HOUSING: PRICE PREDICTION

Submitted by: RAJ

ACKNOWLEDGMENT

I express my sincere gratitude to Flip Robo Technologies for giving me the opportunity to work on this project on Housing Price prediction using machine learning algorithms. I would also like to thank Flip Robo Technologies for providing me with the requisite datasets to work with.

I acknowledge my indebtedness to the authors of papers titled: “House Price Prediction using a Machine Learning Model: A Survey of Literature” and “The impact of housing quality on house prices in eight capital cities, Australia” for providing me with invaluable insights and knowledge of the dynamic relationships that exist in the economics of real estate and housing markets.

INTRODUCTION

Business Problem Framing

Houses are one of the necessary needs of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company. 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 company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:

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

Conceptual Background of the Domain Problem

Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies.

Hedonic Characteristics of Housing Price: A Hedonic approach is preferred for predicting the sale prices in the housing market because the market displays resilience, flexibility and spatial fixity.

Housing Attributes: Studying the structural, locational, and economic attributes of housing properties is crucial in understanding their mutually inclusive relationships with their pricing.

Review of Literature

2 research papers, namely: “House Price Prediction using a Machine Learning Model: A Survey of Literature” and “The impact of housing quality on house prices in eight capital cities, Australia” were reviewed and evaluated to gain insights into all the attributes that influence the price of house.

From studying the papers and analysing the research work it, is learnt that locational attributes and structural attributes are prominent factors in predicting house prices. Studies suggest that there exists a close relationship between House pricing and locational attributes such as distance from the closest shopping center, train station, position offering views of hills or shore, the neighborhood in which the property is situated etc.

Structural attributes of the house like lot size, lot shape, quality and condition of the house, garage capacity, rooms, Lot frontage, number of bedrooms, bathrooms, overall finishing of the house etc play a big role in influencing the house price.

Neighbourhood qualities can be included in deciding house price. Factors like efficiency of public education, community social status, the socio-cultural demographics improve the worth of a property.

The demand side of the housing market is also a necessary component. Although population growth is widely known as a driver in housing demand, the key issue lies in the proportion of people with abundant financial resources.

Variables representing land value such as rents and material costs also demonstrate their influence in explaining house prices, which are positively related to housing prices.

Multiple regression analysis models allow to ascertain price predictions by capturing independent and dependent variable data. In Using multiple regression modelling techniques, we can describe changes brought to a dependent variable with changes in the independent variables.

In this research, various models were built in which the house Sale Price is projected as separate and dependent variable while locational, structural and various other attributes of housing properties were treated as independent variables. Therefore, the house price is set as a target or dependency variable, while other attributes are set as independent variables to determine the main variables by identifying the correlation coefficient of each attribute.

Motivation for the Problem Undertaken

There is a steady rise in house demand with every passing year, and consequently the house prices are rising every year. The problem arises when there are numerous variables such as location and property demand that influence the pricing. Therefore, buyers, sellers, developers and the real estate industry are keen to know the most important factors influencing the house price to help investors make sound decisions and help house builders set the optimal house price. There are many benefits that home buyers, property investors, and house builders can reap from the house-price model. This model aims to serve as a repository of such information and gainful insights to home buyers, property investors and house builders, that will help them determine best house prices. This model can be useful for potential buyers in deciding the characteristics of a house they want

that best fits their budget and will be of tremendous benefit, especially to housing developers and researchers, to ascertain the most significant attributes to determine house prices and to acknowledge the best machine learning model to be used to conduct a study in this field.

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

Various Regression analysis techniques were used to build predictive models to understand the relationships that exist between Housing sales prices and various Housing property attributes. The Regression analysis models were used to predict the Sale price value for changes in Housing property attributes.

Regression modelling techniques were used in this Problem since Sales Price data distribution is continuous in nature.

In order to forecast house price, predictive models such as ridge regression Model, Random Forest Regression model, Decision tree Regression Model, Support Vector Machine Regression model, Extreme Gradient Boost Regression were used to describe how the values of Sale Price depended on the independent variables of various Housing property attributes.

Data Sources and their formats

The dataset was compiled by a US-based housing company named Surprise Housing. The company has collected a data set from the sale of houses in Australia. The dataset was made available in .csv file format.

There are 2 datasets: One for training the predictive machine learning models and the second one to be used by the models for predicting the SalePrice(target variable).

```
HDF.head(50)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVa
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
5	1197	60	RL	58.0	14054	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
6	561	20	RL	NaN	11341	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
7	1041	20	RL	88.0	13125	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	NaN	
8	503	20	RL	70.0	9170	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	40
9	576	50	RL	80.0	8480	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
10	449	50	RM	50.0	8600	Pave	NaN	Reg	Bnk	AllPub	...	0	NaN	NaN	NaN	

Test dataset

```
In [7]: HDF_test.head()
```

Out[7]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	ScreenPorch	PoolArea	PoolQC	Fence	MiscFeatu
0	337	20	RL	86.0	14157	Pave	NaN	IR1	HLS	AllPub	...	0	0	NaN	NaN	Na
1	1018	120	RL	NaN	5814	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	Na
2	929	20	RL	NaN	11838	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	Na
3	1148	70	RL	75.0	12000	Pave	NaN	Reg	Bnk	AllPub	...	0	0	NaN	NaN	Na
4	1227	60	RL	86.0	14598	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	Na

5 rows × 80 columns

Training Dataset contains 1168 entries and 81 variables, while Test Dataset contains 292 entries and 80 variables.

Dataset Description

The Independent Feature columns are:

- MSSubClass: Identifies the type of dwelling involved in the sale.
- MSZoning: Identifies the general zoning classification of the sale.
- LotFrontage: Linear feet of street connected to property
- LotArea: Lot size in square feet
- Street: Type of road access to property
- Alley: Type of alley access to property
- LotShape: General shape of property
- LandContour: Flatness of the property
- Utilities: Type of utilities available
- LotConfig: Lot configuration

- LandSlope: Slope of property
- Neighborhood: Physical locations within Ames city limits
- Condition1: Proximity to various conditions
- Condition2: Proximity to various conditions (if more than one is present)
- BldgType: Type of dwelling
- HouseStyle: Style of dwelling
- OverallQual: Rates the overall material and finish of the house
- OverallCond: Rates the overall condition of the house
- YearBuilt: Original construction date
- YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)
- RoofStyle: Type of roof
- RoofMatl: Roof material
- Exterior1st: Exterior covering on house
- Exterior2nd: Exterior covering on house (if more than one material)
- MasVnrType: Masonry veneer type
- MasVnrArea: 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: 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: Rating of basement finished area
- BsmtFinSF1: Type 1 finished square feet
- BsmtFinType2: Rating of basement finished area (if multiple types)
- BsmtFinSF2: Type 2 finished square feet
- BsmtUnfSF: Unfinished square feet of basement area
- TotalBsmtSF: Total square feet of basement area
- Heating: Type of heating

- HeatingQC: Heating quality and condition
- CentralAir: Central air conditioning
- Electrical: Electrical system
- 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
- TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)
- Functional: Home functionality (Assume typical unless deductions are warranted)
- Fireplaces: Number of fireplaces
- FireplaceQu: Fireplace quality
- GarageType: Garage location
- GarageYrBlt: Year garage was built
- GarageFinish: Interior finish of the garage
- GarageCars: Size of garage in car capacity
- GarageArea: Size of garage in square feet
- GarageQual: Garage quality
- GarageCond: Garage condition
- PavedDrive: Paved driveway
- 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
- Fence: Fence quality
- MiscFeature: Miscellaneous feature not covered in other categories
- MiscVal: $Value of miscellaneous feature
- MoSold: Month Sold (MM)
- YrSold: Year Sold (YYYY)
- SaleType: Type of sale
- SaleCondition: Condition of sale

Target Column:

- SalePrice

Data Preprocessing Done

Checking for null values

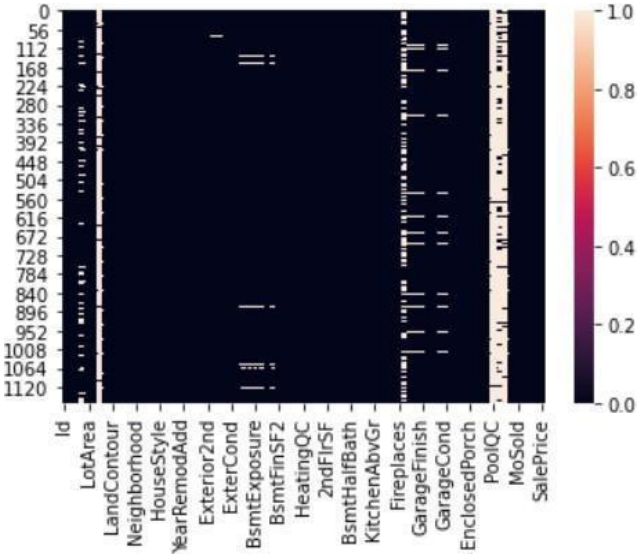
```
HDF[HDF.columns[HDF.isnull().any()]]
```

	LotFrontage	Alley	MasVnrType	MasVnrArea	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinType2	FireplaceQu	GarageType	GarageYr
0	NaN	NaN	None	0.0	Gd	TA	No	ALQ	Unf	TA	Attchd	197
1	95.0	NaN	None	0.0	TA	Gd	Gd	ALQ	Rec	TA	Attchd	197
2	92.0	NaN	None	0.0	Gd	TA	Av	GLQ	Unf	TA	Attchd	199
3	105.0	NaN	BrkFace	480.0	Gd	TA	No	BLQ	Unf	TA	Attchd	197
4	NaN	NaN	Stone	126.0	Gd	TA	No	ALQ	Unf	TA	Attchd	197
5	58.0	NaN	None	0.0	Gd	TA	Av	Unf	Unf	Gd	BuiltIn	200
6	NaN	NaN	BrkFace	180.0	Gd	TA	No	ALQ	Unf	Gd	Detchd	195
7	88.0	NaN	BrkCmn	67.0	TA	TA	No	Rec	BLQ	TA	Attchd	195
8	70.0	NaN	None	0.0	TA	TA	No	ALQ	GLQ	NaN	Detchd	196
9	80.0	NaN	None	0.0	TA	TA	No	Rec	Unf	NaN	Detchd	194
10	50.0	NaN	None	0.0	TA	TA	No	Unf	Unf	Gd	Detchd	193

```
HDF_test[HDF_test.columns[HDF_test.isnull().any()]]
```

	LotFrontage	Alley	MasVnrType	MasVnrArea	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinType2	Electrical	FireplaceQu	GarageType
0	86.0	NaN	Stone	200.0	Ex	TA	Gd	GLQ	Unf	SBrkr	Gd	Attchd
1	NaN	NaN	None	0.0	Gd	TA	Av	GLQ	Unf	SBrkr	Ex	Attchd
2	NaN	NaN	None	0.0	Gd	TA	Av	Unf	Unf	SBrkr	TA	Attchd
3	75.0	NaN	None	0.0	TA	TA	No	Rec	Unf	SBrkr	Gd	Attchd
4	86.0	NaN	Stone	74.0	Gd	TA	Mn	Unf	Unf	SBrkr	Gd	BuiltIn
5	21.0	NaN	None	0.0	Gd	TA	Av	BLQ	GLQ	SBrkr	NaN	NaN
6	35.0	NaN	BrkFace	80.0	Gd	TA	Gd	GLQ	Unf	SBrkr	NaN	Basment
7	107.0	NaN	Stone	436.0	Ex	TA	Gd	GLQ	Unf	SBrkr	Gd	Attchd
8	NaN	NaN	BrkFace	145.0	Gd	TA	Gd	GLQ	Unf	SBrkr	TA	Attchd
9	32.0	NaN	BrkFace	320.0	Ex	TA	No	GLQ	Unf	SBrkr	NaN	Attchd
10	60.0	NaN	None	0.0	TA	TA	No	Unf	Unf	SBrkr	NaN	Detchd

It’s observed that there are 18 columns in train dataframe with null values and 19 columns in test dataframe with null values.



Plotting a heatmap of null values revealed that in both training and testing datasets, Columns titled: Alley, PoolQC,MiscFeature,FireplaceQu,Fence have extremely sparse data with overwhelmingly high percentage of null values and therefore must be dropped.

The ID columns from test and train datasets were also dropped since they don't contribute to building a good model for predicting the target variable values.

Finding the null value percentage in each of the columns in Train and Test datasets

Using the following codes, the percentage of null values in each column was determined to understand how sparse the data is in those columns and to decide upon which imputation techniques to use inorder to eliminate null values.

```
for c in HDF[HDF.columns[HDF.isnull().any()]]:
    perct = HDF[c].isnull().sum()/1168*100
    print(f"null value % in {c} is: {perct}")
```

```
null value % in LotFrontage is: 18.32191780821918
null value % in MasVnrType is: 0.5993150684931506
null value % in MasVnrArea is: 0.5993150684931506
null value % in BsmtQual is: 2.5684931506849313
null value % in BsmtCond is: 2.5684931506849313
null value % in BsmtExposure is: 2.654109589041096
null value % in BsmtFinType1 is: 2.5684931506849313
null value % in BsmtFinType2 is: 2.654109589041096
null value % in GarageType is: 5.47945205479452
null value % in GarageYrBlt is: 5.47945205479452
null value % in GarageFinish is: 5.47945205479452
null value % in GarageQual is: 5.47945205479452
null value % in GarageCond is: 5.47945205479452
```

```
for c in HDF_test[HDF_test.columns[HDF_test.isnull().any()]]:
    perct = HDF_test[c].isnull().sum()/292*100
    print(f"null value % in {c} is: {perct}")
```

```
null value % in LotFrontage is: 15.41095890410959
null value % in MasVnrType is: 0.3424657534246575
null value % in MasVnrArea is: 0.3424657534246575
null value % in BsmtQual is: 2.3972602739726026
null value % in BsmtCond is: 2.3972602739726026
null value % in BsmtExposure is: 2.3972602739726026
null value % in BsmtFinType1 is: 2.3972602739726026
null value % in BsmtFinType2 is: 2.3972602739726026
null value % in Electrical is: 0.3424657534246575
null value % in GarageType is: 5.821917808219178
null value % in GarageYrBlt is: 5.821917808219178
null value % in GarageFinish is: 5.821917808219178
null value % in GarageQual is: 5.821917808219178
null value % in GarageCond is: 5.821917808219178
```

KNN imputation technique was used to impute values to missing data in LotFrontage, while the missing values in the rest of the columns were imputed with the most frequently occurring values of their respective columns.

Data Inputs- Logic- Output Relationships

The Datasets consist mainly of object data type variables and a few float and int data type variables. The relationships between the independent variables and dependent variable were analysed

Features like Lot area, Lot Frontage, Overall Quality, Overall Condition, Basement Finishing, Total Basement Surface Area, first and 2nd Floor square feet, Garage capacity,Total rooms have a positive linear relationship, therefore increase in their values leads to increase in SalePrice. Whereas Age of Housem Remodellling age Garrage age have a linear negative relationship and therefore increase in their values leads to a decrease in SalePrice.

Underlying Assumptions

```
HDF.describe()
```

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	...	WoodDeck
count	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	...	1168.0000
mean	56.767979	70.988470	10484.749144	6.104452	5.595890	1970.930651	1984.758562	101.696918	444.726027	46.647260	...	96.2063
std	41.940650	22.437056	8957.442311	1.390153	1.124343	30.145255	20.785185	182.218483	462.664785	163.520016	...	126.1589
min	20.000000	21.000000	1300.000000	1.000000	1.000000	1875.000000	1950.000000	0.000000	0.000000	0.000000	...	0.0000
25%	20.000000	60.000000	7621.500000	5.000000	5.000000	1954.000000	1966.000000	0.000000	0.000000	0.000000	...	0.0000
50%	50.000000	70.988470	9522.500000	6.000000	5.000000	1972.000000	1993.000000	0.000000	385.500000	0.000000	...	0.0000
75%	70.000000	79.250000	11515.500000	7.000000	6.000000	2000.000000	2004.000000	150.000000	714.500000	0.000000	...	171.0000
max	190.000000	313.000000	164660.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	1474.000000	...	857.0000

8 rows × 37 columns

Based on the statistical information above, the following observations were made:

- Big difference between max value and 75% in SalePrice,MSSubClass,LotFrontage,LotArea,BsmtFinSF 1,BsmtF inSF2, etc indicates presence of outliers.
- A higher std than mean in columns: MasVnrArea,BsmtFinSF1,BsmtFinSF2,WoodDeckSF,O penPorc hSF,EnclosedPorch,3SsnPorch etc indicates presence of skewness.
- An Anomaly is displayed in the relationship between age of house and SalePrice. There is a general negative relationship between House age and Sale Price, ie. increase in age leads to a decrease in SalePrice, however, houses built between 1880 and 1900 sold for the highest. The assumption made in this regard is that those houses were sold for the highest amount because of their antiquity value.

Hardware and Software Requirements and Tools Used

Hardware Used:

- Processor AMD Ryzen 9 5900HX(8 Cores 16 Logical Processors)
- Physical Memory: 16.0GB (3200MHz)
- GPU: Nvidia RTX 3060 (192 bits), 6GB DDR6 VRAM, 3840 CUDA cores.

Software Used:

- Windows 10 Operating System
- Anaconda Package and Environment Manager: Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data- science packages suitable for Windows and provides a host of tools and environment for conducting Data Analytical and Scientific works. Anaconda provides all the necessary Python packages and libraries for Machine learning projects.
- Jupyter Notebook: The Jupyter Notebook is an open-source web application that allows data scientists to create and share documents that integrate live code, equations, computational output, visualizations, and other multimedia resources, along with explanatory text in a single document.
- Python3: It is open source, interpreted, high level language and provides great approach for object-oriented programming. It is one of the best languages used for Data Analytics And Data science projects/application. Python provides numerous libraries to deal with mathematics, statistics and scientific function.
- Python Libraries used:
  - Pandas: For carrying out Data Analysis, Data Manipulation, Data Cleaning etc
  - Numpy: For performing a variety of operations on the datasets.

- matplotlib.pyplot, Seaborn: For visualizing Data and various relationships between Feature and Label Columns
- Scipy: For performing operations on the datasets
- Statsmodels: For performing statistical analysis
- sklearn for Modelling Machine learning algorithms, Data Encoding, Evaluation metrics, Data Transformation,Data Scaling, Component analysis,Feature selection etc

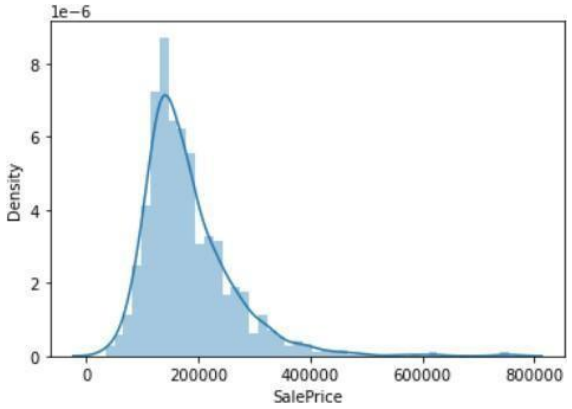
Exploratory Data Analysis

Visualizations

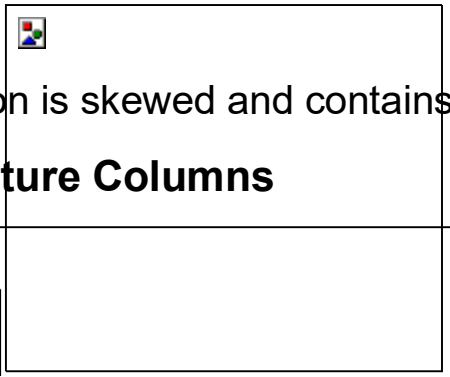
Barplots,Distplots,Boxplots,Countplots,lineplots were used to visualise the data of all the columns and their relationships with Target variable.

Univariate Analysis

Analyzing the Target Class

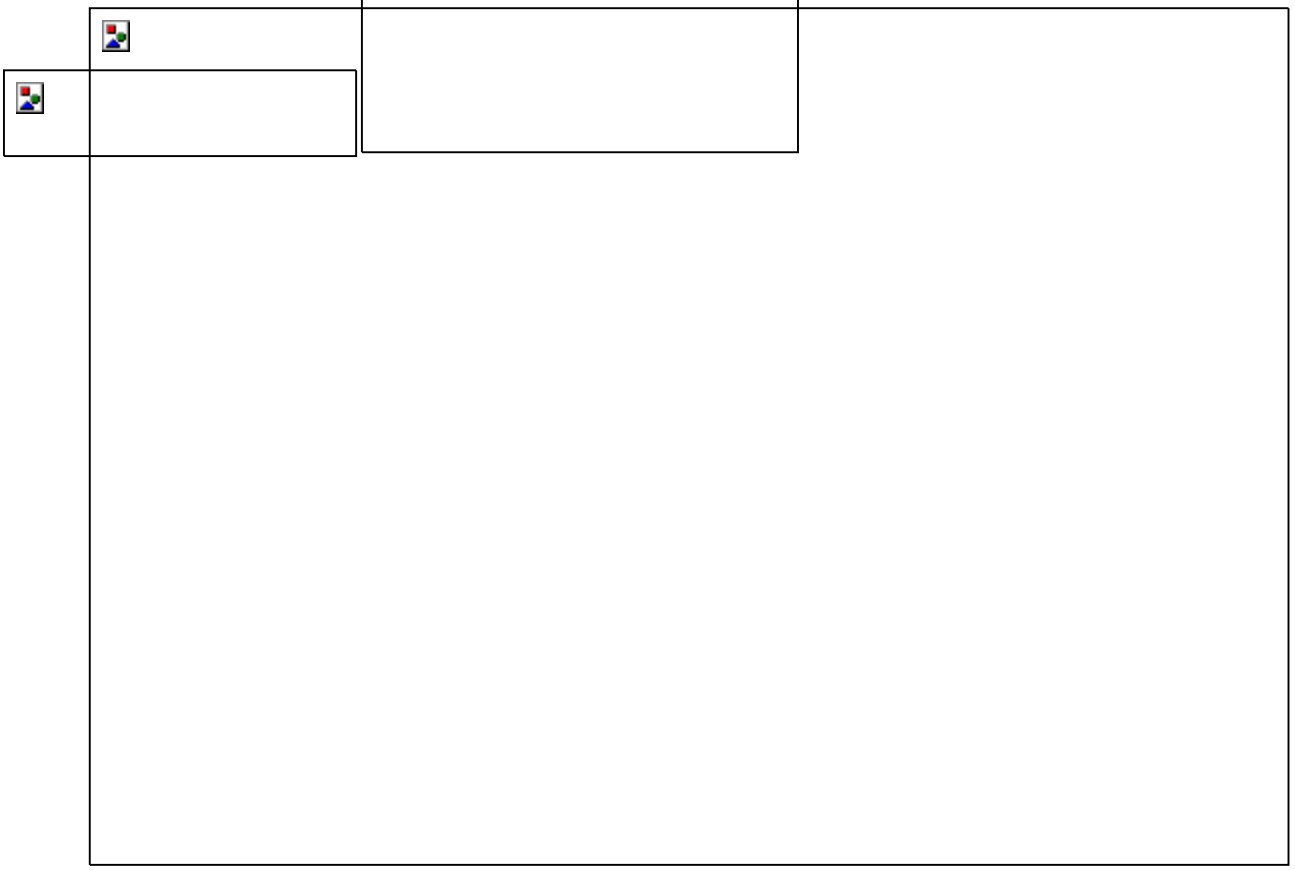


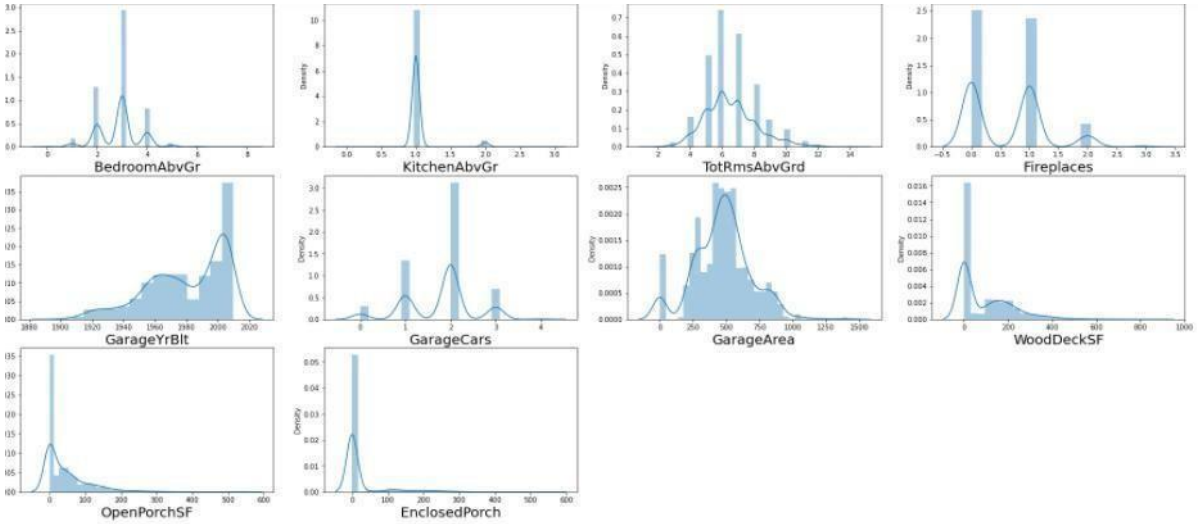
From the graph above it is observed that the Price data forms a continuous distribution with mean of 181477.00 and tails off from 400000 mark.



Distribution is skewed and contains outliers.

Analyzing the Feature Columns

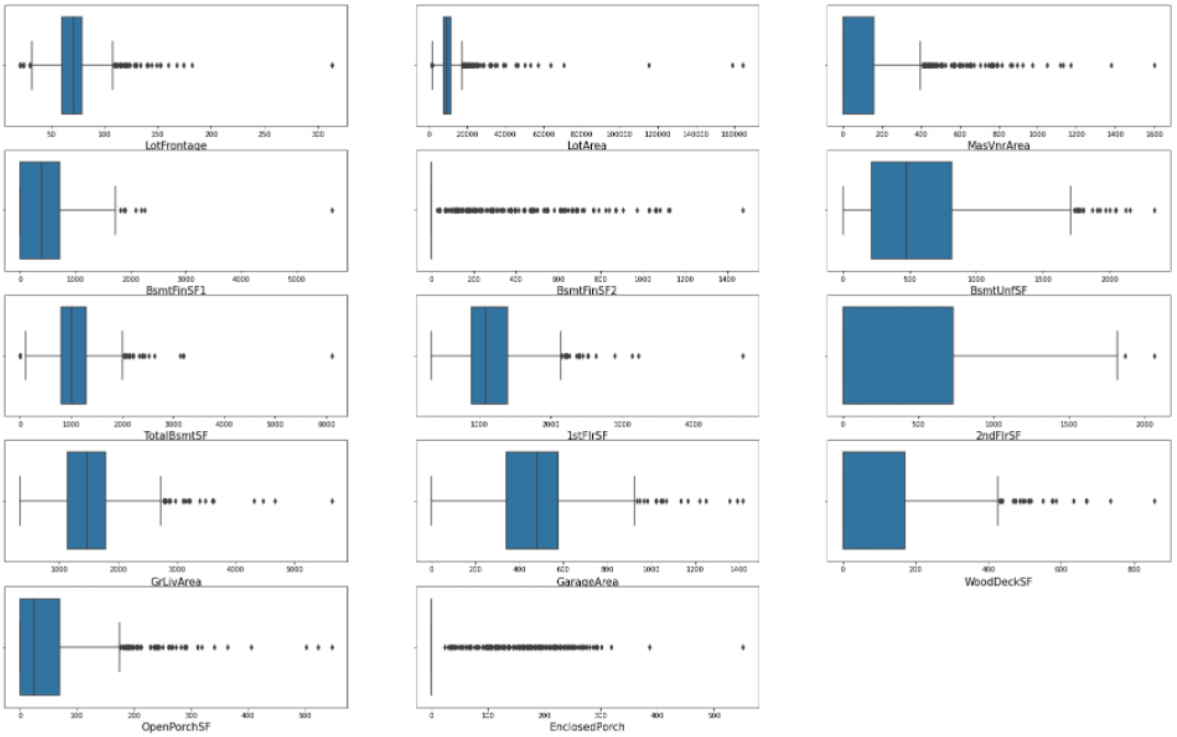




LotFrontage,LotArea,MasVnrArea,BsmtFinSF1,BsmtFin SF2,Bs mtUnFSF,TotalBsmtSF,1stFlrSF,2ndFlrSF,GrLivArea,Wo odDeck SF,OpenPorchSF,EnclosedPorch are skewed and contain outliers

```
MSSubClass       1.422619
LotFrontage      2.710383
LotArea         10.659285
OverallQual      0.175082
OverallCond      0.580714
YearBuilt       -0.579204
YearRemodAdd    -0.495864
MasVnrArea       2.835718
BsmtFinSF1       1.871606
BsmtFinSF2       4.365829
BsmtUnfSF        0.909057
TotalBsmtSF      1.744591
1stFlrSF         1.513707
2ndFlrSF         0.823479
LowQualFinSF     8.666142
GrLivArea        1.449952
BsmtFullBath     0.627106
BsmtHalfBath     4.264403
FullBath         0.057809
HalfBath         0.656492
BedroomAbvGr     0.243855
KitchenAbvGr     4.365259
TotRmsAbvGrd     0.644657
Fireplaces       0.671966
GarageYrBlt     -0.708074
GarageCars      -0.358556
GarageArea       0.189665
WoodDeckSF       1.504929
OpenPorchSF      2.410840
EnclosedPorch    3.043610
3SsnPorch        9.770611
ScreenPorch      4.105741
PoolArea        13.243711
MiscVal         23.065943
MoSold           0.220979
YrSold           0.115765
SalePrice        1.953878
dtype: float64
```

Considerable skewness exists in columns



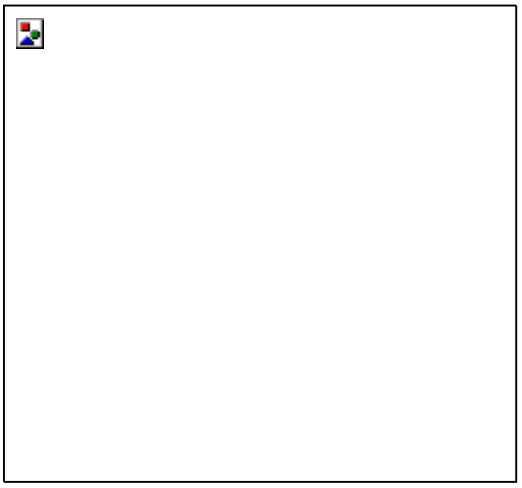
There is a considerable number of outliers in the columns. However, they will not be removed, since we have a very small dataset to work with and removing outliers results in 13.86% of loss in data.

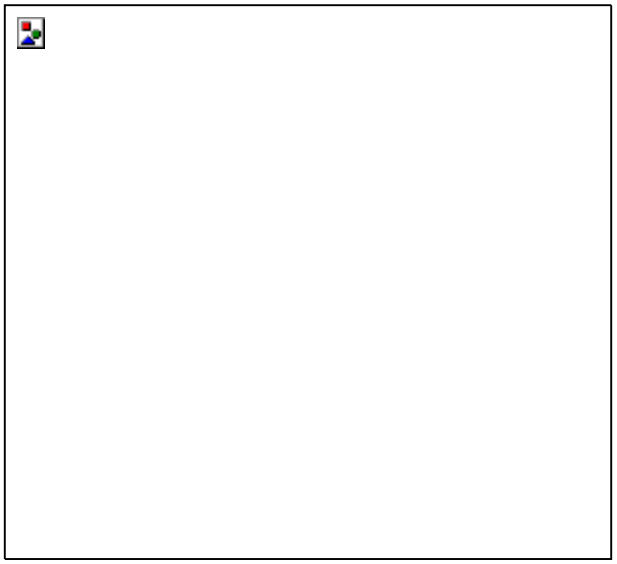
Normalizing Data Distribution using PowerTransformer

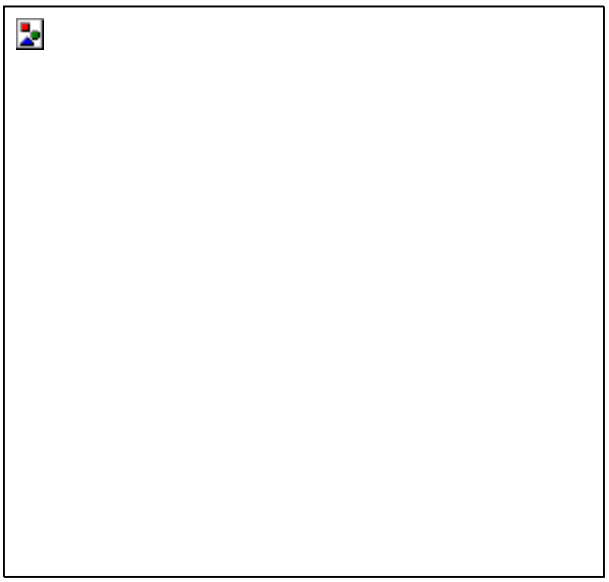
The skewness in Data Distributions of the feature columns was reduced using the Yeo-Johnson Power transformer method

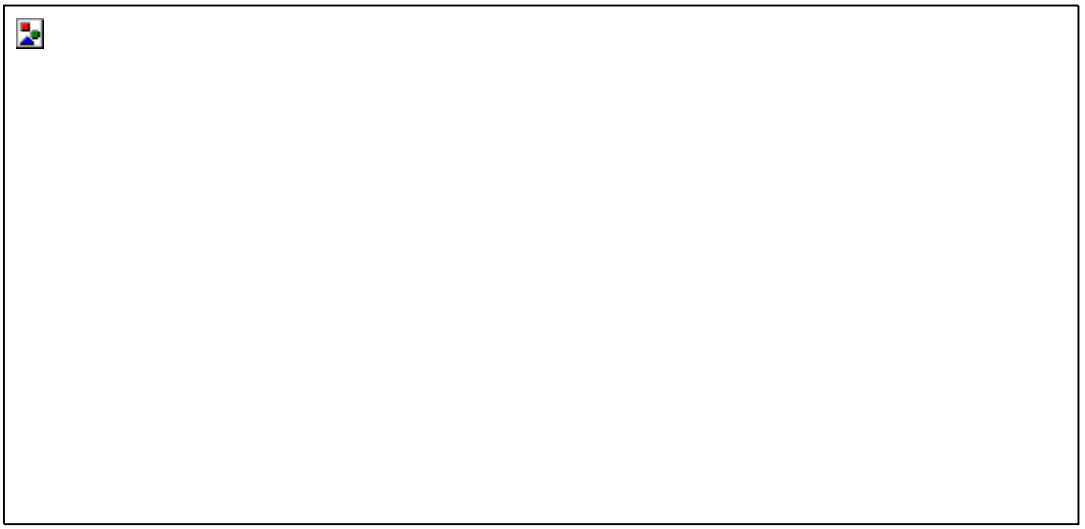
```
LotFrontage      0.161368
LotArea          0.032509
MasVnrArea       0.439526
BsmtFinSF1      -0.404528
BsmtFinSF2       2.394737
BsmtUnfSF       -0.284390
TotalBsmtSF      0.286779
1stFlrSF        -0.002391
2ndFlrSF         0.280208
GrLivArea       -0.000054
GarageArea      -0.320370
WoodDeckSF       0.113026
OpenPorchSF     -0.002749
EnclosedPorch    2.022616
House_Age        0.579204
Remod_Age        0.495864
Garage_age       0.708074
dtype: float64
```

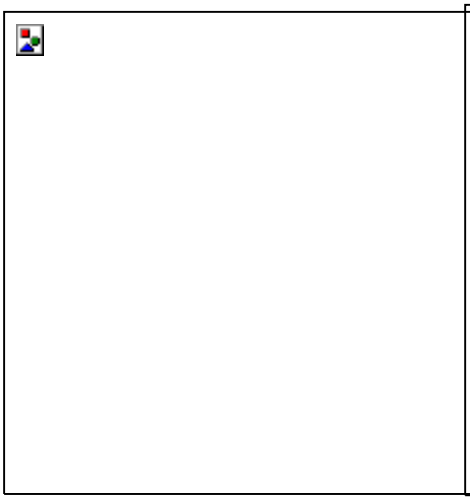
A lot of skewness has been removed.

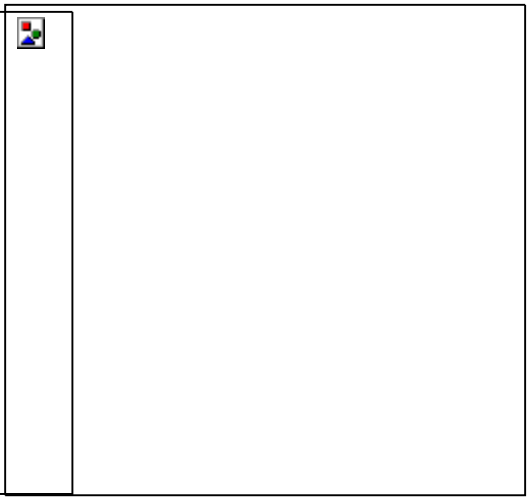












Following observations are made from above graphs:

- Residential Low Density is the most common zoing classification
- Most common Street Type is 'Pave'
- Regular is the most common LotShape, followed by Slightly irregular
- Most Properties have Near Flat/Level LandContour
- All public Utilities are available
- Inside lot is the most common Lot configuration
- Slope of property land is most commonly gentle
- Most Housing properties are situated in Neighborhoods of North Ames, followed by College Creek,Edwards and Old Town
- Most Housing properties are in proximity to Normal conditions
- Most Housing properties are of Single-family Detached type
- Most Housing properties 1 storied and 2 storied
- Most Houses have Gable roof style

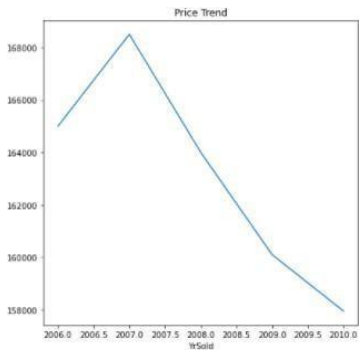
- Most Houses have roofs made of Standard (Composite) Shingle
- Vinyl Siding is the most common exterior covering used
- Most Houses don't have a Masonry veneer type while some have Brick Face
- The quality of the material on the exterior is most commonly average/typical
- The present condition of the material on the exterior is most commonly average/typical
- Two of the most common foundation types are Cinder Block and Poured Concrete
- The height of the basement is usually either Typical (80- 89 inches) or Good (90-99 inches)
- The general condition of the basement is commonly Typical with slight dampness
- Basements most commonly have no exposure
- Most houses have Basements that are usually unfinished followed by houses with basements having Good Living Quarters
- Most houses have Gas forced warm air furnace heating arrangement
- Most houses have Excellent Heating quality and condition
- Most houses have Central air conditioning
- Most houses have Standard Circuit Breakers & Romex Electrical system
- Most houses have Most houses have Typical/Average and Good Kitchen quality
- Most houses have Typical Functionality
- Most houses have a Garage Attached to home
- Most houses have an Unfinished garage
- Garage is usually Typical/Average
- Garage condition is usually Typical/Average

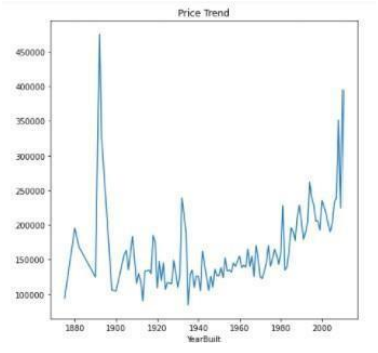
- Most houses have a Paved driveway
- Warranty Deed - Conventional is the most common Type of sale

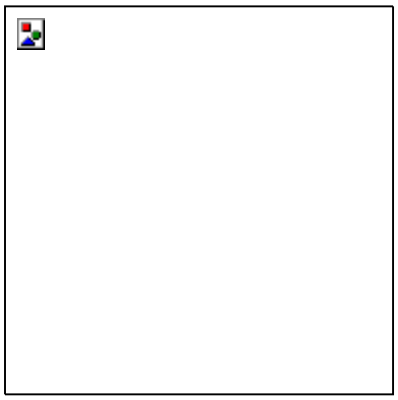
- Condition of sale is most commonly a Normal Sale

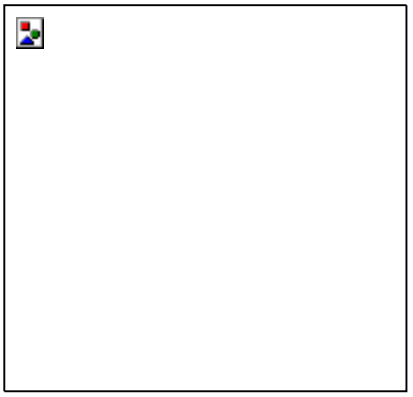
Bivariate Analysis

Interpreting Relationship between Dependent Variable and Independent Variable Columns









From the graph above, it is observed that:

- Sales value peaked between 2006 and 2007 and there has been a general downward trend in sales price since then
- Sales value is higher for houses built after 1990s implying the lesser the age of the house, the higher its value, however houses built between 1880 and 1900 sold for the highest, this could be because of their antiquity value.
- Sales value is higher for houses which were remodelled more recently.

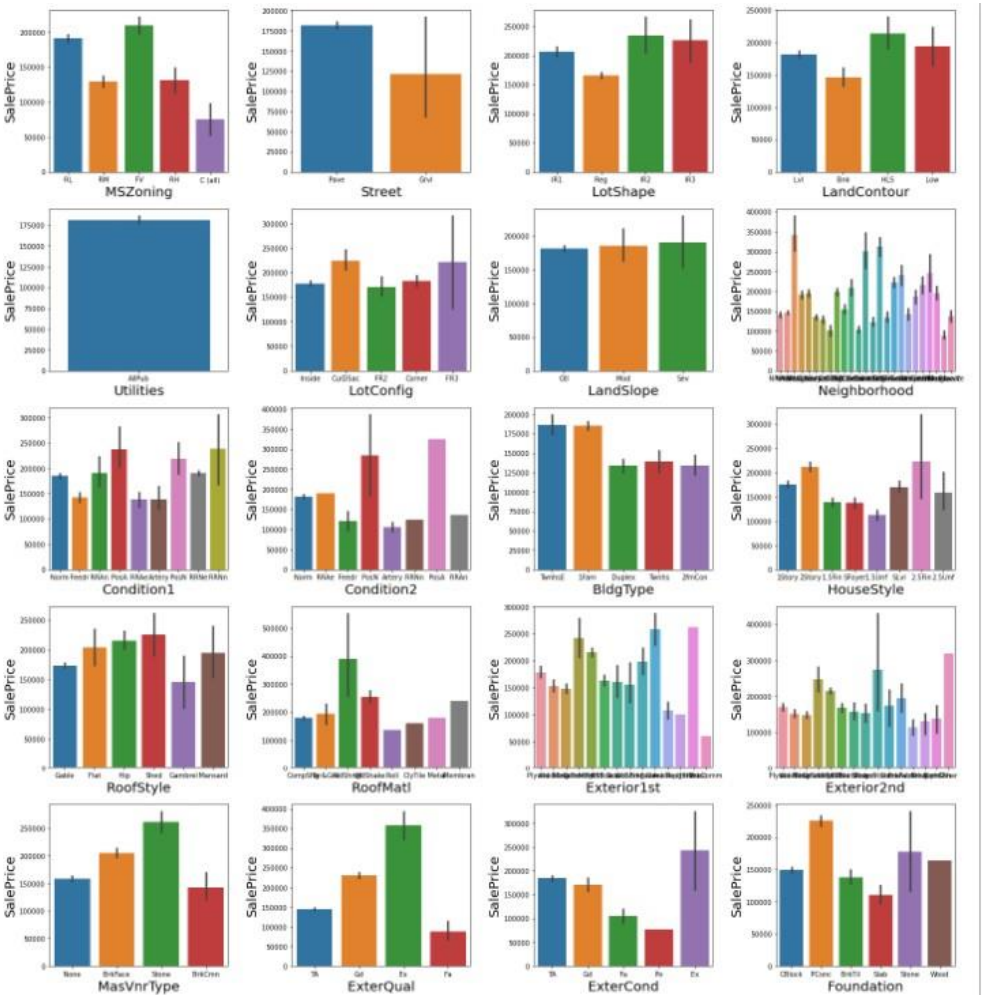
- Sales value is higher for houses whose Garage was built more recently.

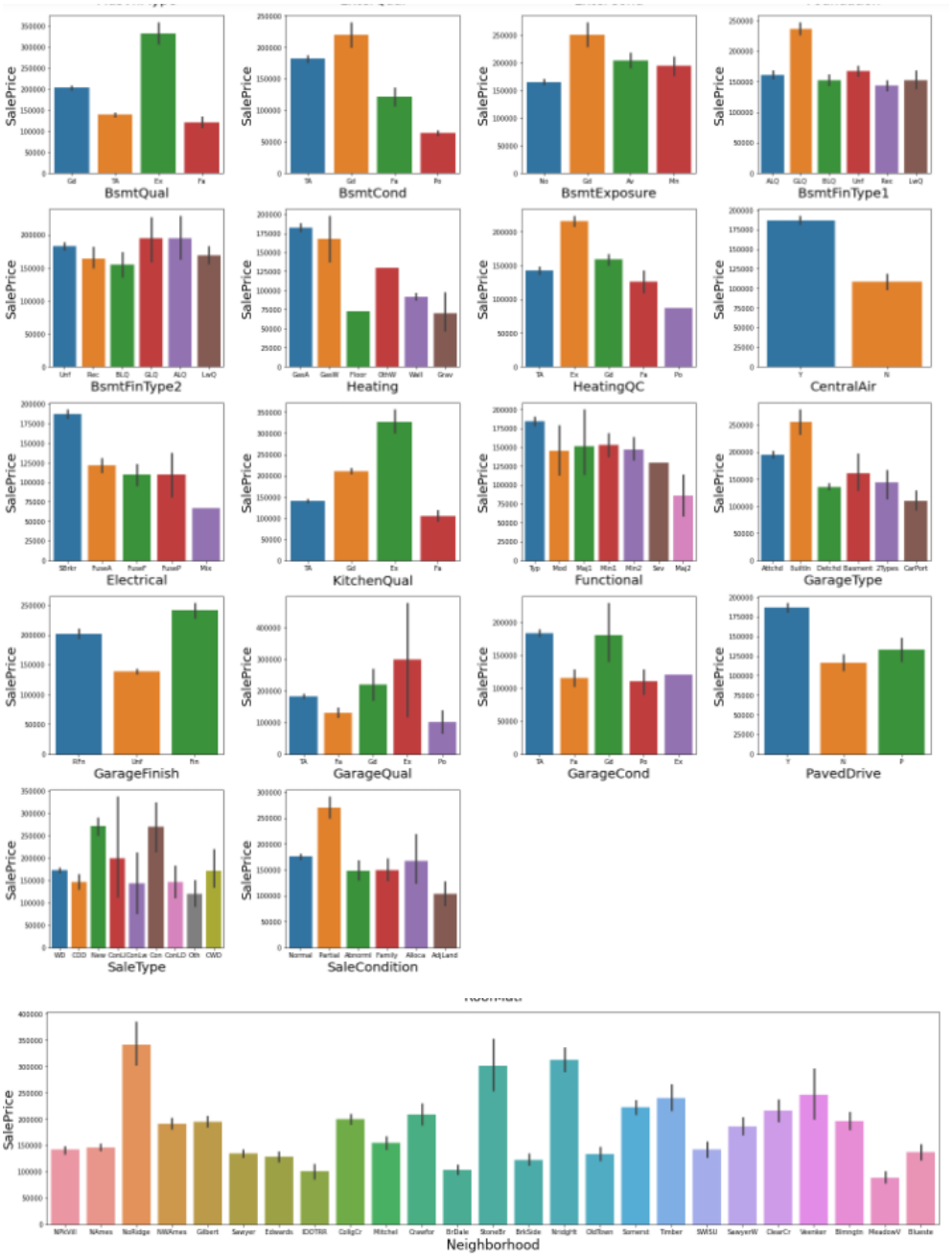
Following Observations are made from graphs above:

- 1 story and 2 and 2.5 story houses built in 1946 and newer fetch the highest amount in sales.
- Houses with LotFrontage between 100 ft and 200 ft are sold for the highest amount.
- Houses wtih Lot area upto 25000 sqft fetch the highest amount.
- There is a Linear positive relation between Overall Quality and SalesPrice

- There is a Linear positive relation between Overall Condition and SalesPrice
- There is a Linear positive relation between Masonry veneer area and SalesPrice
- Most Sales were done for Type 1 Finished basement with area upto 2500 sqft
- There is a Linear positive relation between Type 2 Finished basement area and SalesPrice
- There is a Linear positive relation between Total Basement area and SalesPrice
- There is a Linear positive relation between Total 1st area and 2nd floor area and SalesPrice
- There is a Linear positive relation between low Quality finished square feet and SalesPrice
- There is a Linear positive relation between Above grade living area square feet and SalesPrice
- There is a Linear negative relation between basment half bath and SalesPrice
- There is a Linear positive relation between Full Bathroom and SalesPrice
- There is a Linear positive relation between Total rooms above grade and SalesPrice
- There is a Linear positive relation between Fireplaces and SalesPrice
- There is a Linear positive relation between Garage Car capacity and SalesPrice
- There is a Linear positive relation between Garage area and SalesPrice
- Sales Prices peaked between 0-400 square feet area for Wooden Deck
- Sales Prices peaked between 0-300 square feet area for Open Porch
- Sales Price and Enclosed Porch area have a positive relation
- Sales Price and 3 season Porch area have a positive relation
- Sales Price and screen Porch area have a positive relation
- Sales Price and Month Sold have a positive relation
- Sales Price and Month Sold have a positive relation
- Sales Price and house age have a negative relation

- Sales Price and remodelling age have a negative relation
- Sales Price and Garage age have a negative relation





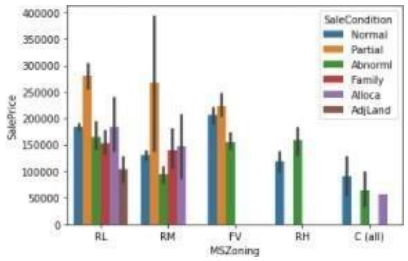
Following Observations are made from graphs above:

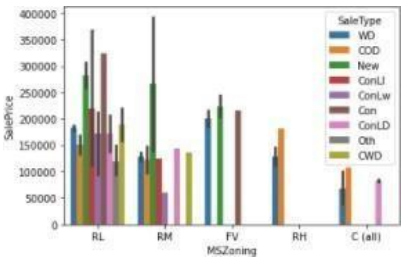
- Saleprice is highest for Floating Village and Low density Residen tial zones
- Saleprice is highest for housing properties near paved streets
- Saleprice is highest for irregular lot shapes
- Hill side properties sell for the highest amount

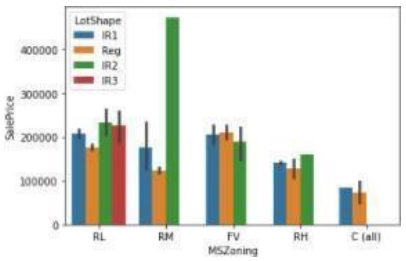
- Utilities & Landslope columns don'tshow a strong relationship with Sales Price
- Housing Properties in Northridge, Stone Brook,Northridge Heig hts,Timberland, Somerset,Veenker fetch the highest Sales amou nt
- Cul-de-sac and 3 sided frontage lot configurations fetch the high est Sales amount
- Proximity to Railroads, Off-site features like parks etc fetch the highest Sales amount
- Townhouse and Single-family Detached are the most valued
- Two story and Two and one-half story: 2nd level finished sell for the highest amount
- Houses with Wood Shingle Roofs sell for the highest amount
- Houses with Exterior covering of Cement Board,Stone,Imitation Stucco sell for the highest amount
- Houses with Stone Mason veneer type sell for the highest amou nt
- Houses with Excellent exterior material quality sell for the highe st amount
- Houses with Excellent exterior material condition sell for the hig hest amount
- Houses with Poured Concrete and stone foundation types sell f or the highest amount
- Houses with Excellent (100+ inches) height of the basement sell for the highest amount
- Houses with Excellent Basement Condition sell for the highest a mount

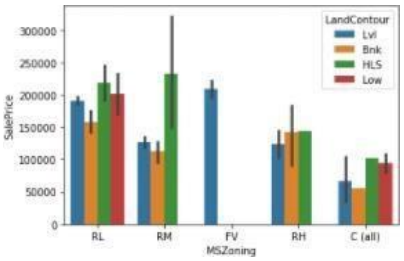
- Houses with Good Basement Exposure sell for the highest amount
- Houses with Good and Average Living Quarters in Basement sell for the highest amount
- Houses with Gas forced warm air furnace and Gas hot water he ating systems sell for the highest amount
- Houses with Excellent Heating quality and condition sell for the highest amount
- Houses with Central Air Conditioning sell for the highest amoun t
- Houses with Standard Circuit Breakers & Romex sell for the high est amount
- Houses with Excellent Kitchen Quality sell for the highest amou nt
- Houses with Built in Garages sell for the highest amount
- Houses with Good / Typical Garage condition sell for the highest amount
- Houses with Finished Garage sell for the highest amount
- Houses with excellent Garage Quality sell for the highest amoun t
- Houses with Paved Driveway sell for the highest amount
- Homes just constructed and sold, Contract 15% Down payment regular terms sell for the highest amount
- New Homes(not completed when last assessed) sell for the high est amount

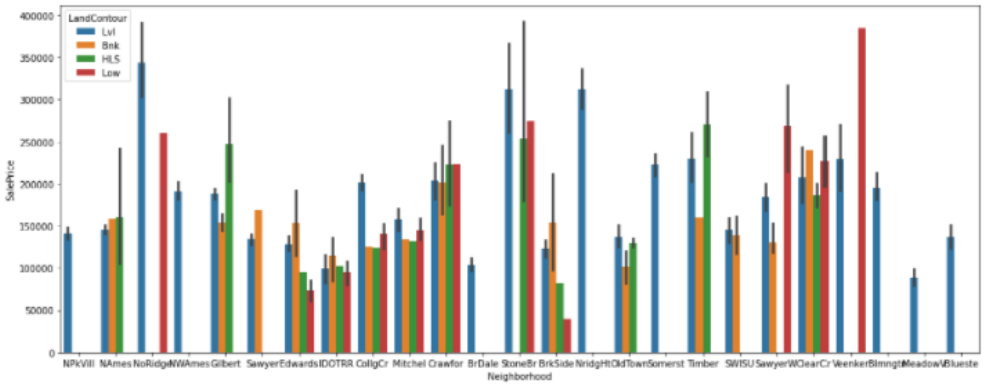
Multivariate Analysis

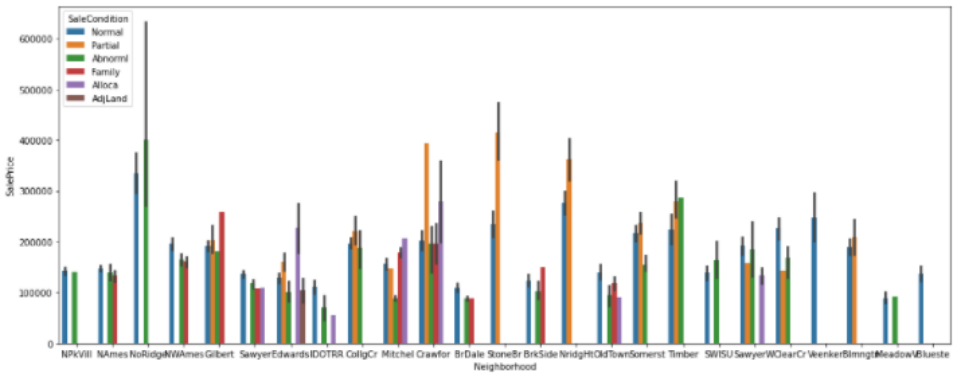


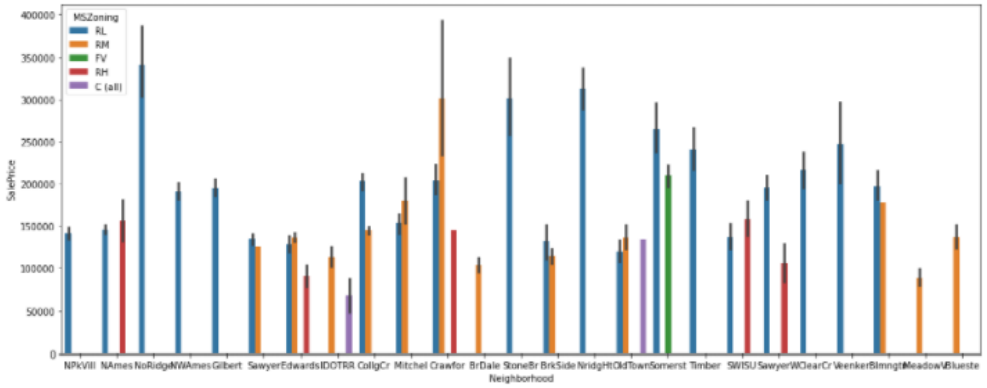












Following Observations are made from graphs above:

- New Homes are the most popular in all types of zoning
- New hourses and Low interest contract are the most popular sale types in low density, medium density and floating village residentials
- Partially irregular and irregular plot shapes are most popular in low and medium residential zones

- Low density and medium density zones settled near hillsides and depressions are mostly sold at higher prices, whereas floating villages are settled in flat regions, and high density zones settle near banked regions sell for the highest prices
- Most housing properties established in levelled regions in North Ridge sell for the highest.
- Most Housing properties in levelled regions of Stone Brook sell for highest followed by banked region and hillsides.
- Houses in levelled region of NorthRidge heights sell for the most while housing properties in depressed regions of Veenker sell for the highest prices.
- Most housing properties that are newly established in Crawford,Stone Brook, Timberlane,North Ridge Heights,Bloomington Heights sell for the highest.
- Most Housing properties in North Ridge sell for trade, foreclosure, short sale and normal sale in North Ridge.
- Most houses sold in North Ridge,North Ridge Heights, Somerset,TimberLane,Veenker, Bloomington Heights are in low density residential zones.
- North Ames has more houses sold in High density residential zones, while Crawford has more houses sold in medium density residential zones.
- Warranty Deed - Conventional,Home just constructed and sold,Contract Low InterestCourt Officer Deed/Estate are the most common sale types.
- Excelent quality of Gas forced warm air furnace and Gas hot water heating systems fetches the highest amount of money.
- 3 sided Frontage properties with Regular plot shape sell for the highest
- Two and one-half story: 2nd level finished housing properties sell for the highest.
- Single-family Detached housing properties sell for the

highest in most neighborhoods

Feature Engineering:

In order to better understand the relationships between age of a housing property and SalePrice, following columns were created based on data of existing columns:

House Age data was extracted from YearBuilt, and was placed in new column titled: House_Age, House Remodeling Age was extracted from YearRemodAdd, and was placed in new column titled: Remod_Age and age of Garage was extracted from GarageYrBuilt nd placed in a new column titled Garage_Age.

Encoding the Categorical Columns

Before Proceeding with finding the correlations of the columns, The data of the categorical columns needs to be encoded using LabelEncoder.

```
from sklearn.preprocessing import LabelEncoder
```

```
labenc = LabelEncoder()
```

```
for col in HDF[HDF.columns[HDF.dtypes == 'object']]:
    HDF[col] = labenc.fit_transform(HDF[col])
```

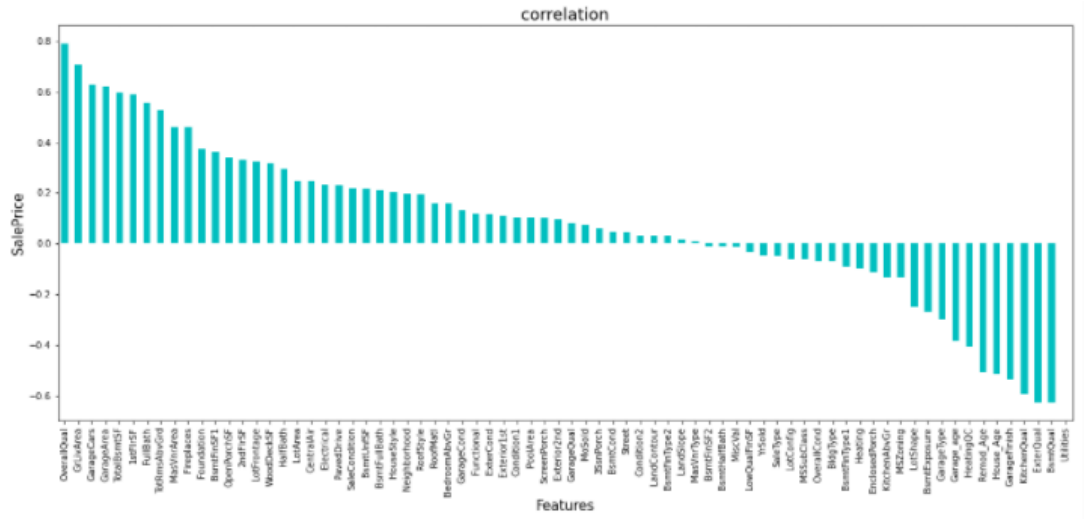
```
HDF['YrSold'] = HDF.YrSold.map({2007:2,2009:4,2006:1, 2008: 3, 2010: 5}) # encoding years in YrSold Column
```

```
HDF['Utilities'] =HDF.Utilities.map({0:1})
```

```
HDF.dtypes[HDF.dtypes != 'object']
```

Finding the correlations

Visualizing correlation of Feature Columns with Label Column



OverallQual,GrLiveArea,GarageCars,GarageArea,TotalBsmtSF,1st FlrSF,Full Bath,TotRmsAbvGrd,MasVnrArea,FirePlaces have the strongest positive correlation with SalePrice while BsmtQual,ExterQual,KitchenQual,GarageFinish,House_age,Remod_age,H eatingQC,Garage_age have the strongest negative correlation with SalePrice.

Model/s Development and Evaluation

Feature Selection

Features were first checked for presence of multicollinearity and then based on Principle Component Analysis and based on the respective ANOVA f-score values, the feature columns were selected that would best predict the Target variable, to train and test machine learning models.

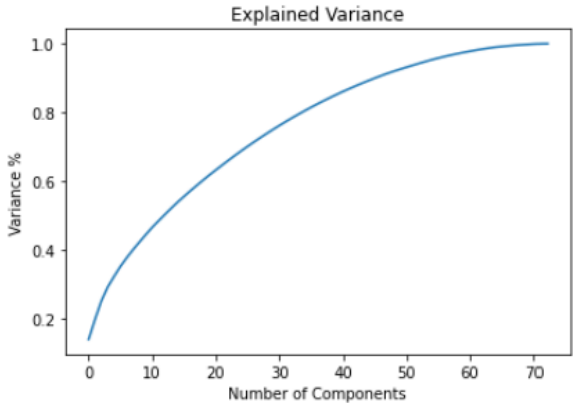
```
In [167]: from statsmodels.stats.outliers_influence import variance_inflation_factor

In [168]: vif = pd.DataFrame()

In [169]: vif["Features"] = X.columns
          vif['vif'] = [variance_inflation_factor(scaled_X,i) for i in range(scaled_X.shape[1])]

In [170]: vif
```

30	BsmtFinType2	4.019272
31	BsmtFinSF2	4.504940
32	BsmtUnfSF	5.323098
33	TotalBsmtSF	6.927810
34	Heating	1.317820
35	HeatingQC	1.696584
36	CentralAir	1.725971
37	Electrical	1.379424
38	1stFlrSF	17.623835
39	2ndFlrSF	17.357365
40	LowQualFinSF	1.532083
41	GrLivArea	28.469724

BsmtFinSF1,1stFlrSF,2ndFlrSF,GrLivArea,GarageCars,GarageArea,Ho use_Age exhibit high multicollinearity.

```
from sklearn.decomposition import PCA
```

```
pca = PCA()
principleComponents = pca.fit_transform(scaled_X)
plt.figure()
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('Number of Components')
plt.ylabel('Variance %')
plt.title('Explained Variance')
plt.show()
```



Based on The Principle Component Analysis it was determined that 70 components explain around 95% variance in Data.

```
from sklearn.feature_selection import SelectKBest, f_classif
bestfeat = SelectKBest(score_func = f_classif, k = 'all')
fit = bestfeat.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
fit = bestfeat.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
dfcolumns.head()
featureScores = pd.concat([dfcolumns,dfscores],axis = 1)
featureScores.columns = ['Feature', 'Score']
print(featureScores.nlargest(75,'Score'))
```

	Feature	Score
14	OverallQual	5.303071
65	MiscVal	3.564855
22	ExterQual	3.514221
41	GrLivArea	2.956955
25	BsmtQual	2.876879
48	KitchenQual	2.617125
54	GarageCars	2.578547
44	FullBath	2.435854
55	GarageArea	2.242545
53	GarageFinish	2.187163
70	House_Age	2.133300
33	TotalBsmtSF	2.070692
38	1stFlrSF	2.060541
4	Street	1.835751
71	Remod_Age	1.813783
34	Heating	1.707885
49	TotRmsAbvGrd	1.656866
1	MSZoning	1.640044
60	OpenPorchSF	1.613273
51	Fireplaces	1.591973
36	CentralAir	1.557680
72	Garage_age	1.535466
24	Foundation	1.528516
5	LotShape	1.407526
3	LotArea	1.405080
29	BsmtFinSF1	1.385210
35	HeatingQC	1.358939
45	HalfBath	1.337597
9	Neighborhood	1.281079
21	MasVnrArea	1.259874
59	WoodDeckSF	1.246439
27	BsmtExposure	1.214154
2	LotFrontage	1.212723
26	BsmtCond	1.190161

Using SelectKBest and f_classif for measuring the respective ANOVA f-score values of the columns, the best 70 features were selected.

Using StandardScaler, the features were scaled by resizing the distribution values so that mean of the observed values in each feature column is 0 and standard deviation is 1.

From sklearn.model_selection's train_test_split, the data was divided into train and test data. Training data comprised 75% of total data where as test data comprised 25% based on the best random state that would result in best model accuracy.

Inorder to find the best random state for the train and test split, the following code was used:

```
from sklearn.ensemble import RandomForestRegressor
maxAcc = 0
maxRS=0
for i in range(1,100):
    x_train,x_test,y_train,y_test = train_test_split(scaled_x_best,y,test_size = .25, random_state = i)
    modRF =  RandomForestRegressor()
    modRF.fit(x_train,y_train)
    pred = modRF.predict(x_test)
    acc  = r2_score(y_test,pred)
    if acc>maxAcc:
        maxAcc=acc
        maxRS=i
print(f"Best Accuracy is: {maxAcc} on random_state: {maxRS}")
```

Best Accuracy was: 0.9174093624145732 on random_state: 72

```
x_train,x_test,y_train,y_test = train_test_split(scaled_x_best,y,test_size = .25, random_state =72)
```

The model algorithms used were as follows:

- **Ridge**: Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. Since the features have multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values. Ridge shrinks the parameters. Therefore, it is used to prevent multicollinearity.
- **DecisionTreeRegressor**: Decision Tree solves the problem of machine learning by transforming the data into a tree representation. Each internal node of the tree representation denotes an attribute and each leaf node denotes a class label.A decision tree does not require normalization of data. A decision tree does not require normalization of data.
- **XGBRegressor**: XGBoost uses decision trees as base learners; combining many weak learners to make a strong learner. As a result it is referred to as an ensemble learning method since it uses the output of many models in the final prediction. It uses the power of parallel processing, supports regularization, and works well in small to medium dataset.

- **RandomForestRegressor**: A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. A random forest produces good predictions that can be understood easily. It reduces overfitting and can handle large datasets efficiently. The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.

- **Support Vector Regressor**: SVR works on the principle of SVM with few minor differences. Given data points, it tries to find the curve. But since it is a regression algorithm instead of using the curve as a decision boundary it uses the curve to find the match between the vector and position of the curve. Support Vectors helps in determining the closest match between the data points and the function which is used to represent them. SVR is robust to the outliers. SVR performs lower computation compared to other regression techniques.

```
from sklearn.linear_model import Ridge
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.svm import SVR
```

```
from sklearn.metrics import r2_score,mean_squared_error
```

```
rf = RandomForestRegressor()
dt = DecisionTreeRegressor()
xg = XGBRegressor()
SV= SVR()
r=Ridge()
```

```
from sklearn.metrics import r2 score,mean squared error

rf                          =
RandomForestRegressor()
dt                          =
DecisionTreeRegressor()
xg = XGBRegressor()
SV= SVR()
r=Ridge()
```

Training the Models

```
nf.fit(x_train,y_train)
xg.fit(x_train,y_train)
SV.fit(x_train,y_train)
r.fit(x_train,y_train)
dt.fit(x train,y train)

DecisionTreeRegressor()
```

All models have been trained.

Ridge Regression Model

```
]: y_r_pred = r.predict(x_test)
```

```
]: 0.8581092136367113
```

Mean Squared Error

```
•]: rean_s q u ar ed_e r re r (y _ test, y_r_p re d}
]: 860008392.6691393
```

Root Mean Squared Error

```
]: | np.sqrt(mean_squared_error(y_test,y_r_pred))
]: 29325.8996907024
```

XGB Regression Model

```
y_xg_p•ed : xg. predict(x_test)
```

```
0.899004570051692
```

Mean £lquared Error

```
mean_squared_erro
612139234.7096407
```

Root Mea n 3qua red Error

```
np.sqrt(mean_squa
```

Random Forest Regression Model

```
]: 0.9127773707952838
```

Mean 3q uared Error

```
]: 528661480.2082185
```

```
]: n p. sq rt l me an_s q ua re d_e rr s \ y_-e s,t y_rf-Are d) !
]: 22992.63969639455
```

Support Vector Regression Model

```
6*&*syzJ11.4Ues64
```

Root Mea n Squa red Error

```
bUyh'.s4.’’s*6we4&y
```

Mean Squared Erro

```
np.sqrt(mean_squa
```

Decision Tree Regression Model

```
y_dt_pred = dt.predict(x_test)
```

R2 Score

```
r2_score(y_test,y_dt_pred)
```

```
0.7788722311025055
```

Mean Squared Error

```
mean_squared_error(y_test,y_dt_pred)
```

```
1340268399.2260275
```

Root Mean Squared Error

```
np.sqrt(mean_squared_error(y_test,y_dt_pred))
```

```
36609.67630594441
```

Mean Squared Error and Root Mean Squared Error metrics were used to evaluate the Model performance. The advantage of MSE and RMSE being that it is easier to compute the gradient. As, we take square of the error, the effect of larger errors become more pronounced than smaller error, hence the model can now focus more on the larger errors.

Cross validation is a technique for assessing how the statistical analysis generalises to an independent data set.It is a technique for evaluating machine learning models by training several models on subsets of the available input data and evaluating them on the complementary subset of the data. Using cross-validation, there are high chances that we can detect over-fitting with ease.

Model Cross Validation scores were then obtained for assessing how the statistical analysis generalises to an independent data set. The models were evaluated by training several models on subsets of the available input data and evaluating them on the complementary subset of the data.

```
from sklearn.model_selection import ShuffleSplit,cross_val_score
```

Ridge Regression

```
cross_val_score(r,scaled_x_best,y,cv=5).mean()
```

```
0.77241873730882
```

Random Forest Regression

```
cross_val_score(rf,scaled_x_best,y,cv=5).mean()
```

```
0.8445427237063778
```

XGB Regression

```
cross_val_score(xg,scaled_x_best,y,cv=5).mean()
```

```
0.8317857020177609
```

SV Regression

```
cross_val_score(SV,scaled_x_best,y,cv=5).mean()
```

```
-0.06178097265098006
```

Decision Tree Regression

```
cross_val_score(dt,scaled_x_best,y,cv=5).mean()
```

```
0.7408266038992578
```

Interpretation of the Results

Based on comparing Accuracy Score results with Cross Validation results, it is determined that Random Forest Regressor is the best model. It also has the lowest Root Mean Squared Error score.

Hyper Parameter Tuning

GridSearchCV was used for Hyper Parameter Tuning of the Random Forest Regressor model.

Random Forest Regressor

```
parameter = {'n_estimators':[30,60,80],'max_depth': [10,20,40],'min_samples_leaf':[1,2,5,10,20,30],'min_samples_split':[5,10,20]
```

```
GridCV = GridSearchCV(RandomForestRegressor(),parameter,cv=5,n_jobs = -1,verbose = 1)
```

```
GridCV.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 972 candidates, totalling 4860 fits

GridSearchCV(cv=5, estimator=RandomForestRegressor(), n_jobs=-1,
             param_grid={'criterion': ['mse', 'mae'], 'max_depth': [10, 20, 40],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_leaf': [1, 2, 5, 10, 20, 30],
                         'min_samples_split': [5, 10, 20],
                         'n_estimators': [30, 60, 80]},
             verbose=1)
```

```
GridCV.best_params_
```

```
{'criterion': 'mae',
 'max_depth': 40,
 'max_features': 'sqrt',
 'min_samples_leaf': 1,
 'min_samples_split': 5,
 'n_estimators': 80}
```

```
Best_mod = RandomForestRegressor(n_estimators = 80,criterion = 'mae', max_depth= 40, max_features = 'sqrt',min_samples_leaf = 1,
```

```
Best_mod.fit(x_train,y_train)
```

```
RandomForestRegressor(criterion='mae', max_depth=40, max_features='sqrt',
                      min_samples_split=5, n_estimators=80)
```

```
rfpred = Best_mod.predict(x_test)
acc = r2_score(y_test,rfpred)
print(acc*100)
```

```
91.11965879968058
```

Based on the input parameter values and after fitting the train datasets

The Random Forest Regressor model was further tuned based on the parameter values yielded from GridsearchCV.

The Random Forest Regressor model displayed an accuracy of 91.11%

This model was then tested using a scaled Test Dataset comprising of 292 entries for 80 features. The model performed with good amount of accuracy.

```
Prediction_accuracy = pd.DataFrame({'Predictions': mod.predict(scaled_xtest_best), 'Actual Values': y[0:292]})
Prediction_accuracy
```

	Predictions	Actual Values
0	330040.16250	128000
1	202195.61250	268000
2	266406.98125	269790
3	162911.83750	190000
4	240113.93750	215000
5	95645.92500	219210
6	146961.07500	121500
7	338230.60000	166000
8	242841.46250	140000
9	169936.18750	118500
10	88305.87500	119500
11	146135.44375	237000

In summary, Based on the visualizations of the feature-column relationships, it is determined that, Features like OverallQual,GrLiveArea,MiscVal,ExterQual,KitchenQual,GaragеCars, GarageArea,TotalBsmtSF,1stFlrSF,FullBath,TotRmsAbvGrd,MasVnrA rea,FirePlaces have the strongest positive correlation with SalePrice and are some of the most important features to predict the label values.

Random Forest Regressor Performed the best out of all the models that were tested. It also worked well with the outlier handling.

CONCLUSION

Key Findings and Conclusions of the Study and Learning Outcomes with respect to Data Science

Based on the in-depth analysis of the Housing Project, The Exploratory analysis of the datasets, and the analysis of the Outputs of the models the following observations are made:

- Structural attributes of the house Structural attributes of the house like lot size, lot shape, quality and condition of the house, garage capacity, rooms, Lot frontage, number of bedrooms, bathrooms, overall finishing of the house etc play a big role in influencing the house price.
- Neighbourhood qualities can be included in deciding house price.
- Various plots like Barplots,Countplots and Lineplots helped in visualising the Feature-label relationships which corroborated the importance of structural and locational attributes for estimating Sale Prices.
- Due to the Training dataset being very small, the outliers had to be retained for proper training of the models.
- Therefore, Random Forest Regressor, being robust to outliers and being indifferent to non linear features, performed well despite having to work on small dataset.

Limitations of this work and Scope for Future Work

While features that focus on structural and locational attributes of housing properties are crucial for estimating the Sale Price of Housing properties, they aren't the only factors that influence the value in the housing market. Data on Demographics(Age,Income,Regional preferences of buyers, purpose of buying a property) is very important for understanding the Housing market. Interest Rates too impact the price and demand of houses. Economic cycles also influence Real Estate prices.
Government Policies, Regulations, Legalizations are also important

factors that may influence the sales of houses. The availability of data on above features would help build a predictive model that would more accurately understand the relationship between the features and target variable and yield more accurate predictions.