

Project Report On

Housing: Price Prediction

Submitted By

Mohd Sharik Hashmi

ACKNOWLEDGMENT

I would like to recognize the invaluable assistance of the team of DataTrained Academy and FlipRobo Technologies for providing guidance to work on real-time data projects which also helped me in doing a lot of Research and fetching out the insights.

All the required information and the dataset are provided by Flip Robo Technologies.

References:

1. <https://towardsdatascience.com/machine-learning-projectpredicting-boston-house-prices-with-regression-b4e47493633d>
2. <https://www.kaggle.com/ammamr111/house-price-predictionan-end-to-end-ml-project>
3. <https://medium.com/codex/house-price-prediction-withmachine-learning-in-python-cf9df744f7ff>

TABLE OF CONTENTS

1. Introduction

- 1.1 Business Problem Framing
- 1.2 Conceptual Background of the Domain Problem
- 1.3 Review of literature
- 1.4 Motivation for the Problem Undertaken

2. Analytical Problem Framing

- 2.1 Mathematical/ Analytical Modelling of the Problem
- 2.2 Data Sources and their format
- 2.3 Data Pre-processing Done

2.4 Data Inputs- Logic- Output Relationships

2.5 Hardware & Software Requirements & Tools Used

3. Model/s Development and Evaluation

3.1 Identification of possible Problem-solving approaches
(Methods)

3.2 Visualizations

3.3 Testing of Identified Approaches (Algorithms)

3.4 Run and Evaluate Selected Models

3.5 Key Metrics for success in solving problem under
consideration

3.6 Interpretation of the Results

4. Conclusion

4.1 Key Findings and Conclusions of the Study

4.2 Learning Outcomes of the Study in respect of Data Science

4.3 Limitations of this work and Scope for Future Work

1. INTRODUCTION

Thousands of houses are sold every day. There are some questions every buyer asks himself like: What is the actual price that this house deserves? Am I paying a fair price? Also Is it the location? Is it the overall quality of the house? Is it the size? Could it be sold at a good price in future? All these questions come in to our mind when we decide to purchase a house. In this

study, a machine learning model is proposed to predict a house price based on data related to the house (its size, the year it was built in, etc.). During the development and evaluation of our model, we will show the code used for each step followed by its output. This will facilitate the reproducibility of our work.

1.1 Business Problem Framing:

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

The project endeavours to extensive data analysis and implementation of different machine learning techniques in python for having the best model with most important features of a house on insight of both business value and realistic perspective.

Business goal: With the help of available independent variables, we need to model the price of the houses. This model will then be used by the management to understand how exactly the prices vary with the variables. They can 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.

1.2 Conceptual Background of the Domain Problem:

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.

The problem statement is related to the US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The company is looking at prospective properties to buy houses to enter the market. It is 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 house?
- How do these variables describe the price of the house?

In this section, we evaluate widely used regression technologies like Linear Regression, regularization, bagging and boosting and many more ensemble techniques to predict the house sale price result.

1.3 Review of Literature:

The relationship between house prices and the economy is an important motivating factor for predicting house prices (Pow, Janowicz, & Liu, 2014). There is no accurate measure of house prices (Pow, Janulewicz, & Liu, 2014). Pow states that Real Estate property prices are linked with economy (Pow, Janulewicz, & Liu, 2014). He also states there is no accurate measure of house prices. A property's value is important in real estate transactions. Pow tries to predict the sold and asking prices of real estate

values without bias to help both buyers and sellers make their decisions. A property's value is important in real estate transactions.

Housing market is important for economic activities (Khamis & Kamarudin, 2014). Traditional housing price prediction is based on cost and sale price 3 comparison. So, there is a need for building a model to efficiently predict the house price.

Based on the sample data provided to us from our client database where we have understood that the company is looking at prospective properties to buy houses to enter the market. The data set explains it is a regression problem as we need 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. House prices trends are not only the concerns for buyers and sellers, but they also indicate the current economic situations. Therefore, it is important to predict the house prices without bias to help both buyers and sellers make their decisions.

1.4 Motivation for the Problem:

Undertaken I have gone thorough many projects before, but this project has given me an idea to handle large number of attributes. By doing this project I have got an idea about how to deal with data exploration where I have used all my analyzation skills to predict the house price using ML models. The model will be a good way for both buyers and sellers to understand the pricing dynamic of a new market. The main objectives of this study are as follows:

- ✓ To apply data pre-processing and preparation techniques in order to obtain clean data.
- ✓ To build machine learning models able to predict house price based on house features.
- ✓ To analyse and compare models' performance in order to choose the best model. By processing the above objects, I will be able to find which

variables are important to predict the price of house? And how do these variables describe the price of the house? The relation between house prices and the economy is an important factor for predicting house prices.

2. ANALYTICAL PROBLEM FRAMING

2.1 Mathematical/ Analytical Modelling of the Problem:

The house price model is based on a demand function for housing services and a standard life-cycle model of utility for a representative household. This is a common approach in academic research into house prices. The study is to predict the sale price of the house and analysing which features are important and how they contribute in the prediction. There are two datasets. One is train dataset which is supervised and another one is test dataset which is unsupervised. The target variable is “Sale Price” and it is a regression type problem. I have used train dataset to build machine learning models and then by using this model I made prediction for the test dataset.

I have observed some columns having more than 85% of zero entries and 70% of null values so, I decided to drop those columns. I have performed both univariate and bivariate analysis to analyse the sale price of the house. I have analysed the categorical and numerical features using categorical plots and numerical plots respectively to get better insights from the data. In this project I have done various mathematical and statistical analysis such as describing the statistical summary of the columns, feature engineering, treating null values, removing outliers, skewness, encoding the data etc. Checked for correlation between the features and visualized it using heat map.

Also, I built many regression algorithms while building machine learning models, used hyper tuning method for best model and saved the best

model. Finally, I predicted the sale price of the house using the saved trained model.

2.2 Data Sources and their formats:

✓ A US-based housing company named Surprise Housing has collected the dataset from the sale of houses in Australia and the data is provided by Flip Robo Company and it is in csv format.

There are 2 data sets: 1. Train dataset 2. Test dataset 5 ✓ Train dataset will be used for training the machine learning models. The dataset contains 1168 rows and 81 columns, out of 81 columns, 80 are independent variables and remaining 1 is dependent variable (Sale Price).

✓ Test dataset contains all the independent variables, but not the target variable. We will apply the trained model to predict the target variable for the test data. The dataset contains 292 rows and 80 columns.

✓ The dataset contains both numerical and categorical data. Numerical data contains both continuous and discrete variables and categorical data contains both nominal and ordinal variables.

✓ I can concatenate both train and test data, but this may cause data leakage so I decided to process both the data separately.

2.3 Data Pre-processing Done

- Firstly, I have imported the necessary libraries and imported both train and test datasets which were in csv format. And process both datasets simultaneously.
- I have done some statistical analysis like checking shape, nunique, column names, data types of the features, info about the features, value counts etc for both train and test data.
- I have dropped the columns "Id" and "Utilities" from both the datasets. Since Id is the unique identifier which contains unique value throughout the data also all the entries in Utilities column were unique. They had no significance impact on the prediction.
- While looking into the value count function I found some of the columns having more than 85% of zero values so, I dropped those columns from both the datasets as they might create skewness which will impact my model.
- Also, I have done some feature extraction as the datasets contained some time variables like YearBuilt, YearRemodAdd, GarageYrBlt and YrSold.

Converting them into age seem more meaningful as they offer more information about the longevity of the features. It is analogous to the fact that, the statement "Mr. X died at the age of 66 years" holds more information for us than the statement "Mr. X died in the year 2019". So, I have extracted age information from the datetime variables by taking the difference in year between the year the house was built and year the house was sold and dropped the year columns.

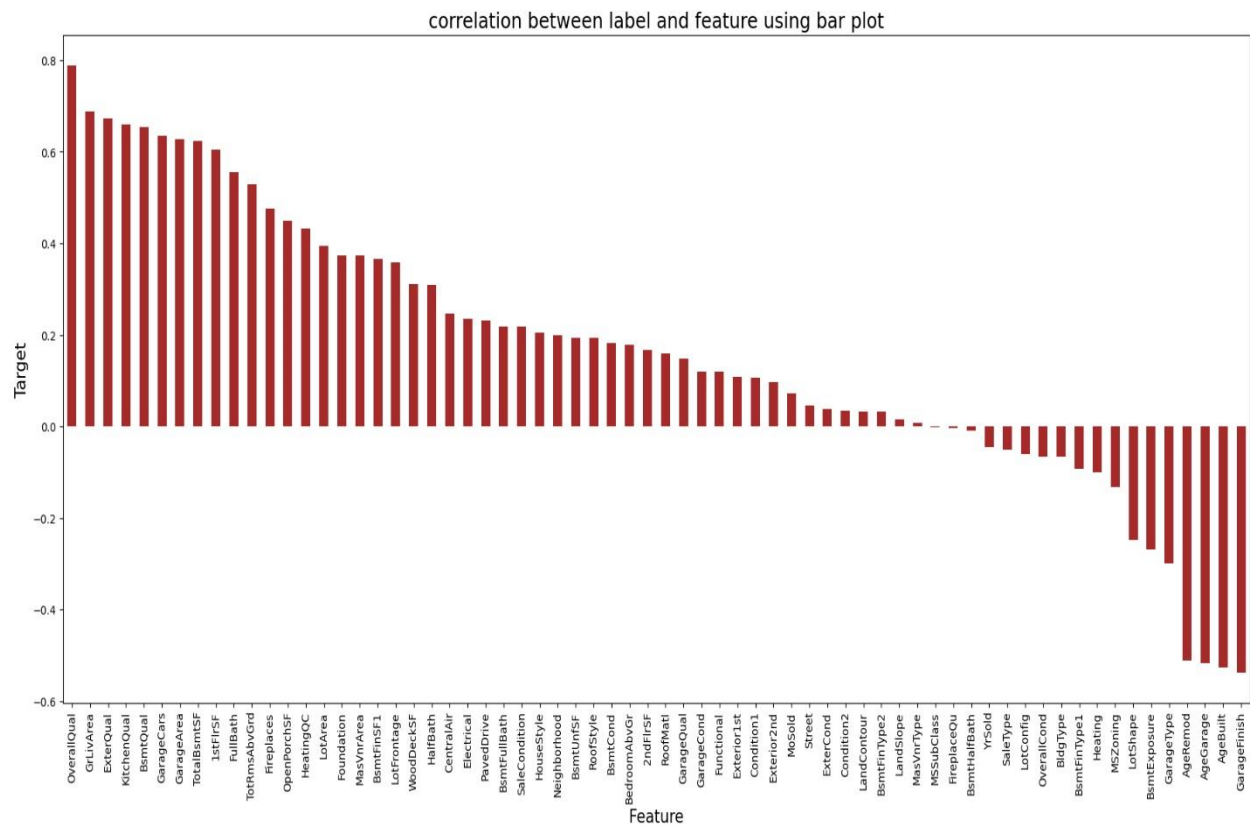
- I checked the null values and found them in some of the columns. So, I imputed null values present in categorical and numerical columns using mode and mean methods respectively. I found some columns having more than 80% of null values so, I dropped those columns to overcome with the skewness.
- Described statistical summary of both train and test datasets.

- Visualized each feature using seaborn and matplotlib libraries by plotting several categorical and numerical plots.
- Identified outliers using box plots in both datasets. I tried to remove them using Zscore method and got huge data loss of around 19% , so removed outliers using percentile method by setting data loss to 2%.
- Checked for skewness and removed skewness in numerical columns using power transformation method (yeo-johnson). Also dropped KitchenAbvGr columns as it contains zero values throughout the data after using power transformation.
- Encoded both train and test data frames using Ordinal Encoder. Also replaced some categorical columns having ratings by numbers based on specific condition.
- Used Pearson's correlation coefficient to check the correlation between label and features.
- Scaled both the datasets using Standard Scalar method and used regression algorithms to build ML models.
- All these steps were performed to both train and test datasets simultaneously.

2.4 Data Inputs- Logic- Output Relationships

- To analyse the relation between features and target I have done EDA where I analysed the relation using many plots like bar plot, reg plot, scatter plot, line plot, swarm plot, strip plot, violin plot etc. And found some of the columns like OverallQual, TotalRmsAbvGrd, FullBath, GarageCars etc have strong positive linear relation with the label.

- I have checked the correlation between the target and features using heat map and bar plot. Where I got the positive and negative correlation between the label and features.



From the above bar plot I can notice the positive and negative correlation between the features and label Sale Price. Below are the correlated features.

Important features that affect Sale Price positively and negatively.

Features having high Positive correlation with label

- OverallQual
- GrLivArea
- ExterQual
- KitchenQual
- BsmtQual
- GarageCars
- GarageArea
- TotalBsmtSF
- 1stFlrSF
- FullBath
- TotRmsAbvGrd

Features having high Negative correlation with label Heating

- MSZoning
- LotShape
- BsmtExposure
- GarageType
- AgeRemod
- AgeGarage
- AgeBuilt
- GarageFinish

2.5 Hardware & Software Requirements & Tools Used:

To build the machine learning projects it is important to have the following hardware and software requirements and tools.

Hardware required:

- Processor: core i5 or above
- RAM: 8 GB or above
- ROM/SSD: 250 GB or above

Software required:

- Anaconda 3- language used Python 3

Libraries required:

Importing important libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

import numpy as np: It is defined as a Python package used for performing the various numerical computations and processing of the multidimensional and single dimensional array elements. The calculations using Numpy arrays are faster than the normal Python array.

import pandas as pd: Pandas is a Python library that is used for faster data analysis, data cleaning and data pre-processing. The data-frame term is coming from Pandas only.

import matplotlib.pyplot as plt: Matplotlib and Seaborn acts as the backbone of data visualization through Python. Matplotlib: It is a Python library used for plotting graphs with the help of other libraries like Numpy and Pandas. It is a powerful tool for visualizing data in Python. It is used for creating statically interferences and plotting 2D graphs of arrays.

import seaborn as sns: Seaborn is also a Python library used for plotting graphs with the help of Matplotlib, Pandas, and Numpy. It is built on the roof of Matplotlib and is considered as a superset of the Matplotlib library. It helps in visualizing univariate and bivariate data.

```
from scipy. stats import zscore from sklearn. preprocessing import  
PowerTransformer
```

```
from sklearn. preprocessing import OrdinalEncoder
```

```
from sklearn.preprocessing import StandardScaler
```

```
from statsmodel.starts.outliers_influence import  
variance_inflation_factor
```

With the above sufficient library, we can perform pre-processing and data cleaning. For building my ML models libraries below are required.

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score

from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import BaggingRegressor
import xgboost as xgb
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn import metrics
```

3.MODEL/S DEVELOPMENT AND EVALUATION

3.1 Identification of possible Problem-solving approaches (Methods):

- I have used imputation methods to treat the null values.
- Used percentile method to remove outliers.
- Removed skewness using power transformation (yeo-johnson) method.
- Encoded the object type data into numerical using Ordinal Encoder.
- I have used Pearson's correlation coefficient method to check the correlation between the dependent and independent variables.

- I have scaled the data using Standard Scalar method to overcome with the data biasness.
- Used many Machine Learning models to predict the sale price of the house.

3.2 Visualizations

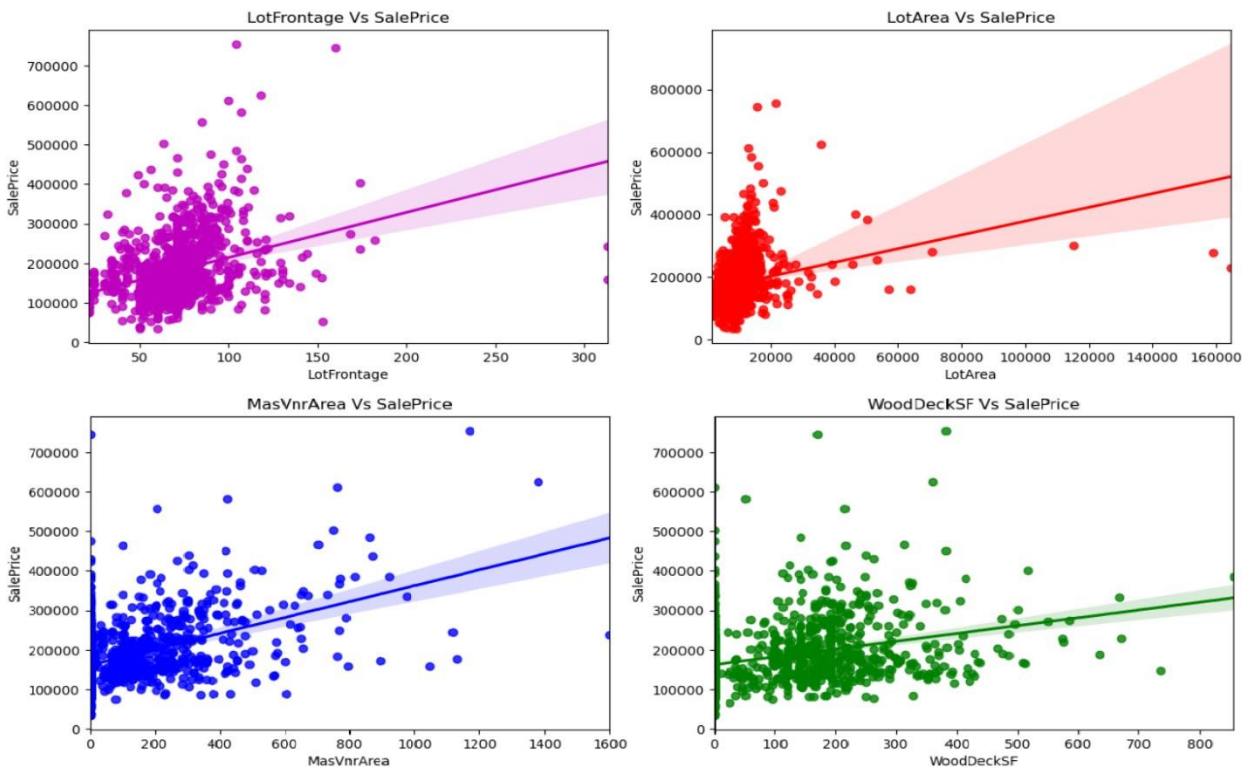
I have analyzed the data using both univariate and bivariate analysis to visualize the data.

In univariate analysis I have used pie plots, count plots and distribution plot and in bivariate analysis I have used bar plots for categorical columns and used reg plots, scatter plots, strip plots, swarm plots, violin plots and line plot to visualize numerical columns.

These plots have given good pattern. Here I will be showing only bivariate analysis to get the better insights of relation between label and the features.

1. Visualizing Continuous variables vs SalePrice

Continuous variables vs SalePrice



Observations:

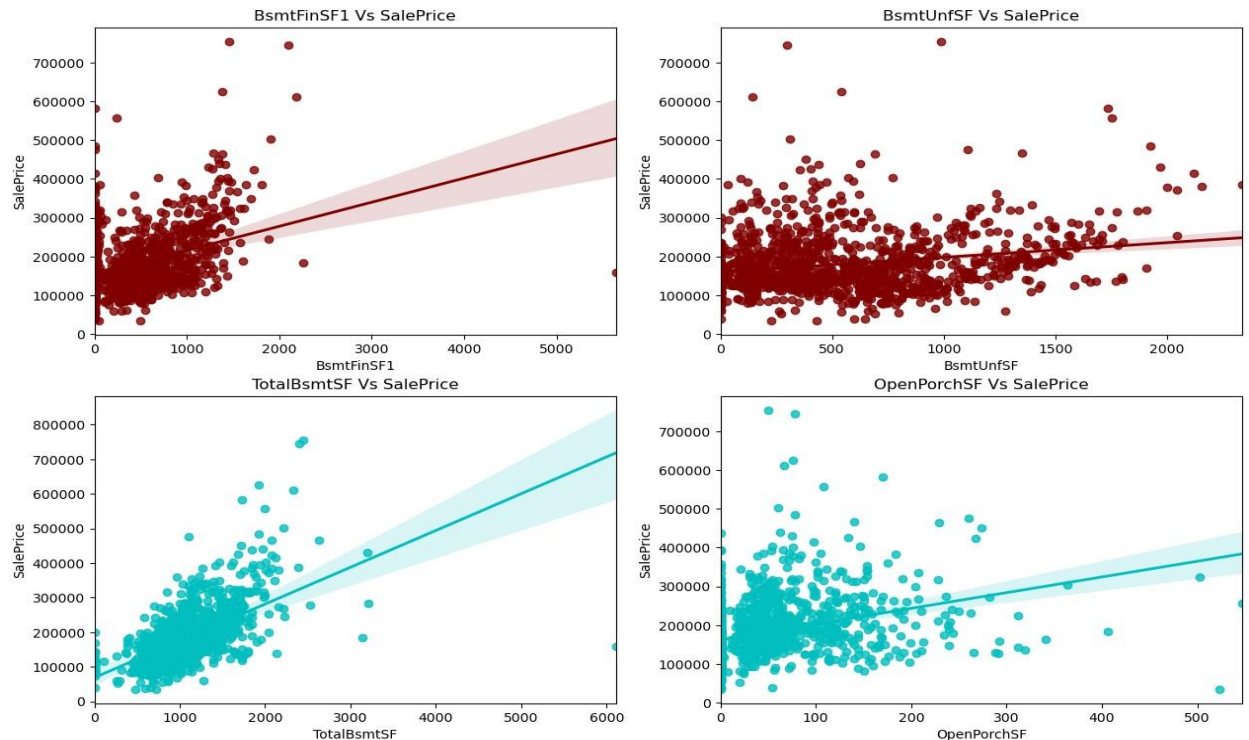
SalePrice vs LotFrontage: From the plot we can observe there is no much linear relation between the label and feature. If the linear feet of street connected to property is more, the sale price is also high.

SalePrice vs LotArea: There is weakly positive linear relation between the label and feature. But the sale price is high when lot size has around 20000 square feet area. Also, as the lot size increases the price is also increasing moderately.

SalePrice vs MasVnrArea: There is bit positive linear relation between feature and target. Also the sale price is high when Masonry veneer area has around 50-400 square feet. So as the Masonry veneer area in square feet increases sale price is also increasing.

SalePrice vs WoodDeckSF: There is weakly positive linear relation between the feature and target. As the Wood deck area increases, sale price is also increasing.

Continuous variables Vs SalePrice



Observations:

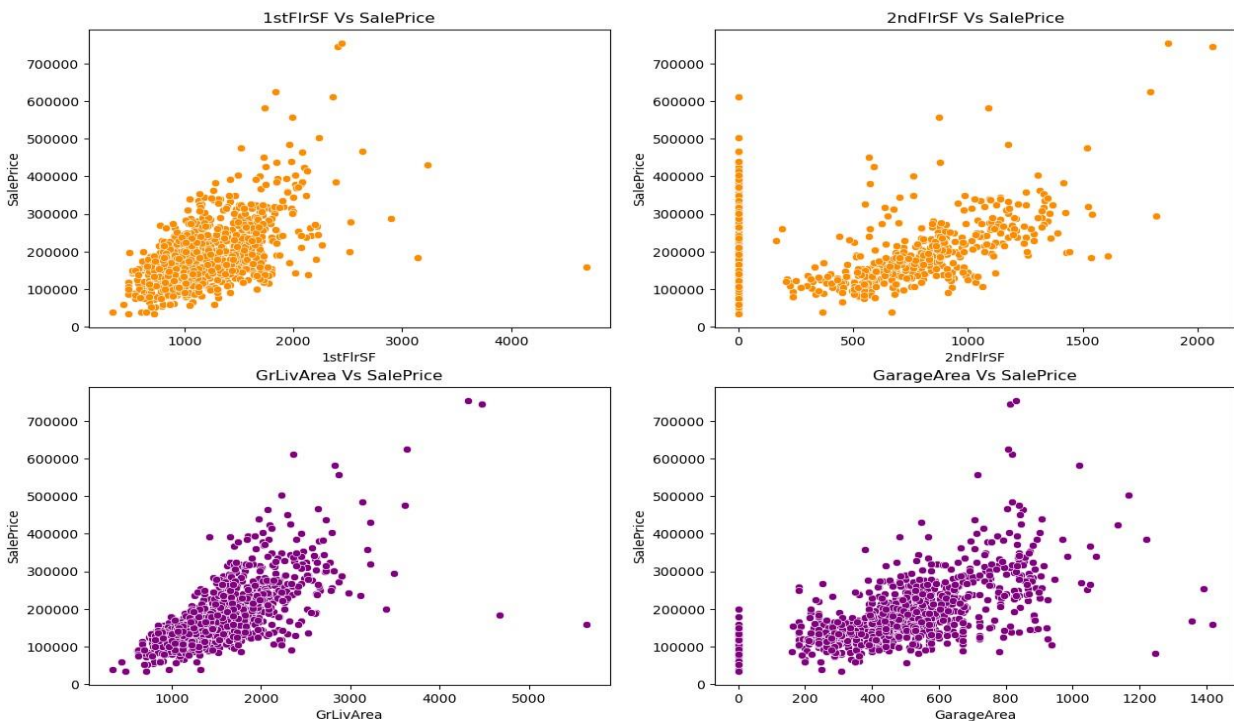
SalePrice vs BsmtFinSF1: There is weakly positive linear relation between feature and label. The sale price is high that is 100000-300000 when basement square feet lie upto 1500 square feet. So as the type 1 basement finished square feet increases, sale price also increases.

SalePrice vs BsmtUnfSF: There is positive linear relation between the target and BsmtUnfSF. When the unfinished basement area is below 1000 square feet, the sale price is high.

SalePrice vs TotalBsmtSF: There is positive linear relation between sale price and TotalBsmtSF. As total basement area increases, sale price also increases.

SalePrice vs OpenPorchSF: There is a linear relation between the label and feature. The sale price is high when Open porch area is below 200sf. Here also as the Open porch area increases, sale price also increases.

Continuous variables Vs SalePrice



Observations:

SalePrice vs 1stFlrSF: There is a linear relation between the label and feature. As we can observe in the plot, the sale price is high when the

first-floor area lies between 500-2000 square feet. So as the 1st floor area increases, sales price also increases moderately.

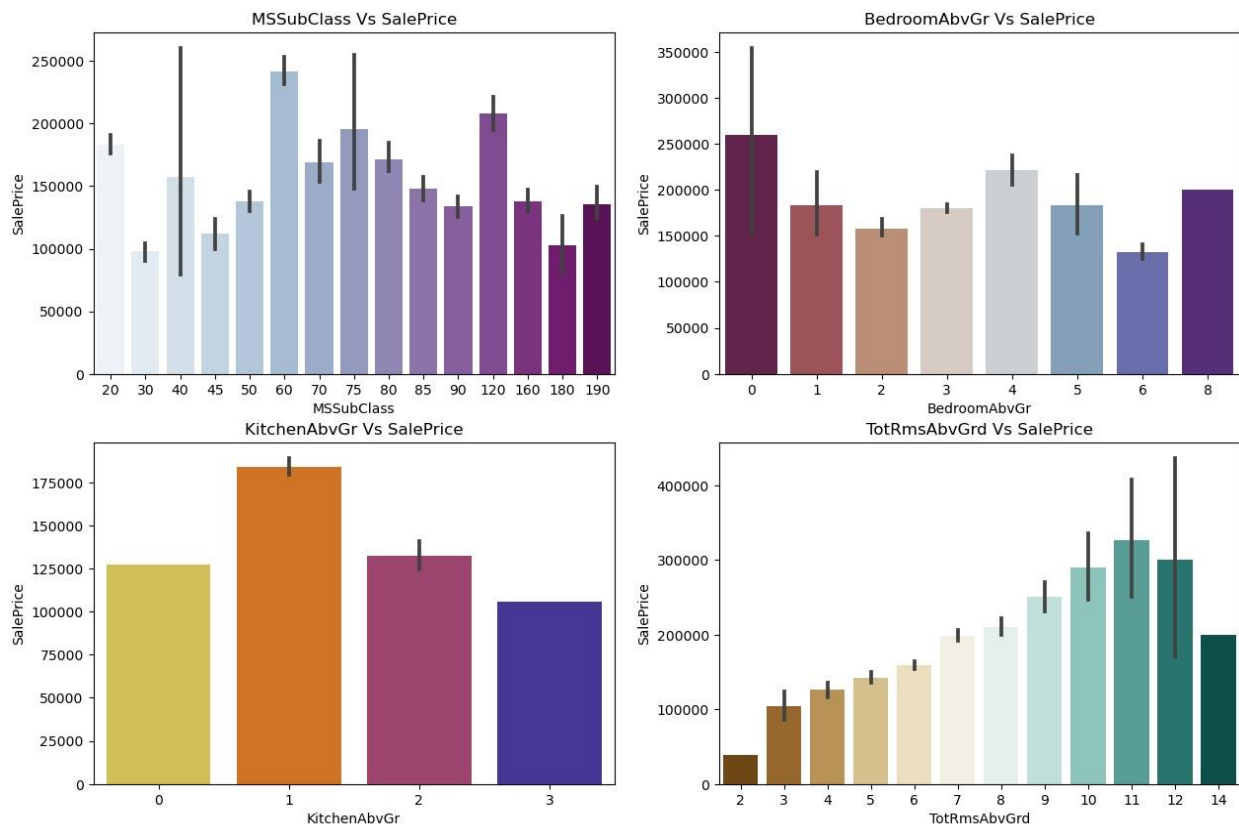
SalePrice vs 2ndFlrSF: There is a positive correlation between SalePrice and 2ndFlrSF. So, it is obvious that the sale price increases based on the floors.

SalePrice vs GrLivArea: Most of the houses have above grade living area. There is a positive correlation between the label and feature. Here as the above grade living area increases, sale price also increases.

SalePrice vs GarageArea: Similar to 2nd floor sf, here also positive linear relation between the label and feature. As size of garage area increases, sale price also increases. The sale price is high when size of garage area is between 200-800 square feet.

2. Visualizing Discrete varia vs SalePrice

Discrete variables Vs SalePrice



bles

Observations:

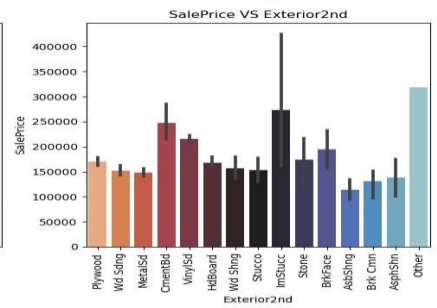
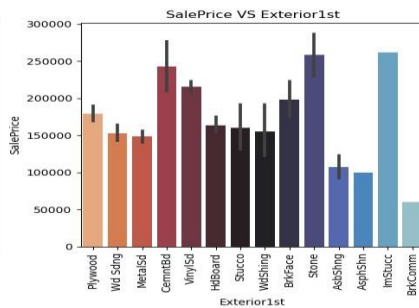
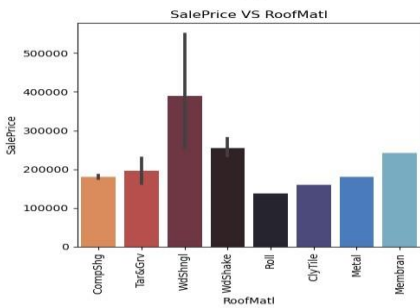
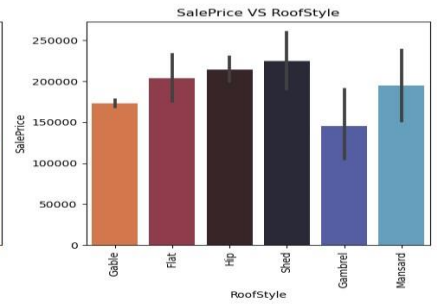
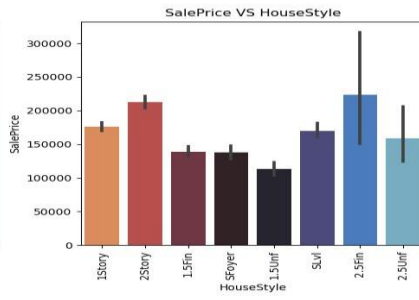
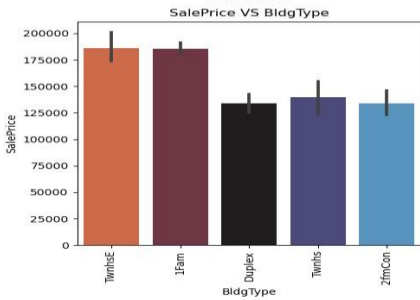
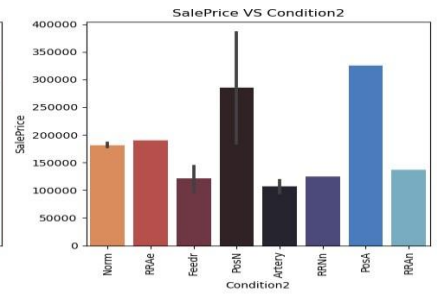
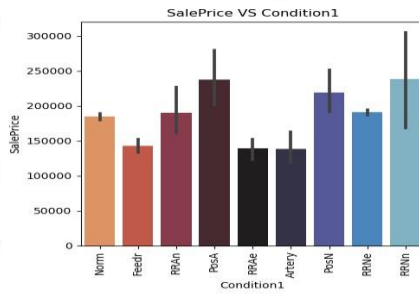
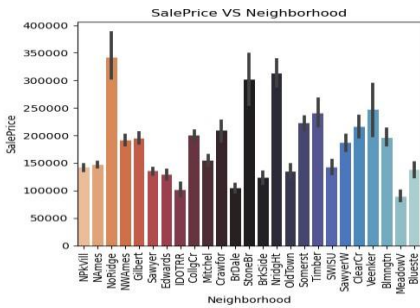
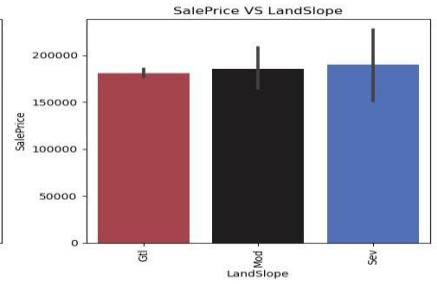
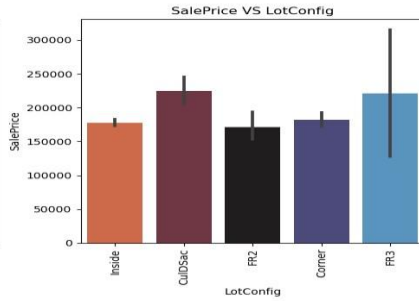
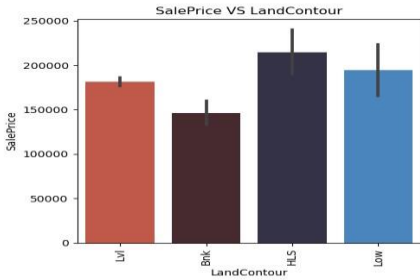
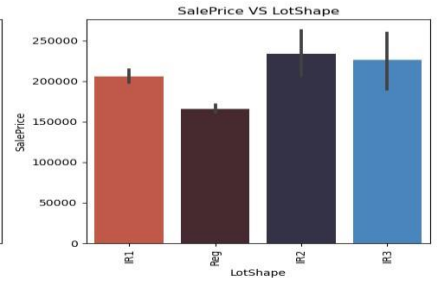
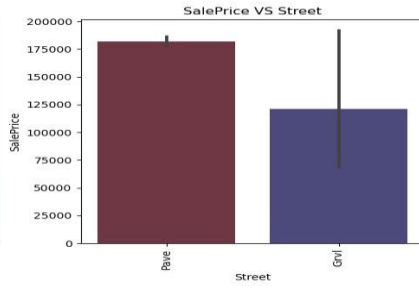
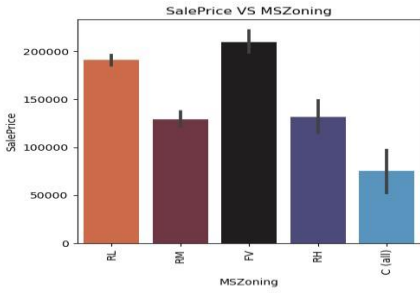
SalePrice vs MSSubClass: The sale price is high for the MSSubClass 60,120 and 20.

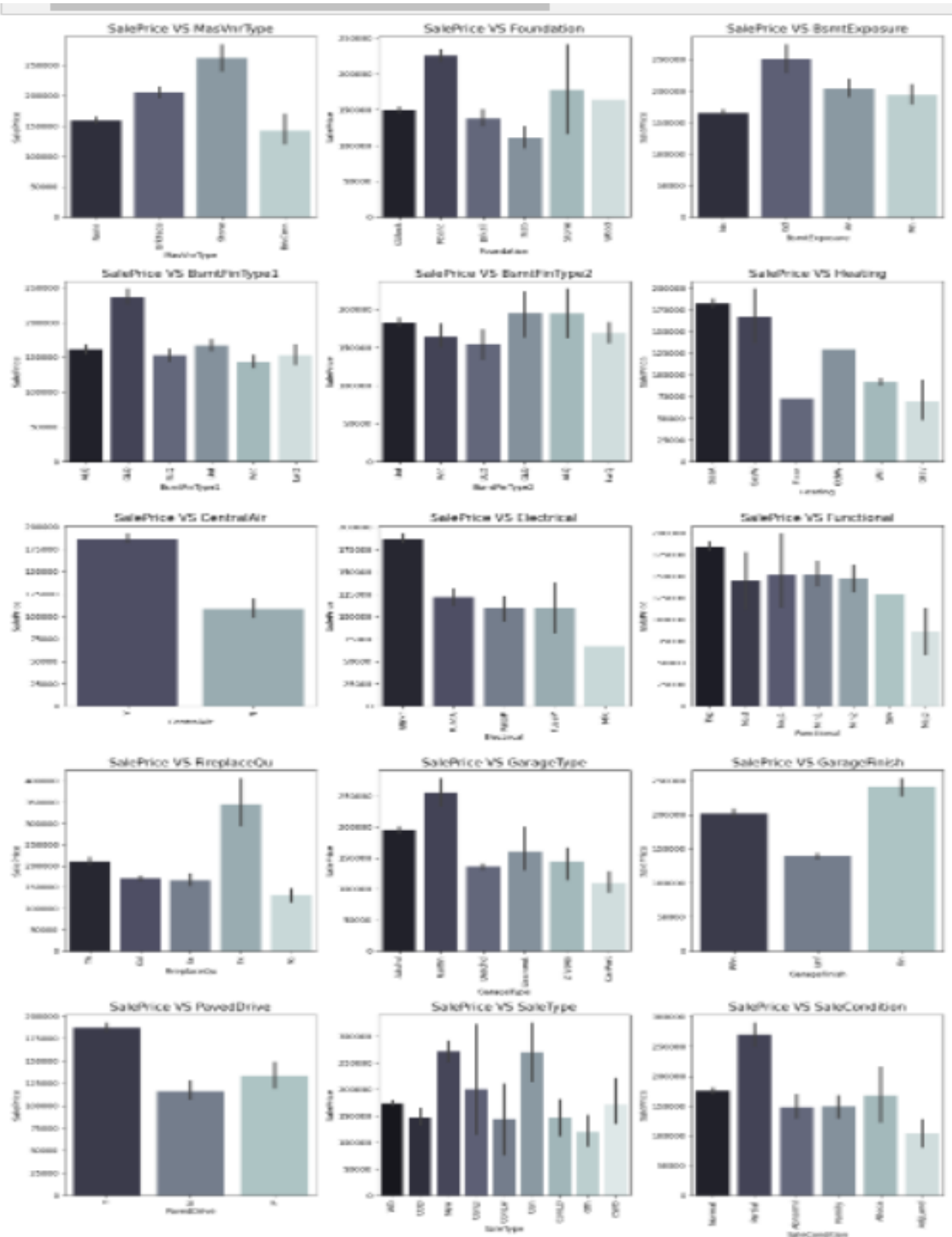
SalePrice vs BedroomAbvGr: Many houses are having 0 and 4 bedrooms have high sales price also houses having 8 bedrooms also have high sales price. Other bedroom grades have average sale price.

SalePrice vs KitchenAbvGr: Most of the houses have single kitchen and few houses have 2 kitchens. The sale price is also high in case of the houses having single kitchen.

SalePrice vs TotRmsAbvGrd: We can observe some linear relation between Total rooms above grade and Sale Prices as the number of rooms increases the sales price also increases.

3. Visualizing Nominal variables vs SalePrice:





Observations:

SalePrice vs MSZoning: Most of the houses are belongs to Floating Village Residential followed by Residential Low Density. The houses from this zone are have high sale price compared to other zones.

SalePrice vs Street: By observing the bar plot, it is obvious that the property of house with Paved type of road have high SalePrice and the houses in gravel roads have very less sale price.

SalePrice vs LotShape: Most of the houses having moderately irregular and irregular shape of property have high sale price and houses with regular type of property have less sale piece compared to others.

SalePrice vs LandContour: The houses having the hillside and depression property flatness have high sale price compared to others.

SalePrice vs LotConfig: Most of the houses with Frontage on 3 sides of property have high sale price compared to others.

SalePrice vs LandSlope: There is no significance difference between the slope of the property. As we can observe the houses having Gentle slope, Moderate Slope and Severe Slope have same sale price.

SalePrice vs Neighborhood: The houses which are located near Northridge have high sale price compared to others.

SalePrice vs Condition1: The houses having the conditions adjacent to positive off-site feature and houses within 200' of North-South Railroad have high sale price compared to others.

SalePrice vs Condition2: The houses having the conditions near positive off-site feature park, greenbelt, etc and adjacent to positive off-site feature have high sale price.

SalePrice vs BldgTypz: Most of the houses are Single-family Detached and Townhouse End Unit and they have higher sale price compared to other categories.

SalePrice vs HouseStyle: Houses which are having style of dwelling 2nd level finished and two story have high sale price compared to other types.

SalePrice vs RoofStyle: The houses having the roof style Flat, Hip and Shed have high sale price and the houses having garble roof style have less sale price.

SalePrice vs RoofMatl: Houses with Wood Shingles root materials have high sale prices.

SalePrice vs Exterior1st: Houses having Imitation Stucco, Stone and Cement Board as 1st exterior cover have high sale price.

SalePrice vs Exterior2nd: Houses having Imitation Stucco and other as 2nd cover have high sale price.

SalePrice vs MasVnrType: Houses having Stone Masonry veneer type have high sale price than other types.

SalePrice vs Foundation: Houses having Poured Concrete as foundation type have high sale price compared to other types.

SalePrice vs BsmtExposure: Houses having good walkout or garden level walls have high sale price compared to others.

SalePrice vs BsmtFinType1: The sale price is high for the houses containing good living quarters basement finished area.

SalePrice vs BsmtFinType2: The sale price is moderately high for the houses having good living quarters and average living quarters.

SalePrice vs Heating: The houses having the heating type gas forced warm air furnace and gas hot water or steam heat have high sale price.

SalePrice vs CentralAir: Most of the houses have central air conditioning so it is obvious that these houses have high sale price.

SalePrice vs Electrical: Most of the houses having standard circuit breakers & romex have high sale price compared to others.

SalePrice vs Functional: The houses having the typical functionality have maximum sales price and others have average sale price.

SalePrice vs FireplaceQu: The houses having excellent exceptional masonry fireplace quality have high sale price and the houses having poor fireplace quality have very less sale price compared to others.

SalePrice vs GarageType: The houses having built-in garage have high sale price compared to others.

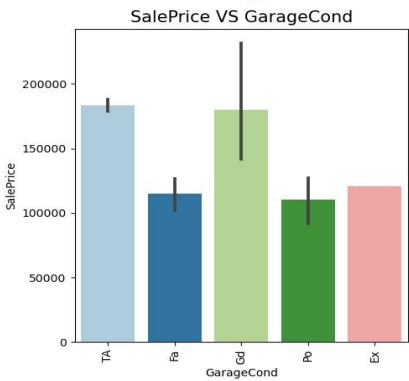
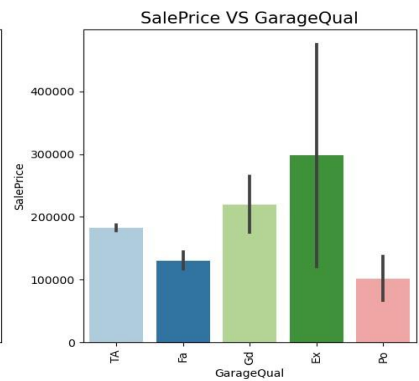
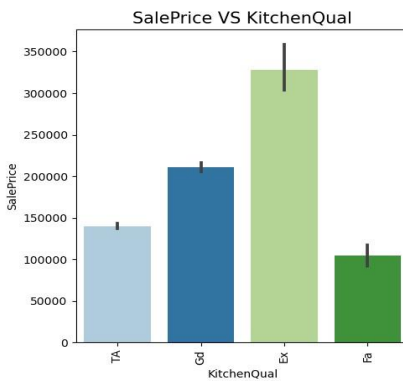
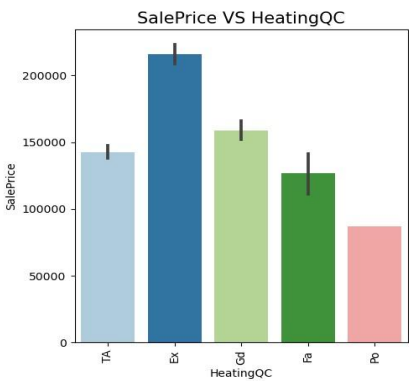
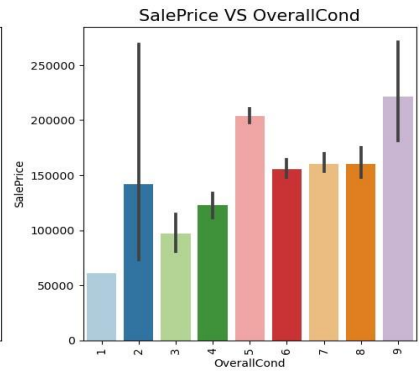
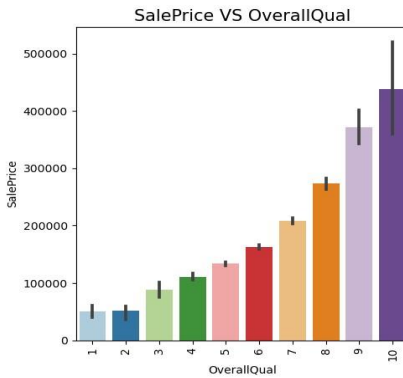
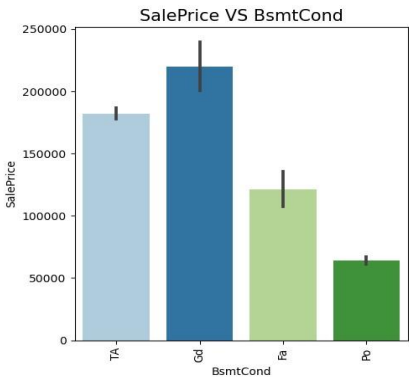
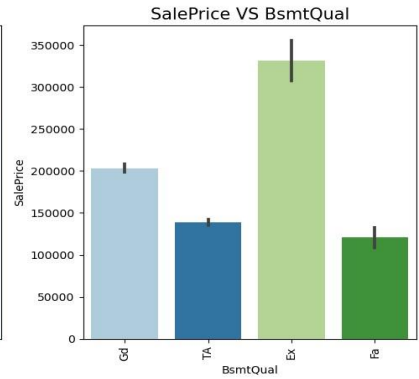
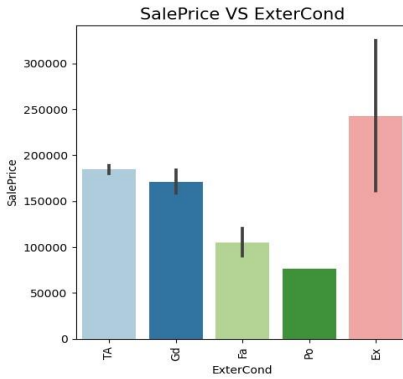
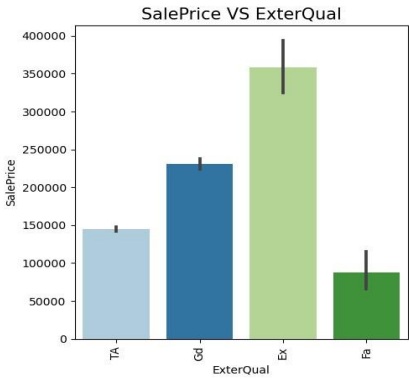
SalePrice vs GarageFinish: Garages located inside the house which is got finished have high sale price.

SalePrice vs PavedDrive: Houses having paved drive ways have high sale price.

SalePrice vs SaleType: Many houses having sale types as just constructed and sold and Contract 15% Down payment regular terms have high sale price.

SalePrice vs SaleCondition: Houses having partial sale condition that is home was not completed when last assessed have high sale price.

4. Visualizing Ordinal Variables vs SalePrice



Observations:

SalePrice vs ExterQual: Houses having excellent quality of the material on the exterior have high sale price and houses having fair quality have very less sale price.

SalePrice vs ExterCond: Houses having excellent condition of the material on the exterior have high sale price and the houses having poor condition of the material on the exterior have very less sale price compared to others.

SalePrice vs BsmtQual: The houses which evaluates the excellent quality of height of the basement have high sale price compared to others.

SalePrice vs BsmtCond: The houses which evaluates the good quality of general condition of the basement have high sale price compared to others.

SalePrice vs OverallQual: The houses which have very excellent overall quality like material and finish of the house have high sale price. Also, we can observe from the plot as the overall quality of the house increases, the sale price also increases. That is there is good linear relation between SalePrice and OverallQual

SalePrice vs OverallCond: The houses having overall condition as excellent and average have very high sale price compared to others.

SalePrice vs HeatingQC: Most of the houses having excellent heating quality and condition have high sale price.

SalePrice vs KitchenQual: Houses having excellent quality of the kitchen have high sale price compared to others.

SalePrice vs GarageQual: The sale price of the house is high for the houses having excellent garage quality.

SalePrice vs GarageCond: Houses having typical/average garage condition have high sale price and the houses having good garage condition also have high sales price compared to others.

3.3 Testing of Identified Approaches (Algorithms)

In this problem SalePrice is my target variable which is continuous in nature, from this I can conclude that it is a regression type problem hence I have used following regression algorithms to predict the sale price of the house. After the pre-processing and data cleaning I left with 67 columns including target and I used these features for prediction.

- 1. Linear Regression**
- 2. Lasso Regressor**
- 3. Ridge Regressor**
- 4. Random Forest Regressor**
- 5. Extra Trees Regressor**
- 6. Gradient Boosting Regressor**
- 7. Bagging Regressor**

3.4 Run and evaluate selected models

Linear Regression:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables. Linear regression algorithm shows a linear relationship between a dependent (y) and

one or more independent (x) variables. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

```
: # Checking r2score for Linear Regression
LR = LinearRegression()
LR.fit(x_train,y_train)

# prediction
predLR=LR.predict(x_test)
print('R2_score:',r2_score(y_test,predLR))

# Mean Absolute Error (MAE)
print('MAE:',mean_absolute_error(y_test, predLR))

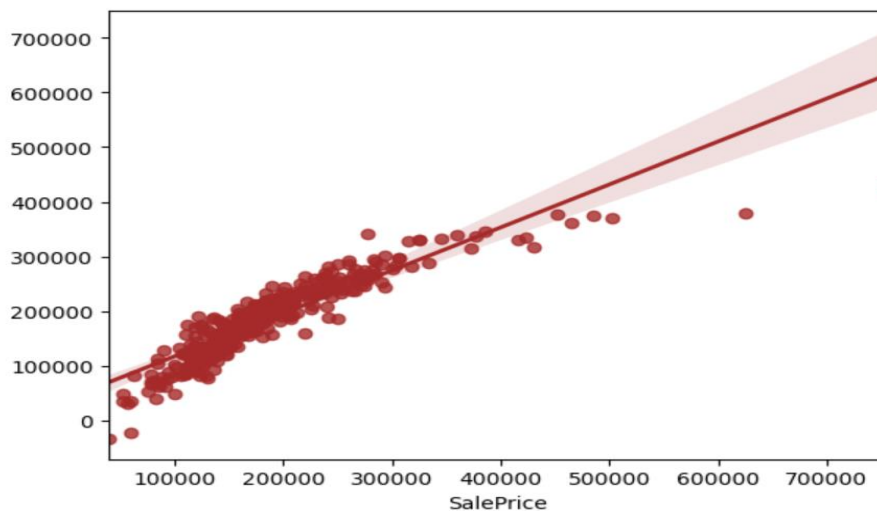
# Mean Squared Error (MSE)
print('MSE:',mean_squared_error(y_test, predLR))

# Root Mean Squared Error (RMSE)
print("RMSE:",np.sqrt(mean_squared_error(y_test, predLR)))

# Checking Cross Validation Score for Linear Regression
score=cross_val_score(LR,train_df_x,y,cv=8)
print(score)
print("cross validation score: ",score.mean())
print("Difference between R2 score and cross validation score is - ",r2_score(y_test,predLR)-abs(score.mean()))

# Visualizing the predicted values
sns.regplot(y_test,predLR,color="brown")
plt.show()

R2_score: 0.8176763159761927
MAE: 22210.516499512276
MSE: 1247984052.8474388
RMSE: 35326.81775715779
[0.79077872 0.83182544 0.78188816 0.78158497 0.71820765 0.83764851
 0.81953795 0.809018 ]
cross validation score: 0.7963111738815182
Difference between R2 score and cross validation score is - 0.02136514209467455
```



- ➔ The R2 score using Linear Regression model is 81.76% and the cross validation is 79.63%.
- ➔ The best fit line shows there is strong linear relation between test data of trained model and predicted values.

Similarly applied all the algorithms, created different models and found out the R2 score, MAE, MSE, RSME and cross validation and the difference between cross validation and R2 score. And the best score is given by Random Forest Regressor model.

Random Forest Regressor:

A Random Forest regressor is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregate their individual predictions to form a final prediction. Created Random Forest Regressor model and getting 90.11% R2 score using this model. From the above plot we can observe the sales price of the house. The best fit line shows there is strong linear relation between test data of trained model and predicted values.

```

# Checking r2score for Random Forest Regressor
RFR = RandomForestRegressor()
RFR.fit(x_train,y_train)

# prediction
predRFR=RFR.predict(x_test)
print('R2_score:',r2_score(y_test,predRFR))

# Mean Absolute Error (MAE)
print('MAE:',mean_absolute_error(y_test, predRFR))

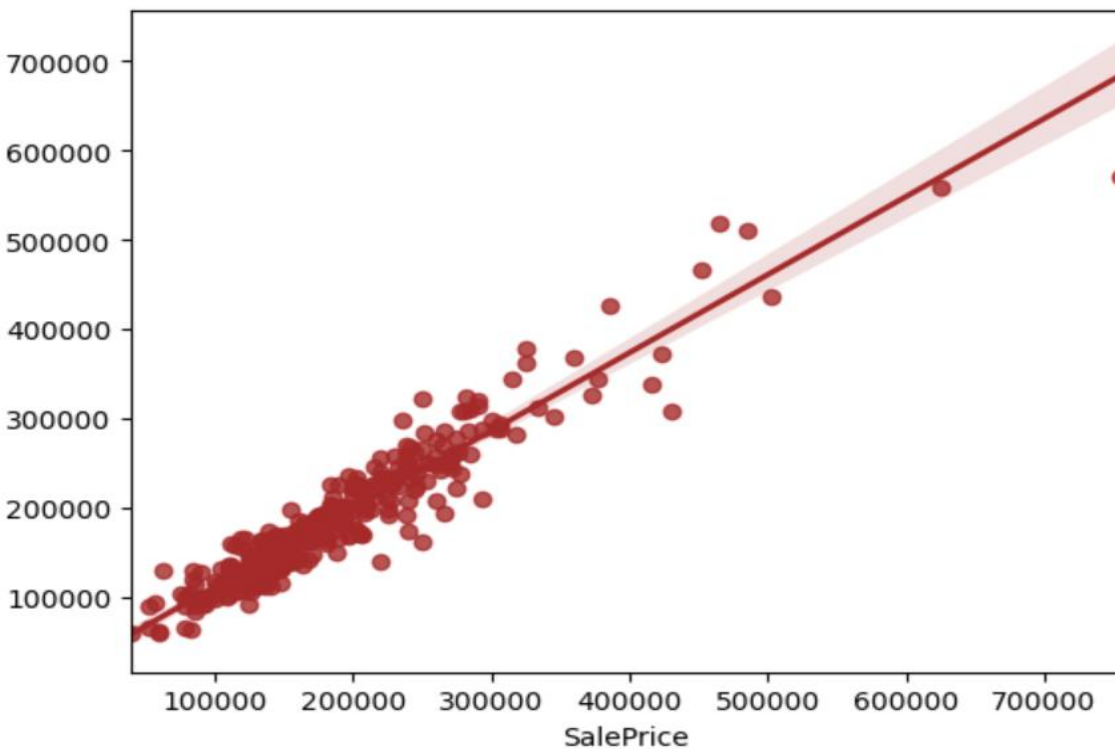
# Mean Squared Error (MSE)
print('MSE:',mean_squared_error(y_test, predRFR))

# Root Mean Squared Error (RMSE)
print("RMSE:",np.sqrt(mean_squared_error(y_test, predRFR)))

# Checking Cross Validation Score for Linear Regression
score=cross_val_score(RFR,train_df_x,y,cv=8)
print(score)
print("cross validation score: ",score.mean())
print("Difference between R2 score and cross validation score is - ",r2_score(y_test,predRFR)-abs(score.mean()))

# Visualizing the predicted values
sns.regplot(y_test,predRFR,color="brown")
plt.show()

```



➔ The R2 score using Random Forest Regressor model is 90.11% and the cross validation is 84.89%.

- ➔ The best fit line shows there is strong linear relation between test data of trained model and predicted values.

Model Selection:

From the difference between R2 score and Cross Validation score I can conclude that Random Forest Regressor as my best fitting model as it is giving less difference compare to other models. Let's perform Hyperparameter tuning to increase the model accuracy.

Hyper Parameter Tuning:

```
: # Lets use GridSearchCV to find the best parameters in Bagging Regressor
parameters = {'n_estimators': [10,50,100,200,500],
              'max_samples': [1.0,5.0,6.0,0.008],
              'max_features': [1.0,10.0,0.0001,5.68],
              'bootstrap': [True,False]}
```

```
: GCV=GridSearchCV(RandomForestRegressor(),parameters,cv=5)
```

```
: GCV.fit(x_train,y_train)
```

```
: GridSearchCV(cv=5, estimator=RandomForestRegressor(),
              param_grid={'bootstrap': [True, False],
                          'max_features': [1.0, 10.0, 0.0001, 5.68],
                          'max_samples': [1.0, 5.0, 6.0, 0.008],
                          'n_estimators': [10, 50, 100, 200, 500]})
```

```
: # Getting best parameters
GCV.best_params_
```

```
: {'bootstrap': True,
  'max_features': 1.0,
  'max_samples': 1.0,
  'n_estimators': 50}
```

I have used GridSearchCV to get the best parameters of Random Forest Regressor. And used all the obtained parameters to get the accuracy of final model.

Creating final model:

```

# Creating final model
Final_model = RandomForestRegressor(bootstrap=True, max_features=1.0, max_samples=1.0, n_estimators=50)

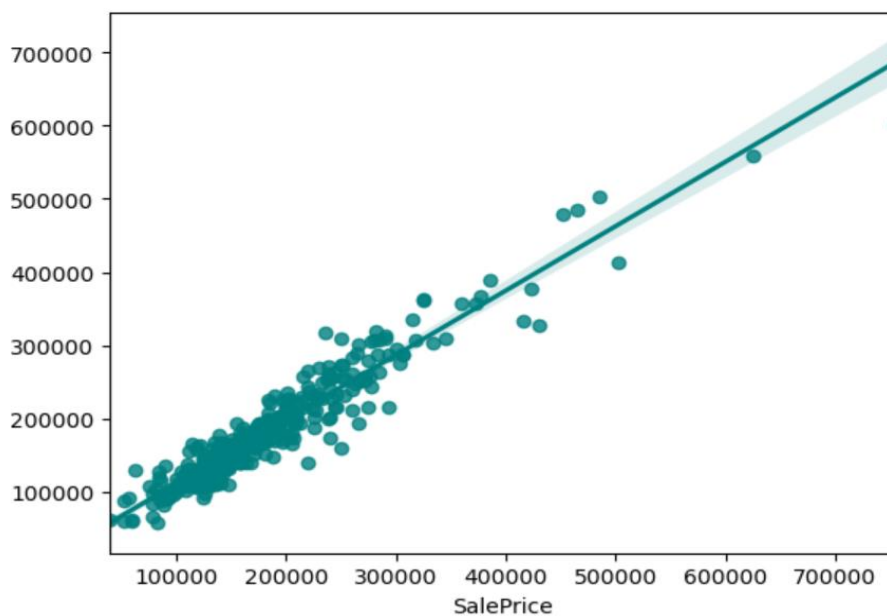
# Prediction
Final_model.fit(x_train, y_train)
pred = Final_model.predict(x_test)
print('R2_Score:', r2_score(y_test, pred)*100)

# Metric Evaluation
print('Mean absolute error:', mean_absolute_error(y_test, pred))
print('Mean squared error:', mean_squared_error(y_test, pred))
print('Root Mean Squared error:', np.sqrt(mean_squared_error(y_test, pred)))

# Visualizing the predicted values
sns.regplot(y_test, pred, color="teal")
plt.show()

```

R2_Score: 90.79657542800147
 Mean absolute error: 17566.27743589743
 Mean squared error: 629963526.6221732
 Root Mean Squared error: 25099.07421842832



The R2 score of Random Forest Regressor has been increased 0.68% after tuning the model. It is giving R2 score as 90.79% which is very good.

The plot gives some strong linear between test and predicted values. Also, I can notice the MAE, MSE and RMSE values have been reduced which means the model trained very well.

Saving the final model and predicting the saved model

- I have saved my final best model using joblib library in. pkl format

```
# Saving the model using .pkl
import joblib
joblib.dump(Final_model, "Housing_SalePrice_Prediction.pkl")

['Housing_SalePrice_Prediction.pkl']
```

- Loading saved model and predicting the sale price.

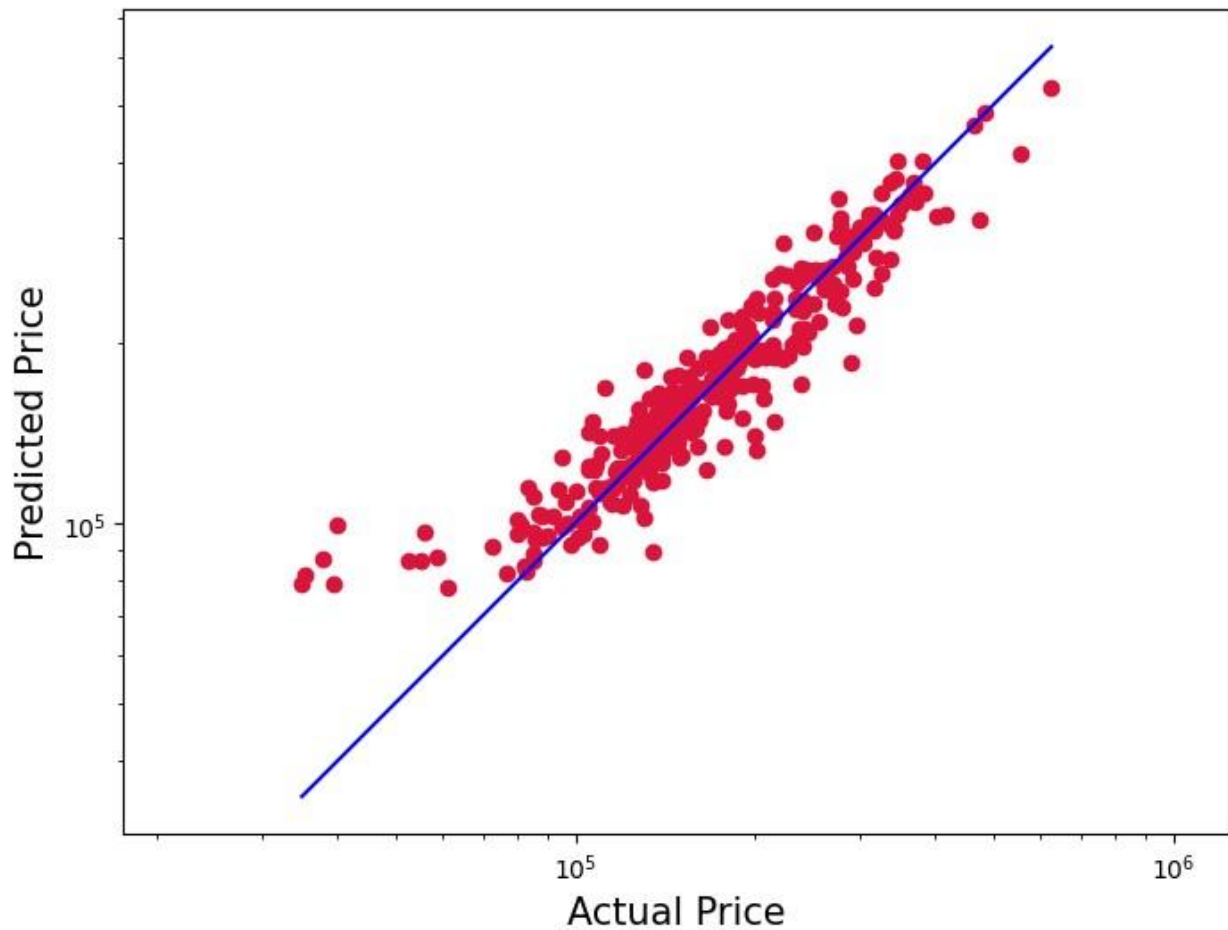
```
# Loading the saved model
Model = joblib.load("Housing_SalePrice_Prediction.pkl")

# prediction
a = np.array(y_test)
predicted = np.array(Model.predict(x_test))
df_com = pd.DataFrame({"Original":a,"Predicted":predicted},index= range(len(a)))
df_com
```

	Original	Predicted
0	120000	141260.00
1	140000	163286.06
2	172500	167131.10
3	244600	262039.00
4	88000	94338.00
5	252000	264438.38
6	176000	161883.00
7	124900	117605.34
8	120000	140426.26
9	87000	103672.00
10	37900	87083.52

Prediction Visualization

```
plt.figure(figsize=(8,6))
plt.scatter(y_test, predicted, c='crimson')
plt.yscale('log')
plt.xscale('log')
p1 = max(max(predicted), max(y_test))
p2 = min(min(predicted), min(y_test))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('Actual Price', fontsize=15)
plt.ylabel('Predicted Price', fontsize=15)
plt.axis('equal')
plt.show()
```

- The plot gives the linear relation between predicted and actual sale price of the house. The best fitting line gives the actual values and red dots gives the predicted values.

Predicting SalePrice of house for test dataset using saved trained model:

```
# Predicting the house sale price from the features of the testing data
Predicted_SalePrice = Model.predict(test_df)
Predicted_SalePrice
```

```
array([197967.84, 146300.2 , 187937.7 , 180716.84, 155618.24, 159995. ,
       157724.74, 183260.28, 155791.68, 154800. , 187595.3 , 193019.32,
       203013.04, 203218.74, 182614.72, 160031.04, 190973.44, 188034.44,
       150955. , 148058. , 157935. , 182829.64, 188334.44, 139016.16,
       187908.74, 194905.6 , 156613.04, 185204.08, 156540. , 160124.12,
       152909.04, 183331.92, 187249.7 , 185843.88, 161512.04, 152570.04,
       158394. , 153245. , 158405. , 153542. , 186392.18, 154457.04,
       153214.24, 189360.86, 151572.04, 154105.18, 190636.92, 205663.32,
       152507. , 189326.28, 158904.28, 187170.24, 168844.08, 157123.08,
       155548.24, 152984.94, 157579.08, 156165.04, 177189.58, 183562.44,
       157355. , 155285. , 199420.68, 150793. , 146134.26, 155432. ,
       189495.64, 185016.72, 194852.8 , 184894.44, 161004. , 187229.88,
       182368.32, 152279.2 , 152218.94, 155425.04, 154540.24, 146967. ,
       196214.8 , 189013.04, 154125.04, 156300.04, 154644. , 191934.48,
       153836.74, 153457.48, 162911.32, 158217.04, 155503.2 , 155773.04,
       184707.28, 148134.4 , 149410.08, 147454.4 , 158653.2 , 154084. ,
       203139.4 , 154275.74, 152535.04, 186587.44, 153553.04, 165454. ,
       197171.28, 155074.04, 157455.04, 155417.2 , 158911.04, 152932.44,
       154546.24, 160951.12, 183794.88, 159457.2 , 185014.48, 150670.74,
       156547.08, 158285.08, 187547.72, 153631.04, 190928.64, 192471.12,
       162636. , 153441. , 157221.74, 157042.04, 198039.32, 158390.74,
       189982.08, 185699.48, 189944.24, 158474. , 188277.08, 160473.48,
       154814. , 157634. , 181286.32, 181067.24, 150431. , 140581.2 ,
       153213.74, 164247.04, 179238.16, 182659.32, 156179.04, 181235.48,
       196187.8 , 155320.24, 169028. , 168161. , 186404.38, 155258.24,
```

These are the predicted sale price of the house for test data set.

Creating DataFrame and saving the predictions

```
#Creating DataFrame for the predicted results  
Prediction = pd.DataFrame()  
Prediction['SalePrice'] = Predicted_SalePrice  
Prediction
```

	SalePrice
0	197967.84
1	146300.20
2	187937.70
3	180716.84
4	155618.24
5	159995.00
6	157724.74
7	183260.28
8	155791.68
9	154800.00
10	187595.30
11	193019.32

I have predicted the SalePrice for test dataset using saved model of train dataset and getting good predictions. I have saved my predictions in csv format for further analysis.

3.5 Key Metrics for success in solving problem under consideration:

The essential step in any machine learning model is to evaluate the accuracy and determine the metrics error of the model. I have used the following metrics for my model evaluation:

Mean absolute error (MAE):

MAE is a popular error metric for regression problems which gives magnitude of absolute difference between actual and predicted values. The MAE can be calculated as follows:

The diagram illustrates the Mean Absolute Error (MAE) formula with the following components and annotations:

- MAE**: The metric being calculated.
- =**: The equals sign.
- $\frac{1}{N}$** : The fraction representing the division by the total number of data points. An arrow points from the text "Divide by total Number of Data Points" to this fraction.
- \sum** : The summation symbol.
- $|Y - \hat{Y}|$** : The absolute value of the difference between the actual output and the predicted output. An arrow points from the text "Actual Output" to Y , and another arrow points from the text "Predicted Output" to \hat{Y} . A yellow bracket underlines the expression $|Y - \hat{Y}|$, with an arrow pointing to the text "Absolute Value of residual".
- Sum Of**: A curved arrow points from this text to the summation symbol \sum .

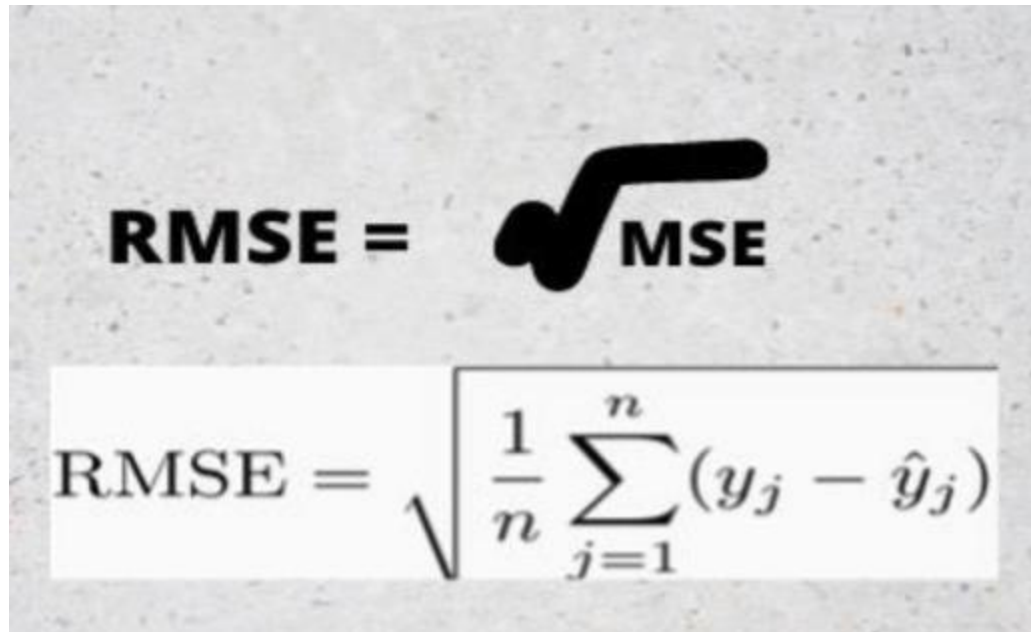
Mean Squared Error (MSE):

MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value. We perform squared to avoid the cancellation of negative terms and it is the benefit of MSE.

$$MSE = \frac{1}{n} \sum \left(\underbrace{y - \hat{y}}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}} \right)^2$$

Root Mean Squared Error (RMSE):

RMSE is an extension of the mean squared error. The square root of the error is calculated, which means that the units of the RMSE are the same as the original units of the target value that is being predicted.

The image shows a hand-drawn formula for RMSE on a piece of paper. At the top, it says 'RMSE = ' followed by a large, bold, hand-drawn square root symbol, and then 'MSE'. Below this, there is a white rectangular box containing the mathematical formula:
$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

RMSE = $\sqrt{\text{MSE}}$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

R2 Score: I have used R2 score which gives the accurate value for the models used. On the basis of R2 score I have created final model.

3.6 Interpretation of the results:

Visualizations:

- ➔ I have used distribution plot to visualize the target variable SalePrice, which was almost normally distributed. From the scatter plot we noticed most of the features like OverallQual, TotalRmsAbvGrd, FullBath, GarageCars etc had some strong linear relation with target as we observed as the quality or area increased, the sale price also tends to increase.
- ➔ The heat map and bar plot helped to understand the correlation between target and features. Also, with the help of heat map I found

multicollinearity problem and I have done feature selection to overcome with the issue. Detected outliers and skewness using box plots and distribution plots. And I found some of the features skewed to right. I got to know the count of each column using count plots and pie plots.

Pre-processing: The dataset should be cleaned and scaled to build the ML models to get good predictions. I have performed many processing steps which I have already mentioned in the pre-processing step.

Modelling: After cleaning and processing both train and test data, I performed train test split to build the model. I have built multiple regression models to get the accurate R2 score, and evaluation metrics. I got Random Forest regressor as best model which gives 90.11% R2 score. This is due to over-fitting, so I checked the cross-validation score. After tuning the best model Random Forest regressor I got 90.79% R2 score and even got minimum MAE, MSE and RMSE values. Less error means no over fitting. And finally, I saved my final model and got the good predictions results for test dataset.

4.CONCLUSION

4.1 Key Findings and Conclusions of the Study

In this study, we have used multiple machine learning models to predict the house sale price. We have gone through the data analysis by performing feature engineering, finding the relation between features and label through

visualizations. And got the important feature and we used these features to predict the price by building ML models. We have got good prediction results. After using hyper parameter tuning, the best model increased by .68% and the R2 score was 90.79% also the errors decreased which means no over-fitting issue.

Findings:

Which variables are important to predict the price of variable?

Overall Quality is the most contributing and highest positive impacting feature for prediction. Also, the features like GarageArea, LotArea, 1stFlrSF, TotalBsmtSF etc have somewhat linear relation with the price variable.

How do these variables describe the price of the house?

The houses which have very excellent overall quality like material and finish of the house have high sale price. Also, we have observed from the plot that as the overall quality of the house increases, the sale price also increases. That is there is good linear relation between SalePrice and OverallQual. So, if the seller builds the house according to these types of qualities that will increase the sale price of the house.

There is a linear relation between the SalePrice and 1stFlrSF. As we have seen as the 1st floor area increases, sales price also increases moderately. So, people like to live in the houses which have only 1-2 floors and the cost of the house also increases in this case.

Also, we have seen the positive linear relation between the SalePrice and GarageArea. As size of garage area increases, sale price also increases.

There is positive linear relation between sale price and TotalBsmtSF. As total basement area increases, sale price also increases.

4.2 Learning Outcomes of the Study in respect of Data Science

While working on this project I learned more things about the housing market and how the machine learning models have helped to predict the price of house which indeed helps the sellers and buyers to understand the future price of the house. I found that the project was quite interesting as the dataset contains several types of data. I used several types of plotting to visualize the relation between target and features. This graphical representation helped me to understand which features are important and how these features describe the sale price. Data cleaning was one of the important and crucial things in this project where I replaced all the null values with imputation methods and dealt with features having zero values and time variables.

Finally, our aim is achieved by predicting the house price for the test data, I hope this will be further helps for sellers and buyers to understand the house marketing. The machine learning models and data analytic techniques will have an important role to play in this type of problems. It helps the customers to know the future price of the houses.

4.3 Limitations of this work and scope for future work

Limitations:

In case of processing train and test dataset, I felt concatenation is not suitable as it causes data leakage. The dataset contains some irrelevant columns, zero values, null values, so it is need to increase the dataset size by filling these values.

The dataset has many limitations, the main limitation is that we have no information potential buyers and environment of the sale. The

factors such as auctions can have an influence on the price of the house.

The dataset does not capture many economic factors. Collecting more accurate and important details about the houses from the buyers will help to analyse the data more clearly.

Future work:

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.

As a recommendation, I advise to use this model by the people who want to buy a house in the area covered by the dataset to have an idea about the actual price. The model can be used also with datasets that cover different cities and areas provided that they contain the same features. I also suggest that people take into consideration the features that were deemed as most important as seen in this study might help them estimate the house price better.

Thank You