HOUSING: PRICE PREDICTION

Submitted by:

ArunPrasad.k

Table of content

***Abstract*—House prices increase every year . so there is a need for a system to predict house prices in the future. House price prediction can help the developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house. There are three factors that influence the price of a house which include physical conditions concept and location. This research aims to predict house prices based on A US based housing company named Surprise Housing which has decided to enter the Australian market. The company uses data analytic to purchase houses at a price below their actual values and flip them at a higher price.**

In the present paper we discuss about the prediction of future housing prices that is generated by machine learning algorithm. For the selection of prediction methods we compare and explore various prediction methods. We utilize Ada Boost regression as our model because of its adaptable and probabilistic methodology on model selection. Our result exhibit that our approach of the issue need to be successful and has the ability to process predictions that would be comparative with other house cost prediction models.

***General Terms*: Data Analytics Exploratory Data Analytics Machine Learning Model Evaluation Data Science.**

***Index Terms*: Machine learning algorithm House prediction; regression analysis;**

1. INTRODUCTION

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

The company's objective is to look at prospective properties to buy houses to enter the market. We 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?

2.Analytical Problem Framing

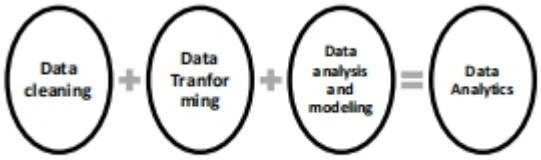


Fig 1: Data Analytics

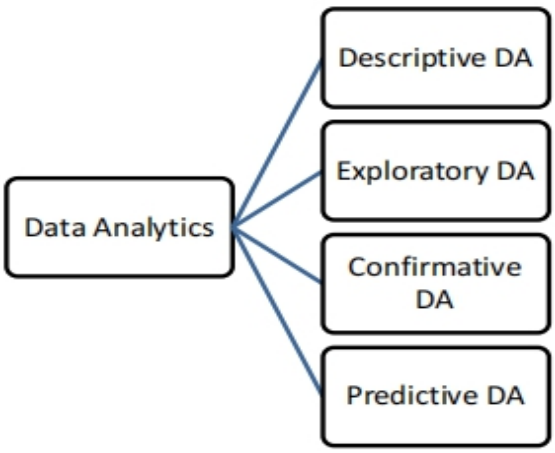


Fig 2: Categories of Data Analytics

3.PROCESS FLOW

There is a step by step approach to choose a particular model for the current problem. We need to decide whether a particular machine learning model is suitable for our problem or not. Here we can see process flow being followed.

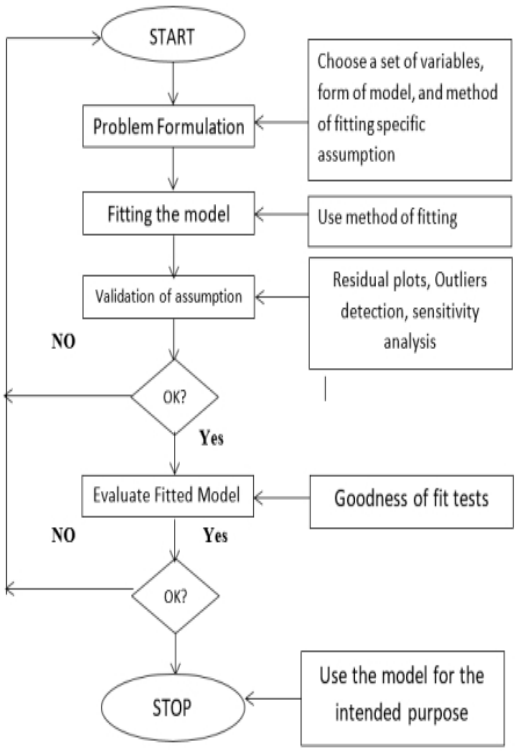


Fig 3: Process of fitting a Machine Learning Model

4.RELATED WORK

4.1 House Price Affecting Factors *:*

There are several factors that affect house prices. Here we divide these factors into three main groups they are

- ✓ physical condition
- ✓ Sale condition
- ✓ Location

Physical conditions are properties possessed by a house that can be observed by human senses including the size of the house the number of bedrooms the availability of kitchen and garage the availability of the garden the area of land and buildings and the age of the house .

Sale Conditions are arrangements for a sale that are stated by the person or company selling the goods or properties which buyer must agree to.

for example, when payment must be made, how goods will be delivered, etc

Location is an important factor in shaping the price of a house. This is because the location determines the prevailing land price . In addition the location also determines the ease of access to public facilities such as schools campus hospitals and health centers as well as family recreation facilities such as malls culinary tours or even offer a beautiful . In general the factors affecting the house prices will be presented below.

MSSubClass	Foundation
MSZoning	BsmtQual
LotFrontage	BsmtCond
LotArea	BsmtExposure
Street	BsmtFinType1
Alley	BsmtFinSF1
LotShape	BsmtFinType2
LandContour	BsmtFinSF2
Utilities	BsmtUnfSF
LotConfig	TotalBsmtSF
LandSlope	Heating
Neighborhood	HeatingQC
Condition1	CentralAir
Condition2	Electrical
BldgType	1stFlrSF
HouseStyle	2ndFlrSF
OverallQual	LowQualFinSF
OverallCond	GrLivArea
YearBuilt	BsmtFullBath
YearRemodAdd	BsmtHalfBath
RoofStyle	FullBath

RoofMatl	HalfBath
Exterior1st	BedroomAbvGr
Exterior2nd	KitchenAbvGr
MasVnrType	KitchenQual
MasVnrArea	TotRmsAbvGrd
ExterQual	Functional
ExterCond	Fireplaces
FireplaceQu	OpenPorchSF
GarageType	EnclosedPorch
GarageYrBlt	3SsnPorch
GarageFinish	ScreenPorch
GarageCars	PoolArea
GarageArea	PoolQC
GarageQual	Fence
GarageCond	MiscFeature
PavedDrive	MiscVal
WoodDeckSF	MoSold
SaleCondition	YrSold
SalePrice.	SaleType

5.DESIGN APPROACH

5.1. Linear regression:

Simple linear regression statistical method allows us to summarize and study the relationship between two continuous quantitative variables.

- ✓ One variable, denoted x, is regarded as the predictor, explanatory, or independent variable.
- ✓ The other variable, denoted y, is regarded as the response, outcome, or dependent variable

5.2. Lasso Regression:

Least Absolute Shrinkage and Selection Operator (Lasso) is an L1 norm regularised regression technique that was formulated by Robert Tibshirani in 1996 [6]. Lasso is a powerful technique that performs regularisation and feature selection. Lasso introduces a bias term, but instead of squaring the slope like Ridge regression, the absolute value of the slope is added as a penalty term. Lasso is defined as:

$$\mathrm{L} = \mathrm{Min}(\text{sum of squared residuals} + \alpha * [\text{slope}])$$

Where $\mathrm{Min}(\text{sum of squared residuals})$ is the Least Squared Error, and $+\alpha * [\text{slope}])$ is the penalty term. However, alpha α is the tuning parameter which controls the strength of the penalty term. In other words, the tuning parameter is the value of shrinkage.

5.3. Ridge Regression:

The Ridge Regression is an L2 norm regularised regression technique that was introduced by Hoerl in 1962. It is an estimation procedure to manage col linearity without removing variables from the regression model. In multiple linear regression, the multicollinearity is a common problem that leads least square estimation to be unbiased, and its variances are far from the correct value. Therefore, by adding a degree of bias to the regression model, Ridge Regression reduces the standard errors, and it shrinks the least square coefficients towards the origin of the parameter space .

Ridge formula is

$$\mathrm{L} = \mathrm{Min}(\text{sum of squared residuals} + \alpha * [\mathrm{Slope}]^2$$

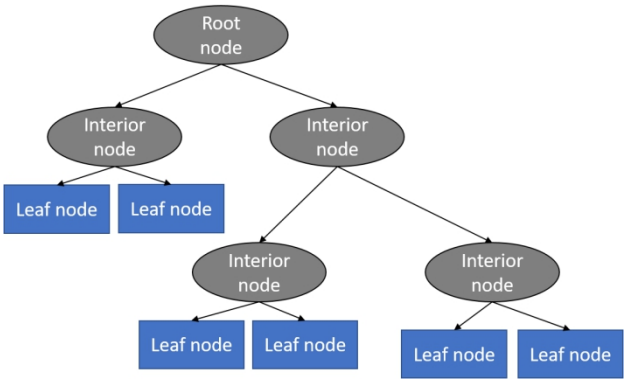
Where $\mathrm{Min}(\text{sum of squared residuals})$ is the Least Squared Error, and $[\mathrm{Slope}]^2$ is the penalty term that Ridge adds to the Least Squared Error.

5.4. Random Forest Regression:

A Random Forest is an ensemble technique qualified for performing classification and regression tasks with the help of multiple decision trees and a method called Bootstrap Aggregation known as Bagging Decision Trees are used in classification and regression tasks, where the model (tree) is

formed of nodes and branches. The tree starts with a root node, while the internal nodes correspond to an input attribute. The nodes that do not have children are called leaves, where each leaf performs the prediction of the output variable.

.



A Decision Tree can be defined as

$$\varphi = x \rightarrow \gamma$$

5.5. Gradient Boosting algorithm:.

Gradient boosting is a machine Taking in strategy to relapse Also arrangement problems, that produces a prediction model in the structure of an group from claiming powerless prediction models. The exactness of a predictive model might be helped to two ways:. Possibly by grasping characteristic building alternately. Toward applying boosting calculations straight far. There are a significant number boosting calculations in.

- ✓ Gradient Boosting
- ✓ XGBoost
- ✓ AdaBoost
- ✓ Gentle Boost etc.

Each boosting algorithm need its own underlying math. Also, a slight variety may be watched same time applying them. Boosting calculation will be a standout among those The greater part capable Taking in thoughts acquainted in the final one twenty A long time. It might have been intended to order problems, yet all the it can be developed should relapse too. The inspiration to gradient boosting might have been An technique. That combines those outputs about large portions "weak" classifiers to process An capable "committee. " a powerless classifier will be person whose slip rate is main superior to irregular guessing.

6.IMPLEMENTATION

6.1. Importing Dataset:

Reading the data to plot the graphs:

```
# Importing the training data for analysis
df_train = pd.read_csv('housing train.csv')
df_train.head(5)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	Mo
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	

6.2 Cleaning Dataset:

Before applying any type of data analytics on the data set, the data should be first cleaned.

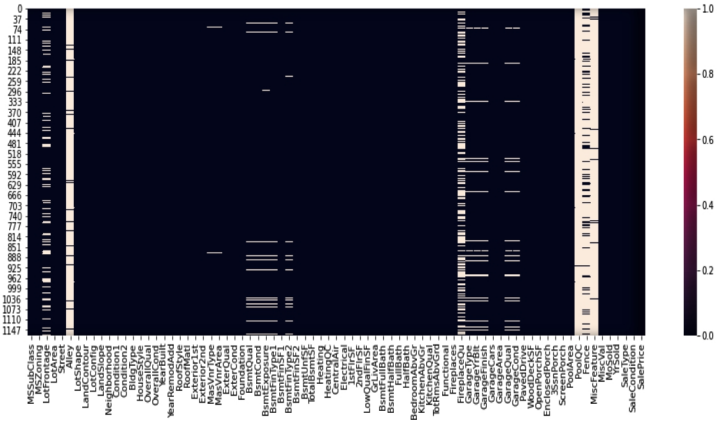


Figure 4: Heatmap on missing values

We find many attributes with missing values.The following is the list of Columns with missing values.

i. LotFrontage
ii. Alley
iii. MasVnrType
iv. MasVnrArea
v. BsmtQual
vi. BsmtCond
vii. BsmtExposure
viii. BsmtFinType1
ix. BsmtFinType2
x. FireplaceQu
xi. GarageType
xii. GarageYrBlt
xiii. GarageFinish
xiv. GarageCond
xv. PoolQC
xvi. Fence
xvii. MiscFeature

It is very important to not loss too much data.But still filling the NaN values with mean or median or mode may result in high bias if we train ML algorithms especially when the column has high percentage of missing values.Using mean value for replacing missing values may not create a great model and hence gets ruled out.As a rule of thumb, when the data goes missing on 40 percent of the variable and above, dropping the variable should be considered.so it is better to drop columns having high percentage of NaN values and fill the remaining missing columns with mean or mode values.

```
#Dropping columns with more than 40% missing values
df_train=df_train.drop("Alley",axis=1)          #93.77% NaN
df_train=df_train.drop("FireplaceQu",axis=1)    #47.26% NaN
df_train=df_train.drop("PoolQC",axis=1)         #99.52% NaN
df_train=df_train.drop("Fence",axis=1)          #80.75% NaN
df_train=df_train.drop("MiscFeature",axis=1)    #80.75% NaN
df_train=df_train.drop("LotFrontage",axis=1)

```

```
# Dropping missing values from other columns
df_train.dropna(axis="rows",inplace=True)
```

Rechecking for the Missing value.Here we find there are no missing values in the data set which has been handled in this case.

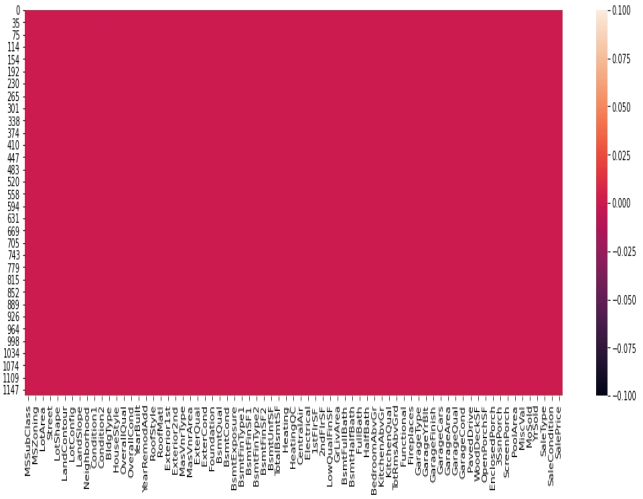


Figure 5: Heatmap on missing values

7. EXPLORATORY DATA ANALYSIS

We are going to perform exploratory data analysis for our problem in the first stage. In exploratory data analysis data set is explored to figure out the features which would influence the Houses sale price. The data is deeply analyzed by finding a relationship between each attribute and our target label.

7.1 Uni-variate Analysis

i) Target variable "saleprice"

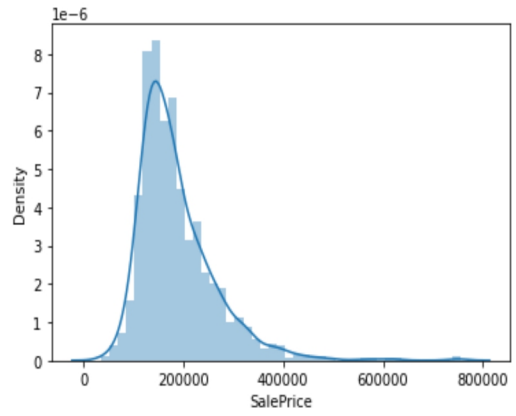
First we will see on the Basic Descriptions on the target variable "sale price".

```
#description about column SalesPrice
df_train["SalePrice"].describe()
```

```
count      1071.000000
mean     187212.879552
std       78367.298698
min       35311.000000
25%      135000.000000
50%      169500.000000
75%      222000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

```
#Let's visualize the distribution of sale price
sns.distplot(df_train['SalePrice'])
```

```
<AxesSubplot:xlabel='SalePrice', ylabel='Density'>
```



Here we observe that there are No missing values. Here we find that the mean is higher than median in most of the columns.If the mean is higher than the median, the distribution is positively skewed.The maximum and the 75% have huge range of difference.we observe that the difference is abnormal

we infer that we may have very less outliers or May be not. From displacement Graph we infer that the sale price of most of the houses were around **200000.**The displacement graph shows the target variable is skewed.

ii) Categorical Features

As we have many columns ,so let's make groups . There are several factors that affect house prices. Here we will divide these factors into three main groups there are

- Location
- physical condition
- Sales Type and condition

Location:

Location is an important factor in shaping the price of a house. This is because the location determines the prevailing land price .

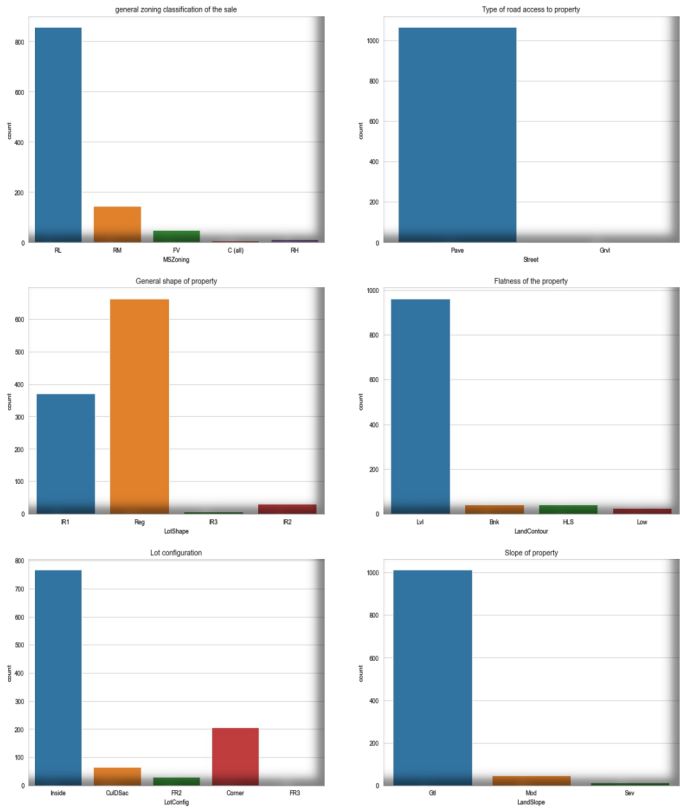


Figure 7 : Count plot on columns related to Location

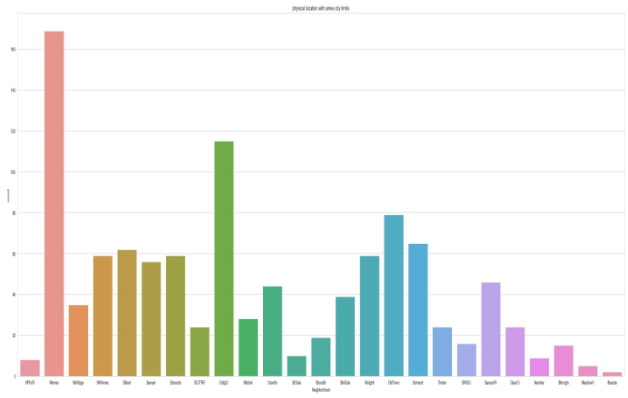


Figure 8 :Count plot on column "Neighbourhood"

Plot Insight:

i. More number of houses were found with low residential density

ii. There were around 1068 houses with paved road access whereas only 3 houses with Gravel road access

iii. Large number of houses where seen to have regular shape and only very few with irregular shape

iv. Regarding flatness of the property we notice houses around 950 in numbers were near flat/level followed by houses with quick and significant rise from street grade to building.

v. Houses with Inside lot configuration where found high in numbers

vi. Mostly houses where found in Gentle slopes,followed by Moderate and severe slope

vii. Considering physical location with ames city limits we see more houses were found in north ames[NAmes]

viii. Least number of houses found in the location of Bluestem of ames city.

Physical Condition:

Physical conditions are properties possessed by a house that can be observed by human senses.

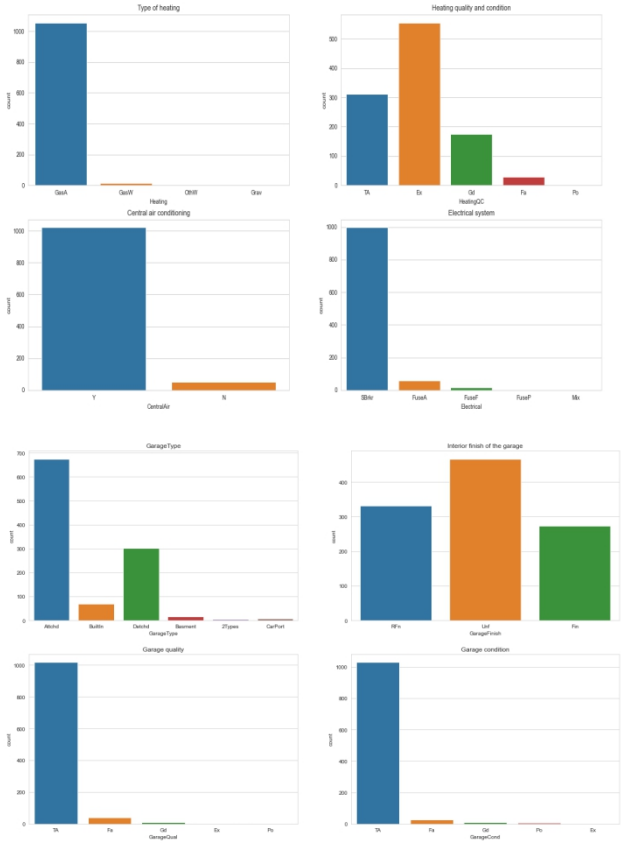


Figure 9 :Count plot on column related to Physical condition of the property

Plot Insight:

i. Gas forced warm air furnace heating sysytem is found in most of the houses

ii. There were more number of houses with excellent Heating quality and condition

iii. Almost all houses is fitted with Central air conditioning system

iv. Around 1000 houses is fitted with Standard Circuit Breakers & Romex as the electrical system

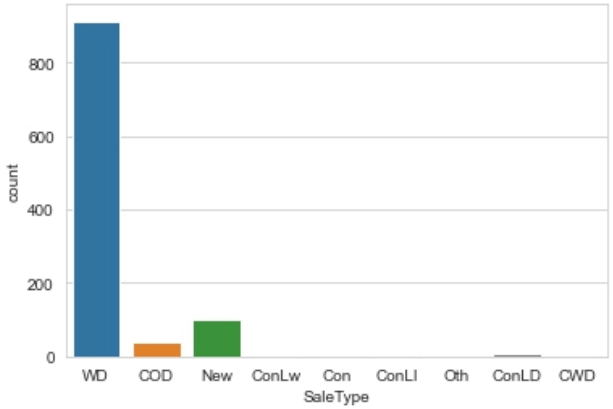
v. We can see most of the houses were attached with garage

vi. Regarding the interior finish of the garage, we can see only 280 garages were in finished condition. Around 480 garages were still unfinished,followed by roughly finished garages(around 320 in numbers).

vii. Many garages were seen with average/typical garage quality

viii. similarly we can see the condition of most of the garages were in average/typical condition .

Sales Type and condition:



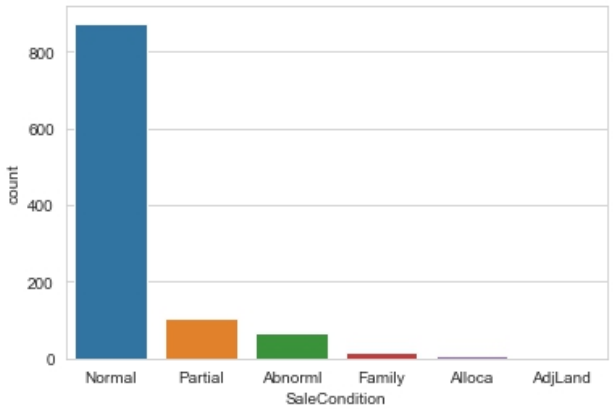


Figure 10: Sales Type and condition

i. Warranty deed acts as a guarantee to the buyer where the seller has all the right to sell the property, and also the property is free from debts and other liabilities,despite the warranty deed. The graph shows most of the properties where Warranty Deed in **Conventional way**

ii. Regarding sales condition,Normal condition where high when considering others

iii) Numerical feature:

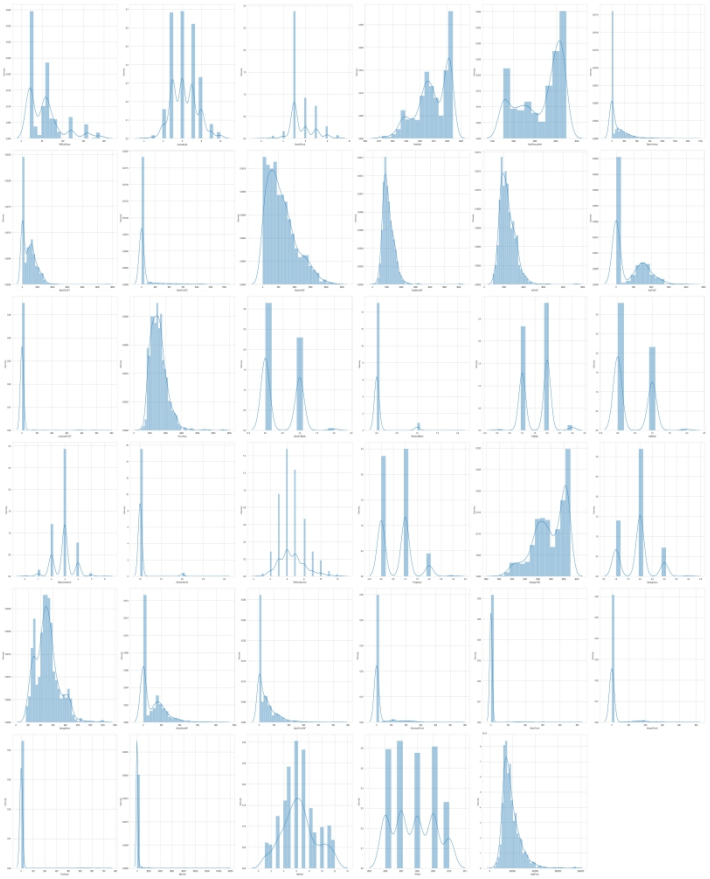


Figure 11 : Displacement Graph

From the displacement graph we infer that all the columns shows Right Skewed Distribution. Hence all the attributes must have Positive skewness.

7.2 Bi-variate Analysis:

Bi-variate analysis is one of the simplest forms of quantitative (statistical) **analysis**. It involves the **analysis** of two variables (often denoted as X, Y), for the purpose of determining the empirical relationship between them. **Bi-variate analysis** can be helpful in testing simple hypotheses of association.

Here we will be seeing the relationship between each attribute and our target column “SalePrice”.

Figure 12: Scatterplot b/w Target varaible and other variables

Plot Insight:

i. From the scatter plot we understand that , with increase in each square feet the sale price of the property increases.

8. Categorical encoding using Label-Encoder

Label Encoding refers to converting the **labels** into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those **labels** must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

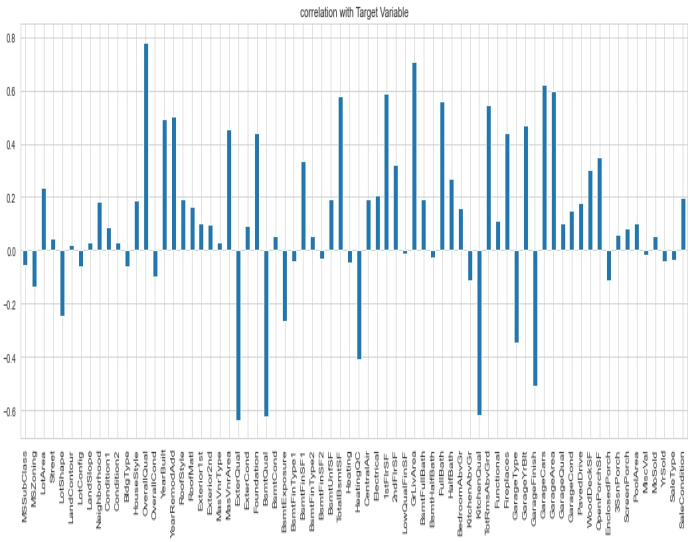
```
#From the given data-set we can infer that all the data types are categorical
#we convert them to integer type by using the label encoder method
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for columns in df.columns:
    df[columns]=le.fit_transform(df[columns])
```

Once the Encoding is done we proceed with Feature Engineering.

9. Feature Engineering

Feature engineering is the most important part of data analytic process. It deals with, selecting the features that are used in training and making predictions. In feature engineering the domain knowledge is used to find features in the dataset which are helpful in building machine learning model. It helps in understanding the dataset in terms of modeling. A bad feature selection may lead to less accurate or poor predictive model. The accuracy and the predictive power depend on the choice of correct features. It filters out all the unused or redundant features.

9.1. Correlation



Here “SalePrice” column is chosen as response column. These features are selected because their values have an impact on the prediction of House Price. These features will be the value of “y” in the bar-plots as shown

above. If wrong features where selected then even the good algorithm may produce the bad predictions. Therefore, feature engineering acts like a backbone in building an accurate predictive model.

9.2. Looking for Outliers

The difference between a good and an average machine learning model is often its ability to clean data. One of the biggest challenges in data cleaning is the identification and treatment of outliers. In simple terms, outliers are observations that are significantly different from other data points. Even the best machine learning algorithms will underperform if outliers are not cleaned from the data because outliers can adversely affect the training process of a machine learning algorithm, resulting in a loss of accuracy.

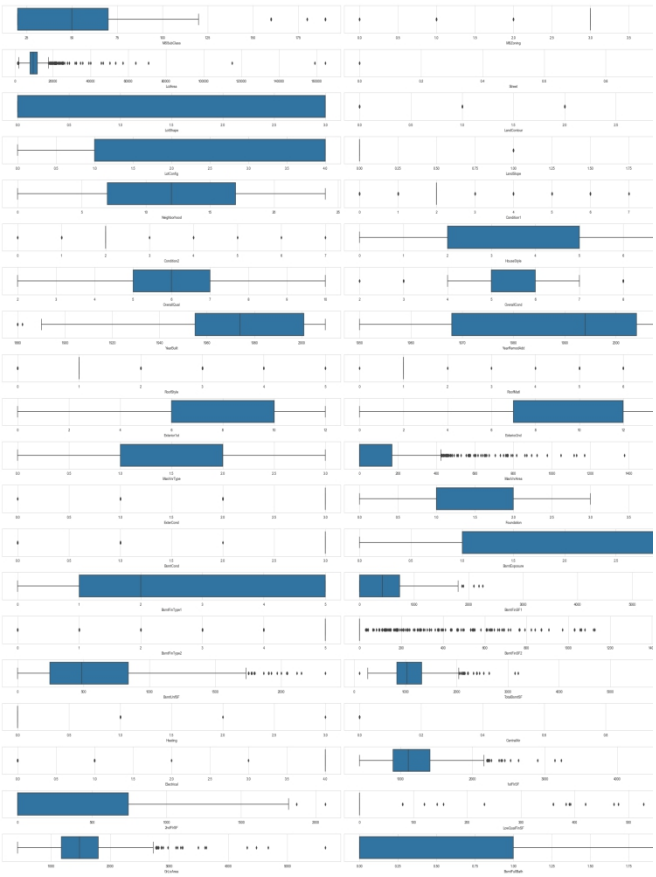


Figure Outliers using BoxPlot

9.3.Treating Outliers:Z score Technique

In this procedure we calculate the z-score for each observation (fix this). Any z-score greater than 3 or less than -3 is considered to be an outlier. This rule of thumb is based on the empirical rule. From this rule we see that almost all of the data (99.7%) should be within three standard deviations from the mean. By calculating the z-score we are standardizing the observation, meaning the standard deviation is now 1. Thus from the empirical rule we expect 99.7% of the z-scores to be within -3 and 3.

```
#Removing Outliers
#Z-score Techinique
from scipy.stats import zscore
z=np.abs(zscore(df_train))
z
```

```
threshold=3
print(np.where(z>3))
df_new=df_train[(z<3).all(axis=1)]
```

Percentage Loss

loss_of_data=(1071-500)/1071*100

loss_of_data=53.314

when we try to remove outliers it will results in loss of data around 53%. so it is not b est practice to remove outliers with high perce ntage loss of data.

From the document Provided we kno w that data is expensive and we cannot lose m ore than 7-8% of the data. so proceeding witho ut outlier removal

10.Splitting data to fit any machine learning model

After we have performed data cleaning, data visualizations, and learned details about our data it is time to fit the first machine learning model into it.

Separating features from the target variable:

We should start with separating features for our model from the target variable. Notice that in our case all columns except 'SalePrice' are features that we want to use for the model. Our target variable is 'SalePrice'. We can use the following code to do target separation.

```
#seperating SalePrice from other attributes
x_l=df_train.drop(["SalePrice"],axis=1)
y=df_train["SalePrice"]
```

Using train_test_split:

Let's now use train_test_split from the function from scikit-learn to divide features data (x) and target data (y) even further into train and test.

```
#importing Libraries
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.20,random_state=1)
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((856, 65), (215, 65), (856,), (215,))
```

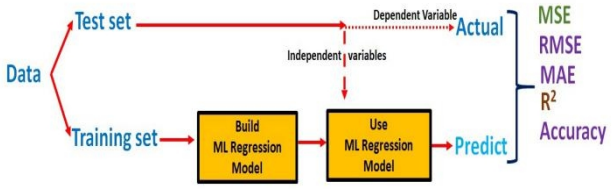
The data that you have prepared is now ready to be fed to the machine learning model.

Scaling Input

Many machine learning algorithms perform better when numerical input variables are scaled to a standard range. This includes algorithms that use a weighted sum of the input, like logical regression, and algorithms that use distance measures, like k-nearest neighbors.

11.Machine Learning Models

Various machine learning models are implemented to validate and predict the House sale price.



First we will import all the necessary Libraries of our Machine Learning Model.

```
#importing our model libraries
from sklearn.linear_model import LinearRegression,Lasso,Ridge,ElasticNet
lr=LinearRegression()
ls=Lasso()
rd=Ridge()
en=ElasticNet()
from sklearn.svm import SVR
svr=SVR()
from sklearn.neighbors import KNeighborsRegressor
knn=KNeighborsRegressor()
from sklearn.tree import DecisionTreeRegressor
dt=DecisionTreeRegressor()
#importing error Metrics
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

```

Once the libraries are imported,we proceed to predict the required output with the help of different Machine Learning algorithm. The Different Machine learning algorithm includes

1. Linear Regression
2. Lasso
3. Ridge
4. Elastic Net
5. SVR
6. K Neighbors Regressor
7. Decision Tree Regressor
8. Random Forest Regressor
9. Ada -Boost Regressor

```
#scoring the model
model=[lr,svr,knn,ls,rd,en]
for m in model:
    m.fit(x_train,y_train)
    print("Training score of ",m,"is",m.score(x_train,y_train))
    predm=m.predict(x_test)
    print("r2_score is :",r2_score(y_test,predm))
    print("Error:")
    print("mean_absolute_error is :",mean_absolute_error(y_test,predm))
    print("mean_squared_error is :",mean_squared_error(y_test,predm))
    print("root mean_absolute_error is :",np.sqrt(mean_squared_error(y_test,predm)))

    print("**************************************************")
    print("\n\n")

```

```
#using Random Forest Regressor
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
rf_pred=rf.predict(x_test)
print("score is",rf.score(x_train,y_train))
print("r2 score is",r2_score(y_test,rf_pred))
print("mean absolute error is : ",mean_absolute_error(y_test,rf_pred))
print("mean squared error is : ",mean_squared_error(y_test,rf_pred))
print("root mean absolute error is : ",np.sqrt(mean_absolute_error(y_test,rf_pred)))

```

```
#using AdaBoostRegressor
from sklearn.ensemble import AdaBoostRegressor
rf=RandomForestRegressor()
ada=AdaBoostRegressor(base_estimator=rf,n_estimators=20,learning_rate=0.1,random_state=1)
ada.fit(x_train,y_train)
ada_pred=ada.predict(x_test)
ada_score=ada.score(x_train,y_train)
print("score is",ada_score)
print("r2 score is",r2_score(y_test,ada_pred))
print("mean absolute error is : ",mean_absolute_error(y_test,ada_pred))
print("mean squared error is : ",mean_squared_error(y_test,ada_pred))
print("root mean absolute error is : ",np.sqrt(mean_absolute_error(y_test,ada_pred)))
```

Model	Training accuracy	Testing accuracy
Linear Regression	81.89%	74.79%
KNN	77.45%	74.79%
Lasso	81.89%	74.78%
Ridge	81.89%	74.81%
ElasticNet	79.82%	74.93%
Random Forest	97.49%	83.22%
AdaBoost	98.62%	84.82%

12.MODEL EVALUATION

The model developed in this research will be tested using several methods such as Mean Absolute Percentage Error(MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). MAPE is calculated by making an average percentage of the absolute error of each predicted result. Thus, MAPE can indicate how much prediction error.

R-squared (R^2), Mean Squared Error (MSE), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are the most commonly used metrics to measure accuracy for continuous variables. In this post, we will observe the coefficient of variables (CoV) effect on the MAE, MSE, R^2, and accuracy. We will apply the same linear regression to 4 different data which has variables with different coefficients to explain how and why the MSE, MAE, R^2, and Accuracy are changing. First, while we keep the MSE and MAE fixed, we will observe the R^2 and accuracy with the change of coefficient of variables. Secondly, while we keep the R^2, and accuracy set constant, we will

observe the MSE and MAE with the change of coefficient of variables.

I) Mean Absolute Error (MAE):

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

If the absolute value is not taken (the signs of the errors are not removed), the average error becomes the Mean Bias Error (MBE) and is usually intended to measure average model bias. MBE can convey useful information, but should be interpreted cautiously because positive and negative errors will cancel out.

II) Mean absolute percentage error (MAPE):

The mean absolute percentage error (MAPE) is a measure of how accurate a forecast system is. It measures this accuracy as a percentage, and can be calculated as the average absolute percent error for each time period minus actual values divided by actual values.

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Where:

n is the number of fitted points,

At is the actual value,

Ft is the forecast value.

Σ is summation notation (the absolute value is summed for every forecasted point in time).

The mean absolute percentage error (MAPE) is the most common measure used to forecast error, and works best if there are no extremes to the data (and no zeros).

III) Root Mean Square Error (RMSE):

RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.

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

MODEL	MAE	MAPE	RMSE
Linear Regression	23190.56	1751	41845.17
KNN	24196.47	1809	42538.72
Lasso	23182	175129	41848.44
Ridge	23171.44	17490	41821.99
ElasticNet	22447.08	174135	41729.54
Random Forest	17512.76	116489	132.33
AdaBoost	16779.58	105435	129.53

13.PREDICTION

Since we have evaluated all models by using **MAE,RMSE,MAPE** we will predict by using model which has highest accuracy.Here we can choose Adaboost Regressor models to predict the Sale Price of the house

13.1 Hyper Tuning using GridSearchCV

Hyperparameters are crucial as they control the overall behavior of a machine learning model. The ultimate goal is to find an optimal combination of hyperparameters that minimizes a predefined loss function to give better results.

```
#AdaBoostRegressor
#using GridSearchCV
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import AdaBoostRegressor
parameters={"n_estimators" :[1,10,100],
            "learning_rate" : [0.15,0.1,0.05,0.01],
            "loss" : ['linear', 'square', 'exponential']}
ada=AdaBoostRegressor()
clf=GridSearchCV(ada,parameters)
clf.fit(x_train,y_train)
print(clf.best_params_)
```

```
{'learning_rate': 0.15, 'loss': 'exponential', 'n_estimators': 100}
```

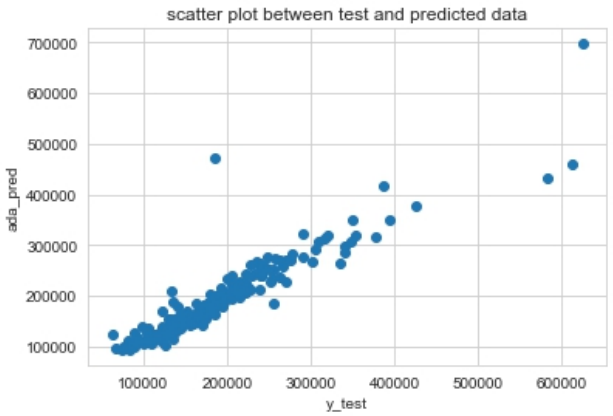
Here we first find the best parameters for the Adaboost regressor Model and indulge it to the model to improve our predicting accuracy. There are several ways for finding the parameters, here we use the most powerful and most commonly used method named GridSearchCV.

13.2Further Evaluation

13.2.1 Cross-validation

The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).

13.2.2 Scatter Plot b/w Test-data and Predicted-data



14. CONCLUSION

Data cleaning is the first step while performing data analysis. Exploratory data analytics helps one to understand the dataset and the dependency among the attributes. EDA is used to figure out the relationship between the features of the dataset.This is done by using various graphical techniques. The one used above barplot ,countplot and heatmaps. By applying EDA some conclusions are drawn and facts are found. In feature engineering the actual parameters to be used while designing the training model and prediction model is found out on the basis of exploratory data analytics process.

We used Machine Learning models to predict the sale price of the house.

Adaboost Regressor proves to be best among all with an accuracy of ***85.55820173733075***. This means the predictive power of Adaboost Regressor in this dataset with the chosen features is very high.

It is clearly stated that the accuracy of the models may vary when the choice of feature modelling is different. Ideally linear regression and Random Forest Regressor are the models which give a good level of accuracy when it comes to classification problem.

The results answer the research questions as follows:

Question 1 – Which machine learning algorithm performs better and has the most

accurate result in house price prediction? And why?

AdaBoost made the best performance overall when both R2 and RMSE scores are taking into consideration. It has achieved the best performance due to its L1 norm regularisation for assigning zero weights to the insignificant features.

Question 2 – What are the factors that have affected house prices in ?

"BldgType",'ExterQual',"BsmtQual","HeatingQC","KitchenQual","GarageType","GarageFinish" has a weak correlation with the house prices. Which means there are lower likelihood relationships between these factors and sale price. However, when these factors increase the house price decrease.

Future Work

Future work on this study could be divided into four main areas to improve the result even further. Which can be done by:

I. The used pre-processing methods do help in the prediction accuracy. However, experimenting with different combinations of pre-processing methods to achieve better prediction accuracy.

II. Make use of the available features and if they could be combined as binning features has shown that the data got improved.

III. The correlation has shown the association in the testing data. Thus, attempting to enhance the Training data is required to make rich with features that vary and can provide a strong correlation relationship.

IV. We find more percentage of missing value.This explains that the data-set needs more engineering towards it.Many true values should be added In the future.

REFERENCES :

1.https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471

2.https://www.sas.com/en_us/insights/analytics/machinelearning.html#machine-learning-importance

3.http://www.wired.co.uk/article/machine-learning-ai-explained

4.https://deeplearning4j.org/ai-machinelearning-deeplearning

5.David E. Rapach , Jack K. Strauss " Forecasting real housing price growth in the Eighth District states"

6. Vasilios Plakandaras+ and Theophilos , Rangan Gupta*, Periklis Gogas "Forecasting the U.S. Real House Price Index"

7. Gupta and Das (2010) Forecasting the US Real House Price Index: Structural and Non-Structural Models with and without Fundamentals

8. Rangan Gupta “Forecasting US real house price returns over 1831– 2013: evidence from copula models”