HOUSING: PRICE PREDICTION

Submitted By:

Mirza Irshadbaig Ismailbaig

Introduction

Business 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 are 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. The real estate industry has been one of the leading researches focusing on modern economics, for its significant implications on relevant industries and fields such as construction, investment, and public welfare. In general, purchasing and investing in any real estate project will involve various transactions between different parties. Thus, it could be a vital decision for both households and enterprises. How to construct a realistic model to precisely predict the price of real estate has been a challenging topic with great potential for further research. Large Real estate companies have various products they need to sell and they have to assign people to handle each of these products. This again bases the prediction of a price tag on a human hence there is room for human error. Additionally, these assigned individuals need to be paid. However, having a computer do this work for you by crunching the heavy numbers can save a lot of time money and provide accuracy which a human cannot achieve.

Domain Problem:

When people first think of buying a house they tend to go online and try to study trends and other related stuff. People do this so they can look for a house which contains everything they need. However, the common customers don't have detailed knowledge and accurate information about what the actual price should be. This can lead to misinformation as they believe the prices mentioned on the internet to be authentic. While searching for a property is to contact various Estate agents. These agents need to be paid a fraction of the amount just for searching a house and setting a price tag for customer. In most cases, this price tag is blindly believed by people because they have no other options.

Review of Literature:

HOUSING: PRICE PREDICTION involves data from US based housing company Surprise Housing which contains 80 variables where it is required to analyze which indicators are effective in predicting prices of house in Australia. The analysis made in project on basis of requirements. The prediction of price have to be done with 1168 data instance, and finding key indicators of price s of house. The project involves, analysis of missing records, imputation with meaningful entities, detecting visualizing distribution of features, detecting outliers and treatment of outliers with data loss consideration. The analysis of skewness to reduce skewenss in data with transformation techniques. The correlation analysis, multicollinearity analysis and feature selection methods are employed to get most effective indicators and to increase interpretability of prediction of prices. The scaling and PCA has to perform trained model with highly correlated components. The training involves ensemble technique or algorithms like XGboostRegressor, Gradient boosting Regreessor, Random Forest regressor and decision tree and SVR which works well on linear as well as non-linear task. The XGboost regressor out-performed best in all metrics of evaluation among all algorithms and finalize as best model to predict prices with hyper parameter tuning.

Analytical Problem Framing

- Analytical and Statistical modeling of problem

A] Descriptive analysis: Descriptive statistics describe, show, and summarize the basic features of a dataset found in a given study, presented in a summary that describes the data sample and its measurements. It helps analysts to understand the data better. The given set of data involves 79 variables in which study of each variable is necessary to know whether it will help in prediction of sale price of house. Also it is important to know data type of values, weather these features are continuous features or categorical features, which crucial step in preprocessing. It is done with done with some methods like describe(), info(), dtype() on dataframe.

Observations:

1] Id is variable which index values of house with respect to instances are.
2] The descriptive stats give an overview of extreme values of data and nature of data and also give idea of unrealistic values observed in variables with domain knowledge application. 3] descriptive analysis will also informed about missing records in features which has to be treated before prediction
4] The continuous variables are mostly observed right skewed which is very crucial analysis while prediction with machine learning.

B] Missing records analysis : The missing records detection and treatment and detection is one of the essential part of machine learning .Because the scikit learn Library doesn't works with missing records.

1] The project involves employment of pandas and visualization tools like matplotlib and seaborn for detection of missing records . There are 18 features with 5558 missing cells in whole data.

2] The data set has 1168 instances and 80 variables. All variables are analyzed with tools to detect missing records where 18 variables have missing records .within which variables 05 variables have more than 20% missing records and remaining have less than 20% records as Nan.

3] During missing records analysis , it has been detected that there is relationship between missing records in data with other data variables . Hence these missing records are missing At Random type upon random value imputation leads to biased estimate of sale price of house in predictive model. It's crucial to interpret the relationship between related variables before imputation.

C] Unnecessary Features : The features variance and distribution has great impact on prediction of response variable. There is also impact on correlation and covariance with other features ad response variables.

Project involves analysis of with pandas methods like unique() and value_counts() for analysis of categories and their distribution .

D] Univariate Analysis :

Graphical

a] Distribution of continuous Variables : In Machine Learning, data satisfying Normal Distribution is beneficial for model building. Normality is an assumption for the ML models. It is not mandatory that data should always follow normality. There are 34 continuous variables in data with target variable.

Project involves analysis distribution with matplotlib and Distplot() in seaborn for all 34 variables .Analysis is done detect variables which have normal-like distribution which are non-normal.

b] Distribution of Categorical variable: The count plot can be thought of as a histogram across a categorical, instead of quantitative, variable. The distribution of categorical plot is necessary for analyzing contribution of categories in feature as well its effect on response variable Sale price of House. There are 46 variables in data having categories.

c] Outliers detection with Boxplot: Outliers are unusual values in your dataset, and they can distort statistical analyses and violate their assumptions. Removing outliers is legitimate only for specific reasons like values are unrealistic and after analyzing data loss after outliers removal. The **boxplot()** is combination graphical and statistical method of Seaborn library in which box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be "outliers. The 34 continuous variables has been analyzed with boxplot and preprocessing techniques has been applies on basis of observations made with it.

Non Graphical

Outliers detection with Z score method : A Z-score is a numerical measurement that describes a value's relationship to the mean of a group of values. Z-score is measured in terms of standard deviations from the mean. If a Z-score is 0, it indicates that the data point's score is identical to the mean score. 1] After computation of data the outliers can be detected with empirical rule of z distribution. The 68% of data lies between 1 Standard deviation of data. The 95% of data lies between 2 standard deviation of data. The 99.7% of data lies between 3 standard deviation of data. The outliers are detected with z value more the 3 standard deviation in respective continuous features. All rows which consist of outliers are simply eliminated from dataset permanently. **Data loss:** After elimination of instance from data can lead to elimination of use full information or crucial information from data, Hence outliers removal is not recommended technique always. It's advised to analyze for data loss and interpretation of outliers instances before removal. If large amount of data is eliminated from dataset isn't good method. It is possible to build model with outliers with some algorithms Z score method detected up to 30% of data instances as Outliers.

b] Outliers detection with IQR method : IQR is used to measure variability by dividing a data set into quartiles. The data is sorted in ascending order and split into 4 equal parts. Q1, Q2, Q3 called first, second and third quartiles are the values which separate the 4 equal parts. IQR is the range between the first and the third quartiles namely Q1 and Q3: *IQR = Q3 – Q1*. The data points which fall below Q1 – 1.5 IQR or above Q3 + 1.5 IQR are outliers the data instances which consist of outliers are eliminated from dataset.
Data loss: IQR method detected up to 57% of data instances as Outliers, which cannot employed for outliers elimination.

c] Skewness Analysis : Skewness is a quantifiable measure of how distorted a data sample is from the normal distribution. In normal distribution, the data is represented graphically in a bell-shaped curve, where the mean and mode are equal. The goal is to reduce skewness to get as close as possible to a normal distribution by using transformations.

The analysis of skewness is performed with skew() method in pandas to know which variables have highly skewed data. Skeweness value more than 1 is considered as highly skewed data. The some continuous feature have high skewness value which has to be treated with transformation like logarithm, square roots, cube root, Power transformation and quintile Transformation of all the data points in continuous variables. The analysis is made with all above mentioned transformation and best and effective transformation is selected for preprocessing which Power transformation with yeo-johnson method from sklearn library.

F] Power Transformation : Power transforms are a family of parametric, monotonic transformations that are applied to make data more Gaussian-like. This is useful for modeling issues related to heteroscedasticity (non-constant variance), or other situations where normality is desired. Currently, Power Transformer supports the Box-Cox transform and the Yeo-Johnson transform. The optimal parameter for stabilizing variance and minimizing skewness is estimated through maximum likelihood. Box-Cox requires input data to be strictly positive, while Yeo-Johnson supports both positive and negative data.

Power transformation can expressed mathematically as below:

$$\psi(y,\lambda) = \begin{cases} \dfrac{(y+1)^{\lambda}-1}{\lambda} & y \geq 0 \text{ and } \lambda \neq 0, \\ \log(y+1) & y \geq 0 \text{ and } \lambda = 0, \\ -\dfrac{(-y+1)^{2-\lambda}-1}{2-\lambda} & y < 0 \text{ and } \lambda \neq 2, \\ -\log(-y+1) & y < 0, \lambda = 2. \end{cases}$$

In above expression, the value of λ is chosen via maximum likelihood estimation.

F] Multivariate Analysis :

a] Analysis with scatterplot : Scatter plots shows how much one variable is affected by another or the relationship between them with the help of dots in two dimensions. It is very important method to analyze. The scatter diagram is for analyzing effect of continuous variables **LotFrontage, LotArea,YearBuilt,YearRemodAdd,MasVnrArea,BsmtFinSF1,BsmtFinSF2','BsmtUnfSF',TotalBsmtSF,1stFlrSF,2ndFlrSF,LowQualFinSF,GrLivArea,BsmtFullBath,BsmtHalfBath,FullBath,HalfBath,BedroomAbvGr,KitchenAbvGr,Fireplaces,TotRmsAbvGrd,GarageYrBlt,GarageCars,GarageArea,WoodDeckSF,OpenPorchSF,EnclosedPorch,3SsnPorch, ScreenPorch,PoolArea,MiscVal,MoSold,YrSold** on SalePrice variable. And also for analyzing relationship among independent variables which violates assumption of algorithms.

b] Analysis with Violin plot: Violin plots are used to visualize data distributions, displaying the range, median, and distribution of the data. Violin plots show the same summary statistics as box plots, but they also include Kernel Density Estimations that represent the shape/distribution of the data. The violin plots are used in House price prediction for analyzing the distribution with categories in categorical variables in dataset.

c] Analysis with correlation matrix : Correlation is an indication about the changes between two variables.. The correlation matrix involves calculation of Pearson's correlation coefficient for every variable to variable pair . The values more than 0.5 indicates that moderate to high level of correlation between pair of variables. There are 78

variables with target. Correlation matrix returns correlation coefficient for all variables to variables pair . It manly signifies multicollinearity and correlation of indicators with target feature.

d] Feature selection with SelectKBest and f_regression : The SelectKBest method selects the features according to the k highest score. The scoring function used is f_regression which will rank features in the same order if all the features are positively correlated with the target. The p-value and scores for each term tests the null hypothesis that the coefficient within feature and target variable is equal to zero (no effect). The higher p-values indicate low correlation while lower indicates higher correlation. All the features in dataset are trained with target variable Sale price and Score and P values are stored in dataframe for analysis. The Best features will be selected on basis of probabilities obtained are less than significance level of 0.01 for null hypothesis to be true. The variables with p values lesser than 0.01 have significant relationship with target variable sale Price of house.

e] Multicollinearity with VIF: VIF determines the strength of the correlation between the independent variables. It is predicted by taking a variable and regressing it against every other variable. VIF score of an independent variable represents how well the variable is explained by other independent variables.

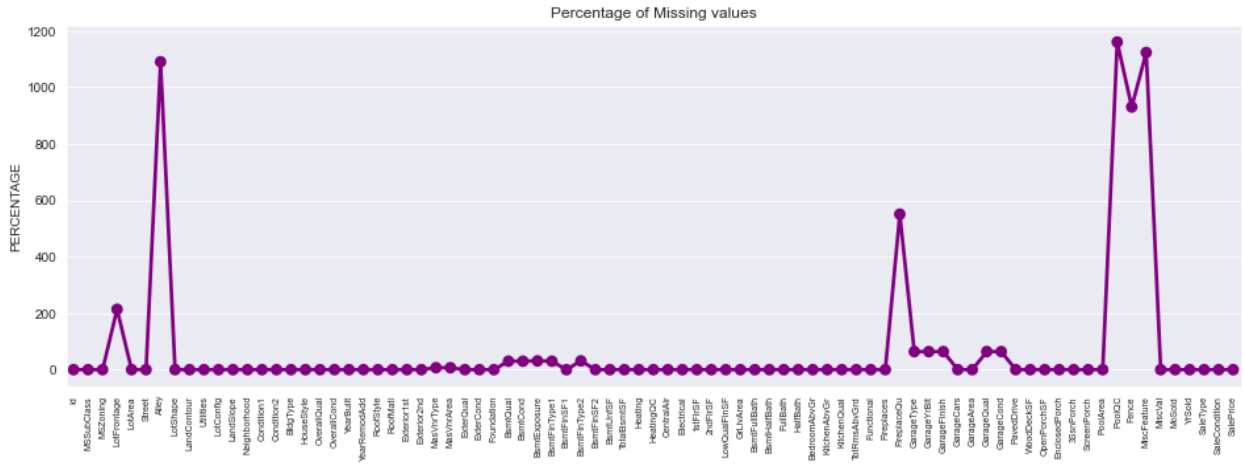
Project involves assessing vif scores for each variable. The variance inflation factor for every input variable is stored in dataframe . The threshold for vif score is decided as 10 as high scores of vif are observe in features in Housing price indicators. The features having score above 10 will be eliminated to improve the interpretability of model.

- ## Data preprocessing Techniques

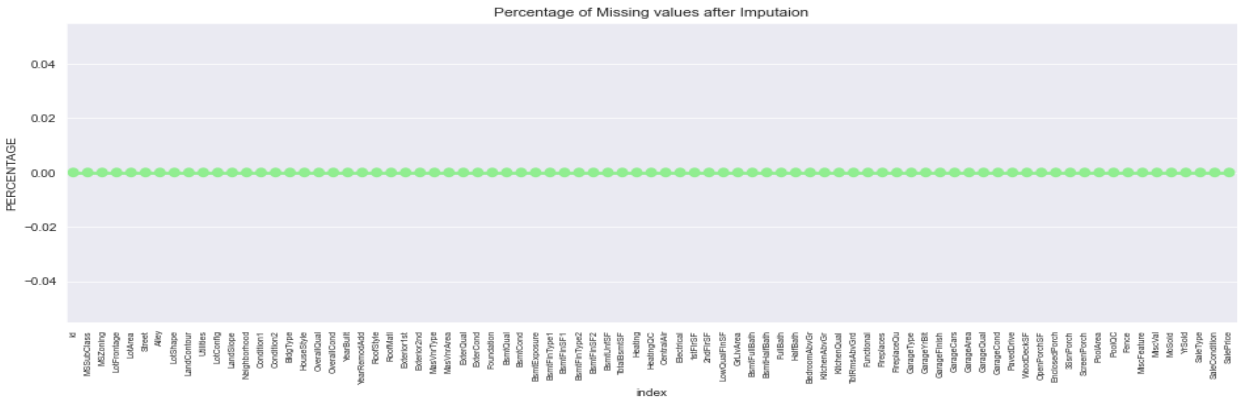
Upon Analysis with all above statistical and mathematical techniques the below observations can be made:

1] The Housing price prediction involves price prediction which is continuous in nature.so the preprocessing techniques must be employed in accordance.
2] Imputation of missing records: Upon analysis of relation, 90% of missing records are categorical which can be categorize as not applicable for respective features hence are imputed with NA or None type categories which are also defined in data description defined by customer.

Example:-The features BsmtCond, BsmtQual, BsmtExposure, BsmtFinType1, BsmtFinType2 are related to Basement .The missing values are observed at TotalBsmtSF is equals to 0 which indicates that no basement hence All above variables will be not applicable NA. Such observations are available in all categorical features.



After Imputation:



3] Upon analysis, it is observed that (a) Id feature is index like feature and which has no predictive power as its unique value for unique instance.
(b) Utilities feature involves only one category in train data for all instances while it has more categories as defined in data description file but these are absent for all instances. (c) Id and Utilities features have eliminated from data to avoid over fitting and confusion to algorithms while training.

4] Outliers treatment: Upon outliers detection with graphical and non-graphical analysis it can be concluded that there are outliers in all continuous variables in housing price prediction data. But these data values are not unrealistic or wrong values. These values are due to natural variation in data variables. The data loss observed with outliers elimination with both Z score and IQR method is up to 30 % and Up to 57% respectively. Due to mentioned analysis, it is decided to train the model with True values and without any elimination of outliers from data, which can cause loss crucial information from data.

5] Encoding categorical data : Sklearn provides a very efficient tool for encoding the levels of categorical features into numeric values. LabelEncoder encode labels with a value between 0 and n_classes-1 where n is the number of distinct labels. Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning. Object data type variables are encoded to numeric values with LabelEncoder in sklearn library. There are 42 variables in Housing Price prediction data which have been encoded with **LabelEncoder.**

```
In [73]: df.head()
```

Out[73]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	LotConfig	LandSlope	...	PoolArea	PoolQC	Fence	MiscFeature	Mis
0	120	RL	70.98847	4928	Pave	NA	IR1	Lvl	Inside	Gtl	...	0	NA	NA	NA	
1	20	RL	95.00000	15865	Pave	NA	IR1	Lvl	Inside	Mod	...	0	NA	NA	NA	
2	60	RL	92.00000	9920	Pave	NA	IR1	Lvl	CulDSac	Gtl	...	0	NA	NA	NA	
3	20	RL	105.00000	11751	Pave	NA	IR1	Lvl	Inside	Gtl	...	0	NA	MnPrv	NA	
4	20	RL	70.98847	16635	Pave	NA	IR1	Lvl	FR2	Gtl	...	0	NA	NA	NA	

5 rows × 79 columns

After Encoding with LabelEncoder:

```
In [78]: df.head()
```

Out[78]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	LotConfig	LandSlope	...	PoolArea	PoolQC	Fence	MiscFeature	Mis
0	120	3	70.98847	4928	1	1	0	3	4	0	...	0	3	4	1	
1	20	3	95.00000	15865	1	1	0	3	4	1	...	0	3	4	1	
2	60	3	92.00000	9920	1	1	0	3	1	0	...	0	3	4	1	
3	20	3	105.00000	11751	1	1	0	3	4	0	...	0	3	2	1	
4	20	3	70.98847	16635	1	1	0	3	2	0	...	0	3	4	1	

5 rows × 79 columns

6] Transformation: The skewness has been detected in continuous variables which have been analyzed with different transformation techniques. Among all these above techniques mentioned in analytical modeling section, The Power transformation works better than all, Hence All values continuous variables except Target variable Sale Price in Housing price prediction are transformed with Power Transformation which involves method called Yeo-Johnson. The continuous variables are transformed from non-normal distribution to somewhat like normal distribution. The Transformation can be observed in given graph below

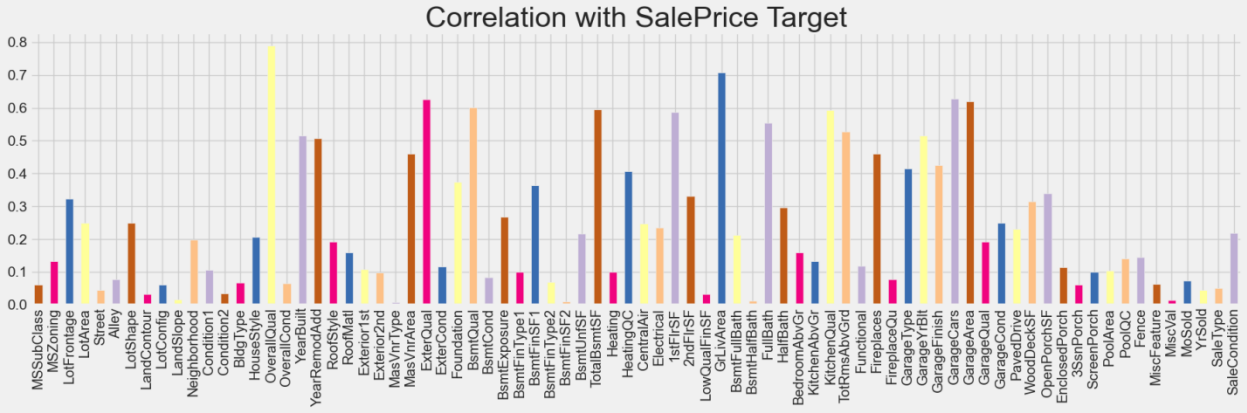

Before Transformation
After Transformation
Density
LotFrontage
LotArea
YearBuilt
YearRemodAdd
MasVnrArea
BsmtFinSF1
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
1stFlrSF
2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
Fireplaces
TotRmsAbvGrd
GarageYrBlt
GarageCars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

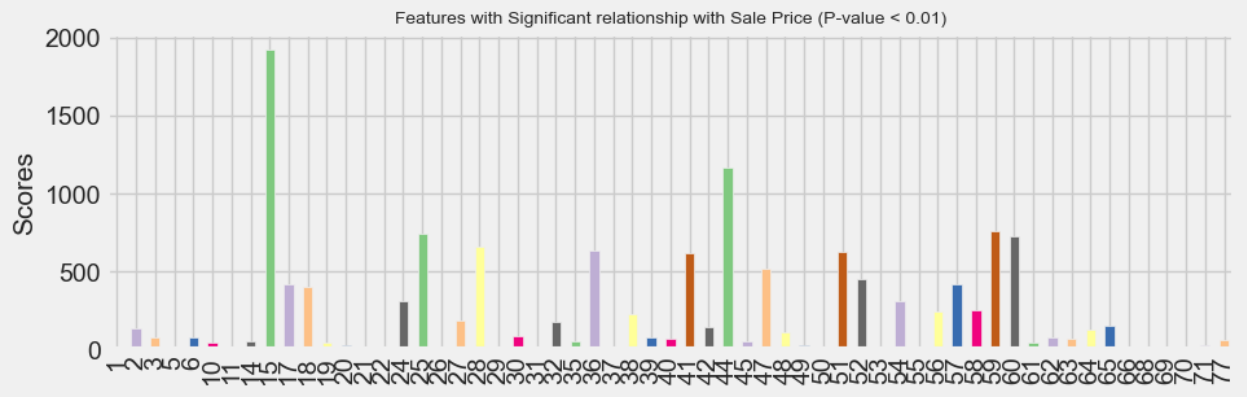
7] Standardization: Standardization is performed to prevent features with wider ranges from dominating the distance metric. But the reason we standardize data is not the same for all machine learning models, and differs from one model to another. The housing price prediction involves variables with different ranges as Encoded categorical features are in range of 1 to 10 or maximum 25 but features involving area in square foot have high ranges from 0 to thousands hence standardization is necessary in this scenario.

StandardScaler: The StandardScaler is preprocessing technique in Sklearn library. Standardize features by removing the mean and scaling to unit variance. The standard score of a data value x is calculated as: z = (x - u) / s, where x is data value of feature, u is mean and s is standard deviation of feature. All features are scale with Standard Scalar.

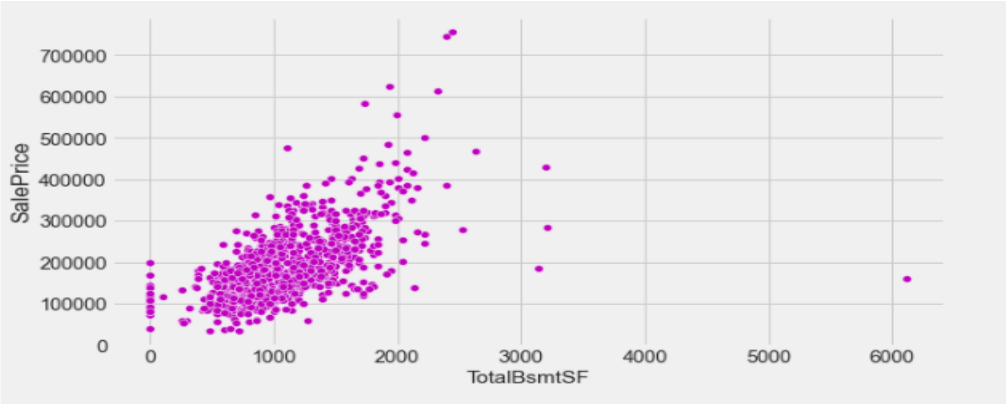
8] PCA: Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components. The importance of each component decreases when going to 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance.

Need of PCA in House price prediction project:
1] The set of feature have very less correlation with response variable Sale Price. It can be observed with SelectKBest feature selection method, there are only 19 features having insignificant relation with Sale Price variable.
2] There is high amount of multicollinearity is observed in feature as 50 variables have more than 10 vif score. Such situation of multicollinearity can't be handled with elimination of correlated features.
3] To handle above scenarios, to reduce dimensionality without losing any variable and information and to Trained model with completely uncorrelated feature there is need of PCA in House prediction project.

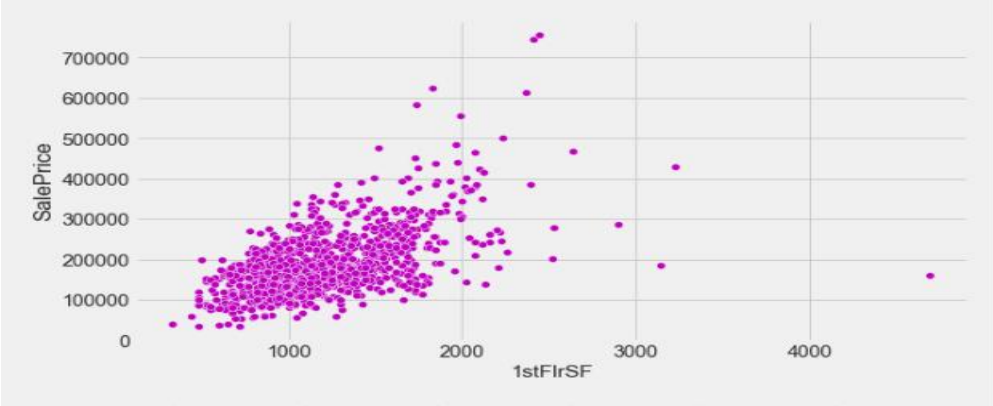




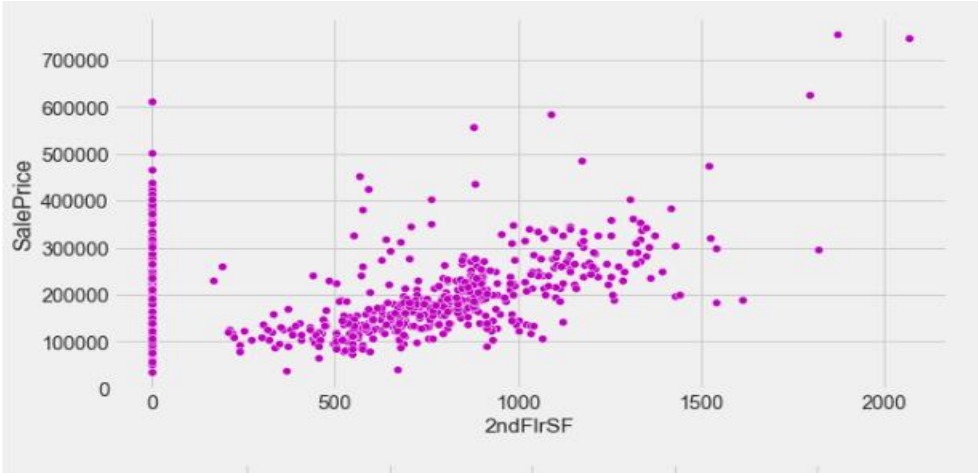
- **Data input-logic-output relationship :**



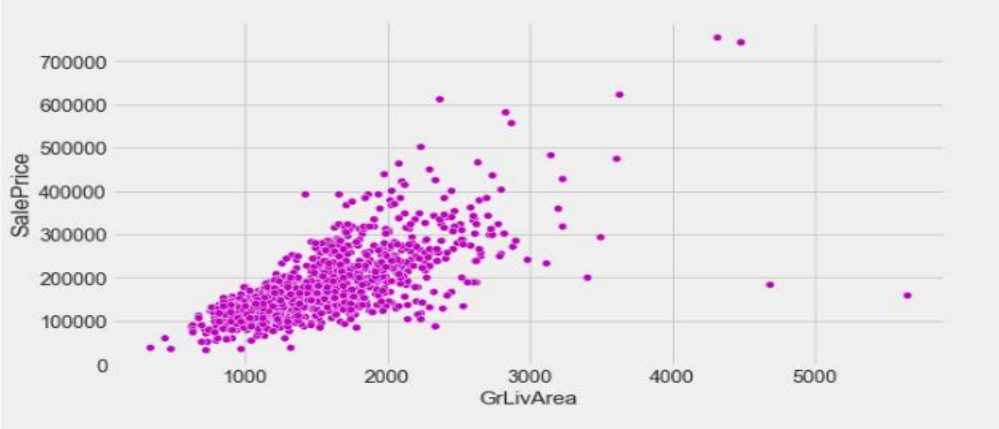
TotalBsmtSF: Total Square feet of basement area are area of basement. The feature is continuous in nature. The increase in area of basement has positive relationship with sale price of house as Area increase the price also increases.



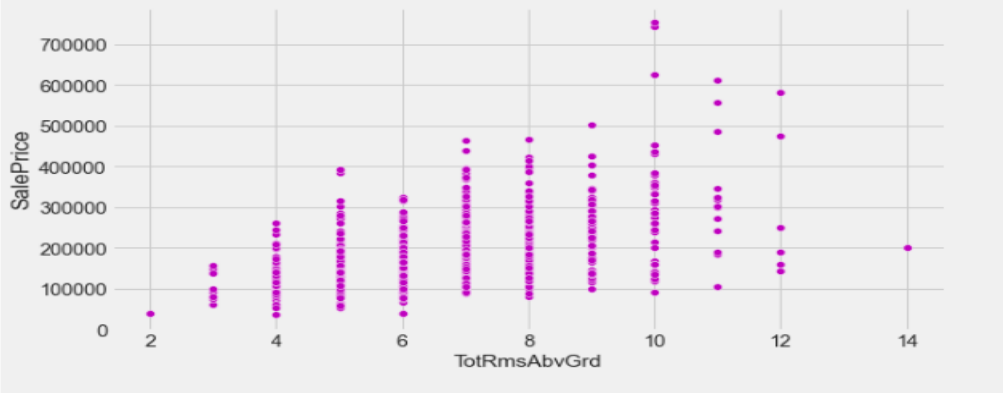
1stFlrSF: The 1stFlrSF is area of first floor in square feet unit. The price of house with large area of first floor area also increases. First floor area is never equal to zero in all house instances of data.



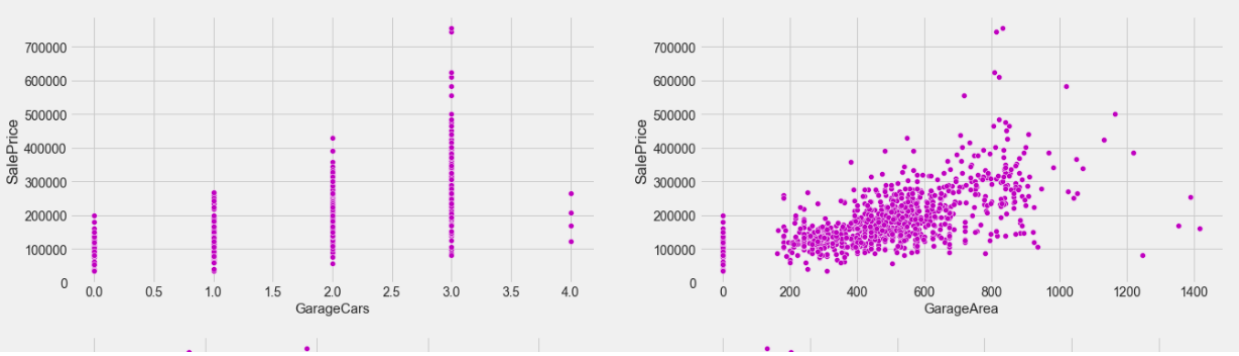
2stFlrSF: The area of second floor in square foot. There are houses without second floor having higher price while there are also houses with second floor having higher prices. Second floor area have relationship with sale price within specific range, there might be other criteria for higher price.



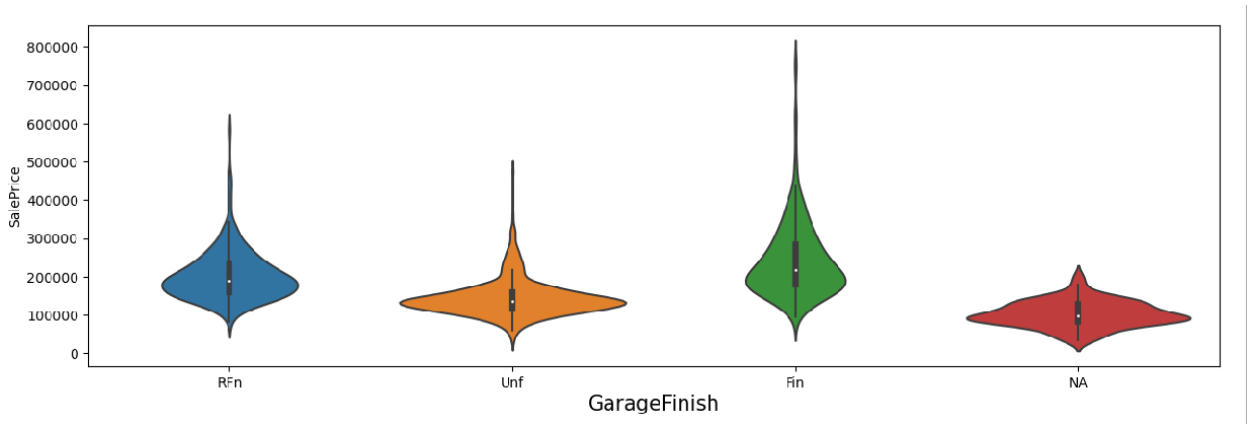
GrLivArea: The feature involves living area in square feet above ground have strong relationship with sale price of house among all variables. As area of living room increase there is price also increases.



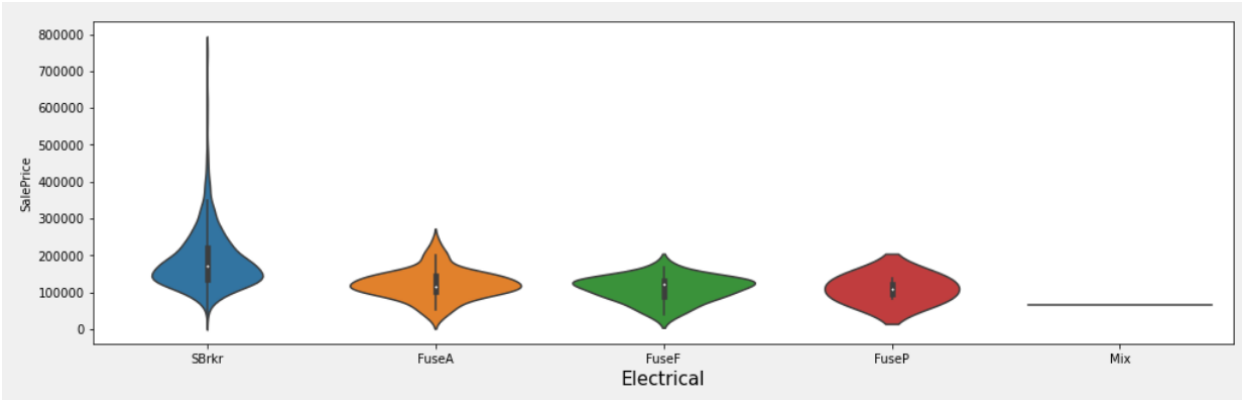
TotRmsAbvGrd: The TotRmsAbvGrd is counts of total rooms above grade which causes effect on price as rooms increases price also increases. The trend can be analyzed with above graph.



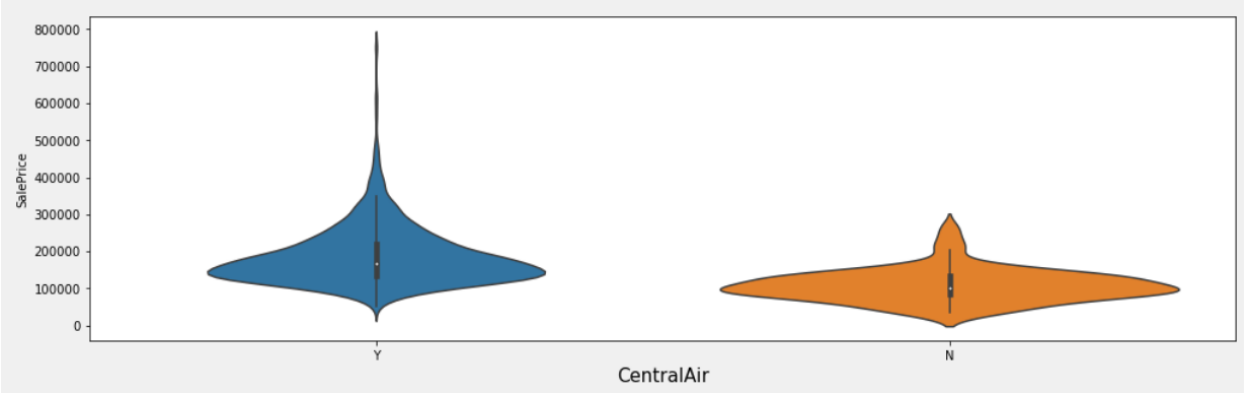
GarageCars and GarageArea: The hoses having garage facilities have high prices as Houses without Garage have low prices can be observed at x axis of right hand side graph. The garage with more numbers car capacity also has high prices. These features have relationship with each other as Number of car capacity increases area also increases.



GarageFinish: The difference between prices of house can be observed with rough finished, finished and unfinished garage types the houses without garage have lower price than houses with garage area.

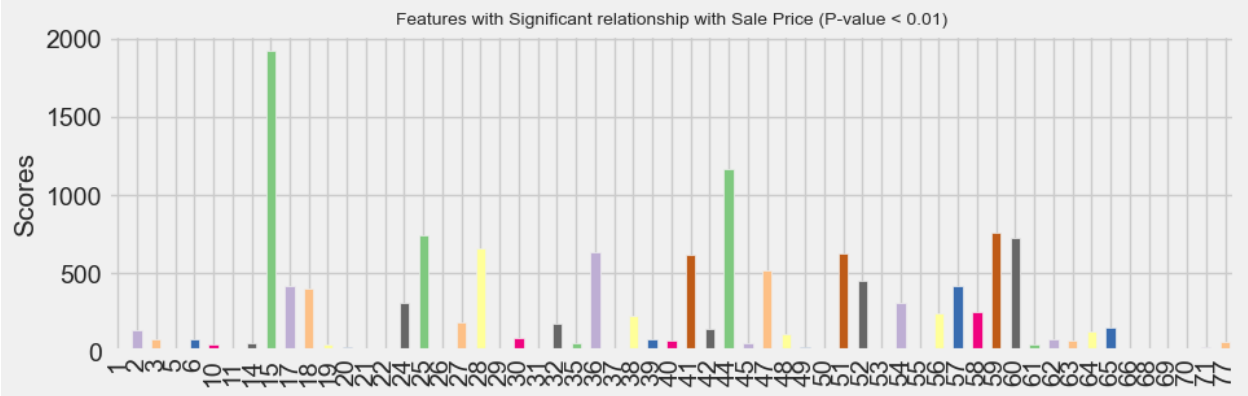


Electrical: The houses with standard circuit breaker and romex system have highest prices. The median also higher than that of houses with other system.



The houses with central air conditioning system have highest price that outliers in prices are houses equipped with central air conditioning system.

The prices in quality based variables are high for excellent quality grade and higher for superior quality which involves OverallQual, ExterQual, BsmtQual, kitchenQual, FireplaceQu, GarageQual, HeatingQc, PoolQC, variables having higher prices for superior quality.



The above features have significant relationship found against 0.01 significance level of null hypothesis. The other remaining features are MSSubClass, Street LandContour, LotConfig, LandSlope, Condition2, BldgType, OverallCond, MasVnrType, BsmtFinType2, BsmtFinSF2, LowQualFinSF, BsmtHalfBath, 3SsnPorch, MiscFeature, MiscVal, MoSold, YrSold, SaleType have not significant relationship with target feature saleprice of house.

- ## Hardware and Software Tools:

Hardware:

1] **Operating System** : Windows 10

2] **Processor** : Intel core i3

3] **RAM** : 8 GB

3] **Hard drive capacity**: 1 TB , SSD 128 MB.

Software:

1] **IDE** : Jupyter Notebook

2] **Library**: Numpy, Pandas, Matplotlib, Seaborn, Scipy, scikit Learn.

Model/s Development and Evaluation

- **Identification of approaches for problem:** The housing price prediction data involves large amount of outliers, upon elimination which causes loss of crucial information, also these outliers are due to natural variation in population Hence it is necessary to train algorithm with outliers. But distance based machine learning algorithms are sensitive to the range and distribution of attribute values. Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models and ultimately poorer results.

  While to trained model with anomalies in data , Tree based algorithms with ensemble techniques will be effective as Tree based algorithms work on basis of True and false for value on either side of split weather it is 10 or 1000 .It doesn't find distance between data values for estimation. The Tree base algorithms are less sensitive towards outliers.

- ## Testing of Identified Approaches:

1] Decision Tree Regressor: Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.
The model is fitted with parameters maximum depth of tree is 18, minimum samples in leaf node are 10 while splitting internal node with more than 15 samples and random state of 17 is used arbitrary to reproduce same results.

http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

2] Gradient Boosting Regressor: The Gradient Boosting Regression is type of algorithms which uses boosting technique where simple models estimation is add to find strong model .These weak model uses gradient descent to minimize the loss and reached the prediction.
The parameters used are learning rate is 0.1; maximum depth of 5, minimum samples in leaf node 2 and minimum samples for splitting internal node is 35 with random state of 10.

http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

3] Random Forest Regressor: A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The model is trained with maximum depth of 15, minimum samples split of 15 and min samples leaf of 10 and random state of 10.

http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

4] Extreme Gradient boosting Regressor: Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm.XGBoost became the go-to method and often the key component in winning solutions for a range of problems in machine learning competitions. The regressor is trained with learning rate is 0.1, max depth of 5, gamma is 0.01 and random state used is 155.

https://xgboost.readthedocs.io/en/stable/python/python_api.html

5] Support Vector Regressor: Support Vector Regression as the name suggests is a regression algorithm that supports both linear and non-linear regressions. This method works on the principle of the Support Vector Machine. SVR the idea is to fit the error inside a certain threshold which means, work of SVR is to approximate the best value within a given margin. The kernel function used for mapping data to higher dimensions. The parameter for training data with SVR are kernel is linear which is used to control error and gamma used to give curvature to decision boundary.

http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

- ## Run and Evaluate selected models:

1] Decision Tree Regressor

```
In [128]: dt1=DecisionTreeRegressor(min_samples_split=15,min_samples_leaf=10,random_state=17,max_depth=18)
          dt1.fit(x_train,y_train)

          Reg_eval(dt1,x_train,y_train,x_test,y_test,train=True)
          Reg_eval(dt1,x_train,y_train,x_test,y_test,train=False)
```

```
********** Training Evaluation ******************

The R squared  for Train data  is 0.862769530401162
------------------------------------------------------------

The RMSE for Train data is 28728.902846085613

********** Testing  Evaluation ******************

The R squared  for Test data is 0.8162210144100108
------------------------------------------------------------

The RMSE for  Test data  is 36380.839759367314
------------------------------------------------------------

The  MSE for Test data  is 1323565501.5967617
------------------------------------------------------------

The MAE for Test data  is 24571.03825158361
------------------------------------------------------------
```

2] GradientBoostingRegressor:

```
In [130]: gbdt1=GradientBoostingRegressor(learning_rate=0.1,min_samples_split=35,max_depth=5,min_samples_leaf=2,random_state=10)
          gbdt1.fit(x_train,y_train)

          Reg_eval(gbdt1,x_train,y_train,x_test,y_test,train=True)
          Reg_eval(gbdt1,x_train,y_train,x_test,y_test,train=False)
```

```
********** Training Evaluation ******************

The R squared  for Train data  is 0.9940228088540418
------------------------------------------------------------

The RMSE for Train data is 5995.73373699189

********** Testing  Evaluation ******************

The R squared  for Test data is 0.8490461830945674
------------------------------------------------------------

The RMSE for  Test data  is 32972.11787677739
------------------------------------------------------------

The  MSE for Test data  is 1087160557.280103
------------------------------------------------------------

The MAE for Test data  is 20486.82869463365
------------------------------------------------------------
```

3] RandomForestRegressor:

In [132]:
```
rf1=RandomForestRegressor(max_depth=15,min_samples_leaf=10, min_samples_split=15,random_state=10)
rf1.fit(x_train,y_train)

Reg_eval(rf1,x_train,y_train,x_test,y_test,train=True)
Reg_eval(rf1,x_train,y_train,x_test,y_test,train=False)
```

```
********** Training Evaluation ******************

The R squared  for Train data  is 0.87529848715006
------------------------------------------------------------

The RMSE for Train data is 27386.06452273302

********** Testing  Evaluation ******************

The R squared  for Test data is 0.8351447292800585
------------------------------------------------------------

The RMSE for  Test data  is 34456.90082261861
------------------------------------------------------------

The  MSE for Test data  is 1187278014.2997754
------------------------------------------------------------

The MAE for Test data  is 21673.972800829953
```

4] XgboostRegressor:

In [134]:
```
xgr1=XGBRegressor(booster='gbtree',learning_rate=0.1,random_state=155,max_depth=5,gamma=2)

xgr1.fit(x_train,y_train)

Reg_eval(xgr1,x_train,y_train,x_test,y_test,train=True)
Reg_eval(xgr1,x_train,y_train,x_test,y_test,train=False)
```

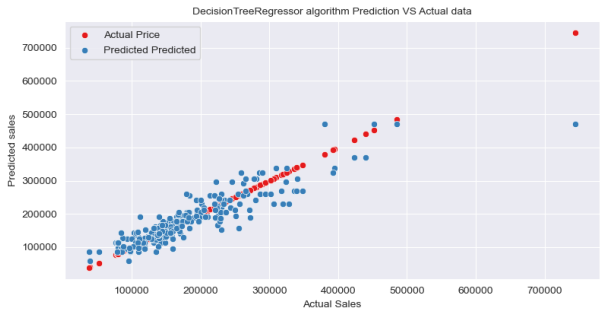
```
********** Training Evaluation ******************

The R squared  for Train data  is 0.9960440005662545
------------------------------------------------------------

The RMSE for Train data is 4877.776266922135

********** Testing  Evaluation ******************

The R squared  for Test data is 0.8764190956006461
------------------------------------------------------------

The RMSE for  Test data  is 29833.243922809008
------------------------------------------------------------

The  MSE for Test data  is 890022442.9578205
------------------------------------------------------------

The MAE for Test data  is 19039.90616653312
```

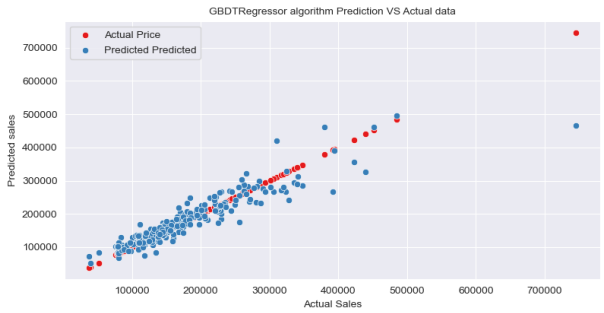
5] SVR:

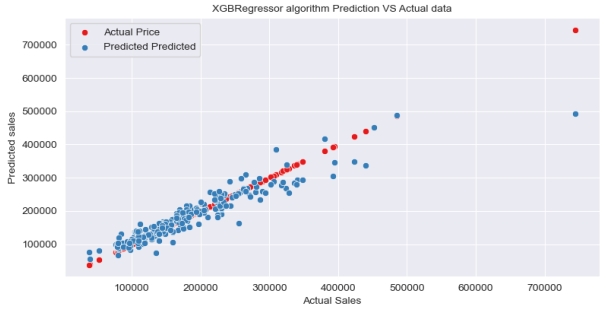
In [136]:
```
svr1=SVR(gamma=0.001,C=10000,kernel='linear')
svr1.fit(x_train,y_train)

Reg_eval(svr1,x_train,y_train,x_test,y_test,train=True)
Reg_eval(svr1,x_train,y_train,x_test,y_test,train=False)
```

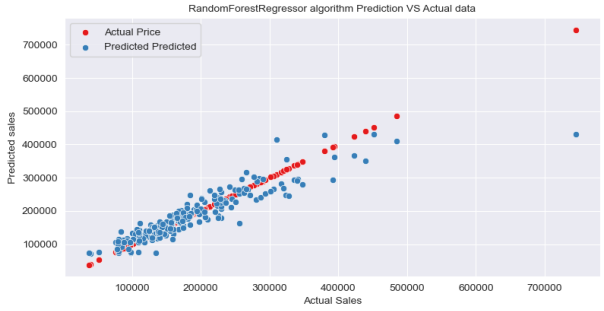
```
********** Training Evaluation ******************

The R squared  for Train data  is 0.8122720411915263
------------------------------------------------------------

The RMSE for Train data is 33601.467490269446

********** Testing  Evaluation ******************

The R squared  for Test data is 0.8044023250336482
------------------------------------------------------------

The RMSE for  Test data  is 37532.42612717981
------------------------------------------------------------

The  MSE for Test data  is 1408683010.9922097
------------------------------------------------------------

The MAE for Test data  is 21288.68133298116
------------------------------------------------------------
```

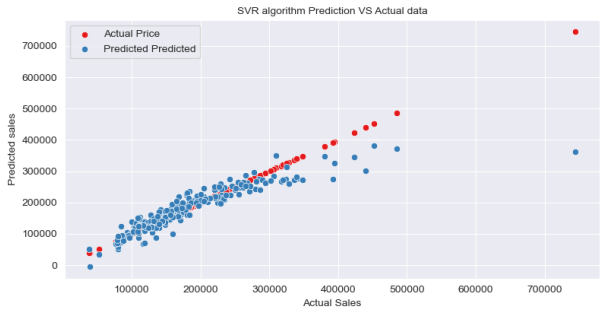
- **The performance of algorithms has evaluated with plotting predicted values against Actual values of Target variables . This can be observed in below graph:**







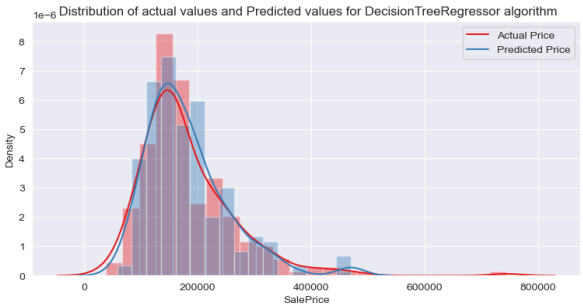


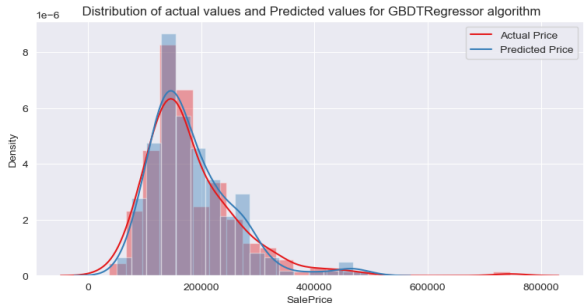


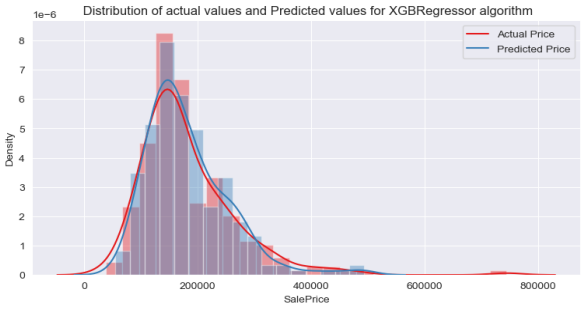
Observations:

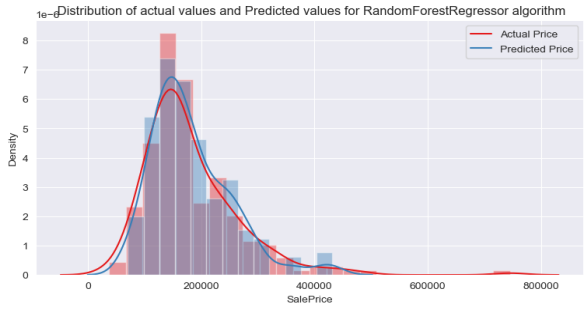
1] The scatter diagram of Predicted prices with actual prices of Test data for all algorithms.
2] The red points on each subplot represent the trend of house prices when prediction and actual prices will be same.
3] While Blue points represents prediction trend Vs. Actual price.
2] XGBoost algorithm has out-performed among all algorithms trained on Housing Price data. As values are observed very close to red graph.
5] The SVR prediction prices are widely spread around red graph which shows low prediction power.

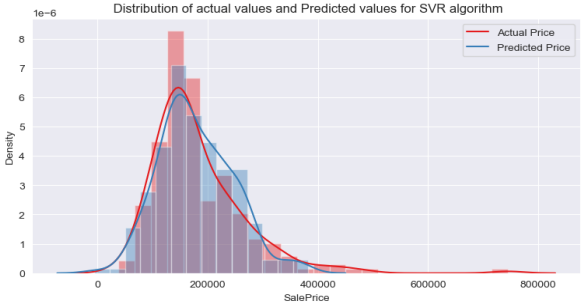
- **Distribution of Actual and Predicted Prices :**
The distribution gives performance of algorithms for different range of predicted prices as compared to actual prices.











Observations:

1] The above five subplots represents distribution for each algorithm respectively.

2] The red distribution is of actual prices in test data, while Blue distribution is for predicted prices with algorithms.

3] All algorithms predicted similar prices for start to middle prices of house. While these models have low predicted power for extreme prices.

4] Again XGboost has out-performed among all algorithms as predicted values distribution is very close to Actual Prices.

- **All Evaluation metrics for each algorithms are collected in dataframe which is as follows:**

```
In [147]: EvalDF
```

Out[147]:

	Regressor	Train R2 square	Test R2 square	RMSE	Normalized RMSE	Validation score	Analysing Overfitting model
0	DecisionTreeRegressor	0.862770	0.816221	36380.839759	0.050522	0.714135	0.102086
1	GBDTRegressor	0.994023	0.849046	32972.117877	0.045788	0.783333	0.065713
2	XGBRegressor	0.996044	0.876419	29833.243923	0.041429	0.832787	0.043632
3	RandomForestRegressor	0.875298	0.835145	34456.900823	0.047850	0.798977	0.036168
4	SVR	0.812272	0.804402	37532.426127	0.052121	0.787041	0.017362

Observations:

1] From model training, evaluation with different metrics and visualization it can be interpreted that models have good performance with higher r2 score in in sample and out sample evaluation.

2] There is cross validation performed on all algorithms to validate model against over fitting, it can be concluded from results that there is no ovefitting observed in any algorithms as error in score are in acceptable range.

3] From all above evaluation with RMSE and R2 score, xgr1 which is XGBoostRegressor out-performed with 0.8764 R2 score and Normalized RMSE of 0.0415. Hence final model is XGBoostRegressor.

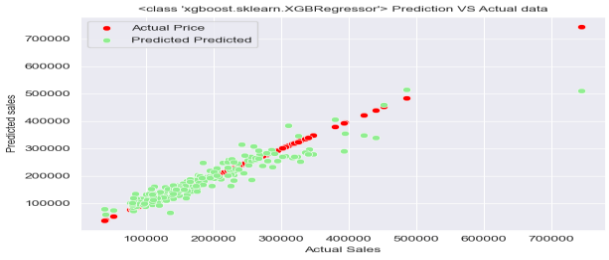
- **Hyper parameter tuning with XGR :**

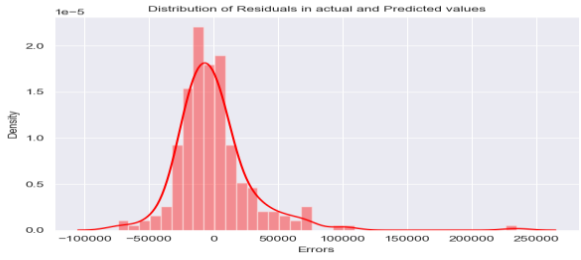
```
In [150]: from sklearn.model_selection  import GridSearchCV

          param2={'gamma':[0.01,0.02,0.05,0.1],
                          'learning_rate':[0.095,0.098,0.1,0.15,0.18],
                            'max_depth' :[3,4,5,6,7,8]}


          grid2=GridSearchCV(XGBRegressor(),param_grid=param2,cv=7)
          grid2.fit(x_train,y_train)
          print(grid2.best_params_)
```

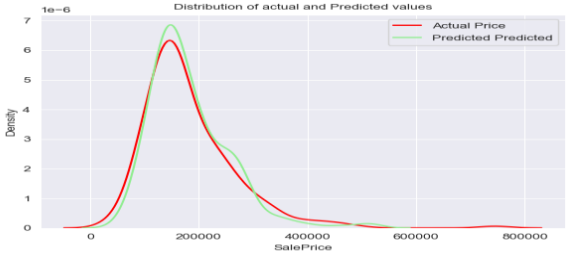
```
{'gamma': 0.01, 'learning_rate': 0.1, 'max_depth': 3}
```

```
In [151]: xgr2=grid2.best_estimator_
          xgr2.fit(x_train,y_train)

          Reg_eval(xgr2,x_train,y_train,x_test,y_test,train=True)
          Reg_eval(xgr2,x_train,y_train,x_test,y_test,train=False)
```

```
********** Training Evaluation ******************

The R squared  for Train data  is 0.9708382538548875
------------------------------------------------------------

The RMSE for Train data is 13243.431466351034

********** Testing  Evaluation ******************

The R squared  for Test data is 0.8729397421051751
------------------------------------------------------------

The RMSE for  Test data  is 30250.29823424302
------------------------------------------------------------

The  MSE for Test data  is 915080543.2606462
------------------------------------------------------------

The MAE for Test data  is 19908.948934962606
------------------------------------------------------------
```







Observation:

The score is not possible to increase with hyper parameter tuning, The xgr1 is Extreme Gradient Boosting algorithm with max_depth booster='gbtree',learning_rate=0.1,random_state=155,gamma=2 is final model.

Key Metrics for success in solving problem under

1] RMSE: In machine learning, it is extremely helpful to have a single number to judge model's performance, whether it is during training, cross validation. Root mean square error is one of the most widely used measurements for this. It is proper scoring rule that is intuitive to understand and compatible with some of the most common statistical assumptions. Root mean square error or root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions. It shows how far predictions fall from measured true values using Euclidean distance.

In house price prediction, it is very useful metrics as the lower the difference between predicted and actual prices will help in firm to make decision in selling and purchasing house. It is absolute measure of fit having same unit as response variable. The RMSE obtained with XGBoost Regressor is 29834. The RMSE is useful for evaluating model with same response variables while to interpret easily it can normalize with Maximum and Minimum values of response variable to compare with other model evaluation.

2] R squared: The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. It is pronounced as R squared and is also known as the coefficient of determination. It works by measuring the amount of variance in the predictions explained by the dataset. If the value of the R2 is 1, it means that the model is perfect and if its value is 0, it means that the model will perform badly on an unseen dataset.

Observation:

1] The models are analyzed against both metrics R squared and RMSE for selecting best model which will have higher R2 and Lower RMSE. The both conditions of selection of model are satisfied with XGR as The Highest R2 of 0.8764 and Lowest RMSE value of 29834 and Normalized RMSE of 0.0415.

2] The features MSSubClass, Street, LandContour, LotConfig, LandSlope, Condition2, BldgType, OverallCond, MasVnrType, BsmtFinType2, BsmtFinSF2, LowQualFinSF, BsmtHalfBath, 3SsnPorch, MiscFeature, MiscVal, MoSold, YrSold, SaleType have not significant relationship with target feature saleprice of house. The purchase can be made with considering except these features.The other feature shows significant relationship with 1 % of risk.

Conclusions

1] In this project, The Housing price prediction involves high amount of outliers or extreme values. The Training of model with outliers with higher accuracy is challenging task. During distribution analysis and skewness analysis it's observed that continuous indicators have non normal distribution. So Training model with linear method in machine learning will exhibit less interpretability and less predictive power. This condition is solved with employing tree base algorithms which are robust to extreme point and also with use transformation technique like yeo-johnsosn.

2] Upon correlation analysis, Multicollinearity and feature selection method, it can be concluded that predictors exhibiting effective association with response also exhibits the multicollinearity with other indicators in data. This scenario makes confusion in real world process. If more than two indicators exhibit association among them, Makes less interpretability to human being. This Condition has been solving with dimensionality reduction technique like PCA.

3] Highly correlated are mentioned in project reports. The impacts of quality defining indicators are more on sale prices. The superior the quality of materials and facilities will make prices superior which can be observed also in real world.

Learning outcomes

1] The data has very less number of instances 1168 rows are very. The missing records in such amount of data are very crucial to handle. It is not recommended in data science to delete any data without knowing its effect on other indicators.

2] The mostly missing records have exhibited relation with other indicators such data can make biased results upon imputation with wrong entity and deletion can make loss of information. The detection and imputation of missing records can't imagine without power tools like pandas seaborn and matplotlib.

3] Also the treatment of extreme values without elimination is highly challenging task which can be done with some lines of code with transformation.

4] While training of various algorithms, it is challenging work to find parameters in ensemble algorithm which exhibit generalized performance without over fitting. It is challenging task in machine learning to enhance score and performance of model without over fitting in model.

5] The hyper parameter tuning of algorithm is challenging work due to system limitation upon feeding large lists of hyper parameters will become Time consuming process**.**

Limitations of Project:

1] The model is not able to predict extreme prices which is due to lack of instances for extreme values. Upon addition of more data instances the prediction will be more accurate with less deviation.

2] The system hardware limitations have made restriction to obtained more accuracy with hyper parameter tuning with GridSearchCV which power full technique in Scikit learn library.

Upon addition of more data and higher configuration of system, it is definitely possible to enhance score and performance of Machine learning model.