House Prices: Advanced Regression Techniques

Name: Arpit Garg
Id: A1784072
Master of Data Science
The University of Adelaide

Categories:

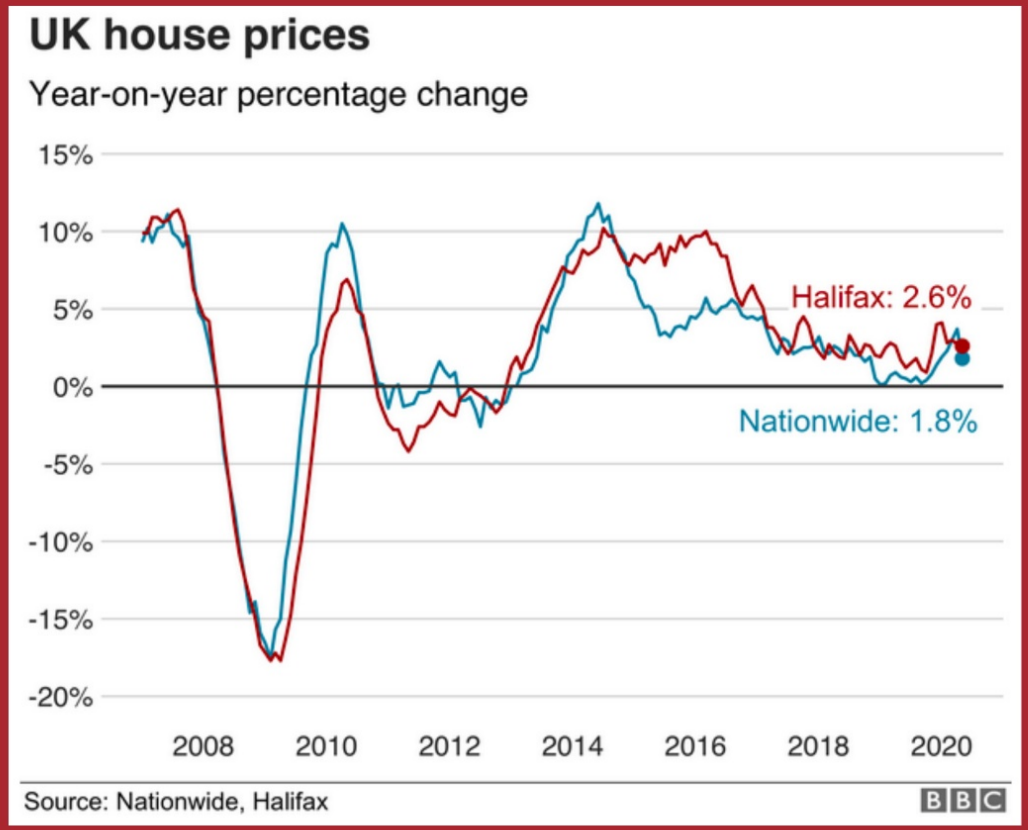
1. Introduction
2. Exploratory Data Analysis
3. Data Pre-Processing
4. Data Modeling
5. Evaluation

House Prices: Advanced Regression Techniques

Name: Arpit Garg
Id: A1784072
Master of Data Science
The University of Adelaide


Introduction
Aim to solve Kaggle Problem to predict houses .
Different parameters that can effect house prices.
We are using AMES housing dataset.
Why Useful?
AMES Dataset

BBC News



For Investors

For Residents

AMES Dataset

79 Different Parameters

Target: Sale Price

Test and Training set

Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley
1461	20	RH	80	11622	Pave	NA
1462	20	RL	81	14267	Pave	NA
1463	60	RL	74	13830	Pave	NA
1464	60	RL	78	9978	Pave	NA
1465	120	RL	43	5005	Pave	NA

House Prices: Advanced Regression Techniques

Name: Arpit Garg
Id: A1784072
Master of Data Science
The University of Adelaide


Correlation
Print Details
Exploratory Data Analysis
1. Print some details like Shape
2. Null Columns and Missing Plot
3. Skewness
4. Plot Numerical and Categorical Data
5. Correlation Matrix
Plot
Missing
Skewness

Information Related to Dataset

Shape

```
#shape of training data
print('****************************************')
print('\n\nShape of training data: ',train.shape,'\n\n')
print('****************************************')
```

```
****************************************


Shape of training data:  (1460, 81)


****************************************
```

```
#shape of testing data
print('****************************************')
print('\n\nShape of testing data: ',test.shape,'\n\n')
print('****************************************')
```

```
****************************************


Shape of testing data:  (1459, 80)


****************************************
```

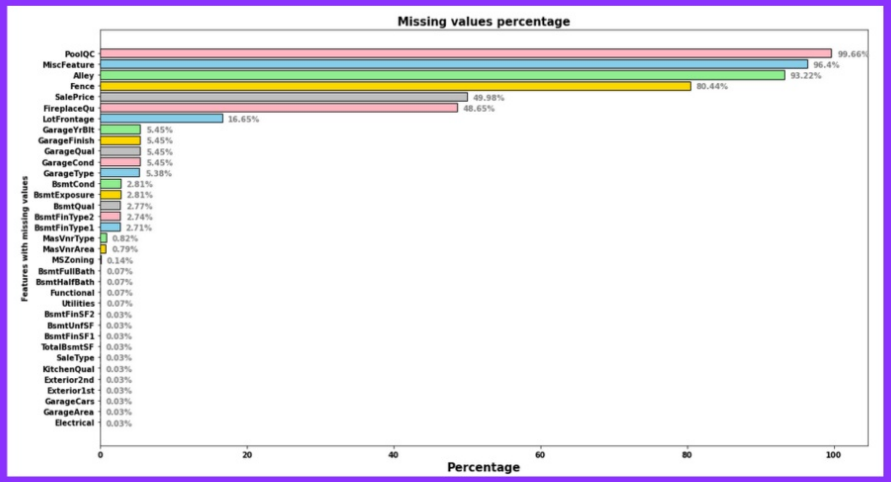
Column Details

```
ame of columns in train data:

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')
```

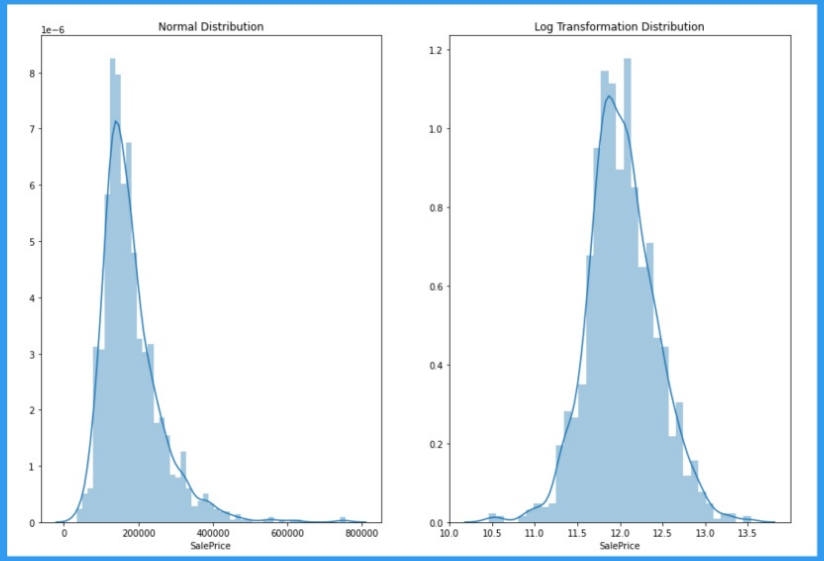
Head Values

```
Displaying top 5 entries in train data:

   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
0   1          60       RL         65.0     8450   Pave   NaN      Reg
1   2          20       RL         80.0     9600   Pave   NaN      Reg
2   3          60       RL         68.0    11250   Pave   NaN      IR1
3   4          70       RL         60.0     9550   Pave   NaN      IR1
4   5          60       RL         84.0    14260   Pave   NaN      IR1

  LandContour Utilities  ... PoolArea PoolQC Fence MiscFeature MiscVal MoSold  \
0         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      2
1         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      5
2         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      9
3         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      2
4         Lvl    AllPub  ...        0    NaN   NaN         NaN       0     12

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500
3   2006        WD        Abnorml     140000
4   2008        WD         Normal     250000

[5 rows x 81 columns]
```

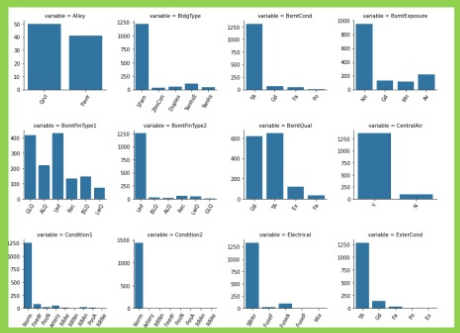
Null Columns

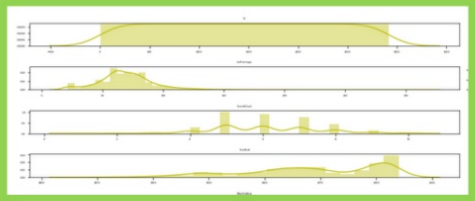


Skewness in target



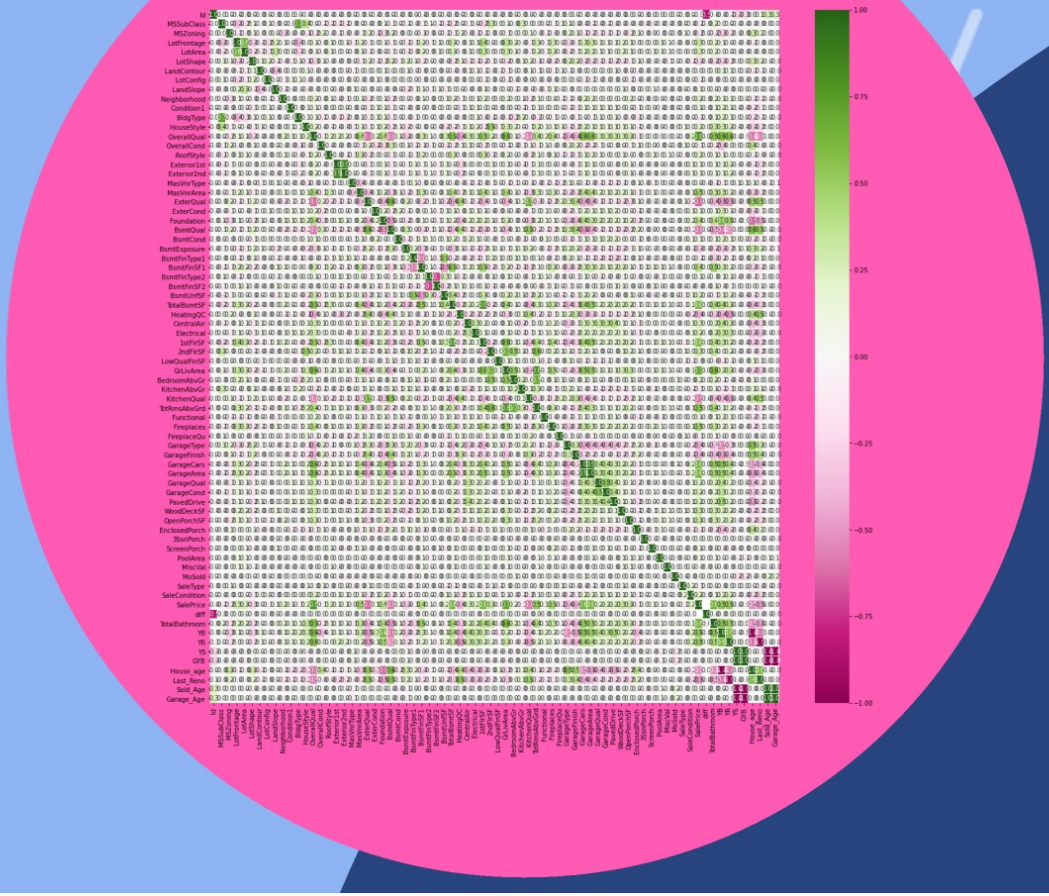
Categorical





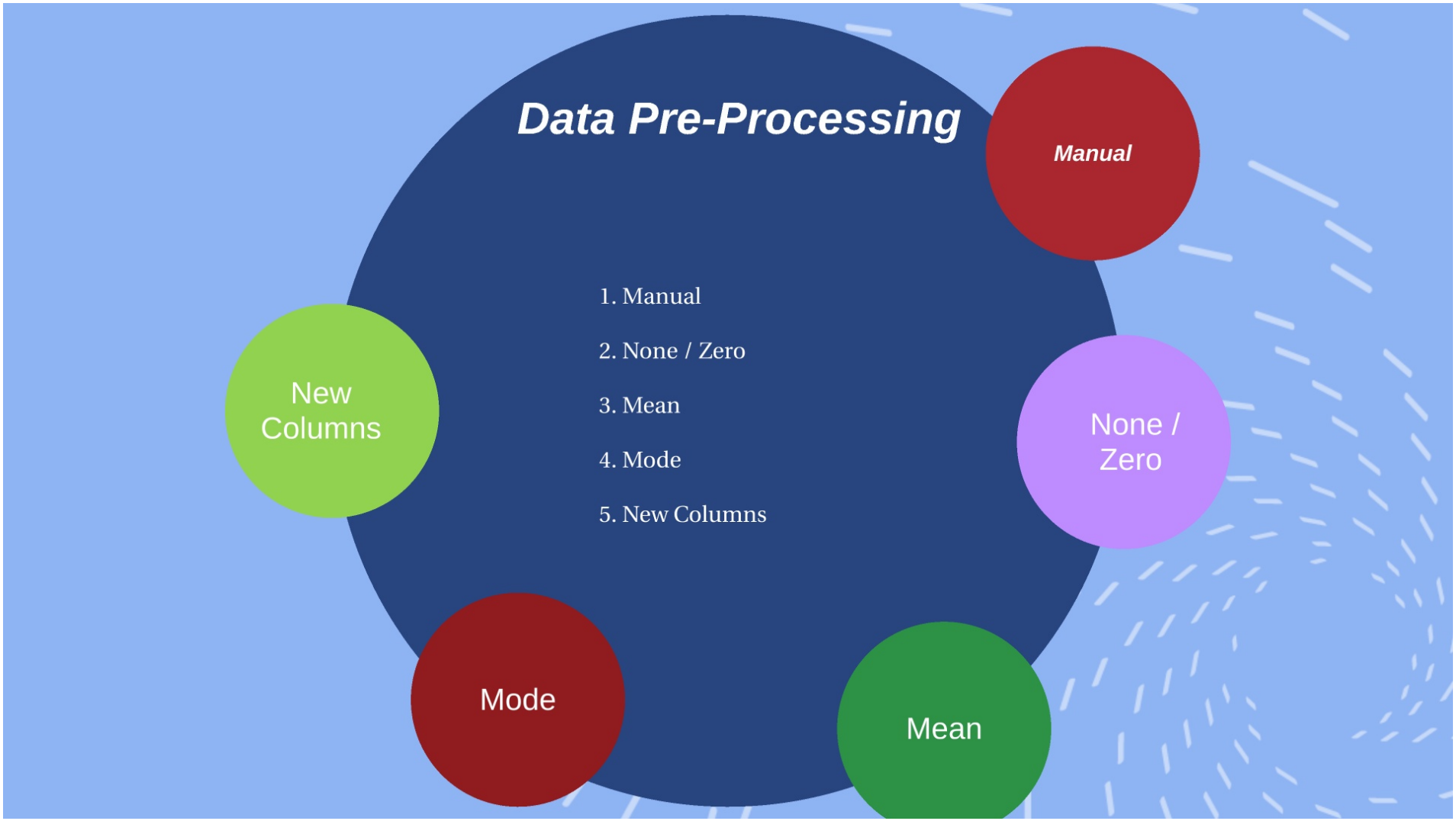
Numerical

Correlation



House Prices: Advanced Regression Techniques

Name: Arpit Garg
Id: A1784072
Master of Data Science
The University of Adelaide


Data Pre-Processing
1. Manual
2. None / Zero
3. Mean
4. Mode
5. New Columns
Manual
None / Zero
Mean
Mode
New Columns

Manual Transformations

1. No information about fireplace then there is no fireplace so none.

2. Relationship between LotFrontage and sqrt(LotArea)

```
Correlation between LotFrontage and LotArea:     0.6476580398617832
```

None / Zero

Some Parameters like Garage and Basement are grouped together. If data type is object then replaced by None, otherwise Zero for numerical.

```python
#Handling all Basement columns at once
print('""""""""""""""""""""""""""""""""""""""""')
print('\n\nHandling Basement Columns\n\n')
print('""""""""""""""""""""""""""""""""""""""""')
bsmt_cols = ['BsmtCond','BsmtExposure','BsmtQual','BsmtFinType2','BsmtFinType1']
df[bsmt_cols][df['BsmtExposure'].isnull()==True]
#Basement Imputation
for cols in bsmt_cols:
    if df[cols].dtype==np.object:
        df.loc[df[cols].isnull(),cols] = 'None'
    else:
        df.loc[df[cols].isnull(),cols] = 0
```

```python
#Handling all garage columns at once
print('""""""""""""""""""""""""""""""""""""""""')
print('\n\nHandling Garage Columns\n\n')
print('""""""""""""""""""""""""""""""""""""""""')
garage_cols=['GarageType','GarageQual','GarageCond','GarageYrBlt','GarageFinish','GarageCars','GarageArea']
df[garage_cols][df['GarageType'].isnull()==True]
#Garage Imputation
for cols in garage_cols:
    if df[cols].dtype==np.object:
        df.loc[df[cols].isnull(),cols] = 'None'
    else:
        df.loc[df[cols].isnull(),cols] = 0
```

Mean

Replacing some missing values with mean of column

```
print('****************************************')
print('\n\nNow Handling All Left Columns\n\n')
print('****************************************')
left_out = ['Exterior2nd','Exterior1st','TotalBsmtSF','Electrical','SaleType','BsmtFinSF1','BsmtUnfSF','KitchenQual
for cols in left_out:
    if df[cols].dtype==np.object:
        temp = df[cols].mode()
        df.loc[df[cols].isnull(),cols] = temp.iloc[0]
    else:
        df.loc[df[cols].isnull(),cols] = df[cols].mean()
```

Mean Value Theorem

If $f(x)$ is continuous on $[a, b]$ and differentiable on (a, b) then there is a c such that

$$f'(c) = \frac{f(b) - f(a)}{b - a}$$

Mode

In some columns missing values are replaced by mode values

```
#Handling functional
print('''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''')
print('\n\nNow Handling Functional Column\n\n')
print('''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''')
df['Functional'].fillna(df.Functional.mode()[0],inplace=True)
```

The mode is the most common value in a data set

New Columns

Some new columns are introduced based on old columns and old ones are dropped.

Years:

```
#Convert the columns
print('''''''''''''''''''''''''''''''''''''''''''')
print('\n\nConverting some columns to more appropriate\n\n')
print('''''''''''''''''''''''''''''''''''''''''''')
df['YB'] = pd.to_datetime(df['YearBuilt'], format='%Y', errors='ignore').dt.year
df['YR'] = pd.to_datetime(df['YearRemodAdd'], format='%Y', errors='ignore').dt.year
df['YS'] = pd.to_datetime(df['YrSold'], format='%Y', errors='ignore').dt.year
df['GYB'] = pd.to_datetime(df['YrSold'], format='%Y', errors='ignore').dt.year
# Construct new variables for years since - Take the difference between the current year and variable year
df["House_age"] = datetime.datetime.now().year - df['YB']
df["Last_Reno"] = datetime.datetime.now().year- df['YR']
df["Sold_Age"] = datetime.datetime.now().year - df['YS']
df["Garage_Age"] = datetime.datetime.now().year - df['GYB']
# Delete old columns
print('''''''''''''''''''''''''''''''''''''''''''''')
print('\n\nAdding new Columns\n\n')
print('''''''''''''''''''''''''''''''''''''''''''''')
print('''''''''''''''''''''''''''''''''''''''''''''')
print('\n\nRemoving old Columns\n\n')
print('''''''''''''''''''''''''''''''''''''''''''''')
df.drop(['YearBuilt', 'YearRemodAdd', 'YrSold', 'GarageYrBlt'], axis=1,inplace=True)
```

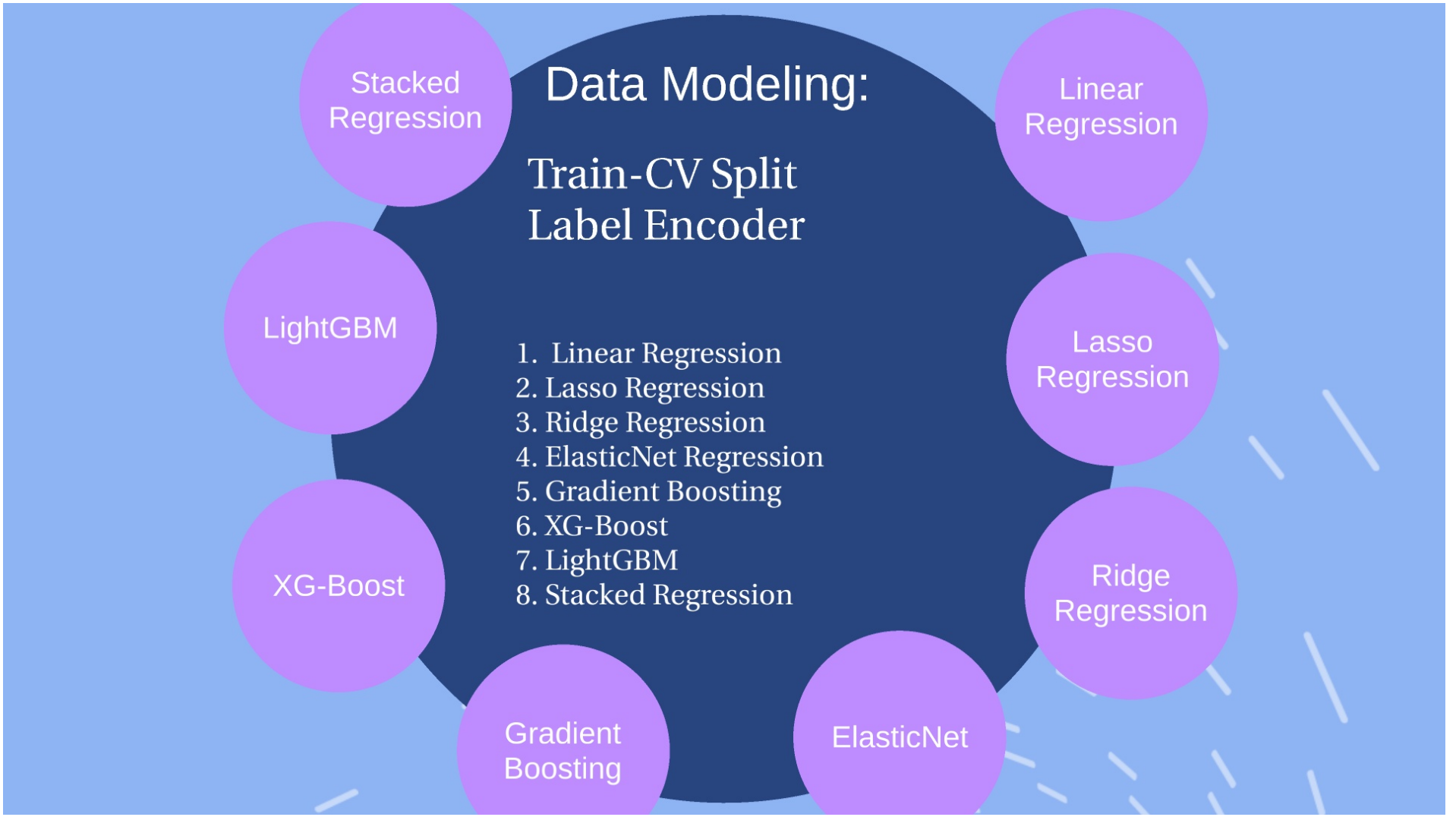
Bathrooms

Bathrooms:

```
#Combining all the bathroom
print('''''''''''''''''''''''''''''''''''''''''''''''')
print('\n\nNow Handling Bathroom Columns\n\n')
print('''''''''''''''''''''''''''''''''''''''''''''''')
df['TotalBathroom'] = df['BsmtFullBath'] + 0.5*df['BsmtHalfBath'] + df['FullBath'] + 0.5*df['HalfBath']
df.drop(['BsmtFullBath','BsmtHalfBath','FullBath','HalfBath'],axis=1,inplace=True)
df['TotalBathroom'].fillna(df['TotalBathroom'].mode()[0],inplace=True)
```

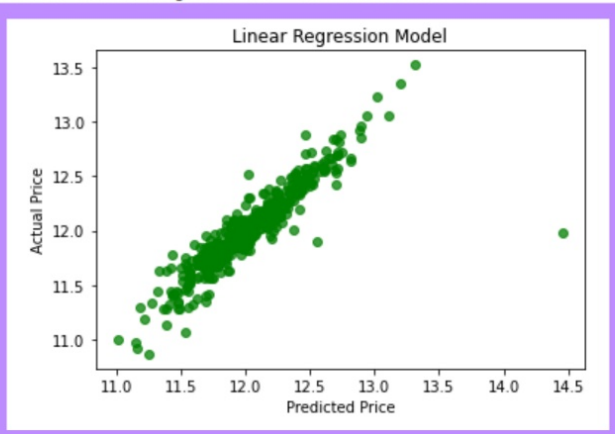
House Prices: Advanced Regression Techniques

Name: Arpit Garg
Id: A1784072
Master of Data Science
The University of Adelaide


Stacked Regression
Data Modeling:
Train-CV Split
Label Encoder
Linear Regression
LightGBM
1. Linear Regression
2. Lasso Regression
3. Ridge Regression
4. ElasticNet Regression
5. Gradient Boosting
6. XG-Boost
7. LightGBM
8. Stacked Regression
Lasso Regression
XG-Boost
Ridge Regression
Gradient Boosting
ElasticNet

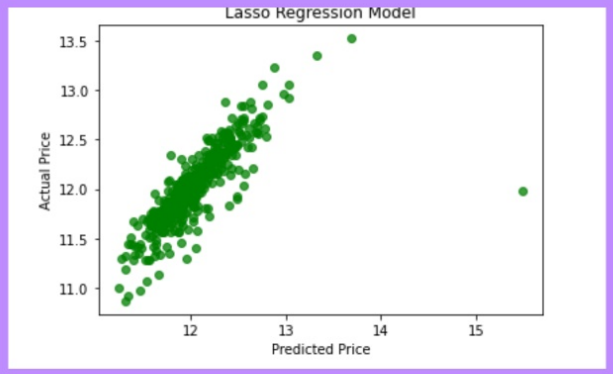
Linear Regression

```
Root Mean Square Error in Linear Regression:  0.1702980079992807

R2 Score in Linear Regression:  0.8049382891288266
```



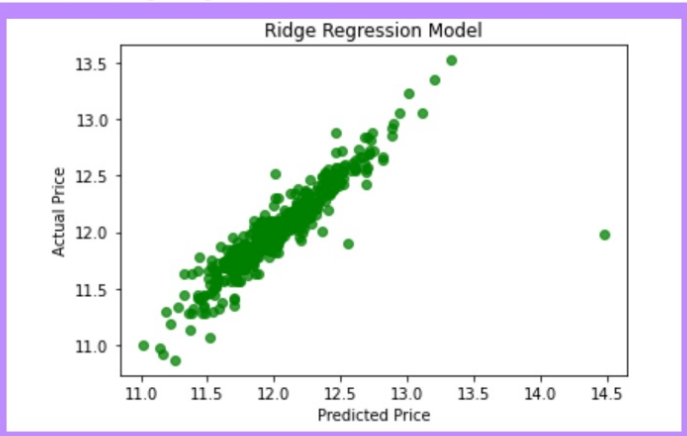
Lasso Regression

```
Root Mean Square Error in Lasso Regression:  0.23942461326020192

R2 Score in Lasso Regression:  0.614441322218327
```



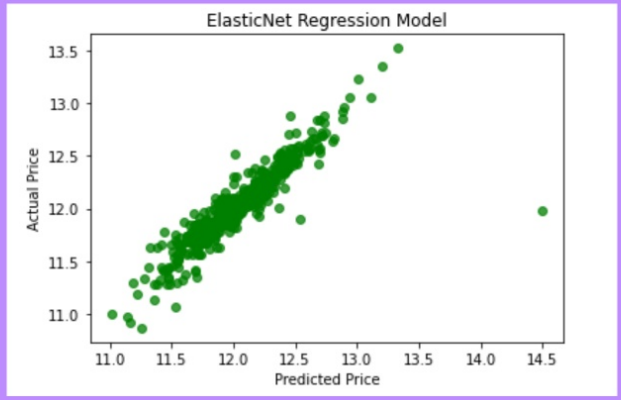
Ridge Regression

```
Root Mean Square Error in Ridge Regression:  0.1706895079758536

R2 Score in Ridge Regression:  0.8040403991745132
```



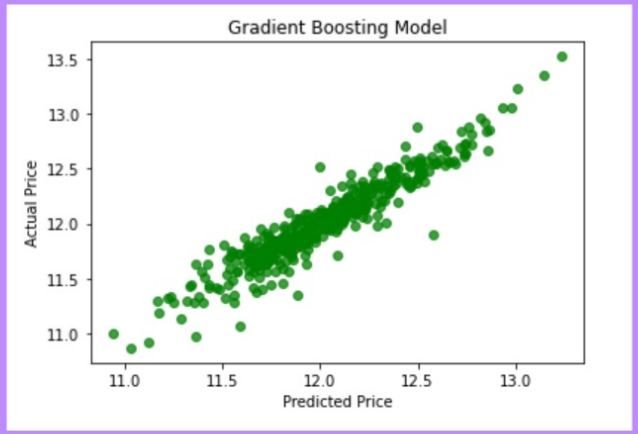
ElasticNet

```
Root Mean Square Error in ElasticNet Regression:  0.17125809946845255

R2 Score in ElasticNet Regression:  0.8027326850339862
```



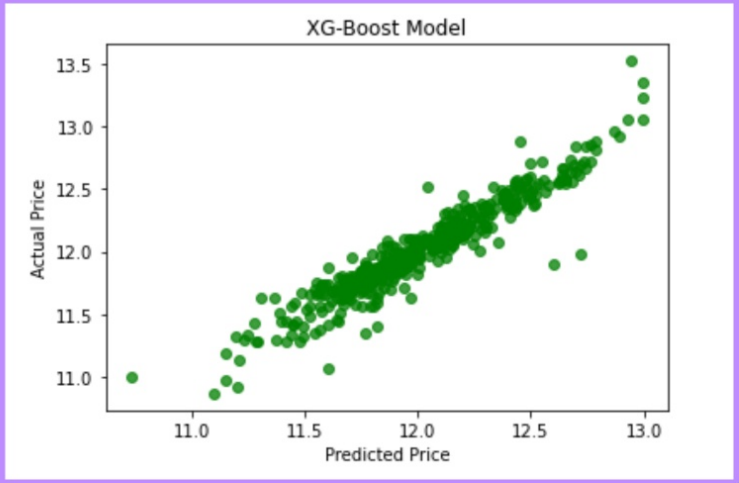
Gradient Boosting

```
Root Mean Square Error in Gradient Boosting:  0.1254698784502168

R2 Score in Gradient Boosting:  0.8941156322756973
```



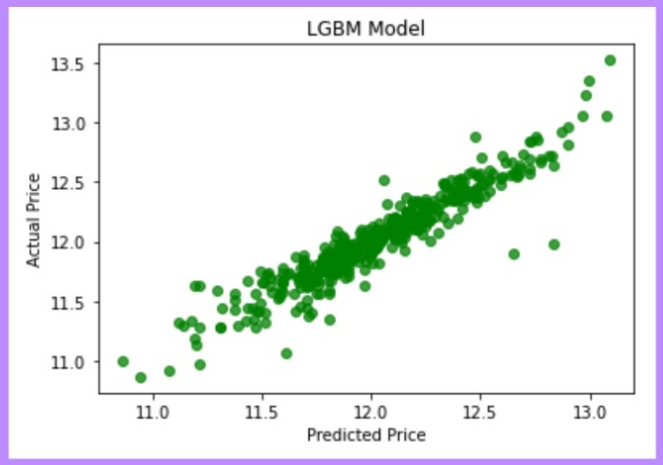
XG-Boost

```
Root Mean Square Error in XGB:  0.12323608253651414

R2 Score in XGB:  0.8978522836438579
```



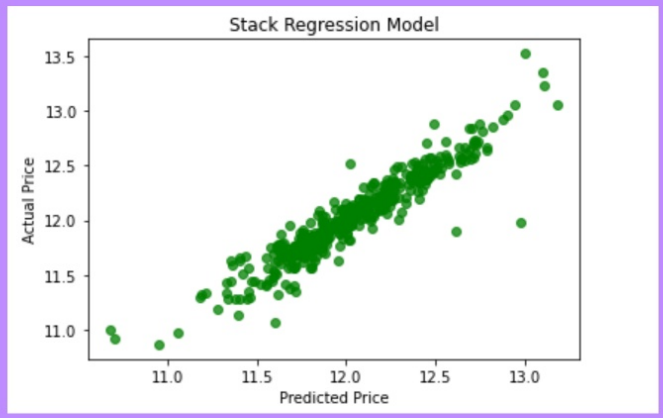
LightGBM

```
Root Mean Square Error in LGBM:  0.12904276385332766

R2 Score in LGBM:  0.8879994370462735
```



Stacked Regression

```
Root Mean Square Error Stacked CV:  0.12927422119750184

Score for Stacked Regression:  0.8875972974645968
```



House Prices: Advanced Regression Techniques

Name: Arpit Garg
Id: A1784072
Master of Data Science
The University of Adelaide

Evaluation

Table 1: Evaluation Metrics

S.No.	Regression	RMSE	R2 Score
1	Linear	0.17029800799928	0.804938289128826
2	Lasso	0.239424613260201	0.614441322218327
3	Ridge	0.170689507975853	0.804040399174513
4	Elastic-Net	0.171258099468452	0.802732685033986
5	Gradient Boost	0.125469878450216	0.894115632275697
6	XG-Boost	0.123236082536514	0.897852283643857
7	LightBGM	0.129042763853327	0.887999437046273
8	Stack	0.129274221197501	0.887597297464596

https://www.kaggle.com/arpit2412/competitions

House Prices: Advanced Regression Techniques

Name: Arpit Garg
Id: A1784072
Master of Data Science
The University of Adelaide


Thank You

House Prices: Advanced Regression Techniques

Name: Arpit Garg
Id: A1784072
Master of Data Science
The University of Adelaide