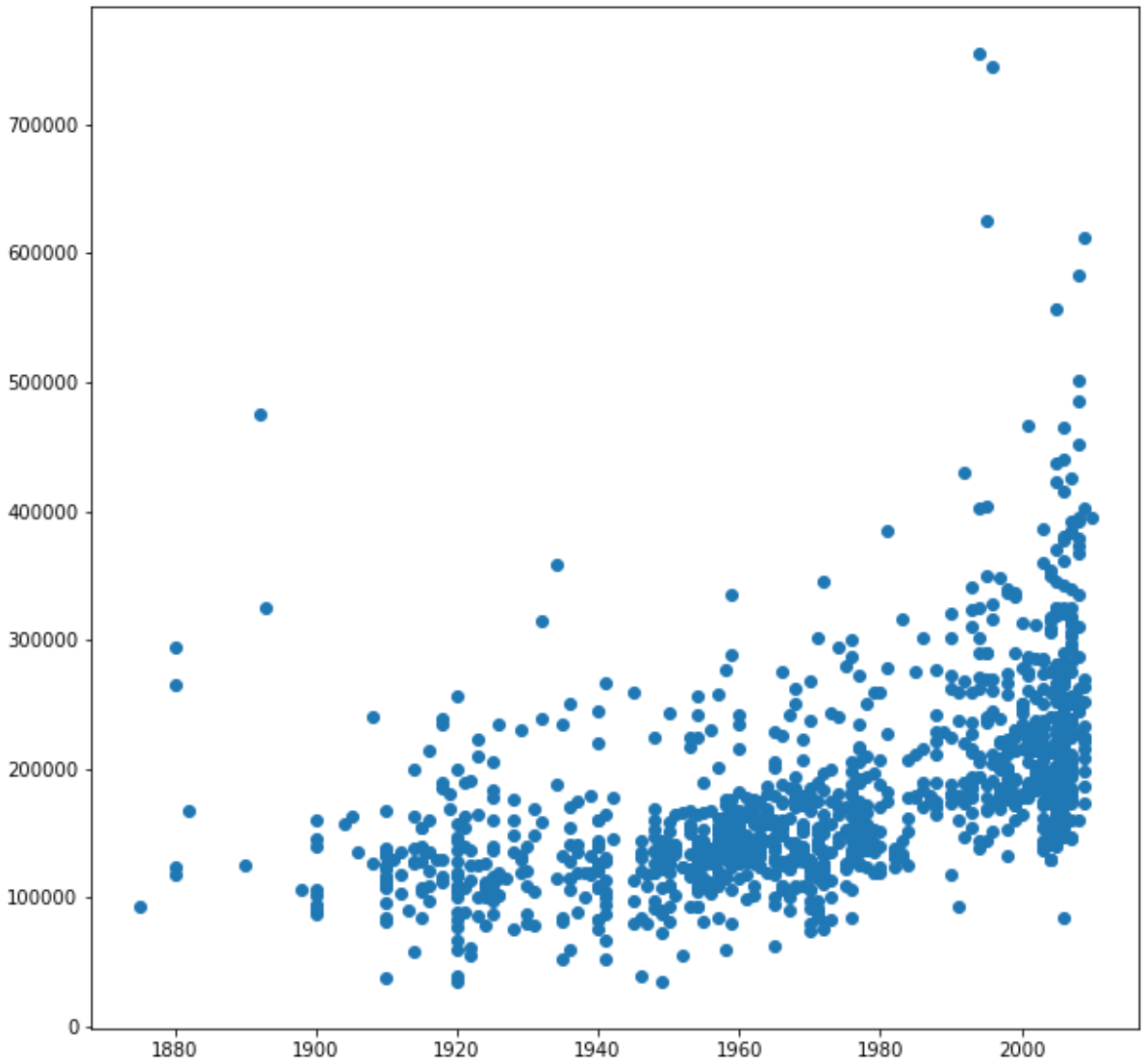
Observation:

As total rooms above ground increases the Price also increases strong positive correlation

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['YearBuilt'],df['SalePrice'])
```

Observation:



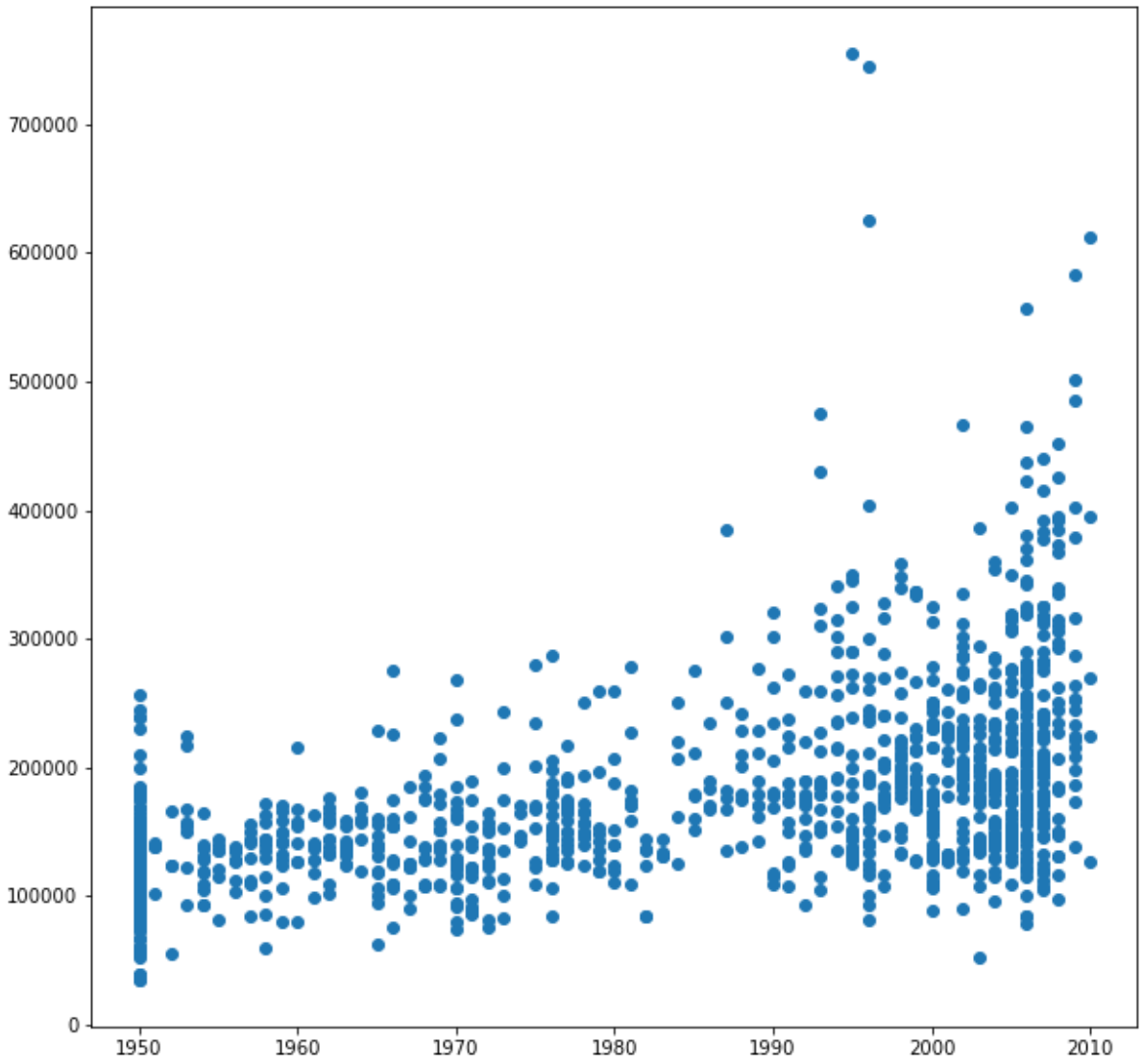
Output:

As the Year Increases price also increases. new house has higher price.

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['YearRemodAdd'],df['SalePrice'])
```

Output:



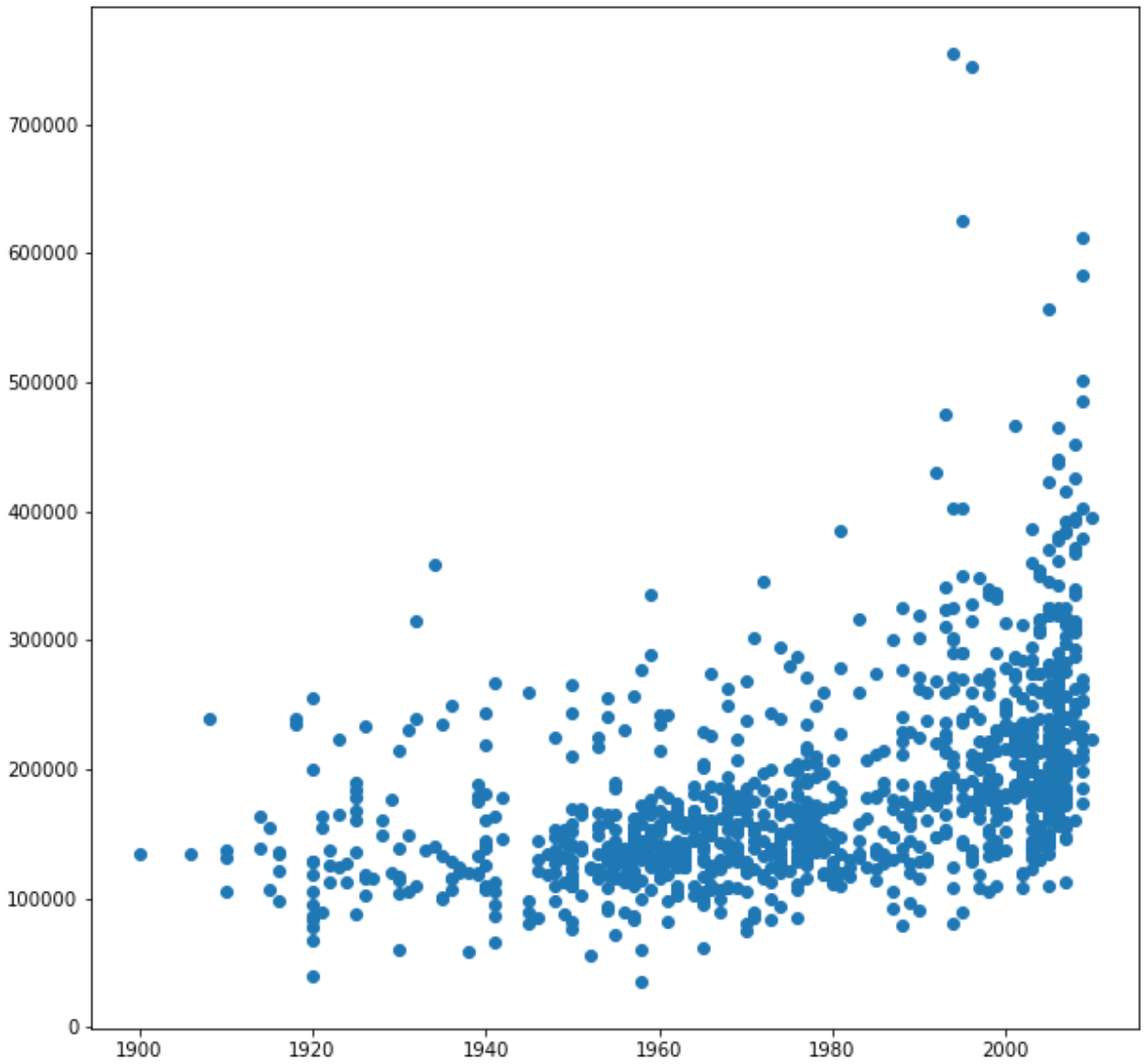
Observation:

as the remodelling or alteration Date increases the price also increases people want house to be reliable

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['GarageYrBlt'],df['SalePrice'])
```

Output:



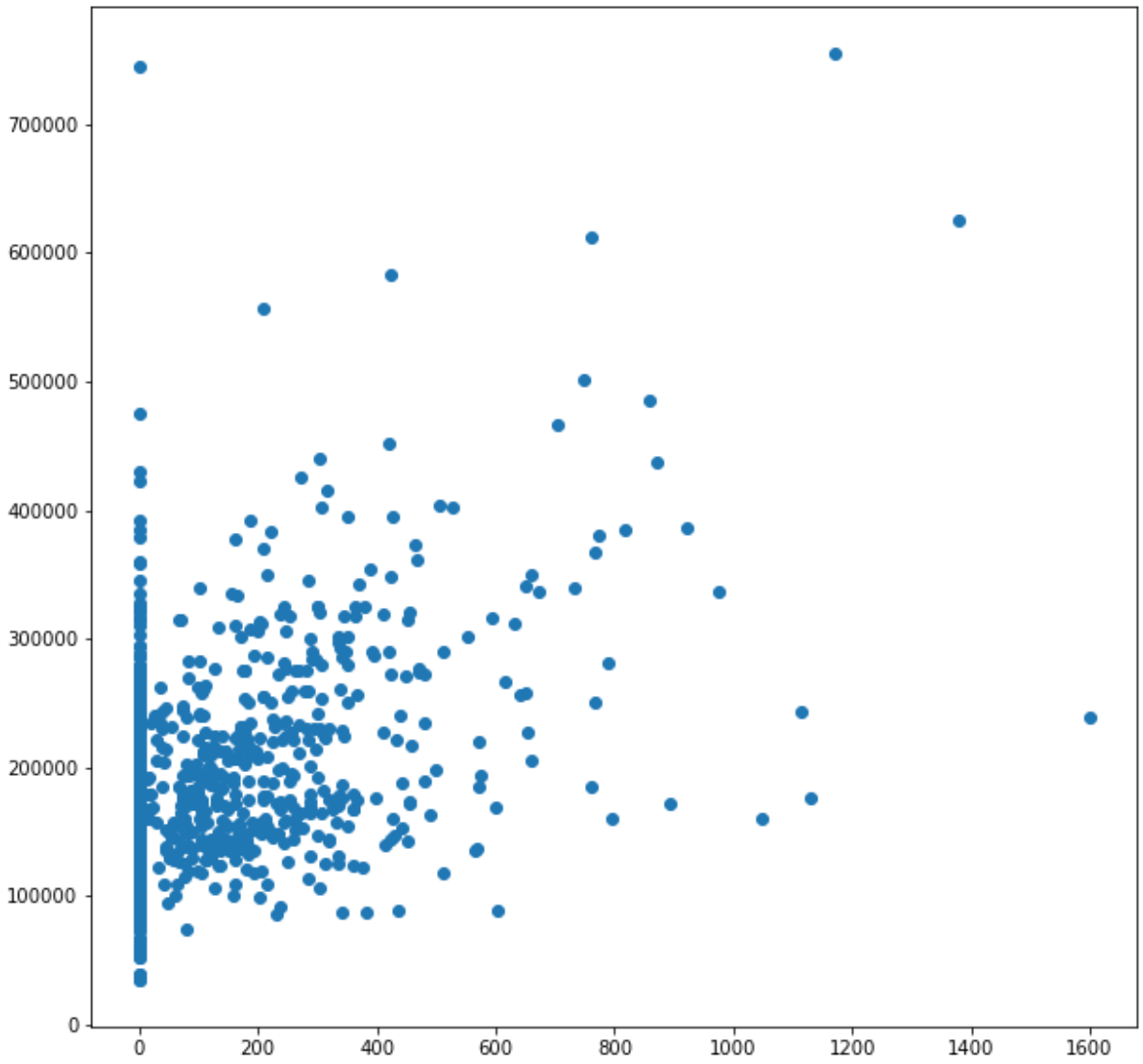
Observation:

As the Garage built year increases the price of the house also increases

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['MasVnrArea'],df['SalePrice'])
```

Output:



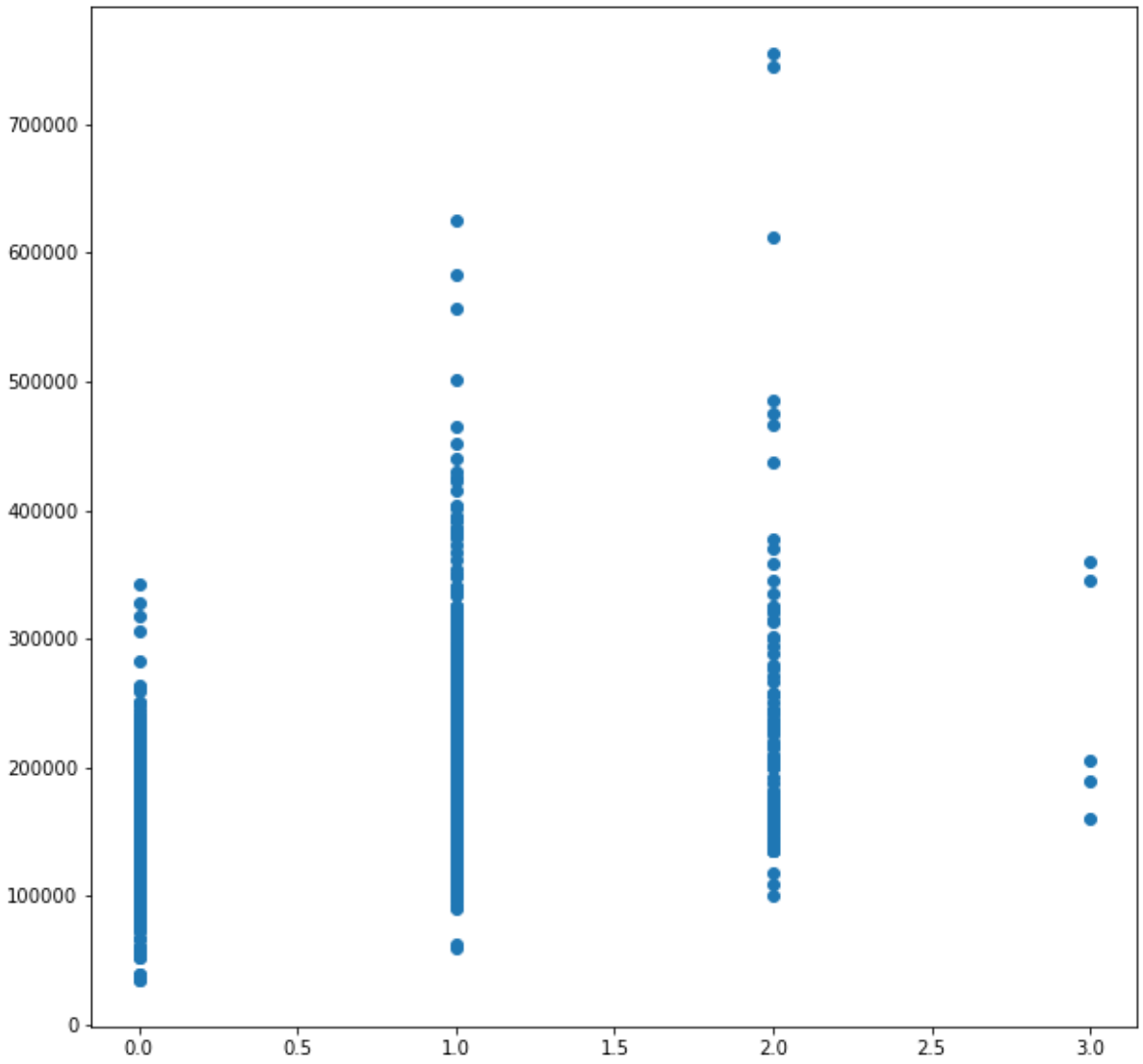
Observation:

Masonry veneer area increases the price also increases

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['Fireplaces'],df['SalePrice'])
```

Output:



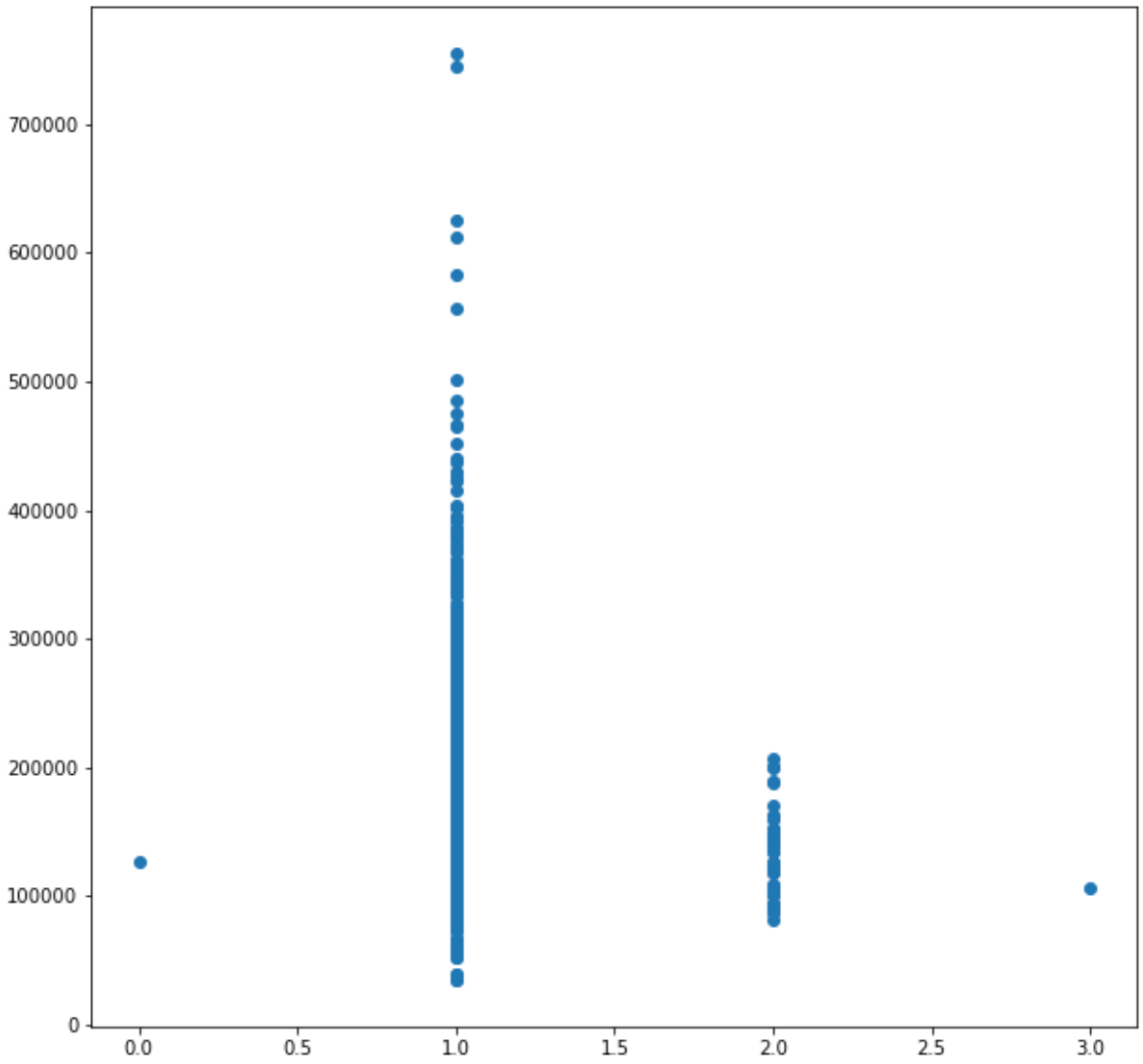
Observation:

as the number of fireplaces increases the price also increases

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['KitchenAbvGr'],df['SalePrice'])
```

Output:



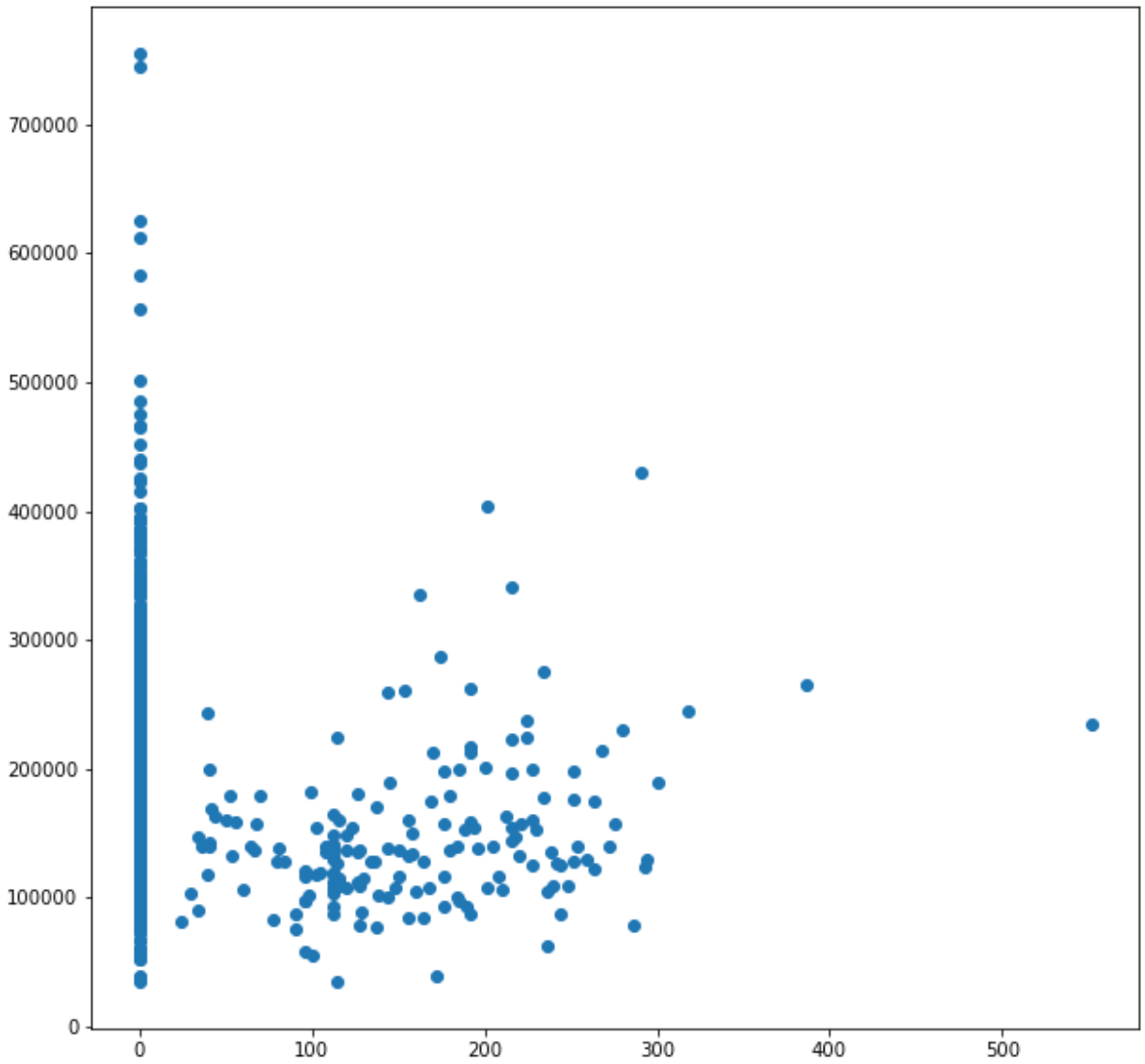
Observation:

as the kitchen above ground increases the price drops negative correlation.

Code:

```
plt.figure(figsize=(10,10))
plt.scatter(df['EnclosedPorch'],df['SalePrice'])
```

Output:



Observation:

as the enclosed porch area increases the price decreases negative correlation

- ## Interpretation of the Results

  Based on the visualizations I can see people are obsessed with size of the house no matter its living area size, garage size, or even basement size. If the size of house is big the price goes up. Even the no of fireplace, no of bathrooms also matters as people obsessed with size and number.

CONCLUSION

- Key Findings and Conclusions of the Study

  I can clearly see the correlation of size variables to price. The size variables I mean are house square feet, first floor square feet, garage area, basement square feet, No of Fireplaces, no of bathrooms. as the size of them increases price increases along with them there is strong positive correlation with size variables I mentioned above. People are obsessed with size of the house.

- Learning Outcomes of the Study in respect of Data Science

  As I mentioned earlier the size is directly proportional to Price. The best algorithm for this dataset is Ridge Regression. Although initially the testing score is 80% and Cross Validation Score is 86%. With the help of grid Search I can clock the Testing Score to 87% and Cross Validation Score to 87%

- Limitations of this work and Scope for Future Work

  The Limitations are some of the Categorical data may have ordinal relationship but we considered they don't have ordinal relationship so in future someone with in depth industry knowledge may distinguish the data and encode them