HOUSING: PRICE PREDICTION

Submitted by:

ASHIKA YASMEEN

ACKNOWLEDGMENT

It gives me a great pleasure in presenting the project report on House Price Prediction Model Using Machine Learning.

References:

Wikipedia

Google.com

https://www.diva-portal.org/smash/get/diva2:1456610/FULLTEXT01.pdf

https://m2pi.ca/project/2020/bc-financial-services-authority/BCFSA-final.pdf

INTRODUCTION

Buying a house is undoubtedly one of the most important decisions of everyone's life. Houses are one of the necessary needs of each and every person around the globe. The price of a house may depend on a variety of factors ranging from house location, its features, property demand, house price and supply in the real estate.

Business Problem:

While purchasing the house, the price of the house is the main factor. The pricing of the house not only depends on the size of the property and number of rooms, but also on the factors like neighbourhood, transportation, banks, schools/colleges, shops etc. Housing is also a part of investment.

House price forecasting is an important topic of real estate. Real estate market is one of the markets which is one of the major contributions 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.

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. Here we build a model using Machine Learning in order to predict the actual value of the properties and decide whether to invest in them or not.

Background:

Machine learning is a subfield of Artificial Intelligence (AI) that works with algorithms and technologies to extract useful information from data. Machine learning methods are appropriate in big data since attempting to manually solve big data would be difficult without the support of machines. Machine learning attempts to solve problems algorithmically rather than purely mathematically. Therefore, it is based on creating algorithms that permit the machine to learn. Machine learning is categorized into two groups, which are supervised and unsupervised.

Supervised machine learning is an approach that is defined by its use of labeled datasets. These datasets are designed to train algorithms into classifying data or predicting outcomes accurately. Supervised learning can be separated into two types, classification and regression.

Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns in data without any human intervention. Unsupervised machine learning models are used for three main tasks, clustering, association and dimensionality reduction.

The performance will be measured upon predicting house prices since the prediction in many regression algorithms relies not only on a specific feature but on an unknown number of attributes that result in the value to be predicted. House prices depend on an individual house specification. Houses have different number of features that may not have the same cost due to its location. A big house may have a higher price if it is located in rich area.

The data used in this project will be handled by using a combination of pre-processing methods to improve the predictions accuracy.

Literature:

Literature survey is the most important in any kind of research. Most of the literature study is based on articles from websites. This literature study attempts to construct a model based on regression techniques, regularization and ensemble methods, and how it can be applied to predict house price.

The literature study gives an overview of the articles that are related to this study, the feature engineering methods and the evaluation metrics that are used to measure the performance of the algorithms.

This paper presents how to make optimal use of Linear Regression, Lasso and Ridge Regression, Decision Tree Regression, KNeighbors Regression, Support Vector Regressor, SGD Regressor, Random Forest Regressor, Ada Boost Regressor and Gradient Boosting Regressor. The system will satisfy customers by providing accurate output and preventing the risk of investing in the wrong house.

Motivation:

Objective: The objective of this project is to predict the house price, to calculate house price depending on area, surrounding environment like school/college, bank, ATM's, shops, hospitals etc. To provide comparison of house pricing to customers.

Motivation: The customer invests huge amount into house they don't have idea about the price of house at the certain area. So, sometimes they give extra money to builder for same construction and same area. But they don't know about the two different prices of the model. Or, in some cases, builder can't predict the price of the current area and therefore he may have loss in his business. So, this project is developed for price prediction.

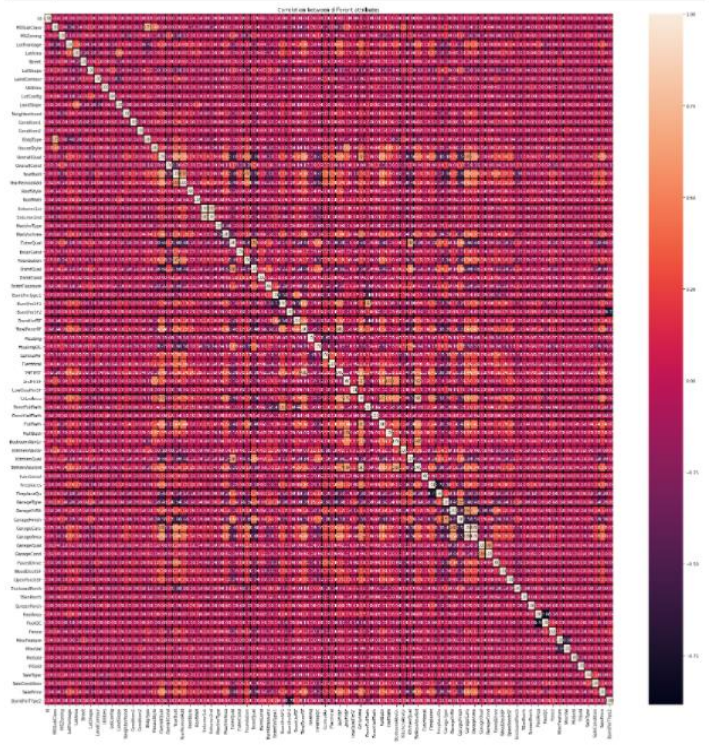
ANALYTICAL PROBLEM FRAMING

Mathematical/Analytical Modeling:

This experiment is done to pre-process the data and evaluate the prediction accuracy of the models. The experiment has multiple stages that are required to get the prediction results. These stages can be defined as:

- Pre-Processing: The dataset will be checked and pre-processed using certain methods. These methods have various ways of handling data. Thus, the pre-processing is done on multiple iterations where each time the accuracy will be evaluated with the used combination.
- Data splitting: Dividing the dataset into two parts to train the model with one and use the other in the testing. The dataset will be split 75% for training and 25% for testing.
- Evaluation: The accuracy of both datasets will be evaluated by measuring the r2 and RMSE rate when training the model with an evaluation of the actual prices on the test dataset with the prices that are being predicted by the model.
- Correlation: Correlation analysis defines the strength of a relationship between two variables, which can be between two independent variables or one independent and one dependent variable. Correlation between the available features and house price will be evaluated to identify whether the features have a negative, positive or zero correlation with the house price.

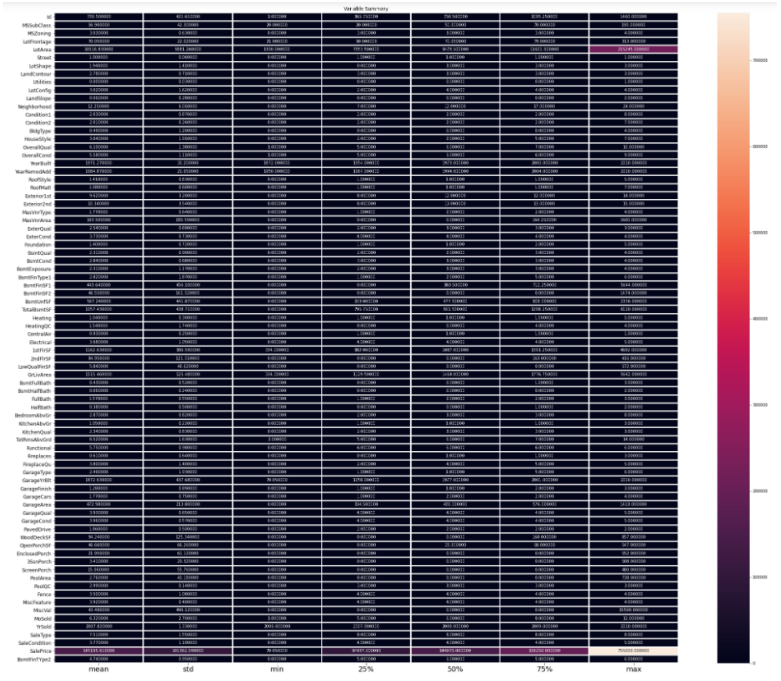
```
plt.figure(figsize=(30,30))
sns.heatmap(cor,annot=True,linewidth=1,linecolor='black',fmt='.2f')
plt.title("Correlation between different attributes")
plt.show()
```



```
cor['SalePrice']
```

```
Id             -0.036351
MSSubClass     -0.041994
MSZoning       -0.117465
LotFrontage     0.255843
LotArea         0.135165
                  ...
YrSold         -0.040265
SaleType       -0.076746
SaleCondition   0.135060
SalePrice       1.000000
```

- Describe of the dataset: Describe of the dataset is plotted using the heatmap which shows us the mean, standard deviation, maximum and minimum value of each column in the given dataset.



- Evaluation Metrics: The prediction accuracy will be evaluated by measuring the R-Squared (R2), and Root Mean Square Error (RSME) of the model used in training. R2 will show if the model is overfitted, whereas RSME shows the error percentage between the actual and predicted data, which in this case, the house prices.
- Outliers are datapoints that are far away from other similar points. They may be due to variability in the measurement or may be due to experimental errors. Outliers should be excluded from the dataset in order to get the efficient and accurate prediction.

Methods to remove outliers:

1. Z-score: Call scipy.stats.zscore() with the given data-frame as its argument to get a numpy array containing the z-score of each value in a data-frame. Call numpy.abs with the previous result to convert each element in the data-frame to its absolute value.
2. Interquartile (IQR) range: It can be used to remove outliers present in the dataframe. IQR can be calculated using scipy.stats.iqr module.

- Skewness: Skewness is a measure of the asymmetry of the probability distribution of a real valued random variable about its mean.

Skewness:

```
skw=house.skew()
skw
```

```
Id               0.000000
MSSubClass       1.407657
MSZoning        -1.735395
LotFrontage      2.384950
LotArea         12.207688
                   ...
YrSold           0.096269
SaleType        -3.868638
SaleCondition   -2.741167
SalePrice        0.715012
BsmtFinTYpe2    -3.406104
Length: 80, dtype: float64
```

Data Description:

In Machine Learning, the training data set is the actual dataset used to train the model for predicting house price.

There are 298 rows and 80 columns.

```
train.columns
```

```
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')
```

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 ATT IC 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

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

LotFrontage: Linear feet of street connected to property

LotArea: Lot size in square feet

Street: Type of road access to property

Grvl Gravel

Pave Paved

Alley: Type of alley access to property

Grvl Gravel

Pave Paved

NA No alley access

LotShape: General shape of property

Reg Regular

IR1 Slightly irregular

IR2 Moderately Irregular

IR3 Irregular

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

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

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

LandSlope: Slope of property

Gtl Gentle slope

Mod Moderate Slope

Sev Severe Slope

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

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

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

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

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

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

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

YearBuilt: Original construction date

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

RoofStyle: Type of roof

Flat Flat

Gable Gable

Gambrel Gabrel (Barn)

Hip Hip

Mansard Mansard

ShedShed

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

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

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

MasVnrType: Masonry veneer type

BrkCmn Brick Common

BrkFace Brick Face

CBlock Cinder Block

None None

Stone Stone

MasVnrArea: Masonry veneer area in square feet

ExterQual: Evaluates the quality of the material on the exterior

Ex Excellent

Gd Good

TA Average/Typical

Fa Fair

Po Poor

ExterCond: Evaluates the present condition of the material on the exterior

Ex Excellent

Gd Good

TA Average/Typical

Fa Fair

Po Poor

Foundation: Type of foundation

BrkTil Brick & Tile

CBlock Cinder Block

PConc Poured Contrete

Slab Slab

Stone Stone

Wood Wood

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

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

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

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

BsmtFinSF1: Type 1 finished square feet

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

BsmtFinSF2: Type 2 finished square feet

BsmtUnfSF: Unfinished square feet of basement area

TotalBsmtSF: Total square feet of basement area

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

HeatingQC: Heating quality and condition

Ex Excellent

Gd Good

TA Average/Typical

Fa Fair

Po Poor

CentralAir: Central air conditioning

N No

Y Yes

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

1stFlrSF: First Floor square feet

2ndFlrSF: Second floor square feet

LowQualFinSF: Low quality finished square feet (all floors)

GrLivArea: Above grade (ground) living area square feet

BsmtFullBath: Basement full bathrooms

BsmtHalfBath: Basement half bathrooms

FullBath: Full bathrooms above grade

HalfBath: Half baths above grade

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

Kitchen: Kitchens above grade

KitchenQual: Kitchen quality

Ex Excellent

Gd Good

TA Typical/Average

Fa Fair

Po Poor

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

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

Fireplaces: Number of fireplaces

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

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

GarageYrBlt: Year garage was built

GarageFinish: Interior finish of the garage

Fin Finished

RFn Rough Finished

Unf Unfinished

NA No Garage

GarageCars: Size of garage in car capacity

GarageArea: Size of garage in square feet

GarageQual: Garage quality

Ex Excellent

Gd Good

TA Typical/Average

Fa Fair

Po Poor

NA No Garage

GarageCond: Garage condition

Ex Excellent

Gd Good

TA Typical/Average

Fa Fair

Po Poor

NA No Garage

PavedDrive: Paved driveway

Y Paved

P Partial Pavement

N Dirt/Gravel

WoodDeckSF: Wood deck area in square feet

OpenPorchSF: Open porch area in square feet

EnclosedPorch: Enclosed porch area in square feet

3SsnPorch: Three season porch area in square feet

ScreenPorch: Screen porch area in square feet

PoolArea: Pool area in square feet

PoolQC: Pool quality

Ex Excellent

Gd Good

TA Average/Typical

Fa Fair

NA No Pool

Fence: Fence quality

GdPrv Good Privacy

MnPrv Minimum Privacy

GdWo Good Wood

MnWw Minimum Wood/Wire

NA No Fence

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

MiscVal: $Value of miscellaneous feature

MoSold: Month Sold (MM)

YrSold: Year Sold (YYYY)

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

SaleCondition: Condition of sale

Normal Normal 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 Pre-Processing:

Data pre-processing is an important step in machine learning to get highly accurate and reliable result. Data pre-processing helps in increasing the quality of data by filling in missing data's(NaN values), removing outliers, scaling the data.

There are many steps involved in data pre-processing:

- Data Cleaning helps to impute the missed values and removing outliers from the dataset.
- Data Integration integrates data from multiple sources into single dataset.
- Data Transformation such as normalization helps in improving the accuracy and efficiency of algorithms involved in machine learning.

- Data Reduction reduces the data size by dropping out redundant features using feature selection and feature extraction techniques.

Treating null values

Sometimes there can be certain columns which may contain the null values used to indicate the missing or unknown values. In our dataset the null values are present in LotFrontage, Alley, PoolQC and SalePrice columns.

```
house.isnull().sum()
```

```
Id                 0
MSSubClass         0
MSZoning           0
LotFrontage      259
LotArea            0
                ...
YrSold             0
SaleType           0
SaleCondition      0
SalePrice        292
source             0
Length: 82, dtype: int64
```

The null values can be filled with fill.na or by replacing with mean, median or mode.

```
house=house.replace(np.NaN,house['LotFrontage'].mean())
house=house.replace(np.NaN,house['Alley'].mode())
house=house.replace(np.NaN,house['PoolQC'].mode())
house=house.replace(np.NaN,house['SalePrice'].mean())
```

Converting labels into numeric

In machine learning, we usually deal with datasets which contain multiple labels in one or more column. These labels can be in the form of alphabets or numbers. To make the data understandable or in human readable form, the training data is often labelled in words.

In our dataset there are columns with categorical values. The columns like MSZoning, Street, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Condition2, BldgType, HouseStyle, RoofStyle, RoofMat1, Exterior1st, Exterior2nd, MasVnrType, ExterQual, ExterCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, Heating, CentralAir, Electrical, 2ndFlrSF, KitchenQual, Functional, FireplaceQu, GarageType, GarageFinish, GarageQual, GarageCond, PavedDrive, PoolQC, Fence, MiscFeature, SaleType, SaleCondition. These columns are converted using Label Encoder.

Label Encoder refers to converting the labels into numeric form so as to convert it into the machine-readable form. It is an important step in data pre-processing.

Label encoding in python can be imported from Sklearn library. Sklearn provides a very efficient tool for encoding.

Label Encoding

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

house['MSZoning']=le.fit_transform(house['MSZoning'])
house['Street']=le.fit_transform(house['Street'])
house['LotShape']=le.fit_transform(house['LotShape'])
house['LandContour']=le.fit_transform(house['LandContour'])
house['Utilities']=le.fit_transform(house['Utilities'])
house['LotConfig']=le.fit_transform(house['LotConfig'])
house['LandSlope']=le.fit_transform(house['LandSlope'])
house['Neighborhood']=le.fit_transform(house['Neighborhood'])
house['Condition1']=le.fit_transform(house['Condition1'])
house['Condition2']=le.fit_transform(house['Condition2'])
house['BldgType']=le.fit_transform(house['BldgType'])
house['HouseStyle']=le.fit_transform(house['HouseStyle'])
house['RoofStyle']=le.fit_transform(house['RoofStyle'])
house['RoofMatl']=le.fit_transform(house['RoofMatl'])
house['Exterior1st']=le.fit_transform(house['Exterior1st'])
house['Exterior2nd']=le.fit_transform(house['Exterior2nd'])
house['MasVnrType']=le.fit_transform(house['MasVnrType'])
house['ExterQual']=le.fit_transform(house['ExterQual'])
house['ExterCond']=le.fit_transform(house['ExterCond'])
house['Foundation']=le.fit_transform(house['Foundation'])
house['BsmtQual']=le.fit_transform(house['BsmtQual'])
house['BsmtCond']=le.fit_transform(house['BsmtCond'])
house['BsmtExposure']=le.fit_transform(house['BsmtExposure'])
house['BsmtFinType1']=le.fit_transform(house['BsmtFinType1'])
house['BsmtFinTYpe2']=le.fit_transform(house['BsmtFinType2'])
house['Heating']=le.fit_transform(house['Heating'])
house['HeatingQC']=le.fit_transform(house['HeatingQC'])
house['CentralAir']=le.fit_transform(house['CentralAir'])
house['Electrical']=le.fit_transform(house['Electrical'])
house['2ndFlrSF']=le.fit_transform(house['2ndFlrSF'])
house['KitchenQual']=le.fit_transform(house['KitchenQual'])
house['Functional']=le.fit_transform(house['Functional'])
house['FireplaceQu']=le.fit_transform(house['FireplaceQu'])
house['GarageType']=le.fit_transform(house['GarageType'])
house['GarageFinish']=le.fit_transform(house['GarageFinish'])
house['GarageQual']=le.fit_transform(house['GarageQual'])
house['GarageCond']=le.fit_transform(house['GarageCond'])
house['PavedDrive']=le.fit_transform(house['PavedDrive'])
house['PoolQC']=le.fit_transform(house['PoolQC'])
house['Fence']=le.fit_transform(house['Fence'])
house['MiscFeature']=le.fit_transform(house['MiscFeature'])
house['SaleType']=le.fit_transform(house['SaleType'])
house['SaleCondition']=le.fit_transform(house['SaleCondition'])
```

Input-Output Relationship:

The dependent variable or target or output depends on the features or input variables given in the dataset.

Column MSSubClass identifies the type of dwelling involved in the house sale which helps in the prediction of house price. MSZoning Identifies the general zoning classification of the sale in which the house price depends on in which zone is the house is present. LotFrontage is the linear feet of street connected to property which is helpful in predicting the house price. LotArea is the lot size in square feet which has high impact in predicting the house price. Street is the

type of road access to property. Alley is the type of alley access to property. LotShape is the general shape of property. LandContour is the flatness of the property. Utilities is the type of utilities available, the greater the utilities available, the higher will be the price of the house. LotConfig is the lot configuration. LandSlope is the slope of property. Neighborhood is the physical locations within city limits, with the good neighborhood, the higher will be the house price. Condition1 and Condition2 is the proximity to various conditions. BldgType is the type of house which effects the house price. HouseStyle is the style of house, the house price depends on it. OverallQual rates the overall material and finish of the house. OverallCond rates the overall condition of the house. YearBuilt is the original construction date, if the house is older, the lesser will be the house price. YearRemodAdd is the remodeling date. RoofStyle is the type of roof. RoofMatl is the roof material used in the construction of roof. Exterior1st and Exterior2nd is the exterior covering on house. MasVnrType is the masonry veneer type. MasVnrArea is the masonry veneer area in square feet. ExterQual evaluates the quality of the material on the exterior. ExterCond evaluates the present condition of the material on the exterior. Foundation is the type of foundation. BsmtQual evaluates the height of the basement. BsmtCond evaluates the general condition of the basement. BsmtExposure refers to walkout or garden level walls. BsmtFinType1 and BsmtFinType2 is the rating of basement finished area. BsmtFinSF1 and BsmtFinSF2 is the type 1 and type 2 finished square feet. BsmtUnfSF is the unfinished square feet of basement area. TotalBsmtSF is the total square feet of basement area. Heating is the type of heating. HeatingQC is the heating quality and condition. CentralAir is the central air conditioning. Electrical is the electrical system. 1stFlrSF is the first-floor square feet and 2ndFlrSF is the second-floor square feet. LowQualFinSF is the low quality finished square feet (all floors). GrLivArea is the above grade (ground) living area square feet. BsmtFullBath is the basement full bathrooms. BsmtHalfBath is the basement half bathrooms. FullBath is the full bathrooms above grade. HalfBath is the half baths above grade. Bedroom is the bedrooms above grade. Kitchen is the kitchens above grade. KitchenQual is the kitchen quality. TotRmsAbvGrd is the total rooms above grade. Functional is the home functionality. Fireplaces is the number of fireplaces. FireplaceQu is the fireplace quality. GarageType is the garage location. GarageYrBlt is the year garage was built. GarageFinish is the interior finish of the garage. GarageCars is the size of garage in car capacity. GarageArea is the size of garage in square feet. GarageQual is the garage

quality. GarageCond is the garage condition. PavedDrive is the paved driveway. WoodDeckSF: Wood deck area in square feet. OpenPorchSF is the open porch area in square feet. EnclosedPorch is the enclosed porch area in square feet. 3SsnPorch is the three-season porch area in square feet. ScreenPorch is the screen porch area in square feet. PoolArea is the pool area in square feet. PoolQC is the pool quality. Fence is the fence quality. MiscFeature is the miscellaneous feature not covered in other categories. MiscVal is the $Value of miscellaneous feature. MoSold is the month Sold (MM). YrSold is the year Sold (YYYY). SaleType is the type of sale. SaleCondition is the condition of sale.

Tools Used:

Hardware: The needed time to train the model depends on the capability of the used system during the experiment. Some libraries use GPU resources over the CPU to take a shorter time to train a model.

Operating System	Windows 10
Processor	CORE i3
RAM	16GB

Language: Python

- Python is widely used in numeric and scientific computing.
- Scipy is a collection of packages for mathematics, science and engineering.
- Pandas is a data analysis and modelling library.

Libraries:

- Pandas
- Numpy
- Matplotlib
- Seaborn
- Scikit Learn

MODEL DEVELOPMENT AND EVALUATION

Model:

Regression model is used. Regression models are used to predict a continuous value. It is a supervised technique. Regression models a target prediction based on independent variables.

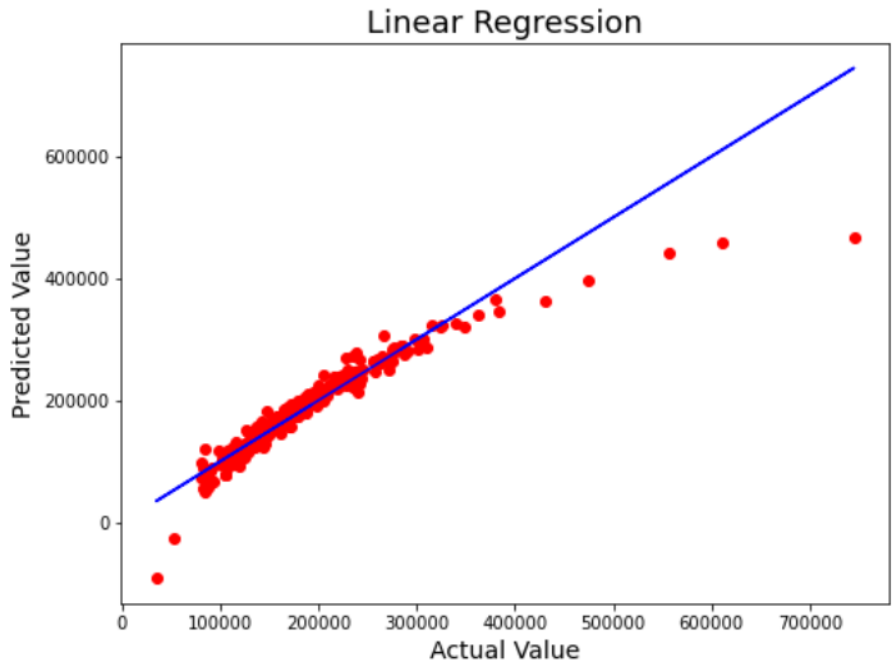
Linear Regression:

Linear regression is a machine learning algorithm based on supervised learning.

$$Y = bX + a$$

Multiple Linear Regression (MLR) is a supervised technique used to estimate the relationship between one dependent variable and more than one independent variables. Identifying the correlation and its cause-effect helps to make predictions by using these relations [4]. To estimate these relationships, the prediction accuracy of the model is essential; the complexity of the model is of more interest. However, Multiple Linear Regression is prone to many problems such as multicollinearity, noises, and overfitting, which effect on the prediction accuracy.

$$Y(x_1, x_2, x_3) = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0$$

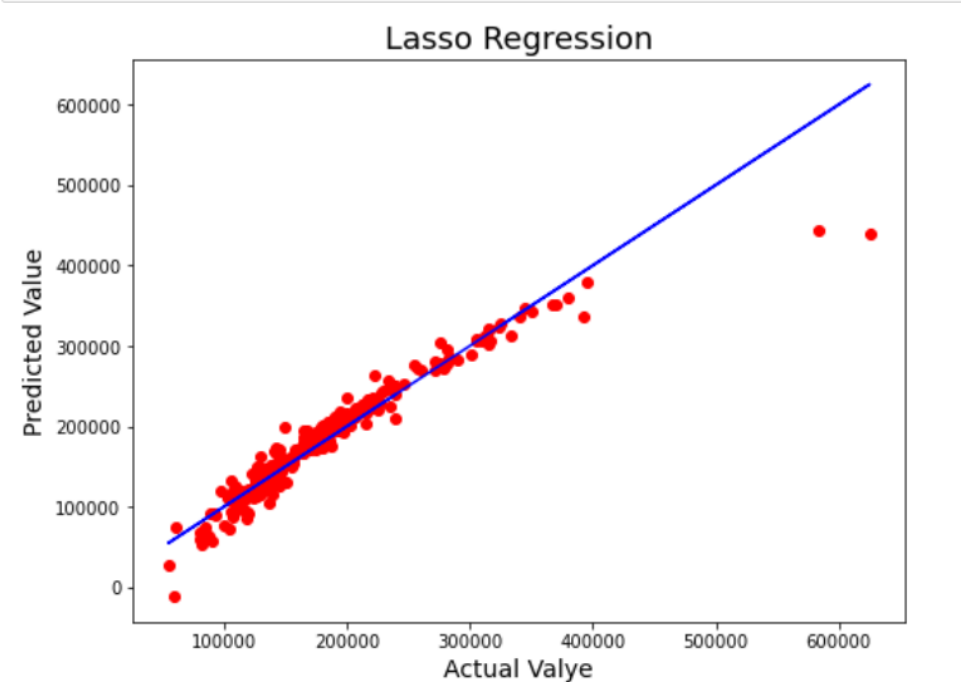


Lasso Regression:

Least Absolute Shrinkage and Selection Operator (Lasso) is a regression technique. Lasso is a powerful technique that performs regularisation and feature selection. In Lasso instead of squaring the slope like Ridge regression, the absolute value of the slope is added as a penalty term. Lasso is defined as:

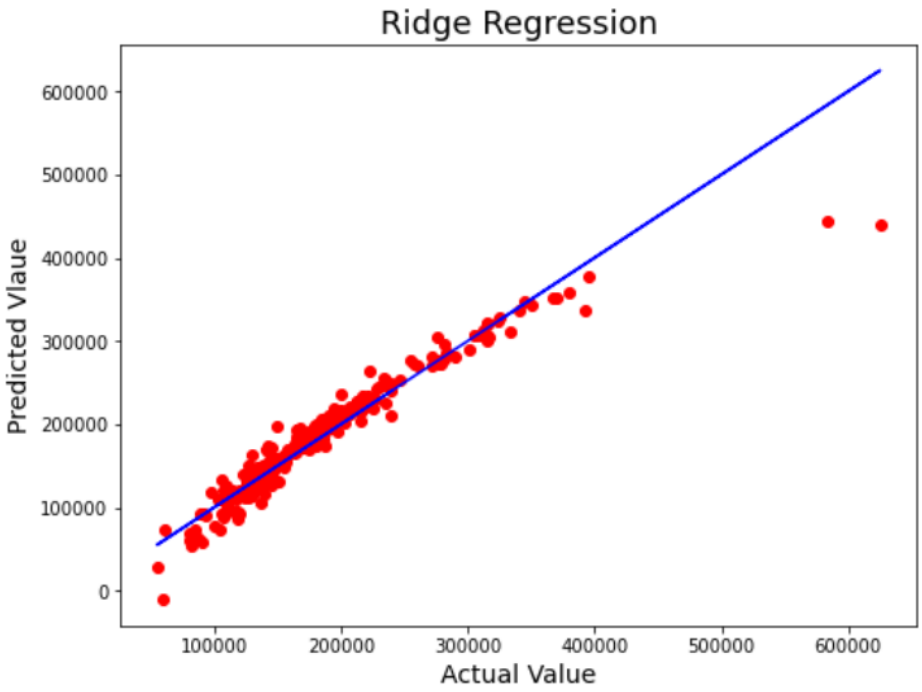
$$L = Min(sum\ of\ squared\ residuals\ +\ \alpha * |slope|)$$

Where $Min\ (sum\ of\ squared\ residuals)$ is the Least Squared Error, and $\alpha * |slope|$ is the penalty term. However, alpha α is the tuning parameter which controls the strength of the penalty term.



Ridge Regression:

The Ridge Regression is supervised regression technique. It is an estimation procedure to manage collinearity without removing variables from the regression model. In multiple linear regression, the multicollinearity is a common problem that leads least square estimation to be unbiased, and its variances are far from the correct value. Therefore, by adding a degree of bias to the regression model, Ridge Regression reduces the standard errors, and it shrinks the least square coefficients towards the origin of the parameter space. Ridge formula is:

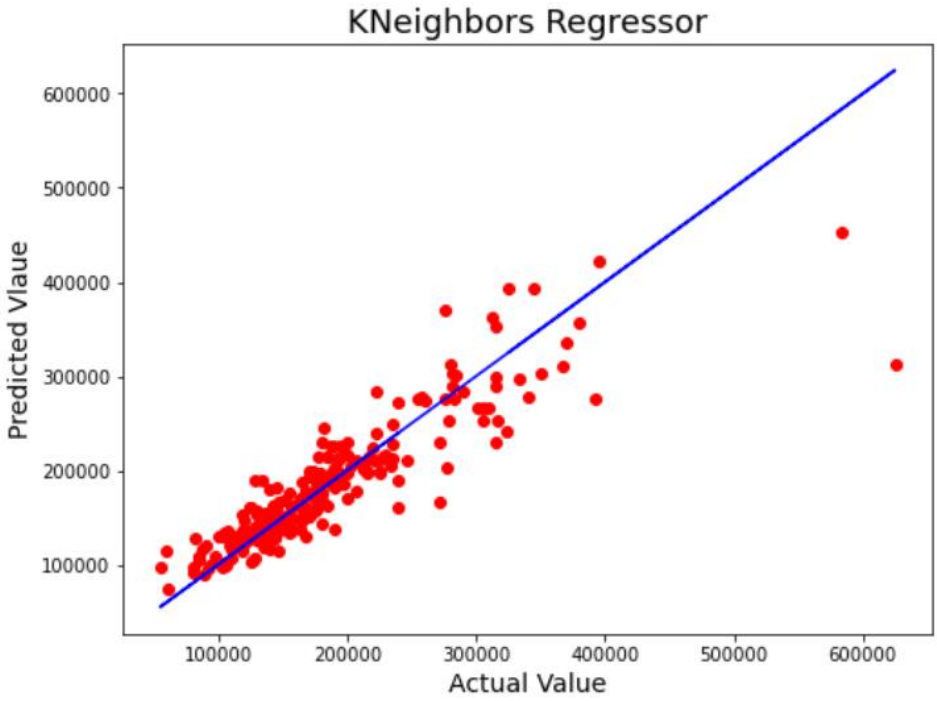


DecisionTree Regressor:

DecisionTree regressor is a supervised regression technique that builds model 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 has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node.

KNeighbors Regressor:

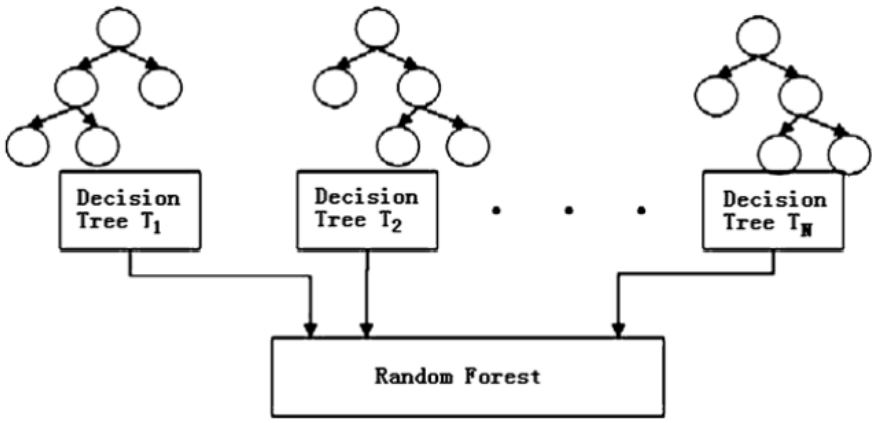
KNeighbors regressor is a supervised regression technique based on k-nearest neighbors. The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set.

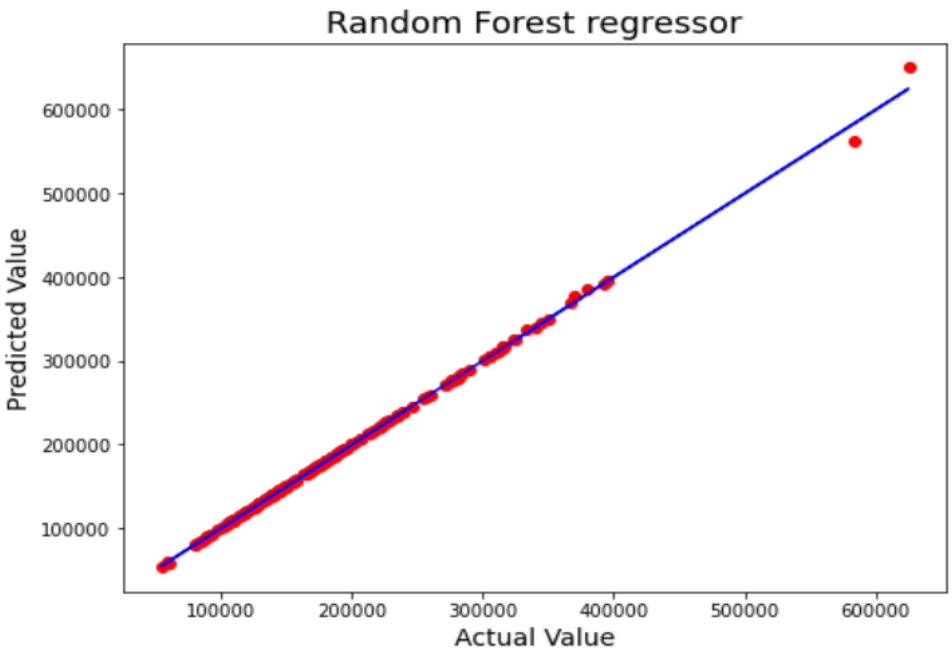


Random Forest Regressor:

A Random Forest is an ensemble technique used for performing regression tasks with the help of multiple decision trees and a method called Bootstrap Aggregation known as Bagging.

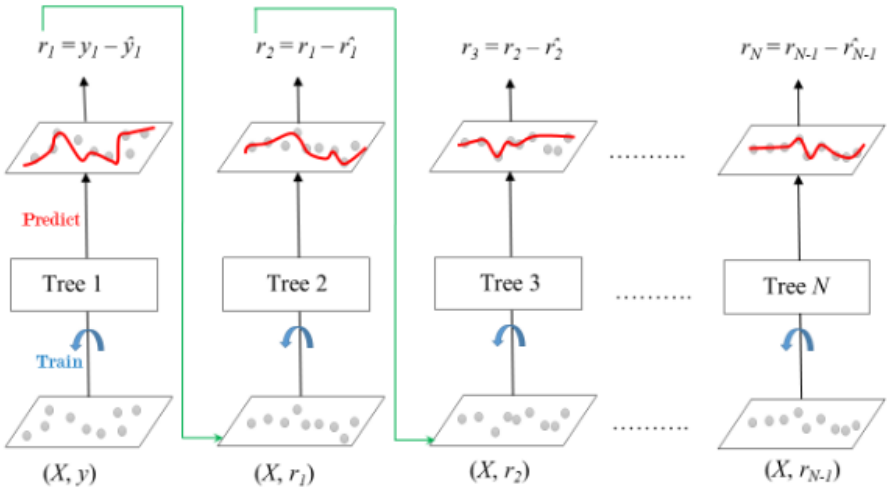
Random Forest is a model that constructs an ensemble predictor by averaging over a collection of decision trees. Therefore, it is called a forest, and there are two reasons for calling it random. The first reason is growing trees with a random independent bootstrap sample of the data. The second reason is splitting the nodes with arbitrary subsets of features. However, using the bootstrapped sample and considering only a subset of the variables at each step results in a 6 wide variety of trees. The variety is what makes Random Forest more effective than individual Decision Tree.

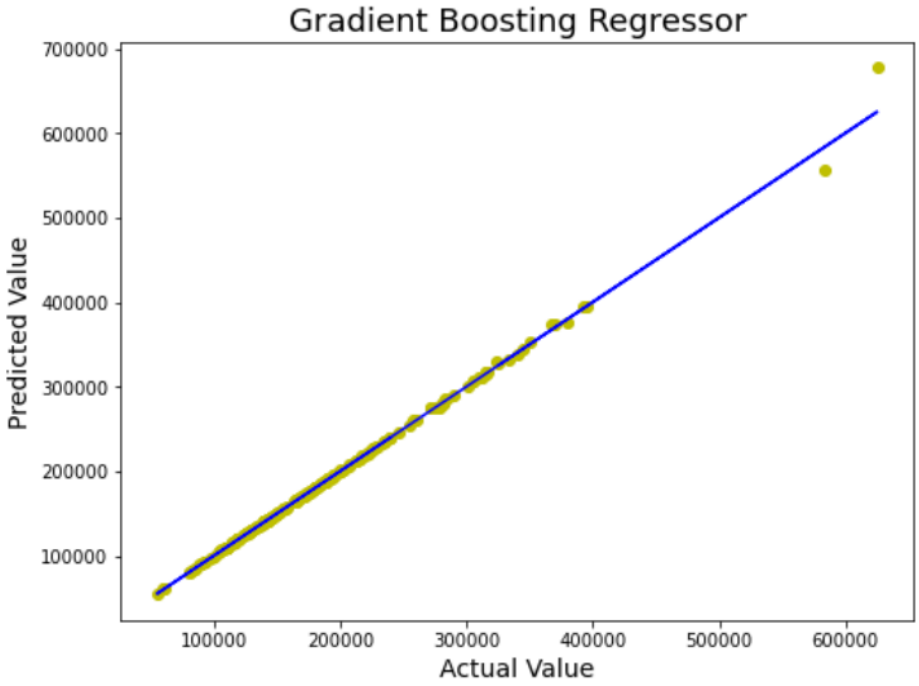




Gradient Boosting Regressor:

Gradient boosting is a machine learning technique for regression, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient boosted trees, which usually performs random forest. It builds the model in a stage-wise fashion like other boosting methods and it generalizes them by allowing optimization of an arbitrary differentiable loss function.





Algorithms Used:

Linear Regression

Regularization:

- Lasso Regression
- Ridge Regression
- Decision Tree Regressor
- KNeighbors Regressor
- SGD Regressor

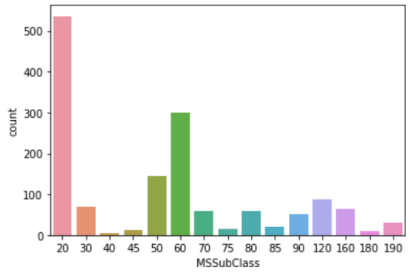
Ensemble Methods:

- Random Forest Regressor
- Gradient Boosting Regressor

Visualizations:

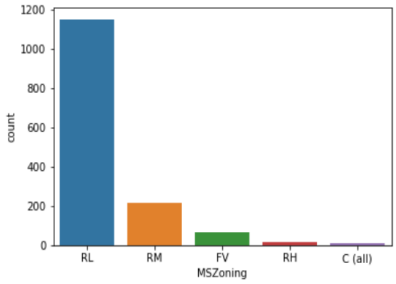
```
sns.countplot(house['MSSubClass'])
```

```
<AxesSubplot:xlabel='MSSubClass', ylabel='count'>
```



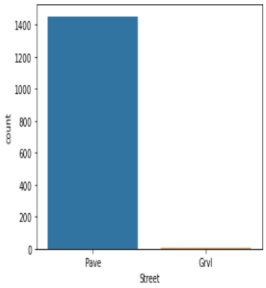

```
sns.countplot(house['MSZoning'])
```

```
<AxesSubplot:xlabel='MSZoning', ylabel='count'>
```



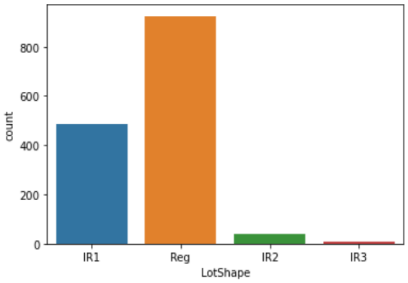

```
sns.countplot(house['Street'])
```

```
<AxesSubplot:xlabel='Street', ylabel='count'>
```



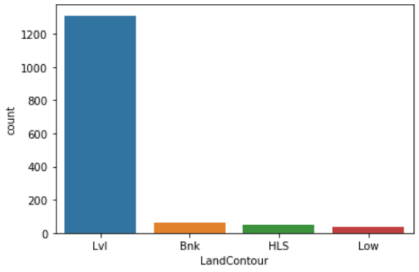

```
sns.countplot(house['LotShape'])
```

```
<AxesSubplot:xlabel='LotShape', ylabel='count'>
```



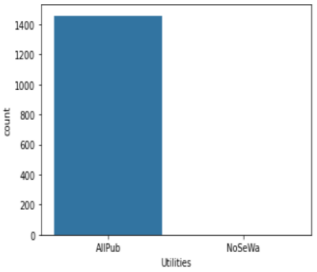

```
sns.countplot(house['LandContour'])
```

```
<AxesSubplot:xlabel='LandContour', ylabel='count'>
```



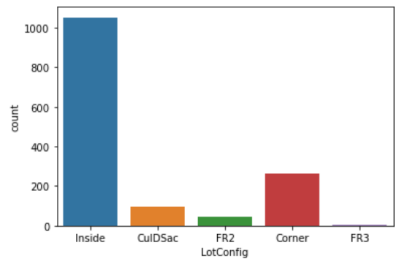

```
sns.countplot(house['Utilities'])
```

```
<AxesSubplot:xlabel='Utilities', ylabel='count'>
```



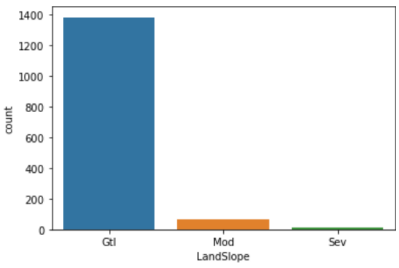

```
sns.countplot(house['LotConfig'])
```

```
<AxesSubplot:xlabel='LotConfig', ylabel='count'>
```



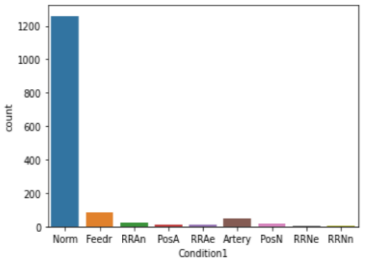

```
sns.countplot(house['LandSlope'])
```

```
<AxesSubplot:xlabel='LandSlope', ylabel='count'>
```



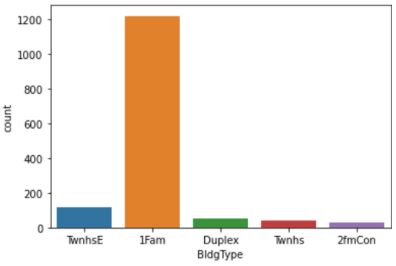

```
sns.countplot(house['Condition1'])
```

```
<AxesSubplot:xlabel='Condition1', ylabel='count'>
```



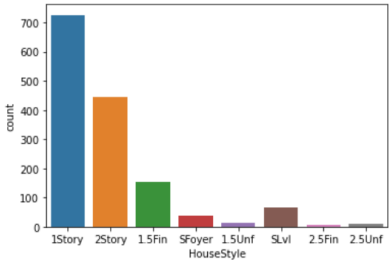

```
sns.countplot(house['BldgType'])
```

```
<AxesSubplot:xlabel='BldgType', ylabel='count'>
```



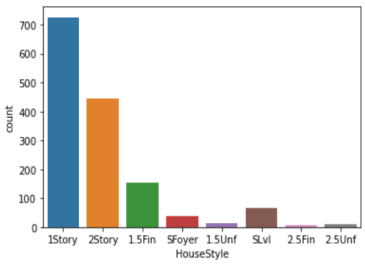

```
sns.countplot(house['HouseStyle'])
```

```
<AxesSubplot:xlabel='HouseStyle', ylabel='count'>
```



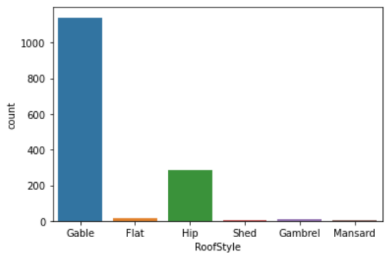

```
sns.countplot(house['HouseStyle'])
```

```
<AxesSubplot:xlabel='HouseStyle', ylabel='count'>
```



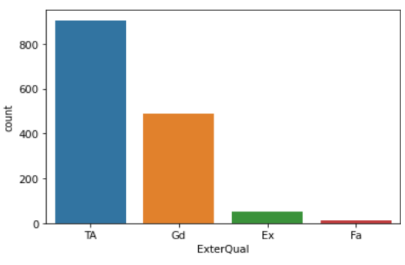

```
sns.countplot(house['RoofStyle'])
```

```
<AxesSubplot:xlabel='RoofStyle', ylabel='count'>
```



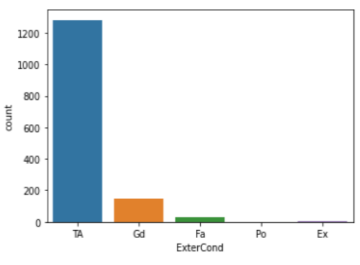

```
sns.countplot(house['ExterQual'])
```

```
<AxesSubplot:xlabel='ExterQual', ylabel='count'>
```



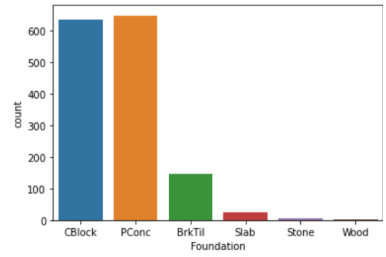

```
sns.countplot(house['ExterCond'])
```

```
<AxesSubplot:xlabel='ExterCond', ylabel='count'>
```



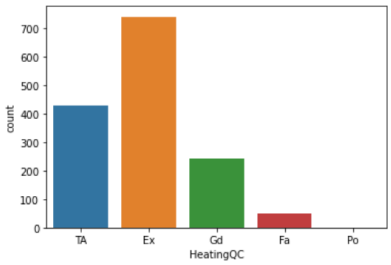
```
sns.countplot(house['Foundation'])
```

```
<AxesSubplot:xlabel='Foundation', ylabel='count'>
```



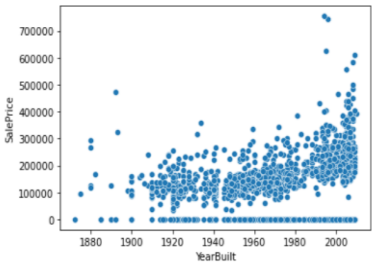

```
sns.countplot(house['HeatingQC'])
```

```
<AxesSubplot:xlabel='HeatingQC', ylabel='count'>
```



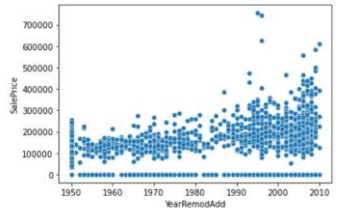

```
sns.scatterplot(x='YearBuilt',y='SalePrice',data=house)
```

```
<AxesSubplot:xlabel='YearBuilt', ylabel='SalePrice'>
```



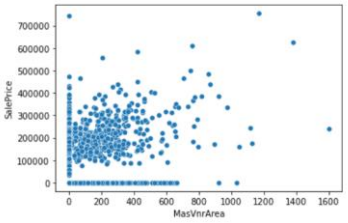

```
sns.scatterplot(x='YearRemodAdd',y='SalePrice',data=house)
```

```
<AxesSubplot:xlabel='YearRemodAdd', ylabel='SalePrice'>
```



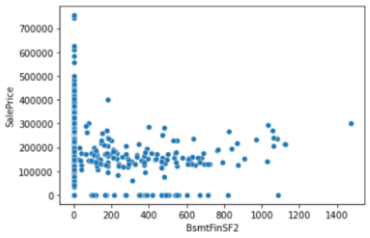

```
sns.scatterplot(x='MasVnrArea',y='SalePrice',data=house)
```

```
<AxesSubplot:xlabel='MasVnrArea', ylabel='SalePrice'>
```




```
sns.scatterplot(x='BsmtFinSF2',y='SalePrice',data=house)
```

```
<AxesSubplot:xlabel='BsmtFinSF2', ylabel='SalePrice'>
```



Interpretation of Results:

Many machine learning algorithms are used to predict. Using these algorithms is beneficial so that the result can be as near to the claimed results. However, the prediction accuracy of these algorithms depends heavily on the given data when training the model. If the data is in bad shape, the model will be overfitted and inefficient, which means that data pre-processing is an important part of this experiment and will affect the final results. Thus, multiple combinations of pre-processing methods need to be tested before getting the data ready to be used in training.

From Visualizations it is clear that how each feature from the dataset is relatable to the house price. In pre-processing, unnecessary feature is removed, outliers are removed and skewness from the input variables are removed. The data's are scaled and transformed for better training purpose. Gradient Boosting algorithm gave the best model with 99% accuracy.

CONCLUSION

The goal is to achieve the system which will reduce the human effort to find a house having reasonable price. Project is focused on predicting the house price according to the area for which machine learning methods are used. The experimental results showed that this technique that are used while developing system will give accurate prediction of house price. The study shows a comparison between the different regression algorithms.

Gradient Boosting Regressor gave the best model with same r2 and cross validation score with accuracy pf 99%.

Limitations:

This study did not cover all regression algorithms; instead, it is focused on the chosen algorithm, starting from the basic regression techniques to the advanced techniques.