Housing Price Prediction Project

Submitted by:

P.V.RAGHAVULU

ACKNOWLEDGMENT

Thanks for giving me the opportunity to work in Flip Robo Technologies as Intern and would like to express my gratitude to Data Trained Institute aswell for trained me in Data Science Domain. This helps me to do my projects well and understand the concepts.

Resources Referred – Google, GitHub, Blogs for conceptual referring.

INTRODUCTION

- Business Problem Framing

  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 company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not.

  For this company wants to know:

  • Which variables are important to predict the price of variable?

  • How do these variables describe the price of the house?

- Conceptual Background of the Domain Problem

  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. The company is looking at prospective properties to buy houses to enter the market.

- Motivation for the Problem Undertaken

- 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.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

  Here our target variable is the “Sale Price” and need to predict the prices of the house. As the data is continuous and our problem is Regression.

- Data Sources and their formats

  The Data is provided by Flip Robo Technologies, and it has Train and Test Data Set and need to train our data in Train dataset and need to load the Test dataset to make the predictions.

```
#Loading the train dataset
df = pd.read_csv('train.csv')
df
```

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

1168 rows × 81 columns

```
#Loading the test dataset
df1 =  pd.read_csv('test.csv')
df1
```

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

292 rows × 80 columns

Data is having Null values and We need to treat the missing values and can be discussed in pre-processing.

- Data Pre-processing Done

```
# Dropping the columns from train dataset which is having more columns of null and unwante

df = df.drop(columns = ['PoolQC','Fence','MiscFeature','FireplaceQu','Alley','Id'], axis =
```

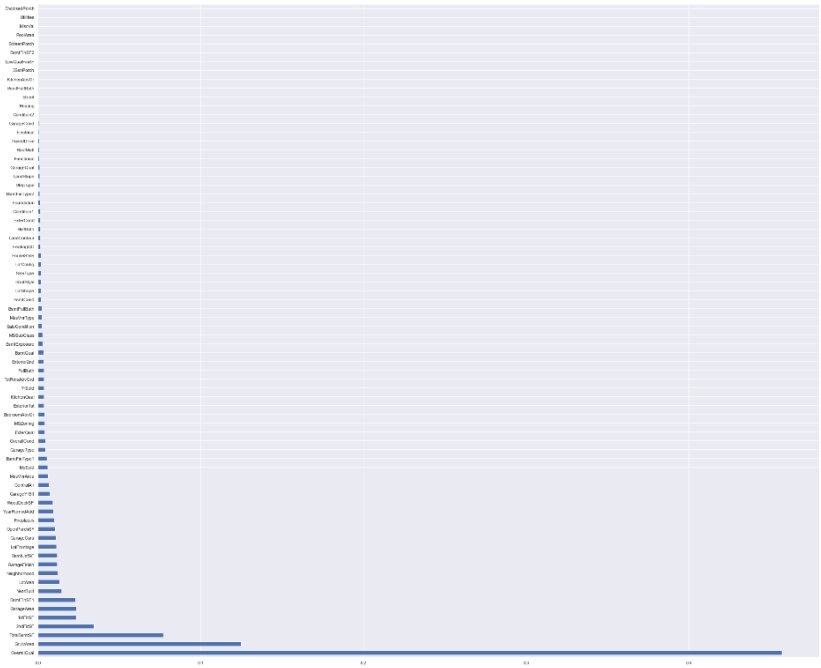
```
# Dropping the columns from test dataset which is having more columns of null and unwanted

df1 = df1.drop(columns = ['PoolQC','Fence','MiscFeature','FireplaceQu','Alley','Id'], axis
```

```
# Splitting the train dataset into numerical and categorical,

trn = df.select_dtypes(exclude = np.object)
trc = df.select_dtypes(include = np.object)
```

```
# Splitting the test dataset into numerical and categorical,

ten = df1.select_dtypes(exclude = np.object)
tec = df1.select_dtypes(include = np.object)
```

```
for i in trn:
    # Getting 75 and 25 %le
    q75, q25 = np.percentile(trn[i],[75,25])

    # calaculating IQR range
    iqr = q75-q25

    #calculating Upper and Lower
    max = q75 + (iqr*1.5)
    min = q25 - (iqr*1.5)

    # replace outliers val to NAN and filling NAN to median,

    trn.loc[trn[i]< min,i] = np.nan
    trn.loc[trn[i]> max,i] = np.nan

    trn[i].fillna(trn[i].median(),inplace = True)
```

```
# applying label encoder to categorical to convert into numerical

le = LabelEncoder()
column = ['MSZoning', 'Street', 'LotShape', 'LandContour', 'Utilities',
       'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2',
       'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st',
       'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation',
       'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2',
       'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual',
       'Functional', 'GarageType', 'GarageFinish', 'GarageQual', 'GarageCond',
       'PavedDrive', 'SaleType', 'SaleCondition']
for i in column:
    trc[column] = trc[column].apply(le.fit_transform)
    tec[column] = tec[column].apply(le.fit_transform)
```

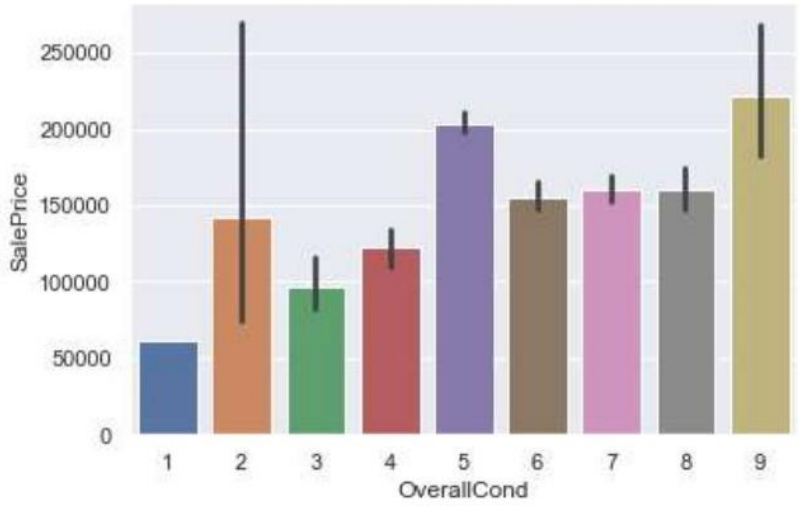


- Data Inputs- Logic- Output Relationships

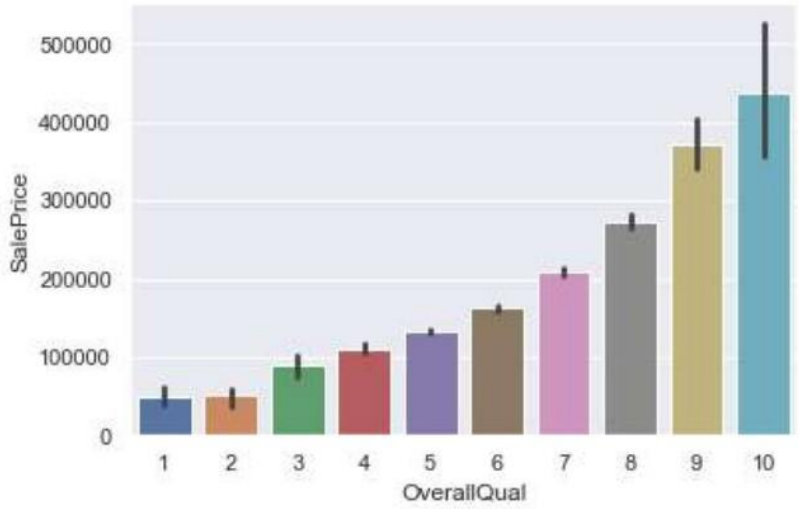
The below plot shows that rating of over all conditions of house which is greater than 5 has price of > 1.5L and the house which has rating of >8 of material and finishing of house price is >30L.

Also, the plot shows that most of the house roof style is shed / hip / flat.

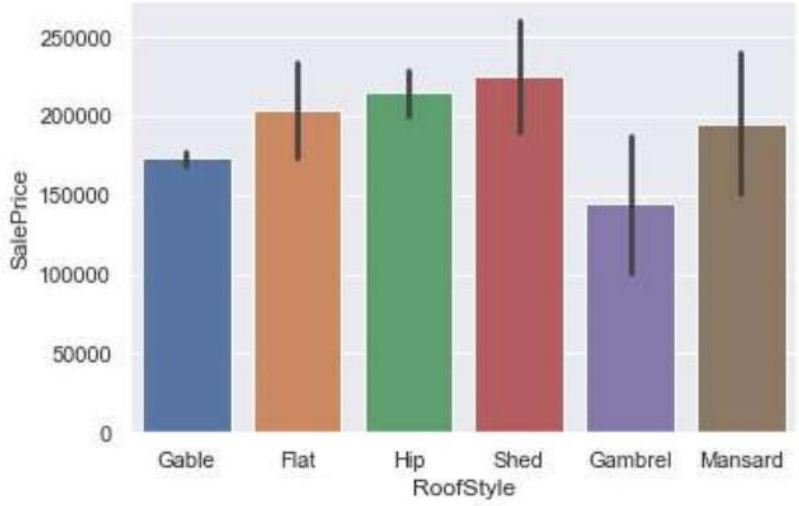
```
sns.set_theme()
sns.barplot(x = 'OverallCond', y = 'SalePrice', data = df)
plt.show()
```




```
sns.barplot(x = 'OverallQual', y = 'SalePrice', data = df)
plt.show()
```

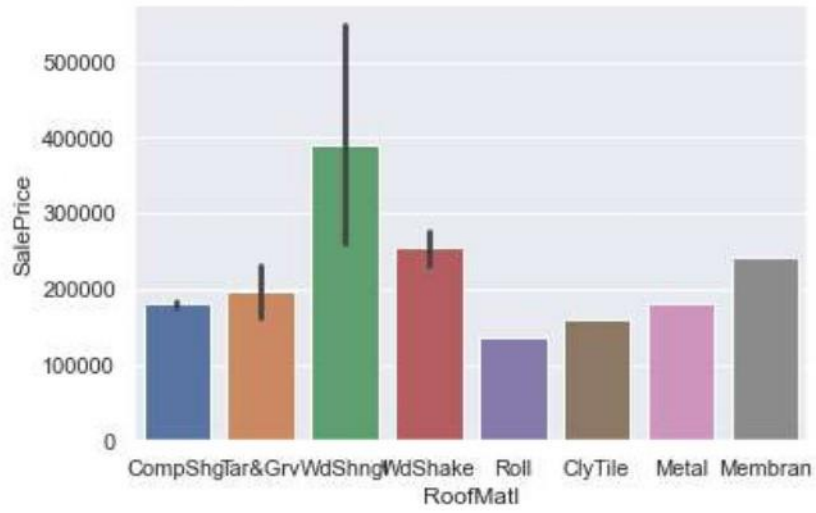



```
sns.barplot(x = 'RoofStyle', y = 'SalePrice', data = df)
plt.show()
```

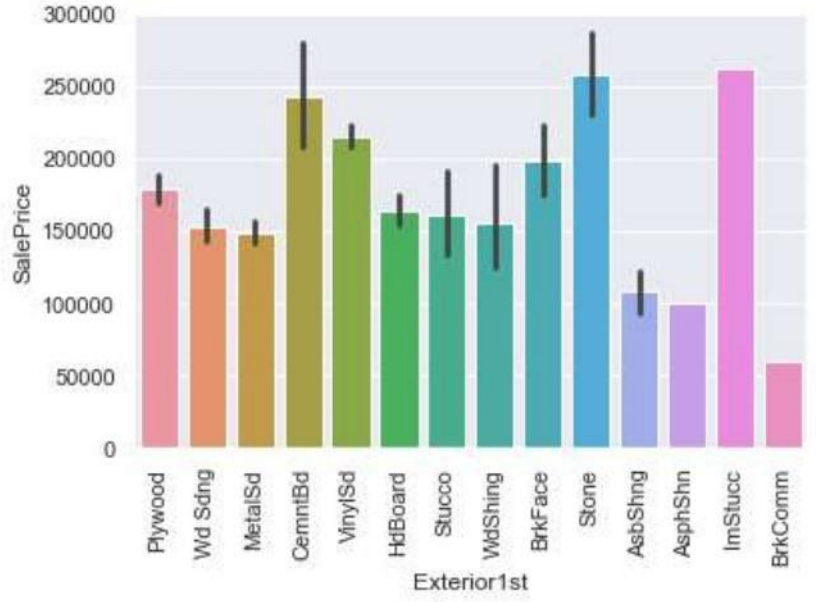


The below plot explains that most of the roof material in houses are Wood Shingles and the exterior used on houses are Imitation Stucco and Masonry veneer type of most of the houses are Stone and the houses which is satisfying these conditions are selling the house price >25L.

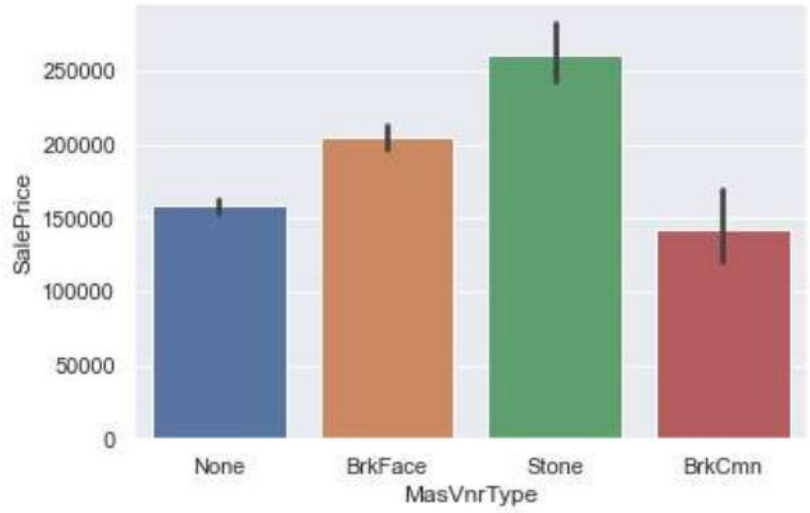
```
sns.barplot(x = 'RoofMatl', y = 'SalePrice', data = df)
plt.show()
```




```
sns.barplot(x = 'Exterior1st', y = 'SalePrice', data = df)
plt.xticks(rotation = 90)
plt.show()
```



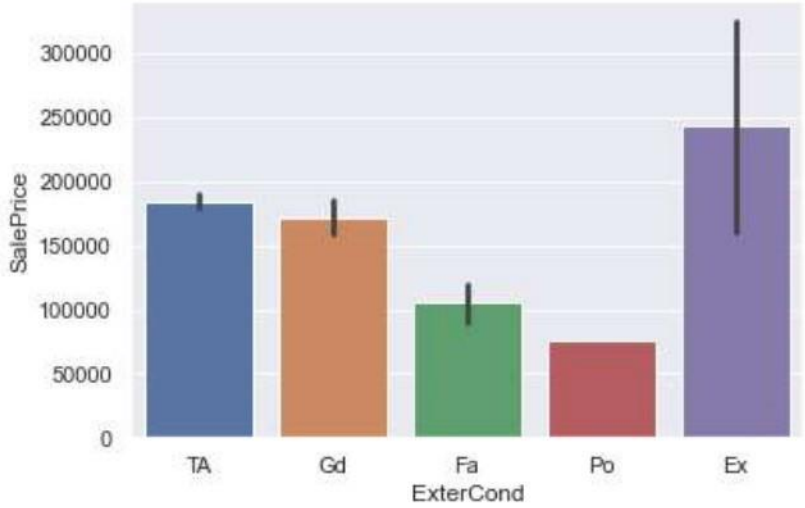
```
sns.barplot(x = 'MasVnrType', y = 'SalePrice', data = df)
plt.show()
```



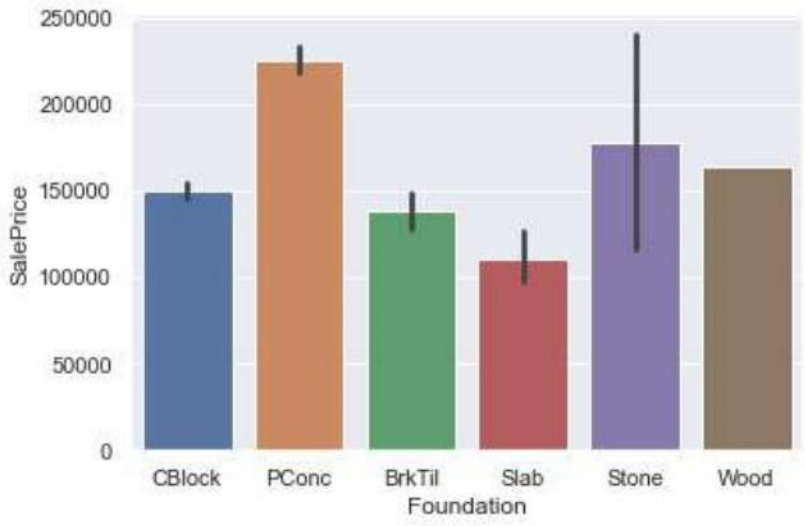
The exterior condition and basement of house which is having rating of Excellent has price of >20L.

Most of the house possess the foundation type as Poured Concrete and it has the high price of >20L.

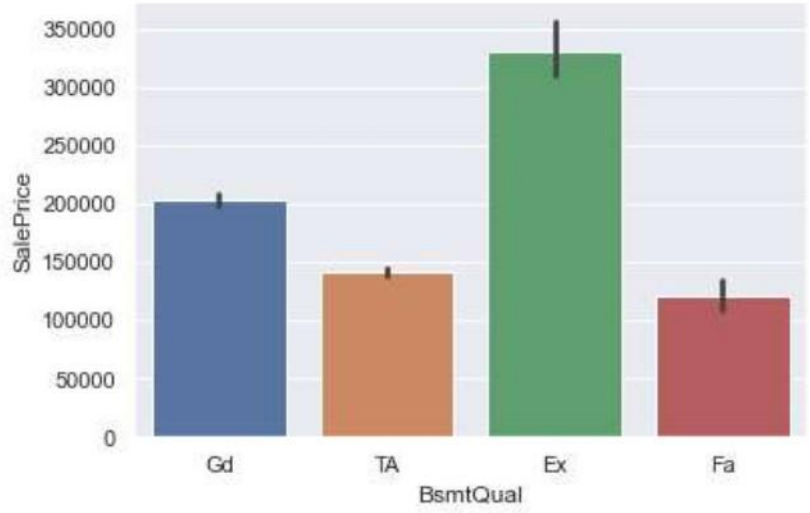
```
sns.barplot(x = 'ExterCond', y = 'SalePrice', data = df)
plt.show()
```




```
sns.barplot(x = 'Foundation', y = 'SalePrice', data = df)
plt.show()
```

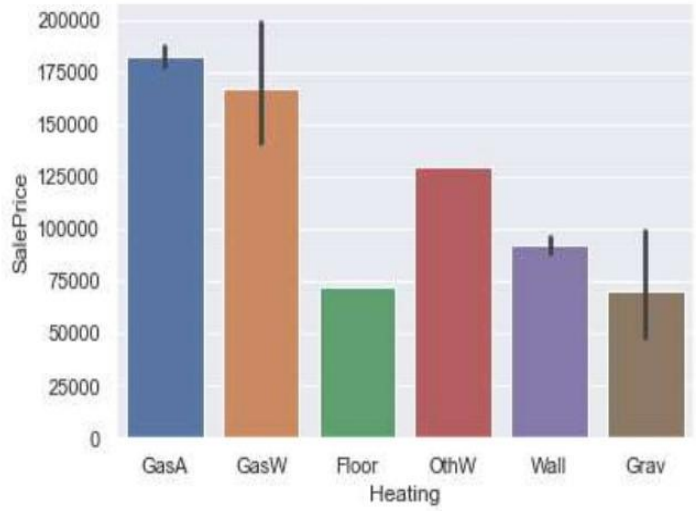


```
sns.barplot(x = 'BsmtQual', y = 'SalePrice', data = df)
plt.show()
```

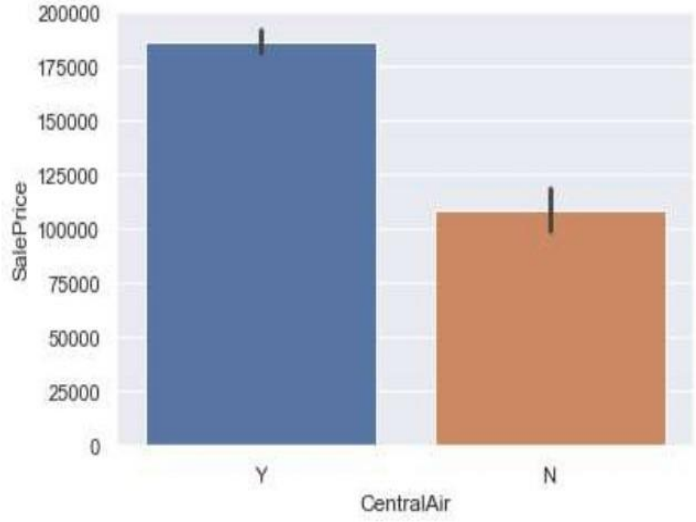


The plot shows that most of the house heating type is Gas forced warm air furnace and central air conditioned and fireplace quality should be excellent has price of >1.5L.

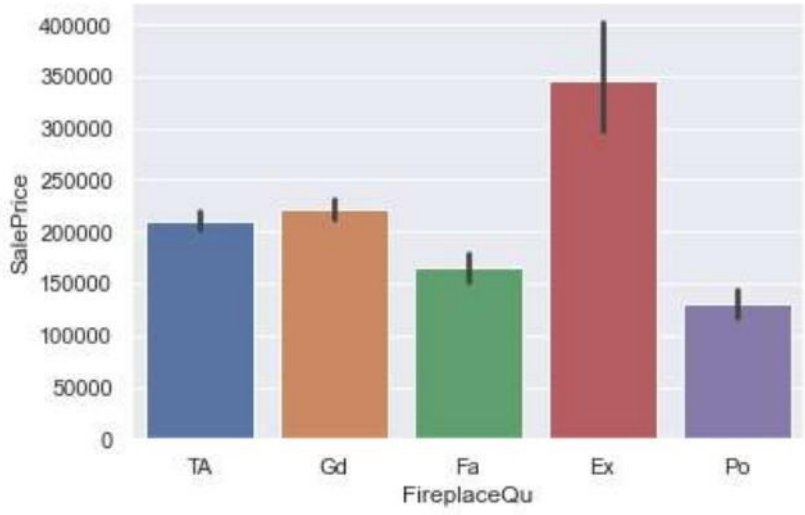
```
sns.barplot(x = 'Heating', y = 'SalePrice', data = df)
plt.show()
```




```
sns.barplot(x = 'CentralAir', y = 'SalePrice', data = df)
plt.show()
```




```
sns.barplot(x = 'FireplaceQu', y = 'SalePrice', data = df)
plt.show()
```



- Hardware and Software Requirements and Tools Used

  Model training was done on Jupiter Notebook. Kernel Version is Python3.

Model/s Development and Evaluation

- Testing of Identified Approaches (Algorithms)

  Random Forest

  K Neighbours

  Ada Boost

  Gradient Boost

  Linear Regression

- Run and evaluate selected models

```
x = dtr.drop(columns = ['HalfBath','ExterCond','Condition1','Foundation','BsmtFinType2','BldgType','LandSlope',
                        'GarageQual','Functional','RoofMatl','PavedDrive','Electrical','GarageCond','Condition2',
                        'Heating','Street','BsmtHalfBath','ScreenPorch','3SsnPorch','EnclosedPorch','KitchenAbvGr',
                        'Utilities',
                        'PoolArea','LowQualFinSF','MiscVal','BsmtFinSF2','SalePrice'],axis = 1)
y = dtr['SalePrice']
```

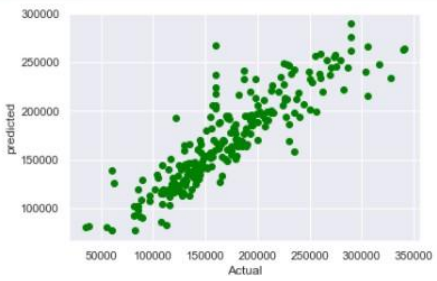
```
scaler = StandardScaler()
x_sc = scaler.fit_transform(x)
```

```
#Train test split

x_train,x_test,y_train,y_test = train_test_split(x_sc,y, test_size = 0.20, random_state = 555)
```

```
# Random forest algorithm and it has cv score of 74.4%

from sklearn.ensemble import RandomForestRegressor

rfr = RandomForestRegressor()
rfr.fit(x_train,y_train)
y_pred = rfr.predict(x_test)
scr_rfr = cross_val_score(rfr,x,y,cv=5)

print("r2_Score", r2_score(y_test,y_pred))
print("CV Score", scr_rfr.mean())
print("MSE",mean_squared_error(y_test,y_pred))
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
print("Train Score", rfr.score(x_train,y_train))
print("Test Score", rfr.score(x_test,y_test))
```

```
r2_Score 0.7742004493059257
CV Score 0.7441250243339593
MSE 777913847.391355
RMSE 27891.106958874097
Train Score 0.9626310453562714
Test Score 0.7742004493059257
```

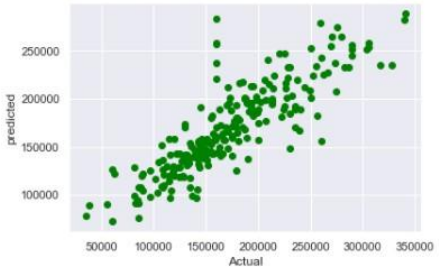
```
plt.scatter(y_test,y_pred, color = 'green')      #Scatter Matrix for Actual VS predicted for the model
plt.xlabel("Actual")
plt.ylabel("predicted")
plt.show()
```



```
# K Neighbors has cv score of 55.2%

from sklearn.neighbors import KNeighborsRegressor

knr = KNeighborsRegressor(n_neighbors = 5)
knr.fit(x_train,y_train)
y_pred = knr.predict(x_test)
scr_knr = cross_val_score(knr,x,y,cv=5)

print("r2_Score", r2_score(y_test,y_pred))
print("CV Score", scr_knr.mean())
print("MSE",mean_squared_error(y_test,y_pred))
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
print("Train Score", knr.score(x_train,y_train))
print("Test Score", knr.score(x_test,y_test))
```

```
r2_Score 0.7240238145058926
CV Score 0.5519686312843624
MSE 950779997.5075214
RMSE 30834.72064909169
Train Score 0.7720316309484047
Test Score 0.7240238145058926
```

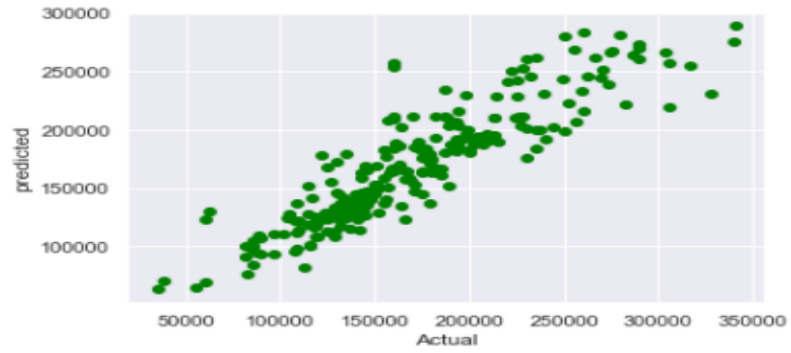
```
plt.scatter(y_test,y_pred, color = 'green')         #Scatter Matrix for Actual VS predicted for the model
plt.xlabel("Actual")
plt.ylabel("predicted")
plt.show()
```




```
#Gradient Boost has cv score of 75%

from sklearn.ensemble import GradientBoostingRegressor
gbr = GradientBoostingRegressor()
gbr.fit(x_train, y_train)
y_pred = gbr.predict(x_test)
scr_gbr = cross_val_score(gbr,x,y,cv=5)

print("r2_Score", r2_score(y_test,y_pred))
print("CV Score", scr_gbr.mean())
print("MSE",mean_squared_error(y_test,y_pred))
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
print("Train Score", gbr.score(x_train,y_train))
print("Test Score", gbr.score(x_test,y_test))
```

```
r2_Score 0.7911588863970327
CV Score 0.7509469795610286
MSE 719489448.3934989
RMSE 26823.30047539823
Train Score 0.9156286309462047
Test Score 0.7911588863970327
```

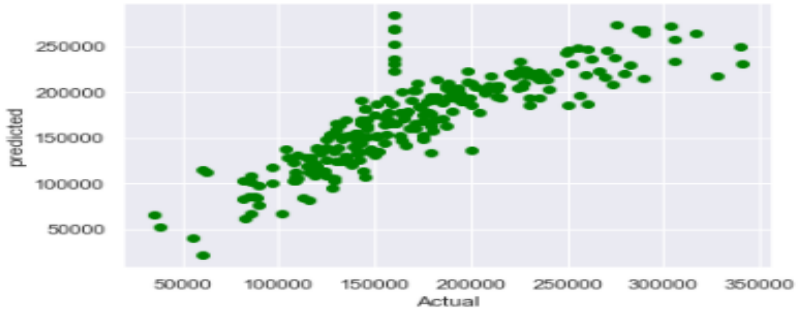
```
plt.scatter(y_test,y_pred, color = 'green')     #Scatter Matrix
plt.xlabel("Actual")
plt.ylabel("predicted")
plt.show()
```



```
# Linear Regression has cv score of 69.5%

from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(x_train,y_train)
y_pred = lr.predict(x_test)
scr_lr = cross_val_score(lr,x,y,cv=5)

print("r2_Score", r2_score(y_test,y_pred))
print("CV Score", scr_lr.mean())
print("MSE",mean_squared_error(y_test,y_pred))
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
print("Train Score", lr.score(x_train,y_train))
print("Test Score", lr.score(x_test,y_test))
```

```
r2_Score 0.7194377394441989
CV Score 0.6952036967821108
MSE 966579724.6757176
RMSE 31089.865304882194
Train Score 0.7244934918276908
Test Score 0.7194377394441989
```

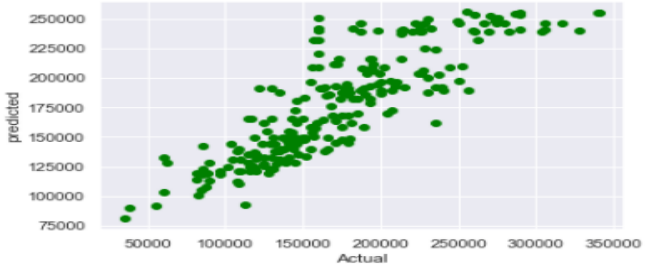
```
plt.scatter(y_test,y_pred, color = 'green')      #Scatter Matrix
plt.xlabel("Actual")
plt.ylabel("predicted")
plt.show()
```




```
# Ada boost model of cv score of 68.2%

from sklearn.ensemble import AdaBoostRegressor
abr = AdaBoostRegressor()
abr.fit(x_train, y_train)
y_pred = abr.predict(x_test)
scr_abr = cross_val_score(abr,x,y,cv=5)

print("r2_Score", r2_score(y_test,y_pred))
print("CV Score", scr_abr.mean())
print("MSE",mean_squared_error(y_test,y_pred))
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
print("Train Score", abr.score(x_train,y_train))
print("Test Score", abr.score(x_test,y_test))
```

```
r2_Score 0.7060329381303554
CV Score 0.6818308073758554
MSE 1012761307.1080794
RMSE 31823.91093357445
Train Score 0.7461556023702922
Test Score 0.7060329381303554
```

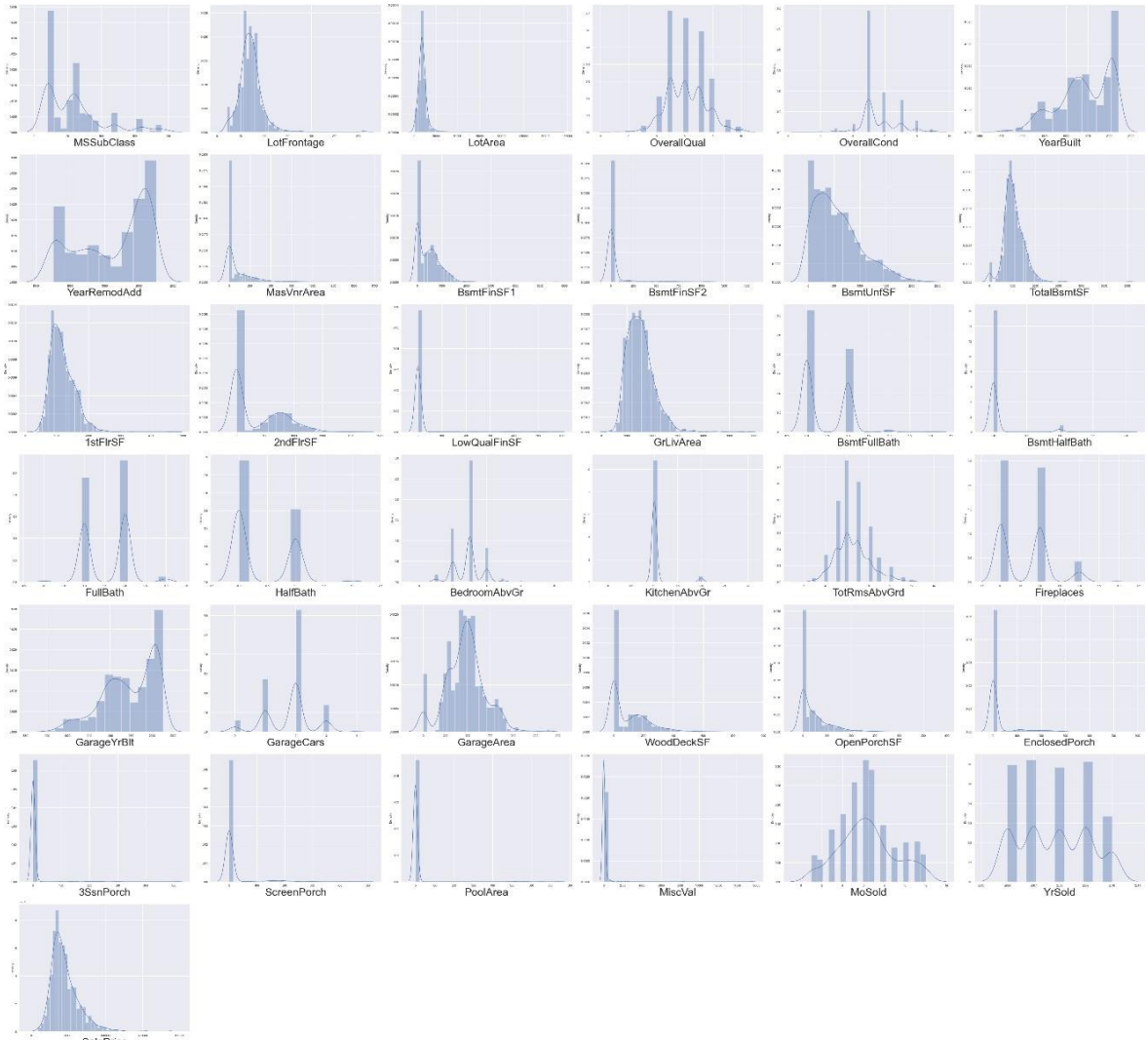
```
plt.scatter(y_test,y_pred, color = 'green')     #Scatter Matrix
plt.xlabel("Actual")
plt.ylabel("predicted")
plt.show()
```

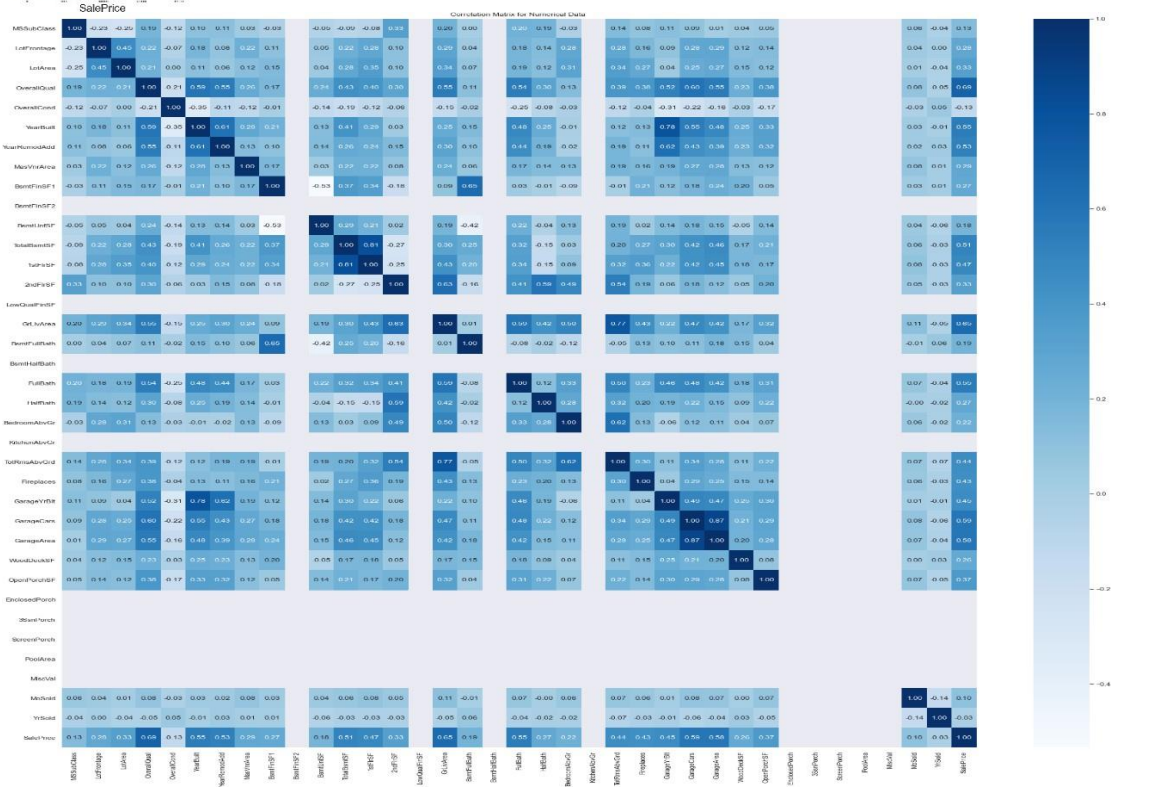


- Visualizations

  Box plot and Dist. plot,


MSZoning
Street
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
BldgType
HouseStyle
RoofStyle
RoofMatl
Exterior1st
Exterior2nd
MasVnrType
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinType2
Heating
HeatingQC
CentralAir
Electrical
KitchenQual
Functional
GarageType
GarageFinish
GarageQual
GarageCond
PavedDrive
SaleType
SaleCondition


MSSubClass
LotFrontage
LotArea
OverallQual
OverallCond
YearBuilt
YearRemodAdd
MasVnrArea
BsmtFinSF1
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
1stFlrSF
2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
TotRmsAbvGrd
Fireplaces
GarageYrBlt
GarageCars
GarageArea
WoodDeckSF
OpenPorchSF
EnclosedPorch
3SsnPorch
ScreenPorch
PoolArea
MiscVal
MoSold
YrSold
SalePrice


Correlation Matrix for Numerical Data

- Interpretation of the Results

```
# Applying randomized search CV to increase the accuracy,

rg = RandomizedSearchCV(gbr, param_distributions = param, cv= 5)
rg.fit(x_train,y_train)
rg.best_params_
```

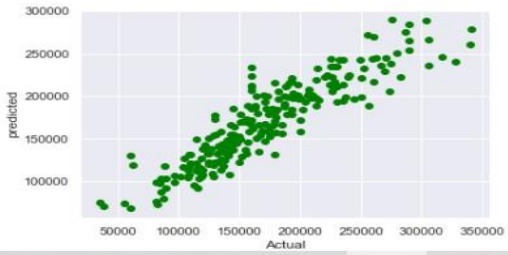
```
{'n_estimators': 100,
 'min_samples_leaf': 15,
 'loss': 'lad',
 'learning_rate': 0.2,
 'criterion': 'mse'}
```

```
final = GradientBoostingRegressor(loss = 'lad',n_estimators= 100,criterion= 'mse', learning_rate = 0.2 , min_samples_leaf = 35 )
final.fit(x_train, y_train)
y_pred = final.predict(x_test)

print("r2_Score", r2_score(y_test,y_pred))
print("MSE",mean_squared_error(y_test,y_pred))
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
print("Train Score", final.score(x_train,y_train))
print("Test Score", final.score(x_test,y_test))
```

```
r2_Score 0.8188525273540491
MSE 624080636.8217899
RMSE 24981.60596962873
Train Score 0.8228689645521672
Test Score 0.8188525273540491
```

Our model has given the accuracy score ie r2 score of 82% which is high than the CV score of model and r2 score of model.

```
plt.scatter(y_test,y_pred, color = 'green')   #Scatter Matrix for Actual VS predicted for the model
plt.xlabel("Actual")
plt.ylabel("predicted")
plt.show()
```



CONCLUSION

- Key Findings and Conclusions of the Study

As this project is about predicting the price of house in US market, The problem is Regression and I have used 5 algorithms to build the model and among all Gradient Boosting algorithm is the best model.

I have used hyper parameter tuning to increase the accuracy score as well and price which predicted was slightly varying from the actual price.

We have 2 datasets for training and testing the data and it has more columns but less rows and removed the columns which is less and no importance using feature importance feature engineering.

So, this will helps the clients to understand the variables which are important to predict the price of house and helps them to get success in their business.

- ## Learning Outcomes of the Study in respect of Data Science

  Random forest and Boosting algorithms work best and Randomized search CV is faster than Grid search CV.

  As this dataset has 2 data files and we need to do all pre-processing, EDA to both datasets and train dataset has target variable and test dataset will not have target variable which we need to predict it.

  This is kind of different and time taking process but helps me to learn more and most important is that I am getting hands-on experience more on Data Science Concepts.