House Price Prediction

: NIMESH KUMAR ROY

Nimesh Kumar Roy

ACKNOWLEDGMENT

(1) www.researchgate.net

(2) S. Abhishek.:Ridge regression vs Lasso,How these two popular ML Regression techniques work. Analytics India magazine,2018.

(3) Hands -on Machine Learning with Scikit-Learn,Keras,Tensorflow

2nd Edition.

(4) Video tutorials from "DATA TRAINED"

(5) https://www.datacamp.com/

(6) https://www.upgrad.com/

Problem Statement :-

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. Our problem is related to one such housing company.

A US-based housing company named **Surprise Housing** has decided to enter the Australian market. The company uses

data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same

purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file

below.

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?

By the help of machine learning model we can predict the price of a particular area from particular data which helps the people to decide the buying strategy .

Predictive models for determining the sale price of houses in cities is still remaining as more challenging and tricky task. The sale price of properties depends on a number of interdependent factors.

Key factors that might affect the price include area of the

property, location of the property and its amenities . The data set has 81 features. In this project an attempt has been made to construct a predictive model for evaluating the price based on the factors that affect the price.

Modeling explorations apply some regression techniques such as linear regression, Lasso and Ridge

regression models, KNeighbor Regressor,Random Forest Regressor,regression, and boosting algorithms such as Extreme Gradient Boost Regression (XG Boost). Such models are used to build a predictive model, and to

pick the best performing model by performing a comparative

analysis on the predictive errors obtained between these models.

Here, the attempt is to construct a predictive model for

evaluating the price based on factors that affects the price.

Linear regression is one of the statistical techniques for assessing the relationship between the (dependent) target variable and several independent variables. Regression techniques are widely used to build a model based on several factors to predict price. We have considered five prediction models, they are ordinary least squares model, Lasso and Ridge regression models, KNeighbors, Random Forest Regressor , Decision Tree Regressor and XGBoost regression model. Once we get a good fit, we can use the model to forecast monetary value of that particular housing property .

Motivation for the Problem Undertaken -

As a common man every person's dream is he/she has their own house, but the problem is how to found right area and right price with good amenities. This project is only a step to help people to get their dream home at predicted price .

DATA UNDERSTANDING AND PRE-PROCESSING

Data Description *:*

The two data sets-train set and test data considered in the project . It consists of features that describe house-property . There are 81 features in both the data sets. The features can be explained as follows:

(1) 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

(2) 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

(3) LotFrontage: Linear feet of street connected to property

(4) LotArea: Lot size in square feet

(5) Street: Type of road access to property

Grvl Gravel

Pave Paved

(6) Alley: Type of alley access to property

Grvl Gravel

Pave Paved

NA No alley access

(7) LotShape: General shape of property

Reg Regular

IR1 Slightly irregular

IR2 Moderately Irregular

IR3 Irregular

(8) LandContour: Flatness of the property

Lvl Near Flat/Level

Bnk Banked - Quick and significant rise from street grade to building

HLS Hillside - Significant slope from side to side

Low Depression

(9) 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

(10) 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

(11) LandSlope: Slope of property

Gtl Gentle slope

Mod Moderate Slope

Sev Severe Slope

(12) Neighborhood: Physical locations within Ames city limits

Blmngtn Bloomington Heights

Bluestem	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

(13) 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

(**14) 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

(15) 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

(16) 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

(17) 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

(18) 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

(19) YearBuilt: Original construction date

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

(21) RoofStyle: Type of roof

Flat Flat

Gable Gable

Gambrel Gabrel (Barn)

Hip Hip

Mansard Mansard

Shed Shed

(22) 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

(23) 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

(24) 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

(25) MasVnrType: Masonry veneer type

BrkCmn Brick Common

BrkFace Brick Face

CBlock Cinder Block

None None

Stone Stone

(26) MasVnrArea: Masonry veneer area in square feet

(27) ExterQual: Evaluates the quality of the material on the exterior

Ex Excellent

Gd Good

TA Average/Typical

Fa Fair

Po Poor

(28) ExterCond: Evaluates the present condition of the material on the exterior

Ex Excellent

Gd Good

TA Average/Typical

Fa Fair

Po Poor

(29) Foundation: Type of foundation

BrkTil Brick & Tile

CBlock Cinder Block

PConc Poured Contrete

SlabSlab

Stone Stone

Wood Wood

(30) 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

(31) 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

(32) 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

(33) 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

(34) BsmtFinSF1: Type 1 finished square feet

(35) 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

(36) BsmtFinSF2: Type 2 finished square feet

(37) BsmtUnfSF: Unfinished square feet of basement area

(38) TotalBsmtSF: Total square feet of basement area

(39) 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

(40) HeatingQC: Heating quality and condition

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

(41) CentralAir: Central air conditioning

N No

Y Yes

(42) 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

(43) 1stFlrSF: First Floor square feet

(44) 2ndFlrSF: Second floor square feet

(45) LowQualFinSF: Low quality finished square feet (all floors)

(46) GrLivArea: Above grade (ground) living area square feet

(47) BsmtFullBath: Basement full bathrooms

(48) BsmtHalfBath: Basement half bathrooms

(49) FullBath: Full bathrooms above grade

(50) HalfBath: Half baths above grade

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

(52) Kitchen: Kitchens above grade

(53) KitchenQual: Kitchen quality

Ex Excellent

Gd Good

TA Typical/Average

Fa Fair

Po Poor

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

(55) 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

(56) Fireplaces: Number of fireplaces

(57) 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

(58) 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

(59) GarageYrBlt: Year garage was built

(60) GarageFinish: Interior finish of the garage

Fin Finished

RFn Rough Finished

Unf Unfinished

NA No Garage

(61) GarageCars: Size of garage in car capacity

(62) GarageArea: Size of garage in square feet

(63) GarageQual: Garage quality

Ex Excellent

Gd Good

TA Typical/Average

Fa Fair

Po Poor

NA No Garage

(64) GarageCond: Garage condition

Ex Excellent

Gd Good

TA Typical/Average

Fa Fair

Po Poor

NA No Garage

(65) PavedDrive: Paved driveway

Y Paved

P Partial Pavement

N Dirt/Gravel

(66) WoodDeckSF: Wood deck area in square feet

(67) OpenPorchSF: Open porch area in square feet

(68) EnclosedPorch: Enclosed porch area in square feet

(69) 3SsnPorch: Three season porch area in square feet

(70) ScreenPorch: Screen porch area in square feet

(71) PoolArea: Pool area in square feet

(72) PoolQC: Pool quality

Ex Excellent

Gd Good

TA Average/Typical

Fa Fair

NA No Pool

(73) Fence: Fence quality

GdPrv Good Privacy

MnPrv Minimum Privacy

GdWo Good Wood

MnWw Minimum Wood/Wire

NA No Fence

(74) 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

(75) MiscVal: $Value of miscellaneous feature

(76) MoSold: Month Sold (MM)

(77) YrSold: Year Sold (YYYY)

(78) 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

(79) SaleCondition: Condition of sale

NormalNormal Sale

Abnorml Abnormal Sale - trade, foreclosure, short sale

AdjLand Adjoining Land Purchase

Alloca Allocation - two linked properties with separate deeds, typically condo with a garage unit

Family Sale between family members

Partial Home was not completed when last assessed (associated with New Homes)

Data understanding and basic EDA :-

The purpose is to create a model that can estimate housing prices. We divide the set of data into functions and target variable. In this section, we will try to understand overview of original data set, with its original features and then we will make an exploratory analysis of the data set and attempt to get useful observations. The train data set consists of 1168 records with 81 explanatory variables. In test data set, there were around 292 records with 80 variables. While building regression models we are often required to convert the categorical i.e. text features to its numeric representation. The two most common

ways to do this is to use label encoder or one hot encoder. Label encoding in python can be achieved by using sklearn library.

Label encoder encodes labels with a value between 0 and n-1. If a label repeats, it attributes the same value as previously assigned . One hot encoding refers to splitting the column that contains numerical categorical data to many columns depending on the number of categories present in that column. Each column contains “0” or “1” corresponding to which column it has been placed .

The dataset includes quite a few categorical variables (both train and test data set) for which we will need to create dummy variables or use label encoding to convert into numerical form.

These would be fake/dummy variables because they are placeholders for actual variable and are created by ourselves. Also, there are a lot of null values present as well, so we will need to treat them accordingly

The features appears as categorical variables are-

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, SaleCondition

The features appears as numerical variables are-

Id, MSSubClass, LotFrontage, LotArea, Alley, OverallQual, OverallCond YearBuilt, YearRemodAdd, MasVnrArea, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, 1stFlrSF, 2ndFlrSF, LowQualFinSF,GrLivArea BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, TotRmsAbvGrd, Fireplaces, GarageYrBlt, GarageCars GarageArea, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch ScreenPorch, PoolArea, MiscVal, MoSold, YrSold, SalePrice

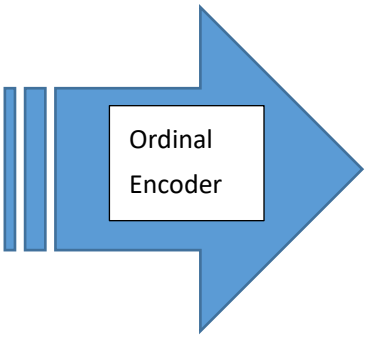
Data pre-processing :-

The general steps in data pre-processing are:

(1) Converting categorical features into numerical variables in order to fit linear regression model.

```
from sklearn.preprocessing import OrdinalEncoder
oe=OrdinalEncoder()
for i in train_cat.columns:
    if train_cat[i].dtypes=='object':
        train_cat[i]=oe.fit_transform(train_cat[i].values.reshape(-1,1))
```

MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
RL	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl
RL	Pave	NaN	IR1	Lvl	AllPub	Inside	Mod
RL	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl
RL	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl
RL	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl
...	...	...	...	...	...	...	...
RL	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl
RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl
RL	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl
C (all)	Pave	Pave	Reg	Lvl	AllPub	Inside	Gtl



MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
3.0	1.0	0.0	0.0	3.0	0.0	4.0	0.0
3.0	1.0	0.0	0.0	3.0	0.0	4.0	1.0
3.0	1.0	0.0	0.0	3.0	0.0	1.0	0.0
3.0	1.0	0.0	0.0	3.0	0.0	4.0	0.0
3.0	1.0	0.0	0.0	3.0	0.0	2.0	0.0
...	...	...	...	...	...	...	...
3.0	1.0	0.0	0.0	3.0	0.0	4.0	0.0
3.0	1.0	0.0	3.0	3.0	0.0	4.0	0.0
3.0	1.0	0.0	3.0	3.0	0.0	2.0	0.0
0.0	1.0	1.0	3.0	3.0	0.0	4.0	0.0
3.0	1.0	0.0	0.0	3.0	0.0	4.0	0.0

(2) Imputing null records with appropriate values.

Random Sample Imputation :-

This Imputation technique can be used for both the Numerical and Categorical variables.The process of Random Sampling involves replacing the missing values by taking a random observation from the pool of available observations for the variable.

Here, we have not limited to only one(1) Random observation, rather take as many random observations as the number of missing values for the variable.

Thus, by doing so we can ensure that the mean and standard deviation of the variables is not changed (preserved) for Numerical variables. Whereas for the Categorical variable, the frequency of the categories is preserved(not changed).

Another thing we can keep a note of here is that, since the variable distribution is preserved, we can use this method for Linear models.

To use this method for Missing Data At Random, we need to use a missing data indicator in combination with this technique.

Assumptions :-

- The data is Missing Completely At Random(MCAR).

- We replace the missing values with the values having the same distribution of original variables.
- The probability of selecting the value is dependent on its frequency, i.e. higher the frequency of a value higher the probability of selecting it. Thus, the variance & distribution of variables is preserved.
- Missing values are not more than 5% of the complete dataset.

Advantages :-

- It is easy to implement.
- We can get the complete dataset very quickly.
- The variance & distribution of variables is preserved.

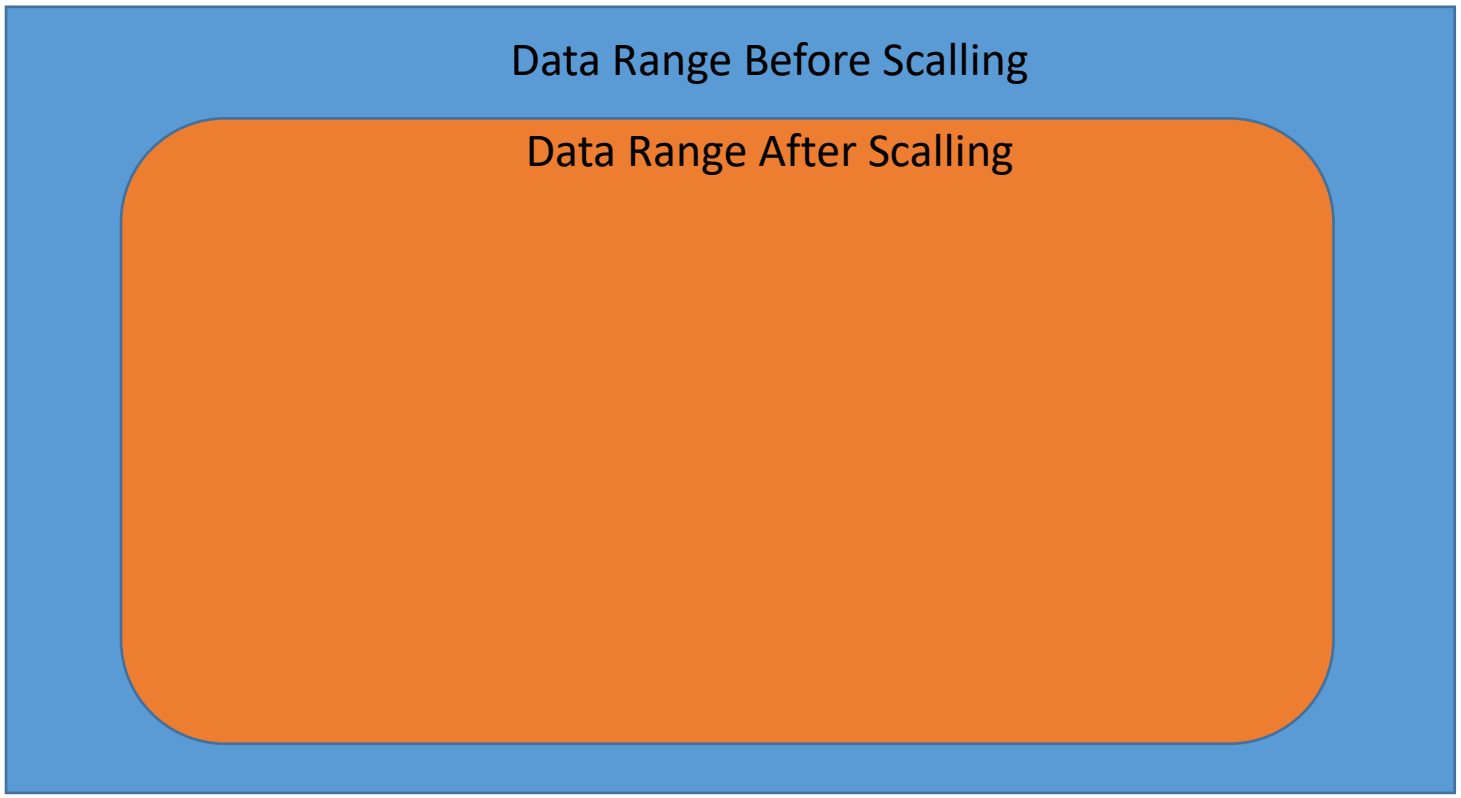
Limitations :-

- Randomness -- Since the value of missing data is selected at random, there is always a chance of getting different values for the same observation. This can be controlled by using "seed" during the process.
- In case we have more missing values, then the relationship of the imputed variable with other variables may be affected.
- For extracting the values for Test Set we need to store the train set in memory, as the missing values should always be

replaced by the values of the Train set only. Thus, in the case of huge data sets, it becomes a memory-intensive operation.

(3) Scaling of data .

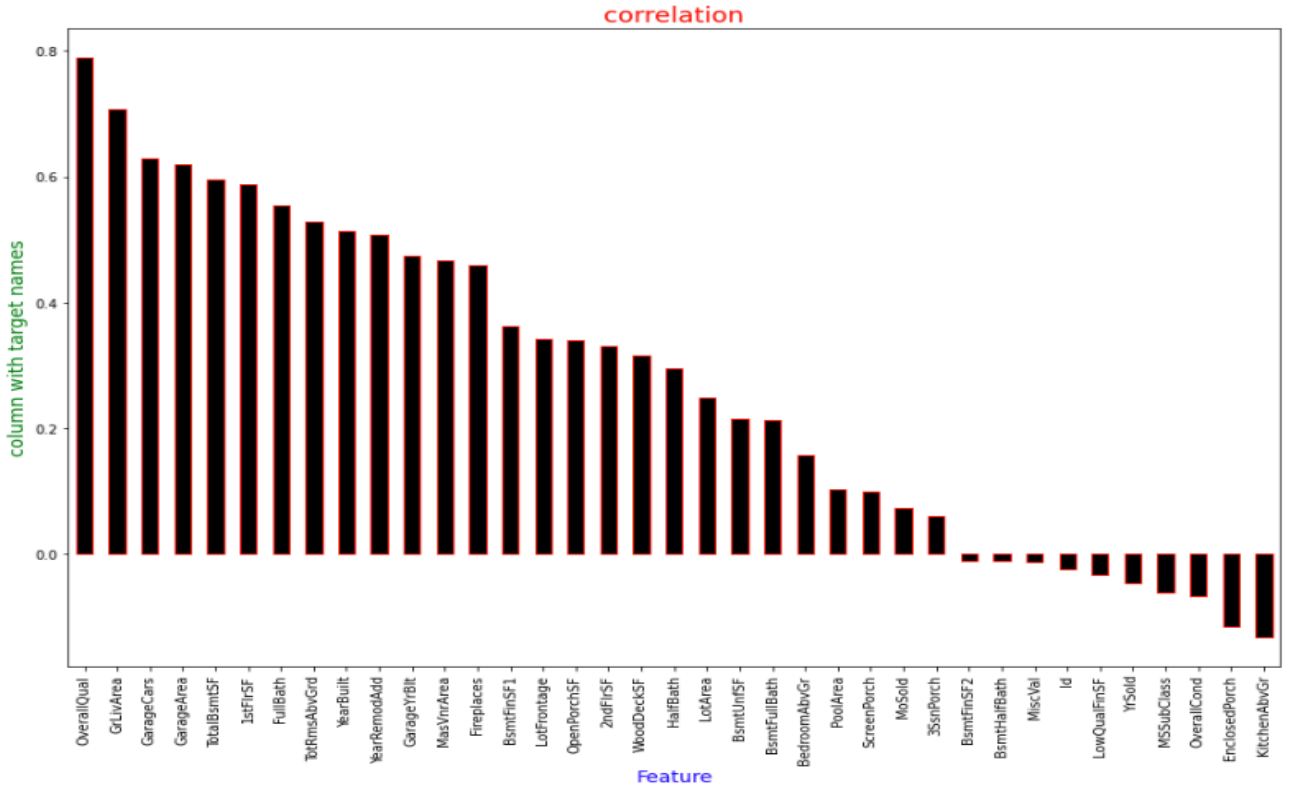
```
scaler=MinMaxScaler()
x=pd.DataFrame(scaler.fit_transform(x),columns=x.columns)
x
```

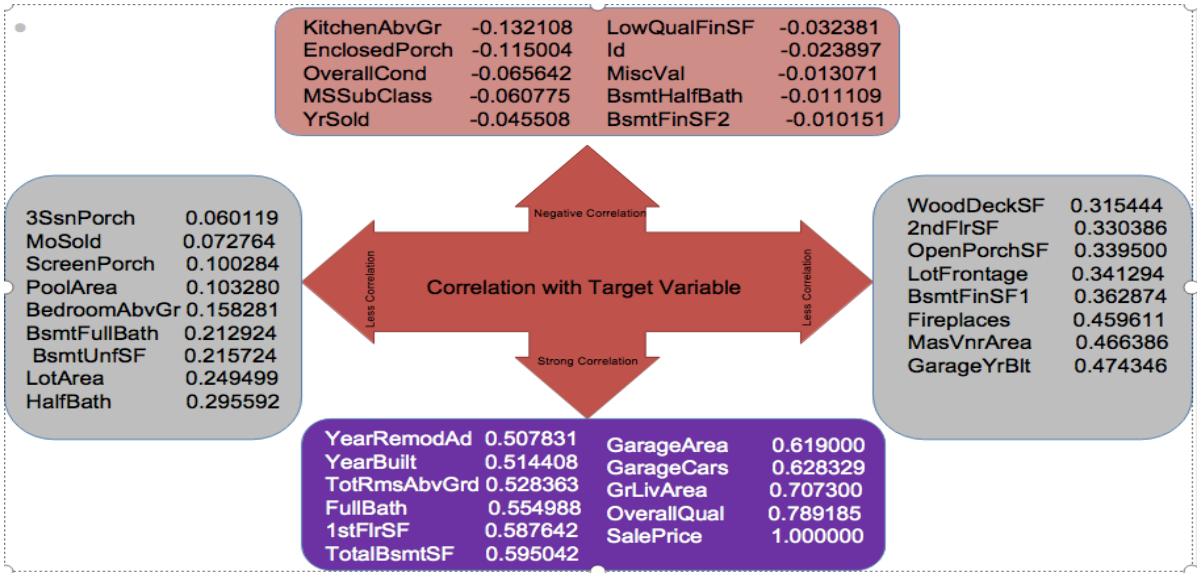


both the data sets as it doesn't add much to the

For each value in a feature, MinMaxScaler **subtracts the minimum value in the feature and then divides by the range**. The range is the difference between the original maximum and original minimum. MinMaxScaler preserves the shape of the original distribution

(4) Correlation with target Variable :-





We can see that varaibles

KitchenAbvGr, EnclosedPorch, OverallCond, MSSubClass, YrSold, Id MiscVal, BsmtHalfBath, BsmtFinSF2 .

have negative correlation with target variable i.e. these variables give negative impact on “Sale Price”.

Whereas variables

YearRemodAdd, YearBuilt, TotRmsAbvGrd, FullBath, TotalBsmtSF, GarageArea, GarageCars, GrLivArea, OverallQual .

Have strong positive correlation with target varaiable i.e. these variable give positive imapct on “Sale Price”.

(5) Treating Outliers :-

An outlier is an observation that is unlike the other observations. It is rare, or distinct, or does not fit in some way.*We will generally define outliers as samples that are exceptionally far from the mainstream of the data.*

We treat the outliers with-

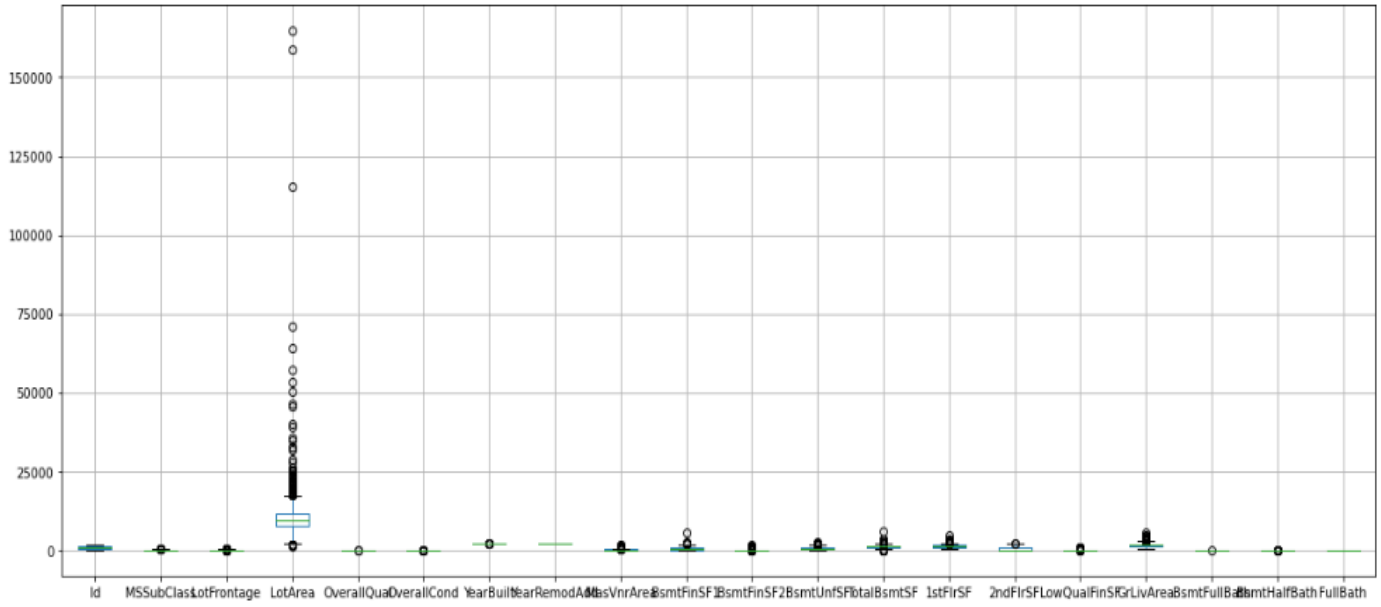
Interquartile Range Method

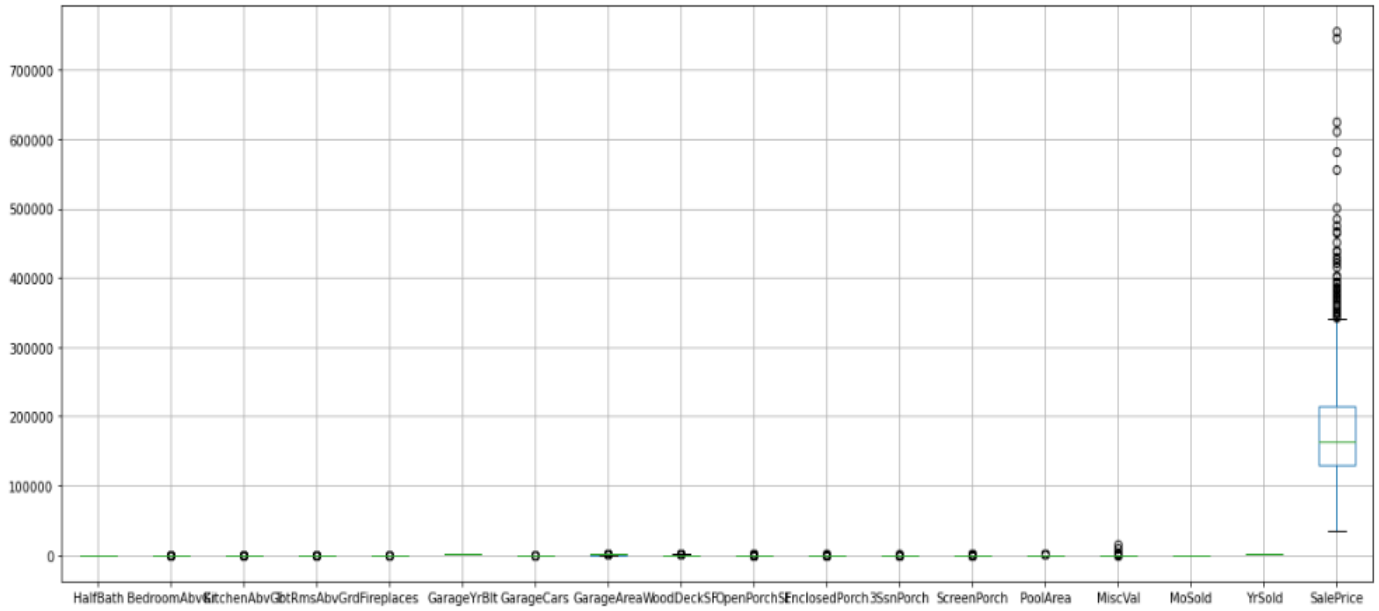
Not all data is normal or normal enough to treat it as being drawn from a Gaussian distribution.

A good statistic for summarizing a non-Gaussian distribution sample of data is the Interquartile Range, or IQR for short.

The IQR is calculated as the difference between the 75th and the 25th percentiles of the data and defines the box in a box and whisker plot.

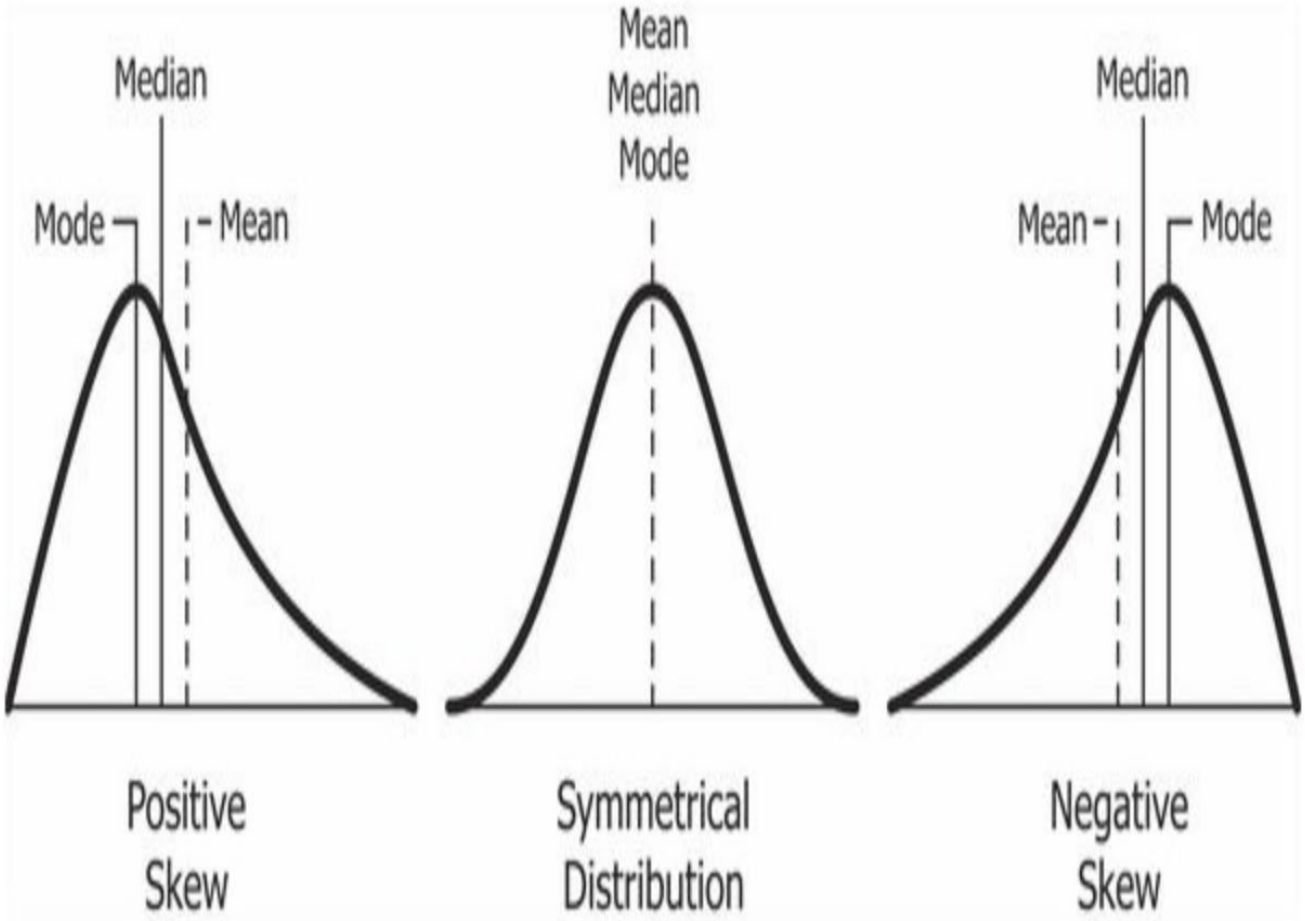
On a box and whisker plot, these limits are drawn as fences on the whiskers (or the lines) that are drawn from the box. Values that fall outside of these values are drawn as dots.





(6) Removing the Skewness :-

skewness is the measure of how much the probability distribution of a random variable deviates from the normal distribution.



it is better to transform the skewed data to normally distributed data.

We have seen heavy skewness on some variables and we treat them with “Power Transform”.

Skewness before power transform -

```
1 x.skew()
```

```
Id                0.026526
MSSubClass        1.016094
LotFrontage       0.086094
LotArea           0.184194
OverallQual       0.175082
                    ...
PoolQC          -10.887630
Fence            -3.185107
MiscFeature     -17.238424
SaleType         -3.660513
SaleCondition    -2.671829
Length: 80, dtype: float64
```

Skewness after power transform

```
from sklearn.preprocessing import power_transform
x_tran=power_transform(x)
x=pd.DataFrame(x_tran,columns=x.columns)
```

```
x.skew()
```

```
Id              -0.268486
MSSubClass       0.040499
LotFrontage     -0.023819
LotArea         -0.002361
OverallQual      0.021658
                    ...
PoolQC           5.155070
Fence            1.116688
MiscFeature      9.291637
SaleType        -2.067563
SaleCondition   -0.353292
Length: 80, dtype: float64
```

(7) After Scaling PCA applied to dataset :-

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

the idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible.

Before applying PCA -

```
1  x.shape
```

```
(1168, 80)
```

After applying PCA -

```
1  pca=PCA(0.95)
2  xpca=pca.fit_transform(x)
3  xpca.shape
```

```
(1168, 47)
```

(8) Model Selections :-

(A) Linear Regression .

(B) Decision Tree Regressor.

(C) Random Forest Regressor.

(D) KNeighbor Regressor.

(E) XGBoost Regressor.

(A) Linear Regression :-

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

This form of analysis estimates the coefficients of the linear equation, involving one or more independent variables that best predict the value of the dependent variable. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values. There are simple linear regression calculators that use a “least squares” method to discover the best-fit line for a set of paired data. You then estimate the value of X (dependent variable) from Y (independent variable).
In this project our we evaluated model’s performance using R2 score.

```
max_r_score=0
r_state=0
for i in range(1,200):
    x_train_pca,x_test_pca,y_train,y_test=train_test_split(xpca,y,test_size=0.25,random_sta
    reg=LinearRegression()
    reg.fit(x_train_pca,y_train)
    y_pred=reg.predict(x_test_pca)
    r2_scr=r2_score(y_test,y_pred)
    if r2_scr>max_r_score:
        max_r_score=r2_scr
        r_state=1
print('Max r2 score :-', max_r_score, "on random state :-",r_state)
```

```
Max r2 score :- 0.7642564994161046 on random state :- 1
```

Lasso Regression -

LASSO means least absolute shrinkage, and the selection operator is an LR technique that also regularizes functionality. It is identical to ridge regression, except that it varies in the values of regularisation. The absolute values of the sum of regression coefficients are taken into consideration. It even sets the coefficients to zero so it completely reduces the errors. So selection of features are resulted by lasso regression. In the previously mentioned ridge equation, the component ' e ' has absolute values instead of squared values .

It is to be noted that computationally Lasso regression technique is far more intensive than Ridge regression technique.

```
ls=Lasso(alpha=10,random_state=0)
ls.fit(x_train_pca,y_train)
ls_score_training=ls.score(x_train_pca,y_train)
pred_ls=ls.predict(x_test_pca)
ls_score_training*100

```

```
80.71078617657844
```

(B) Decision Tree Regressor :-

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which

corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

```
depth=list(range(3,30))
param_grid=dict(max_depth=depth)
tree=GridSearchCV(DecisionTreeRegressor(),param_grid,cv=10)
tree.fit(x_train_pca,y_train)
```

```
GridSearchCV(cv=10, estimator=DecisionTreeRegressor(),
             param_grid={'max_depth': [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
                                       15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
                                       25, 26, 27, 28, 29]})
```

```
y_train_pred=tree.predict(x_train_pca)
y_test_pred=tree.predict(x_test_pca)
```

```
r2_score(y_train.values,y_train_pred)
```

```
0.7447404685606169
```

```
treescore=cross_val_score(tree,xpca,y,cv=5)
treec=treescore.mean()
print('Cross Validation Score :', treec*100)
```

```
Cross Validation Score : 62.328909662879674
```

(C) Random Forest Regressor :-

Random forest is an ensemble of decision trees. This is to say that many trees, constructed in a certain “random” way form a Random Forest. Each tree is created from a different sample of rows and at each node, a different sample of features is selected for splitting. Each of the trees makes its own individual prediction. These predictions are then averaged to produce a single result.

The averaging makes a Random Forest better than a single Decision Tree hence improves its accuracy and reduces overfitting.

A prediction from the Random Forest Regressor is an average of the predictions produced by the trees in the forest.

```
parameters={'criterion':['mse','mae'],'max_features':['auto','sqrt','log2']}
rf=RandomForestRegressor()
clf=GridSearchCV(rf,parameters)
clf.fit(x_train_pca,y_train)
print(clf.best_params_)
```

```
{'criterion': 'mae', 'max_features': 'auto'}
```

```
from sklearn.model_selection import cross_val_score
rf=RandomForestRegressor(criterion='mae',max_features='auto')
rf.fit(x_train_pca,y_train)
rf.score(x_train_pca,y_train)
pred_decision=rf.predict(x_test_pca)
rfs=r2_score(y_test,pred_decision)
print('R2 Score :', rfs*100)
rfscore=cross_val_score(rf,xpca,y,cv=5)
rfc=rfscore.mean()
print('Cross Validation Score :', rfc*100)
```

```
R2 Score : 71.39893543047012
Cross Validation Score : 75.90491249318563
```

(D) KNeighbor Regressor :-

K-Nearest Neighbors regressor. This non-parametric regression method keeps track of the last window_size training samples. Predictions are obtained by aggregating the values of the closest n_neighbors stored-samples with respect to a query sample.

```
k_range=list(range(1,30))
params=dict(n_neighbors=k_range)
knn_regressor=GridSearchCV(KNeighborsRegressor(),params,cv=10,scoring='neg_mean_squared_err
knn_regressor.fit(x_train_pca,y_train)
```

```
GridSearchCV(cv=10, estimator=KNeighborsRegressor(),
             param_grid={'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
                                         13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
                                         23, 24, 25, 26, 27, 28, 29]},
             scoring='neg_mean_squared_error')
```

```
y_train_pred=knn_regressor.predict(x_train_pca)
y_test_pred=knn_regressor.predict(x_test_pca)
```

```
r2_score(y_train.values, y_train_pred)
```

```
0.7668626110862068
```

```
knnscore=cross_val_score(knn_regressor,xpca,y,cv=5)
knnc=knnscore.mean()
print('Cross Validation Score :', knnc*100)
```

```
Cross Validation Score : -189959562492.4212
```

(E) XGBoost Regressor :-

XGBoost stands for extreme gradient boosting which is most efficient technique for either regression or classification problem. It is decision tree based algorithm that make use of gradient boosting framework. It provides the features that greatly have impact on performance of model.

This technique helps in developing a model that have less variance and more stability. In addition, the execution speed is fast when compared to other algorithms.

```
import xgboost as xgb
from xgboost import XGBRegressor
```

```
xg_reg = xgb.XGBRegressor(objective ='reg:linear', colsample_bytree = 0.3, learning_rate = 0.1, max_depth = 5, alpha = 10, n_estimators = 10)

xg_reg.fit(x_train_pca,y_train) preds = xg_reg.predict(x_test_pca)

mse = mean_squared_error(y_test, preds) print("RMSE: %f" % (mse))
```

```
xgbr=XGBRegressor(verbosity=0)
print(xgbr)
```

```
XGBRegressor(base_score=None, booster=None, colsample_bylevel=None,
             colsample_bynode=None, colsample_bytree=None,
             enable_categorical=False, gamma=None, gpu_id=None,
             importance_type=None, interaction_constraints=None,
             learning_rate=None, max_delta_step=None, max_depth=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=100, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, reg_alpha=None, reg_lambda=None,
             scale_pos_weight=None, subsample=None, tree_method=None,
             validate_parameters=None, verbosity=0)
```

```
xgbr.fit(x_train_pca,y_train)
score=xgbr.score(x_train_pca,y_train)
print("Training Score :",score)
```

```
Training Score : 0.999996892365239
```

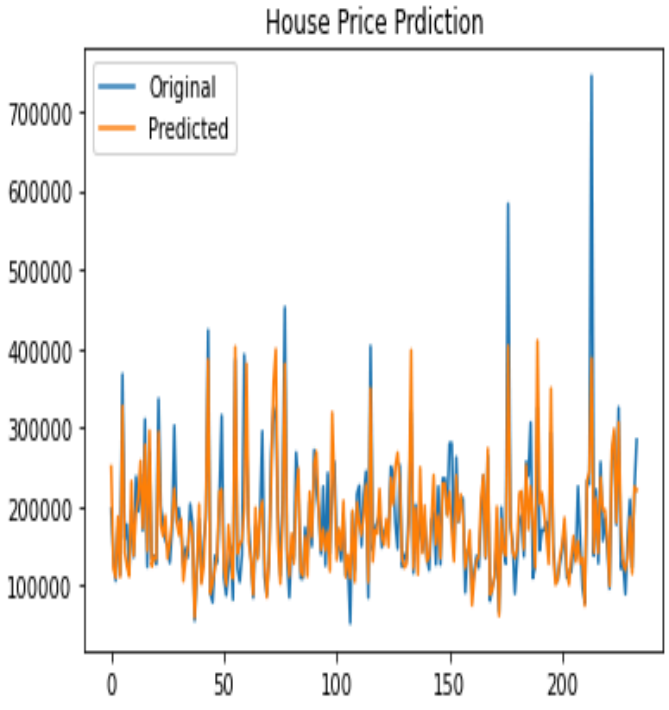
```
scoret=xgbr.score(x_test_pca,y_test)
print("Testing Score :",scoret)
```

```
Testing Score : 0.7281065856993663
```

```
cv_score=cross_val_score(xgbr,x_train_pca,y_train,cv=10)
print("CV mean score :", cv_score.mean())
```

```
CV mean score : 0.7831118690758827
```

```
x_ax=range(len(y_test))
plt.plot(x_ax,y_test,label='Original')
plt.plot(x_ax,ypred,label='Predicted')
plt.title('House Price Prdiction')
plt.legend()
plt.show()
```



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

On reviewing the data we can say that not all the variables present in the data are equally related with the target variable. Some has Negative effect and some has Positive effect, whereas some has no effect on target variable.Although the data covered most of the area which affects the price of the house but some factors like local environment,locality,human impact,weather conditions etc. Like many other factors still not covered in the data.

On working in this project I have learned so many things as a newcomer in this field like how to do proper data study, How to visualize the data, proper way of EDA, Solution findings of the problems, how to chose proper algorithms for better prediction of result. Still I am not perfect in these things but working on this project I can say that I can do better than this in upcoming projects.

MODEL APPLICABILITY :-

It is necessary to check before deciding whether the built model should or should not be used in a real-world setting .The data has been collected in different conditions and by different people. So, it is very much essential to look into the relevancy of data today. We can't say that the characteristics present in the data set are sufficient to describe house prices.

The dataset considered is quite wast in nature and there are a lot of features, like the presence of pool or not, parking lot and others, that remain very relevant when considering a house price. The property has to be categorized either as a flat or villa or independent house. Data collected from a big urban city would not be applicable in a rural city, as for equal value of feature prices, which will be comparatively higher in the urban area.

CONCLUSIONS AND FUTURE SCOPE :-

An optimal model does not necessarily represent a robust model. A model that frequently use a learning algorithm that is not suitable for the given data structure. Sometimes the data itself might be too noisy or it could contain too few samples to enable a model to accurately capture the target variable which implies that the model remains fit.

When we observe the evaluation metrics obtained for advanced regression models, we can say both behave in a similar manner. We can choose either one for house price prediction compared to basic model. With the help of box plots, we can check for outliers. If present, we can remove outliers and check the model's performance for improvement.

We can build models through advanced techniques namely random forests, neural networks, and particle swarm optimization to improve the accuracy of predictions.