HOUSING: PRICE PREDICTION

Submitted by:

LAVINA VAIDYA

ACKNOWLEDGMENT

I would like to thank our project mentor for their valuable help, support and guidance throughout this project and the necessary resources and support they provided us. I am truly grateful to all the individuals who helped us in completing this project successfully and in a timely manner. I would like to thank all the online resources, blogs and tutorials that helped us gain a better understanding of the concepts and technologies used in the project. I am thankful to our friends and family for their encouragement and support.

INTRODUCTION

Housing prices are the amount of money that a buyer pays for a housing unit such as a house, condo, or apartment. Housing prices are influenced by a variety of factors such as location, size, amenities, and quality of construction. In addition, factors such as the local economy, interest rates, and availability of housing can also affect housing prices. Housing prices can also be affected by national and global economic conditions. Housing prices also tend to increase over time, as demand for housing increases and supply remains limited.

- ## Business Problem Framing

  The housing price problem is a problem that deals with predicting the prices of houses in a certain area. This problem can be related to the real world because it is a problem that affects everyone. People need to know the prices of houses in order to make informed decisions about whether to buy or rent a home. In addition, this problem is important for businesses and investors who are looking to buy and sell homes in the area. Understanding the local housing market is essential for making profitable investments. Overall, the housing price problem helps people make better decisions about their real estate investments.

- ## Conceptual Background of the Domain Problem

  1. Appreciation: The increase in the value of a house over time due to market forces such as inflation, changes in the local economy, or the quality of the neighbourhood.
  2. Depreciation: The decrease in the value of a house over time due to market forces such as inflation, changes in the local economy, or the quality of the neighbourhood.
  3. Equity: The difference between the current market value of a house and any outstanding debts owed on the property.
  4. Location: The geographic area in which a house is located, which can have a significant impact on its market value.

5. Supply and demand: The economic forces that drive the price of a house and the volume of houses sold in a given area.

6. Property taxes: Taxes that are assessed on a house based on its market value, which can have an impact on a seller or buyer's decision to purchase a home.

7. Mortgage rates: The interest rate charged on a loan taken out to purchase a house, which can have an impact on the amount of money a buyer is able to borrow and ultimately the price of the house. 8. Property condition: The physical condition of a house, including the quality

8. Supply and Demand: The amount of available homes on the market and the amount of people looking to buy them will affect the price of a house. When the demand is higher than the supply, prices tend to rise.

- **Business Goal**

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

- **Motivation for the Problem Undertaken**

  The objective behind making a house price prediction project is to provide an accurate and reliable prediction of the value of a given house based on data about the house and its surroundings.

  This project is motivated by the need to accurately price houses for sale, rental prices and mortgages. By predicting the house price, real estate agents, buyers, sellers, and lenders are able to make better decisions on investments and transactions. Additionally, the project can provide a better understanding of the factors that affect the value of a particular house.

ANALYTICAL PROBLEM FRAMING

Mathematical/Analytical Modeling of the Problem:

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

There are two datasets. One is train dataset which is supervised and another one is test dataset which is unsupervised. The target variable is "SalePrice" 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.

Firstly, I have to treat test dataset. I checked for null values, in some columns null values are more than 50% so we drop them, similarly in some columns zero values are greater than 70% I dropped then also. I treated train dataset for null values and fill them but before that I differentiate the data frame in categorical and numerical column then numerical columns null value fill with mean and categorical column fills with mode value

After treating null values and zero values I dropped some columns which find to irrelevant relation to our target variable, then I am going for visualization part to fling relation with target variable(bivariant visualization)

Now its time to find skewness treated them and create a new data frame without outliers.

After finishing this I go for regration model find best fil model and make predication for given test data.

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: • Train dataset • Test dataset ✓ 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 (SalePrice). ✓ 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 nominal and ordinal variables.

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, 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, GarageYrBit and YrSold. Converting them into age seem more meaningful as they offer more information about the longevity of the features. 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.

➢ Used correlation coefficient to check the correlation between label and features.

➢ While checking the correlation I came across multicollinearity problem, I checked VIF values and removed GrLivArea to overcome with the multicollinearity issue.

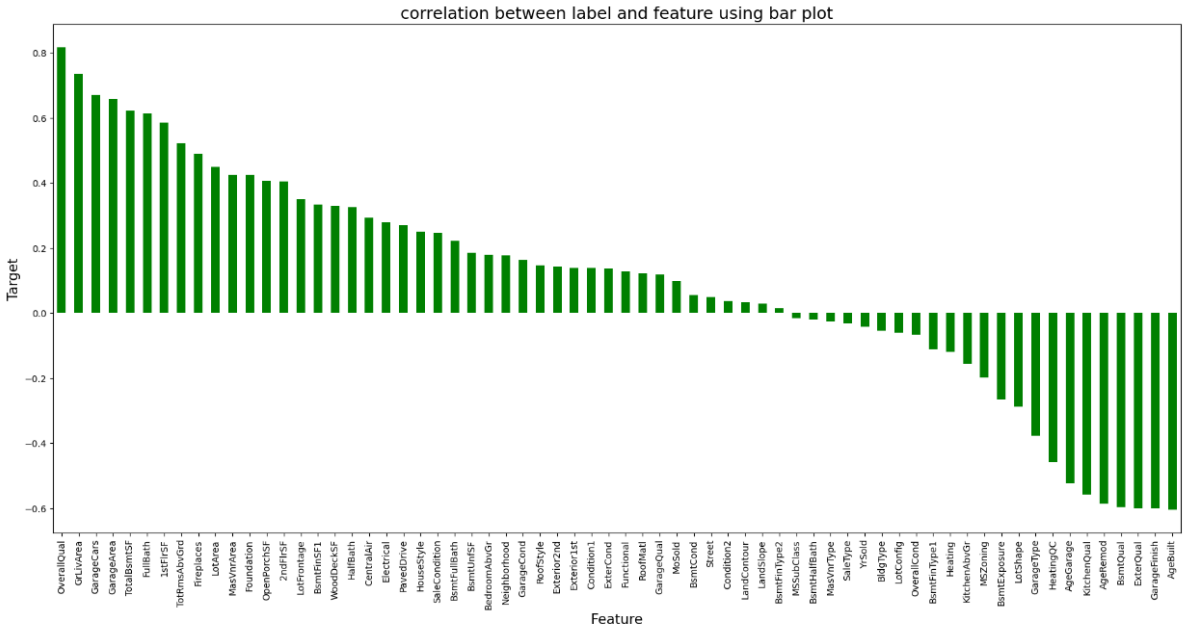
➢ Scaled train datasets using Standard Scalar method and used regression algorithms to build ML models.

➢ All these steps were performed to train datasets.

Data Inputs- Logic- Output Relationships

To find out relationship between data inputs to target variable I done various visual analysis such as count plot, dist plot, reg plot etc. after that I checked the corelation because of that I know which feature variable is positively corelated and which one is negatively corelated.

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 SalePrice. Below are the correlated features. Important features that affect SalePrice positively and negatively Features having high Positive correlation with label

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

Features having high Negative correlation with label

- Heating

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

Hardware and Software Requirements and 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 Python3

Library used

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

```
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
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 Lasso, Ridge
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.metrics import classification_report
from sklearn import metrics
```


Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  I have used imputation methods to treat the null values..
- Removed skewness using power transformation method.
- Encoded the object type data into numerical using Label Encoder.
- I have used 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
- Testing of Identified Approaches (Algorithms)
  I used following algorithms on train dataset
  - LinearRegression
  - RandomForestRegressor
  - ExtraTreesRegressor

- LassoRegressor, RidgeRegressor
- GradientBoostingRegressor
- BaggingRegressor

RUN AND EVALUATE SELECTED MODELS

Linear regression

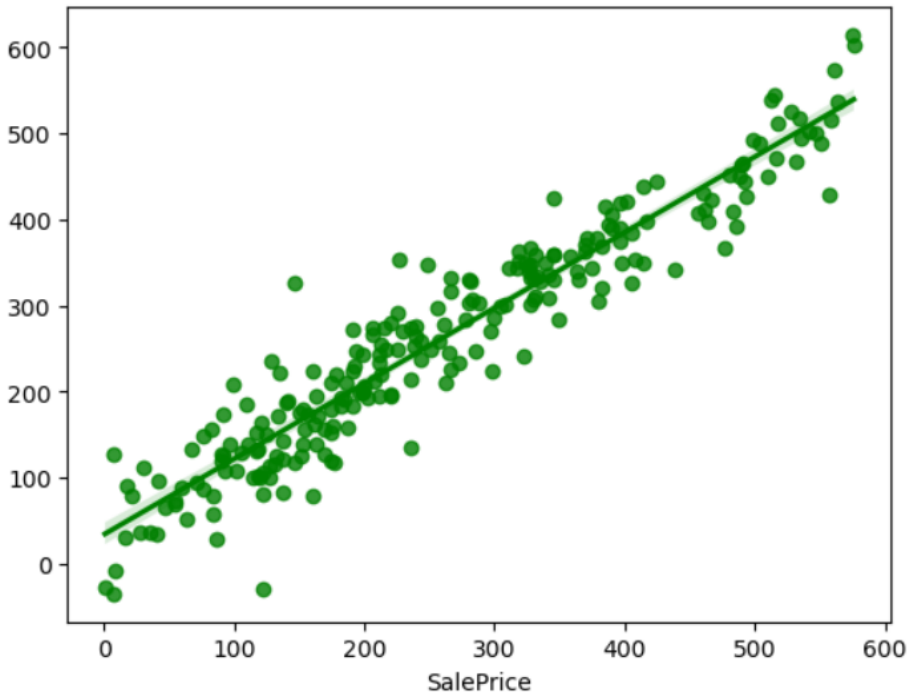
Linear regression is a type of supervised machine learning algorithm that is used to predict a continuous numerical target variable given a set of independent features. I used for house price prediction by using features to predict the value of the house. The linear regression model can then be used to make predictions about the future price of the house given the current set of features.

In [68]:
```
# checking r2 score for linear regression
lr= LinearRegression()
lr.fit(x_train,y_train)

# prediction
pred_test=lr.predict(x_test)
print('R2_score:',r2_score(y_test,pred_test))

# mean Absolute error (MAE)
print('Mean Absolute Error:',mean_absolute_error(y_test, pred_test))

# mean Squared error (MSE)
print('Mean Squared Error:',mean_squared_error(y_test, pred_test))

# root mean Squared error (RMSE)
print("Root Mean Squared Error:",np.sqrt(mean_squared_error(y_test, pred_test)))
```

```
R2_score: 0.9040382002279451
Mean Absolute Error: 34.71576192525394
Mean Squared Error: 2063.018031464253
Root Mean Squared Error: 45.42045829209843
```

In [72]:
```
# checking cross validation score for Linear Regression
print("Cross Validation Score:",cross_val_score(lr,x,y,cv=6).mean())

# visualizing the predicted values
sns.regplot(y_test,pred_test,color="g")
plt.show()
```

```
Cross Validation Score: 0.8822510368175775
```



Created linear regression model and getting 0.8976 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.

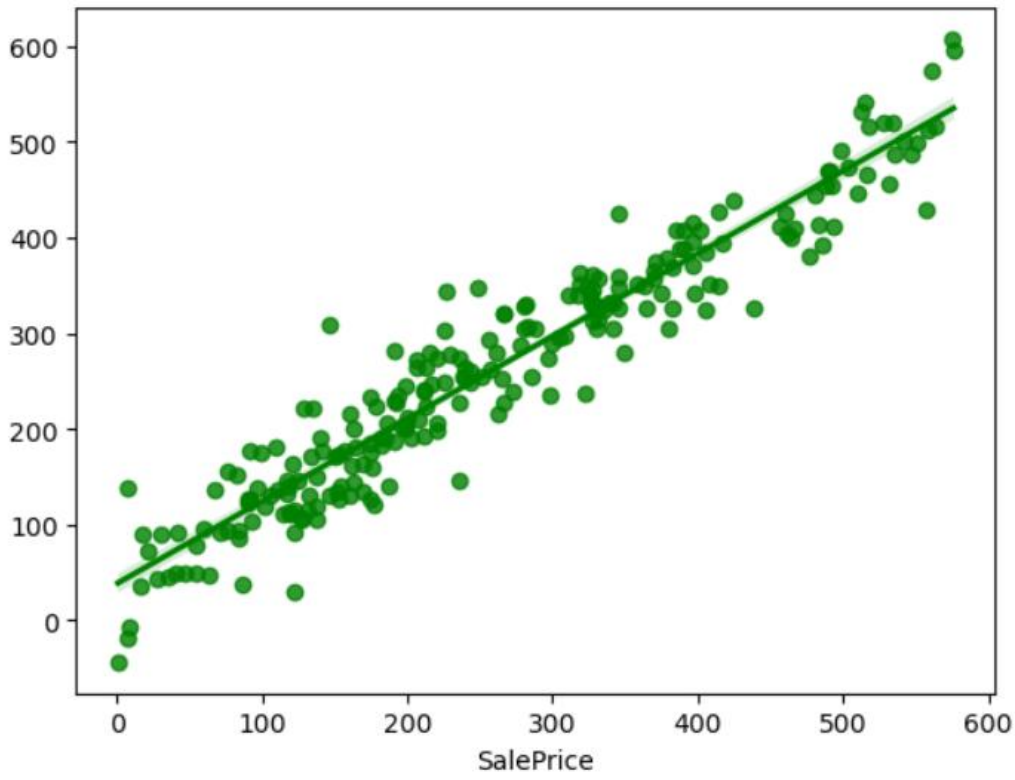
REGULARIZATION

Lasso Regressor

The Lasso Regressor is a linear model that uses an L1 penalty, which is also known as the “Lasso” penalty. It is an alternative to the ridge regression and is used when there are multiple predictors with multicollinearity, as it can perform feature selection. The Lasso Regressor reduces the magnitude of the coefficients of the less important features, which can help reduce overfitting and improve generalization. It is also useful for selecting important features from a dataset.

In [75]:
```
# checking r2score for Lasso Regressor
laso=Lasso(alpha=1)

# predictions
lasso.fit(x_train,y_train)
lasso.score(x_train,y_train)
pred_lasso=lasso.predict(x_test)

print('R2_Score:',r2_score(y_test,pred_lasso))
print('MAE:',metrics.mean_absolute_error(y_test, pred_lasso))
print('MSE:',metrics.mean_squared_error(y_test, pred_lasso))
print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test, pred_lasso)))

# checking cv score for lasso regression
print("Cross Validation Score:",cross_val_score(lasso,x,y,cv=6).mean())

# Visualizing the predicted values
sns.regplot(y_test,pred_lasso,color="g")
plt.show()
```

```
R2_Score: 0.9120824423614655
MAE: 33.30703533074402
MSE: 1890.080293631725
RMSE: 43.47505369325868
Cross Validation Score: 0.8863906780656152
```



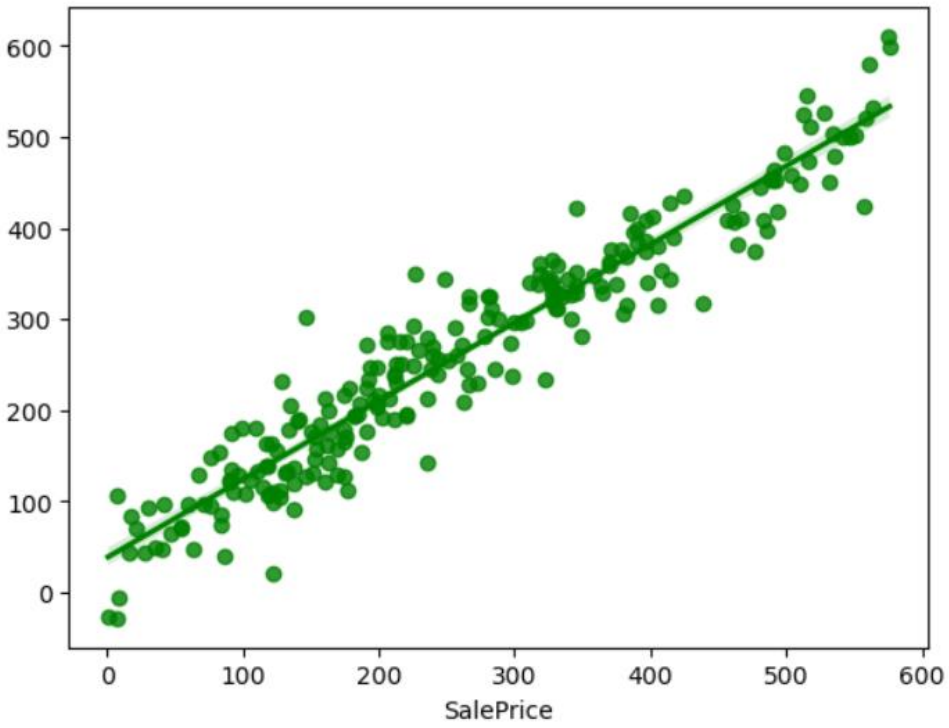
Created Lasso regressor model and getting 0.90 R2 score using this model. From the above plot we can observe the sales price of the house. The best fit line shows there is strong linear relation between test data of trained model and predicted values.

Ridge Regressor

Ridge regression is a type of regularized linear regression that adds an additional penalty term to the loss function. This penalty term is a L2 regularization, which penalizes weights with large magnitudes by adding a term to the loss function. The penalty term is weighted by a regularization parameter, which determines the strength of the regularization and can be adjusted to achieve different results. By adding this penalty term, ridge regression can help avoid overfitting, which is when a model performs well on the training data but poorly on new data.

In [77]:
```
# checking r2score for Ridge Regressor
ridge=Ridge(alpha=100)

# predictions
ridge.fit(x_train,y_train)
ridge.score(x_train,y_train)
pred_ridge=ridge.predict(x_test)

print('R2_Score:',r2_score(y_test,pred_ridge))
print('MAE:',metrics.mean_absolute_error(y_test, pred_ridge))
print('MSE:',metrics.mean_squared_error(y_test, pred_ridge))
print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test, pred_ridge)))

# checking cv score for Ridge regression
print("Cross Validation Score:",cross_val_score(ridge,x,y,cv=6).mean())

# Visualizing the predicted values
sns.regplot(y_test,pred_ridge,color="g")
plt.show()
```

```
R2_Score: 0.911063347112672
MAE: 33.966002664815356
MSE: 1911.9891352649015
RMSE: 43.72629798262027
Cross Validation Score: 0.8847662924755207
```



Created Ridge regressor model and getting 0.9059 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.

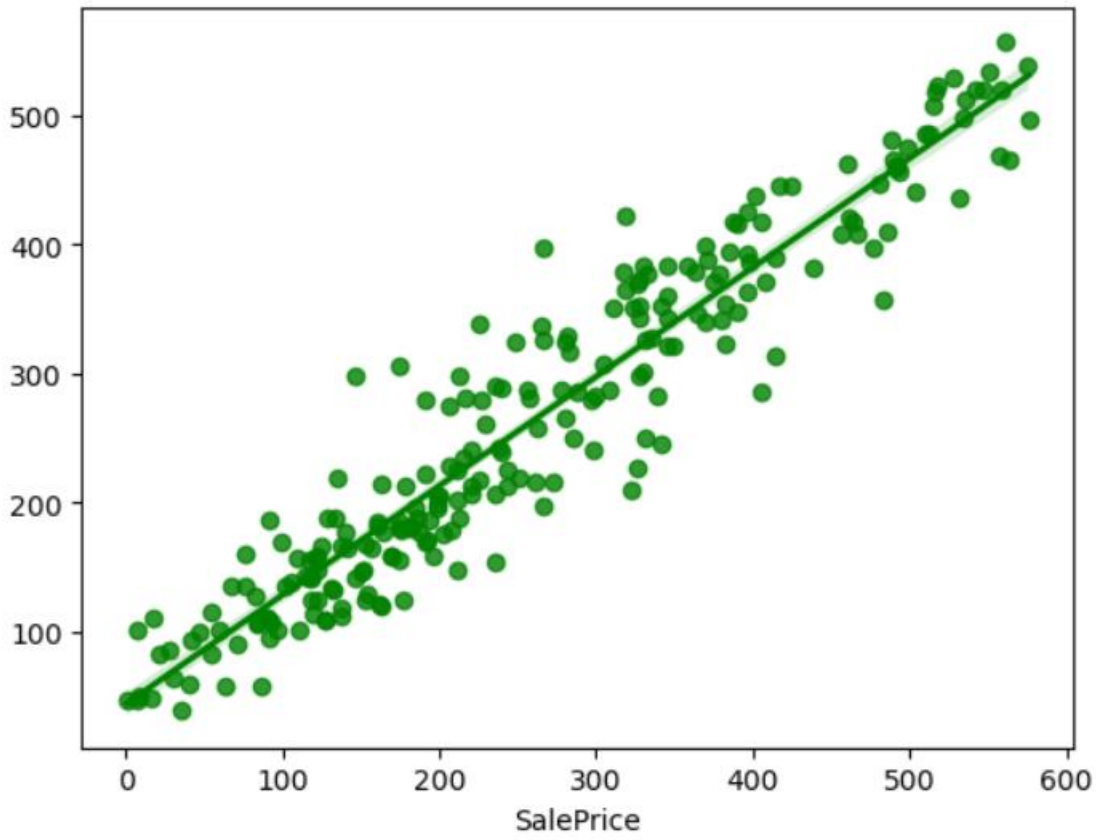
ENSEMBLE TECHNIQUES

Random Forest Regressor

The Random Forest Regressor is a type of supervised machine learning algorithm used for regression analysis. It works by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. The Random Forest Regressor works by randomly selecting a subset of data points and variables at each node and using them to create a decision tree. It then takes the average of the predictions from each tree to make a final prediction. This method makes it highly effective for dealing with high-dimensional datasets. It also reduces overfitting as the trees are built independently and thus are less likely to overfit the data.

```
]: # checking r2score for Random Forest Regressor
   RFR=RandomForestRegressor()
   RFR.fit(x_train,y_train)

   # predictions
   predRFR=RFR.predict(x_test)
   print('R2_Score:',r2_score(y_test,predRFR))

   # metric evaluation
   print('MAE:',metrics.mean_absolute_error(y_test, predRFR))
   print('MSE:',metrics.mean_squared_error(y_test, predRFR))
   print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test, predRFR)))

   # checking cv score for Random Forest regression
   print("Cross Validation Score:",cross_val_score(RFR,x,y,cv=6).mean())

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

```
R2_Score: 0.9016794717396346
MAE: 35.38876068376069
MSE: 2113.726744871795
RMSE: 45.97528406515608
Cross Validation Score: 0.878833534204467
```



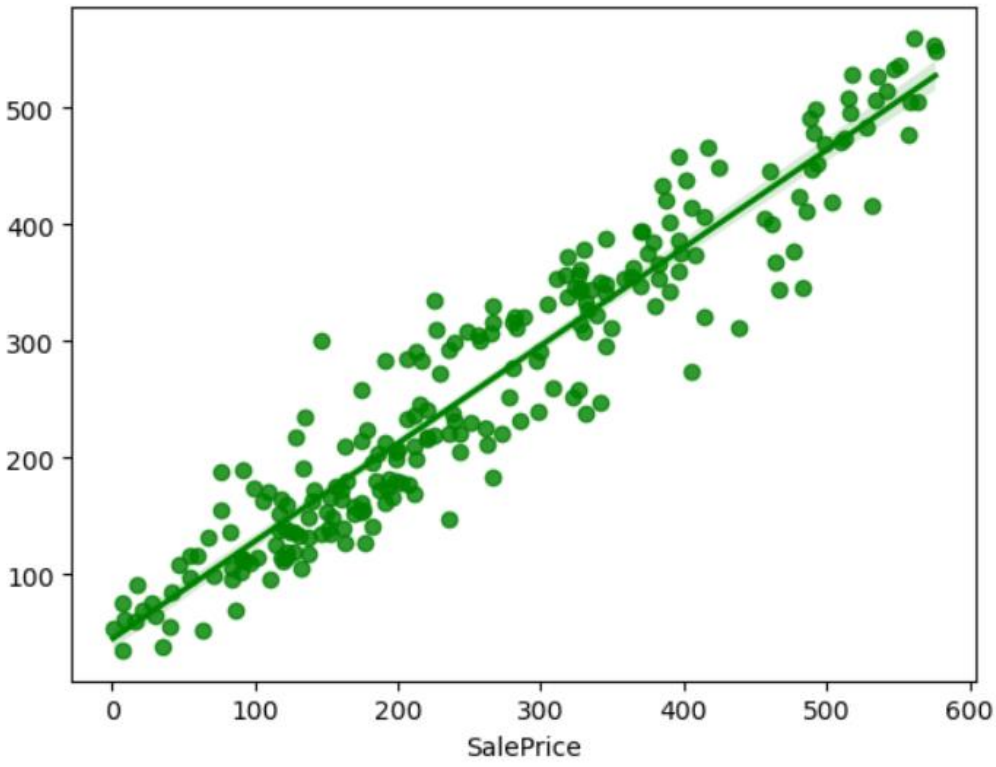
Created Random Forest Regressor model and getting 0.9013 score using this model. From the above plot we can observe the sales price of the house. The best fit line shows there is strong linear relation between test data of trained model and predicted values.

Extra Trees Regressor

The Extremely Randomized Trees Regressor is an ensemble learning algorithm that belongs to the forest of randomized trees. It is an extension of the Random Forest algorithm which uses random splits instead of the best possible split and random thresholds instead of the best possible ones. The algorithm is designed to improve the accuracy of the predictions and reduce the variance of the model. The algorithm works by constructing a number of decision trees on different samples of the data, with each tree being trained on a different subset of the data. The predictions of the trees are then combined and averaged to produce the final prediction. The algorithm can be used for both regression and classification problems.

In [79]:
```
# checking r2score for Extra Trees Regressor
XT=ExtraTreesRegressor()
XT.fit(x_train,y_train)

# predictions
predXT=XT.predict(x_test)
print('R2_Score:',r2_score(y_test,predXT))

# metric evaluation
print('MAE:',metrics.mean_absolute_error(y_test, predXT))
print('MSE:',metrics.mean_squared_error(y_test, predXT))
print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test, predXT)))

# checking cv score for Extra Trees regression
print("Cross Validation Score:",cross_val_score(XT,x,y,cv=6).mean())

# Visualizing the predicted values
sns.regplot(y_test,predXT,color="g")
plt.show()
```

```
R2_Score: 0.8990774680035738
MAE: 35.822606837606834
MSE: 2169.6654688034187
RMSE: 46.57966797652618
Cross Validation Score: 0.8878029398077452
```



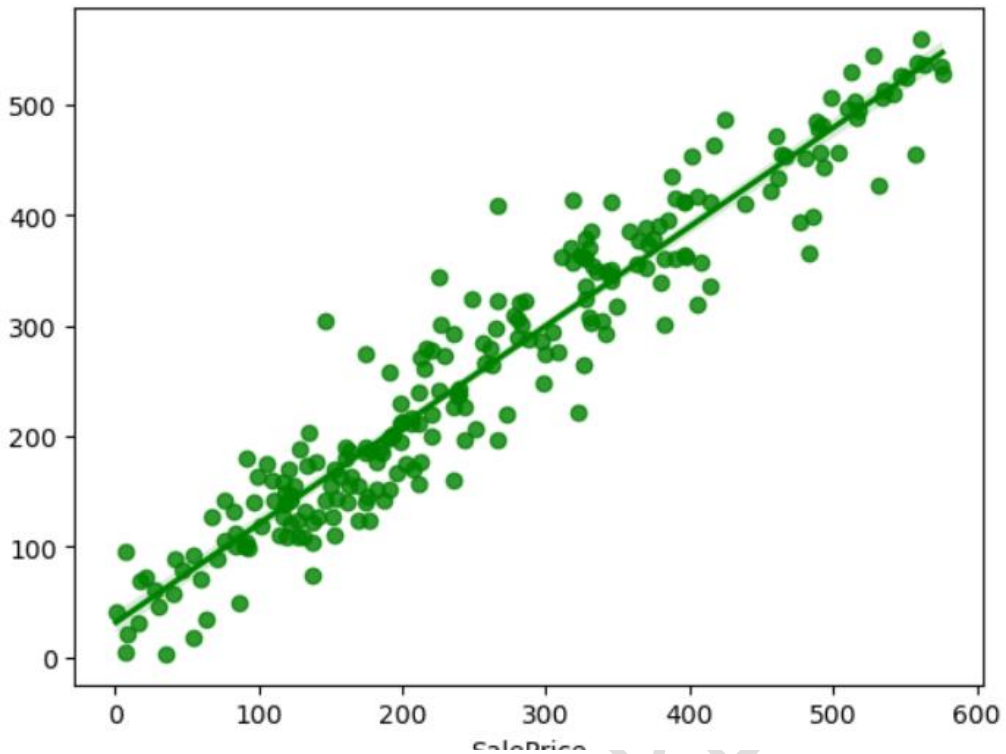
Created Extra Trees Regressor model and getting 0.8878 score using this model. From the above plot we can observe the sales price of the house. The best fit line shows there is strong linear relation between test data of trained model and predicted values.

Gradient Boosting Regressor

Gradient Boosting Regressor is a supervised learning algorithm that can be used in regression and classification problems. It is an ensemble method which combines multiple weak learners (usually decision trees) to create a powerful predictive model. The weak learners are called "base learners". It is iterative in nature and works by gradually improving the predictions of the model by adding weak learners one at a time. The process is repeated until a certain number of weak learners have been added or a certain accuracy has been achieved. It is a powerful algorithm that can produce highly accurate predictions. It is also quite robust to outliers and can handle non-linear relationships between features and target variables.

In [80]:
```
# checking r2score for GradientBoosting Regressor
GB=GradientBoostingRegressor()
GB.fit(x_train,y_train)

# predictions
predGB=GB.predict(x_test)
print('R2_Score:',r2_score(y_test,predGB))

# metric evaluation
print('MAE:',metrics.mean_absolute_error(y_test, predGB))
print('MSE:',metrics.mean_squared_error(y_test, predGB))
print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test, predGB)))

# checking cv score for Gradient Boosting regression
print("Cross Validation Score:",cross_val_score(GB,x,y,cv=6).mean())

# Visualizing the predicted values
sns.regplot(y_test,predGB,color="g")
plt.show()
```

```
R2_Score: 0.9177174985002392
MAE: 32.38584491004318
MSE: 1768.936021116854
RMSE: 42.05872110652978
Cross Validation Score: 0.901436319594861
```



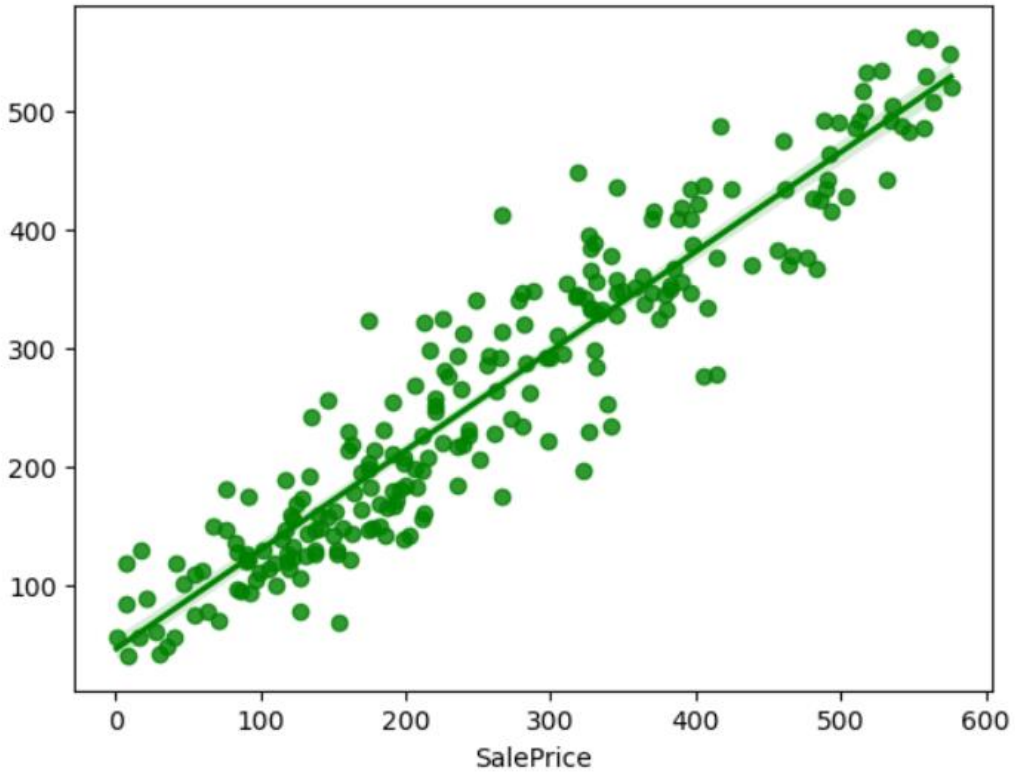
Created gradient boosting Regressor model and getting 0.9177 score using this model. From the above plot we can observe the sales price of the house. The best fit line shows there is strong linear relation between test data of trained model and predicted values.

Bagging Regressor

Bagging Regressor is an ensemble learning technique used in Machine Learning. It is a type of ensemble meta-estimator that fits multiple base models such as decision trees, linear regression, and neural networks on various sub-samples of the dataset and then aggregates their individual predictions to form a final prediction. The base estimators are built in parallel, each on its own sub-sample of the data. Bagging Regressor is an instance of bootstrapping technique, which is a general purpose procedure used to reduce the variance in a predictive model. It is also known as Bootstrap Aggregating.

In [81]:
```
# checking r2score for Bagging Regressor
BR=BaggingRegressor()
BR.fit(x_train,y_train)

# predictions
predBR=BR.predict(x_test)
print('R2_Score:',r2_score(y_test,predBR))

# metric evaluation
print('MAE:',metrics.mean_absolute_error(y_test, predBR))
print('MSE:',metrics.mean_squared_error(y_test, predBR))
print("RMSE:",np.sqrt(metrics.mean_squared_error(y_test, predBR)))

# checking cv score for Bagging regression
print("Cross Validation Score:",cross_val_score(BR,x,y,cv=6).mean())

# Visualizing the predicted values
sns.regplot(y_test,predBR,color="g")
plt.show()
```

```
R2_Score: 0.8797921398818128
MAE: 39.46709401709401
MSE: 2584.2677350427352
RMSE: 50.83569351393502
Cross Validation Score: 0.8642978301373031
```



Created bagging Regressor model and getting 0.8797 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.

Difference between R2 score and Cross Validation Score

Linear Regressor = 1.70%

Lasso Regressor = 2.24%

Ridge Regressor = 2.26%

Random Forest Regressor = 1.82%

Extra Trees Regressor = 1.45%

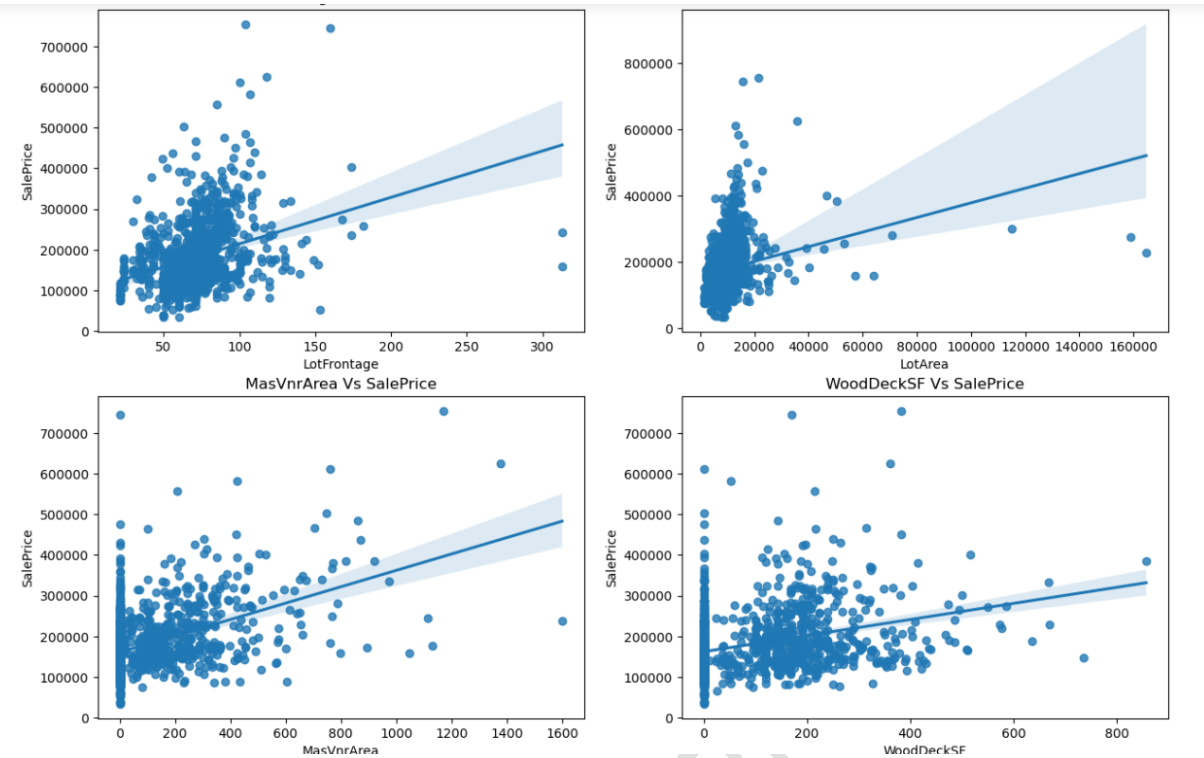
Gradient Boosting Regressor = 1.91%

Bagging regressor=2.64%

From the difference between R2 score and Cross validation score I can conclude that Linear Regressor as my best fitting model as it is giving less difference compare to other models.

VISUALIZATIONS

I have analysed 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.



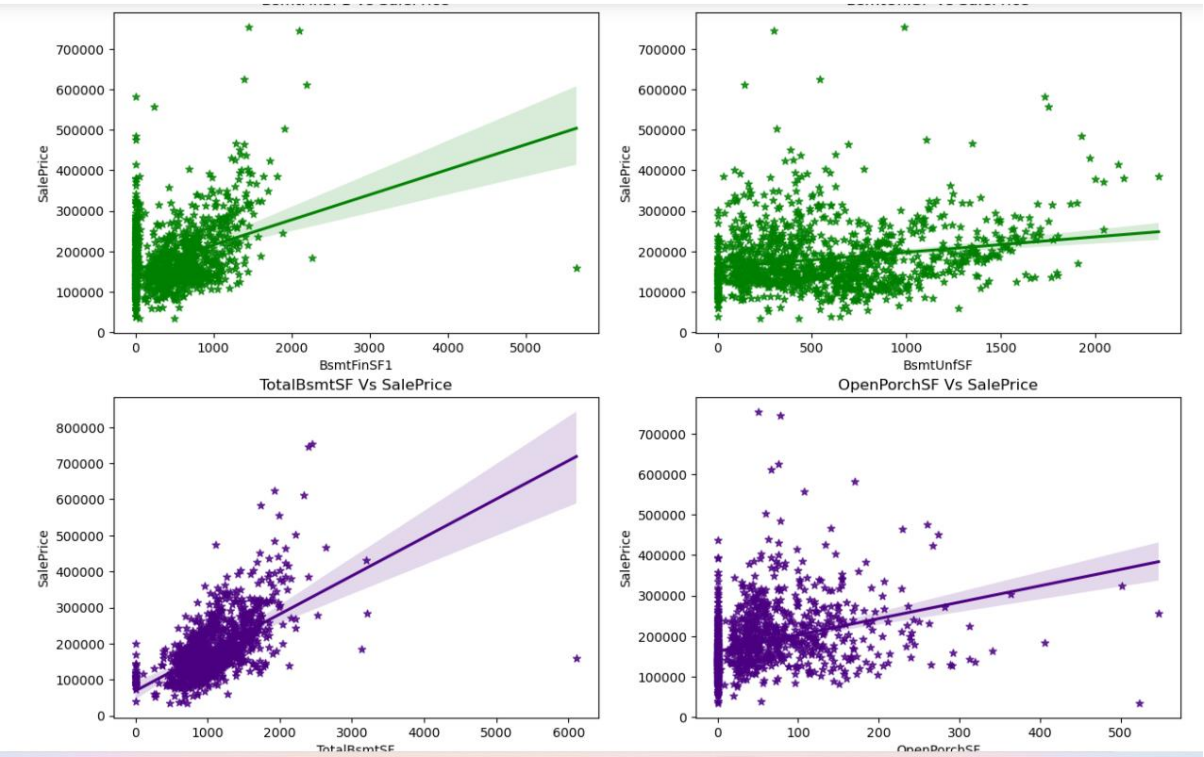
Observations:

• 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 increases.



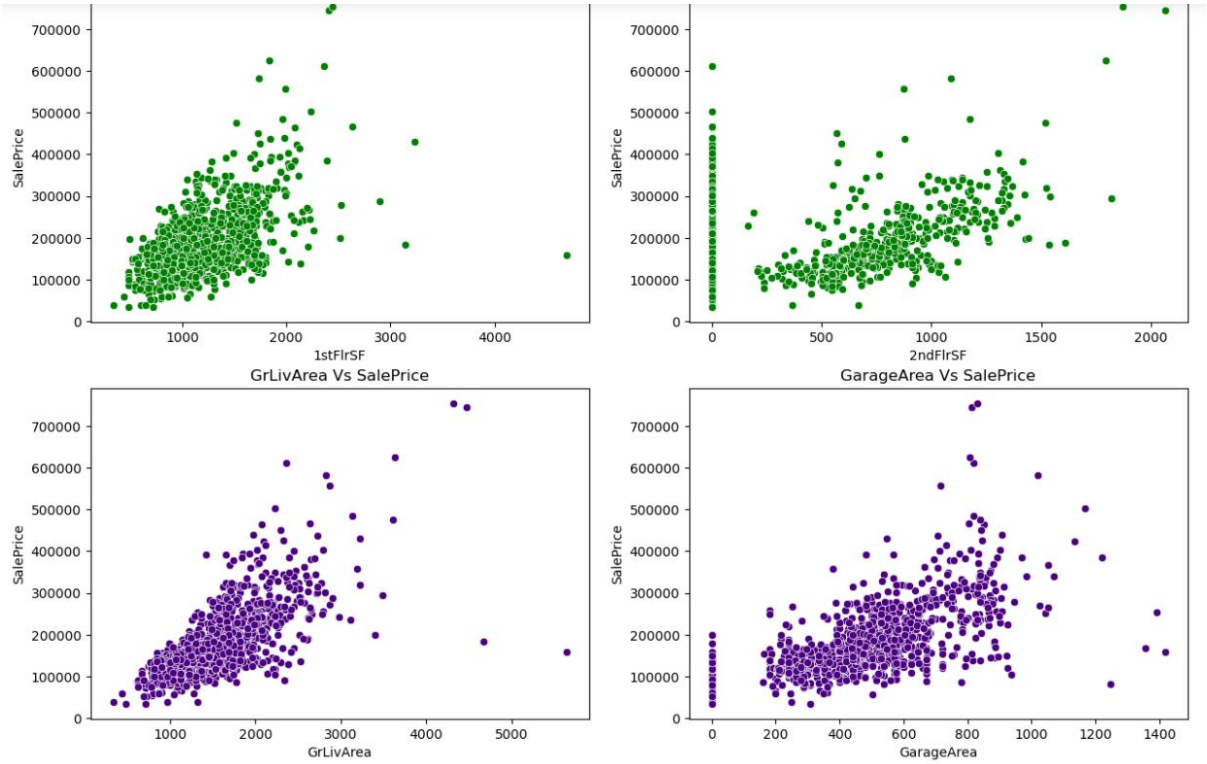
Observations:

• 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 is 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 is also increases.



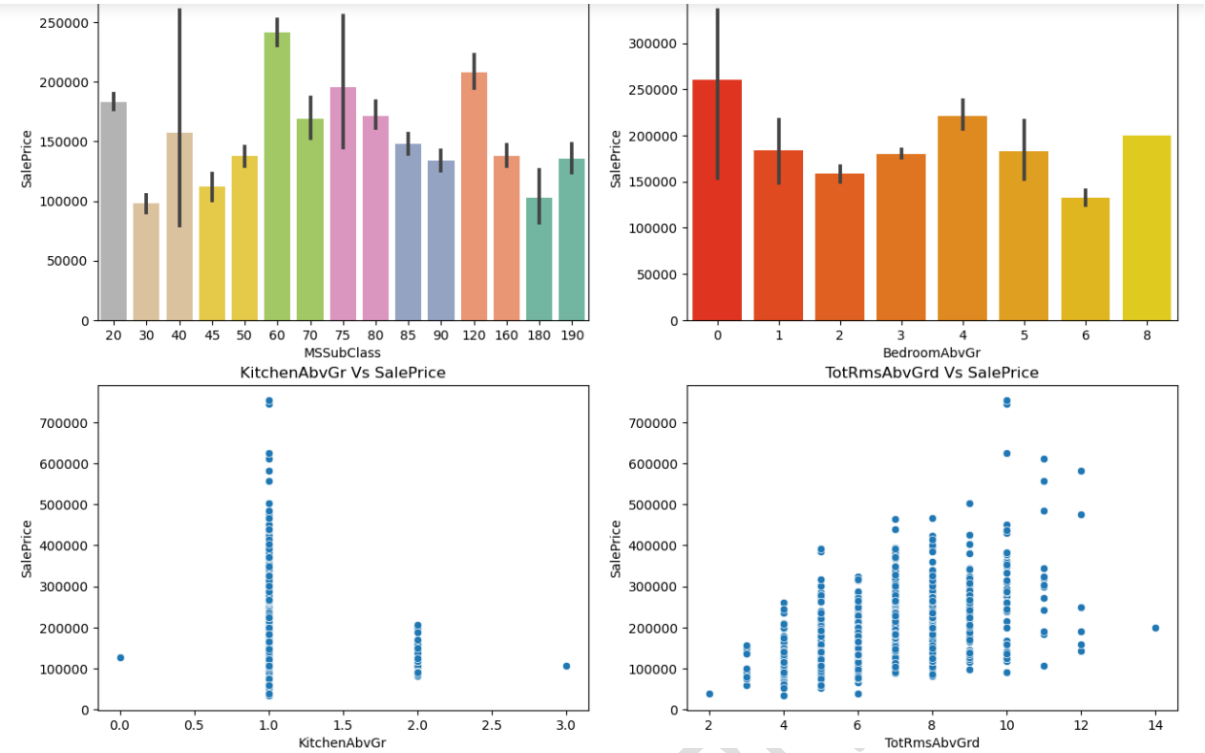
Observation:

• SalePrice vs 1stFlrSF: There is a linear relation between the label and feature. As we can observein 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 floorSF, 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.



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 sale price also increases.

Key Metrics for success in solving problem under consideration 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 linear regressor as best model which gives 89.76% R2 score. This is due to over-fitting , so I checked the cross-validation score. And finally, I saved my final model and got the good predictions results for test dataset. CONCLUSION 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.

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. 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.

LIMITATIONS OF THIS WORK AND SCOPE FOR FUTURE WORK LIMITATIONS:

- The house price prediction project is limited in its ability to predict future house prices.

- The data used to train the model is only a snapshot of the current housing market and does not take into account factors such as economic conditions, population growth, and other external factors that may affect house prices.
- The accuracy of the model is also limited due to the limited amount of data available. The model may not accurately predict house prices in different regions or markets due to the lack of data.
- The model is limited in its ability to explain the factors that contribute to house price movements. As the model is only a prediction tool, it does not provide insight into the underlying reasons for the price changes.

FUTURE WORK:

- Incorporate external factors such as economic conditions, population growth, local infrastructure, and other external factors into the model to improve accuracy.

- Utilize more advanced machine learning techniques such as deep learning to improve the accuracy of the model.
- Increase the amount of data available to train the model, such as historical data from different markets.
- Explore the use of advanced algorithms such as reinforcement learning for the prediction of house prices.
- 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.

References:

- https://www.academia.edu/38358601/House Price Prediction
- https://www.geeksforgeeks.org/house-price-prediction-using-machine-learning-in-python/

- https://www.researchgate.net/publication/342302491Housing Market Prediction Problem using Different Machine Learning Algorithms A Case Study
- https://scholarworks.sjsu.edu/cgi/viwecontent.cgi?article=1540&context=etd projects
- https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques