Housing Price Prediction

Submitted by:

Prerna Jain

Acknowledgement

I would like to express my gratitude to my guide Srishti maan (SME, Flip Robo) for his constant guidance, continuous encouragement, and unconditional help towards the development of the project. It was he who helped me whenever I got stuck somewhere in between. The project would have not been completed without the support and confidence heshowed towards me.

Lastly, I would l like to thank all those who helped me directly or indirectly toward the successful completion of the project.

Introduction

Business Problem Framing

Houses are one of the necessary needs of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors to 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, improve their marketing strategies and focus 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.

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. Investment is a business activity that most people are interested in this globalization era. The result of this census indicates that the younger generation will need a house or buy a house in the future. Based on preliminary research conducted, there are two standards of house price which are valid in buying and selling transaction of a house that is house price based on the developer. There are several approaches that can be used to determine the price of the house, one of them is the prediction analysis.

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.

Conceptual Background of the Domain Problem

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 prices.

Review of Literature

From the dataset, I get to know that it is a Regression problem and the Sale Price of a house varieson its properties. And there are so many features which help to find it.

Motivation for the Problem Undertaken

I am doing this for practice, to get more hands-on data exploration, Feature extraction and Model building.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

I have used Log transformation for transforming the continuous numerical variable containing non-zero elements only as during analysis I found that these variables were not normally distributed, so transformed them using log normal transformation so that the features will be close to normally distributed. I have done some testing separately to check the importance of categorical variables with respect to the Sale Price of the House. Use of Mean, Median to replace the Missing Values in features. Use of Correlation matrix to check the importance and correlation of numerical variables with respect to target variable Sale price and Feature scaling using Min Max scaler as we have positive data points.

Data Sources and their formats

Data I get from the Flip Robo the format was in CSV (Comma Separated Values).
The data source is Kaggle. There is a total of 1168 observations and 81 features including the target feature Sale Price in the train dataset, and 292 observations and 80 features including target feature Sale Price in the test dataset. For the train dataset, there are 3 features having float data types, 35 features having integer data type and 43 features having object data type.
For the test dataset, all is the same except the target feature (Sale Price).

Data Pre-processing Done

I have handled the missing values in both train and test data set. Based on the Data description I have imputed the missing data. Most of the features have nan values which were described as an absence of a feature in data description, so I have replaced them with 'not available for each feature having nan value. For LotFrontage I have imputed the missing values by the median value of LotFrontage grouped by LotConfig. Removal of ID column as it was not important for our model and also dropped Utilities Feature as all the values in this feature were same so there was no variance. Dropped GarageYrBlt I was taking the difference of GarageYrBlt and Year Sold to check the age of the Garage however I found out that where ever there was nan value in GarageYrBlt it meant that the property has no Garage so, if it was subtracted from Year sold then it would have some unrealistic value which should not be done.

Data Inputs- Logic- Output Relationships

I have found out that with continuous numerical variables there is a linear relationship with the Sale Price. And for categorical variables, I have used Boxplot for each categorical feature that shows the relation with the median sale price for all the sub categories in each categorical

variable. For continuous numerical variables, I have used a scatter plot to show the relationshipbetween a continuous numerical variable and a target variable.

Hardware and Software Requirements and Tools Used

The system requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view. System requirements are all of the requirements at the system level that describe the functions which the system as a whole should fulfil to satisfy the stakeholder needs and requirements, and is expressed in an appropriate combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

Hardware requirements: -

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software requirements: -

Anaconda

Libraries: -

From sklearn.preprocessing import StandardScaler

As these columns are different in **scale**, they are **standardized** to have a common **scale** while building machine learning model. This is useful when you want to compare data that correspond to different units.

from sklearn.preprocessing import Label Encoder

Label Encoder and One Hot Encoder. These two encoders are parts of the SciKit Learn library in Python, and they are used to convert categorical data, or text data, into numbers, which our predictive models can better understand.

from sklearn.model_selection import train_test_split,cross_val_score

Train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train_test_split will make random partitions for the two subsets.

The algorithm is trained and tested K times, each time a new set is used as testing set while remaining sets are used for training. Finally, the result of the K-Fold Cross-Validation is the average of the results obtained on each set.

from sklearn.neighbors import KNeighborsRegressor

K Nearest Regressor (KNN) is a very simple, easy to understand, versatile and one of the topmost machine learning algorithms. KNN used in the variety of applications such as finance, healthcare, political science, handwriting detection, image recognition and video recognition

from sklearn.linear_model import LinearRegression

The library sklearn can be used to perform linear regression in a few lines as shown using the LinearRegression class. It also supports multiple features. It requires the input values to be in a specific format hence they have been reshaped before training using the fit method.

from sklearn.tree import DecisionTreeRegressor

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

For feature transformation, I have used Log normal transformation to make the continuous non zero variables close to normal distributed. Use of Annona test to check the importance of categorical features. Use of Pearson's correlation coefficient to check the correlation between dependent and independent features. Use of Min-Max scaler to scale down the features and one label encoding to encode categorical features in numeric.

Testing of Identified Approaches (Algorithms)

Listing down all the algorithms used for the training and testing.

- KNN = KNeighborsRegressor ()
- LR = LinearRegression ()
- SVR = SVR ()
- DT = DecisionTreeRegressor ()
- RF = RandomForestRegressor ()

I applied all these algorithms to the dataset.

Run and Evaluate selected models

```
********************************************************************************
accuracy score of -> LinearRegression()
R2 Score:                 0.9113837734325645
Mean Absolute Error:      0.47784883608422013
Mean Squared error:       0.42284386724508316
Root Mean Squared Error:  0.6502644594663645
[0.90727752 0.89210579 0.82948024 0.84882482 0.87421068 0.88186246
 0.85334359 0.85028035]
cross validation score:   0.867173182946023
Difference between R2 score and cross validatio score is -  0.04421059048654152
********************************************************************************
********************************************************************************
accuracy score of -> RandomForestRegressor()
R2 Score:                 0.8862569560623512
Mean Absolute Error:      0.5344597755047049
Mean Squared error:       0.5427397490709321
Root Mean Squared Error:  0.7367087274295943
[0.880604   0.8800722  0.84128969 0.83891715 0.82659683 0.86085467
 0.8441217  0.86136286]
cross validation score:   0.854227387154818
Difference between R2 score and cross validatio score is -  0.03202956890753317
********************************************************************************
********************************************************************************
accuracy score of -> DecisionTreeRegressor()
R2 Score:                 0.7569946457032288
Mean Absolute Error:      0.777061555170274
Mean Squared error:       1.1595316992415008
Root Mean Squared Error:  1.0768155363113503
[0.76629374 0.77954493 0.63544928 0.68570269 0.7613614  0.73077597
 0.68704723 0.76114596]
cross validation score:   0.7259151505676221
Difference between R2 score and cross validatio score is -  0.031079495135606705
********************************************************************************
********************************************************************************

********************************************************************************
accuracy score of -> SVR()
R2 Score:                 0.8146326124882534
Mean Absolute Error:      0.5978720827091031
Mean Squared error:       0.8845046334368329
Root Mean Squared Error:  0.9404810649007416
[0.79300881 0.84891188 0.86470883 0.83630515 0.77154394 0.8129203
 0.84748976 0.78670912]
cross validation score:   0.8201997231479142
Difference between R2 score and cross validatio score is -  -0.005567110659660868
********************************************************************************
```

```
********************************************************************************
accuracy score of -> KNeighborsRegressor()
R2 Score:                  0.8345887836390523
Mean Absolute Error:       0.6463375374064052
Mean Squared error:        0.7892811635186331
Root Mean Squared Error:   0.8884149725880542
[0.82616525 0.8402331  0.79970383 0.7980662  0.79066369 0.81590406
 0.82988288 0.78027468]
cross validation score:    0.8101117125261218
Difference between R2 score and cross validatio score is -  0.024477071112930515
********************************************************************************
********************************************************************************
accuracy score of -> GradientBoostingRegressor()
R2 Score:                  0.8933426982886384
Mean Absolute Error:       0.5244104678123298
Mean Squared error:        0.50892920712707
Root Mean Squared Error:   0.7133927439545976
[0.90083111 0.87988951 0.82988453 0.84172053 0.84025601 0.86758695
 0.84297851 0.86017494]
cross validation score:    0.8579152616752614
Difference between R2 score and cross validatio score is -  0.035427436613376995
********************************************************************************
********************************************************************************
accuracy score of -> Ridge()
R2 Score:                  0.9113712736887932
Mean Absolute Error:       0.47784548174348024
Mean Squared error:        0.4229035114005673
Root Mean Squared Error:   0.6503103193096103
[0.90727764 0.89210966 0.82952873 0.84885923 0.87418526 0.88185626
 0.85333425 0.85029317]
cross validation score:    0.8671805261311302
Difference between R2 score and cross validatio score is -  0.04419074755766306
********************************************************************************
--------------------------------------------------------------------------------
```

Hyper Parameter Tuning

```
#import the randomized search CV
from sklearn.model_selection import RandomizedSearchCV
parameters = {"n_estimators":[100,200,300,400,500,600,700,800],
              "max_depth":[1,2,3,4,5,6,7,8,9,10,15,20],
              "max_features": [3,5,7,9],
              "min_samples_leaf":[2,3,4,5,6],
              "max_features":['auto','sqrt'],
              "min_samples_split":[2,5,8,10,12,18]}
```

```
clf = RandomizedSearchCV(RandomForestRegressor(), parameters, cv=8)
clf.fit(x_train, y_train)
clf.best_params_  #Best parameters
```

```
{'n_estimators': 200,
 'min_samples_split': 10,
 'min_samples_leaf': 2,
 'max_features': 'auto',
 'max_depth': 20}
```

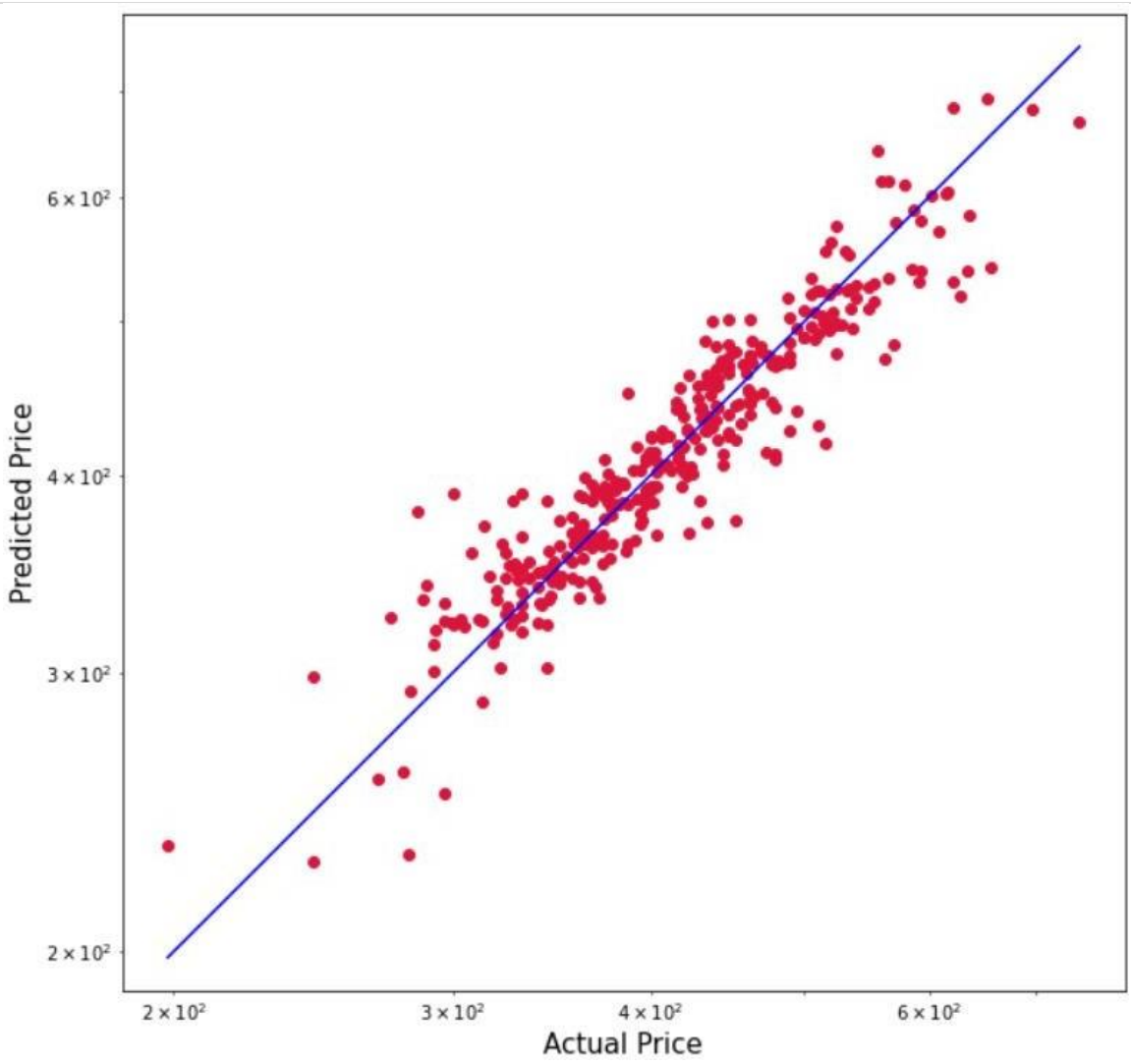
```
clf_pred = clf.best_estimator_.predict(x_test)

r2_score(y_test, clf_pred)
```

```
0.8851699605089569
```

Saving the model

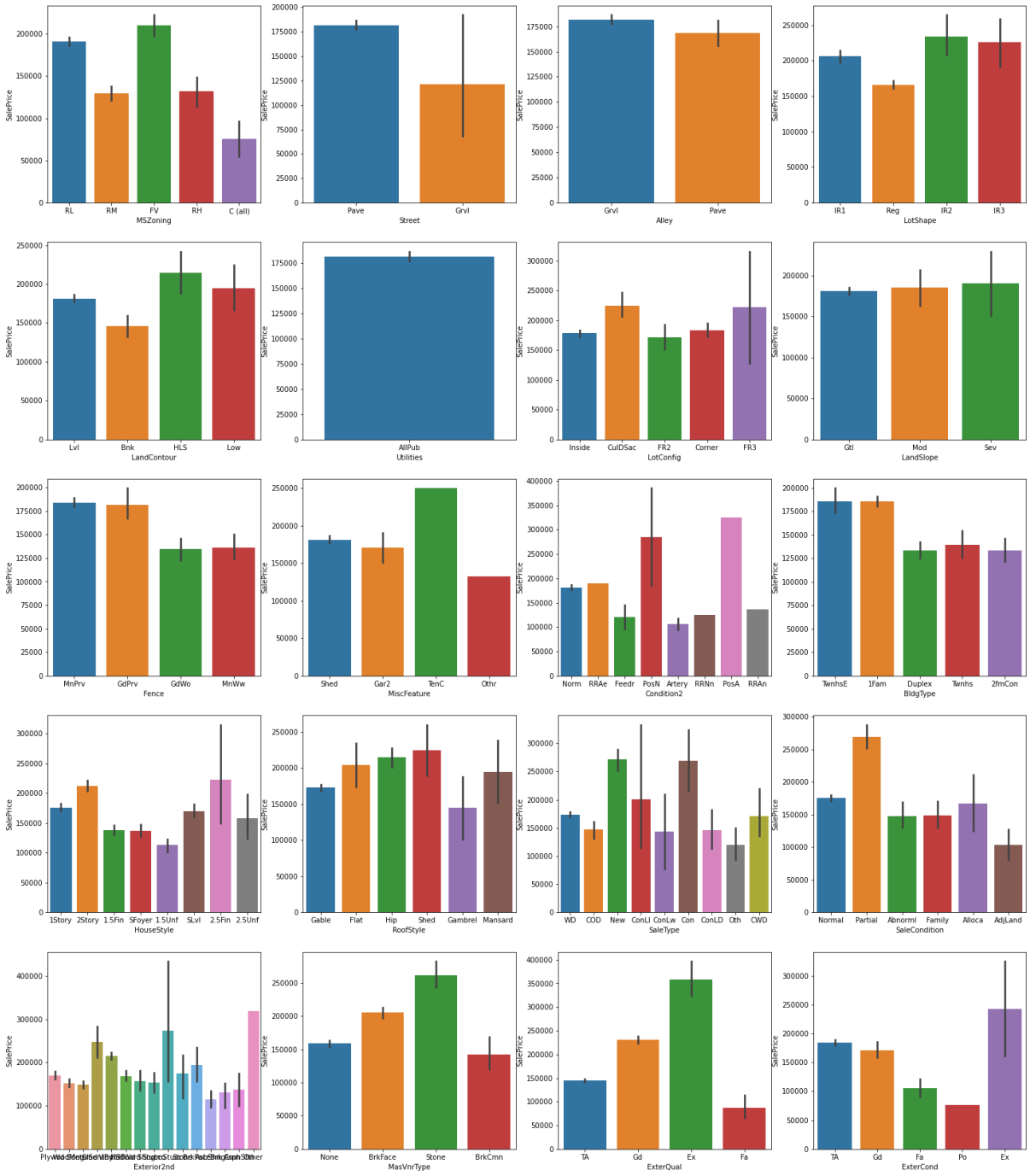
```
import joblib
joblib.dump(clf.best_estimator_,"Housing.obj")
RF_from_joblib=joblib.load('Housing.obj')
Predicted = RF_from_joblib.predict(x_test)
plt.figure(figsize=(10,10))
plt.scatter(y_test, Predicted, c='crimson')
plt.yscale('log')
plt.xscale('log')
p1 = max(max(Predicted), max(y_test.max(axis=1)))
p2 = min(min(Predicted), min(y_test.min(axis=1)))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('Actual Price', fontsize=15)
plt.ylabel('Predicted Price', fontsize=15)
plt.axis('equal')
plt.show()
```

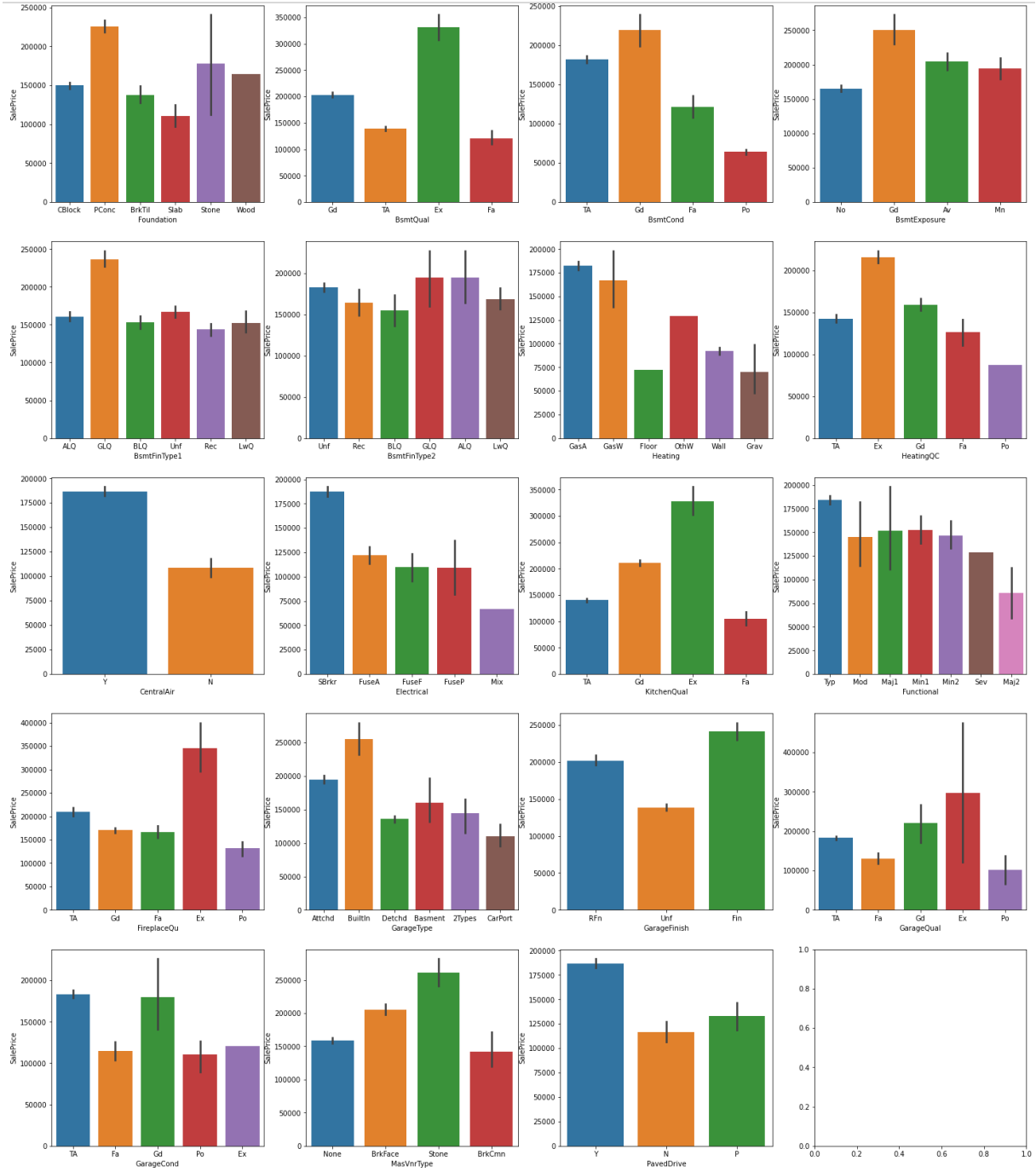

Predicted Price
Actual Price
6×10²
4×10²
3×10²
2×10²
2×10²
3×10²
4×10²
6×10²

Key Metrics for success in solving problem under consideration

As this is a regression problem, we are required to predict the continuous feature (Sale Price) I have used R2 score, mean absolute error, mean squared error and root mean squared error.

Visualizations





Observation:

- Floating Village Residential, Residential Low Density has high sales price and commercial buildings has less price
- Properties with paved streets has high sales prices compared to gravel roads.
- Properties with paved alleys has high sales prices.
- Regular shaped properties have less price moderately irregular properties have high

prices

- Hill side properties has high sales price
- The data we collected every building has all the utilities.
- Properties with Cul-de-sac kind of lot configuration has high sale price
- Properties with moderate to severe slope has high sale prices
- Properties in Northridge, Northridge Heights, Stone Brook locations tops in sales prices
- Properties within With-in 200' of North-South Railroad has highest price followed by properties with Adjacency to positive off-site feature
- Townhouse Inside Unit, Single-family Detached type of properties has high sale prices
- Two and one-half story properties with 2nd level finished has high sale prices followed by 2 story buildings
- Properties with shed type of roofs has high sale prices followed by hip and flat roofs.
- Properties which just got constructed has high sale prices and properties with 15% Contract Down payment regular terms have also high sale prices.
- Homes that are partially constructed during last assessment has high sale prices.
- Properties with stone type of Masonry veneers has high prices.
- Properties with excellent exterior material quality and excellent exterior condition has higher sale prices.
- Properties with poured concrete type of foundation has high sale prices.
- Properties with basement height of 100+ inches have high sale price
- Properties with good basement condition has higher sale price whereas properties with poor basement condition has low price
- Properties with good exposure to walkout or garden level walls has high sales price
- Properties with Good Living Quarters has high sales prices
- Properties with Gas forced warm air furnace and Gas hot water or steam heat has high sale prices.
- Properties with excellent heating conditions tend to have high prices
- Properties with central air conditioning has high sales prices.
- Properties with Standard Circuit Breakers & Romex has high sales values
- Properties with excellent kitchen quality has high price.
- Typical functioning homes has high sale price.
- Properties with excellent fireplace quality has high price.
- Properties with built in garages has high sales price followed by attached garages. Properties which do not have any garage has less sales price
- If the interior of garage is completely finished, those properties have high sale price
- Properties with garages in good condition with excellent quality has high sale prices.
- Properties with paved drive ways has high sales prices.
- Properties with excellent pool quality has high prices
- Properties with 2-STORY built in 1946 or NEWER has high sales prices.

Interpretation of the Results

Most of the features were having missing values. Numerical continuous variables were right-skewed, so I used log-normal transformation. There were some categorical variables which did not have much variance. There were some outliers in the data but due to data shortage, I have not removed the outliers. Most of the continuous numerical variables were having positive linear relation with the target variable. From Random Forest Regressor R2 score was 88.51% of the variance of the dependent variable being studied is explained by the variance of the independent variable.

Higher the R2 score means the model is well fit for the data. However, if the R2 score is very high, it might be a case of overfitting. Other metrics Mean Absolute Error, Mean Squared Error and Root Mean Squared Error, with gradient boosting these scores are less than compared to other models. If these errors are fewer that means the model shows less errors.

Conclusion

Key Findings and Conclusions of the Study

From this dataset, I get to know that each feature plays a very important role to understand the data. Data format plays a very important role in the visualization and Appling the models andalgorithms.

Learning Outcomes of the Study in respect of Data Science

The power of visualization is helpful for the understanding of data into the graphical representation its help me to understand what data is trying to say, Data cleaning is one of the most important steps to remove missing value or null value fill it by mean, median or by mode or by 0.
Various algorithms I used in this dataset to get out best result and save that model. The best algorithm is Random Forest Regressor.

Limitations of this work and Scope for Future Work

The limitations of this project are, it has lots of outliers. If we try to fix outliers by some techniquethe accuracy goes down. If we dope the outliers then we are losing everything.

In future, if someone do the proper and detailed study of this dataset's each column then we will not lose much amount of data and the accuracy will be so high.