Report on

Housing Price Prediction

Submitted by:

PRANKUL JAIN

ACKNOWLEDGMENT

Foremost, I would like to express my deepest and sincere gratitude towards the team at “Flip Robo Technologies” for their continuous support towards the project and other facets of the assignment along with offering a position as an Intern.

Besides them, extending my gratitude towards the academic team at “DataTrained” for their continuous mentoring sessions and their support in the knowledge transfer sessions.

Finally, from the very onset on my assignment I’ve extensively used online platforms like stackoverflow, w3school, kaggle and many others. I’m obligated to pay my gratitude to every personage involved in making content on these platforms.

INTRODUCTION

- Business Problem Framing

Houses are one of the necessary need of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.

- Conceptual Background of the Domain Problem

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?

• Review of Literature

With the acceleration of housing prices in the United States there has been increased attention given to the causes and effects of the fluctuations in housing prices and investment. Models have been built in the past to investigate the movement in the prices using the past data. Under certainty, the model exhibits a balanced aggregate growth, but with underlying sectoral change. In particular, the model shows how housing prices can have a “bubbly” appearance in which housing wealth rises faster than income for an extended period, then collapses and experiences an extended decline.

The primary objective of most prediction models is to establish a relationship between the current economy and the house price. Most buyers look forward to buying homes with optimum value for the amenities available. While a seller focus on the return on investment after property evaluation.

- Motivation for the Problem Undertaken

Designing a model to predict house prices that are mostly apart of a volatile market with intangible facets contributing to its nature is a challenge on its own. Apart from this primarily as a civil engineer I take deep interest in contributing to an industry closely related to me. While this model helps the organization manipulate their business strategy to yield high returns, also contributing to better understand the dynamics of the market.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  The dataset constitutes two forms- one, training data and the other is testing data. Our goal is to build a model that effectively can predict the house price by learning the fed data. The dataset is a combination of multiple datatypes like objects, int (integer) and float. The target data (house price) is observed to be a float data indicating that it is a regression model. A few features in the dataset have NaN values present (Not a Number), there input features are filled with relevant data. Furthermore, visualization analysis helped understand the relationship between the input features and the target column. Adding to this, checking for outliers and skewness were done before removing noise through relevant methods like zscore and yeo-johnson method. Eventually building multiple regression models to find the best fit.

- Data Sources and their formats

  The dataset was assigned to me by “Flip Robo technologies” where I’m designated duties of an Intern. The dataset is of CSV (comma separated values) format.

The train dataset has the 1168 rows and 81 columns while the test dataset has 292 rows and 80 columns.

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1163	289	20	RL	NaN	9819	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1164	554	20	RL	67.0	8777	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1165	196	160	RL	24.0	2280	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1166	31	70	C (all)	50.0	8500	Pave	Pave	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1167	617	60	RL	NaN	7861	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0

1168 rows × 81 columns

The train dataset

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	ScreenPorch	PoolArea	PoolQC	Fence	MiscFea
0	337	20	RL	86.0	14157	Pave	NaN	IR1	HLS	AllPub	...	0	0	NaN	NaN	
1	1018	120	RL	NaN	5814	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	
2	929	20	RL	NaN	11838	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
3	1148	70	RL	75.0	12000	Pave	NaN	Reg	Bnk	AllPub	...	0	0	NaN	NaN	
4	1227	60	RL	86.0	14598	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
287	83	20	RL	78.0	10206	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
288	1048	20	RL	57.0	9245	Pave	NaN	IR2	Lvl	AllPub	...	0	0	NaN	NaN	
289	17	20	RL	NaN	11241	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	S
290	523	50	RM	50.0	5000	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
291	1379	160	RM	21.0	1953	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	

292 rows × 80 columns

The train dataset

- Data Preprocessing Done

  - First step is to import all the necessary libraries. I have initially imported Pandas, Numpy , Matplotlib.pyplot and seaborn.
  - Statically analysis like value_counts, unique values, shape and a few others.
  - Check the information available on the dataset. This dataset was observed to have close to 95% of NaN (not a number) values. As we proceed I’ll include

all the assumptions made and methods used to fill the null values.

- NaN values in “PoolQC” were assumed to be houses without Pool.
- NaN values in “MiscFeature” were assumed to have no miscellaneous features.
- NaN values in “Alley” were assumed to have no alley access.
- NaN values in “Fence” were assumed to have No fence.
- NaN values in “FireplaceQu” were assumed to have no fire place.
- NaN values all basement related features were assumed to have no basement.
- The same assumtions were followed for both test and train datasets.
- While for NaN values in the columns “Electrical”,”MasVnrArea” and “MasVnrType” in the test dataset were filled by mode, mean and mode methods respectively.
- Since the ID column has all unique values, this feature wouldn’t help in prediction.

- Data Inputs- Logic- Output Relationships

Exploratory data analysis (EDA) is conducted across the dataset. The object here is to establish a relationship between the target column and the input data. To achieve this I’ve made use of boxplot,kdeplot/distplot, scatter plot,

barplot and strip plot. Most numeric continuous data have a linear relationship with the taget column (house price).

- # Hardware and Software Requirements and Tools Used

Hardware necessary:

- RAM- 8GB or above
- Processor – Core i5 or above
- SSD- 250GB or above

Software necessary:

- Anaconda

Libraries:

- import pandas as pd: pandas is a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a numpy matrix array. This makes pandas a trusted ally in data science and machine learning.
- import numpy as np: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting,

selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

- import seaborn as sns: Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.
- Import matplotlib.pyplot as plt: matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.
- from sklearn.ensemble import RandomForestRegressor
- from sklearn.tree import DecisionTreeRegressor
- from sklearn.ensemble import GradientBoostingRegressor
- from sklearn.model_selection import cross_val_score

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  As mentioned the null values are filled by making a few assumptions and appropriate methods (mean/mode) and the outliers are checked and removed by percentile method since zscore method had over 10% of data loss. Yeo-Johnson method is used to removed skewness while correlation is calculated using Pearson's correlation coefficient. Along with these methods I've also made used of VIF to determine the multi-collinearity.

- Testing of Identified Approaches (Algorithms)

  Given my target column is continuous in nature the model building is made using regression. My final model is selected based on the r2 score and cross validation score by determining their difference. To assess the best fit model, I've run multiple regression models, the models are enlisted below.

  - RandomForestRegressor
  - GradientBoostingRegressor
  - DecisionTreeRegressor

- Run and Evaluate selected models

  - Random Forest Regressor:
    With accuracy of 90%.

RandomForest Regressor

In [125]:
```
RFR=RandomForestRegressor()
RFR.fit(X_train,y_train)
pred=RFR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(RFR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 90.04671127256715
mean_squared_error: 599093012.4047372
mean_absolute_error: 16793.267464387463
root_mean_squared_error: 24476.376619196257

Cross validation score : 82.72414647013342

R2_Score - Cross Validation Score : 7.322564802433732
```

  - Decision Tree Regressor:
    With accuracy of 70.9%

DecisionTree Regressor

In [124]:
```
from sklearn.tree import DecisionTreeRegressor
from sklearn import metrics
from sklearn.model_selection import cross_val_score
DTR=DecisionTreeRegressor()
DTR.fit(X_train,y_train)
pred=DTR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(DTR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 70.98521219681808
mean_squared_error: 1746413382.079772
mean_absolute_error: 29288.803418803418
root_mean_squared_error: 41790.11105608325

Cross validation score : 60.07149948862425

R2_Score - Cross Validation Score : 10.913712708193835
```

- Gradient Boost Regressor:
  With accuracy score 91.1%

GradientBoosting Regressor

In [126]:
```
from sklearn.ensemble import GradientBoostingRegressor
GBR=GradientBoostingRegressor()
GBR.fit(X_train,y_train)
pred=GBR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(GBR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 91.11573192554091
mean_squared_error: 534748169.12215793
mean_absolute_error: 15592.048095145745
root_mean_squared_error: 23124.622572534194

Cross validation score : 83.58436444216586

R2_Score - Cross Validation Score : 7.53136748337505
```

- Grid search CV

GridSearch Cv for RandomForest Regressor

```
from sklearn.model_selection import GridSearchCV
param = {"criterion":["squared_error","absolute_error","poisson"],"max_features":["auto","sqrt","log2"],"bootstrap":[True,False],
clf = GridSearchCV(RFR,param_grid=param)
```

```
clf.fit(X_train,y_train)
```

```
GridSearchCV(estimator=RandomForestRegressor(),
             param_grid={'bootstrap': [True, False],
                         'criterion': ['squared_error', 'absolute_error',
                                       'poisson'],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'n_estimators': [50, 100, 150, 200]})
```

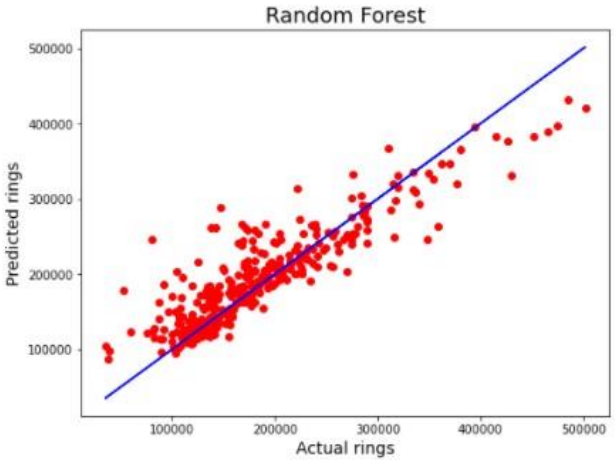
```
print(clf.best_params_)
print(clf.best_score_)
```

```
{'bootstrap': False, 'criterion': 'poisson', 'max_features': 'log2', 'n_estimators': 100}
0.7076408619318525
```

```
RFR=RandomForestRegressor(bootstrap=False,criterion='poisson',max_features='log2',n_estimators=100,random_state=42)
RFR.fit(X_train,y_train)
pred=RFR.predict(X_test)
R2_score = r2_score(y_test,pred)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred)))

#cross validation score
scores = cross_val_score(RFR, X, y, cv = 10).mean()*100
print("\nCross validation score :", scores)

#difference of accuracy and cv score
diff = R2_score - scores
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_score: 78.42112167331264
mean_squared_error: 1298842581.0877426
mean_absolute_error: 24767.58894586894
root_mean_squared_error: 36039.458668073006

Cross validation score : 75.38534929947181

R2_Score - Cross Validation Score : 3.035772373840828
```

```
plt.figure(figsize=(8,6))
plt.scatter(x=y_test,y=pred,color="r")
plt.plot(y_test,y_test,color="b")
plt.xlabel("Actual rings",fontsize=14)
plt.ylabel("Predicted rings",fontsize=14)
plt.title("Random Forest",fontsize=18)
plt.show()
```



- Grid search CV

GridSearch Cv for GradientBoosting Regressor

```
param = {"loss":["squared_error", "absolute_error", "huber", "quantile"],"criterion":["friedman_mse","squared_error","mse","mae"]
clf = GridSearchCV(GBR,param_grid=param)
```

```
clf.fit(X_train,y_train)
```

```
GridSearchCV(estimator=GradientBoostingRegressor(),
             param_grid={'criterion': ['friedman_mse', 'squared_error', 'mse',
                                       'mae'],
                         'loss': ['squared_error', 'absolute_error', 'huber',
                                  'quantile'],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'n_estimators': [50, 100, 150, 200]})
```

```
print(clf.best_params_)
print(clf.best_score_)
```

```
{'criterion': 'mae', 'loss': 'huber', 'max_features': 'sqrt', 'n_estimators': 200}
0.8245020684372859
```

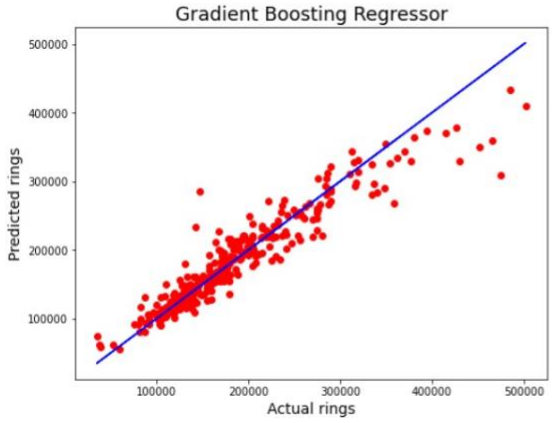
```
GBR=GradientBoostingRegressor(criterion="mse",loss="huber",max_features="log2",n_estimators=100,random_state=9)
GBR.fit(X_train,y_train)
pred_gbr=GBR.predict(X_test)
R2_score = r2_score(y_test,pred_gbr)*100
print('R2_score:',R2_score)
print('mean_squared_error:',metrics.mean_squared_error(y_test,pred_gbr))
print('mean_absolute_error:',metrics.mean_absolute_error(y_test,pred_gbr))
print('root_mean_squared_error:',np.sqrt(metrics.mean_squared_error(y_test,pred_gbr)))
```

```
R2_score: 89.23696348789251
mean_squared_error: 647832102.8595073
mean_absolute_error: 16224.063932714307
root_mean_squared_error: 25452.546097777868
```

```
#difference of accuracy and cv score
scores = cross_val_score(GBR, X, y, cv = 9)
scores=scores.mean()
diff = R2_score - (scores)*100
print("\nR2_Score - Cross Validation Score :", diff)
```

```
R2_Score - Cross Validation Score : 4.695520142467998
```

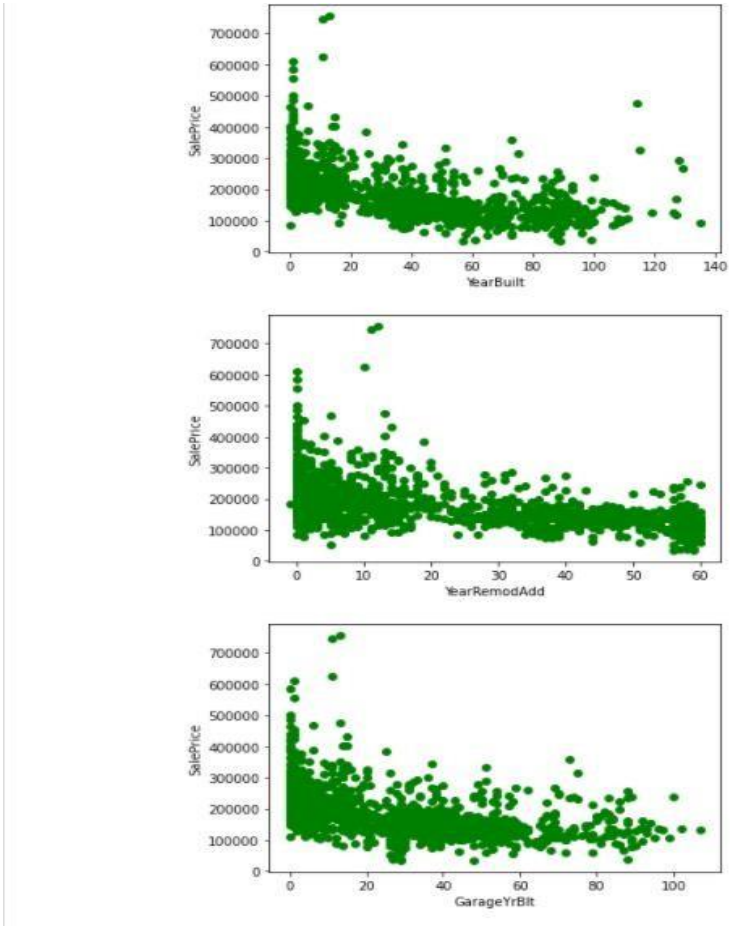
```
plt.figure(figsize=(8,6))
plt.scatter(x=y_test,y=pred_gbr,color="r")
plt.plot(y_test,y_test,color="b")
plt.xlabel("Actual rings",fontsize=14)
plt.ylabel("Predicted rings",fontsize=14)
plt.title("Gradient Boosting Regressor",fontsize=18)
plt.show()
```



- Key Metrics for success in solving problem under consideration

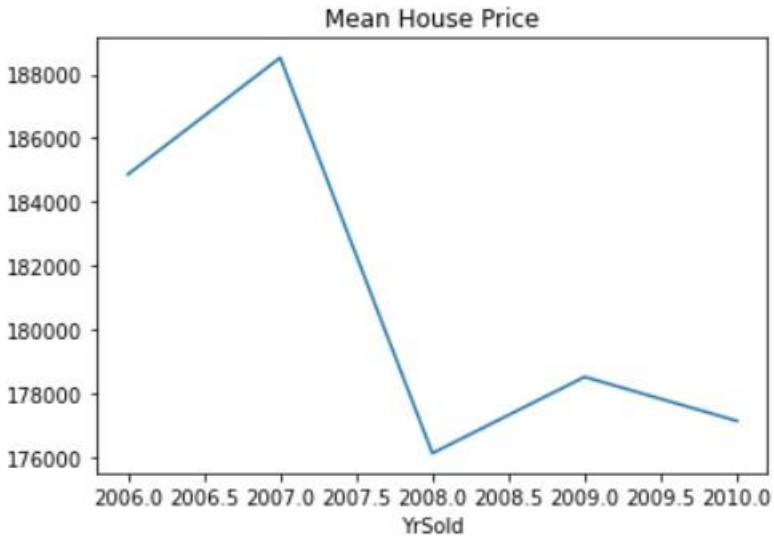
  Below are the metrics used:
  - mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation
  - root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions.
  - r2 score which tells us how accurate our model is.

- Visualizations



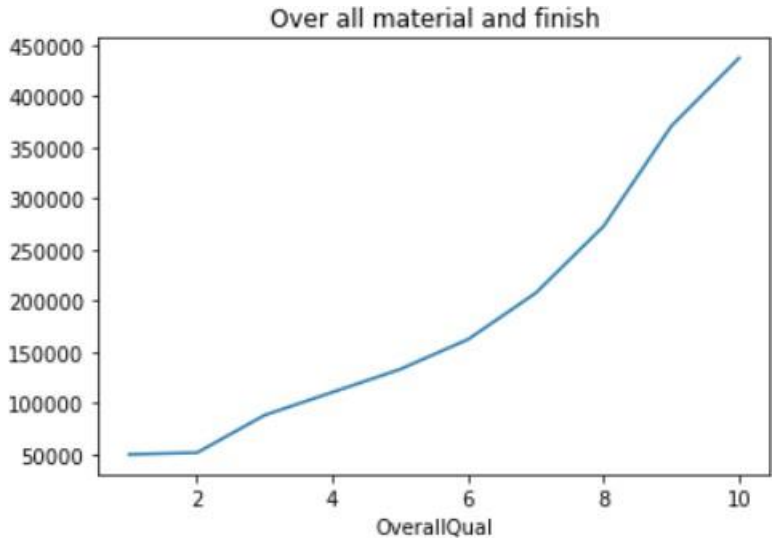
Observation:

1. Newer houses have higher price.
2. Newer house remodeling has higher prices
3. Newer garage construction has higher price.



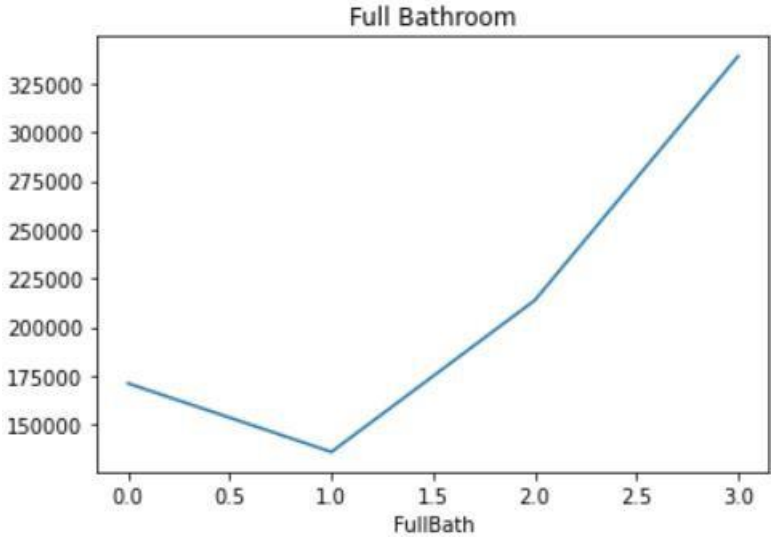
Observation:

1. evidently there was a crash in the real estate towards the end of 2007 and begin of 2008. This could be a result of economic crisis during the period. While a peak was noticed between 2006 and 2007.



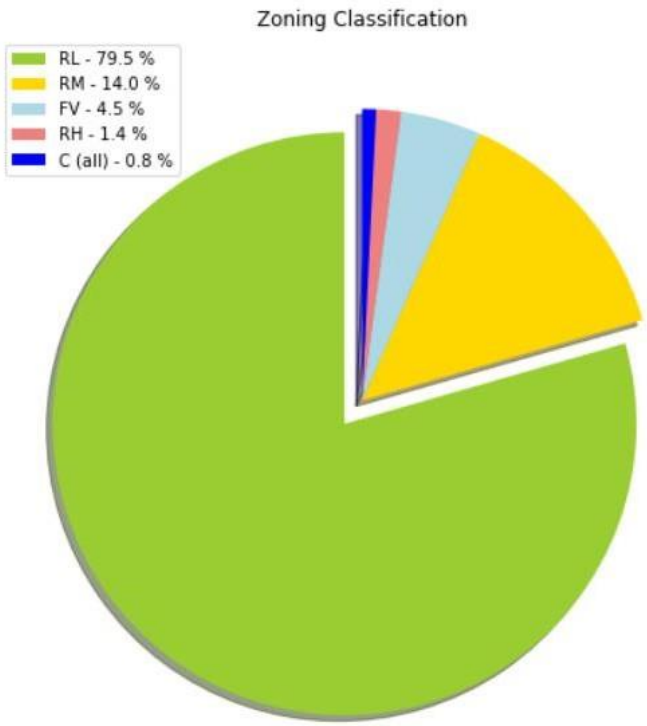
Observation:

1. *As the Over quality of the house increases the price also increases.*



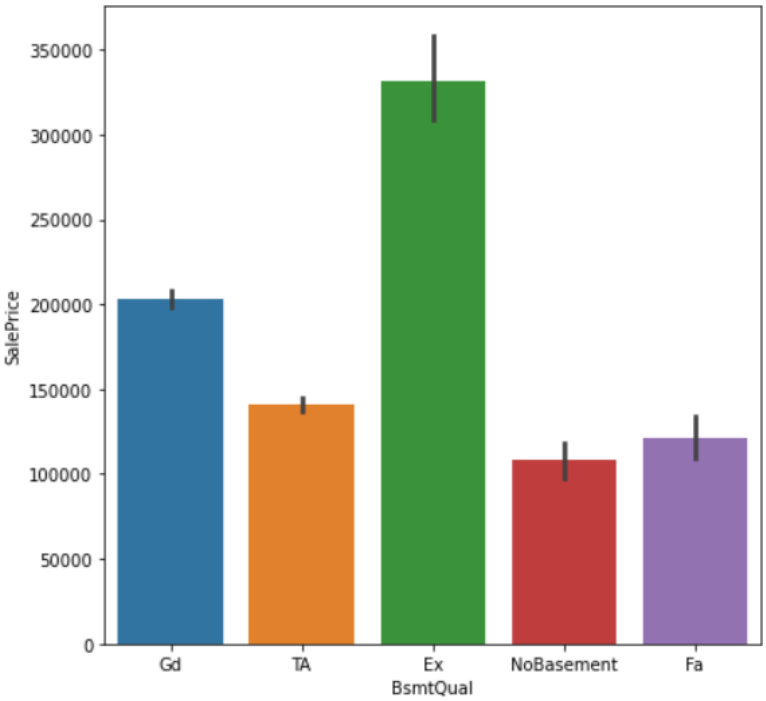
Observation:

1. *Full bathroom over 1.0 contributes in linear increase in price.*



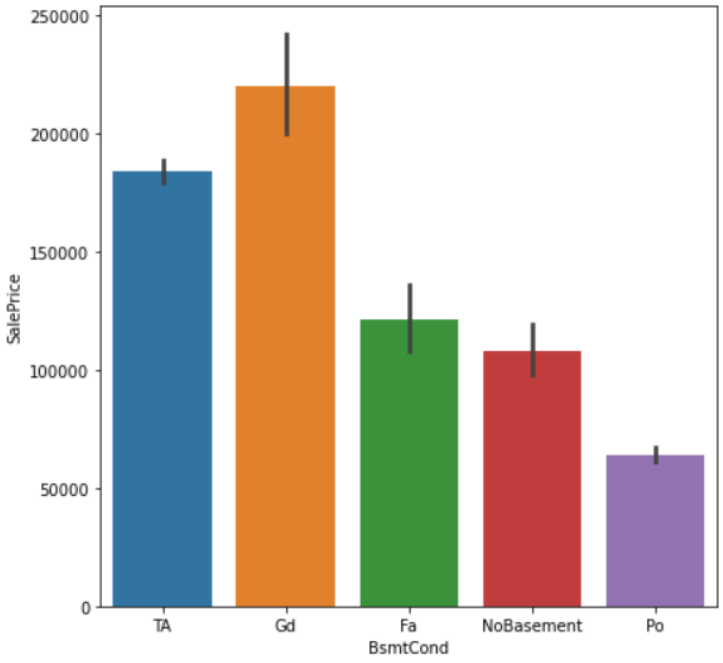
Observation

1. RL consumes over 79.5% of the total.
2. C only contributes 0.8%



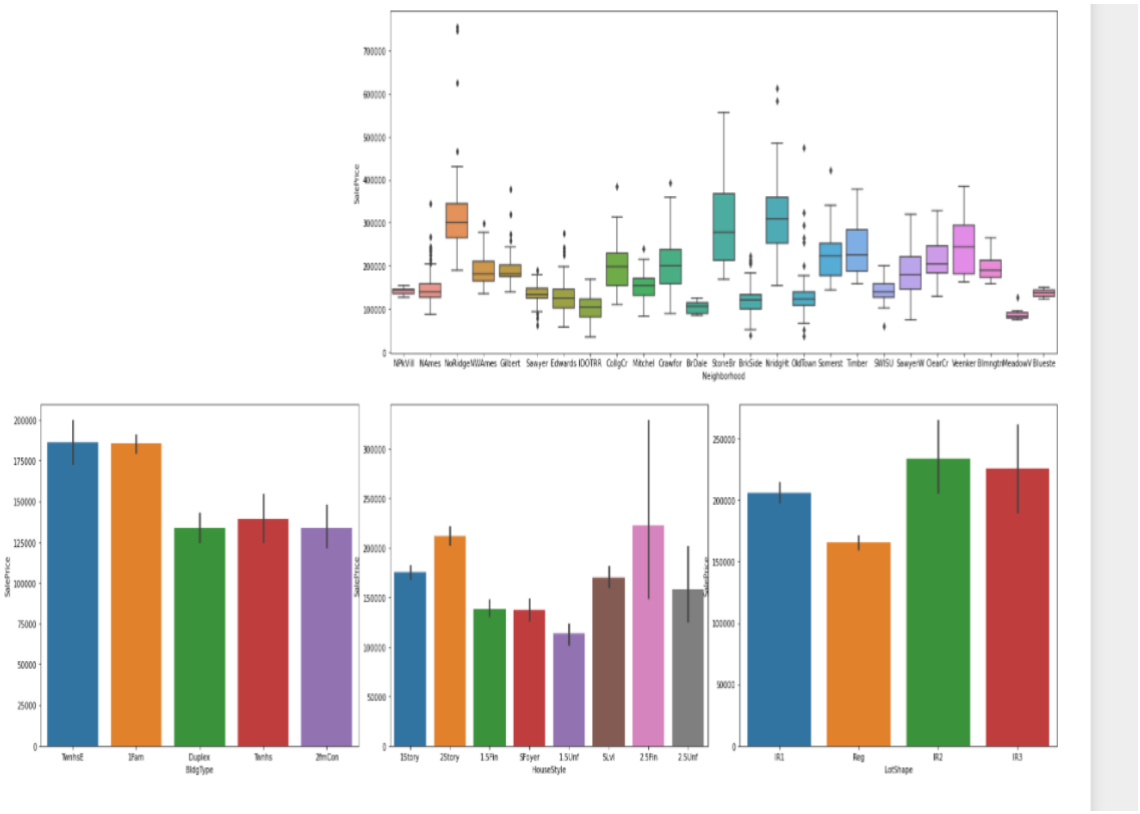
Observation:

1. *Most houses having excellent and good basement have higher prices.*



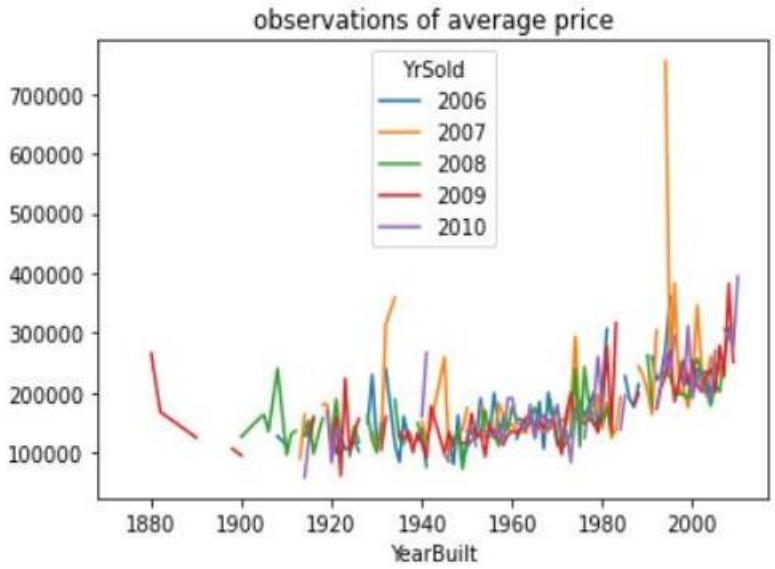
Observation:

1. Most basements are good/typical that contribute to higher price.



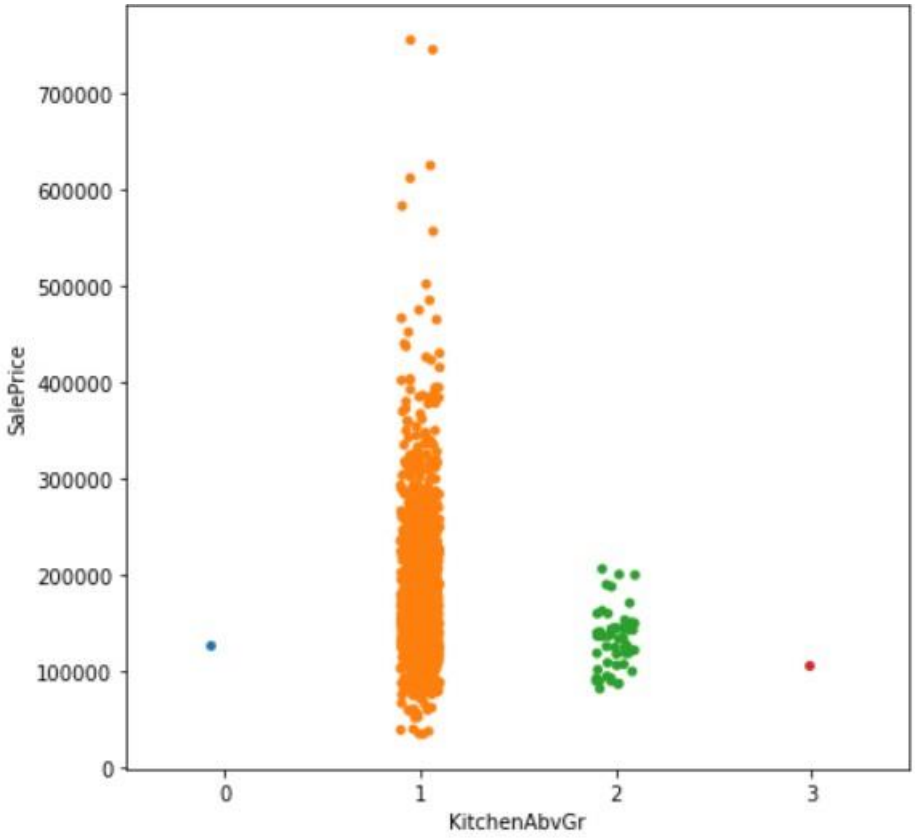
Observation:

1. 2.5 storey dwelling are priced higher.
2. Townhouse end unit are priced higher.
3. slightly irregular properties are priced higher.



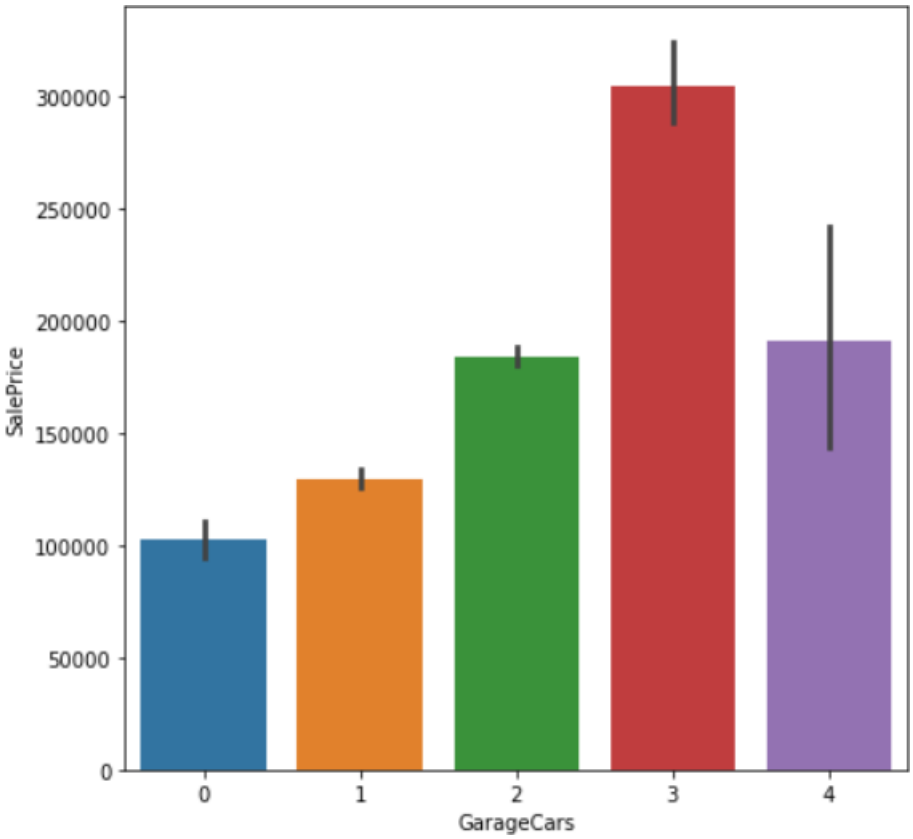
Observation:

1. The year 2007 has seen a sharp peak.



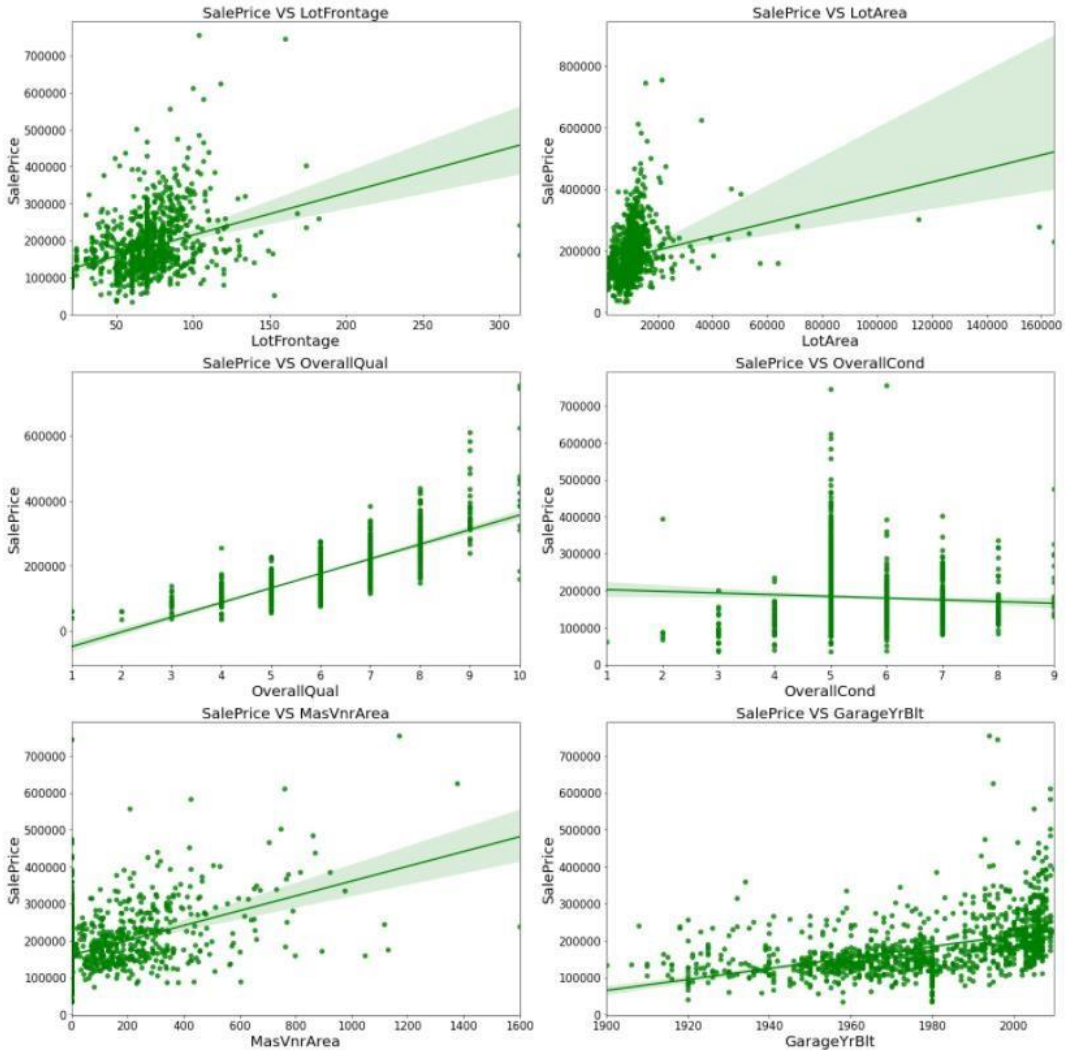
Observation

1. Most kitchens have 1 kitchen above grade.



Observation:

1. A notable increase in the price as the garage capa city increases. At 4 cars there seems to be a dip in price.



Observation:

1. *As linear feet of street increses the price increases.*
2. *As lot area increses there is an increse in the sale price.*
3. *As overall quality increses the sale prices increses.*
4. *With increse in garage built year there isn't a sharp increase in the price.*
5. *At overall condition 5 (average) has the highest sale price while the consecutive condition see a declined trend.*

- Interpretation of the Results

To begin with, the dataset is composed of two separate datasets – test and train. All pre-processing was simultaneously done on both the train and the test data. While the dataset has several features with null values, these values were filled with appropriate assumptions and methods as described earlier. In the next step, the dataset presented itself with outliers and skewness, they were dealt by using methods like percentile, z-score and Yeo-Johnson method. Multiple regression models were built after scaling the dataset. To pick the best fit model we ran hyper parameter tuning. Eventually used the model to predict the house price in the test data.

CONCLUSION

- Key Findings and Conclusions of the Study

The study throws light on how volatile the real estate market can be and how complicated the intangible features associated them can be. Understanding and deal with such data is complex on its own, with prediction this adds a new dimension to the complexity. Given the inconsistency of the dataset in terms on NaN values the project posed new complexes. By identifying the correlation between the target feature and the input features we can establish the features that are more relevant to us. Hence, making building a model easier with definitive accuracy. To conclude, although the application of this model is only in the early stage with research in progress I'm certain we can achieve higher accuracy for most geographical locations with more features.

- Learning Outcomes of the Study in respect of Data Science

Machine learning and its application has helped serval organisations in advancing themselves both in terms of brand name and monetization. With new

advances in machine learning we have consumed more power. One such power is visualization, the projection of this helps us understand the interaction between the target and the input features. Data cleaning, the process of replacing the missing or null values with its closest replacement with relation to its feature column.

This study has only been an experiment in attempting to understand the implementation of machine learning algorithms in the prediction of housing prices.

- ## Limitations of this work and Scope for Future Work

The dataset is inconsistent with plenty of null values. Followed by data leakage that takes place once the test and the train data are merged. Another drawback would be the extreme amount of outliers and the skewness present in the dataset. While these are dealt with appropriate methods but perfectly engineered dataset would make all our lives easier. Engineering a stronger model could easily eliminate the presence on any third party(agents/brokers/evaluators) interventions in price negotiations or in evaluations with reference to the current economic condition.