HOUSING PRICE PREDICTION PROJECT

Submitted by:

SUDHANSU MANDAL

INTRODUCTION

➢ Business Problem Framing

Houses are one of the necessary needs of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors 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. A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file below. The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not.

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

As we know real estate market is one of the markets which is one of the major contributions in the world's economy. There are many companies working in this domain which make this industry very competitive for players. They find very challenging to predict the purchase house at below their actual values and sell them at higher price to gain profit margin. Hence Data Science helps them to predict the prices of house based on various independent variables which plays very important role while predict the price of house. The model helps them to decide whether to invest on the particular properties or not and also helps them to set the correct selling price of the properties based on the properties various attributes. 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.

➢ Review of Literature

This project observed those features, attribute or factor which impact on predict the house price. To complete the project, we have followed below process:

1. **Understand the problem:** Before getting the data, we need to understand the problem statement and the objective of the project. We need to understand the problem we are trying to solve with the help of data science.
2. **Get Data:** In this step, we need to download the dataset. Here we have two files of Training and Test Dataset. We need to determine which features play important role in predicting house sale price.
3. **Data Exploration:** In this section, we will need to deep dive into the dataset and tries to understand the nature of variables, so that they can be treated properly. This step will involve to creates lots of charts, graphs (Univariate, Bivariate and Multivariate analysis) and cross tables to understand the behaviours of features.
4. **Data Pre-Processing:** Here we will clean our dataset based on our model requirement. We will find the null value and impute them with correct value. Identifying the datatype of features so that we can do feature engineering accordingly.
5. **Feature Engineering:** In this step we need to impute with various imputation technique so model can predict the sale price with high performance. We will also look into the best features which contribute in predicting house price.
6. **Model Training:** Using various algorithm which is suitable for the dataset and we train the model on the given training dataset.
7. **Model Evaluation:** This is very important steps to know that how the model has been trained in provided dataset. We will evaluate the model performance based on various technique.
8. **Model Testing:** At last we will identify the best model for the dataset which provide less error with high performance score and will predict the sale price on given test dataset.

➢ Motivation for the Problem Undertaken

Real Estate is very big industry in the world's economy. Many players presence in market who deal with real estate so the competition is very tough. To become successful in this industry the companies must analyse the various factors which can predict the sale price of house. The company must know to invest on properties which returns high profit or margin. The sale prices would depend on various factor or attributes which will analyse in this project through various techniques.

Here Data Science can help the Real estate to align their strategies and marketing mix based on the attributes which are highly correlated with sale price.

ANALYTICAL PROBLEM FRAMING

➢ Mathematical/ Analytical Modeling of the Problem

In this project our dataset having two files which contains Training Dataset and Testing Dataset. The Training Dataset having 1168 observation and 81 columns or features including target variable and Testing Dataset having 292 observation and 80 columns or features.
To understand our Training and Testing data, we have done basic data exploration.

- **Statistic Summary**

Here we have tried to understand the Statistic of data with the help of describe () and Info () function. Mention below are the observation made from this summary:

1. **Data Type:** As we have seen that Training Dataset having 43 columns object datatype, 35 integer datatype and 3 float datatypes and Testing Dataset having 42 columns object datatype, 34 integer and 4 float datatypes.
2. **Null Values:** We have seen that null values in training and testing dataset, which we need to impute with proper technique.
3. Id column need to drop as this column would not require for our analysis, this is only for the row identification.
4. We can analyse that data distribution for columns are not normal as there are difference in value of mean and 50% data value also there would be skewed columns data which we would deal later.

- **Feature Selection**

We have seen that 81 features in Training dataset and 80 features in Testing Dataset, many of which have analysed that not necessary or not much information about the target variable. We have to select only most significant features to make Machine learning models more efficient and cost effective.

- **Data Visualization**

While doing EDA, we have found mentioned below important points for our dataset:

a. Target Variable Skewed Distribution
b. Outliers in Training and Testing Dataset
c. Maximum Data are right skewed data.
d. Most Significate Numerical and Categorical features on predicting Sale Price.

- **Data Normalization**

Since the data was not normal and having right skewed so we have normalized all the features including our target variable.

➢ Data Sources and their formats

The data sources have been provided to us in .csv format and also excel file of having details description about the dataset. There are separate Training and Testing Dataset for our model which we have imported and converted into Pandas dataframe for analysing purpose. The dataset columns having float, int and object data type as shown in Fig 1.

Fig 1:

Training Dataset:

```
dtypes: float64(3), int64(35), object(43)
memory usage: 739.2+ KB
```

Testing Dataset:

```
dtypes: float64(4), int64(34), object(42)
memory usage: 182.6+ KB
```

➢ Data Pre-processing

As we know Pre-Processing of our data is an important step for our model to perform better in predicting the sale price of house. Below is the step for data Preprocessing done:

- **Checking unique value of dataset.**
  As shown in Fig 2, we have checked the unique value in dataset, which helps us to know the categorical features level in the particular columns.

  Fig: 2

  Training Dataset:

```
Out[559]: Id             1168
          MSSubClass       15
          MSZoning          5
          LotFrontage     108
          LotArea         892
                            ...
          MoSold           12
          YrSold            5
          SaleType          9
          SaleCondition     6
          SalePrice       581
          Length: 81, dtype: int64
```

  Testing Dataset:

```
Out[12]: Id             292
         MSSubClass      15
         MSZoning         4
         LotFrontage     65
         LotArea        249
                          ...
         MiscVal          8
         MoSold          12
         YrSold           5
         SaleType         6
         SaleCondition    4
         Length: 80, dtype: int64
```

- **Checking for null value**

In real world data, sometimes we would be missing value in the dataset which occur due to various reason like corrupt data, failure to load the information or incomplete extraction, and sometime due to human error. Handling these missing values is one of the most important challenges faced by analysts because it plays very important role in making the right decision on how to handle it, which helps in to make robust data models.

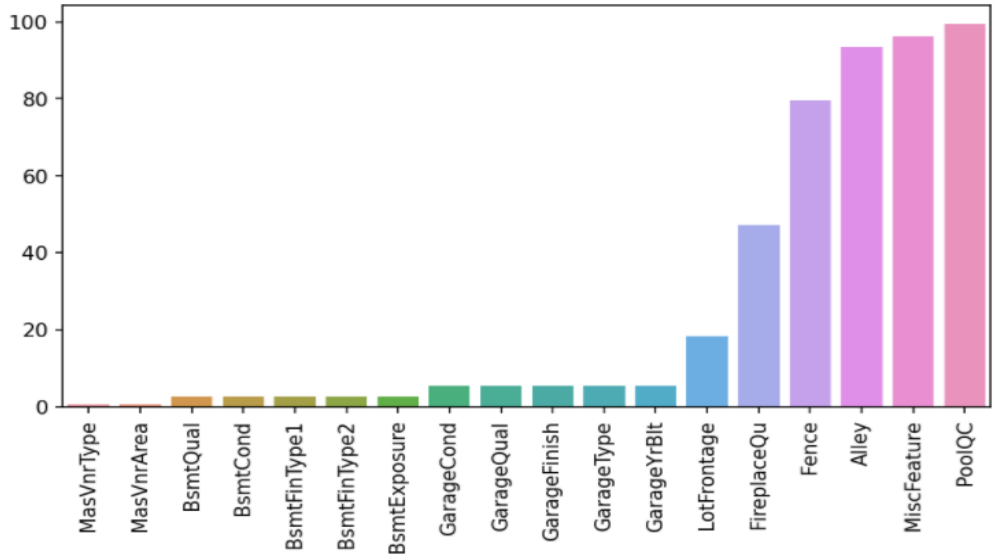
As Shown in Fig 3, we have lots of Null values in our both the dataset. There are some columns where the null value is more than 90% of data.

Fig 3:

Training Dataset:

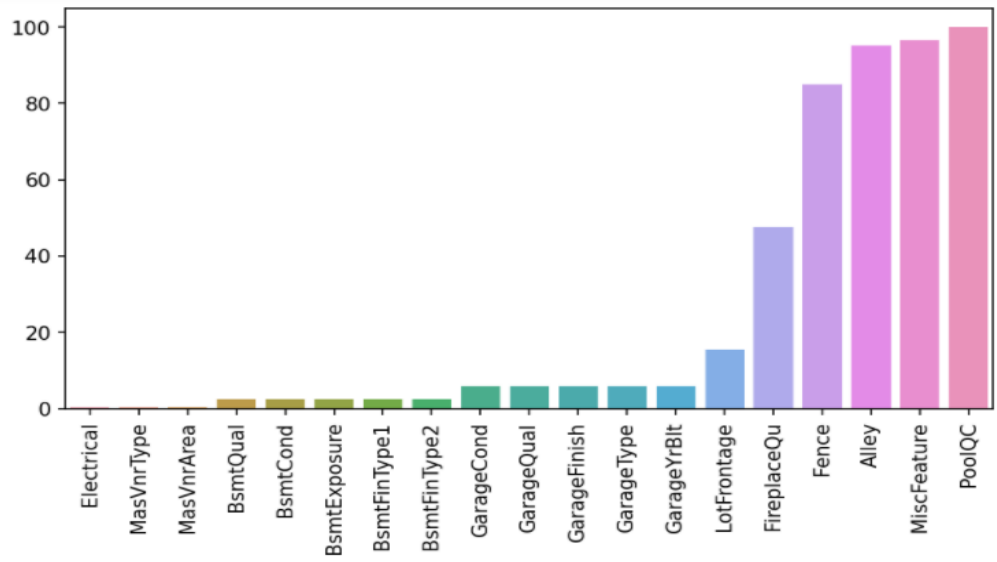
Null Value Training Dataset:

```
Out[17]: MasVnrType       0.599315
         MasVnrArea       0.599315
         BsmtQual         2.568493
         BsmtCond         2.568493
         BsmtFinType1     2.568493
         BsmtFinType2     2.654110
         BsmtExposure     2.654110
         GarageCond       5.479452
         GarageQual       5.479452
         GarageFinish     5.479452
         GarageType       5.479452
         GarageYrBlt      5.479452
         LotFrontage     18.321918
         FireplaceQu     47.174658
         Fence           79.708904
         Alley           93.407534
         MiscFeature     96.232877
         PoolQC          99.400685
```



Null Value Testing Dataset:

```
Out[21]: Electrical       0.342466
         MasVnrType       0.342466
         MasVnrArea       0.342466
         BsmtQual         2.397260
         BsmtCond         2.397260
         BsmtExposure     2.397260
         BsmtFinType1     2.397260
         BsmtFinType2     2.397260
         GarageCond       5.821918
         GarageQual       5.821918
         GarageFinish     5.821918
         GarageType       5.821918
         GarageYrBlt      5.821918
         LotFrontage     15.410959
         FireplaceQu     47.602740
         Fence           84.931507
         Alley           95.205479
         MiscFeature     96.575342
         PoolQC         100.000000
         dtype: float64
```



- **Dealing with Null Values**

In this project we have done mentioned below technique to impute our missing value from the columns.

✓ Imputed with NA, None, Mean, Median or Mode:

As we have seen some of the columns where we have imputed our both training and testing dataset's Null Value with NA, None, Mean, Median and Mode value in dataset, as shown in Fig 4.

Fig 4:

```
In [28]: train_df["MasVnrType"]=train_df["MasVnrType"].fillna("None")

In [29]: train_df[train_df["MasVnrType"].isnull()]

Out[29]:
```

Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSo

0 rows × 81 columns

```
In [30]: train_df["MasVnrArea"].isnull().sum()

Out[30]: 7

In [31]: train_df["MasVnrArea"]=train_df["MasVnrArea"].fillna(0)

In [32]: train_df["MasVnrArea"].isnull().sum()

Out[32]: 0
```

```
In [37]: train_df["BsmtQual"].value_counts()
Out[37]: TA    517
         Gd    498
         Ex     94
         Fa     29
         Name: BsmtQual, dtype: int64

In [38]: mode=train_df["BsmtQual"].mode()
```

We will fill na with the mode value of column.

```
In [39]: train_df["BsmtQual"]=train_df["BsmtQual"].fillna(mode[0])

In [40]: train_df["BsmtQual"].isnull().sum()
Out[40]: 0

In [41]: train_df["BsmtQual"].value_counts()
Out[41]: TA    547
         Gd    498
         Ex     94
         Fa     29
         Name: BsmtQual, dtype: int64
```

✓ Imputed Mean/Median Value with respect of other columns:

As shown in Fig 5, we have imputed Lot Frontage features null value with the median value in respect of Neighbourhood columns.

Fig: 5

```
In [74]: train_df["LotFrontage"].isnull().sum()
Out[74]: 214

In [75]: lot_frontage_by_neighborhood = train_df['LotFrontage'].groupby(train_df['Neighborhood'])

In [76]: for key, group in lot_frontage_by_neighborhood:
             idx = (train_df['Neighborhood'] == key) & (train_df['LotFrontage'].isnull())
             train_df.loc[idx, 'LotFrontage'] = group.median()

In [77]: train_df["LotFrontage"].isnull().sum()
Out[77]: 0
```

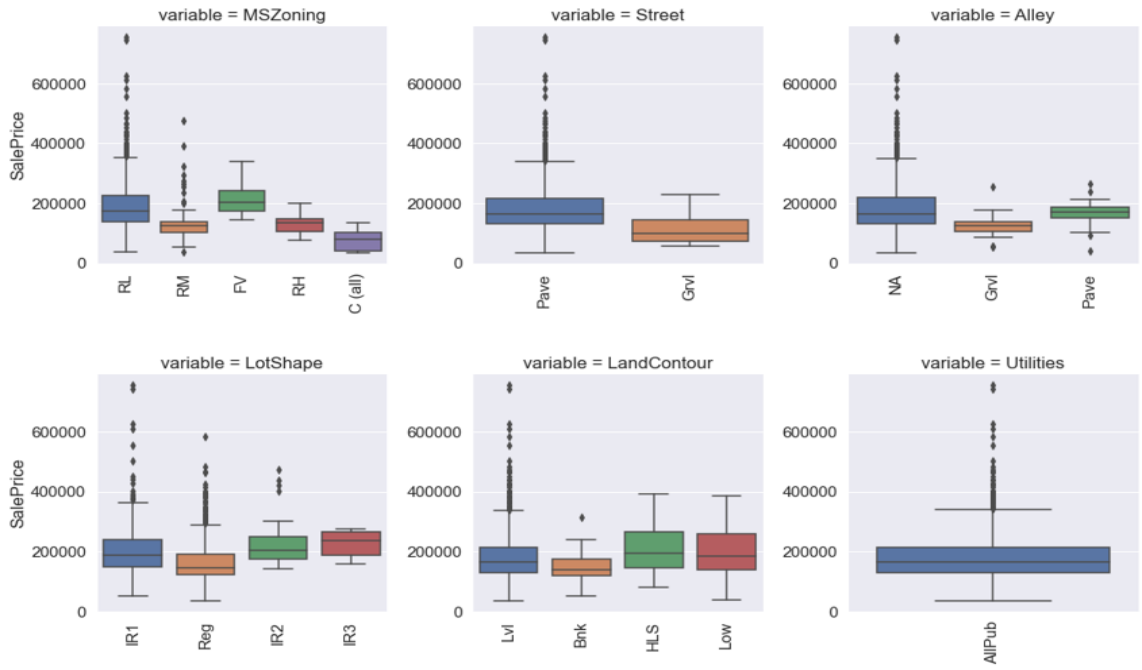
- **Dropping unwanted Columns**

As there are some columns which has only for identification purpose or those columns we believe that those columns are not have any significant for our target variable so we have removed from the dataset. Like Id columns.

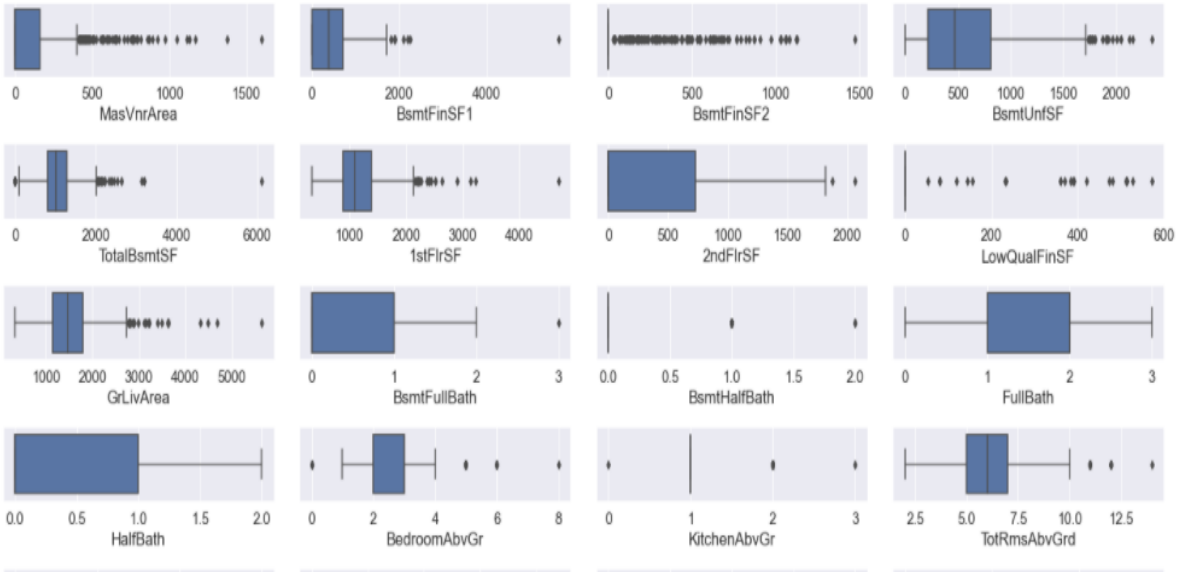
- **Checking for outlier's presence in Dataset.**

In EDA we have visualized that as shown in Fig 6, that there are lot of Categorical Columns where we have outliers with respect of Sale Price in both the training and testing dataset.

Fig: 6



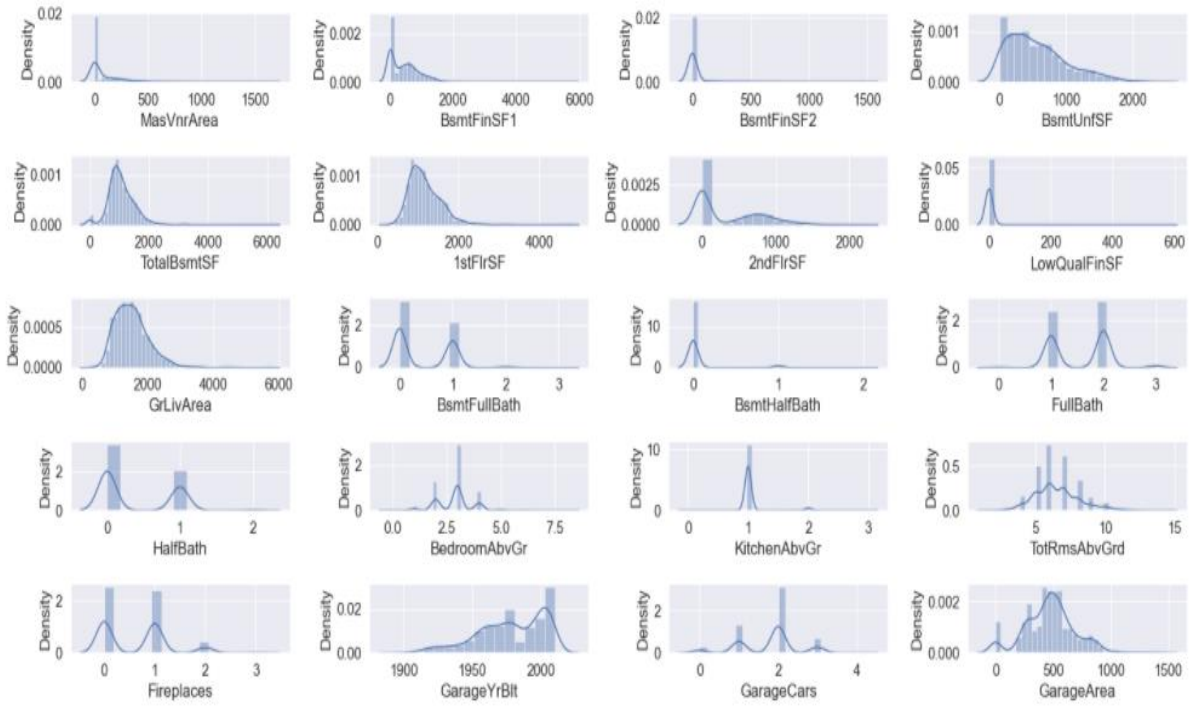
Also, as shown in Fig 7, Outliers present in Numerical features too



- **Dealing with Skewed Data**

As shown in Fig 8, we have also seen that our both datasets having right skewed data. To Avoid any loss of information we have removed data skewedness with log transformation technique.

Fig: 8



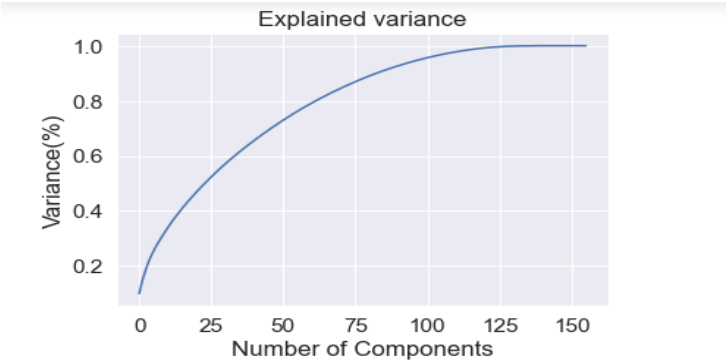

```
In [305]: numeric_features = [f for f in train_df.columns if train_df[f].dtype != object]
          numeric_features.remove("SalePrice")
```

```
In [306]: from scipy.stats import skew
          skewed = train_df[numeric_features].apply(lambda x: skew(x.dropna().astype(float)))
          skewed = skewed[skewed > 0.75]
          skewed = skewed.index
          train_df[skewed] = np.log1p(train_df[skewed])
```

➢ Data Inputs- Logic- Output Relationships

In this dataset, our input variable having int, float and object data type. Our target variable is Continuous data type.
We have seen there are 80 features and 1 target variable in dataset but after doing Feature Engineering our features has increased and now having 156 features.
We have also applied Dimension reduction technique through Principal Component Analysis (PCA) where we have found that 90 number of components explained almost more than 90 % variance, as shown in Fig 9. But we will see our model performance without applying PCA First and based on performance we will decide whether to apply PCA data as input for model or not.

Fig: 9



➢ Hardware and Software Requirements and Tools Used

In this project the mention below hardware, software and tools used to complete this project:

- ✓ Hardware:

Processor	Intel(R) Pentium(R) CPU 3825U @ 1.90GHz 1.90 GHz
Installed RAM	4.00 GB
System type	64-bit operating system, x64-based processor

- ✓ Software:

Edition	Windows 10 Home Single Language
Anaconda	

- ✓ Library & Tools:

Jupyter Notebook
Pandas
NumPy
Matplotlib
Seaborn

Model/s Development and Evaluation

➢ Identification of possible problem-solving approaches (methods)

As we have seen that our target variable is in Continuous type, so we need to apply Regression task to the dataset.
We have seen that our data is not distributed normally and having outliers, so we have normalized the features columns with **Standard Scaler** techniques.
Also, we have done PCA **(Principal component Analysis)** for dimension reduction of our feature's dataset, as shown in fig 9, we can choose 90 components for our model, but we will see the performance of model without removing any features and then will decide whether need to use dimension reduction technique or not.
We have seen that without dimension reduction technique model is performing better.

As shown in Fig 10, we have found that in Linear Regression Model is Overfitting, so we need to apply Regularization technique with Ridge or Lasso and have evaluated the model performance.

Fig 10:

Linear Regression Model Performance on Training and Testing across different Kfolds.

➢ Testing of Identified Approaches (Algorithms)

As we know our target variable having continuous type of data, so we need apply regression algorithm for model building. Since we have outliers in dataset, so we have applied linear model, tree-based model and ensemble model algorithm to the dataset and have observed their performance with different Cross Validation.
Below are the Regression algorithms have applied to the model.

1. Linear Regression
2. Ridge Regression
3. Lasso Regression
4. Random Forest Regression
5. Decision Tree Regressor
6. Xtreme Gradient Boosting Regressor.

We have applied mentioned above model in our training and test dataset. Here in project we have split training 75% and testing 25% of data.

➢ Run and Evaluate selected models

As shown in Fig 11, we have separated the training and test data set into 75% and 25%.

Fig: 11

```
In [472]: x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.25,random_state=2347)
```

Here we have chosen Best Random State for the model as shown in Fig 12.

Fig: 12

```
In [471]: from sklearn.model_selection import train_test_split
          from sklearn.metrics import r2_score
          mlr=LinearRegression()
          for i in range(2000,3000):
              x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.25,random_state=i)
              mlr.fit(x_train,y_train)
              pred_train=mlr.predict(x_train)
              pred_test=mlr.predict(x_test)

              if round(r2_score(y_train,pred_train)*100,1)==round(r2_score(y_test,pred_test)*100,1):
                  print("At Random State",i,"Model perform well")
                  print("At Random State",i)
                  print("Training R2_score is-",r2_score(y_train,pred_train)*100)
                  print("Testing R2_score is-",r2_score(y_test,pred_test)*100)
```

Now we have evaluated our model, based on R2_Score, MAE and RMSE.

1. Linear Regression

Evaluation Matrix:

Fig: 13

```
In [474]: from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

In [475]: pred_test=mlr.predict(x_test)

In [476]: MAE=mean_absolute_error(y_test,pred_test)

In [477]: MAE
Out[477]: 0.08262247519126195

In [478]: RMSE=np.sqrt(mean_squared_error(y_test,pred_test))

In [479]: RMSE
Out[479]: 0.10986126172314514

In [480]: mlr_score=r2_score(y_test,pred_test)

In [481]: mlr_score
Out[481]: 0.9127026543049666
```

Here we can see the Linear Regression R2 Score is 91.27%.
MAE is 0.08
RMSE is 0.11

Also, we have calculated Cross Validation Score to understand our model performance and have seen that our Cross-Validation Score is negative. Hence, it indicates that model having issue with overfitting or underfitting problems.

To understand this, we have plotted training and testing performance of Linear Regression model in different folds and observed the performance of model and error as shown in Fig 14.

Fig 14:

As seen in Above Fig 14, The model performance is not good in test dataset while checking model with different 15 Folds of cross validation. As we can infer from the plot that model is overfitting as in training set model score is better than testing dataset.

Since we have overfitting issue so we will apply regularization technique to deal with overfitting issue.

2. Ridge Regression

Evaluation Matrix:

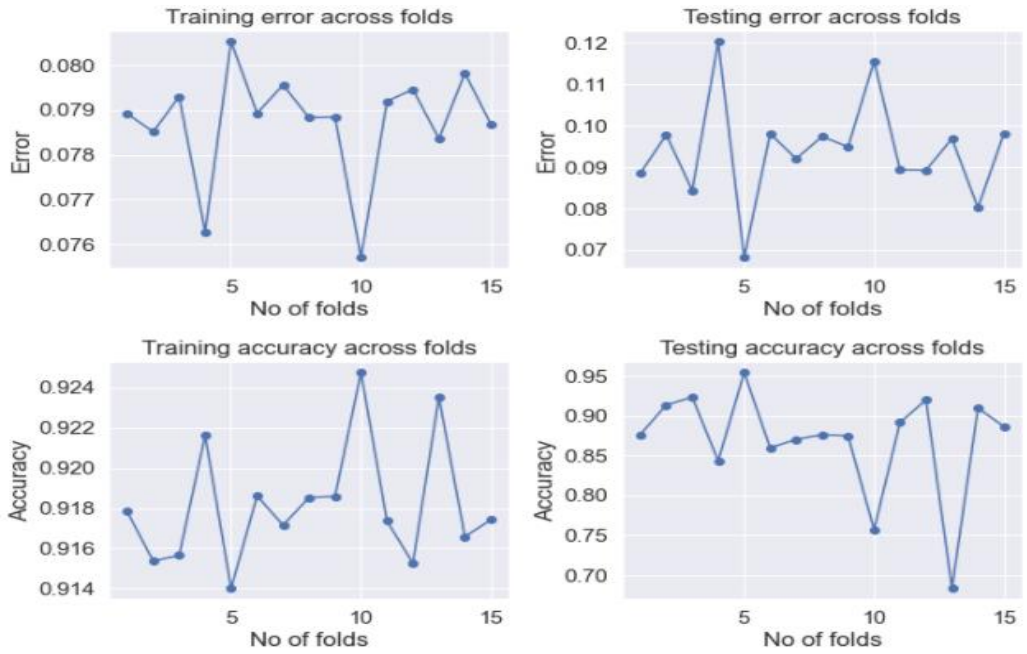
Fig: 15

```
In [493]: pred_test=ridge_model.predict(x_test)

In [494]: MAE=mean_absolute_error(y_test,pred_test)

In [495]: MAE
Out[495]: 0.08246348960621604

In [496]: RMSE=np.sqrt(mean_squared_error(y_test,pred_test))

In [497]: RMSE
Out[497]: 0.10974871866009565

In [498]: ridge_score=r2_score(y_test,pred_test)

In [499]: ridge_score
Out[499]: 0.9128814193811494
```

Here we can see the Ridge Regression R2 Score is 91.28%.
MAE is 0.08
RMSE is 0.11
Also, we have calculated Cross Validation Score to understand our model performance and have seen that our Cross-Validation Score is 86% better than Linear Regression.

To understand better, we have plotted training and testing performance of Ridge Regression model in different folds and observed the performance of model and error as shown in Fig 16.

Fig 16:



As seen in Above Fig 16, The model performance far better than Linear Regression in training and test dataset while checking model with different 15 Folds of cross validation. As we can infer from the plot that model has very minimum difference in training and testing in respect of error and accuracy.

3. Lasso Regression

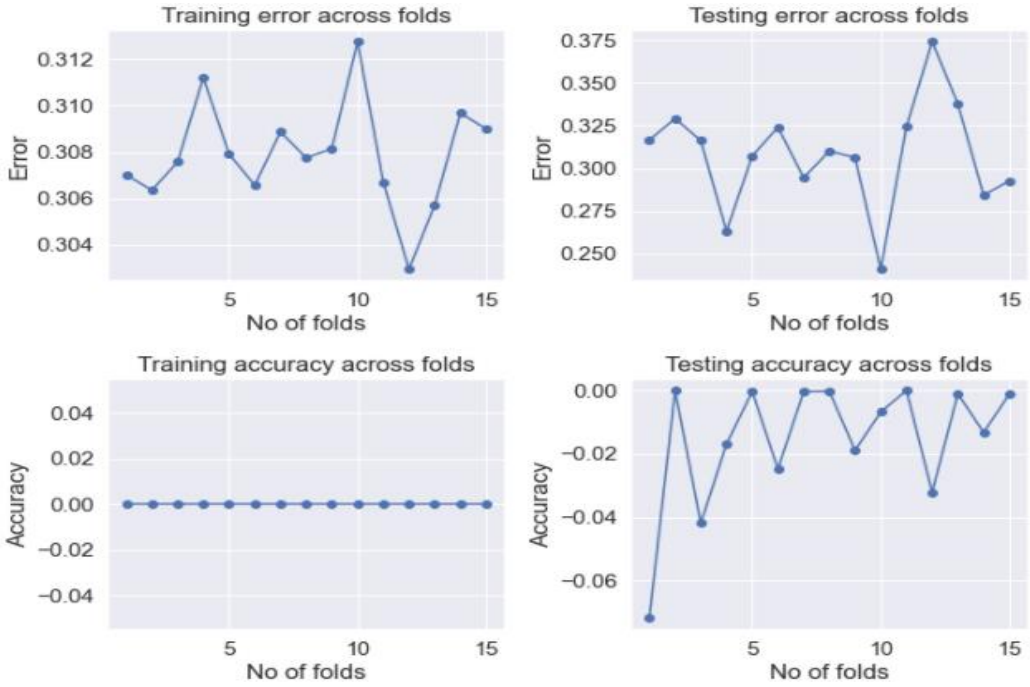
Evaluation Matrix:

Fig: 17

```
In [510]: pred_test=lasso_model.predict(x_test)

In [511]: MAE=mean_absolute_error(y_test,pred_test)

In [512]: MAE
Out[512]: 0.2885092567076428

In [513]: RMSE=np.sqrt(mean_squared_error(y_test,pred_test))

In [514]: RMSE
Out[514]: 0.3730450525769711

In [515]: lasso_score=r2_score(y_test,pred_test)

In [516]: lasso_score
Out[516]: -0.006547891435053188
```

Here we can see the Lasso Regression R2 Score is negative i.e. -0.006%.
MAE is 0.28
RMSE is 0.37
Also, we have calculated Cross Validation Score to understand our model performance and have seen that our Cross-Validation Score is also negative and model is worse than Linear and Ridge Regression.
To understand better, we have plotted training and testing performance of Lasso Regression model in different folds and observed the performance of model and error as shown in Fig 18.

Fig 18:



As seen in Above Fig 18, The model performance is worse than Linear and Ridge regression in training and test dataset while checking model with different 15 Folds of cross validation. As we can infer from the plot that model has very high error in training and testing and R2 score also bad in training and testing dataset.

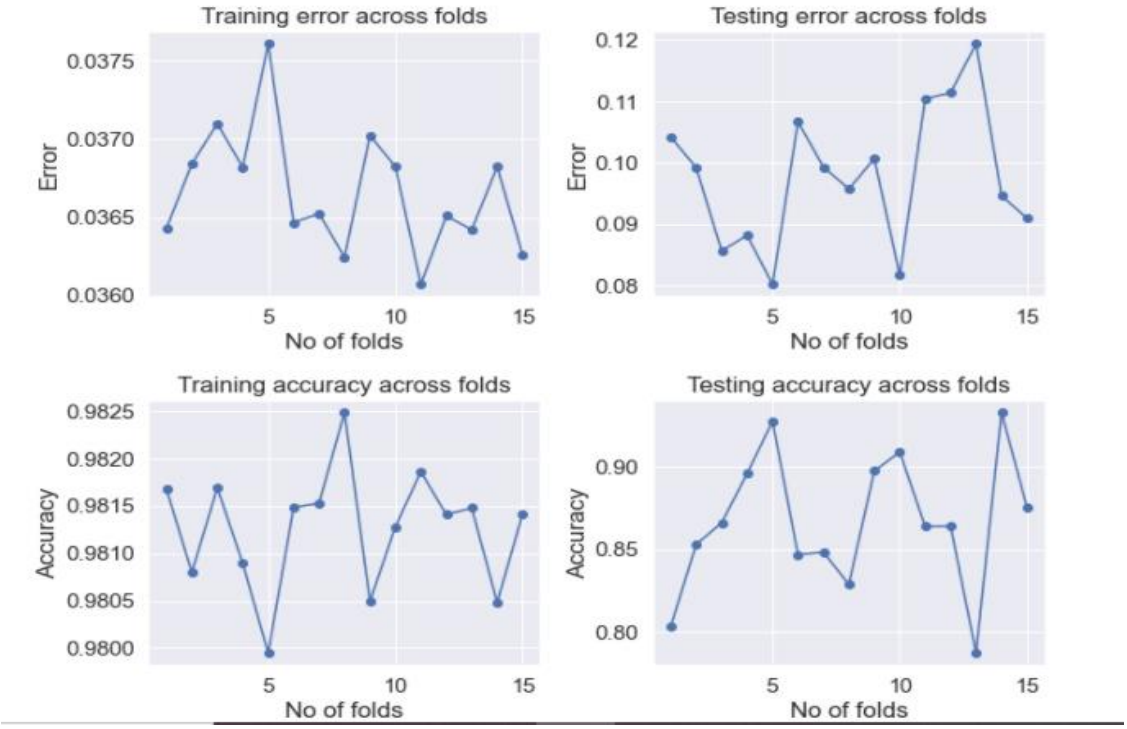
4. Random Forest Regressor

Evaluation Matrix:
Fig: 19

```
In [622]: pred_test=rfr.predict(x_test)

In [623]: MAE=mean_absolute_error(y_test,pred_test)

In [624]: MAE
Out[624]: 0.09277652779816971

In [625]: RMSE=np.sqrt(mean_squared_error(y_test,pred_test))

In [626]: RMSE
Out[626]: 0.1287904476289392

In [627]: rfr_score=r2_score(y_test,pred_test)

In [628]: rfr_score
Out[628]: 0.8800281987339206
```

As shown in Fig 19, we can see the Random Forest Regressor R2 Score is 88.00%.
MAE is 0.09
RMSE is 0.13
Also, we have calculated Cross Validation Score to understand our model performance and have seen that our Cross-Validation Score is 86.29%.
To understand better, we have plotted training and testing performance of Random Forest Regressor model in different folds and observed the performance of model and error as shown in Fig 20.

Fig 20:



As seen in Above Fig 20, The model performance is performing better than Linear Regression and Lasso Regression in training and test dataset while checking model with different 15 Folds of cross validation. As we can infer from the plot that model has high error compare to Ridge Regression and model R2 score is high in training than Testing dataset.

5. Decision Tree Regressor

Evaluation Matrix:
Fig: 21

```
In [639]: pred_test=dtr.predict(x_test)

In [640]: MAE=mean_absolute_error(y_test,pred_test)

In [641]: MAE
Out[641]: 0.13875569352367087

In [642]: RMSE=np.sqrt(mean_squared_error(y_test,pred_test))

In [643]: RMSE
Out[643]: 0.1865478497502966

In [644]: dtr_score=r2_score(y_test,pred_test)

In [645]: dtr_score
Out[645]: 0.7482946948923428
```

As shown in Fig 21, we can see the Decision Tree Regressor R2 Score is 74.83%.
MAE is 0.13
RMSE is 0.19
Also, we have calculated Cross Validation Score to understand our model performance and have seen that our Cross-Validation Score is 70.25%.
To understand better, we have plotted training and testing performance of Decision Tree Regressor model in different folds and observed the performance of model and error as shown in Fig 22.

Fig 22:

As seen in Above Fig 22, The model performance has high error rate in both training and testing and the R2 score also is 100% in training dataset where in testing R2 score is lesser while checking model with different 15 Folds of cross validation. Here the model is overfitting as model working too good in training dataset.

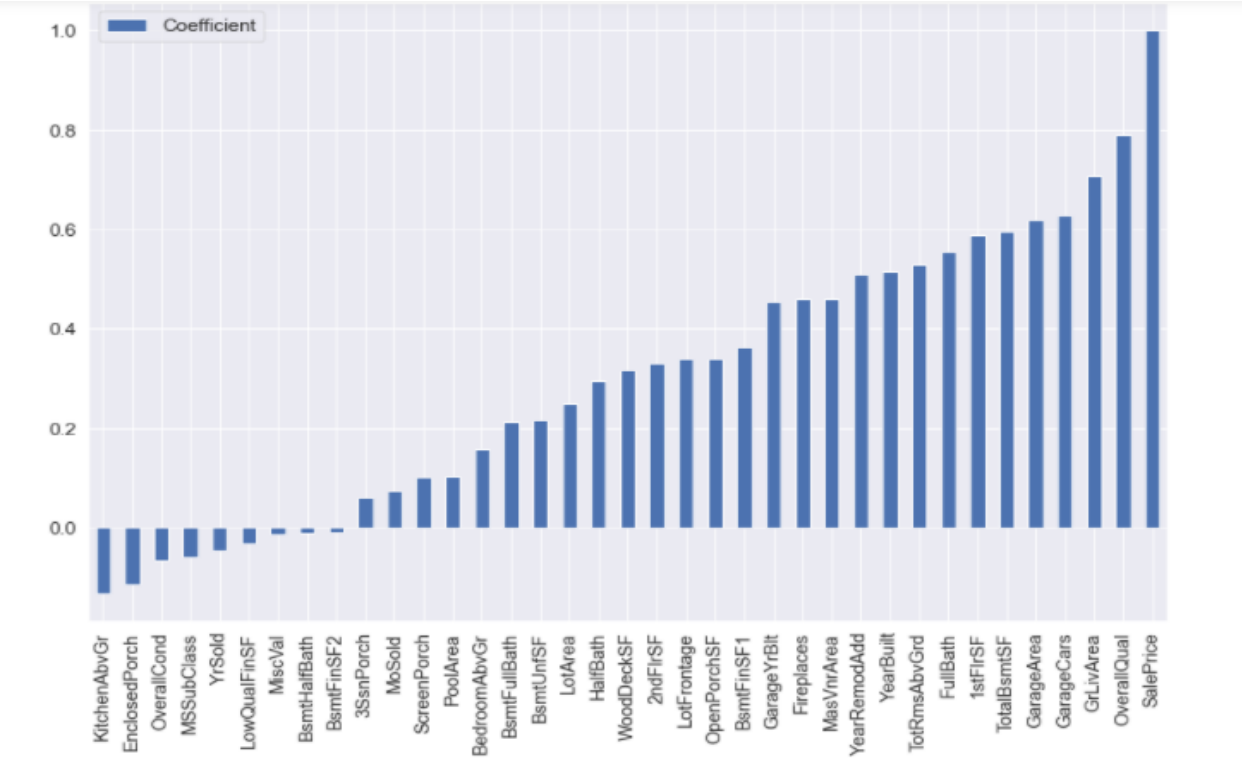
➢ Visualizations

In this Section we have visualized the features with various graph and plot to understand or analyse our features for our model.

- **Numerical Feature:**

As shown in Fig 23, we have found the most important Numerical features which affect the sale price positively and features which affect the sale price negatively.

Fig: 23

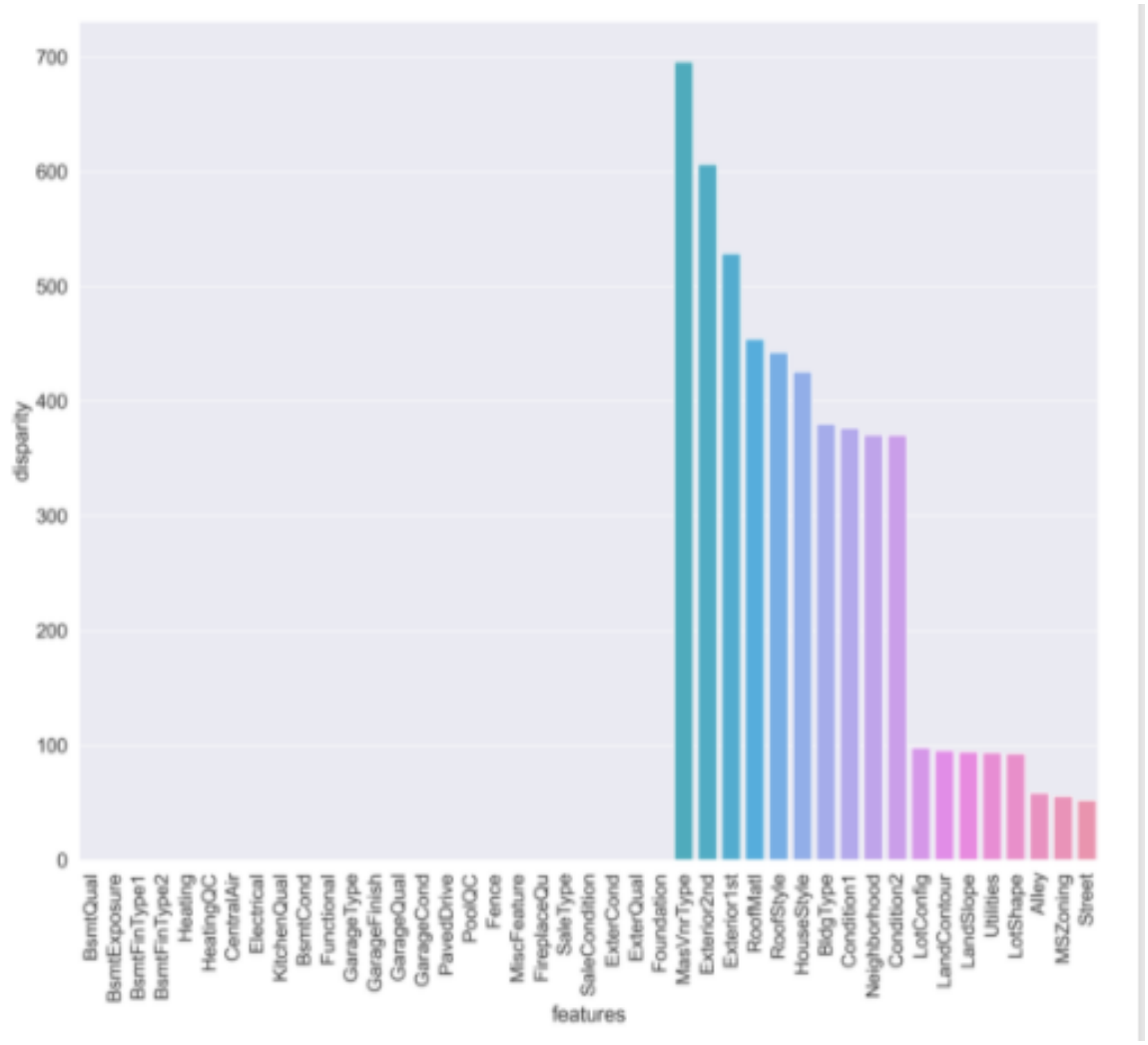


- **Categorical Feature:**

As shown in Fig 24, we have tried to find most important categorical variable with ANOVA technique.

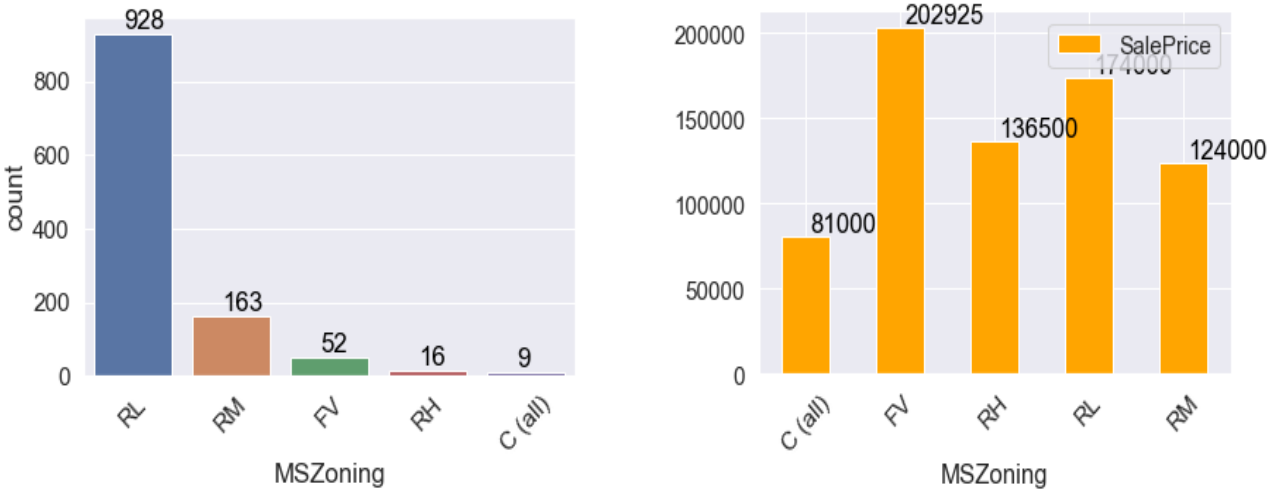
Fig 24:

```
In [211]: cat_data = train_df.select_dtypes(exclude=[np.number])

In [212]: cat = [f for f in train_df.columns if train_df.dtypes[f] == 'object']
          def anova(frame):
              anv = pd.DataFrame()
              anv['features'] = cat
              pvals = []
              samples = []
              for c in cat:
                      for cls in frame[c].unique():
                              s = frame[frame[c] == cls]['SalePrice'].values
                              samples.append(s)
                      pval = stats.f_oneway(*samples)[1]
                      pvals.append(pval)
              anv['pval'] = pvals
              return anv.sort_values('pval')

          cat_data['SalePrice'] = train_df.SalePrice.values
          k = anova(cat_data)
          k['disparity'] = np.log(1./k['pval'].values)
          plt.figure(figsize=(15,12),dpi=350)
          sns.barplot(data=k, x = 'features', y='disparity')
          plt.xticks(rotation=90)
          plt
```



- **MS Zoning**

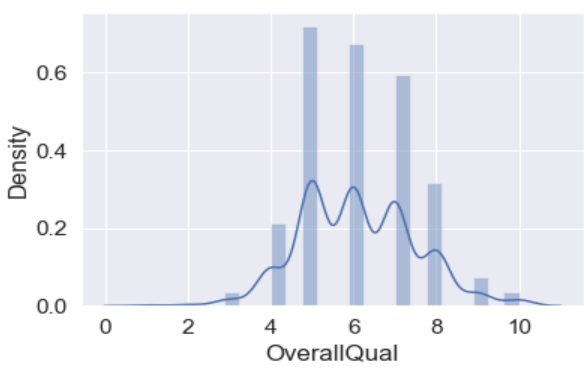
Fig: 25

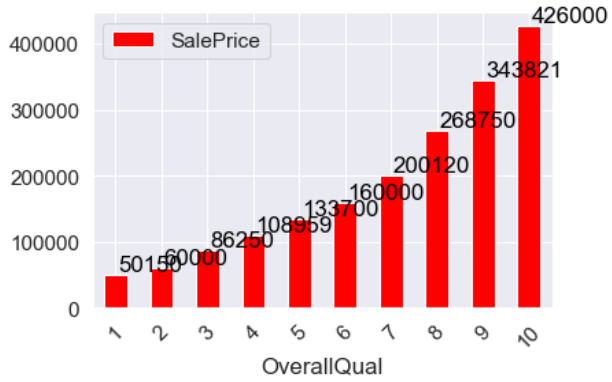


As we can infer from fig 25 that more houses are available in Residential Low-Density Zone. While checking the median value of Sale Price for different zone, we have found that the house median price is more which located at Floating Village Residential zone.

- **OverallQual:**

Fig: 26

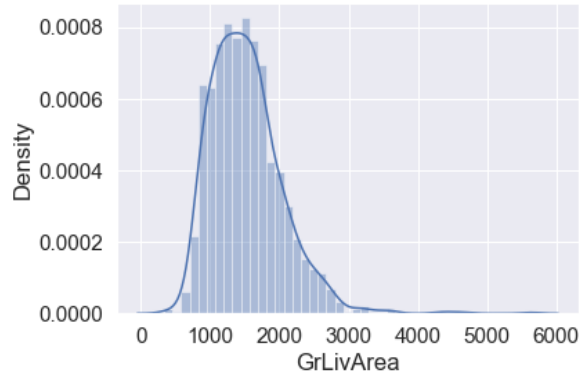


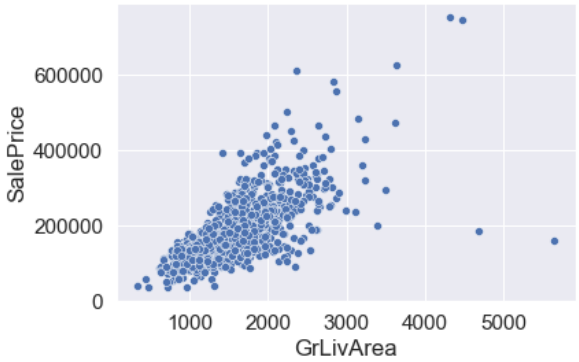


As we can infer from Fig 26, that Overall quality of house is 5,6 and 7 is high. And also, we can see that as the overall qual scale is increasing our house median price also increasing. So, the house prices would be high if the overall quality rating is high for the property.

- **GrLivArea:**

Fig: 27

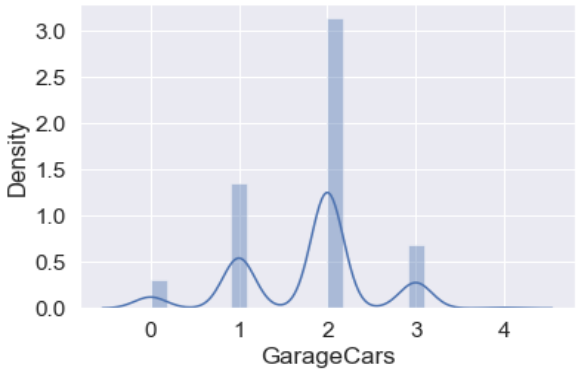


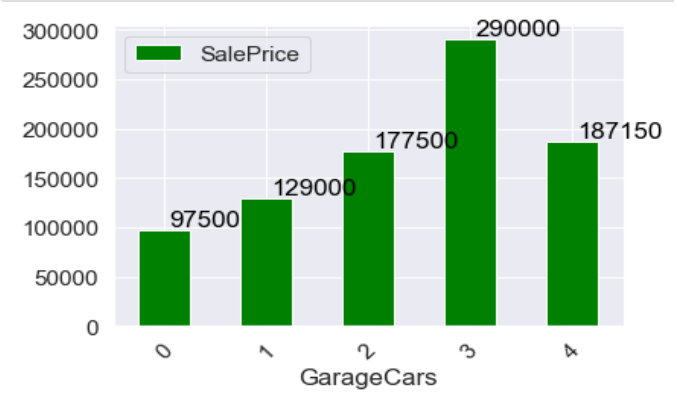


As we can infer from above fig 27, that living area square feet features data is looks like normal distribution but having some outliers and also, we can see the GrLivArea and sale prices having positive correlation as GrLivArea sq. feet increases the house sale price also increasing.

- **GarageCars:**

Fig: 28

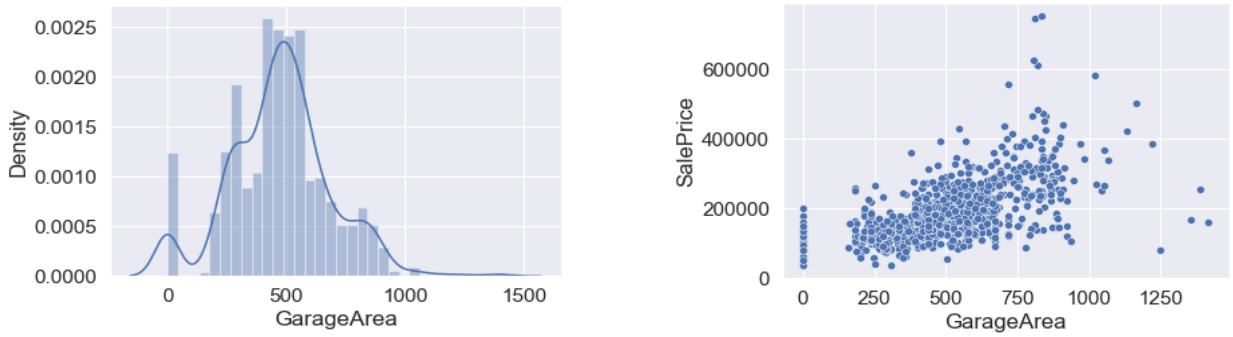




We can infer from above Fig 28, the maximum houses having garage for 1 and 2 cars and the median sale price of house is high if the house having garage capacity of 3 and 4 Cars.

- **GarageArea:**

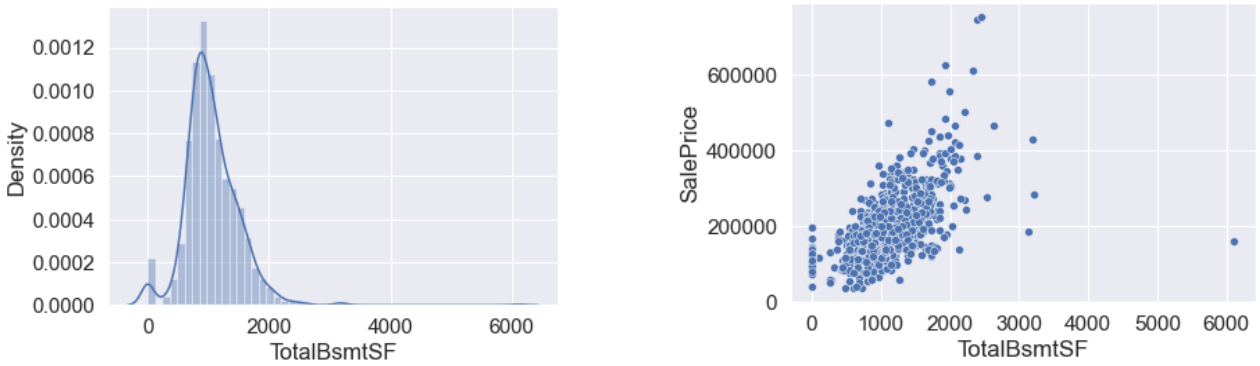
Fig: 29



We can infer from Fig 29 that maximum house having 400 to 600 sq. feet of Garage and GarageArea and Sale price having linear regression positive relationship.

- **TotalBsmtSF**

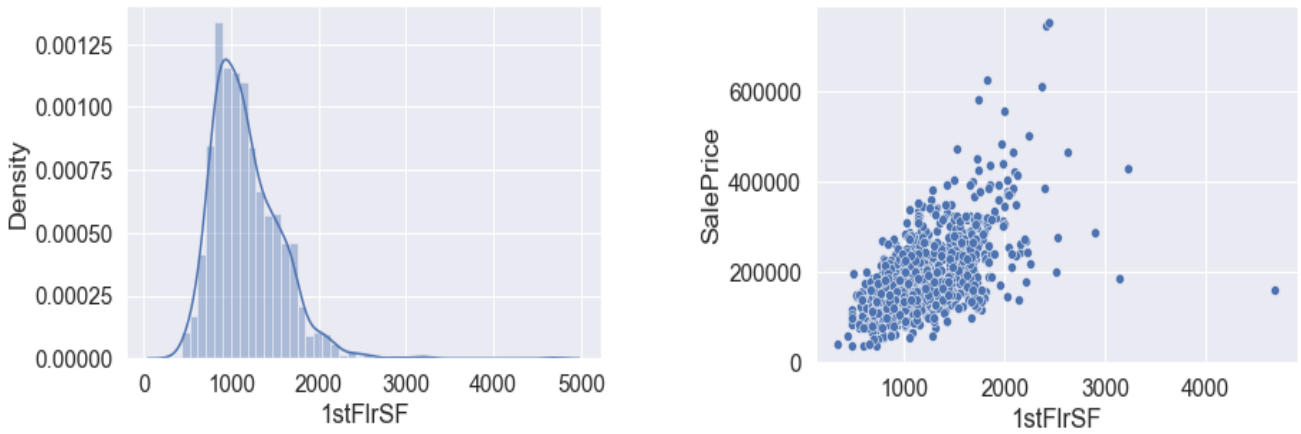
Fig: 30



We can infer from fig 30 that Total basement square feet area features data looks like normal distribution with having some outliers in that, where total area of basement is more but house sale price is very less. The TotalBsmtSF and Sale price having linear regression positive relationship.

- **1stFlrSF:**

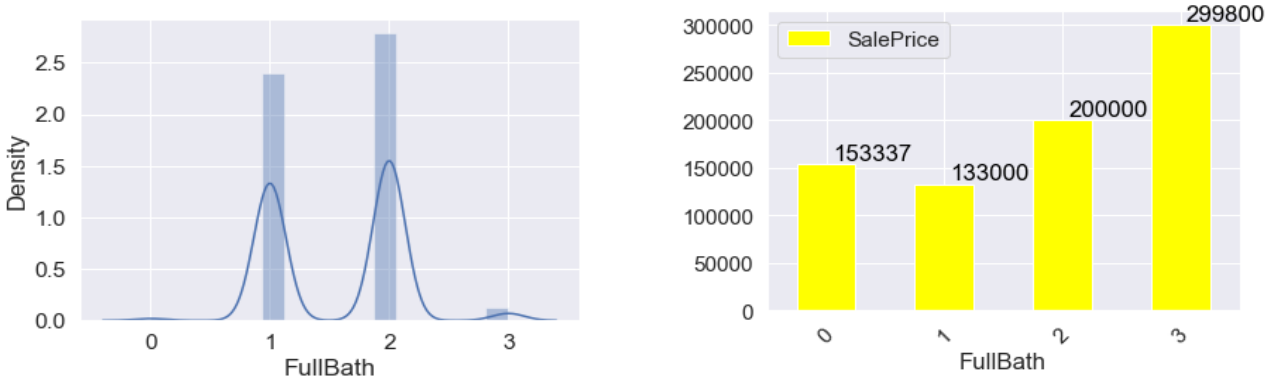
Fig: 31



We can infer from above Fig 31 that maximum house having 1st floor sq. feet area is from 900 to 1200 sq. ft. Sale price and 1stFlrSf having linear relationship.

- **FullBath :**

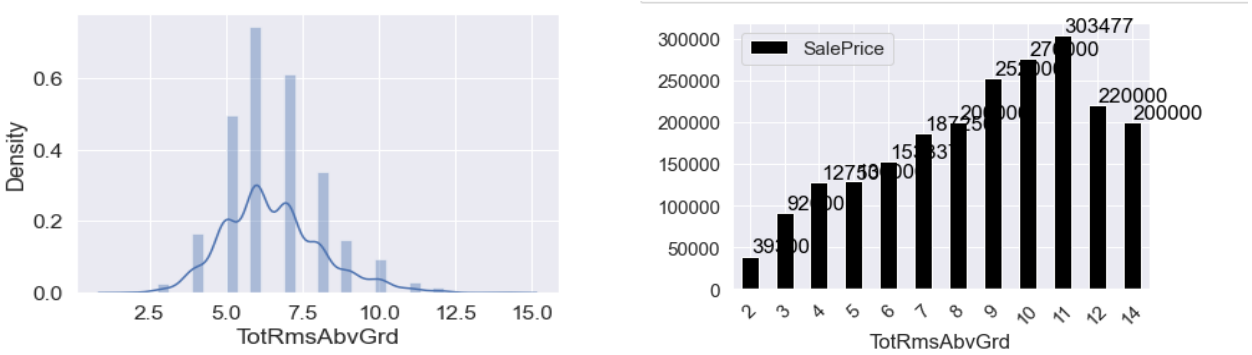
Fig: 32



We can infer from Fig 32 that maximum house having 1 and 2 full bathroom and the median of sale price of house is high if house have 2 or 3 bathrooms.

- **TotRmsAbvGrd:**

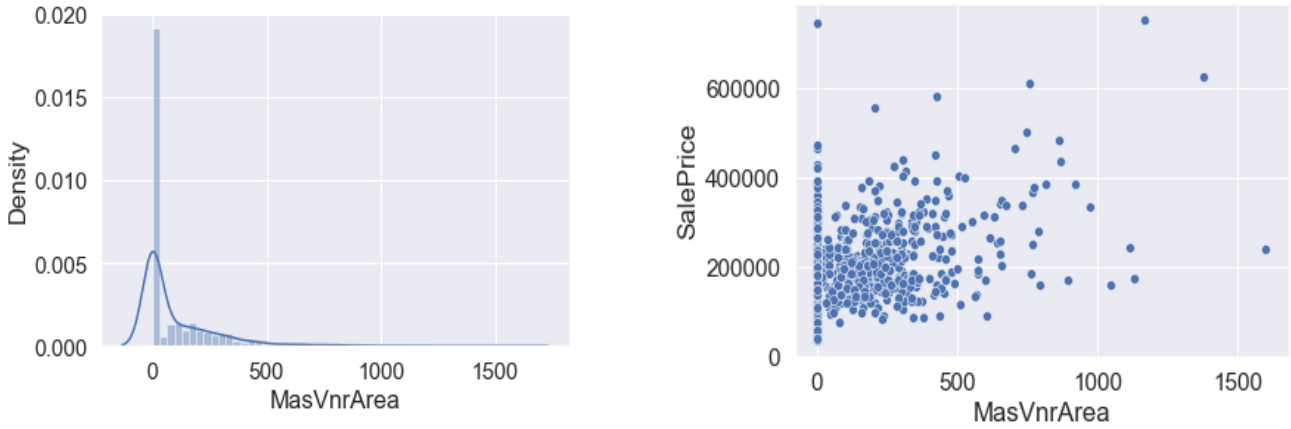
Fig: 33



We can infer from Fig 33 that the house has maximum room is 14 and minimum room is 2. The house sale price is high for those houses where total room is from 9 to 11 excluding bathroom.

- **MasVnrArea:**

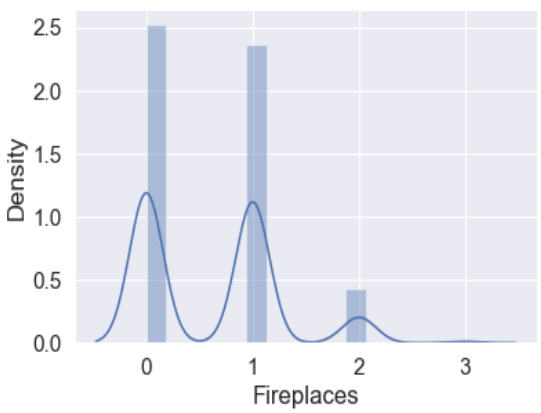
Fig: 34

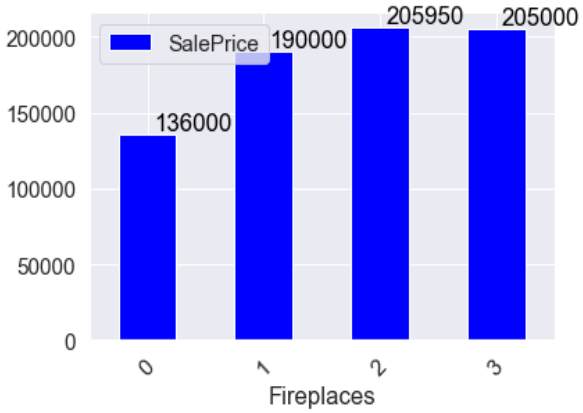


We can infer from Fig 34 that maximum houses have Masonry veneer area under 500 in square feet. There are lot of house where there is no Masonry veneer area but the sale price of house is relatively high. This data having lot of outliers so the feature data is right skewed.

- **FirePlaces:**

Fig: 35

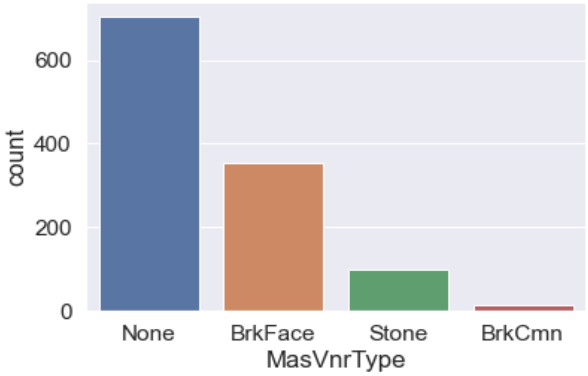


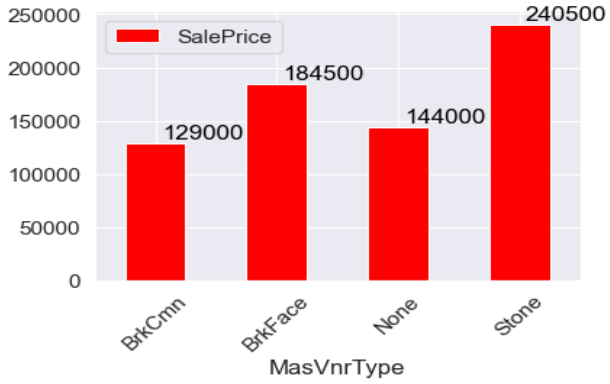


As we can infer from above fig that there are lot of house with 0 and 1 fireplaces and the house prices increases if the number of fireplaces increases.

- **MasVnrType:**

Fig: 36

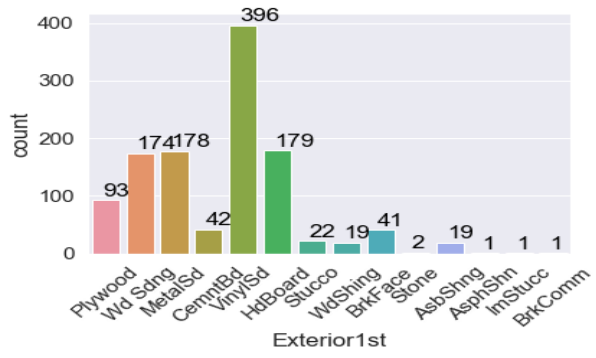


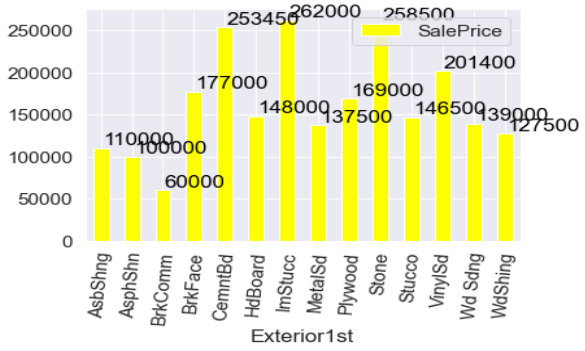


As we can infer from above fig that there are lot of house where no Masonry veneer is available. The house with Stone Masonry Veneer type having more sale price compared to Brick common, Brick Face.

- **Exterior1st:**

Fig: 37

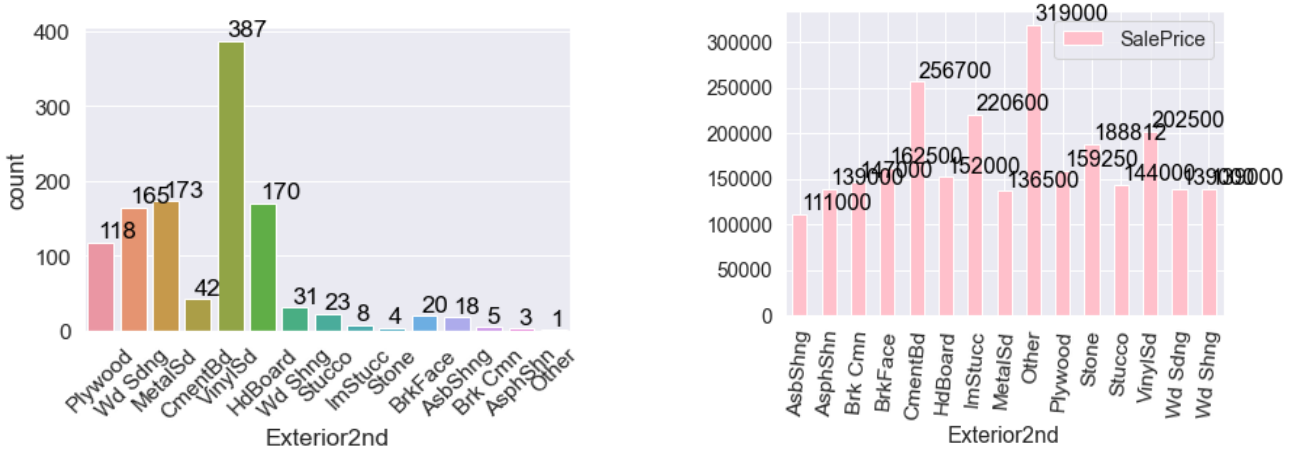




We can see in Fig 37, that maximum house's exterior is covering with Vinyl Siding, Hard Board, Wood Siding and Metal Siding. The avg price of house is more which has exterior covering with Cement Board, Imitation Stucco and Stone.

- **Exterior2nd:**

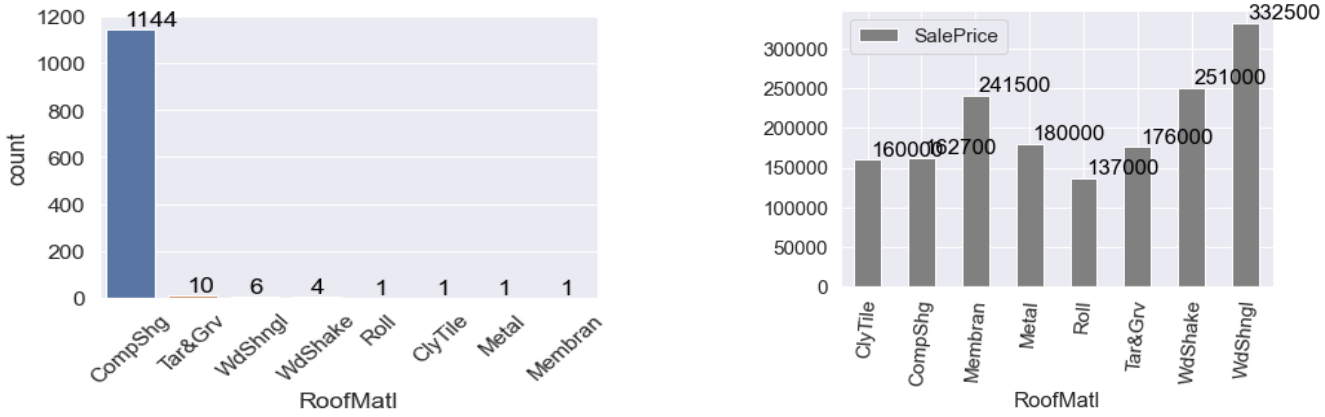
Fig: 38



We can infer from Fig 38 that as Vinyl Siding has used in Exterior as other material but the avg price of exterior covering is more for Cement Board, Imitation Stucco and Vinyl Siding.

- **RoofMatl:**

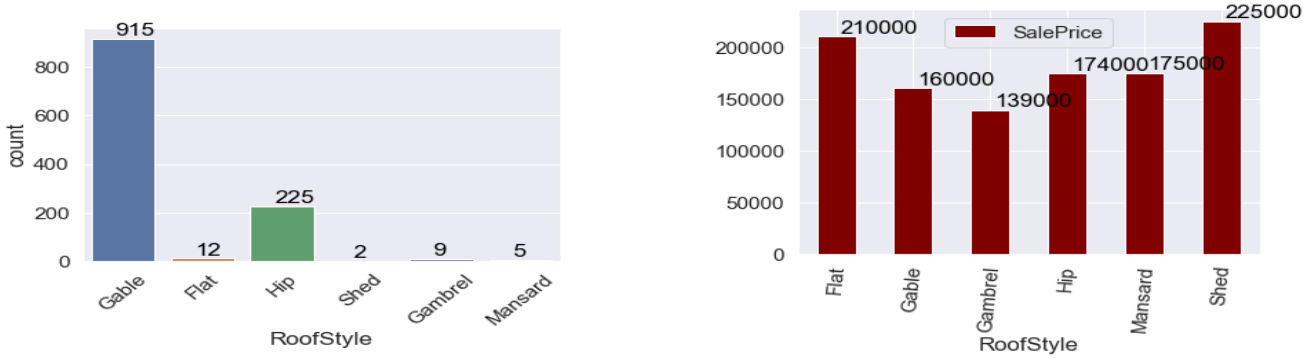
Fig: 39



As we can infer from Fig 39, that maximum houses has used Standard (Composite) Shingle as Roof Material and the prices of house is more if the Roof has made with Wood Shakes & Shingles.

- **RoofStyle:**

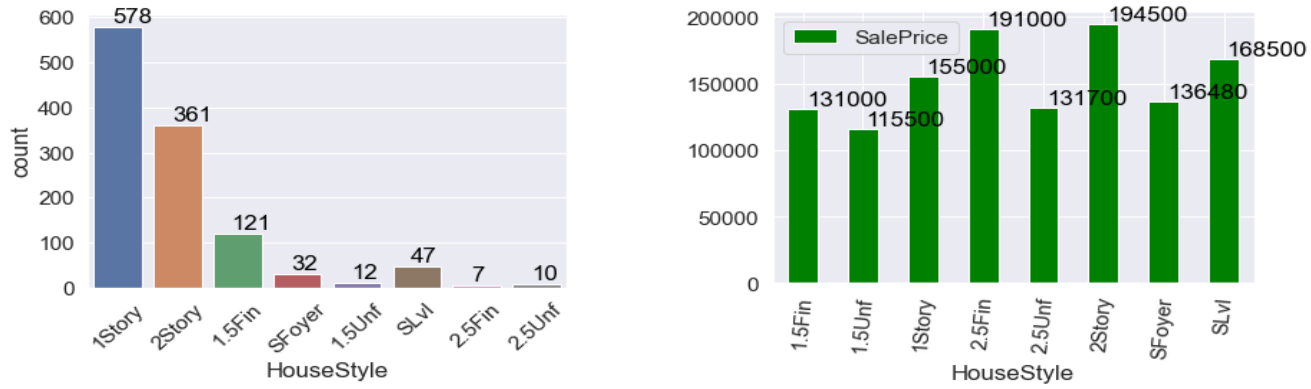
Fig: 40



As we can infer from above fig that maximum house having Gable type roof style and the house prices is more if the roof style are Shed and Flat.

- **HouseStyle:**

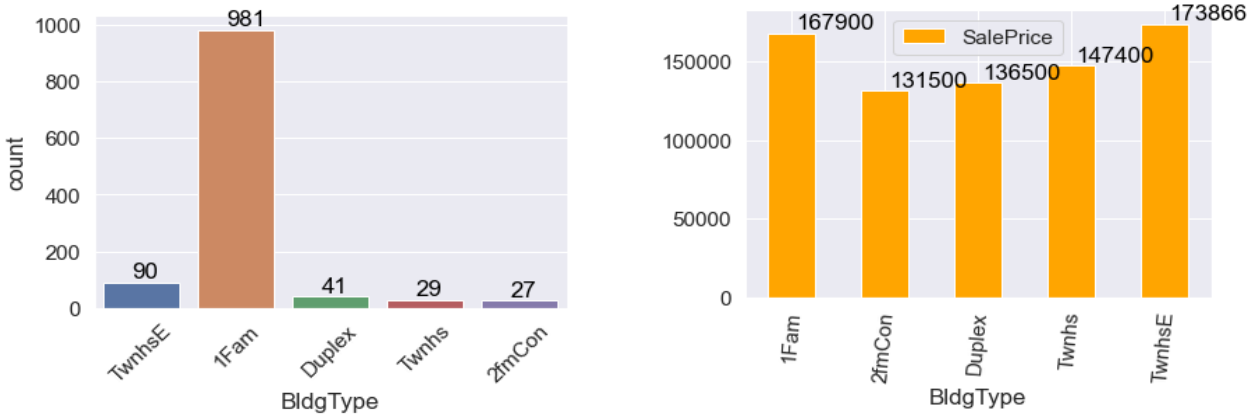
Fig: 41



We can infer from above graph that that 1Story type of houses are maximum followed with 2 Store type. The house prices are more if the house style is two and half Store type and 2 store type.

- **BldgType:**

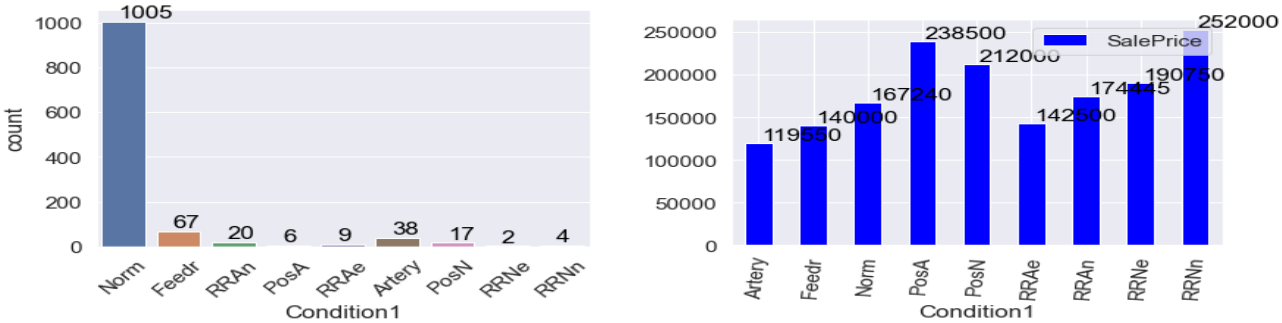
Fig: 42



From the above plot we can see that Single-family Detached type of dwelling house is maximum and the prices of house is more if the house have dwelling type Single family Detached and Townhouse End Unit.

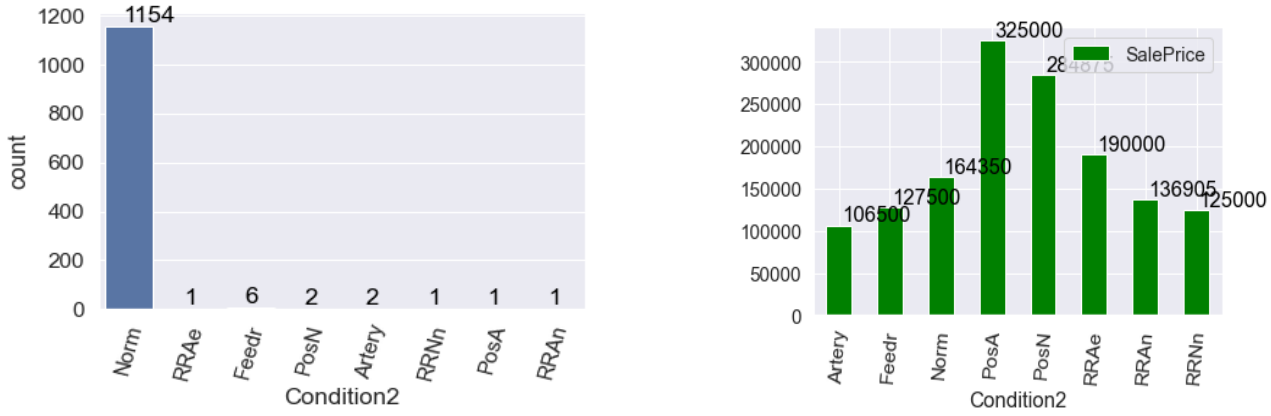
- **Condition1:**

Fig: 43



- **Condition2:**

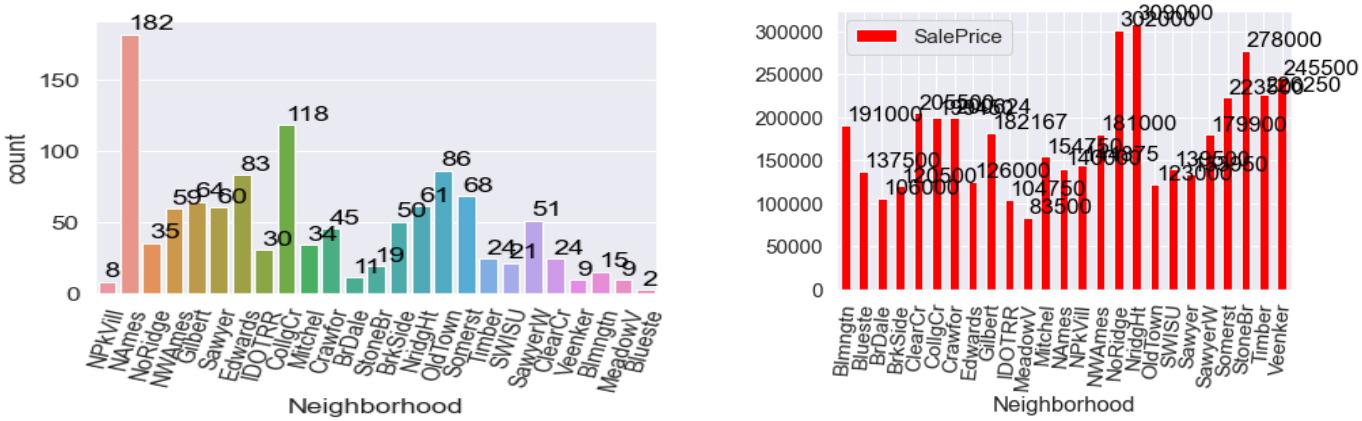
Fig: 44



We can infer from above Fig 43 and 44 that the condition of maximum houses is normal and if the houses is located Adjacent to positive off-site feature then the house price is more.

- **Condition1:**

Fig: 45



As we can infer from Fig 45 that maximum housed is located at Northwest Ames and College Creek. The price of house is more if is located at Northridge and Northridge Heights.

➢ Interpretation of the Results

Below are the interpreted from the visualizations and modelling:

✓ We have seen that R2 score with compare to CV Score with different 15 k fold and MAE error is providing good performance in Ridge Regression Model compare to others applied model like linear, tree based, ensemble etc.as shown in Fig 46.

Fig 46:

Out[604]:

	R2 Score	Cross Validation Score	Difference	MAE
Linear_Regression	0.91	-1.177315e+24	1.177315e+24	0.082622
Ridge_Regression	0.91	8.600000e-01	5.000000e-02	0.082463
Lasso_Regression	-0.01	-1.000000e-02	0.000000e+00	0.288509
Random_Forest_Regression	0.88	8.600000e-01	2.000000e-02	0.090164
Decision_Tree_Regressor	0.73	7.200000e-01	1.000000e-02	0.142268

✓ As shown in Fig 23, we have seen the most important Numerical Features which impact on housing price in positive or negative way are listed below (Fig 47) for top 15 features and bottom 5 features.

Fig 47:

```
OverallQual     0.789185
GrLivArea       0.707300
GarageCars      0.628329
GarageArea      0.619000
TotalBsmtSF     0.595042
1stFlrSF        0.587642
FullBath        0.554988
TotRmsAbvGrd    0.528363
YearBuilt       0.514408
YearRemodAdd    0.507831
MasVnrArea      0.460535
Fireplaces      0.459611
GarageYrBlt     0.453840
BsmtFinSF1      0.362874
Name: SalePrice, dtype: float64
**********************************************
YrSold          -0.045508
MSSubClass      -0.060775
OverallCond     -0.065642
EnclosedPorch   -0.115004
KitchenAbvGr    -0.132108
Name: SalePrice, dtype: float64
```

✓ As shown in Fig 24, the most important Categorical Features which impact on house price prediction. Below are the categorical Features shown in Fig 48 is the important features for model prediction.

Fig 48:

```
MasVnrType
Exterior2nd
Exterior1st
RoofMatl
RoofStyle
HouseStyle
BldgType
Condition1
Neighborhood
Condition2
LotConfig
LandContour
LandSlope
Utilities
LotShape
Alley
MSZoning
Street
```

CONCLUSION

➢ Key Findings and Conclusions of the Study

- ✓ We have shown in Fig 15; Ridge Regression is performing better compared to Linear Regression, Lasso Regression, Random Forest Regressor, Decision Tree Regressor.
- ✓ While doing Principal Components Analysis, as shown in Fig 9 we can go with 90 number of components to get more than 90% of variance explained, but we have tried the model performance without PCA applied in model.
- ✓ While doing visual analysis and under EDA we have got like the some of the most important feature for prediction of housing price are OverallQual, GrLivArea, GarageCars, GarageArea, TotalBsmtSF, 1stFlrSF, Full Bath, TotalRmsAbvGrd, MasVnrType, Exterior2nd, Exterior1st, RoofMatl, RoofStyle, HouseStyle, BldgType etc as shown in Fig 47 and Fig 48.
- ✓ We have Confirmed that based on R2 Score, CV Score and MAE error Ridge Regression Model is performing better.
- ✓ We have seen other model R2 Score also is good like Linear Regression, Random Forest Regressor but have seen that model having Overfitting issue So we have done the Regularization with Ridge Regression.

➢ Learning Outcomes of the Study in respect of Data Science

Below are the Learning Outcome of the Study:

- ✓ Handling huge number of Feature more than 80.
- ✓ Having more features and through PCA can select significant features only for model.
- ✓ Feature data pattern are not normally distributed.
- ✓ Having lots of Outliers and dealing with each feature outlier will consume lot of time so we have done features transformation with Standard Scaler technique.
- ✓ Target variable are not normally distributed as data was right skewed which we have rectified by doing log transformation.
- ✓ Selecting the Best Model based on R2 Score, CV Score and MAE error.

➢ Limitations of this work and Scope for Future Work

Mention below are the limitations and future scope steps:

- ✓ Data set having more features we can do more feature engineer with combining various features and could see the impact on price target variable.
- ✓ We have seen that model having issue like overfitting or underfitting so we have applied regularization technique, we could apply PCA and analyse the applied model performance.