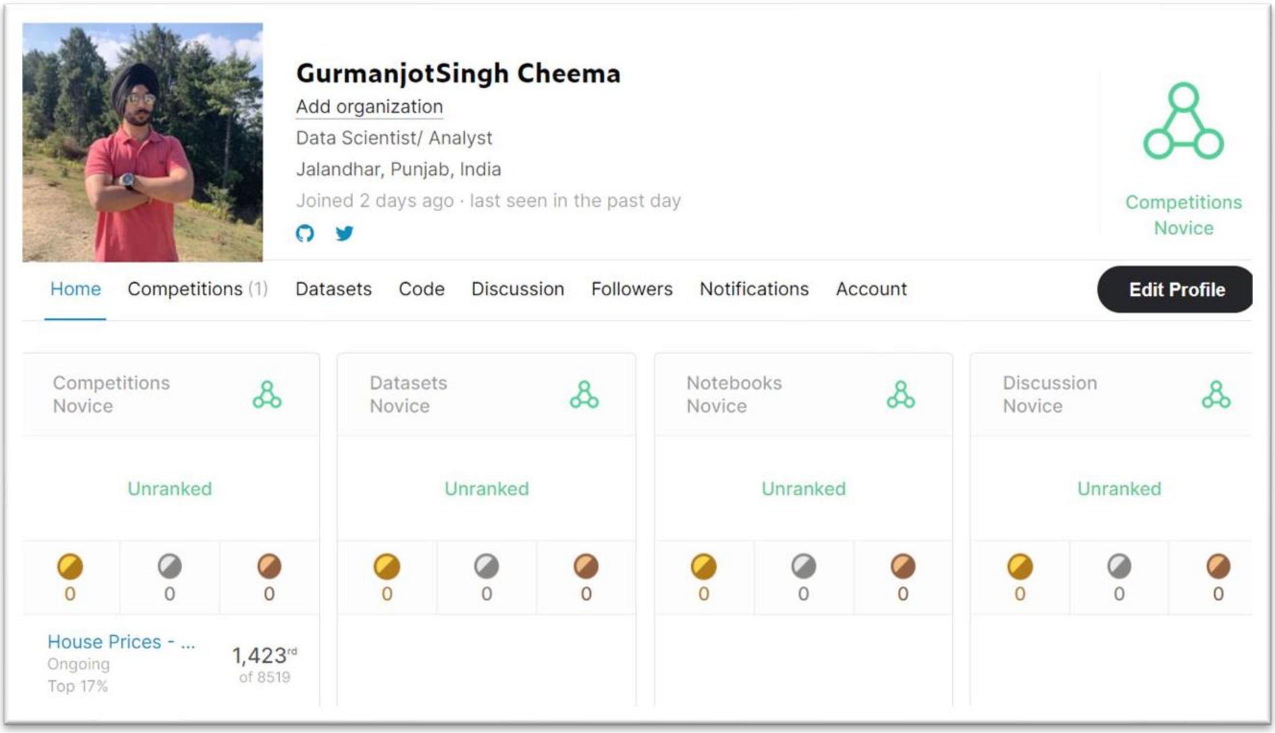
Kaggle House Price Competition

Data Science & Machine Learning in Canada

Prepared by



Background Information

There has been a significant advancement in the technological environment in every field from health sector to business and trade. It is the technology and Big Data solutions that are driving these sectors towards more growth. Machine learning tools combined with Big Data can give a perfect foresight about any industry. Housing prices are an important reflection of the economy, and housing price ranges are of great interest for both buyers and sellers. In this project. house prices will be predicted given explanatory variables that cover many aspects of residential houses. As continuous house prices, they will be predicted with various regression techniques.

Purpose

The purpose of this report is to showcase the nature of the dataset briefly and what steps were taken in order to handle and prepare the dataset for the model training phase. The primary goal is to choose the best model with high accuracy and low root mean square log error as it was the criteria for ranking the predictions made by the model on Kaggle. Various performance metrics such as accuracy score, mean absolute error, mean squared error, root mean square error, root mean square log error, confusion matrix is used to select the model wisely. They are explained through this report. Some visualisations and graphs are also provided in this report for better elaboration of the patterns deduced from the dataset.

Audience

This report aims to provide my understanding of the problem statement, dataset, and regression algorithms being used to the primary audience, i.e., Prof. Moez Ali. My utmost effort is to make this assignment report more informative and easier to understand when compared with the code. Therefore, fellow students and companions are the secondary audiences for this report. Even with basic knowledge of data science and python programming language, the code can be easily understood as each, and every step is elaborated. This report has provided several visualizations to make it easy to understand and informative to my primary and secondary audience.

Table of Contents

List of Figures

1. Dataset Description

The dataset used in this assignment is called as **House Prices dataset and** is retrieved from Kaggle **House Prices- Advanced Regression Techniques** competition. This dataset has **81 features** and **1460 instances.** Along with the dataset, the test dataset was provided separately to test the predictions made by our model. Those predictions were submitted to Kaggle for ranking. A sample submission was also provided through this competition. The most important file after the training data was the “data description” file which gave a detailed overview of the features used in the dataset. The features and their description are listed below: -

- SalePrice - the property's sale price in dollars. This is the target variable that you're trying to predict.
- MSSubClass: The building class
- MSZoning: The general zoning classification
- LotFrontage: Linear feet of street connected to property
- LotArea: Lot size in square feet
- Street: Type of road access
- Alley: Type of alley access
- LotShape: General shape of property
- LandContour: Flatness of the property
- Utilities: Type of utilities available
- LotConfig: Lot configuration
- LandSlope: Slope of property
- Neighborhood: Physical locations within Ames city limits
- Condition1: Proximity to main road or railroad
- Condition2: Proximity to main road or railroad (if a second is present)

- BldgType: Type of dwelling
- HouseStyle: Style of dwelling
- OverallQual: Overall material and finish quality
- OverallCond: Overall condition rating
- YearBuilt: Original construction date
- YearRemodAdd: Remodel date
- RoofStyle: Type of roof
- RoofMatl: Roof material
- Exterior1st: Exterior covering on house
- Exterior2nd: Exterior covering on house (if more than one material)
- MasVnrType: Masonry veneer type
- MasVnrArea: Masonry veneer area in square feet
- ExterQual: Exterior material quality
- ExterCond: Present condition of the material on the exterior
- Foundation: Type of foundation
- BsmtQual: Height of the basement
- BsmtCond: General condition of the basement
- BsmtExposure: Walkout or garden level basement walls
- BsmtFinType1: Quality of basement finished area
- BsmtFinSF1: Type 1 finished square feet
- BsmtFinType2: Quality of second finished area (if present)
- BsmtFinSF2: Type 2 finished square feet
- BsmtUnfSF: Unfinished square feet of basement area
- TotalBsmtSF: Total square feet of basement area

- Heating: Type of heating
- HeatingQC: Heating quality and condition
- CentralAir: Central air conditioning
- Electrical: Electrical system
- 1stFlrSF: First Floor square feet
- 2ndFlrSF: Second floor square feet
- LowQualFinSF: Low quality finished square feet (all floors)
- GrLivArea: Above grade (ground) living area square feet
- BsmtFullBath: Basement full bathrooms
- BsmtHalfBath: Basement half bathrooms
- FullBath: Full bathrooms above grade
- HalfBath: Half baths above grade
- Bedroom: Number of bedrooms above basement level
- Kitchen: Number of kitchens
- KitchenQual: Kitchen quality
- TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)
- Functional: Home functionality rating
- Fireplaces: Number of fireplaces
- FireplaceQu: Fireplace quality
- GarageType: Garage location
- GarageYrBlt: Year garage was built
- GarageFinish: Interior finish of the garage
- GarageCars: Size of garage in car capacity
- GarageArea: Size of garage in square feet

- GarageQual: Garage quality
- GarageCond: Garage condition
- PavedDrive: Paved driveway
- WoodDeckSF: Wood deck area in square feet
- OpenPorchSF: Open porch area in square feet
- EnclosedPorch: Enclosed porch area in square feet
- 3SsnPorch: Three season porch area in square feet
- ScreenPorch: Screen porch area in square feet
- PoolArea: Pool area in square feet
- PoolQC: Pool quality
- Fence: Fence quality
- MiscFeature: Miscellaneous feature not covered in other categories
- MiscVal: Value of miscellaneous feature
- MoSold: Month Sold
- YrSold: Year Sold
- SaleType: Type of sale
- SaleCondition: Condition of sale

2. Data Understanding & Pre-processing

2.1 Importing libraries & Data loading

The very first step is to import the libraries to be used for loading load the dataset. The dataset is then read using the panda’s library. First five rows are displayed to get the brief overview of loaded dataset.

Figure 1: The dataset

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

5 rows × 81 columns

2.2 Getting some basic information about the dataset: -

To understand the shape of the dataset and whether there are some **null values**, and whether the dataset has **duplicate values** or not, this step is priority before any other cleaning techniques are applied. Therefore, some snippets of various steps are displayed below: -

Figure 2: Information on dataset

```
df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
Id	1460.0	730.500000	421.610009	1.0	365.75	730.5	1095.25	1460.0
MSSubClass	1460.0	56.897260	42.300571	20.0	20.00	50.0	70.00	190.0
LotFrontage	1201.0	70.049958	24.284752	21.0	59.00	69.0	80.00	313.0
LotArea	1460.0	10516.828082	9981.264932	1300.0	7553.50	9478.5	11601.50	215245.0
OverallQual	1460.0	6.099315	1.382997	1.0	5.00	6.0	7.00	10.0
OverallCond	1460.0	5.575342	1.112799	1.0	5.00	5.0	6.00	9.0
YearBuilt	1460.0	1971.267808	30.202904	1872.0	1954.00	1973.0	2000.00	2010.0
YearRemodAdd	1460.0	1984.865753	20.645407	1950.0	1967.00	1994.0	2004.00	2010.0
MasVnrArea	1452.0	103.685262	181.066207	0.0	0.00	0.0	166.00	1600.0

Figure 3: duplicate data

```
duplicate = df[df.duplicated()]
print(duplicate)
```

```
Empty DataFrame
Columns: [Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour,
e, Neighborhood, Condition1, Condition2, BldgType, HouseStyle, OverallQual, OverallCond, YearBu
ofMatl, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, ExterCond, Foundation, Bsm
smtFinType1, BsmtFinSF1, BsmtFinType2, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, Heating, HeatingQC,
F, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAb
TotRmsAbvGrd, Functional, Fireplaces, FireplaceQu, GarageType, GarageYrBlt, GarageFinish, Garag
arageCond, PavedDrive, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea
scVal, MoSold, YrSold, SaleType, SaleCondition, SalePrice]
Index: []

[0 rows x 81 columns]
```

2.3 Information on the datatypes of attributes: -

The dataset contains a blend of various data types depending on the attributes. To find out about it, the following piece of code depicts the **data type of each attribute and the memory usage**: -

Figure 4: Information on data types

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
```

2.4 Handling the “NaN” values in the dataset: -

The “NaN” values present in the dataset needed to be dealt with before putting the dataset in the training phase. Therefore, to handle these values the dataset was analysed first comparing it with the dataset description.

- the unknown “NaN” values were present in the categorical columns were dealt by replacing the “None” value which was being detected by Pandas as null to their appropriate value. For example,

  For “Alley” column, if there is no alley access the value in the dataset was given as “None” which was interpreted by Pandas as “Nan”. Therefore, the “None” value was replaced by “no_alley_access” for better understanding”.

- Apart from the categorical columns, some of the numeric columns were having missing values. These missing values were imputed using respective techniques such as mean or median.

  The following piece of code displays the handling on “NaN” value in just one cell: -

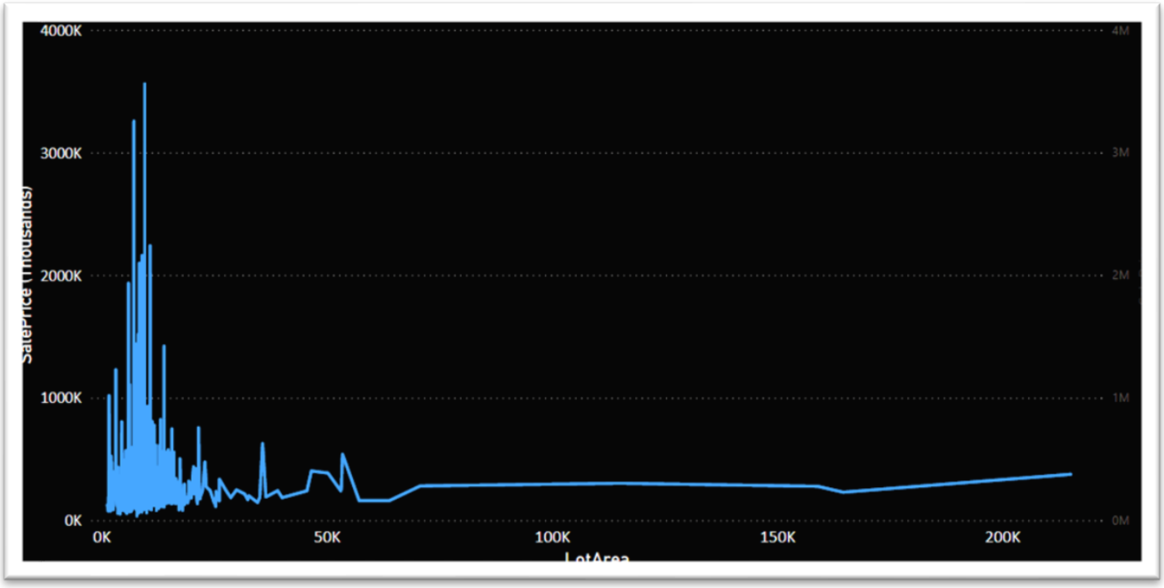
Figure 5: Handling NaN values

```
df=df.fillna({'LotFrontage':df['LotFrontage'].median()})
df=df.fillna({'Alley':'No_alley_access'})
df=df.fillna({'BsmtQual':'No_basement'})
df=df.fillna({'BsmtCond':'No_basement'})
df=df.fillna({'BsmtExposure':'No_basement'})
df=df.fillna({'BsmtFinType1':'No_basement'})
df=df.fillna({'BsmtFinType2':'No_basement'})
df=df.fillna({'FireplaceQu':'No_fireplace'})
df=df.fillna({'GarageType':'No_garage'})
df=df.fillna({'GarageFinish':'No_garage'})
df=df.fillna({'GarageQual':'No_garage'})
df=df.fillna({'GarageCond':'No_garage'})
df=df.fillna({'PoolQC':'No_pool'})
df=df.fillna({'Fence':'No_fence'})
df=df.fillna({'MiscFeature':'None'})
df=df.fillna({'MasVnrType':'None'})
df=df.fillna({'Electrical':'SBrkr'})
df=df.fillna({'MasVnrArea':df['MasVnrArea'].median()})
df=df.fillna({'GarageYrBlt':2005.0})
```

3.Exploratory Data Analysis

3.1 Relationship between SalePrice & LotArea

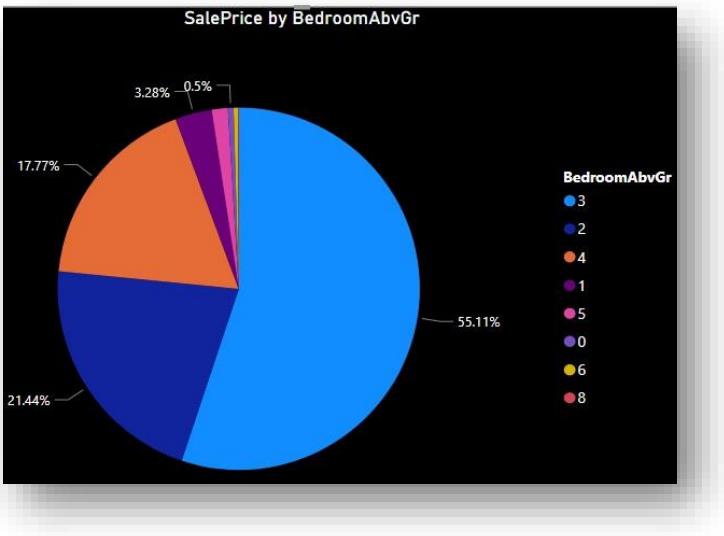
Figure 6: SalePrice Vs. LotArea



We can observe with the above line plot that SalePrice for most of the houses is maximum where the LotArea is less than 50K. Still, there are some houses which have LotArea greater than 50K but with the lower selling price.

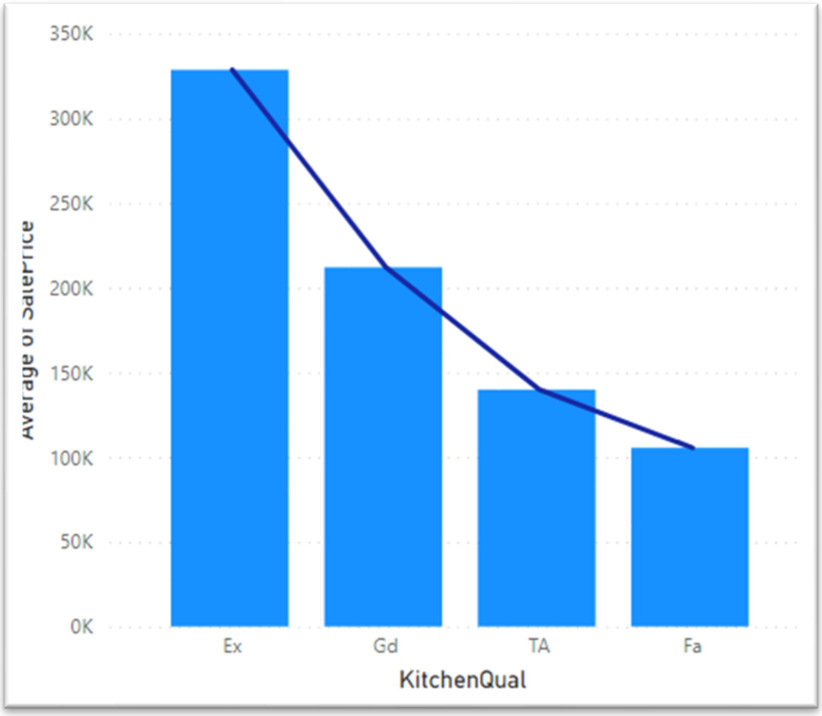
3.2 Relationship between SalePrice by No. of bedrooms

Figure 7: SalePrice Vs. No. of Bedrooms



3.3 Relationship between Sale Price and Kitchen quality

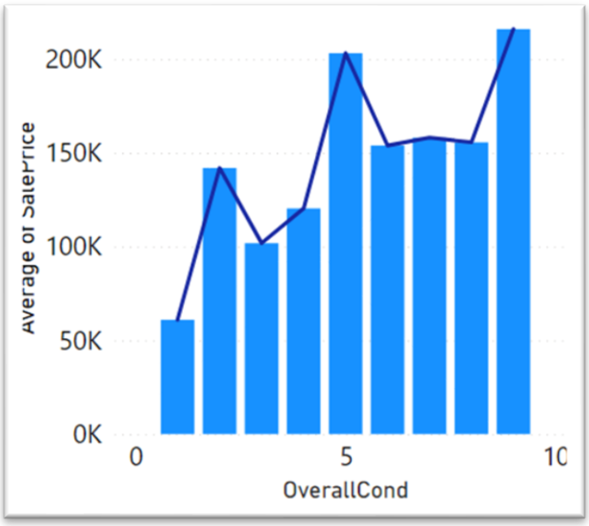
Figure 8: SalePrice Vs. Kitchen Quality



It can be observed from the above graph, the house having excellent kitchen quality have the best-selling price followed by other categories. Therefore, quality of kitchen determines the selling price of a house.

3.4 Relationship between Sale price and overall condition of the house

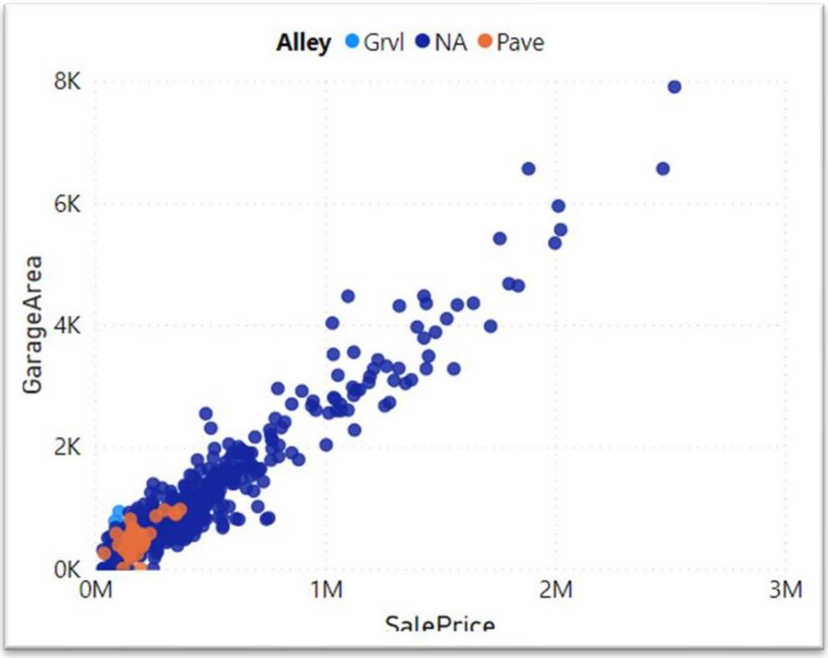
Figure 9: SalePrice Vs. Overall Condition



The best-selling price is determined by houses having overall condition as 5 and 10.

3.5 Relationship between sale price and garage area w.r.t Alley path too.

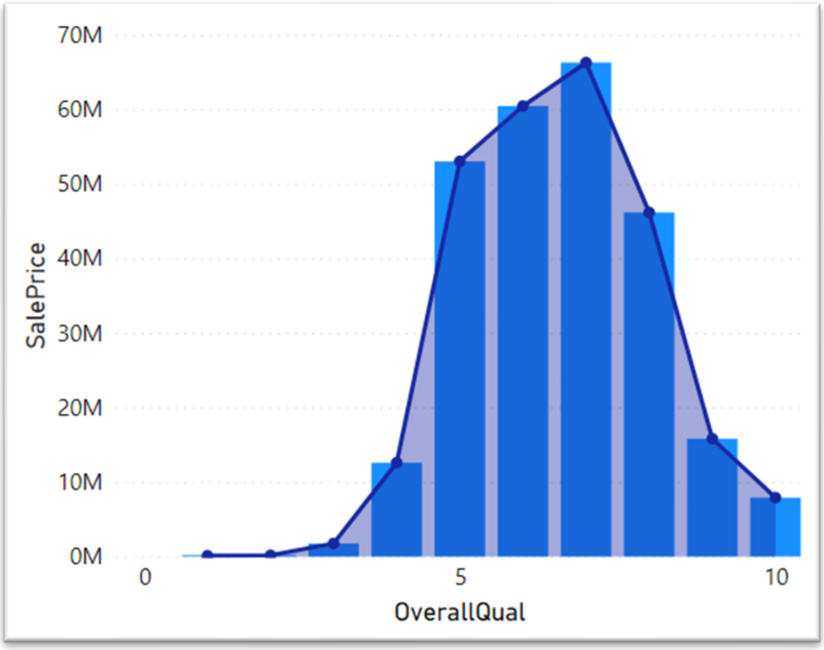
Figure 10: SalePrice Vs. Garage Area



The above Scatter plot depicts that most of the houses having good selling price and more garage area do not have alley access. The scatter plot also gives the hint as the values more than 2M of selling price can be potential outliers.

3.6 Relationship between sale price and overall quality of the house

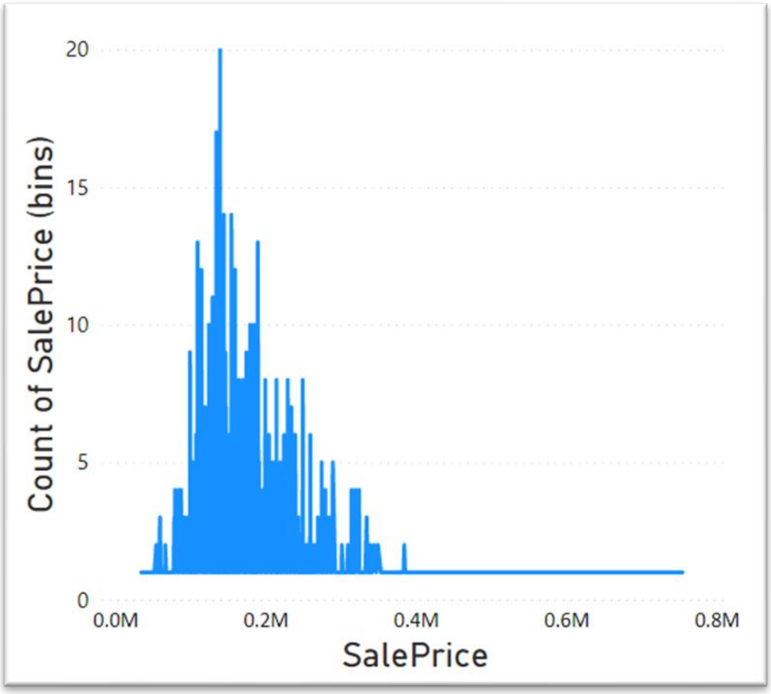
Figure 11: SalePrice Vs. Overall Quality



The above clustered column chart depicts that houses having their overall quality rates in the range of 5 to 8 have the best-selling prices.

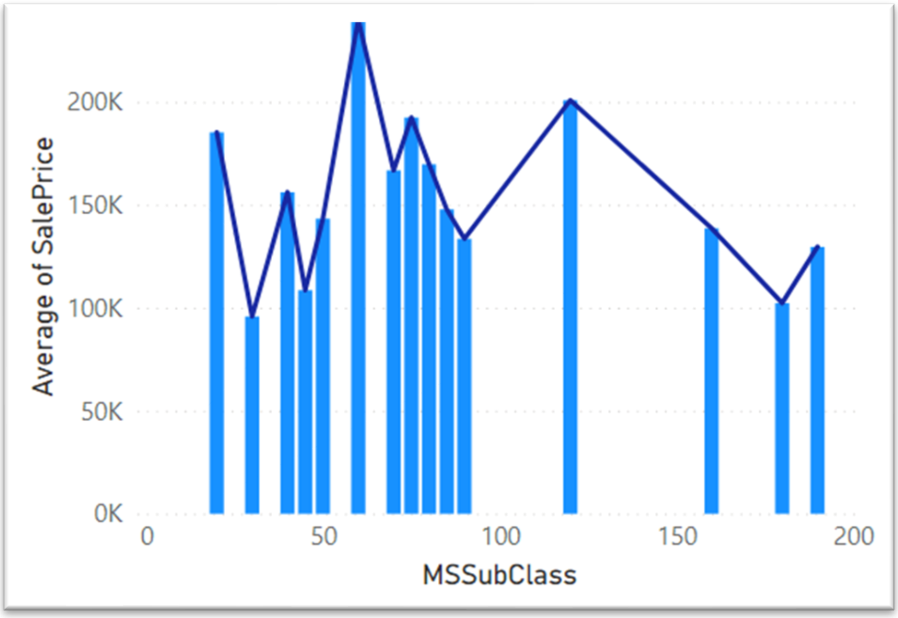
3.7 Sale price distribution

Figure 12: SalePrice Distribution



3.8 Relationship between Sale price and MSSubclass

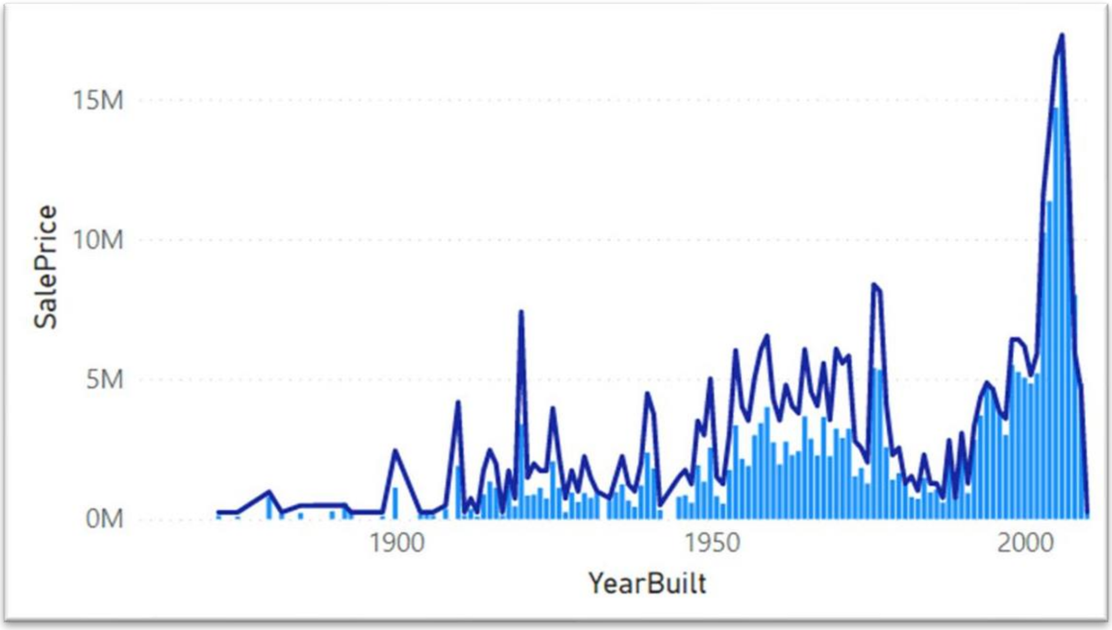
Figure 13: SalePrice Vs. MSSubClass



It can be observed that the houses having MSSubClass around 60-80 meaning they are having 2-story and multi-level split have the best-selling prices as compared to other classes.

3.9 Relationship between Sale Price and Year built

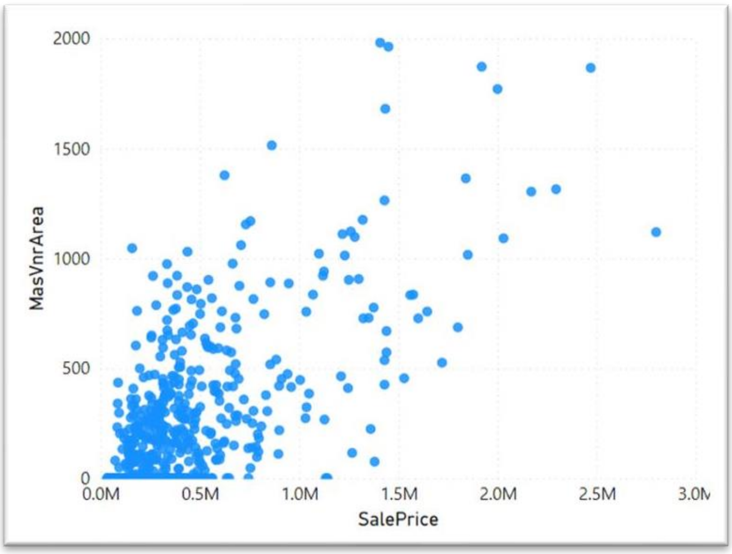
Figure 14: SalePrice Vs. Year built



Some of the old houses still have a better selling price than 90's but as the graph depicts that houses in early 2000's has the maximum sale price.

3.10 Relationship between Sale price and MasVnrArea.

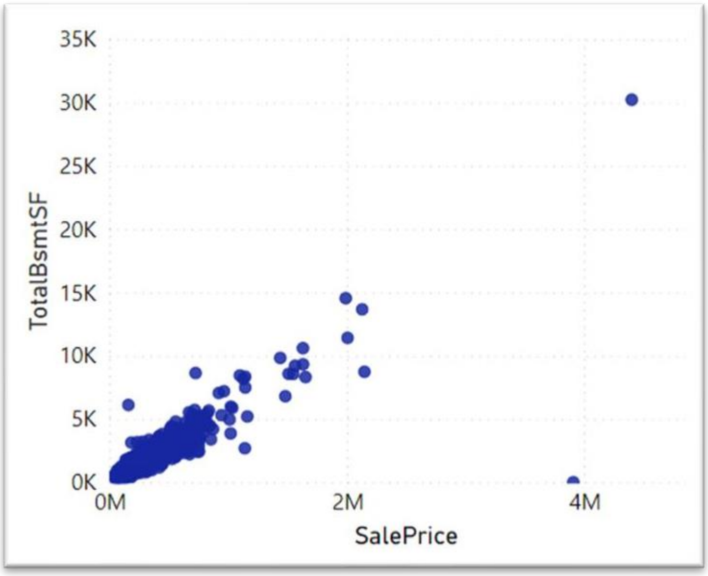
Figure 15: Sale price Vs. Masonry Veneer area



The cluster plot shows that as the masonry veneer area increase the Sale price also increases. It also hints for some potential outliers present in this column who have the selling price more than it should be.

3.11 Relationship between Sale price and Total Basement area

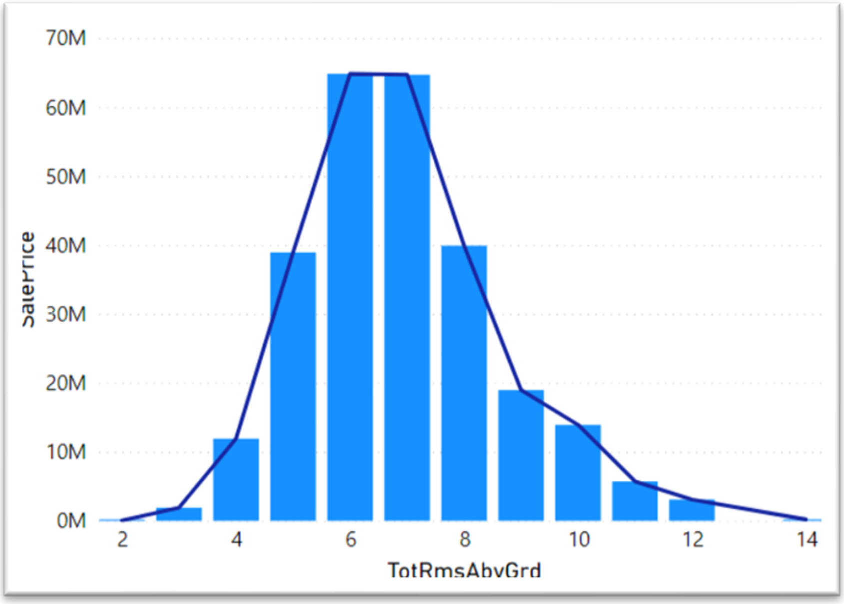
Figure 16: Sale price Vs. Total basement area



This scatter plot between the square feet area of basement and sale price gives a clear trend indication that increase in basement area increases the Sale price of house.

3.12 Relationship between Sale Price and total Rooms in house.

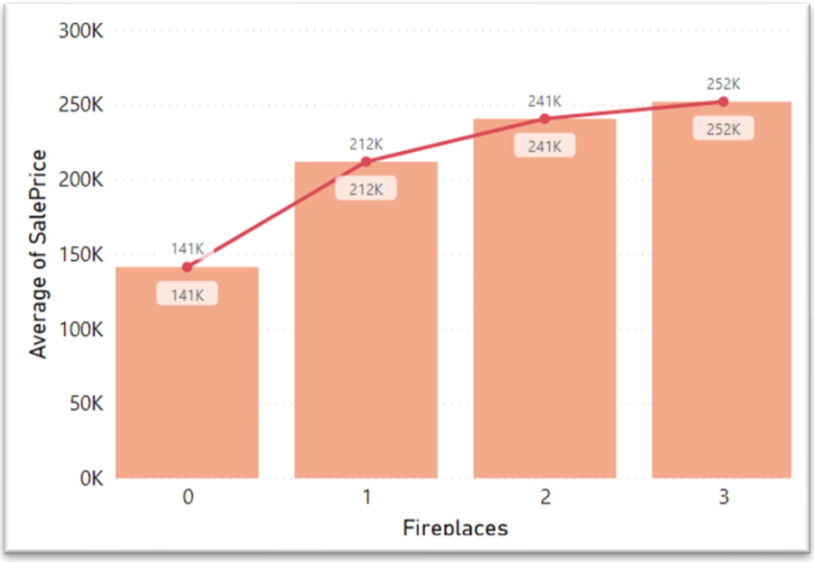
Figure 17: Sale price Vs. Total Rooms



The column and line chart depicts that sale price is maximum for houses having total rooms in the range of 6-7.

3.13 Relationship between Sale Price and Fireplaces in the house.

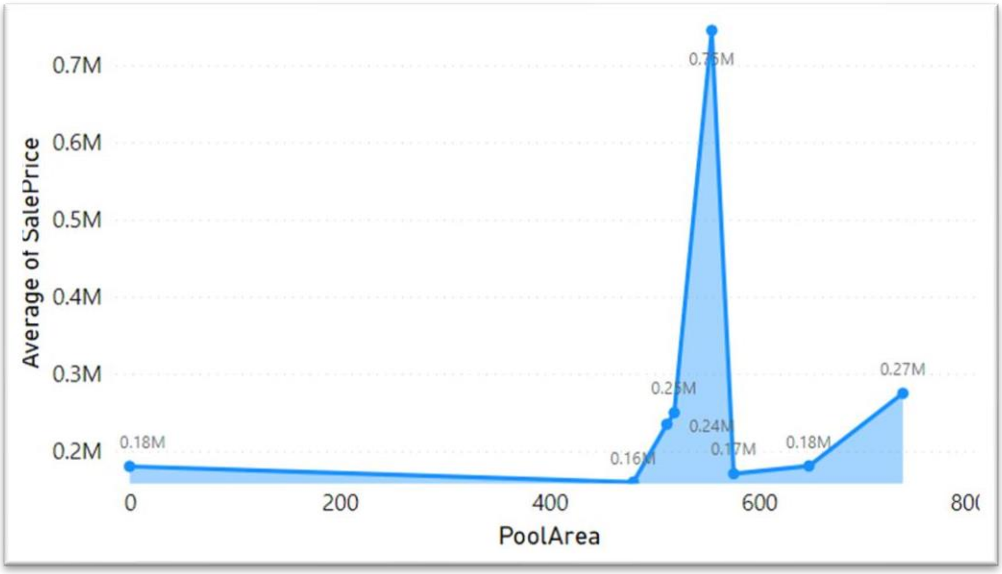
Figure 18: Sale price Vs. Fireplaces



Fireplaces are important for heating purposes specially in old houses. It can be observed from the above pattern that more the no. of fireplace in the house, more is the selling price.

3.14 Relationship between Sale price and Pool area in the house

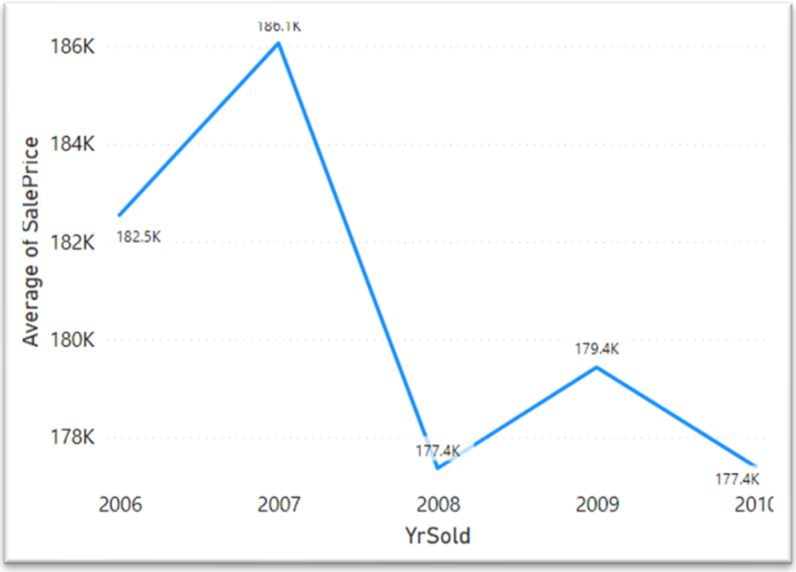
Figure 19: Sale price Vs. Pool Area



Most of the houses have average selling price less than 0.3 M irrespective of the pool area. Only in some cases the prices increase with increase in pool area that may depend on other factors too.

3.15 Relationship between Sale price and year sold

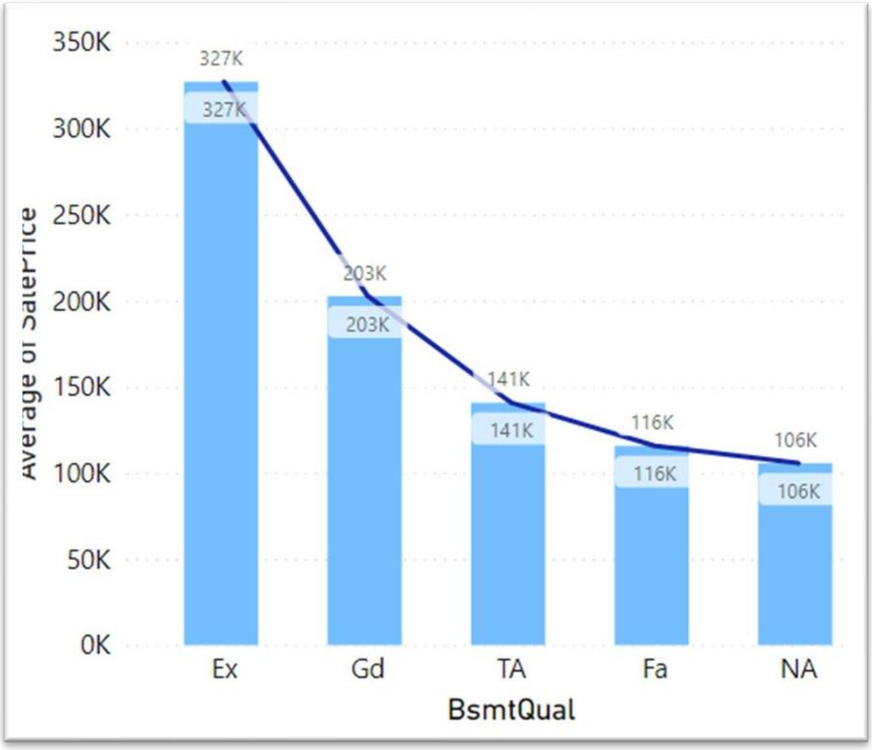
Figure 20: SalePrice Vs. Year Sold



The selling price was lowest in the year 2008. These houses now can be sold at greater price with some renovations.

3.16 Relationship between Sale Price and Basement quality

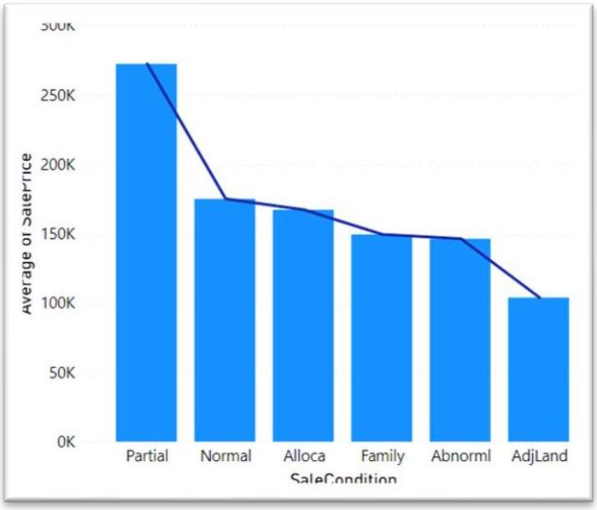
Figure 21: Sale price Vs. Basement quality



It is clear from the above trend that as the basement quality drops from excellent to fair, the selling price of houses also decreases.

3.17 Relationships between Sale price and selling condition of the house

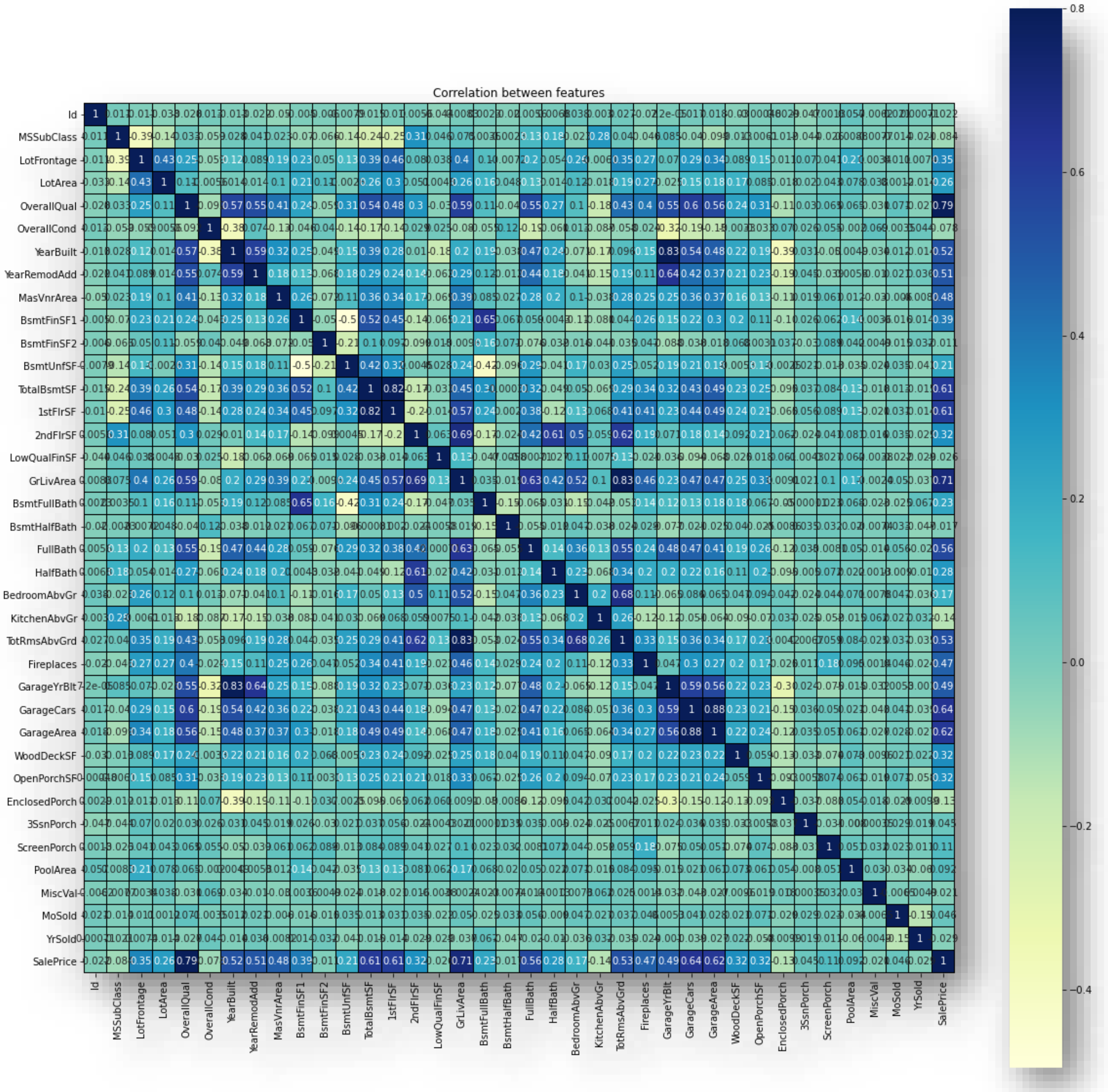
Figure 22: Sale Price Vs. Sale condition



The houses sold in Partial condition have the maximum sale price as compared to others.

3.18 Co-relation plot

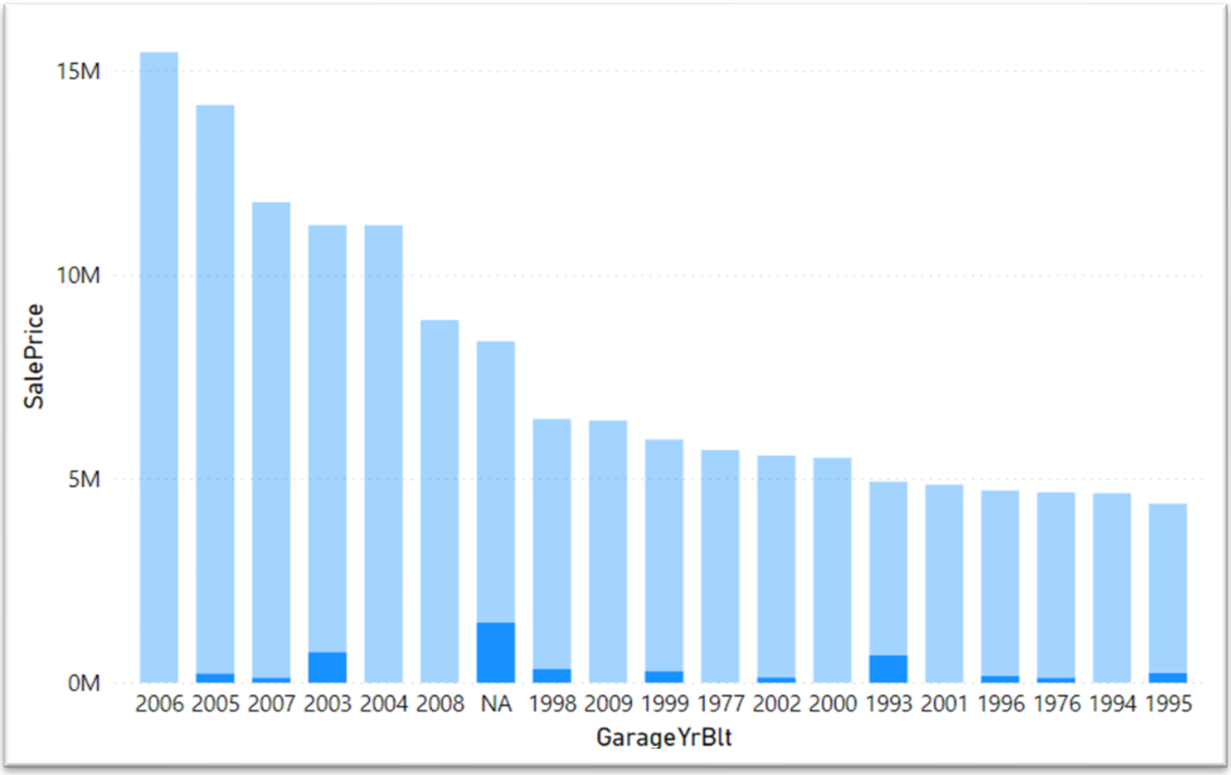
Figure 23: Co-relation plot



The corelation plot was necessary to filter out some columns that are most positively corelated to Sale price column.

3.19 Relationship between Sale price and Garage Year built

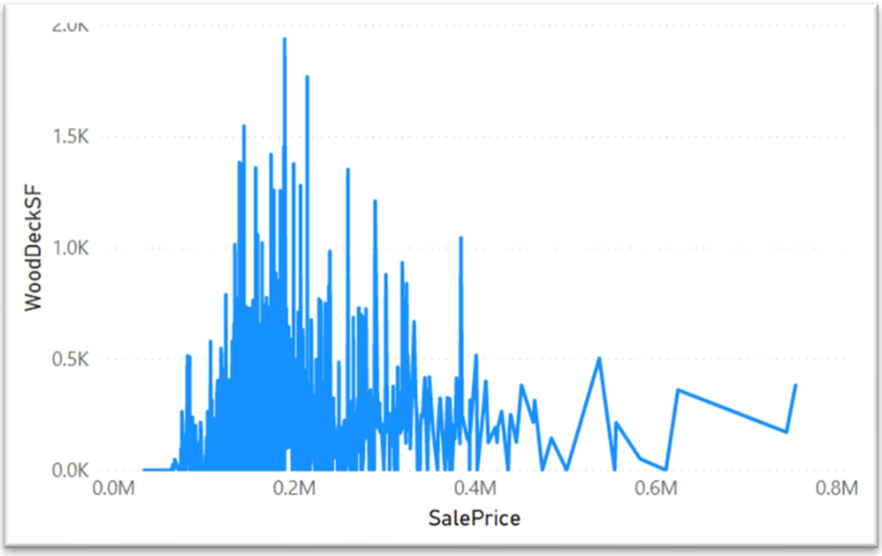
Figure 24: Sale Price Vs. Garage Year build



The sale price is maximum for the latest garages after the 90's.

3.20 Relationship between Sale price and Wood deck surface area

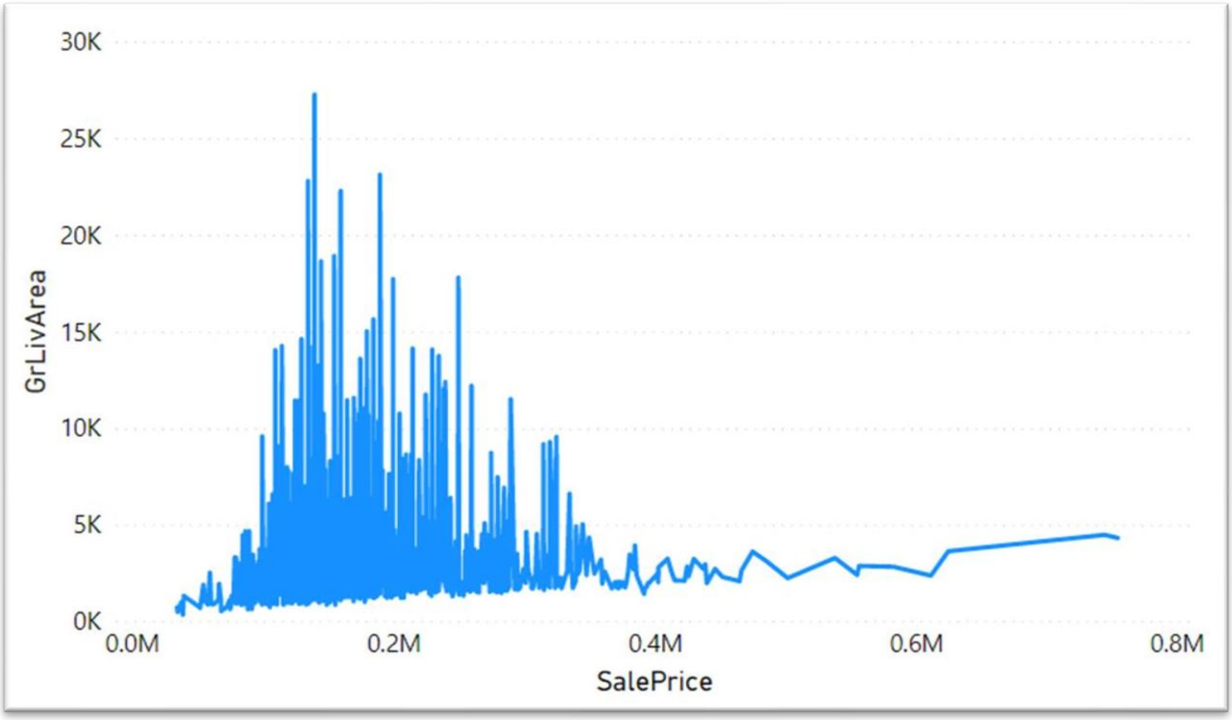
Figure 25: SalePrice Vs. Wood deck Area



It can be observed from the above line plot that as the Wood deck surface area in a house increases the sale prices also goes up.

3.21 Relationship between Sale price and greater living area

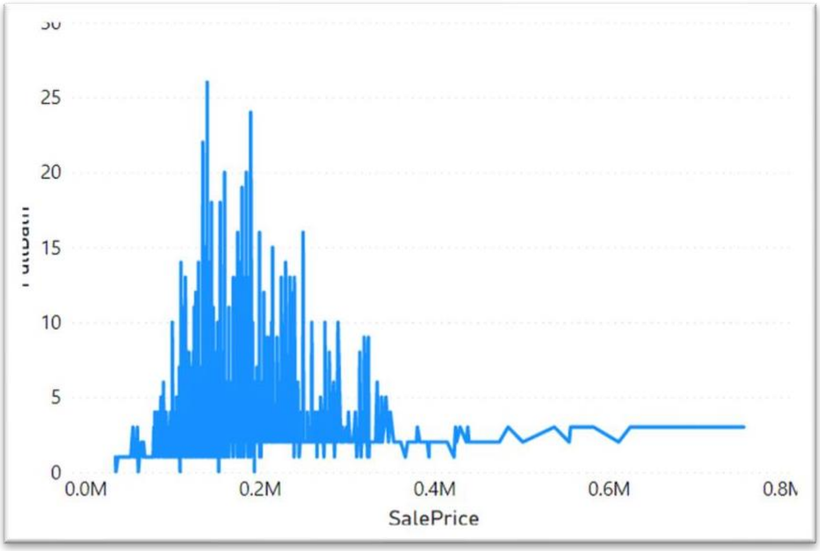
Figure 26: SalePrice Vs. Living Area



Greater the living area in a house, result in more spacious living area and increasing the cost of selling price.

3.22 Relationship between Sale price and full bathrooms above grade

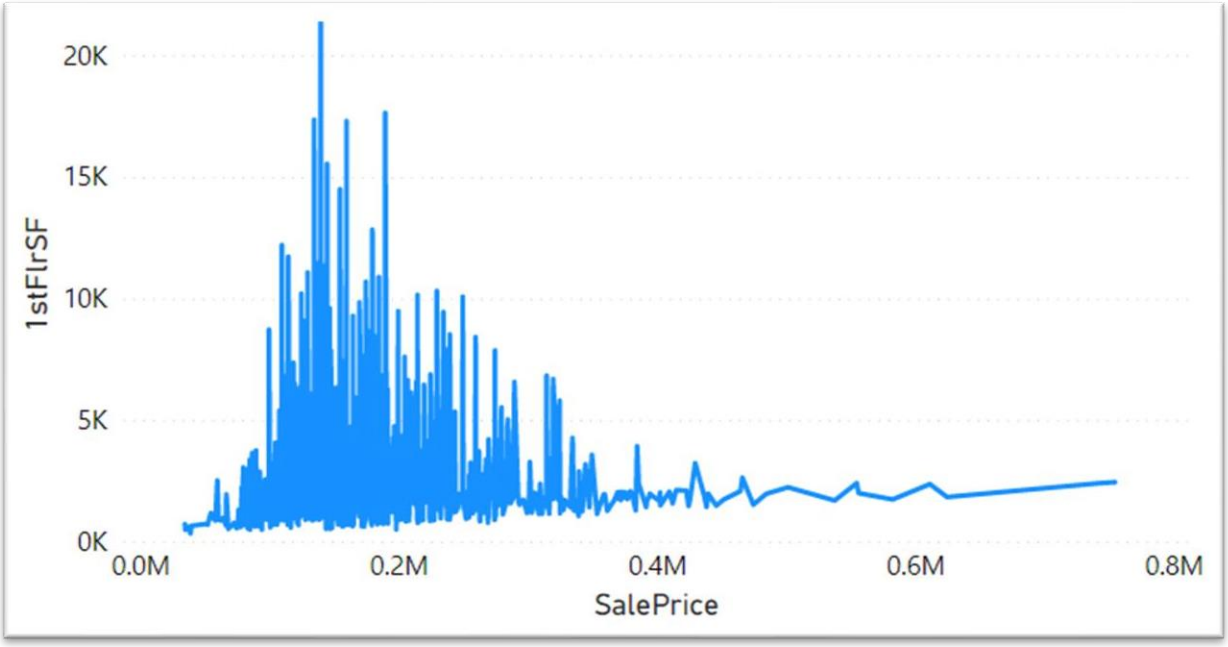
Figure 27: Sale price Vs. Full bathroom



Most of the houses are having full bathrooms in the range of 5-10 have their selling price in a moderate range less than 0.4M.

3.23 Relationship between Sale price and First floor square feet area

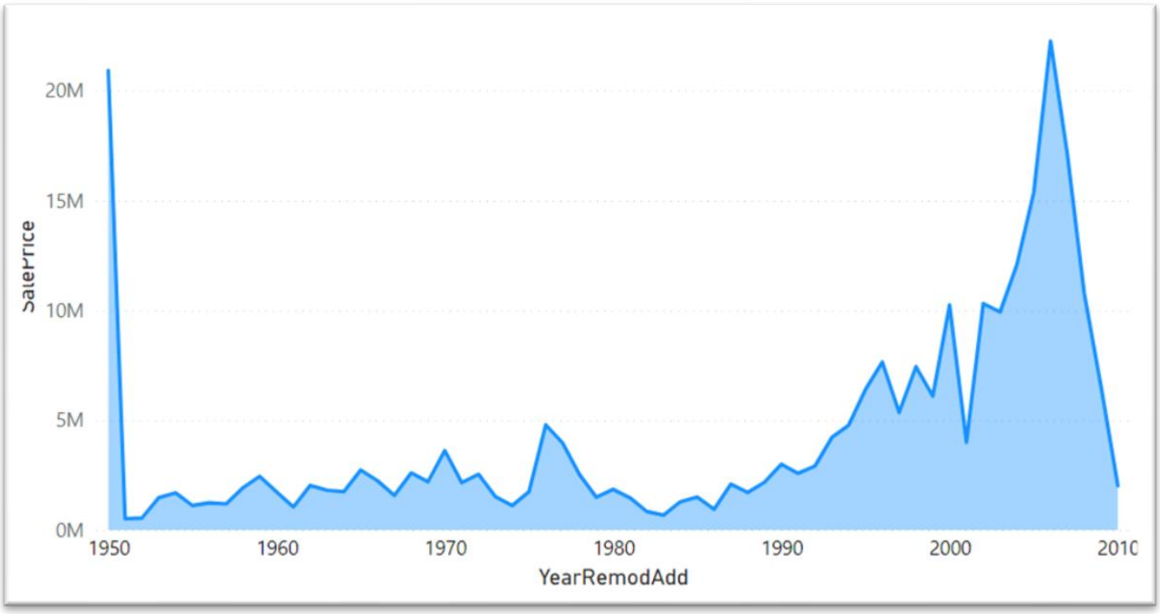
Figure 28: Sale price Vs. First floor area



Most of the houses are having first floor square foot area bathrooms in the range of 2-10 have their selling price in a moderate range less than 0.3M.

3.24 Relationship between Sale price and Year remodelled

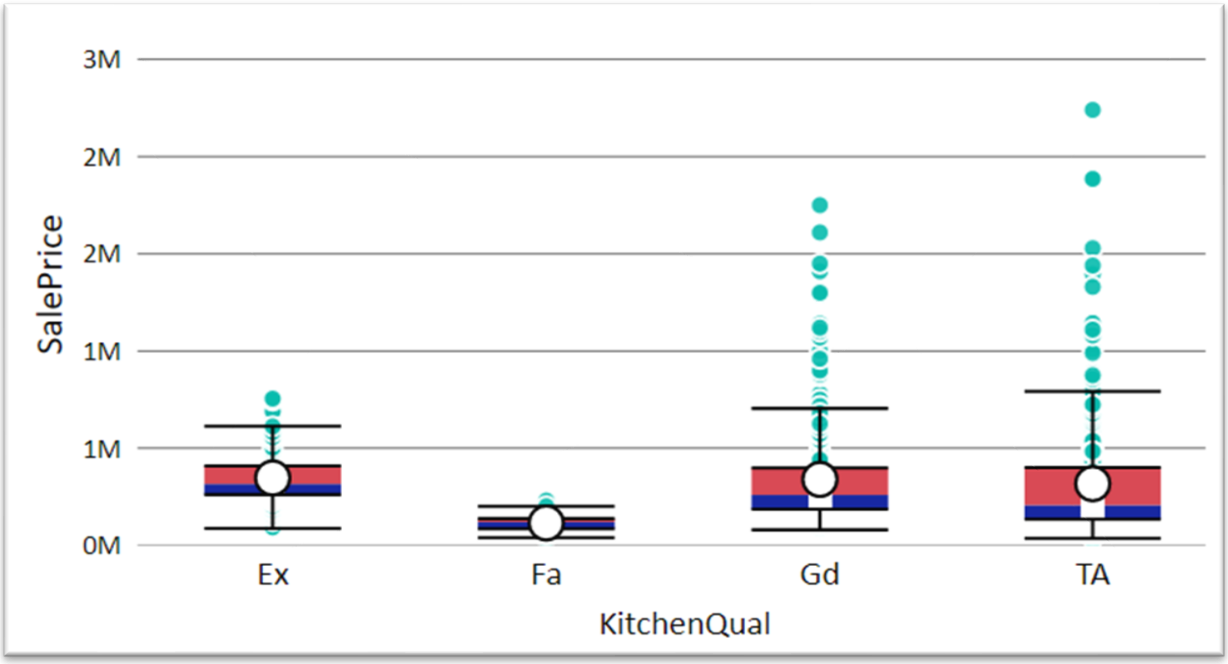
Figure 29: Sale price Vs. Remodelled Year



The houses which were built in 1950's and the latest remodelled houses have the maximum selling price as indicated by the graph above.

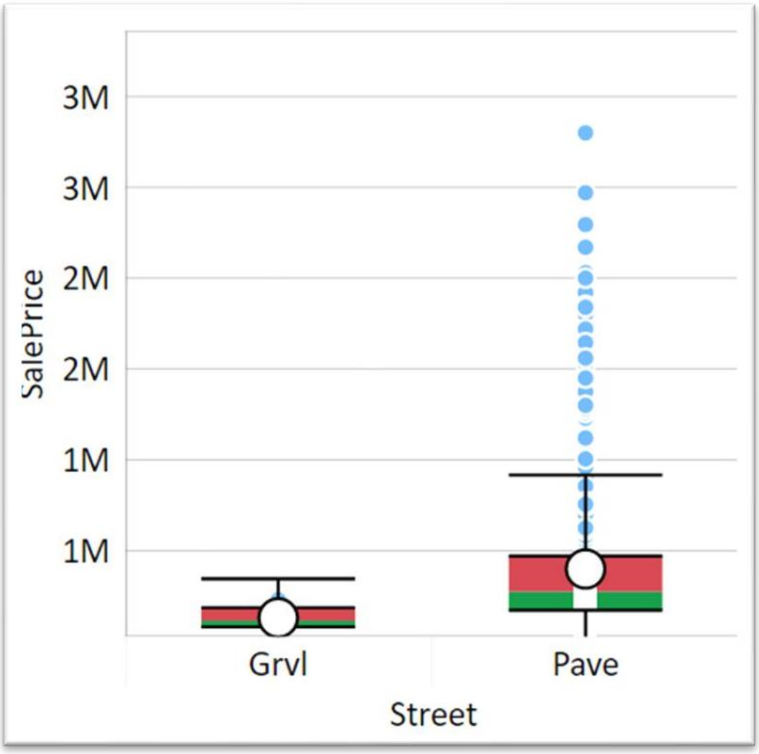
3.25 Detecting outliers in Kitchen Quality

Figure 30: Sale price Vs. Kitchen quality



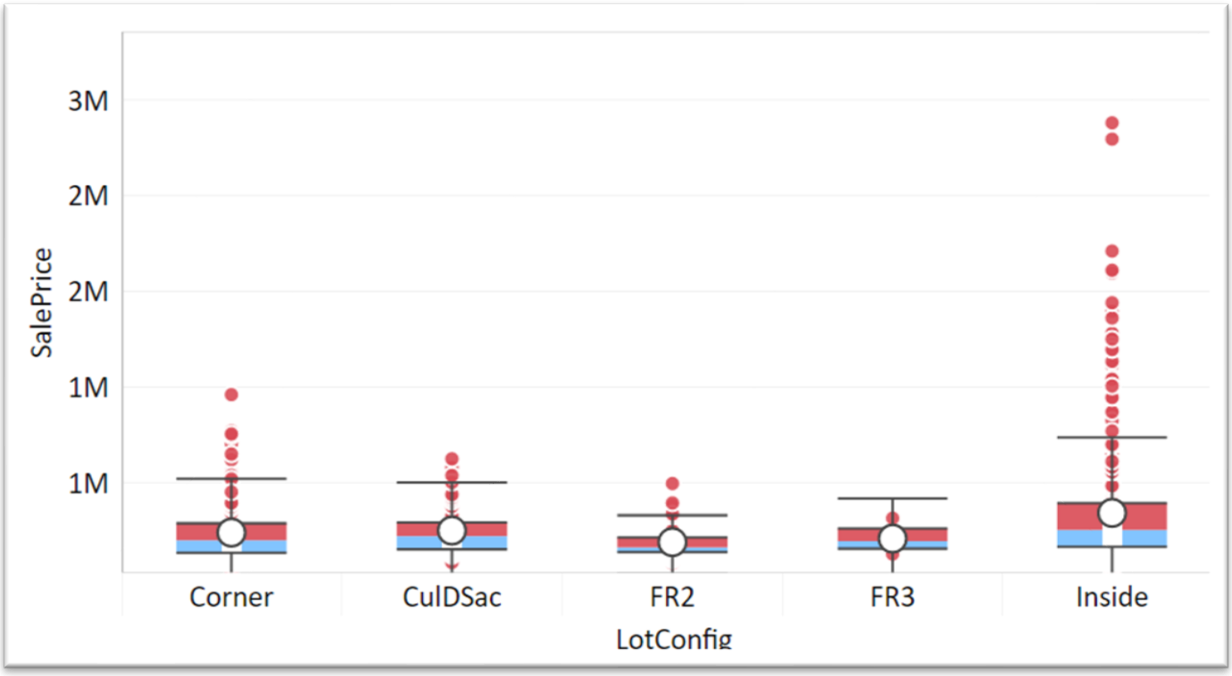
3.26 Detecting Outliers in Street column

Figure 31: Sale price Vs. Street type



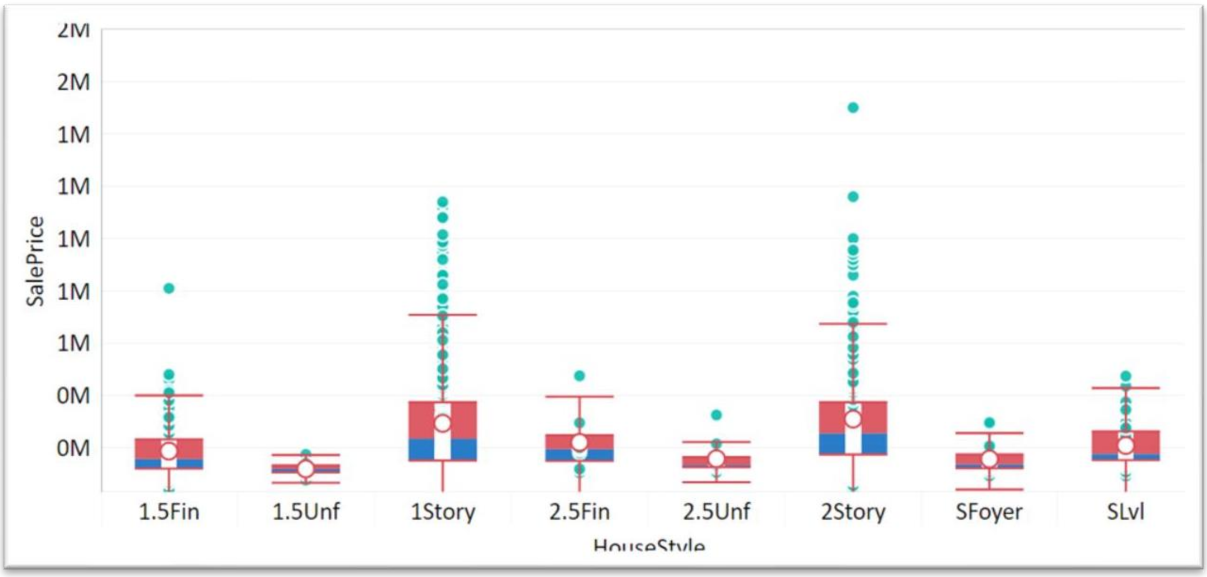
3.27 Detecting outliers in Lot configuration

Figure 32: Sale price Vs. Configuration of Lot



3.28 Detecting outliers in House style column

Figure 33: Sale price Vs. House style



4. Model Training & Evaluation

4.1 Catboost Regressor: -

CatBoost builds upon the theory of decision trees and gradient boosting. The main idea of boosting is to sequentially combine many weak models (a model performing slightly better than random chance) and thus through greedy search create a strong competitive predictive model. Because gradient boosting fits the decision trees sequentially, the fitted trees will learn from the mistakes of former trees and hence reduce the errors. This process of adding a new function to existing ones is continued until the selected loss function is no longer minimized.

In the growing procedure of the decision trees, CatBoost does not follow similar gradient boosting models. Instead, CatBoost grows oblivious trees, which means that the trees are grown by imposing the rule that all nodes at the same level, test the same predictor with the same condition, and hence an index of a leaf can be calculated with bitwise operations. The oblivious tree procedure allows for a simple fitting scheme and efficiency on CPUs, while the tree structure operates as a regularization to find an optimal solution and avoid overfitting.

4.2 Evaluation metrics: -

After training the model, specific evaluation metrics were selected to check the model’s accuracy: -

- **Mean Absolute Error (MAE):** - mean absolute error is a measure of errors between paired observations expressing the same phenomenon

Figure 34: Mean Absolute Error

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error

y_i = prediction

x_i = true value

n = total number of data points

- **Mean Squared Error (MSE): -** mean squared error (MSE) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value.

Figure 35: Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

- **Root Mean Squared Error (RMSE): -** Root-mean-square error is a frequently used measure of the differences between values (sample or population values) predicted by a model and the values observed.

Figure 36: Root Mean Squared Error

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

RMSD = root-mean-square deviation
i = variable i
N = number of non-missing data points
x_i = actual observations time series
$\hat{x}_i$ = estimated time series

- **Root Mean Squared Log Error (RMSLE)**: - RMSLE can be defined using a slight modification on sklearn's mean_squared_log_error function, which itself a modification on the familiar Mean Squared Error (MSE) metric.

Figure 37: Root Mean Squared Log Error

RMSLE = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$

Where:

n is the total number of observations in the (public/private) data set,

p_i is your prediction of target, and

a_i is the actual target for i.

$log(x)$ is the natural logarithm of x ($log_e(x)$).

All the metrics used in the code is depicted in the following figure: -

Figure 38: Evaluation Metrics

```
Mean Absolute Error:      4015.111878978296
Mean Squared Error:       27012165.967486776
Root Mean Squared Error: 5197.322961630033
R Squared:                0.9955661182684862
Root Mean Squared log Error: 0.0013115235797421828
```

As the score metrics used by Kaggle for ranking was RMSLE, therefore utmost effort was put to improve the RMSLE score using Catboost regression algorithm.

4.3 Hyper- parameter tuning using Grid Search CV: -

GridSearchCV is a function that comes in Scikit-learn's model_selection package. This function helps to loop through predefined hyperparameters and fit your any (model) on the training set. So, in the end, we can select the best parameters from the listed hyperparameters. The following figure presents the usage of Grid Search CV for finding the best parameters for cat boost regression: -

Figure 39: Grid Search CV

```python
from sklearn.model_selection import GridSearchCV
parameters = {'depth': [6,8,10],
                  'learning_rate' : [0.01, 0.05, 0.1],
                  'iterations'    : [30, 50, 100]
                 }
grid = GridSearchCV(estimator=cb,param_grid = parameters, cv = 2, n_jobs=-1)
grid.fit(X_train, y_train)
# Results from Grid Search
print("\n========================================================")
print(" Results from Grid Search " )
print("========================================================")

print("\n The best estimator across ALL searched params:\n",
          grid.best_estimator_)

print("\n The best score across ALL searched params:\n",
          grid.best_score_)
print("\n The best parameters across ALL searched params:\n",
          grid.best_params_)
print("\n ========================================================")
```

The parameters obtained from the above code were then used as new input parameters for the catboost resulting in improved RMSLE square: -

Figure 40: Ideal Parameters

```
The best score across ALL searched params:
0.8488338671200568

The best parameters across ALL searched params:
{'depth': 6, 'iterations': 100, 'learning_rate': 0.1}
```

5. Conclusion

The House price dataset used in this assignment was prepared, cleansed, analyzed, and visualized to get some actionable insights from the data. It was trained on various different algorithms among which the best model i.e. catboost regressor was chosen, and it's parameters were tuned to get better accurate results. Therefore, after thorough analysis, and modelling predcitions were made on the test dataset.

The predictions made on the test dataset were then saved and submitted as submission to Kaggle competition.

Figure 41: Final Submission

```
sample['SalePrice'] = y_pred
sample.head()
```

	Id	SalePrice
0	1461	127565.438073
1	1462	161058.561940
2	1463	188562.152255
3	1464	192091.478858
4	1465	180725.748646

```
sample.to_csv("Kaggle_score8_catboost.csv",index=False)
```

6. References

- https://towardsdatascience.com/catboost-regression-in-6-minutes-3487f3e5b329
- https://www.kaggle.com/c/house-prices-advanced-regression-techniques
- https://www.linkedin.com/learning/power-bi-dashboards-for-beginners/getting-started-with-power-bi?u=56968457