FLIP ROBO

Housing price prediction

Submitted by:

ROSHAN KUMAR

ACKNOWLEDGMENT

I would like to express my deep and sincere gratitude towards Fliprobo technologies for providing me the internship opportunity and a great chance for learning and professional development.

I am also greatful to my mentor mr. Shankar .

I am also greatful to my SME mrs. **Gulshana Chaudhary** for continuous support.

There dynamism, vision, sincerity and motivation have deeply inspired me. They taught me the methodology to carry out the task and to present the project works as clearly as possible. It was a great privilege and honour to work and study under there guidance. I am extremely grateful for what has been offered to me.

They provide valuable time for this work. No words are adequate toexpress my gratitude towards him. I would also like to thank them for his friendship and empathy.

INTRODUCTION

• Business Problem Framing

Real estate is the least transparent industry in our ecosystem. Housing prices keep changing day in and day out and sometimes are hyped rather than being based on valuation. Predicting housing prices with real factors is the main objective of our project. Here we aim to make our evaluations based on every basic parameter that is considered while determining the price

With a large amount of unstructured resources and documents, the Real estate industry has become a highly competitive business. The data science process in such an industry provides an advantage to the developers by processing those data, forecasting future trends and thus assisting them to make favourable knowledge-driven decisions.

House prices increase every year, so there is a need for a system to predict house prices in the future. House price prediction can help the developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house.

Prediction house prices are expected to help people who plan to buy a house so they can know the price range in the future, and then they can plan their finance well. In addition, house price predictions are also beneficial for property investors to know the trend of housing prices in a certain location.

There are three factors that influence the price of a house which include physical conditions, concept and location. There are several approaches that can be used to determine the price of the house, one of them is the prediction analysis. . We use various regression techniques in this pathway

• Conceptual Background of the Domain Problem

The real estate market is one of the most crucial components of any national economy. Hence, observations of the real estate market and accurate predictions of real estate prices are helpful for real estate buyers and sellers as well as economic specialists. However, real estate forecasting is a complicated and difficult task owing to many direct and indirect factors that inevitably influence the accuracy of predictions.

In general, factors influencing real estate prices could be quantitative or qualitative. The quantitative factors possibly include macroeconomic factors, business cycles, and real estate attributes. The macroeconomic factors contain unemployment rates, share index, current account of a country, industrial production, and gross domestic product.

Attributes of real estate, for example, includes past sale prices, land area, years of constructions, floor space, surface area, number of floors and building conditions, etc. The qualitative factors refer to subject preferences of decision makers, such as views, building styles, and living environment.

The real estate sector is a sector whose reach is vast; it is therefore affected by many social, political and economic factors which mean that there is a huge amount of complex data available. It is through analysing and understanding this data that models can be created which aim to replicate the changes in the sector and evolve in order to anticipate what we might see in the future.

• Review of Literature

Real estate price prediction is crucial for the establishment of real estate policies and can help real estate owners and agents make informative decisions. The aim of this study is to employ actual transaction data and machine learning models to predict prices of houses. The actual transaction data contain attributes and transaction prices of real estate that respectively serve as independent variables and dependent variables for machine learning models.

Real estate is one of the most fast-paced and emerging industries today. Nowadays everyone wants to be the owner of their house rather than live on rent. Therefore, people are very cautious in searching for the most suitable house. Different people have different budgets and so are their desires. This project draws attention to the house rate predictions based on different objectives like financial status and expectations of non-house holders.

Housing is the utmost need of any individual; it can be either bought or rented. .With time more and more people are drawn towards buying their own houses. Each has a different set of budgets and priorities for their house. People also surf the internet to search the house of their choice. For fulfilling their needs, machine learning engineers from across the world with data scientists are working to predict precise results to the shareholders and customers.

The models use variables regarding house facilities like interior square feet of the property, number of bedrooms, number of bathrooms, total number of rooms, the quality score assigned for rooms based on buyer reviews, the quality score assigned for bathroom based on buyer reviews, the quality score assigned for bedroom based on buyer reviews, the overall quality score assigned for the property, the sale condition of the house and the type of building.

• Motivation for the Problem Undertaken

House Price prediction is important to drive Real Estate efficiency. As earlier, House prices were determined by calculating the acquiring and selling price in a locality.
The House Price prediction model is very essential in filling the information gap and improves Real Estate efficiency. The goal of the paper is to predict the efficient house pricing for real estate customers with respect to their budgets and priorities. By analysing previous market trends and price ranges, and also upcoming developments future prices will be predicted. This model will help customers to invest in an estate without approaching an agent. It also decreases the risk involved in the transaction. Nowadays, e-education and e-learning is highly influenced. Everything is shifting from manual to automated systems. The objective of this project is to predict the house prices so as to minimize the problems faced by the customer. The present method is that the customer approaches a real estate agent to manage his/her investments and suggest suitable estates for his investments. But this method is risky as the agent might predict wrong estates and thus leading to loss of the customer's investments.
In this project, the main focus is on developing a model which not only predicts the sale price of properties for a customer according to his\her interests, but also recognizes the most preferred location of real estate and other utilities and features in any given area.
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 are one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.
A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file below. The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:

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

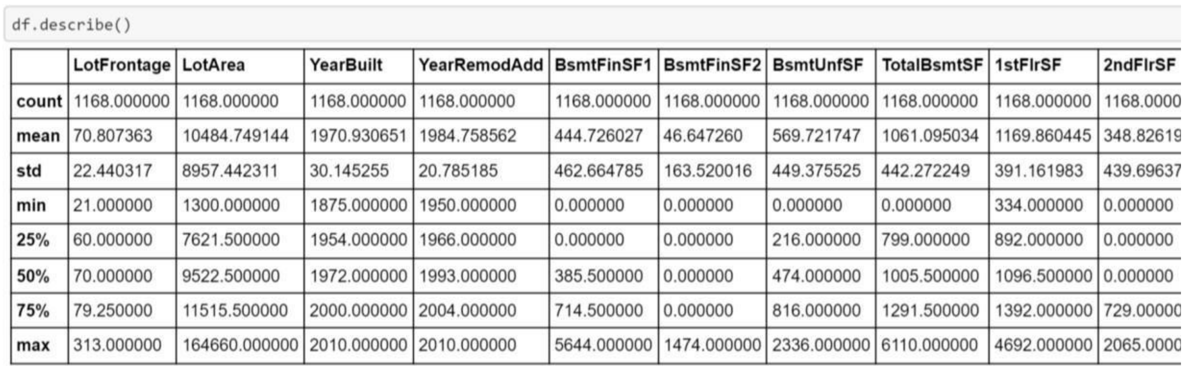
We are required to model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices

vary with the variables. Real estate developers can then accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.
Thus, the machine learning-based model is a substantial and feasible way to forecast real estate prices, and can provide relatively competitive and satisfactory results.

Analytical Problem Framing

- ### Mathematical/ Analytical Modeling of the Problem

The statistical analysis are carried out on each numeric feature using the dataframe.describe () method which gives all the statistical information such as mean of the feature, median, standard deviation, maximum and minimum count.

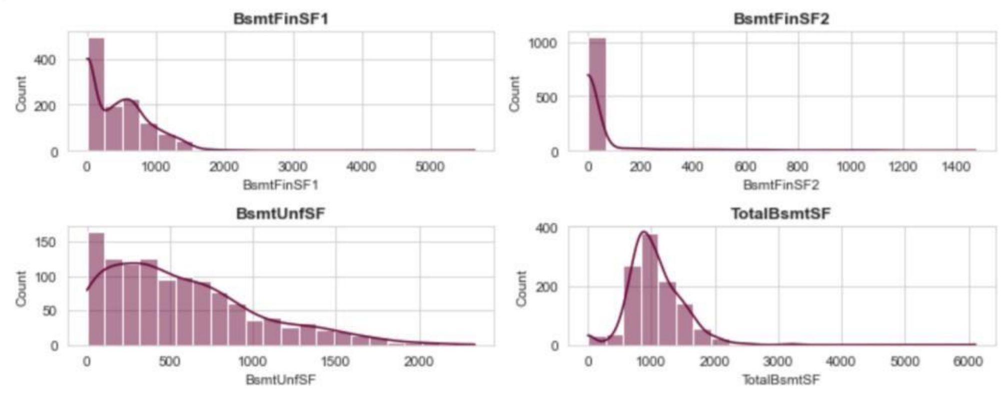


```
df.describe()
```

	LotFrontage	LotArea	YearBuilt	YearRemodAdd	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF
count	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.0000
mean	70.807363	10484.749144	1970.930651	1984.758562	444.726027	46.647260	569.721747	1061.095034	1169.860445	348.82619
std	22.440317	8957.442311	30.145255	20.785185	462.664785	163.520016	449.375525	442.272249	391.161983	439.69637
min	21.000000	1300.000000	1875.000000	1950.000000	0.000000	0.000000	0.000000	0.000000	334.000000	0.000000
25%	60.000000	7621.500000	1954.000000	1966.000000	0.000000	0.000000	216.000000	799.000000	892.000000	0.000000
50%	70.000000	9522.500000	1972.000000	1993.000000	385.500000	0.000000	474.000000	1005.500000	1096.500000	0.000000
75%	79.250000	11515.500000	2000.000000	2004.000000	714.500000	0.000000	816.000000	1291.500000	1392.000000	729.00000
max	313.000000	164660.000000	2010.000000	2010.000000	5644.000000	1474.000000	2336.000000	6110.000000	4692.000000	2065.0000

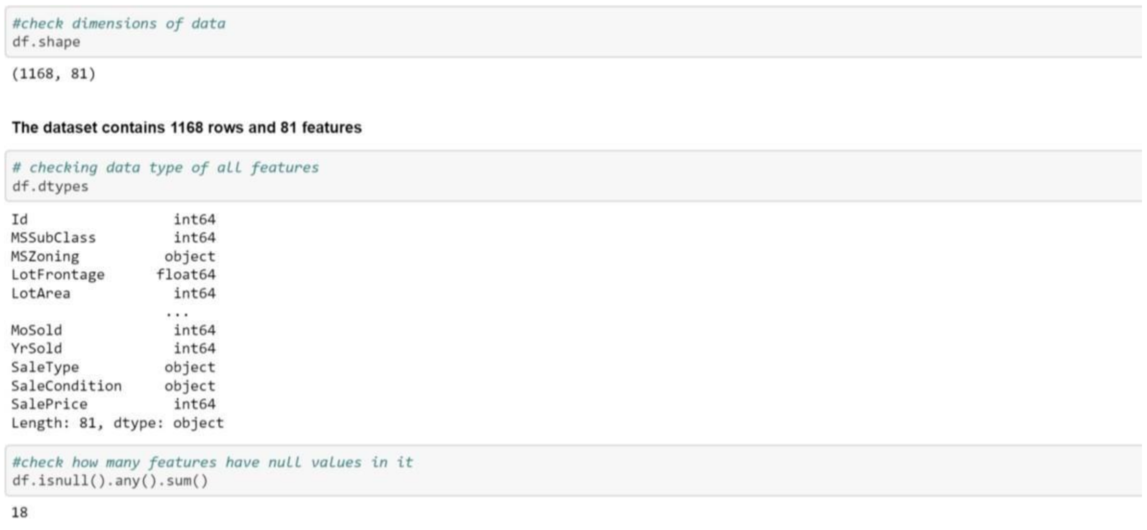
Z-score is one such method used to make analyses of the outliers present.

The skewness and distribution of various data is also studied using visualizations



.

Another EDA included checking for dimensionality of data and its characteristics. Which are shown below:



```
#check dimensions of data
df.shape
```

```
(1168, 81)
```

The dataset contains 1168 rows and 81 features

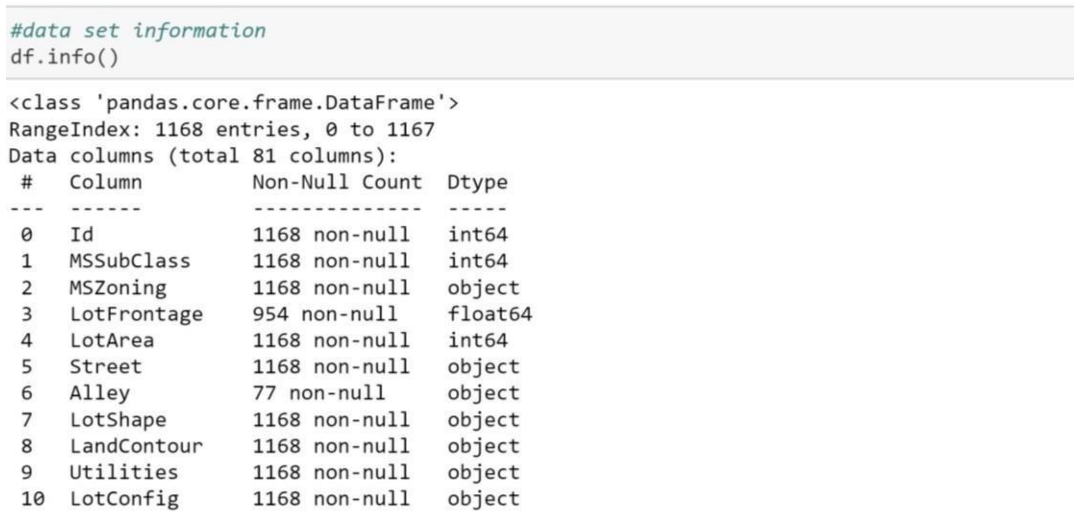
```
# checking data type of all features
df.dtypes
```

```
Id                 int64
MSSubClass         int64
MSZoning          object
LotFrontage      float64
LotArea            int64
                   ...
MoSold             int64
YrSold             int64
SaleType          object
SaleCondition     object
SalePrice          int64
Length: 81, dtype: object
```

```
#check how many features have null values in it
df.isnull().any().sum()
```

```
18
```

The entire data set features could be analysed using dataframe.info()



```
#data set information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1168 entries, 0 to 1167
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1168 non-null   int64
 1   MSSubClass     1168 non-null   int64
 2   MSZoning       1168 non-null   object
 3   LotFrontage    954 non-null    float64
 4   LotArea        1168 non-null   int64
 5   Street         1168 non-null   object
 6   Alley          77 non-null     object
 7   LotShape       1168 non-null   object
 8   LandContour    1168 non-null   object
 9   Utilities      1168 non-null   object
 10  LotConfig      1168 non-null   object
```

• Data Sources and their formats

1. A US-based housing company named Surprise Housing has decided to enter the Australian market and has collected a data set from the sale of houses in Australia.
2. The data is provided in the CSV file .
3. The entire dataset is divided into train and test datasets.
4. The train dataset has 81 columns and a total of 1168 records (rows).Where there are 80 inputs/features and 1 target column(label) .Sale price is our target variable.
5. The test dataset has 80 columns and a total of 292 records (rows) for which we need to make the predictions.
6. Data contains Null values. Which needs to be treated using the domain knowledge and own understanding.

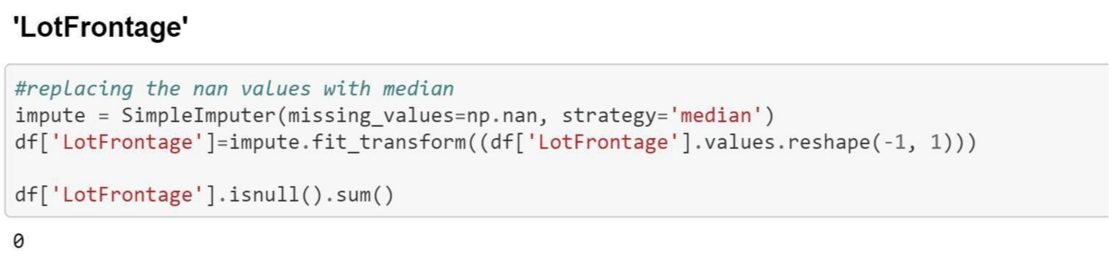
7. The train data had 3 float64 features,35 columns with int64 and 43 features with obje ct data type
8. Almost 18 features have Nan values
9. Data contains numerical as well as categorical variable. We need to handle them accordingly.
10. We will have to build Machine Learning models, apply regularization and determine the optimal values of Hyper Parameters, find important features which affect the price positively or negatively
11. We will train on train.csv dataset and predict on test.csv file.
12. There are few outliers (about 9%) in the data that needs to be handled.

• Data Pre-processing Done

Data is everywhere. The real trick lies in how can we parse it, make sense of it and extract insights. Then, the next step is usually what insights should be prioritized. How to create as many variables and granularity in these variables should be studied and evaluated. Before we feed the data to the model we should make sure our data is clean and well structured.

Few pre-processing steps carried out on the dataset are:

1. **Treating Nan values:** There were 18 features with Nan values out of which we have dropped features which had more than 90% Nan values.
For rest of the features in case of integer or float a simple imputer is used to fill the Nan values with different strategies



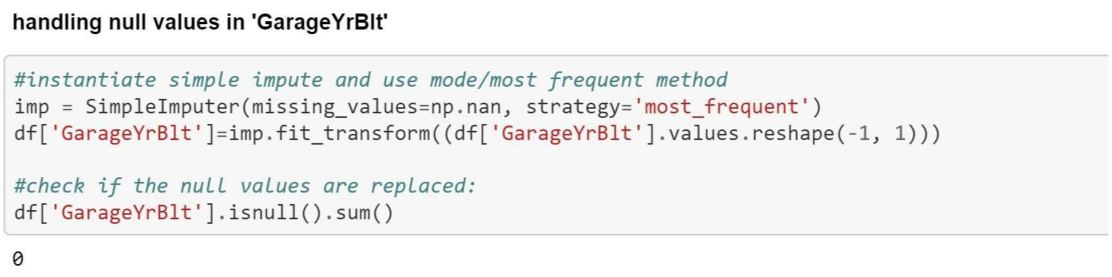
'LotFrontage'

```
#replacing the nan values with median
impute = SimpleImputer(missing_values=np.nan, strategy='median')
df['LotFrontage']=impute.fit_transform((df['LotFrontage'].values.reshape(-1, 1)))

df['LotFrontage'].isnull().sum()
```

```
0
```

.



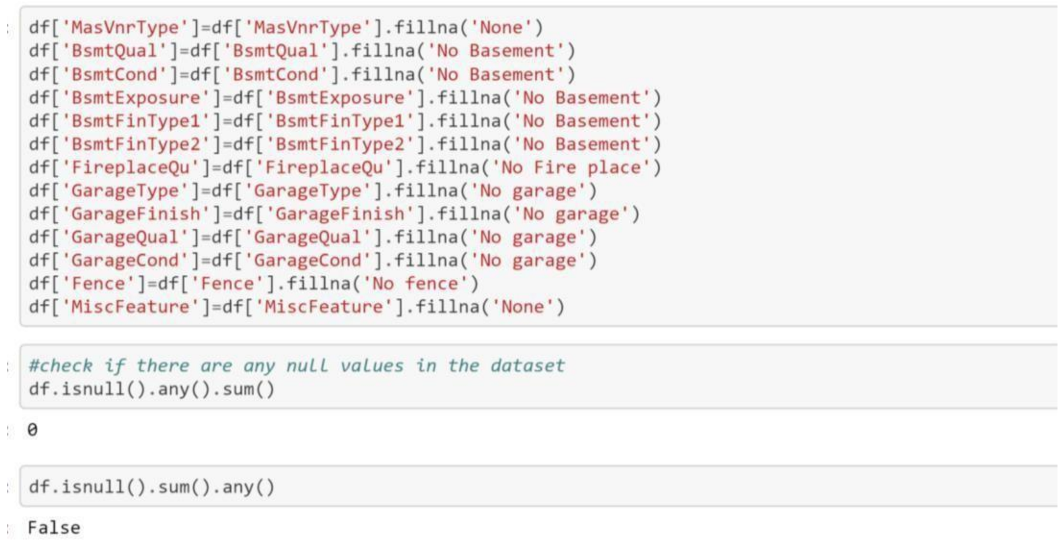
handling null values in 'GarageYrBlt'

```
#instantiate simple impute and use mode/most frequent method
imp = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
df['GarageYrBlt']=imp.fit_transform((df['GarageYrBlt'].values.reshape(-1, 1)))

#check if the null values are replaced:
df['GarageYrBlt'].isnull().sum()
```

```
0
```

For categorical columns a separate class is assigned in place of Nan values referring to the data description provided by the firm.



```
df['MasVnrType']=df['MasVnrType'].fillna('None')
df['BsmtQual']=df['BsmtQual'].fillna('No Basement')
df['BsmtCond']=df['BsmtCond'].fillna('No Basement')
df['BsmtExposure']=df['BsmtExposure'].fillna('No Basement')
df['BsmtFinType1']=df['BsmtFinType1'].fillna('No Basement')
df['BsmtFinType2']=df['BsmtFinType2'].fillna('No Basement')
df['FireplaceQu']=df['FireplaceQu'].fillna('No Fire place')
df['GarageType']=df['GarageType'].fillna('No garage')
df['GarageFinish']=df['GarageFinish'].fillna('No garage')
df['GarageQual']=df['GarageQual'].fillna('No garage')
df['GarageCond']=df['GarageCond'].fillna('No garage')
df['Fence']=df['Fence'].fillna('No fence')
df['MiscFeature']=df['MiscFeature'].fillna('None')
```

```
#check if there are any null values in the dataset
df.isnull().any().sum()
```

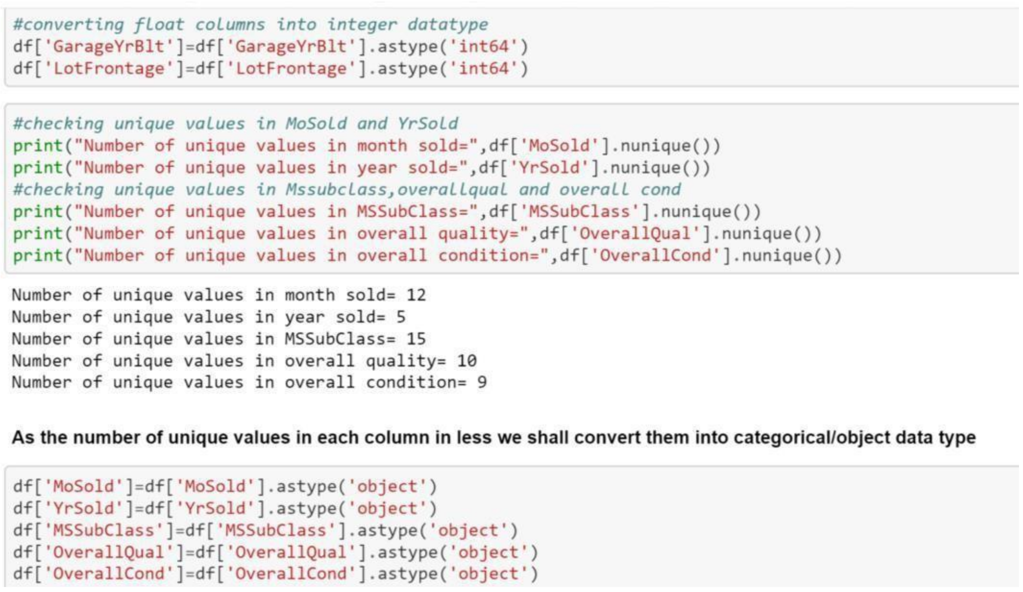
```
0
```

```
df.isnull().sum().any()
```

```
False
```

2. **Data-Type Conversions**

The float columns were converted into integer and a few columns which were numeric but had very less unique values were converted to categorical data type in order to treat them wisely.



```
#converting float columns into integer datatype
df['GarageYrBlt']=df['GarageYrBlt'].astype('int64')
df['LotFrontage']=df['LotFrontage'].astype('int64')
```

```
#checking unique values in MoSold and YrSold
print("Number of unique values in month sold=",df['MoSold'].nunique())
print("Number of unique values in year sold=",df['YrSold'].nunique())
#checking unique values in Mssubclass,overallqual and overall cond
print("Number of unique values in MSSubClass=",df['MSSubClass'].nunique())
print("Number of unique values in overall quality=",df['OverallQual'].nunique())
print("Number of unique values in overall condition=",df['OverallCond'].nunique())
```

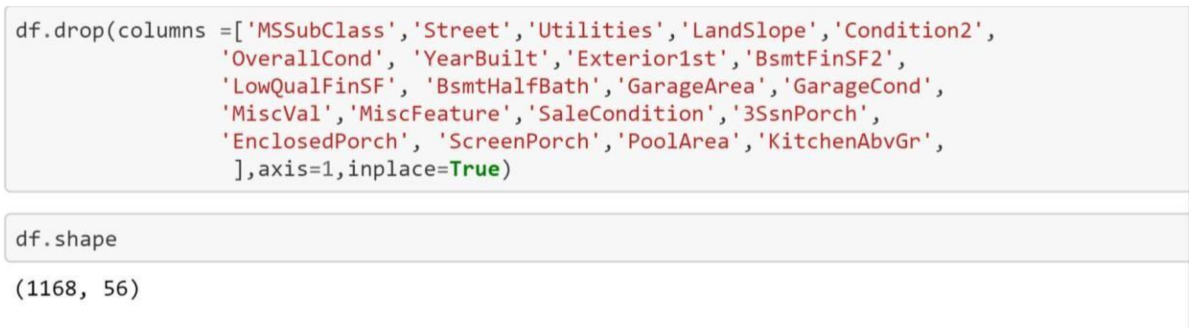
```
Number of unique values in month sold= 12
Number of unique values in year sold= 5
Number of unique values in MSSubClass= 15
Number of unique values in overall quality= 10
Number of unique values in overall condition= 9
```

As the number of unique values in each column in less we shall convert them into categorical/object data type

```
df['MoSold']=df['MoSold'].astype('object')
df['YrSold']=df['YrSold'].astype('object')
df['MSSubClass']=df['MSSubClass'].astype('object')
df['OverallQual']=df['OverallQual'].astype('object')
df['OverallCond']=df['OverallCond'].astype('object')
```

3. **Dropping unnecessary features**: features with 0 correlations with the target variable are dropped some redundant features are also dropped in order to increase model efficiency.
Features that were dropped due huge number of Nan values are : "Alley", " PoolQc" , "MasvnrArea" .All these feature contained more than 70 % Nan values and hence did not provide any significant information so it is relevant to drop them.

Few other features were dropped depending upon its correlation with the target.The features were as follows:



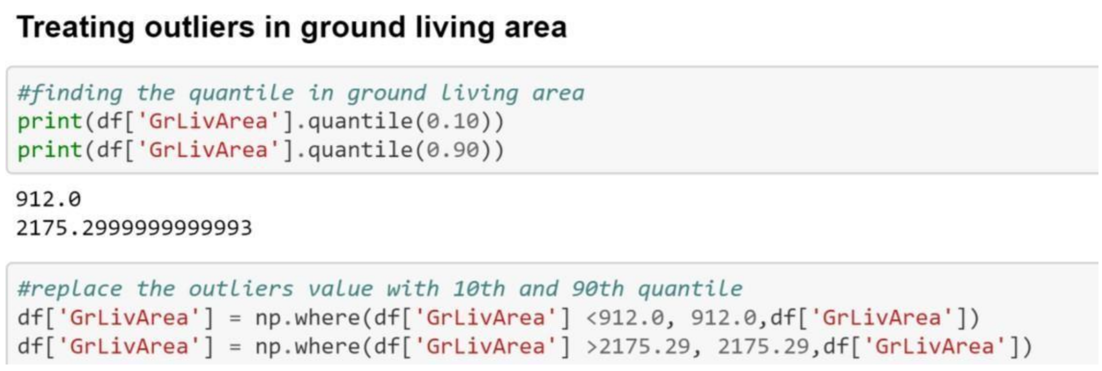
```
df.drop(columns =['MSSubClass','Street','Utilities','LandSlope','Condition2',
                  'OverallCond', 'YearBuilt','Exterior1st','BsmtFinSF2',
                  'LowQualFinSF', 'BsmtHalfBath','GarageArea','GarageCond',
                  'MiscVal','MiscFeature','SaleCondition','3SsnPorch',
                  'EnclosedPorch', 'ScreenPorch','PoolArea','KitchenAbvGr',
                   ],axis=1,inplace=True)
```

```
df.shape
```

```
(1168, 56)
```

- 'MSSubClass':Has a very small correlation with target (-0.06)
- 'Street': Has a very small correlation with target (0.04)
- 'Uitlities' has only one unique value for all rows (All public Utilities (E,G,W,& S))
- 'LandSlope':low correlation (0.02)
- 'Condition2':Condition1 and condition2 are almost same as condition1 as high correlation than condition2 drop condition2.
- 'OverallCond': has low negative correlation -0.07
- YearBuilt': Original construction date and Remodel date (same as construction date if no remodelling or additions) has same correlation (0.51)so to keep one any of them let us keep renewed date and drop year built.
- 'Exterior1st': exterior2 and exterior1st are almost same take one with high correlation i.e. exterior2, drop exterior1.
- 'BsmtFinSF2':low correlation (-0.01)
- 'LowQualFinSF':has low negative correlation of -0.03
- 'BsmtHalfBath':has low negative correlation of -0.01
- 'GarageArea': area according to sq. and garage cars is giving same info let's keep cars as there are few outliers in the same and has high correlation.
- 'GarageCond': garage condition and 'GarageQual' both are giving same info (0.29) let's keep garage quality and drop 'GarageCond'.
- 'MiscVal': has very low co-relation with the target (-0.01)
- 'MiscFeature': has low correlation (0.08) and 1124 values are None which provide no significant information. *'SaleType', 'SaleCondition', both have correlation of (0.37) sale condition is giving no great info let's keep sale type
- '3SsnPorch':has a low correlation co-efficient of 0.06
- 'PoolArea' have 1161 values as 0 and corr of (0.10)
- 'EnclosedPorch', has low correlation of -0.12
- 'ScreenPorch' has low correlation of 0.10 and too many outliers
- 'KitchenAbvGr' : correlation -0.13 and most of the values are 1

4. **Treatment of outliers**: flooring and capping method is used to reduce the outliers in few variables. There were total 9% outliers in the dataset before treating them.

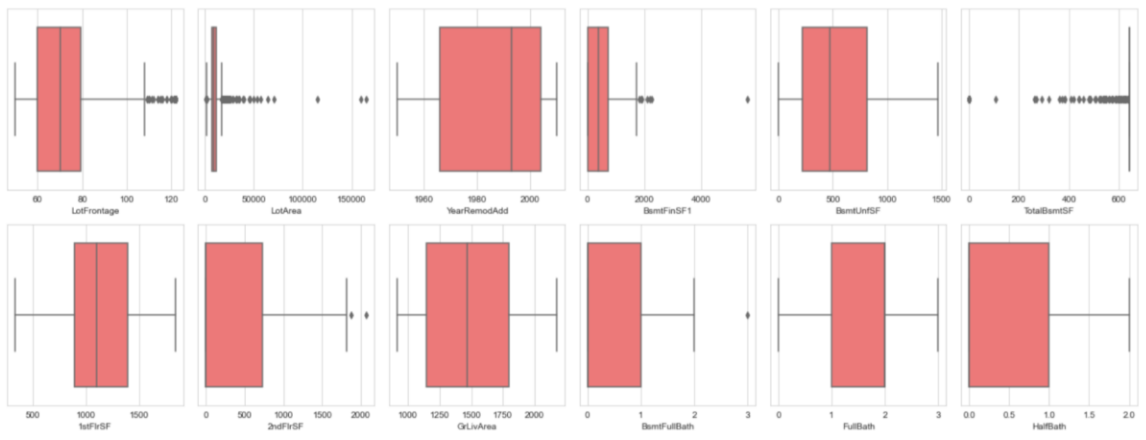

Treating outliers in ground living area

```
#finding the quantile in ground living area
print(df['GrLivArea'].quantile(0.10))
print(df['GrLivArea'].quantile(0.90))
```

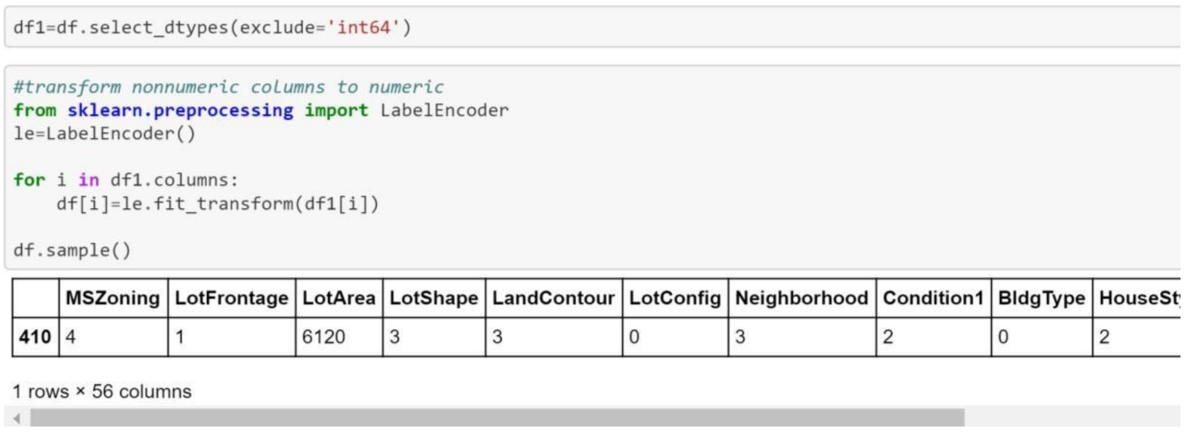
```
912.0
2175.2999999999993
```

```
#replace the outliers value with 10th and 90th quantile
df['GrLivArea'] = np.where(df['GrLivArea'] <912.0, 912.0,df['GrLivArea'])
df['GrLivArea'] = np.where(df['GrLivArea'] >2175.29, 2175.29,df['GrLivArea'])
```

The outliers can be visualized using box-plot as below



5. **Label encoding:** The machine learning model can only understand numbers hence we need to convert/encode all the categorical columns into numbers. Label encoder has been used for the same.



```
df1=df.select_dtypes(exclude='int64')
```

```
#transform nonnumeric columns to numeric
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

for i in df1.columns:
    df[i]=le.fit_transform(df1[i])

df.sample()
```

	MSZoning	LotFrontage	LotArea	LotShape	LandContour	LotConfig	Neighborhood	Condition1	BldgType	HouseSt
410	4	1	6120	3	3	0	3	2	0	2

1 rows × 56 columns

6. **Skewness reduction**: as the features had uneven distribution power transformation is used to reduce the skewness. The method used is yeo-Johnson.

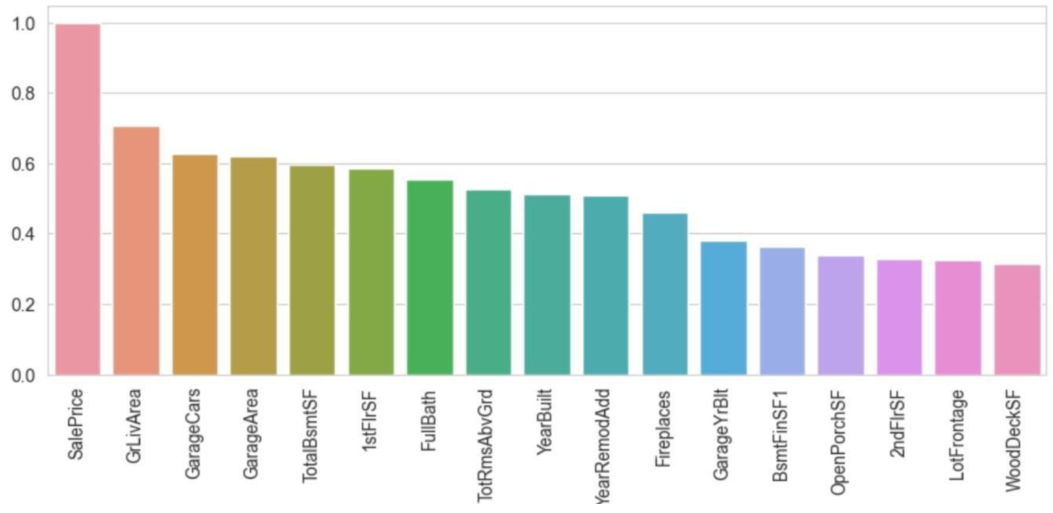
```python
# power transformation to reduce the skewness
from sklearn.preprocessing import power_transform
x=power_transform(x,method='yeo-johnson')
```

• Data Inputs- Logic- Output Relationships

To get an insight of the relationship between inputs and the target variable(output) we have studied the correlation using pandas corr() function.
The features with correlation of more than 0.3 are shown below.

```python
corr = abs(df.corr()["SalePrice"]).sort_values(ascending = False)
large_corr = corr[corr > 0.3]

plt.figure(figsize = (10, 4), dpi = 100)
sns.barplot(x = large_corr.index, y = large_corr.values)
plt.xticks(rotation = 90)
plt.show()
```



The correlation of numeric inputs with respect to output is as follows:

• SalePrice	1.000000
• GrLivArea	0.707300
• GarageCars	0.628329
• GarageArea	0.619000
• TotalBsmtSF	0.595042
• 1stFlrSF	0.587642
• FullBath	0.554988
• TotRmsAbvGrd	0.528363
• YearBuilt	0.514408
• YearRemodAdd	0.507831
• Fireplaces	0.459611
• GarageYrBlt	0.381997
• BsmtFinSF1	0.362874
• OpenPorchSF	0.339500

2ndFlrSF	0.330386
LotFrontage	0.323851
WoodDeckSF	0.315444
HalfBath	0.295592
LotArea	0.249499
BsmtUnfSF	0.215724
BsmtFullBath	0.212924
BedroomAbvGr	0.158281
PoolArea	0.103280
ScreenPorch	0.100284
3SsnPorch	0.060119
BsmtFinSF2	-0.010151
BsmtHalfBath	-0.011109
MiscVal	-0.013071
LowQualFinSF	-0.032381
EnclosedPorch	-0.115004
KitchenAbvGr	-0.132108

For the relationship between categorical inputs and numeric target we have used the dython tool. Which yield the following correlation co-efficient:

Columns with high positive correlation with target

- OverallQual(0.79)
- Neighborhood(0.73)
- BsmtQual(0.69)
- KitchenQual(0.68)
- ExterQual(0.68)
- GarageFinish(0.55)
- FireplaceQu(0.54)
- GarageType(0.50)
- Foundation(0.50)

- Columns with least correlation:
  - MSSubClass(-0.06)
  - Street (0.04)
  - Landslope(0.02)
  - MiscFeature(0.08)
  - MoSold(0.07)
  - YrSOld(-0.05)
  - OverallCond(-0.07)

- Columns with same correlation:
  - GarageQual and GarageCond (0.29)
  - SaleType and SaleCondition (0.37)
  - Exterior2nd and MsVnrType (0.41) Exterior1 (0.40)
  - Condition1 and Condition2 (0.19 and 0.10)

• Hardware and Software Requirements and Tools Used

Open source web-application used for programming:

1. **Jupyter Notebook**

Python Libraries / Packages used were:

1. **Pandas**: pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

   We have used pandas to import the csv file using pd.read_csv all data analysis have been done using the pandas and numpy libraries. The data characteristics have been studied using pandas functions like df.shape(), df .dtypes, df.columns etc.

2. **NumPy:** NumPy is an open-source numerical Python library. NumPy contains amulti-dimensional array and matrix data structures. It can be utilised to perform anumber of mathematical operations on arrays such as trigonometric, statistical, and algebraic.

   We have used the np.where function many times while dealing with the z-scores.np.abs () function has also been used to find the zscore and some mathematical operations like square root

3. **Matplotlib**: library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

   The Matplotlib libraries pyplot function is used for making plots ,plt.show() ,plt.figure(figsize) that has been used is a part of matplotlib library.

4. **Seaborn**: Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

   All the visualizations made are built using the seaborn library. Alias used for seaborn is sns.

   Sns.boxplot(), sns.heatmap(), sns.distplot(), sns.scatterplot() ,sns.stripplot,sns.swarmplot ,heatmap are few of the libraries used

5. **SciPy**: SciPy, a scientific library for Python is an open source library for mathematics, science and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The main reason for building the SciPy library is that, it should work with NumPy arrays.

In this particular project scipy functions such as scipy.stats is used .Zscores are also obtained via scipy.stats library

6. **Sklearn:** Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, and clustering and dimensionality reduction via a consistence interface in Python.
   All the Machine Learning regression algorithms have been imported from the sklearn package. Simple imputer used is also a part of sklearn

   The evaluation metrics, RMSE,MSE,MAE functions are also imported from same.

7. **Dython:** is a set of Data analysis tools in python 3.x. which gives us measures of association for categorical features, Plots features correlation and association for mixed data-sets (categorical and continuous features) in an easy and simple way.

Model/s Development and Evaluation

- ### Identification of possible problem-solving approaches (methods)

The goal of this statistical analysis is to help us understand the relationship between house features and how these variables are used to predict house price.

One of the most crucial steps of building a machine learning model is data exploration. It is done to gain various insights of our dataset and figure out the errors. Some of the common errors in datasets are missing values, outliers and skewness Dataset is first cleaned, before predicting the accuracy.

We use the Train set to make the algorithm learn the data's behaviour and shall make predictions for the test dataset.

The very first step includes identifying the type of problem i.e. whether it is a classification problem or a regression problem. As our target variable "Sale Price" is continuous in nature our problem should be a regression problem

The very first thought that shall strike us upon looking at the dataset is what features might affect our prediction either positively or negatively. Hence it becomes vital to observe correlation between all the input variables and the target.

The corr() function has helped us determine the features of utmost importance.

Also the dython tool has shown us the correlation between categorical features and the numeric target variable.

After selection of the most relatable features the very next aspect is to reduce the assymmetricity in the data using skewness reduction or outlier detection and treatment.

As most of the data was categorical in nature the amount of outliers was less (9 %) hence we had reduced it by flooring and capping method instead of dropping them entirely.

Not all the outliers are treated or dropped because our problem defines real estate characteristics which might contain these outliers; for an example a property with 4 BHK is always a possibility though not much likely hence we should not ignore the possibility.

In our project, 75% of the readings have been selected for the train set and the other 25% for the test set. The values for dependent variables are determined by applying certain algorithms on the training set. Then, a training set algorithm is put in the testing data for only independent variables. The obtained dependent variables and original values are compared and the one with the minimum error is selected.

The sale price depends on all the features combined rather than on just two features. This study will further show that the sale price of a house depends on various features combined and not just a few correlated features.

• Testing of Identified Approaches (Algorithms) ,Run and Evaluate selected models

In data science there is always a scope for improvement and we cannot say that a single model will always be a perfect fit for any problem statement. An algorithm might work best in some cases but the same algorithm won't perfectly yield the needed output upon some minor changes. That is why it always a good practice to try and build our final model only after trying multiple permutations and combinations on the data available, until and unless we are satisfied with the end results.

1. **Linear Regression**:

   Linear Regression is used between the target variable and other independent variables to find a linear relationship between them. It is a regression model, used for predicting continuous values. Given, a dataset $\{y \ , xi1, \dots \ , xip\} \ i=1 \qquad n$ of n observations presumes that p-vector of x regression and y, the dependent variable, has a linear relationship. There is a term for disturbance referred to as ε, the variable for error that adds unwanted noise in the relationship between the regressors and the dependent variables.

The equation formulates as.

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = x_i^T \beta + \varepsilon_i, \quad i = 1, \ldots, n,$$

Where T is the transpose of the matrix.

In general, *y* is written as, $y = X\beta + \varepsilon$

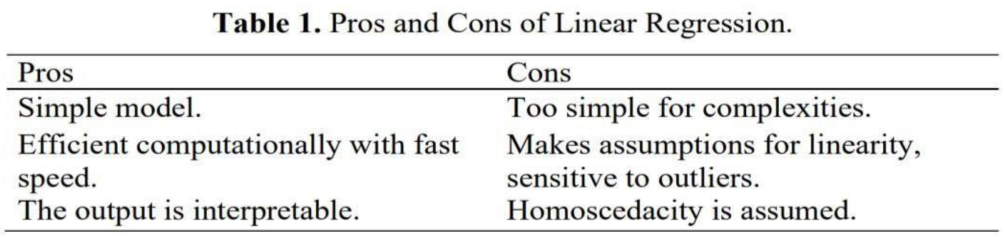
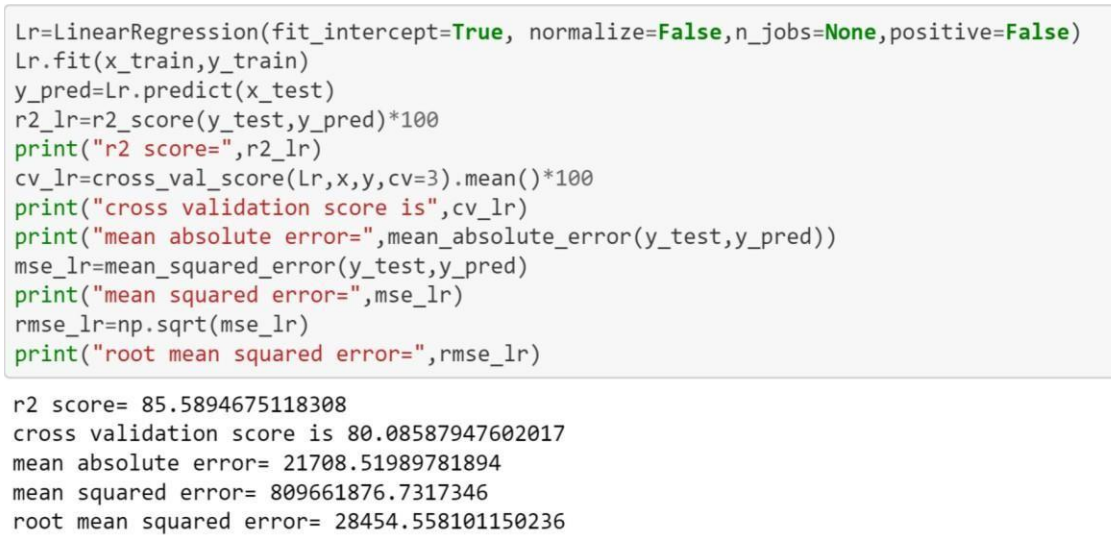


Table 1. Pros and Cons of Linear Regression.

Pros	Cons
Simple model.	Too simple for complexities.
Efficient computationally with fast speed.	Makes assumptions for linearity, sensitive to outliers.
The output is interpretable.	Homoscedacity is assumed.

The Linear regression has been implemented and the results are as follows:



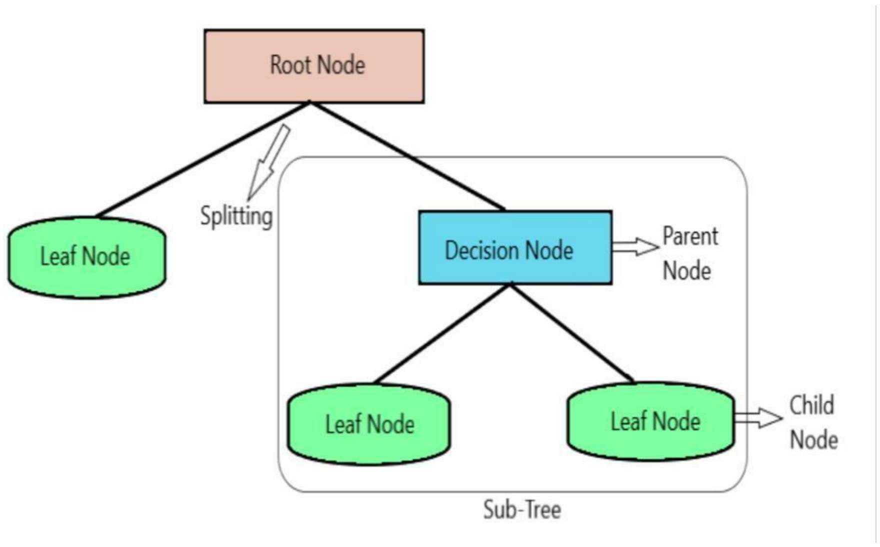
```
Lr=LinearRegression(fit_intercept=True, normalize=False,n_jobs=None,positive=False)
Lr.fit(x_train,y_train)
y_pred=Lr.predict(x_test)
r2_lr=r2_score(y_test,y_pred)*100
print("r2 score=",r2_lr)
cv_lr=cross_val_score(Lr,x,y,cv=3).mean()*100
print("cross validation score is",cv_lr)
print("mean absolute error=",mean_absolute_error(y_test,y_pred))
mse_lr=mean_squared_error(y_test,y_pred)
print("mean squared error=",mse_lr)
rmse_lr=np.sqrt(mse_lr)
print("root mean squared error=",rmse_lr)
```

```
r2 score= 85.5894675118308
cross validation score is 80.08587947602017
mean absolute error= 21708.51989781894
mean squared error= 809661876.7317346
root mean squared error= 28454.558101150236
```

2. Decision Trees Regression:

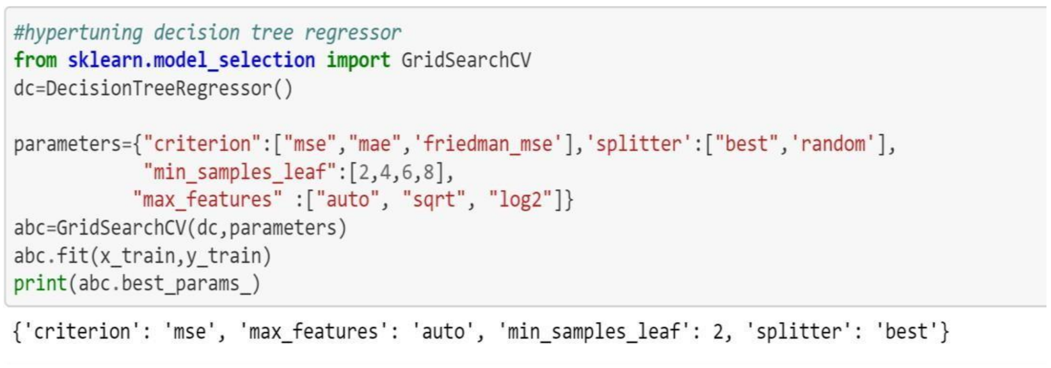
The decision tree is a sort of algorithm that requires a label for functioning, and hence they come under supervised learning. The main aim to use the Decision Tree algorithm involves creating a training model used for predicting the target variable's class or value by studying easy decision order.

Some of the terms used in decision trees are leaf/terminal node, which refers to the nodes which do not have any nodes branching out from them, decision node, refers to the nodes which have child nodes, the root node, refers to the highest node without any parent node, splitting, refers to the process of branching, sub-tree/branch, refers to a tree that is a child of another node, parent node, refers to a parent of a certain node and child node refers to the child of certain parent node.



The above figure shows an overview of these terms

Finding the best parameters for decision tree regression

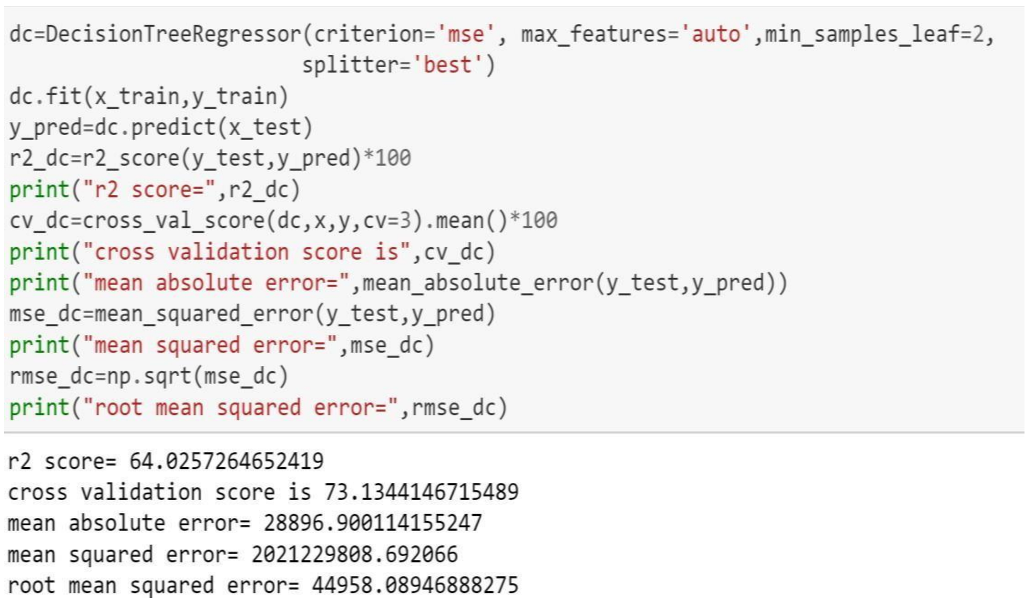


```
#hypertuning decision tree regressor
from sklearn.model_selection import GridSearchCV
dc=DecisionTreeRegressor()

parameters={"criterion":["mse","mae",'friedman_mse'],'splitter':["best",'random'],
            "min_samples_leaf":[2,4,6,8],
           "max_features" :["auto", "sqrt", "log2"]}
abc=GridSearchCV(dc,parameters)
abc.fit(x_train,y_train)
print(abc.best_params_)
```

```
{'criterion': 'mse', 'max_features': 'auto', 'min_samples_leaf': 2, 'splitter': 'best'}
```

Using the parameters obtained above:

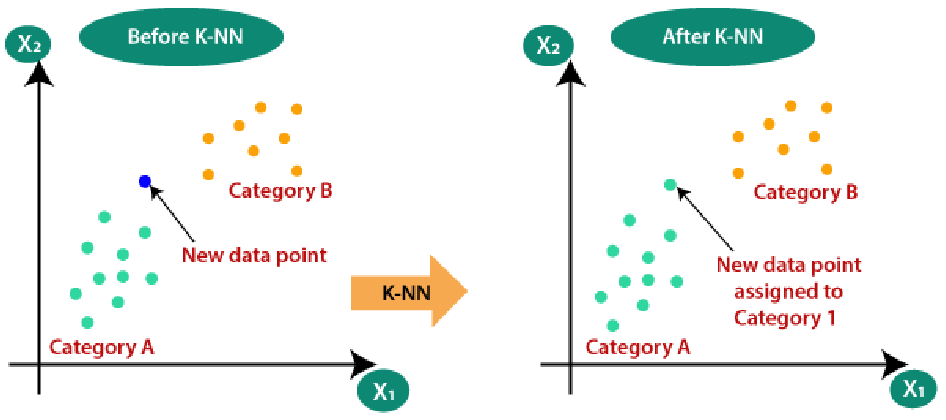


```
dc=DecisionTreeRegressor(criterion='mse', max_features='auto',min_samples_leaf=2,
                         splitter='best')
dc.fit(x_train,y_train)
y_pred=dc.predict(x_test)
r2_dc=r2_score(y_test,y_pred)*100
print("r2 score=",r2_dc)
cv_dc=cross_val_score(dc,x,y,cv=3).mean()*100
print("cross validation score is",cv_dc)
print("mean absolute error=",mean_absolute_error(y_test,y_pred))
mse_dc=mean_squared_error(y_test,y_pred)
print("mean squared error=",mse_dc)
rmse_dc=np.sqrt(mse_dc)
print("root mean squared error=",rmse_dc)
```

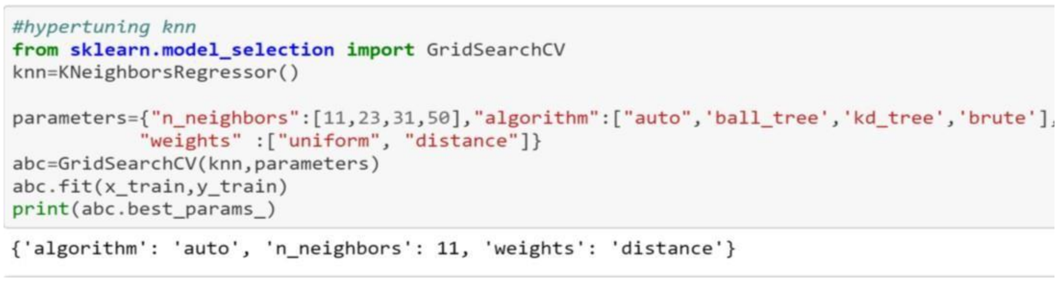
```
r2 score= 64.0257264652419
cross validation score is 73.1344146715489
mean absolute error= 28896.900114155247
mean squared error= 2021229808.692066
root mean squared error= 44958.08946888275
```

3. KNN Regression:

KNN algorithm can be used for both classification and regression problems. The KNN algorithm uses 'feature similarity' to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.



Hyper tuning KNN Regressor

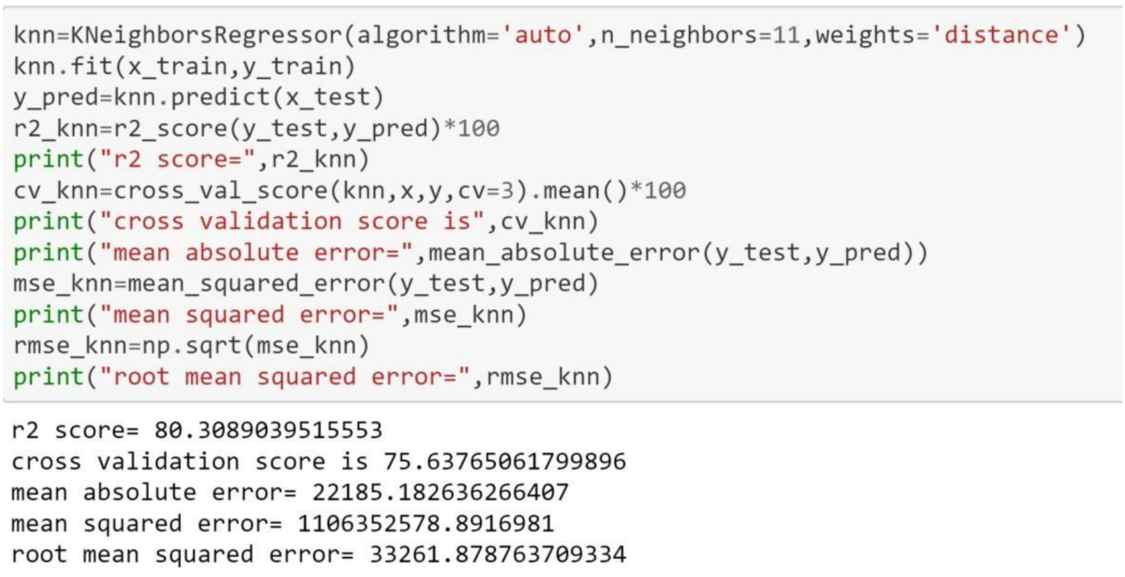


```
#hypertuning knn
from sklearn.model_selection import GridSearchCV
knn=KNeighborsRegressor()

parameters={"n_neighbors":[11,23,31,50],"algorithm":["auto",'ball_tree','kd_tree','brute'],
           "weights" :["uniform", "distance"]}
abc=GridSearchCV(knn,parameters)
abc.fit(x_train,y_train)
print(abc.best_params_)
```

```
{'algorithm': 'auto', 'n_neighbors': 11, 'weights': 'distance'}
```

Using the best parameters for KNN

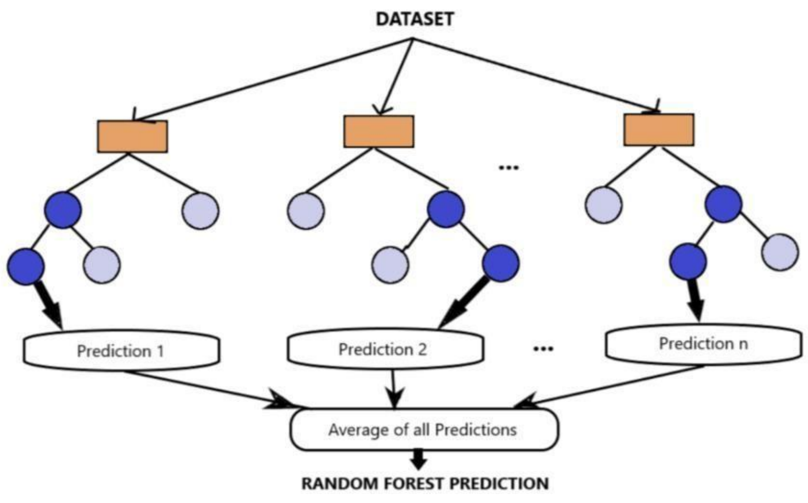


```
knn=KNeighborsRegressor(algorithm='auto',n_neighbors=11,weights='distance')
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
r2_knn=r2_score(y_test,y_pred)*100
print("r2 score=",r2_knn)
cv_knn=cross_val_score(knn,x,y,cv=3).mean()*100
print("cross validation score is",cv_knn)
print("mean absolute error=",mean_absolute_error(y_test,y_pred))
mse_knn=mean_squared_error(y_test,y_pred)
print("mean squared error=",mse_knn)
rmse_knn=np.sqrt(mse_knn)
print("root mean squared error=",rmse_knn)
```

```
r2 score= 80.3089039515553
cross validation score is 75.63765061799896
mean absolute error= 22185.182636266407
mean squared error= 1106352578.8916981
root mean squared error= 33261.878763709334
```

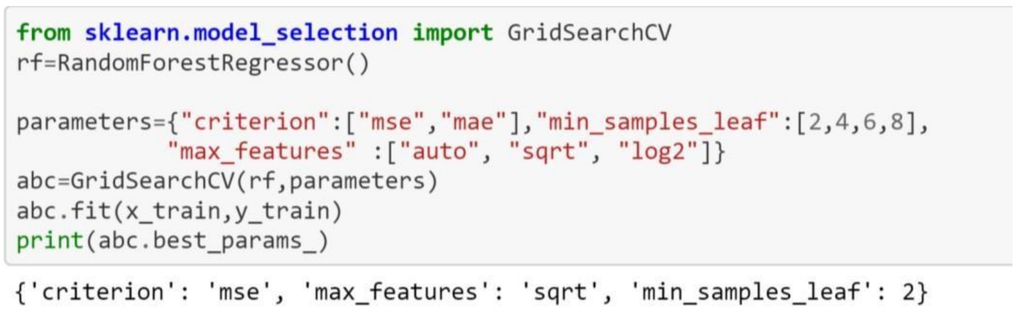
4. **Random Forest Regression** :

Random Forest is a supervised learning algorithm. The decision tree is contemplated as a base of the Random Forest algorithm .Random Forest can be used for regression as well as classification just like the Decision Tree algorithm .For the classification model it works with a categorical target, whereas in the regression model, it predicts the values of a continuous variable. Random Forest is a collection of several decision trees, just like a real forest that consists of trees. It is often also referred to an ensemble learning technique. For the processing, a set of samples of data are picked arbitrary which comprises different decision trees. Each tree gives the prediction, and the average of them is selected for the case in the regression model. In this paper, the regression algorithm of random forests is used. First of all, it divides the dataset into multiple sets, each set produces a decision tree based on different evaluation metrics used. Later the mean of all the predictions is considered as the final prediction.



Overview of Random Forest Regression

Hyper tuning Random forest regression to find optimal parameters



```
from sklearn.model_selection import GridSearchCV
rf=RandomForestRegressor()

parameters={"criterion":["mse","mae"],"min_samples_leaf":[2,4,6,8],
            "max_features" :["auto", "sqrt", "log2"]}
abc=GridSearchCV(rf,parameters)
abc.fit(x_train,y_train)
print(abc.best_params_)
```

```
{'criterion': 'mse', 'max_features': 'sqrt', 'min_samples_leaf': 2}
```

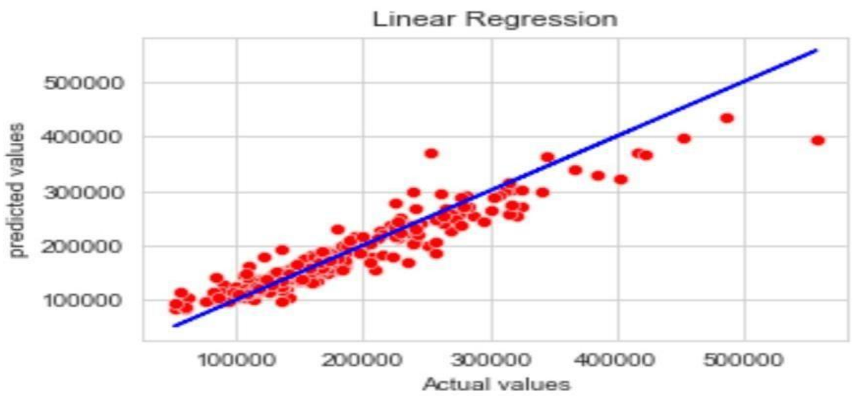
Using the best parameters for random forest regression.

```
#using the parameters obtained above
rf=RandomForestRegressor(criterion='mse',max_features='sqrt',min_samples_leaf=2)
rf.fit(x_train,y_train)
y_pred=rf.predict(x_test)
r2_rf=r2_score(y_test,y_pred)*100
print("r2 score=",r2_rf)
cv_rf=cross_val_score(rf,x,y,cv=3).mean()*100
print("cross validation score is",cv_rf)
print("mean absolute error=",mean_absolute_error(y_test,y_pred))
mse_rf=mean_squared_error(y_test,y_pred)
print("mean squared error=",mse_rf)
rmse_rf=np.sqrt(mse_rf)
print("root mean squared error=",rmse_rf)
```

```
r2 score= 88.47672625773646
cross validation score is 85.10617197502054
mean absolute error= 17090.358375781656
mean squared error= 647440020.1321051
root mean squared error= 25444.842702050744
```

Best fit line for random forest regression.

```
plt.figure(figsize=(5,3))
sns.scatterplot(x=y_test,y=y_pred,color='r')
plt.plot(y_test,y_test,color='b')
plt.xlabel("Actual values")
plt.ylabel("predicted values")
plt.title("Linear Regression")
plt.show()
```



Various evaluation metrics are used on different algorithms. The results of them are given below in the snapshots of each algorithm. Mean Squared Error computes the difference between the mean squared of the predicted values and the actual values; Root Mean Square Error computes the square root of the Mean Squared Error. RMSE is applied to all the four algorithms. R squared or the coefficient of determination is the measure of the closeness of predicted data with the actual data.

With all the comparison metrics, it is observed that Random Forest Regression shows the least Root Mean Square Error with a value of 25552.9797. Therefore, there are fewer errors in this metric as compared to others. Along with this, the R-Squared Score is 0.883786, which is 88.37% for Random forest Regression.

Mean Squared Error is also lowest for the Random Forest Regression algorithm with the value of 652952128.330.

The other evaluation metric used, Mean absolute Error also gives the least value with Random Forest. Apart from Random Forest Regression, Linear Regression also showed satisfactory results with R-Squared score for the same as 0.8558, or 85.58%. The predictions, however, were not very good.

- **Key Metrics for success in solving problem under consideration**

The objective of Regression is to find a line that minimizes the prediction error of all the data points. The essential step in any machine learning model is to evaluate the accuracy of the model. The Mean Squared Error, Mean absolute error, Root Mean Squared Error, and R-Squared or Coefficient of determination metrics are used to evaluate the performance of the model in regression analysis.

- The Mean absolute error represents the average of the absolute difference between the actual and predicted values in the dataset. It measures the average of the residuals in the dataset.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}|$$

Where,

$\hat{y}$ − *predicted value of y*

$\bar{y}$ − *mean value of y*

- Mean Squared Error represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2$$

- Root Mean Squared Error is the square root of Mean Squared error. It measures the standard deviation of residuals.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

- The coefficient of determination or R-squared represents the proportion of the variance in the dependent variable which is explained by the linear regression model. It is a scale-free score i.e. irrespective of the values being small or large, the value of R square will be less than one.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

- Adjusted R squared is a modified version of R square, and it is adjusted for the number of independent variables in the model, and it will always be less than or equal to R^2.In the formula below n is the number of observations in the data and k is the number of the independent variables in the data.

$$R^2_{adj} = 1 - \left[\frac{(1 - R^2)(n-1)}{n-k-1}\right]$$

- Differences among these evaluation metrics

  Mean Squared Error (MSE) and Root Mean Square Error penalizes the large prediction errors vi-a-vis Mean Absolute Error (MAE). However, RMSE is widely used than MSE to evaluate the performance of the regression model with other random models as it has the same units as the dependent variable (Y-axis).

  MSE is a differentiable function that makes it easy to perform mathematical operations in comparison to a non-differentiable function like MAE. Therefore, in many models, RMSE is used as a default metric for calculating Loss Function despite being harder to interpret than MAE.

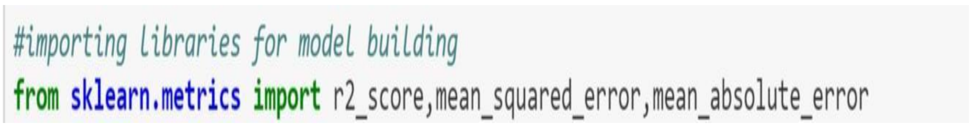
  MAE is more robust to data with outliers.

The lower value of MAE, MSE, and RMSE implies higher accuracy of a regression model. However, a higher value of R square is considered desirable.

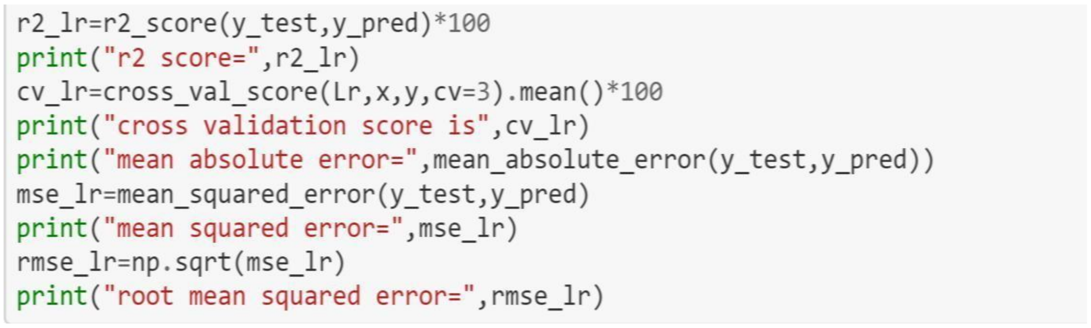
R Squared & Adjusted R Squared is used for explaining how well the independent variable in the linear regression model explains the variability in the dependent variable. R Squared value always increases with the addition of the independent variables which might lead to the addition of the redundant variables in our model. However, the adjusted R-squared solves this problem.

Adjusted R squared takes into account the number of predictor variables, and it is used to determine the number of independent variables in our model. The value of Adjusted R squared decreases if the increase in the R square by the additional variable isn't significant enough.

For comparing the accuracy among different linear regression models, RMSE is a better choice than R Squared.



```
#importing libraries for model building
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
```



```
r2_lr=r2_score(y_test,y_pred)*100
print("r2 score=",r2_lr)
cv_lr=cross_val_score(Lr,x,y,cv=3).mean()*100
print("cross validation score is",cv_lr)
print("mean absolute error=",mean_absolute_error(y_test,y_pred))
mse_lr=mean_squared_error(y_test,y_pred)
print("mean squared error=",mse_lr)
rmse_lr=np.sqrt(mse_lr)
print("root mean squared error=",rmse_lr)
```

Therefore, if comparing the prediction accuracy among different linear regression (LR) models then RMSE is a better option as it is simple to calculate and differentiable. However, if your dataset has outliers then choose MAE over RMSE.

Besides, the number of predictor variables in a linear regression model is determined by adjusted R squared, and choose RMSE over adjusted R squared if you care about evaluating prediction accuracy among different LR models.

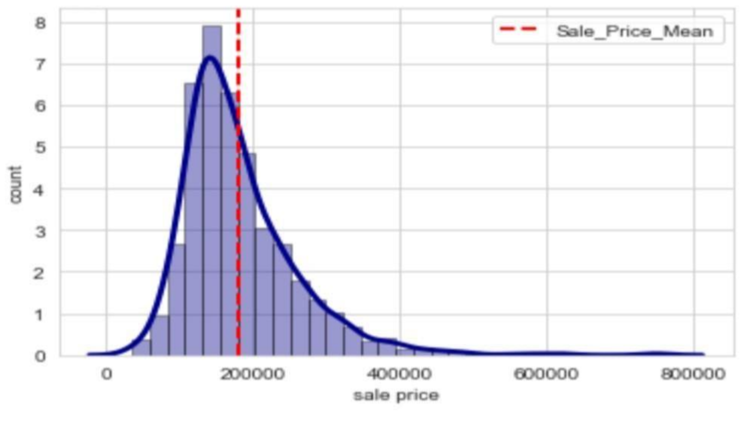
	Model	r2 Score	Cross Validation score	RMSE
1	RandomForest regressor	88.476726	85.106172	25444.842702
0	Linear regression	85.589468	80.085879	28454.558101
3	KNN regressor	80.308904	75.637651	33261.878764
2	Decision Tree regressor	64.025726	73.134415	44958.089469

Different evaluation metrics are used on the models to predict the accuracy with the least error. A comparison of different algorithms of Machine Learning is shown in above table.

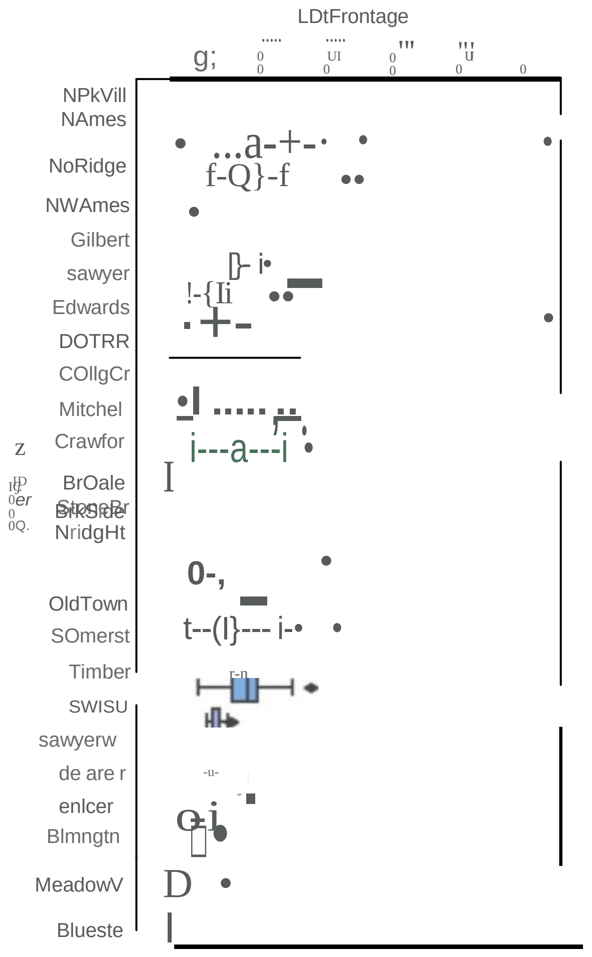
• Visualizations

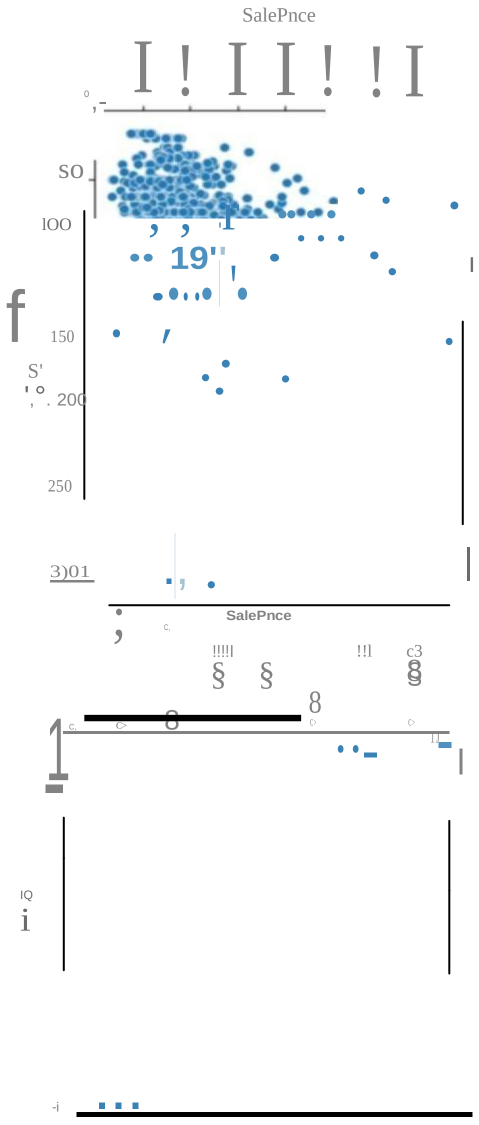
As a data scientist, we always question the amount of time we put into data visualization. Don’t we? Ideally we should put more emphasis and efforts into ensuring a thorough analysis rather than just making the graphs pretty and investing lot of time in just decorating them .Prettier graphs won’t necessarily mean great analysis. Data visualization plays a very important role of presenting data in a powerful and credible way.

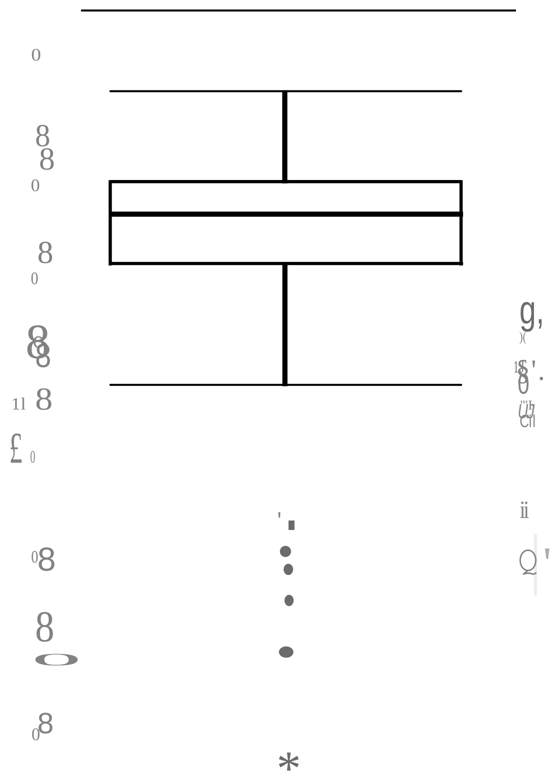
It is quit useful to have a quick overview of different features distribution v/s house price. Hence we shall have a quick look at few important features through visualizations.

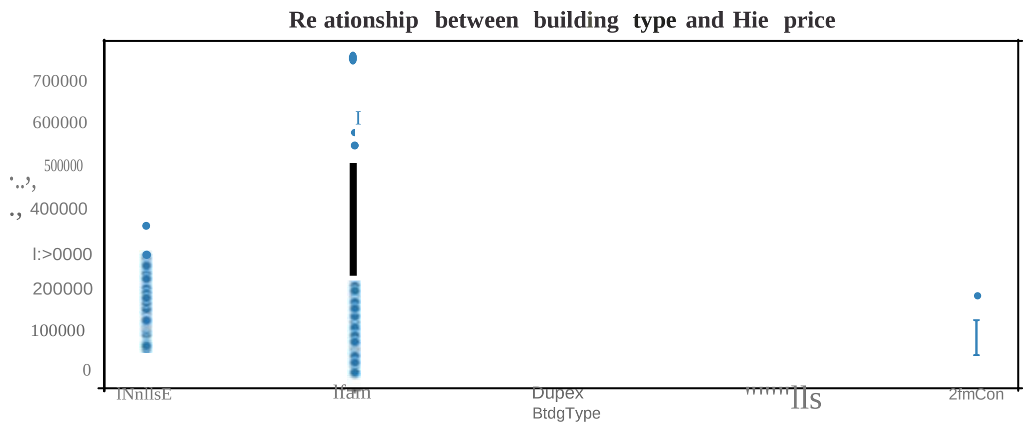


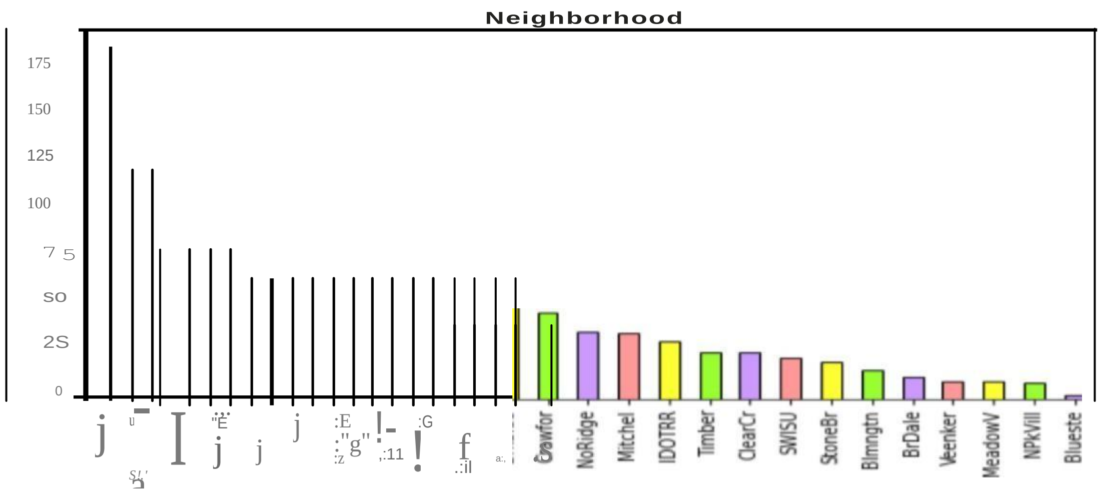
```
The mean value of sale price is 181477.0059931507
Highest price= 755000
minimum sale price= 34900
```

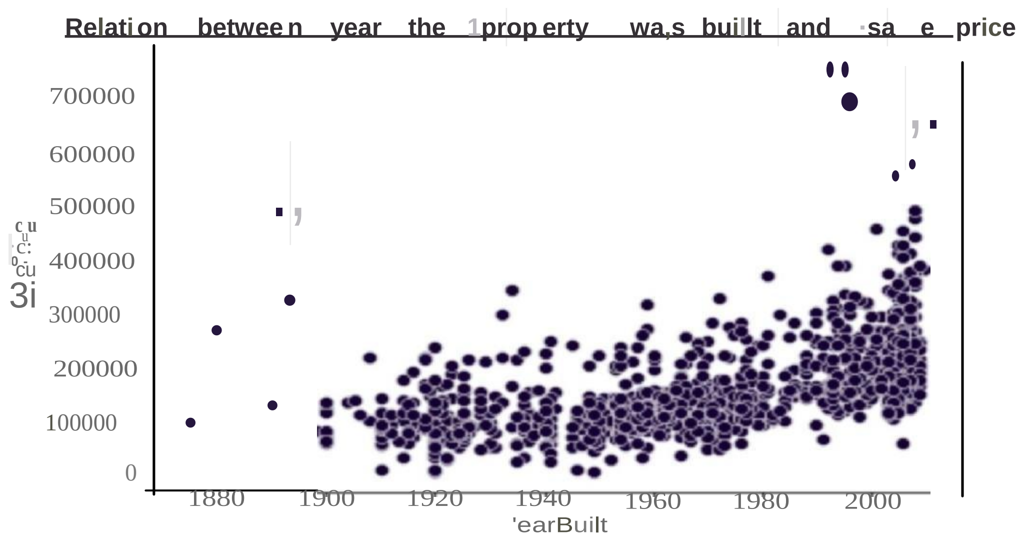

LDtFrontage
NPkVill
NAmes
NoRidge
NWAmes
Gilbert
sawyer
Edwards
DOTRR
COllgCr
Mitchel
Crawfor
BrOale
StoneBr
BrkSide
NridgHt
OldTown
SOmerst
Timber
SWISU
sawyerw
de are r
enlcer
Blmngtn
MeadowV
Blueste

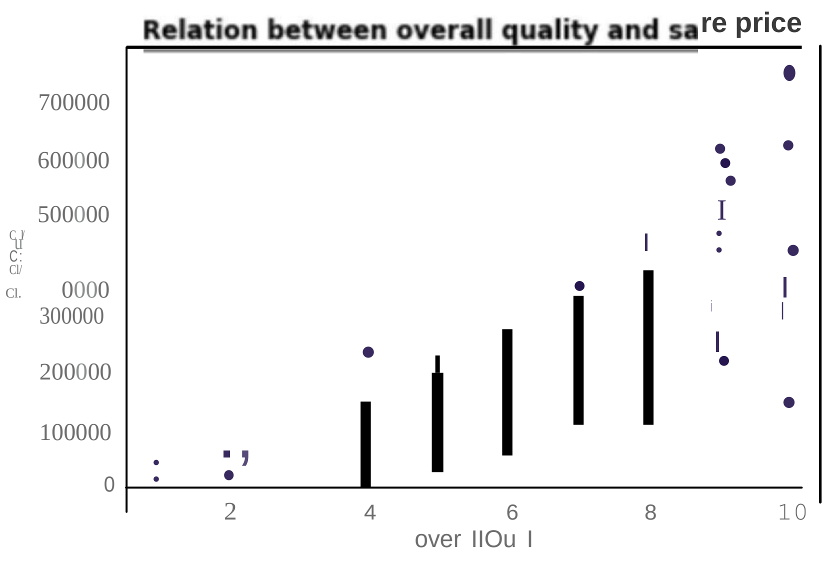

SalePnce
SO
lOO
150
200
250
SalePnce

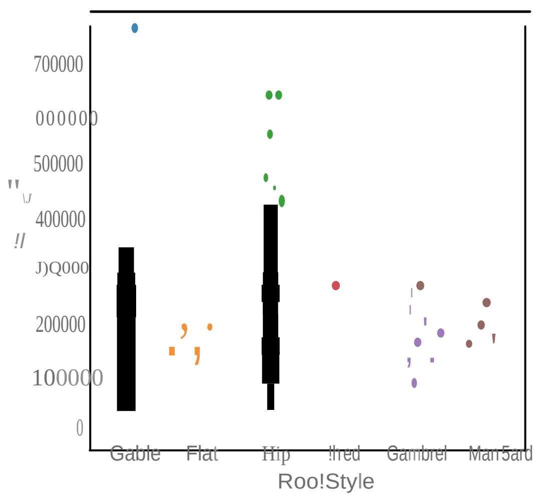


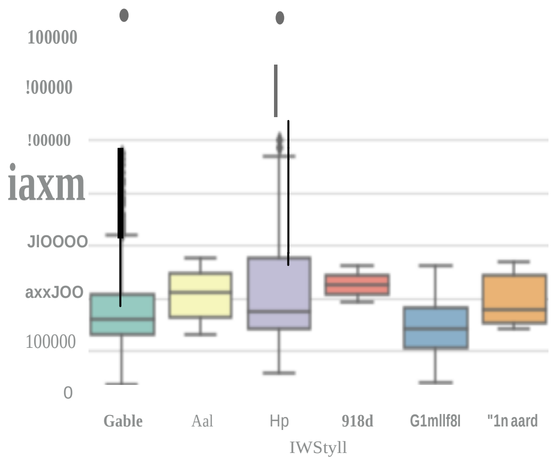

Re ationship between building type and Hie price
700000
600000
500000
400000
200000
100000
0
1Fam
Dupex
BtdgType
2fmCon


Neighborhood
175
150
125
100
0
Crawfor
NoRidge
Mitchel
IDOTRR
Timber
ClearCr
SWISU
StoneBr
Blmngtn
BrDale
Veenker
MeadowV
NPkVill
Blueste

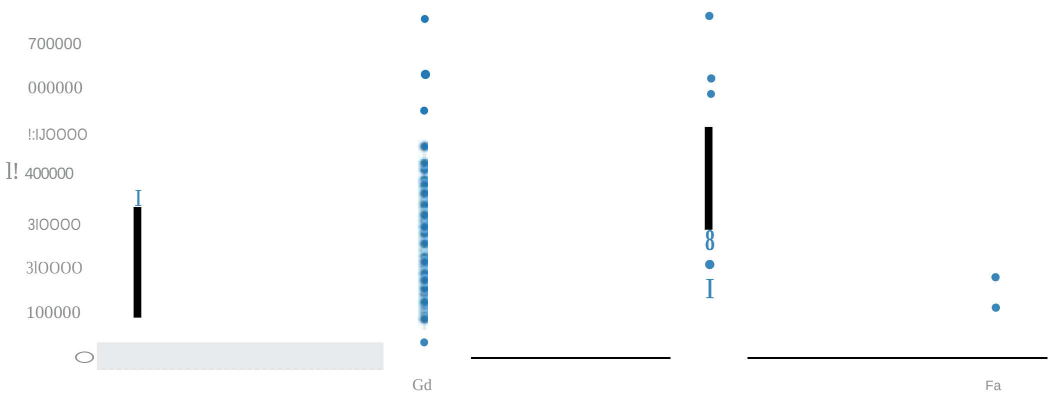

Relation between year the property was built and sale price
700000
600000
500000
400000
300000
200000
100000
0
1880
1900
1920
1940
1960
1980
2000
'earBuilt

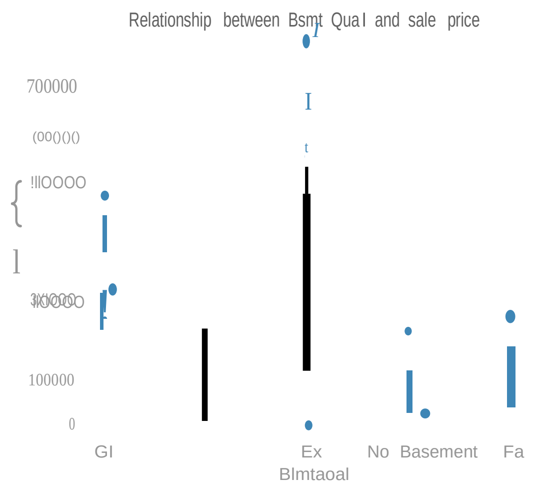


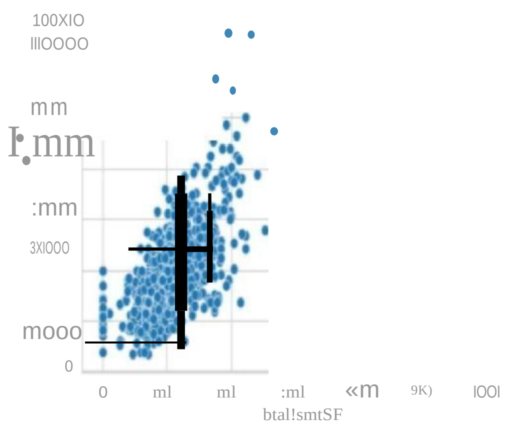


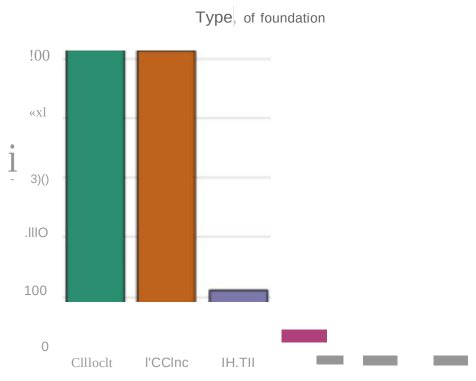


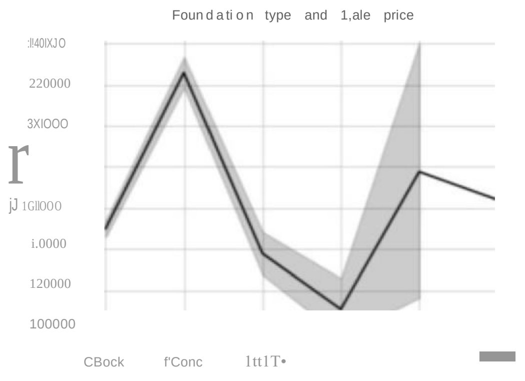
Relationship b tween exterior quality and sa'le price

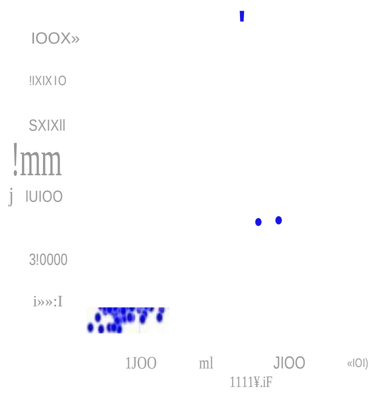


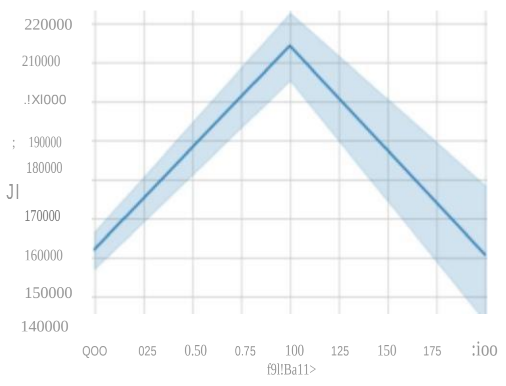

Relationship between Bsmt Qual and sale price
700000
100000
0
GI
Ex
Blmtaoal
No Basement
Fa

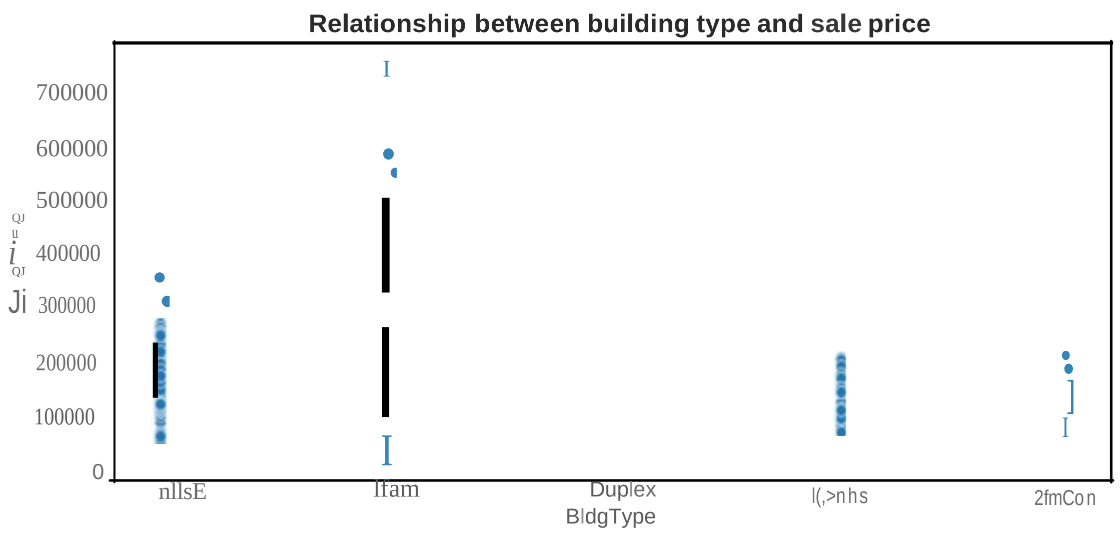

3XI000
mooo
0
btal!smtSF

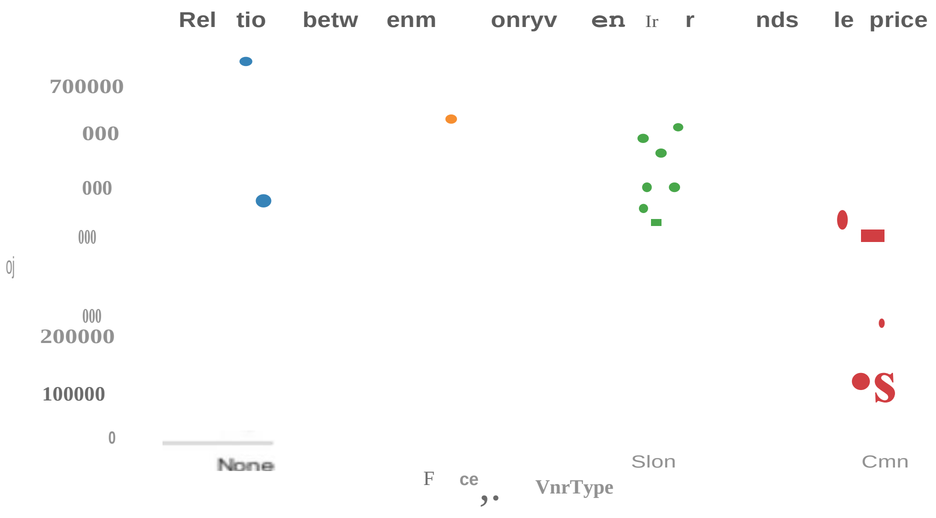

Type, of foundation
!00
100
0
CIllocIt
I'CCInc
IH.TII

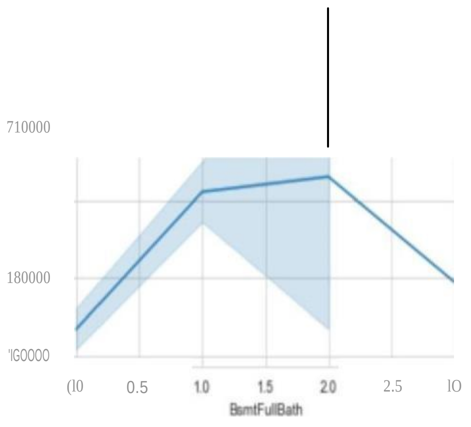

Foundation type and 1,ale price
220000
120000
100000
CBock
f'Conc

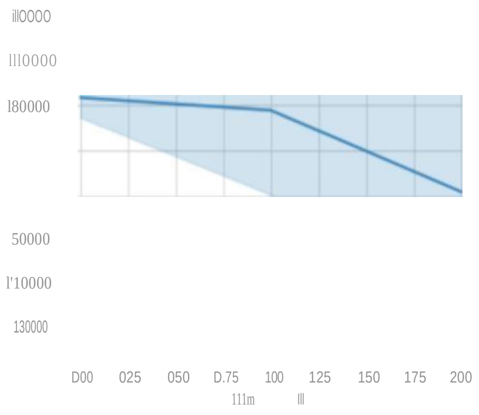

IOOX»
SXIXII
3!0000
1JOO
JIOO

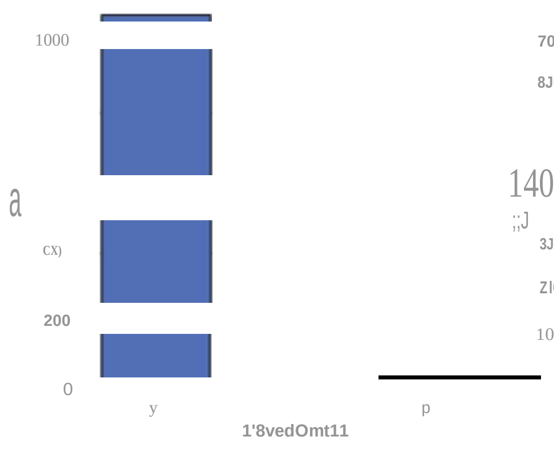

220000
210000
190000
180000
170000
160000
150000
140000
QOO
025
0.50
0.75
100
125
150
175


Relationship between building type and sale price
700000
600000
500000
400000
300000
200000
100000
0
nllsE
Ifam
Duplex
l(,>nhs
2fmCon
BldgType

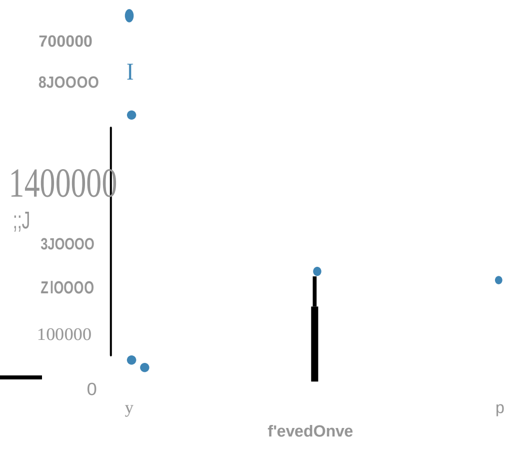

Rel tio betw enm onryv en Ir r nds le price
700000
000
000
000
000
200000
100000
0
None
F ce
VnrType
Slon
Cmn


710000
180000
'IG0000
(I0
0.5
1.0
1.5
2.0
2.5
IO
BsmtFullBath

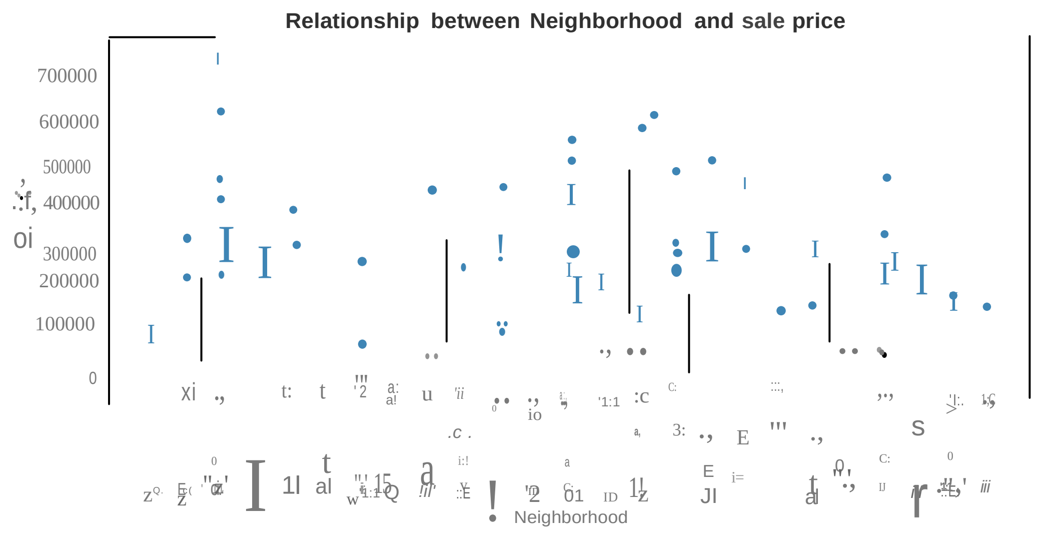

ill0000
lll0000
l80000
50000
l'10000
130000
D00
025
050
D.75
100
125
150
175
200



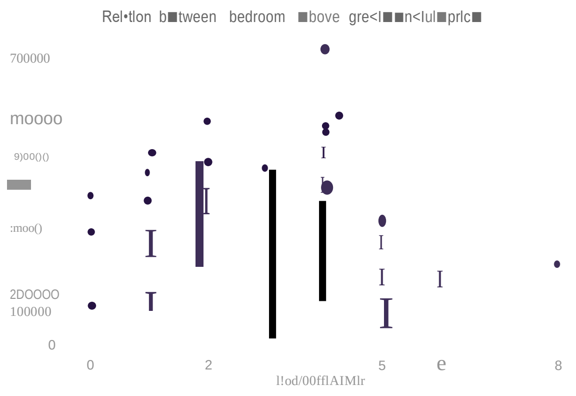
1000
200
0
y
p
1'8vedOmt11



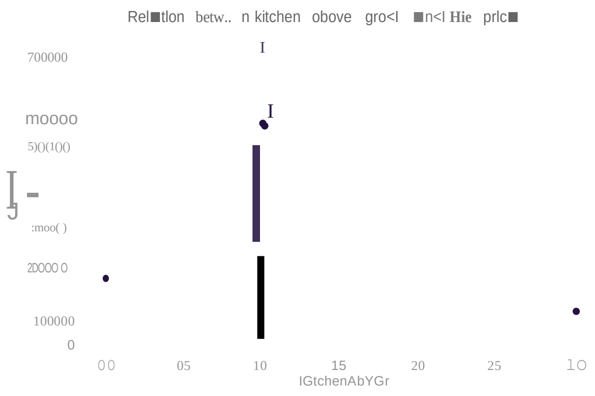
700000
1400000
100000
0
y
p
f'evedOnve



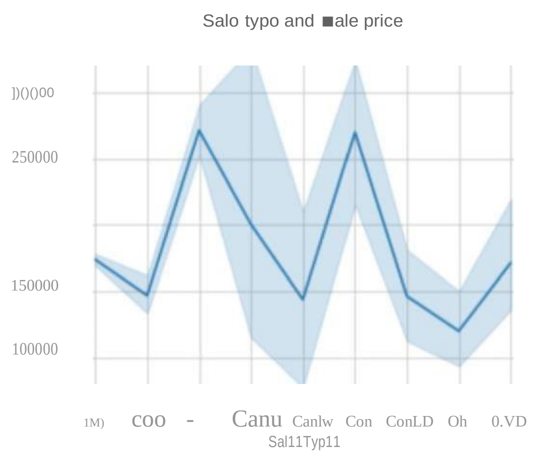
Relationship between Neighborhood and sale price
700000
600000
500000
400000
300000
200000
100000
0
Neighborhood

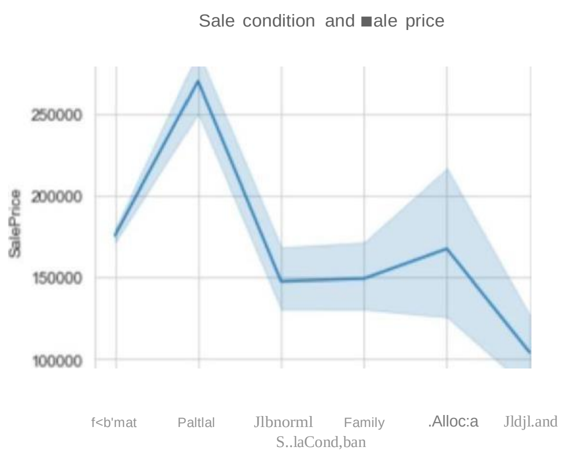


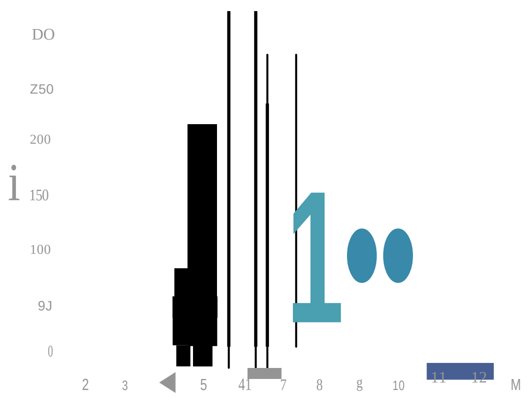
Rel•tlon b■tween bedroom ■bove gre<l■■n<lul■prlc■
700000
100000
0
2
5
8

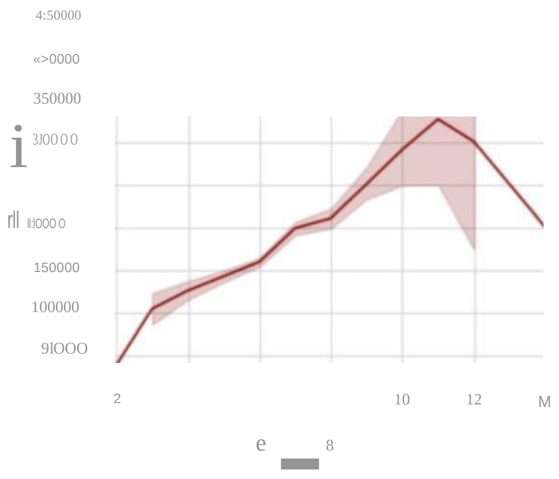


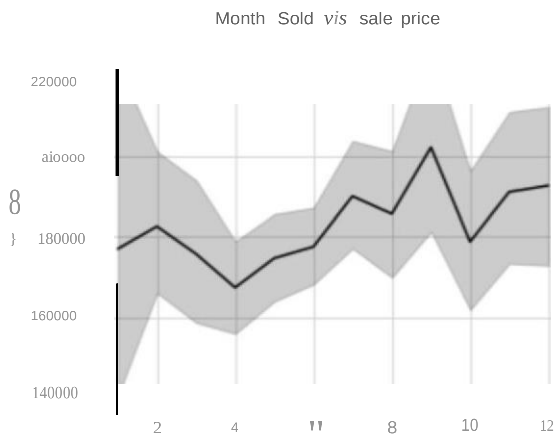
Rel■tlon betw.. n kitchen obove gro<l ■n<l Hie prlc■
700000
100000
0
05
10
15
20
25
IGtchenAbYGr

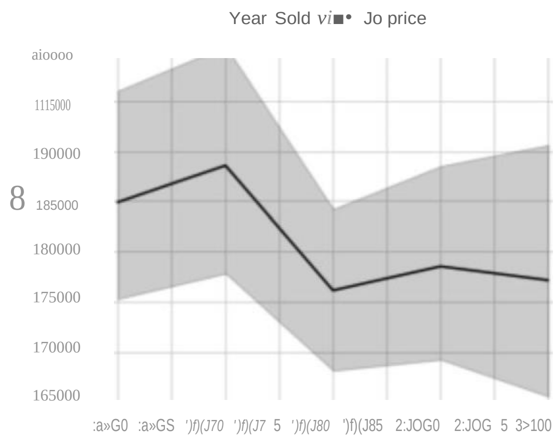

Salo typo and ■ale price
])()()00
250000
150000
100000
1M)
coo
-
Canu
Canlw
Con
ConLD
Oh
0.VD
Sal11Typ11


Sale condition and ■ale price
250000
200000
150000
100000
SalePrice
f<b'mat
Paltlal
Jlbnorml
Family
.Alloc:a
Jldjl.and
S..laCond,ban


DO
Z50
200
150
100
9J
0
2
3
5
41
7
8
g
10
11
12
M


4:50000
«>0000
350000
3J0000
ll:l0000
150000
100000
9lOOO
2
10
12
M
e
8


Month Sold vis sale price
220000
aioooo
180000
160000
140000
2
4
8
10
12


Year Sold vi■• Jo price
aioooo
1115000
190000
185000
180000
175000
170000
165000

- ## Interpretation of the Results

After performing exploratory data analysis the following inferences can be drawn:

The numerical data was not symmetrical and needed cleaning and pre-processing

There were a few null values in the dataset which were filled using simple imputer and assigning different class to categorical Nan values.

Out of 80 features we have built a model using only 55 features which were most relevant to the target.

Columns with highest positive correlation with the target are:

a. GrLivArea(0.71)
b. GarageCars (0.63)
c. GarageArea (0.62)
d. TotalBsmtSF(0.60)
e. 1stFlrSF (0.59)
f. FullBath(0.55)
g. TotRmsAbvGrd(0.53)
h. YrBuilt(0.51)
i. YrRemodAdd(0.51)

High Standard deviations could be seen in

a. LotArea
b. BsmtFinSf1
c. BsmtFinSF2
d. BsmtunfSF
e. WoodDeckSF
f. Miscall Val
g. Sale Price

There were few Columns with no outliers, they are:

a. YrRemodAdd,
b. Full bath,
c. HalfBath ,
d. GarageCars,
e. Mosold and Yrsold,

The total basement area distribution was dense between 1000 to 2000 sq. and the sale prices there ranged from 100k to 450k

The above grade living area ranged from 334 square feet to maximum 5642 sq.ft.

Most of the properties had areas as 864 sq.ft ,1040 sq.ft and 890 sq.ft

The mean value of living area above grade is 1525

Maximum number of houses (516) had foundation of type CBlock Cinder Bloc,PConc Poured Contrete type of foundation could be seen in 513 properties.Stone and wood type of foundation was the least. BrkTil Brick & Tile foundation was available for 112 properties.Slab Foundations showed the least price that was 110000.Properties with stone foundations had higher prices than properties with slab foundation and wood foundation.

For Height of basement most of the houses had average height and Good heights. Very few properties had no basement at all. There were many properties with No walkout or garden level walls. When the basement height was good and typical average (TA) most of the properties had no brick and garden walls. No basement heights were rated as poor. For excellent Basement quality the sale prices were very high few as 70K.

As for all the properties utilities were same that were basic electricity, gas, water and septic tanks these factors did not affect the Sale price of properties as these are the basic facilities that a property would include.

Neighbourhood i.e. Physical locations within Ames city limits had a great impact on the sale prices. Neighbourhood had a co-relation co-efficient of 0.73 with the target. Norridge i.e. Northridge and NridgHt i.e. Northridge Heights had the sale prices starting from a higher rate as well as ending with maximum sale prices as compared to other locations. Northridge and North Ridge heights had higher lot frontage area. North park villa, Bluestem, Briar Dale and Meadow areas had less sale prices. Hence we could say that if a person wishes to buy a property with high Lot Frontage area properties available in North Ridge and Northridge Height he will have to pay high prices for the house the price could be in a range from 2 lakhs to 7 lakh and anywhere in between it.

Talking about the overall quality of the properties it is quite evident that people would prefer good quality finish and material for the house, this in return has quite great effect on the sale prices. The properties with very excellent quality of material and finish had sale prices starting from 200000 and went as high as 750000.Properties with poor, very poor and fair kind of qualities had the minimum sale prices that ranged between 1, 00,000 to 2, 00,000.The better the overall quality of properties, higher the sale prices. No wonder the Overall Quality feature had a correlation of 0.79 with the target (Sale price).

In terms of Exterior quality and the present condition of material most of the houses had Average/Typical and Good quality. There were no houses in poor condition. Sale prices were high for good and excellent exterior quality.

Poured Concrete type of foundation could be seen in 513 properties. Properties with poured concrete foundation recorded the highest price which was 230000.Slab Foundations showed the least price that was 110000.Properties with stone foundations had higher prices than properties with slab foundation and wood foundation

Taking into consideration the time period during which the property was built or renewed and its effect on the sale price, it is quite evident from the graphs provided that the earlier built properties did not gain good sale prices. Customer often preferred recently constructed and renewed properties as is it a general assumption that the older properties would have been degraded over time in terms of quality and other physical aspects such as colour, fancy infrastructures etc. The year built feature had a good correlation of 0.51 with the sale price. The properties that were built during year 1900 to 1980 had sale prices ranged from 1 lakh to 2 lakh with few outliers but for the year 1990 and 2000 the minimum sale increased from 1 lakh to 1.5 lakh with a maximum of 7.5 lakh. Also properties built before 1960 underwent renovation in large amount.

Most of the houses did not had masonry veneer area.354 houses had Brick Face type of masonry veneer area. Masonry veneer area with Brick Common was the least. There were no houses with Cinder Block masonry veneer area houses with Stone. Stone masonry veneer area was moderate in number (98). Prices for No masonry area were dense in range 50000 to 350000.Houses with Brick common masonry veneer area had the lowest prices concentrated between 100000 to 150000

For roof style most of the properties had Gable and Hip type of roofs out of which Hip was more expensive.

The above grade living area had following characteristics:

The above grade living area ranged from 334 square feet to maximum 5642 sq.ft.Most of the properties had areas as 864 sq.ft ,1040 sq.ft and 890 sq.ft.The mean value of living area above grade is 1525

In case of electrical system Standard Circuit Breakers & Romex were the maximum. The sale price for standard CB were the maximum it was above 180000FuseA Fuse Box over 60 AMP and all Romex wiring (Average) was second highest. For mix type of electrical systems the sale prices were below 50000 which was the least among all. Properties with FuseF and FuseP type of electrical system had same prices (~110000).

Properties with paved drive had higher sale prices. Also most of them had paved drive way.

Basement Quality: the sale price for properties with no basement ranged from 10000 to 20000.For excellent basement quality i.e. basement with 100+ inches the sale prices started at 150000 and went up to 750000.For typical average the sale price were from 50000 to 470000.For fair basements that were between 70-79 inches the price range was up to 200000.

1 family detached family building type was maximum and the sale prices could be seen over all the range.

CONCLUSION

Real estate is a rapidly growing business. Each year, more and more people are buying houses. People take into consideration several features before buying a house. House price prediction predicts house pricing based on different features. Real estate prices vary due to a wide variety of attributes. The machine learning-based model is a substantial and feasible way to forecast real estate prices, and can provide relatively competitive and satisfactory results.

In this project, 55 features are used to predict the sale price .Many evaluation techniques like Mean Squared Error, Root Mean Square Error, Mean absolute Error, and R-Squared Score are used on different machine learning algorithms, which are Linear Regression, KNN Regression, Decision Trees Regression, and Random Forests Regression. After comparing all the algorithms, it is concluded that Random Forest Regression gives the best result. This helps the buyers to predict the price of the housing more accurately.

We have cut down the features from 80 to 56 with the help of inferences from visualizations and EDA performed which reduces the redundancy and has helped built a better model. There is a scope to reduce the number of features too but we have limited it to 55 so that the prediction prices are dependent on more than just few factors which shall be helpful in future.

With all the comparison metrics, it is observed that Random Forest Regression shows the least Root Mean Square Error with a value of 25552.9797. Therefore, there are fewer errors in this metric as compared to others. Along with this, the R-Squared Score is 0.883786, which is 88.37% for Random forest Regression.

Mean Squared Error is also lowest for the Random Forest Regression algorithm with the value of 652952128.330.

The other evaluation metric used, Mean absolute Error also gives the least value with Random Forest. Apart from Random Forest Regression, Linear Regression also showed satisfactory results with R-Squared score for the same as 0.8558, or 85.58%. The predictions, however, were not very good.

Features that greatly contribute to house price predictions were locality (neighbourhood) ,above grade living area ,parking capacity of the properties ,Basement area and area above grade, facilities like electricity, water, gas septic tanks, fire place, number of full and half baths, the year of construction, Overall quality of the house including exterior and material quality.

Few features like MS Zone, Miscellaneous features, land slope, month in which property was sold, and etc. did not make any significant impact on the prediction of sale prices.

Thus Machine learning models, including linear regression, KNN regression, decision tree regression and random forest regression, were used in this investigation to forecast house prices.

• Learning Outcomes of the Study in respect of Data Science

There is a huge pressure on real estate industry to unlock the potential of data science and incorporate machine learning evidences-based approaches in their work flows. Property valuation is an imprecise science. Individual appraisers and valuers bring their own experience, metrics and skills to a job. Consistency is difficult, with UK and Australian-based studies suggesting valuations between two professionals can differ by up to 40%.

Perhaps a well-trained machine could perform this task in place of a human, with greater consistency and accuracy. Thanks to machine learning and data analytics, real estate professionals and investors are now able to make more accurate property assessments than ever before.

In today's real estate world, it has become tough to store such huge data and extract them for one's own requirement. Also, the extracted data should be useful. The system makes optimal use of the Regression Algorithm. The system makes use of such data in the most efficient way. The regression algorithm helps to fulfil customers by increasing the accuracy of estate choice and reducing the risk of investing in an estate. A lot's of features that could be added to make the system more widely acceptable.

The decision to purchase real estate is undeniably very essential in the life of most adults. Thus, the appraisal and prediction of real estate can provide useful information to help facilitate real estate transactions. Real estate prices vary due to a wide variety of attributes. The machine learning-based model is a substantial and feasible way to forecast real estate prices, and can provide relatively competitive and satisfactory results.

In the near future, data science will have an important role to play, as it will be able to not only improve a business strategy but also improve the way and quality of our lives. Data science which works in tandem with Artificial Intelligence (AI) will be able to analyse behaviour, interests and preferences in order to propose the ideal apartment for each client. This will mean that clients interested in a property will be able to visit it on their smartphone, projecting themselves into what it would be like to live there, by eliminating a wall or changing the colours for example.

A successful data-driven approach can yield powerful insights. In one example, an application combining a large database of traditional and non-traditional data was used to forecast the three-year rent per square foot for multifamily buildings in Seattle. These machine-learning models predicted rents with an accuracy rate that exceeded 90 per cent.

Adopting data analytics and AI into existing processes can be particularly valuable Technologies such as AI and machine learning can use data to improve efficiency by identifying patterns and opportunities, predicting future scenarios, and even automating certain tasks

• Limitations of this work and Scope for Future Work

To date, most developers and investors continue to make heuristic, or instinct-based, decisions rather than informed ones. Many are unaware of the wide variety of datasets, lack the analytical capabilities to generate insights, and/or are resistant to change. Some may be unsure where to start; others may not know which new skills and capabilities should be added to begin. Here are some of the key barriers:

Lack of awareness about new datasets and analytical techniques

Data risks and uncertainty of return on investment (ROI). New technologies come with their own risks. While the new datasets have merit, there are several inherent risks, perhaps more than using traditional data. Investors and managers would have to do due diligence to verify the authenticity of the origin of the data. There could be privacy risks if the sourced data includes personally identifiable information or material non-public information. If there is a lack of transparency about the data collection method, there will likely be concerns around its accuracy and validity. Finally, investor organizations may not be confident about the ROI of using alternative data.

For future work, diverse data types such as comments of real estate attributes, prices from social media, images from Google maps, and economic indicators are possible sources added as inputs for machine learning models to improve forecasting accuracy.

Also, a mobile application can be developed by the researchers to sort the houses of their choice and need, with the help of advanced machine learning algorithms for intelligently showing the data of customers' interest. Further, more emphasis can be given to modelling the house prices through tree-based boosting techniques such as XGBoost and CatBoost.

One of the major future scopes is adding estate database of more cities which will provide the user to explore more estates and reach an accurate decision. More factors like recession that affect the house prices shall be added. In-depth details of every property will provide ample details of a desired estate. This will help the system to run on a larger level

Another potential opportunity for future research might be the use of deep learning techniques to forecast real estate prices.

Real estate companies can start investing in property matching online software, which can determine if a property is a good match in terms of investment for each customer. Models that allow this have been developed by using a variety of public data and market information such as prices per square metre on past transactions, number of bedrooms and the quality of the neighbourhood. This provides clients with much wider parameters than could be provided by a single estate agent and allows the estate agent to offer greater accuracy in their information, for by example, giving clients accurate house prices instead of estimations.

• References:

1. Fik T J, Ling D C and Mulligan G F 2003 Modeling spatial variation in housing prices: a variable interaction approach. Real Estate Economics, 31(4), 623-46.
2. Varma A, Sarma A, Doshi S and Nair R 2018 House Price Prediction Using Machine Learning and Neural Networks. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) 1936-39. IEEE.
3. Bhagat N, Mohokar A and Mane S 2016 House price forecasting using data mining. International Journal of Computer Applications, 152(2), 23-26.
4. Amri S and Tularam G A 2012 Performance of multiple linear regression and nonlinear neural networks and fuzzy logic techniques in modelling house prices. Journal of Mathematics and Statistics, 8(4), 419-434.
5. Vineeth N, Ayyappa M and Bharathi B 2018 House Price Prediction Using Machine Learning Algorithms. In International Conference on Soft Computing Systems, 425 - 33.
6. Alfiyatin A N, Febrita R E, Taufiq H and Mahmudy W F 2017 Modeling house price prediction using regression analysis and particle swarm optimization. International Journal of Advanced Computer Science and Applications, 8.
7. Afonso B, Melo L, Oliveira W, Sousa S and Berton L 2019 Housing Prices Prediction with a Deep Learning and Random Forest Ensemble. In Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional 389-400. SBC.
8. McKinsey & Company 2018|Article Getting ahead of the market: How big data is transforming real estate