FLIP ROBO

House Price Prediction Using Machine Learning Techniques

Submitted by:

HINAL SETH

ACKNOWLEDGMENT

I would like to thank for Flip Robo Technologies for giving me this golden opportunity to work on this project.

I also would to thank my mentor, Ms. Swati Mahaseth for her guidance, suggestions, and patience toward my project.

In addition, I would like to thank my family for their supports.

Lastly, I would like to thank The Almighty for making me whoever I am today.

INTRODUCTION

- ## Business Problem Framing

The connection between house costs and the economy is a significant propelling factor at foreseeing house costs. There is no precise proportion of house costs. A property's estimation is significant in land exchanges. House costs patterns are not just the worries for purchasers and venders, but they additionally demonstrate the current monetary circumstances. In this way, it is critical to anticipate the house costs without predisposition to assist both purchasers and venders with settling on their choices, i.e., without bias to help both buyers and sellers make their decisions.

There are different AI/Machine Learning calculations to anticipate the house price.

I am expected to assemble a model involving Machine Learning to foresee the genuine worth of the planned properties and choose whether to contribute in them or not.

For this I also want to know :

i) Which variables are important to predict the price of the house ?
ii) How do these variables describe the price of the house ?

Target Variable:

Selling Price of the house.

- ## Conceptual Background of the Domain Problem

Houses are one of the necessary need 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.

In this study we can have two clients :

i) Client House Buyer : This client needs to observe their next dream home with a sensible sticker price. They have their areas of interest prepared. Presently, they need to know whether the house cost matches the house estimation. With this review, they can comprehend which highlights (ex. Number of restrooms, area, and so forth) impact the last cost of the house. Assuming everything matches, they can guarantee that they are getting a fair cost.

ii) Client House Seller : Consider the normal house-flipper. This client needs to exploit the elements that impact a house value the most. They commonly need to purchase a house at a low cost and contribute on the highlights that will give the best yield. For instance, purchasing a house at a decent area yet little area. The client will contribute on making rooms at a little expense to get an enormous return.

- Review of Literature

House is one of human existence's most fundamental requirements, alongside other key necessities like food, water, and significantly more. Interest for houses developed quickly throughout the years as individuals' expectations for everyday comforts gotten to the next level. While there are individuals who make their home as a venture and property, yet the vast majority all over the planet are purchasing a house as their asylum/shelter or as their business.

Real estate markets emphatically affect a nation's money, which is a significant public economy scale. Mortgage holders will buy merchandise like furnishings and family hardware for their home, and homebuilders or workers for hire will buy unrefined substance to construct houses to fulfill house interest, which means that the financial wave impact made by the new house supply. Other than that, purchasers have cash-flow to make an enormous speculation, and the development business is in great condition should be visible through a nation's significant degree of house supply.

Housing prices are an important reflection of the economy, and housing price ranges are of great interest for both buyers and sellers. In this project. house prices will be predicted given explanatory variables that cover many aspects of residential houses. As continuous house prices, they will be predicted with various regression techniques including Linear Regression, Ridge, Elastic Net, SGD Regressor, Decision Tree regression, and

Random Forest regression, etc.

The data contains 1460 entries each having 81 variables, hence we may also use PCA for dimension reduction.

- Motivation for the Problem Undertaken

The benefit of this study that we can have two clients, Buyer and Seller.

i) Objective: To build a model to effectively predict the house price.

There are many advantages that home purchasers, property financial backers, and house manufacturers can harvest from the house-value model. This model will give a great deal of data and information to home purchasers, property financial backers and house manufacturers, for example, the valuation of house costs in the current market, which will assist them with deciding house costs. In the mean time, this model can assist possible purchasers with choosing the attributes of a house they need as indicated by their financial plan. Past investigations zeroed in on breaking down the characteristics that influence house cost and anticipating house cost in light of the model of AI independently. Be that as it may, this article consolidates such a both anticipating house cost and characteristics together.

ii) Motivation: House is significantly established in the monetary, monetary, and political construction of every country. All things considered, announced that the vacillation of house costs has forever been an issue for house proprietors, structures and land, other than expressed that house has become unreasonably expensive as there is significant value development in a few nations in the lodging area. Inhabitants' personal satisfaction as well as public economy relies upon the potential house cost increment. Eventually, this issue will influence financial backers who are making their home as a speculation.
An increment in house request happens every year, in a roundabout way causing house cost expands each year. The issue emerges when there are various factors, for example, area and property request that might impact the house value, along these lines most partners including purchasers and designers, house manufacturers and the land business might want to know the specific ascribes or the precise variables affecting the house cost to assist financial backers with simply deciding and assist with lodging developers set the house cost.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  The data contains 1460 entries each having 81 variables, hence we may use PCA for dimension reduction.

  Since the target is continuous variable, we will use regression algorithms like Linear Regression, Lasso, Ridge, Elastic Net, KNeighbours Regression, Search Vector Machine, SGD Regression, Decision Tree, Random Forest, Gradient Boosting Regression, etc.

- Data Sources and their formats

  Training Dataset: The preparation information comprises of 1,168 instances of houses with 81 highlights portraying each part of the house. We are given sale prices (labels) for each house. The preparation information is what we will use to "instruct" our models.

  Testing Dataset: The test informational collection comprises of 292 models with similar number of highlights as the preparation information. Our test informational collection prohibits the deal cost since this is the thing we are attempting to anticipate. When our models have been constructed we will run the best one on the test dataset.

- Data Preprocessing Done

  Generally speaking, machine learning projects follow the same process. Data ingestion, data cleaning, exploratory data analysis, feature engineering and finally machine learning.

  I started by removing duplicates from the data, checked for missing or NaN (not a number) values. It's important to check for NaNs (and not just because it's socially moral) because these cause errors in the machine learning models.

There are a lot of categorical variables that are marked as N/A when a feature of the house is nonexistent. For example, when no alley is present. I identified all the cases where this was happening across the training and test data and replaced the N/As with something more descriptive. N/As can cause errors with machine learning later down the line so get rid of them.

The data consisted of many null values, outliers and skewness. I have dropped all the features with 60%+ missing values. Removed the outliers for training dataset having zscore more than 4 and treated the skewness of all numerical columns using log transformation and power transformation techniques.

- About Data and Features

About the columns:

1. 'Id': It gives the ID of the house

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

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

190 2 FAMILY CONVERSION - ALL STYLES AND AGES

3. 'MSZoning':Identifies the general zoning classification of the sale.
   A Agriculture
   C Commercial
   FV Floating Village Residential
   I Industrial
   RH Residential High Density
   RL Residential Low Density
   RP Residential Low Density Park
   RM Residential Medium Density

4. 'LotFrontage': Linear feet of street connected to property

5. 'LotArea': Lot size in square feet

6. 'Street': Type of road access to property
   Grvl Gravel
   Pave Paved

7. 'Alley': Type of alley access to property
   Grvl Gravel
   Pave Paved
   NA No alley access

8. 'LotShape':General shape of property
   Reg Regular
   IR1 Slightly irregular
   IR2 Moderately Irregular
   IR3 Irregular

9. 'LandContour': General shape of property
   Reg Regular
   IR1 Slightly irregular
   IR2 Moderately Irregular

IR3 Irregular

10. 'Utilities': Type of utilities available

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

11. 'LotConfig': Lot configuration

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

12. 'LandSlope': Slope of property

    Gtl Gentle slope
    Mod Moderate Slope
    Sev Severe Slope

13. 'Neighborhood': Physical locations within Ames city limits

    Blmngtn Bloomington Heights
    Blueste Bluestem
    BrDale Briardale
    BrkSide Brookside
    ClearCr Clear Creek
    CollgCr College Creek
    Crawfor Crawford
    Edwards Edwards
    Gilbert Gilbert
    IDOTRR Iowa DOT and Rail Road
    MeadowV Meadow Village
    Mitchel Mitchell
    Names North Ames

NoRidge	Northridge
NPkVill	Northpark Villa
NridgHt	Northridge Heights
NWAmes	Northwest Ames
OldTown	Old Town
SWISU	South & West of Iowa State University
Sawyer	Sawyer
SawyerW	Sawyer West
Somerst	Somerset
StoneBr	Stone Brook
Timber	Timberland
Veenker	Veenker

14. 'Condition1': Proximity to various conditions

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

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

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

RRNe	Within 200' of East-West Railroad
RRAe	Adjacent to East-West Railroad

16. 'BldgType': Type of dwelling

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

17. 'HouseStyle': Style of dwelling

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

18. 'OverallQual': Rates the overall material and finish of the house

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

19. 'OverallCond': Rates the overall condition of the house

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

20. 'YearBuilt': Original construction date

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

22. 'RoofStyle': Type of roof

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

23. 'RoofMatl': Roof material

ClyTile Clay or Tile
CompShg Standard (Composite) Shingle
Membran Membrane
Metal Metal
Roll Roll

Tar&Grv	Gravel & Tar
WdShake	Wood Shakes
WdShngl	Wood Shingles

24. 'Exterior1st': Exterior covering on house

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

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

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

MetalSd	Metal Siding
Other	Other
Plywood	Plywood
PreCast	PreCast
Stone	Stone
Stucco	Stucco
VinylSd	Vinyl Siding
Wd Sdng	Wood Siding
WdShing	Wood Shingles

26. 'MasVnrType': Masonry veneer type

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

27. 'MasVnrArea': Masonry veneer area in square feet

28. 'ExterQual': Evaluates the quality of the material on the exterior

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

29. 'ExterCond': Evaluates the present condition of the material on the exterior

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair

Po Poor

30. 'Foundation': Type of foundation

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

31. 'BsmtQual': Evaluates the height of the basement

Ex	Excellent (100+ inches)
Gd	Good (90-99 inches)
TA	Typical (80-89 inches)
Fa	Fair (70-79 inches)
Po	Poor (<70 inches
NA	No Basement

32. 'BsmtCond': Evaluates the general condition of the basement

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

33. 'BsmtExposure': Refers to walkout or garden level walls

Gd	Good Exposure
Av	Average Exposure (split levels or foyers typically score average or above)
Mn	Mimimum Exposure

No	No Exposure
NA	No Basement

34. 'BsmtFinType1': Rating of basement finished area

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

35. 'BsmtFinSF1': Type 1 finished square feet

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

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

37. 'BsmtFinSF2': Type 2 finished square feet

38. 'BsmtUnfSF': Unfinished square feet of basement area

39. 'TotalBsmtSF': Total square feet of basement area

40. 'Heating': Type of heating

Floor	Floor Furnace

GasA Gas forced warm air furnace
GasW Gas hot water or steam heat
Grav Gravity furnace
OthW Hot water or steam heat other than gas
Wall Wall furnace

41. 'HeatingQC': Heating quality and condition

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

42. 'CentralAir': Central air conditioning

N No
Y Yes

43. 'Electrical': Electrical system

SBrkr Standard Circuit Breakers & Romex
FuseA Fuse Box over 60 AMP and all Romex wiring (Average)

FuseF 60 AMP Fuse Box and mostly Romex wiring (Fair)
FuseP 60 AMP Fuse Box and mostly knob & tube wiring (poor)
Mix Mixed

44. '1stFlrSF': First Floor square feet

45. '2ndFlrSF': Second floor square feet

46. 'LowQualFinSF': Low quality finished square feet (all floors)

47. 'GrLivArea': Above grade (ground) living area square feet

48. 'BsmtFullBath': Basement full bathrooms

49. 'BsmtHalfBath': Basement half bathrooms

50. 'FullBath': Full bathrooms above grade

51. 'HalfBath': Half baths above grade

52. 'BedroomAbvGr': Bedrooms above grade (does NOT include basement bedrooms)

53. 'KitchenAbvGr': Kitchens above grade

54. 'KitchenQual': Kitchen quality

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

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

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

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

Sal Salvage only

57. 'Fireplaces': Number of fireplaces

58. 'FireplaceQu': Fireplace quality

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

59. 'GarageType': Garage location

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

60. 'GarageYrBlt': Year garage was built

61. 'GarageFinish': Interior finish of the garage

Fin Finished
RFn Rough Finished
Unf Unfinished
NA No Garage

62. 'GarageCars': Size of garage in car capacity

63. 'GarageArea': Size of garage in square feet

64. 'GarageQual': Garage quality

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

65. 'GarageCond': Garage condition

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

66. 'PavedDrive': Paved driveway

Y Paved
P Partial Pavement
N Dirt/Gravel

67. 'WoodDeckSF': Wood deck area in square feet

68. 'OpenPorchSF': Open porch area in square feet

69. 'EnclosedPorch': Enclosed porch area in square feet
70. '3SsnPorch': Three season porch area in square feet
71. 'ScreenPorch': Screen porch area in square feet

72. 'PoolArea': Pool area in square feet
73. 'PoolQC': Pool quality

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

74. 'Fence': Fence quality

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

75. 'MiscFeature': Miscellaneous feature not covered in other categories

Elev Elevator
Gar2 2nd Garage (if not described in garage section)
Othr Other
Shed Shed (over 100 SF)
TenC Tennis Court
NA None

76. 'MiscVal': (dollar)Value of miscellaneous feature
77. 'MoSold': Month Sold (MM)
78. 'YrSold': Year Sold (YYYY)

79. 'SaleType': Type of sale

WD Warranty Deed - Conventional
CWD Warranty Deed - Cash

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

80. 'SaleCondition': Condition of sal

81. 'SalePrice': Selling price of the house.
It is out target variable.

- ## Hardware and Software Requirements and Tools Used

I used Python and Jupyter notebooks for the project building.

Libraries: These are frameworks in python to handle commonly required tasks.

i) Pandas – For handling structured data
ii) Numpy – For linear algebra and mathematics
iii) Scikit Learn – For Machine Learning
iv) Seaborn – For data visualization.
v) Matplotlib – For data visualization

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Both Lasso and Ridge use the concepts of coefficients of Regression model to rank the feature importance. The higher the coefficients, the more important the features. Both work well when data is in linear shape and not too many noisy data exist. Since our data do not have many noisy data exist, we assume the Lasso and Ridge will work good. Lasso and Ridge are regularization models. Lasso is L1 Regularization. It adds penalty to the loss function with a term $\alpha\sum|wi|$. The weak features have zero coefficients in Lasso model. Increase alpha parameter in Lasso function will produce more zeros in the coefficient.

  Random Forest is a calculation worked with numerous decision trees. Each hub is a highlight condition. Sklearn has RandomForestRegressor() with built-in highlight significance work. In the wake of fitting with RandomForestRegressor(), we can call feature importances to get the significance score for each compo

  -nent. The higher the score, the more significant the highlight is.

- Testing of Identified Approaches (Algorithms)

  I have used Following algorithms in my project:

  i) Linear Regression
  ii) Ridge
  iii) Elastic Net
  iv) SGD Regressor
  v) K Neighbours Regressor
  vi) Decision Tree Regressor
  vii) Random Forest Regressor
  viii) Gradient Boosting Regressor

- Run and Evaluate selected models

i) Linear Regression:

Linear regression is used to predict the relationship between two variables by applying a linear equation to observed data. There are two types of variable, one variable is called an independent variable, and the other is a dependent variable. Linear regression is commonly used for predictive analysis. The main idea of regression is to examine two things. First, does a set of predictor variables do a good job in predicting an outcome (dependent) variable? The second thing is which variables are significant predictors of the outcome variable ?.

1. Linear Regression

```
In [35]: xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.3, random_state = 5 )
lr= LinearRegression()
lr.fit(xtrain,ytrain)
lr.coef_
pred_train=lr.predict(xtrain)
pred_test=lr.predict(xtest)
print('Linear Regression Score:',lr.score(xtrain,ytrain))
print('Linear Regression r2_score:',r2_score(ytest,pred_test))
print("Mean squared error of Linear Regression:",mean_squared_error(ytest,pred_test))
print("Root Mean Square error of Linear Regression:",np.sqrt(mean_squared_error(ytest,pred_test)))
```

```
Linear Regression Score: 0.8470502825605732
Linear Regression r2_score: 0.8581493069513298
Mean squared error of Linear Regression: 775129481.9495882
Root Mean Square error of Linear Regression: 27841.147281489462
```

The accuracy of Linear Regression is 85.81% which is quite good.

ii) Ridge:

The Ridge regression is a procedure which is particular to investigate multiple regression data which is multicollinearity in nature.

2. Ridge

```
In [36]: from sklearn.linear_model import Ridge, ElasticNet
ridge = Ridge(alpha = 0.5)
ridge.fit(xtrain, ytrain)
pred_test_r= ridge.predict(xtest)
print('Ridge Regression Score:',ridge.score(xtrain,ytrain))
print('Ridge Regression r2_score:',r2_score(ytest,pred_test_r))
print("Mean squared error of Ridge Regression:",mean_squared_error(ytest,pred_test_r))
print("Root Mean Square error of Ridge Regression:",np.sqrt(mean_squared_error(ytest,pred_test_r)))
```

```
Ridge Regression Score: 0.8470501202143325
Ridge Regression r2_score: 0.8581727907666389
Mean squared error of Ridge Regression: 775001156.8973557
Root Mean Square error of Ridge Regression: 27838.842592632252
```

The accuracy of Ridge Regression is 85.81%, same as Linear Regression.

iii) Elastic Net:

Sklearn provides a linear model named **ElasticNet** which is trained with both L1, L2-norm for regularisation of the coefficients. The advantage of such combination is that it allows for learning a sparse model where few of the weights are non-zero like Lasso regularisation method, while still maintaining the regularization properties of Ridge regularisation method.

3. Elastic Net

In [37]:
```
en = ElasticNet(alpha = 0.01)
en.fit(xtrain, ytrain)
pred_test_en= en.predict(xtest)
print('ElasticNet Regression Score:',en.score(xtrain,ytrain))
print('ElasticNet Regression r2_score:',r2_score(ytest,pred_test_en))
print("Mean squared error of ElasticNet Regression:",mean_squared_error(ytest,pred_test_en))
print("Root Mean Square error of ElasticNet Regression:",np.sqrt(mean_squared_error(ytest,pred_test_en)))
```

```
ElasticNet Regression Score: 0.8470427260250485
ElasticNet Regression r2_score: 0.8583010995581436
Mean squared error of ElasticNet Regression: 774300025.8351724
Root Mean Square error of ElasticNet Regression: 27826.24706702599
```

The accuracy of ElasticNet Regression is 85.83%, same as Linear and Ridge Regression.

iv) SGD Regressor:

SGD represents Stochastic Gradient Descent: the inclination of the misfortune is assessed each example at a time and the model is refreshed en route with a diminishing strength plan (otherwise known as learning rate).

The regularizer is a punishment added to the shortfall work that psychologists model boundaries towards the zero vector utilizing either the squared euclidean standard L2 or the outright standard L1 or a blend of both (Elastic Net). On the off chance that the boundary update crosses the 0.0 worth as a result of the regularizer, the update is shortened to 0.0 to take into consideration learning inadequate models and accomplish online component determination.

This execution works with information addressed as thick numpy varieties of drifting point values for the highlights.

4. SGD Regressor

In [38]:
```
sgd=SGDRegressor()
sgd.fit(xtrain,ytrain)
pred_train_sgd=sgd.predict(xtrain)
pred_test_sgd=sgd.predict(xtest)
print('SGD Regressor Score:',sgd.score(xtrain,ytrain))
print('SGD Regressor r2_score:',r2_score(ytest,pred_test_sgd))
print("Mean squared error of SGD Regressor:",mean_squared_error(ytest,pred_test_sgd))
print("Root Mean Square error of SGD Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_sgd)))
```

```
SGD Regressor Score: 0.846357845480139
SGD Regressor r2_score: 0.8562163476396343
Mean squared error of SGD Regressor: 785691952.3732618
Root Mean Square error of SGD Regressor: 28030.197151880002
```

The accuracy of SGDRegressor is 85.62%

v) KNeighbours Regressor:

KNeighborsRegressor. The K for the sake of this regressor addresses the k closest neighbors, where k is a whole number worth determined by the client. Subsequently, as the name recommends, this regressor carries out learning in light of the k closest neighbors. The decision of the worth of k is reliant upon information.

5. K-Neighbors Regressor

In [39]:
```
knr = KNeighborsRegressor()
knr.fit(xtrain,ytrain)
pred_train_knr=knr.predict(xtrain)
pred_test_knr=knr.predict(xtest)
print('K Neighbors Regressor Score:',knr.score(xtrain,ytrain))
print('K Neighbors Regressor r2_score:',r2_score(ytest,pred_test_knr))
print("Mean squared error of K Neighbors Regressor:",mean_squared_error(ytest,pred_test_knr))
print("Root Mean Square error of K Neighbors Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_knr)))
```

```
K Neighbors Regressor Score: 0.8858618586776874
K Neighbors Regressor r2_score: 0.855375812005502
Mean squared error of K Neighbors Regressor: 790284978.5802021
Root Mean Square error of K Neighbors Regressor: 28112.007729441917
```

The accuracy of KNeighbors Regressor is 85.53%

vi) Decision Tree Regressor:

Decision tree regression notices highlights of an article and trains a model in the construction of a tree to anticipate information in the future to deliver significant consistent result. Persistent result implies that the result/result isn't discrete, i.e., it isn't addressed just by a discrete, known arrangement of numbers or values.

6. Decision Tree Regressor

In [40]:
```
dtr=DecisionTreeRegressor(criterion='mse')
dtr.fit(xtrain,ytrain)
pred_train_dtr=dtr.predict(xtrain)
pred_test_dtr=dtr.predict(xtest)
print('Decision Tree Regressor Score:',dtr.score(xtrain,ytrain))
print('Decision Tree Regressor r2_score:',r2_score(ytest,pred_test_dtr))
print("Mean squared error of Decision Tree Regressor:",mean_squared_error(ytest,pred_test_dtr))
print("Root Mean Square error of Decision Tree Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_dtr)))
```

```
Decision Tree Regressor Score: 1.0
Decision Tree Regressor r2_score: 0.7824081215082161
Mean squared error of Decision Tree Regressor: 1189009911.9494948
Root Mean Square error of Decision Tree Regressor: 34482.02302576655
```

The accuracy of Decision Tree Regressor is 78.24%

vii) Random Forest Regressor:

Random forest regression is an **ensemble learning technique**. In ensemble learning, we take different algorithms or same algorithm on numerous occasions and set up a model that is more impressive than the first.

7. Random Forest Regressor

In [41]:
```
rf=RandomForestRegressor()
rf.fit(xtrain,ytrain)
pred_train_rf=rf.predict(xtrain)
pred_test_rf=rf.predict(xtest)
print('Random Forest Regressor Score:',rf.score(xtrain,ytrain))
print('Random Forest Regressor r2_score:',r2_score(ytest,pred_test_rf))
print("Mean squared error of Random Forest Regressor:",mean_squared_error(ytest,pred_test_rf))
print("Root Mean Square error of Random Forest Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_rf)))
```

```
Random Forest Regressor Score: 0.9793599261784415
Random Forest Regressor r2_score: 0.885237262879892
Mean squared error of Random Forest Regressor: 627109949.6179727
Root Mean Square error of Random Forest Regressor: 25042.163437250638
```

The accuracy of Random Forest Regressor is 88.52% which is really very good.

viii) Gradient Boosting Regressor:

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions as each predictor corrects its predecessor's error.

8. Gradient Boosting Regressor

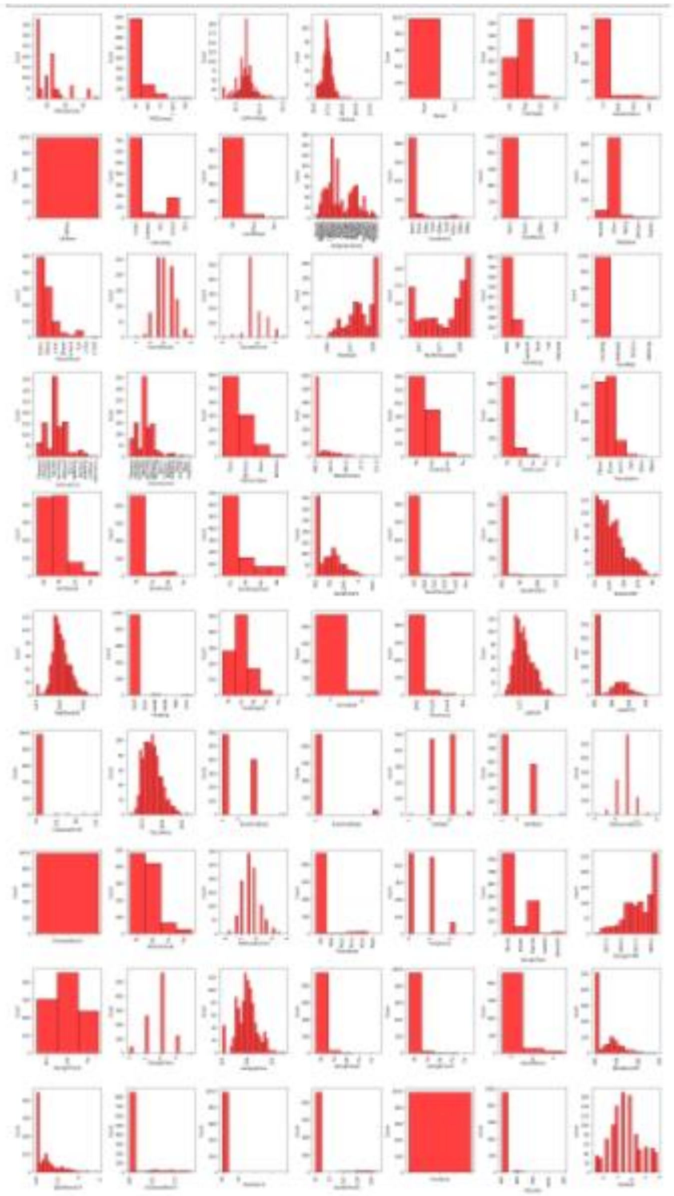
In [42]:
```
from sklearn.ensemble import GradientBoostingRegressor
gb=RandomForestRegressor()
gb.fit(xtrain,ytrain)
pred_train_gb=gb.predict(xtrain)
pred_test_gb=gb.predict(xtest)
print('Gradient Boosting Regressor Score:',gb.score(xtrain,ytrain))
print('Gradient Boosting Regressor r2_score:',r2_score(ytest,pred_test_gb))
print("Mean squared error of Gradient Boosting Regressor:",mean_squared_error(ytest,pred_test_gb))
print("Root Mean Square error of Gradient Boosting Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_gb)))
```

```
Gradient Boosting Regressor Score: 0.9803416144519056
Gradient Boosting Regressor r2_score: 0.88722865697388848
Mean squared error of Gradient Boosting Regressor: 616228167.9413441
Root Mean Square error of Gradient Boosting Regressor: 24823.94344058462
```

The accuracy of Gradient Boosting Regressor is 88.72% which is highest of all.

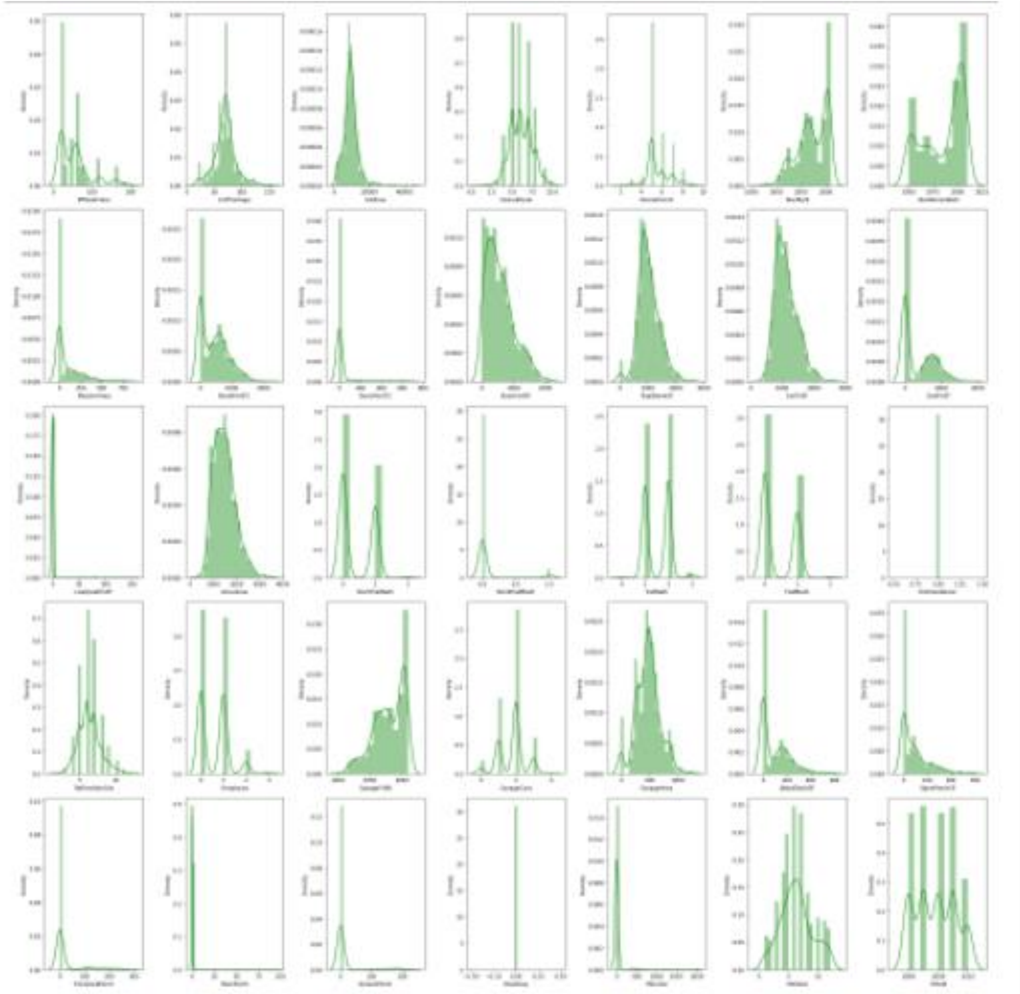
- Visualizations

i) Histogram:



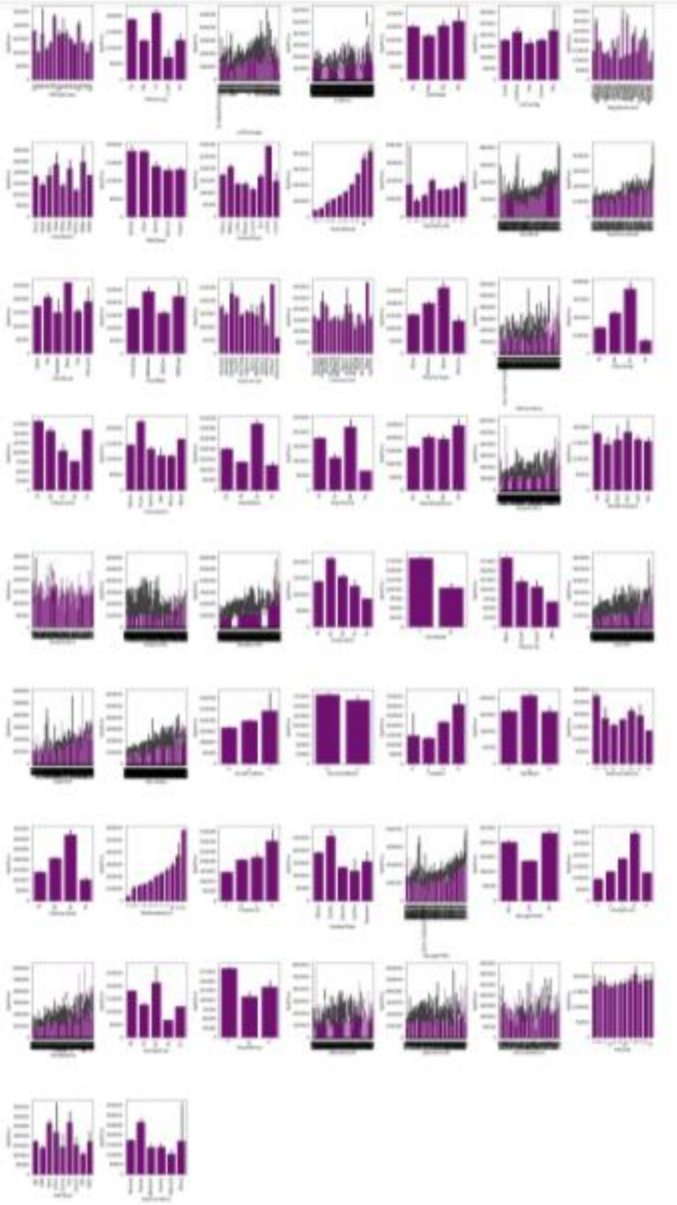
We have observed from the above histogram plot that the columns like 'Street', 'Utilities', 'LotContour', 'KitchenAbvGr', '3SsnPorch', 'PoolArea', and 'YrSold' have least contribution in the data prediction as they contains of almost only same values in them.

ii) Distribution plot:



It tells about the distribution of data.

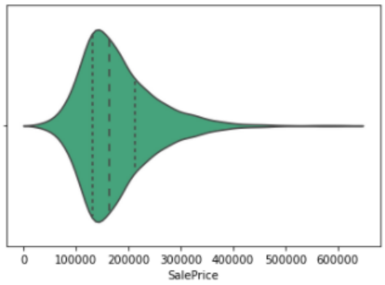
iii) Bar Graph:



Bar graph gives the relationship between the target variable and all the features.

iv) Violin Plot:

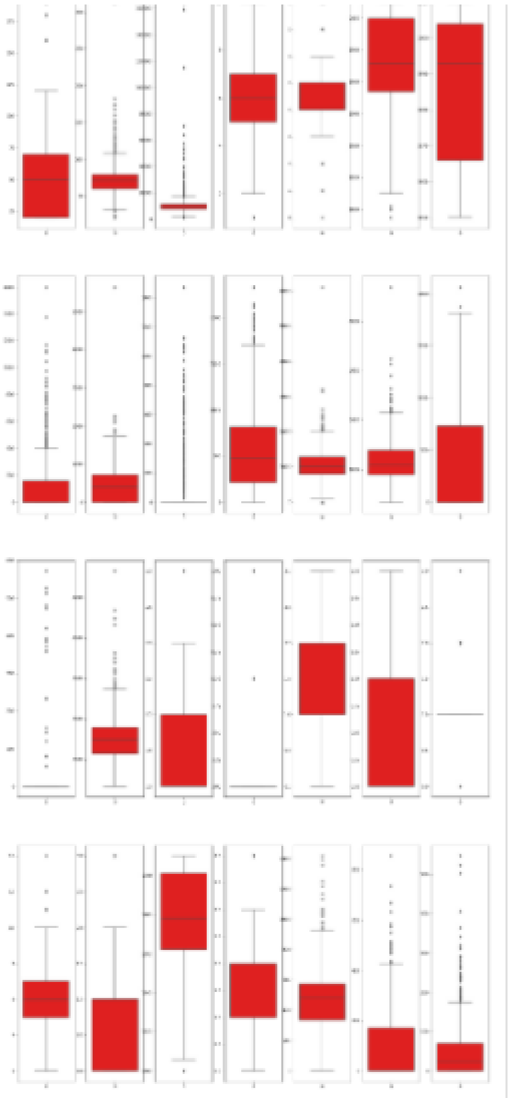
```
In [24]: sn.violinplot(x=dftrain['SalePrice'], inner="quartile", color="#36B37E")

Out[24]: <AxesSubplot:xlabel='SalePrice'>
```



Our dataset contains a lot of variables, but the most important one for us to explore is the target variable. We need to understand its distribution. First, we start by plotting the violin plot for the target variable. The width of the violin represents the frequency.

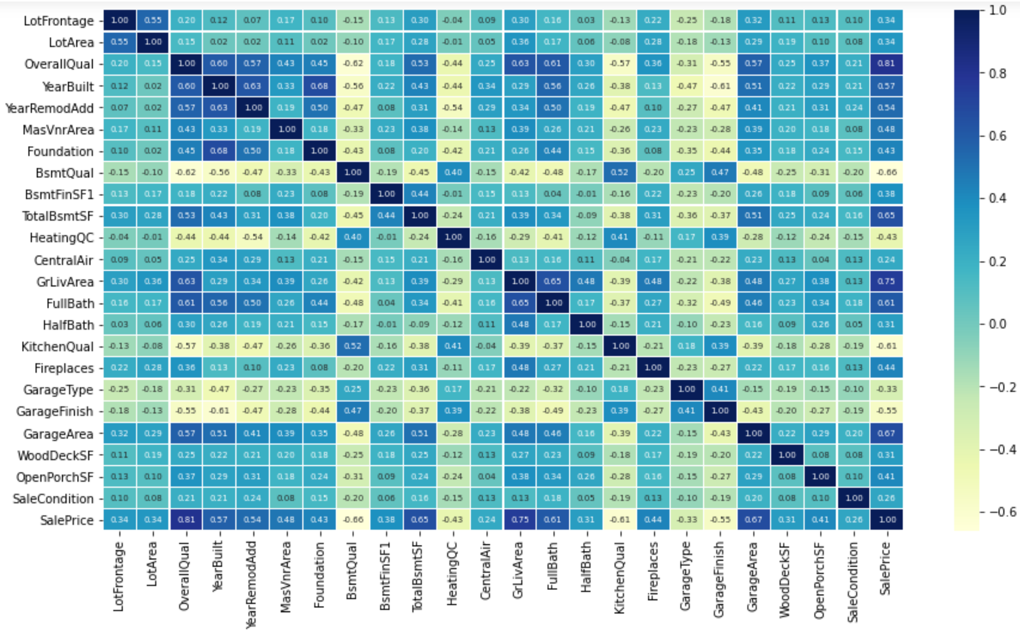
We can see from the plot that most house prices fall between 100,000 and 250,000. The dashed lines represent the locations of the three quartiles Q1, Q2 (the median), and Q3.

v) Boxplot:



Boxplot tells us about the quartiles, interquartile range and his work over there.

vi) Heatmap:



Heatmap gives the correlation between the features and feature-target.

With that, we have finished the Data Visualization process. Our next step is to select and define the dependent variables and the independent variables and split them into a train set and test set.

- Interpretation of the Results

Give a summary of what results were interpreted from the visualizations, preprocessing and modelling.

After visualization and preprocessing, we have reduced the dimension of the dataset effectively. We have all the columns that does not contribute in prediction of target, or which has the least correlation with the target or having the maximum multicollinearity with other features.

CONCLUSION

- ## Key Findings and Conclusions of the Study

  Here, we constructed serveral relapse models to anticipate the cost of some house given some of the house highlights. We eveluated and contrasted each model with decide the one with most noteworthy execution. We additionally saw how a few models rank the highlights as indicated by their significance.

  In this project, we followed the information science process beginning with getting the information, then, at that point, cleaning what's more preprocessing the information, trailed by investigating the information and building models, then, at that point, assessing the outcomes and discussing them with representations

- ## Learning Outcomes of the Study in respect of Data Science

  We have used 8 different algorithms and all are working very fine.
  The Score of each algorithms is above 0.8 0r 80%.
  The r2 Score (Accuracy) of each model is 85%+ except for Decision Tree Classifier (It has only 78% accuracy).
  Coming to the case of choosing best model, Random Forest Regressor turned out to be the one being most accurate compared from others followed by Gradient Boosting Regressor.
  The accuracy of Random Forest is 89% with a score of 0.98 which can be used to solve real world problems easily.

- ## Limitations of this work and Scope for Future Work

  What might be more fascinating in my view is; on the off chance that we could add second layer to the model result or might be second step where results from this model are taken care of into second model which would then conjecture region house costs a half year, year and a half, etc into what's to come. This would permit the open door not exclusively to anticipate the house costs yet additionally to see what's on the horizon at the house costs. Furthermore this is by and large the sort of bits of knowledge Real Estate Investment groups need to make right speculations.

As a suggestion, I encourage to utilize this model (or an adaptation of it prepared with later information) by individuals who need to purchase a house in the space covered by the dataset to have a thought regarding the genuine cost. The model can be utilized additionally with datasets that cover various urban communities and regions given that they contain similar highlights. I additionally recommend that individuals think about the highlights that were considered as most significant as found in the past segment; this may help them gauge the house value better.