FLIP ROBO

Housing: Price Prediction

Submitted by:

Yatika Taneja

ACKNOWLEDGMENT

.

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

I wish to thank, all the faculties in data trained academy as this project utilized knowledge gained from every course that formed the Data science program.

INTRODUCTION

Objective of the study

The objective of our project is 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?

Business Model

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. For the same purpose, the company has collected a data set from the sale of houses in Australia.

You are required to model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.

Literature Survey

The latest worldwide financial crisis restored a sharp enthusiasm toward both academic and strategy circles on the part of asset costs and specifically lodging costs clinched alongside monetary movement. As Lamer (2007) notes those lodging showcase predicted eight of the ten post globe War ii recessions, acting Concerning illustration An heading woman for those true segment of the economy.

Truth be told he dives Likewise significantly Concerning illustration with state that “Housing is those benefits of the business cycle”. Vargas and silva (2008) contend that lodging costs alterations assume a paramount part in the determination of the stage of the business cycle. When those economy booms, development and work in the lodging division expand quickly should react should overabundance demand, quickly pushing ostensible house costs upwards. Throughout those withdrawal phase, the drop in private money lessens aggravate interest Also ostensible house costs.

By ostensible house costs normally fall sluggishly since householders would unwilling on bring down their costs. The majority of the conformity will be attained through declines clinched alongside bargains volume bringing about an drop in the development segment and the lodging built vocation.

Moreover, Throughout withdrawal and subsidence true house costs fall quickly Likewise general inflationary patterns diminish true house costs much with sticky perceived costs.

Recently, a few writers scope to experimental discoveries that house costs can make instrumental molding to determining yield.

(Forni etc, 2003; stock and Watson, 2003; Gupta Furthermore Das, 2010; das etc, 2009; 2010; 2011; Gupta and Hartley, 2013). Those lodging development division speaks to an expansive and only aggregate monetary action communicated in the GDP.

Consequently, Concerning illustration it reflects an extensive parcel of the general riches of the economy, house costs variances can make a pointer of the Development about GDP (Case etc, 2005).

Concerning illustration it is those body of evidence with different assets, those development for house costs can make Additionally an pointer of the future course from claiming expansion (Gupta Also Kabundi, 2010).

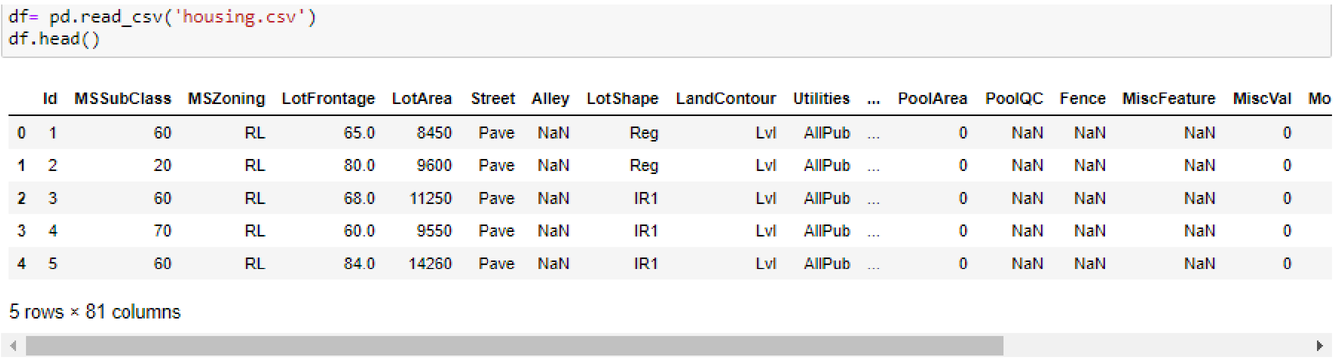
Overall, exact determining of the Development way from claiming house costs could make a suitable apparatus both on house business members and fiscal strategy powers. There is huge literature writing in regards to U.S. house prices. Rapach Furthermore strauss (2007) use an auto regressive dispersed slack (ARDL) model framework, holding 25 determinants with conjecture genuine lodging cost development to the unique states of the elected Reserve's eighth region. They discover that ARDL models tend should beat a benchmark AR model.

Analytical Problem Framing

Data Sources

The sample data is provided to us from our client database. The dataset has 1470 observations and 81 features. SalePrice is the target variable

Dataset:



```
df= pd.read_csv('housing.csv')
df.head()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	Mo
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	

5 rows × 81 columns

Data Description

Variable	**Description**
MSSubClass	Identifies the type of dwelling involved in the sale.
MSZoning	Identifies the general zoning classification of the sale.
LotFrontage	Linear feet of street connected to property
LotArea	Lot size in square feet

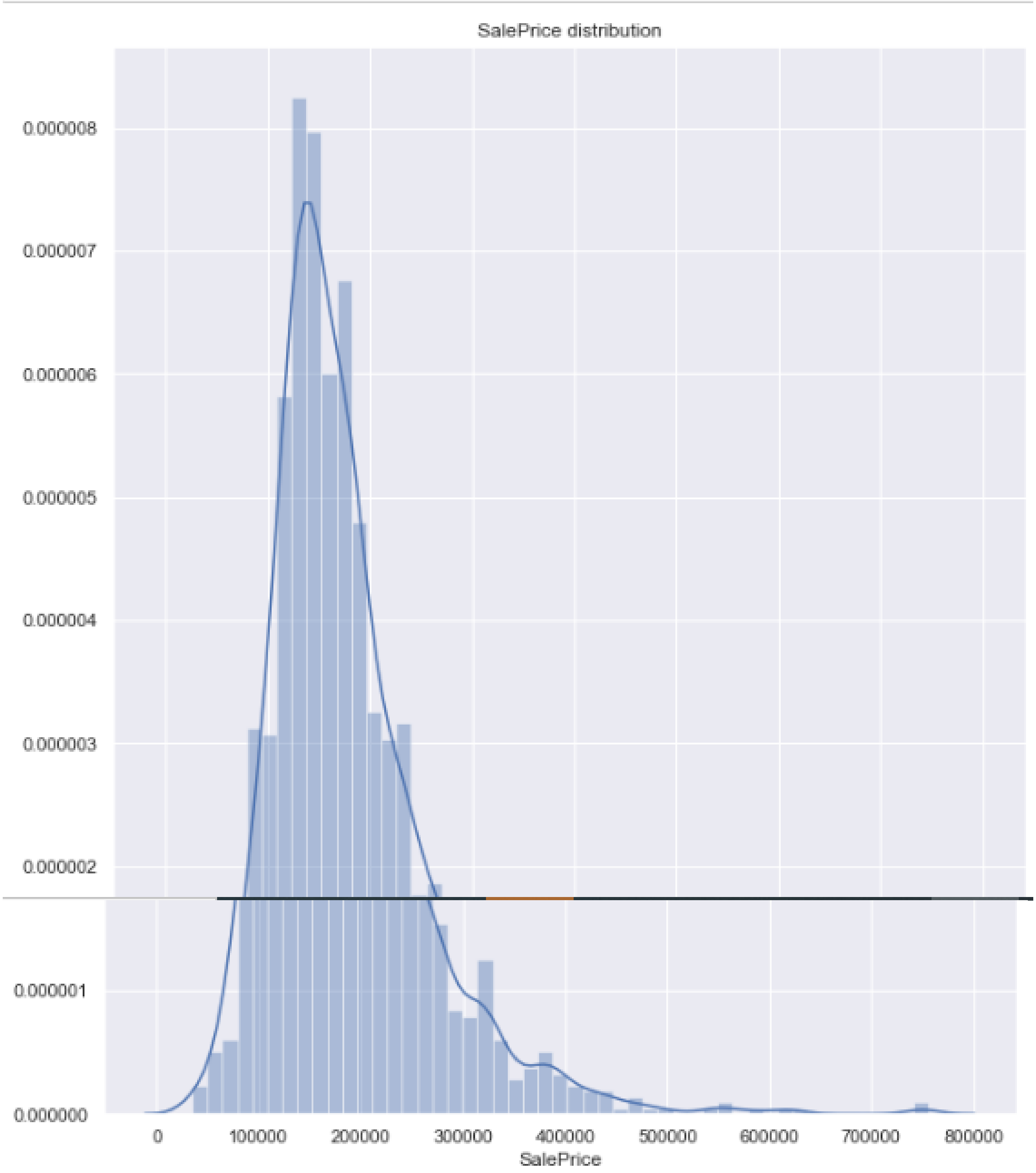
Street	Type of road access to property
Alley	Type of alley access to property
LotShape	General shape of property
LandContour	Flatness of the property
Utilities:	Type of utilities available
LotConfig	Lot configuration
LandSlope	Slope of property
Neighborhood	Physical locations within Ames city limits
Condition1	Proximity to various conditions
Condition2	Proximity to various conditions (if more than one is present)
BldgType	Type of dwelling
HouseStyle	Style of dwelling
OverallQual	Rates the overall material and finish of the house
OverallCond	Rates the overall condition of the house
YearBuilt	Original construction date
YearRemodAdd	Remodel date (same as construction date if no remodeling or additions)
RoofStyle	Type of roof
RoofMatl	Roof material
Exterior1st	Exterior covering on house
Exterior2nd	Exterior covering on house (if more than one material)
MasVnrType	Masonry veneer type
MasVnrArea	Masonry veneer area in square feet
ExterQual	Evaluates the quality of the material on the exterior
ExterCond	Evaluates the present condition of the material on the exterior

Foundation	Type of foundation
BsmtQual	Evaluates the height of the basement
BsmtCond	Evaluates the general condition of the basement
BsmtExposure	Refers to walkout or garden level walls
BsmtFinType1	Rating of basement finished area
BsmtFinSF1	Type 1 finished square feet
BsmtFinType2	Rating of basement finished area (if multiple types)
BsmtFinSF2	Type 2 finished square feet
BsmtUnfSF	Unfinished square feet of basement area
TotalBsmtSF	Total square feet of basement area
Heating	Type of heating
HeatingQC	Heating quality and condition
CentralAir	Central air conditioning
Electrical	Electrical system
1stFlrSF	First Floor square feet
2ndFlrSF	Second floor square feet
LowQualFinSF	Low quality finished square feet (all floors)
GrLivArea	Above grade (ground) living area square feet
BsmtFullBath	Basement full bathrooms
BsmtHalfBath	Basement half bathrooms
FullBath	Full bathrooms above grade
HalfBath	Half baths above grade
Bedroom	Bedrooms above grade (does NOT include basement bedrooms)
Kitchen	Kitchens above grade
KitchenQual	Kitchen quality

TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)
Functional	Home functionality (Assume typical unless deductions are warranted)
Fireplaces	Number of fireplaces
FireplaceQu	Fireplace quality
GarageType	Garage location
GarageYrBlt	Year garage was built
GarageFinish	Interior finish of the garage
GarageCars	Size of garage in car capacity
GarageArea	Size of garage in square feet
GarageQual	Garage quality
GarageCond	Garage condition
PavedDrive:	Paved driveway
WoodDeckSF	Wood deck area in square feet
OpenPorchSF	Open porch area in square feet
EnclosedPorch	Enclosed porch area in square feet
3SsnPorch	Three season porch area in square feet
ScreenPorch	Screen porch area in square feet
PoolArea	Pool area in square feet
PoolQC	Pool quality
Fence	Fence quality
MiscFeature	Miscellaneous feature not covered in other categories
MiscVal	$Value of miscellaneous feature
MoSold	Month Sold (MM)
YrSold	Year Sold (YYYY)
SaleType	Type of sale
SaleCondition	Condition of sale

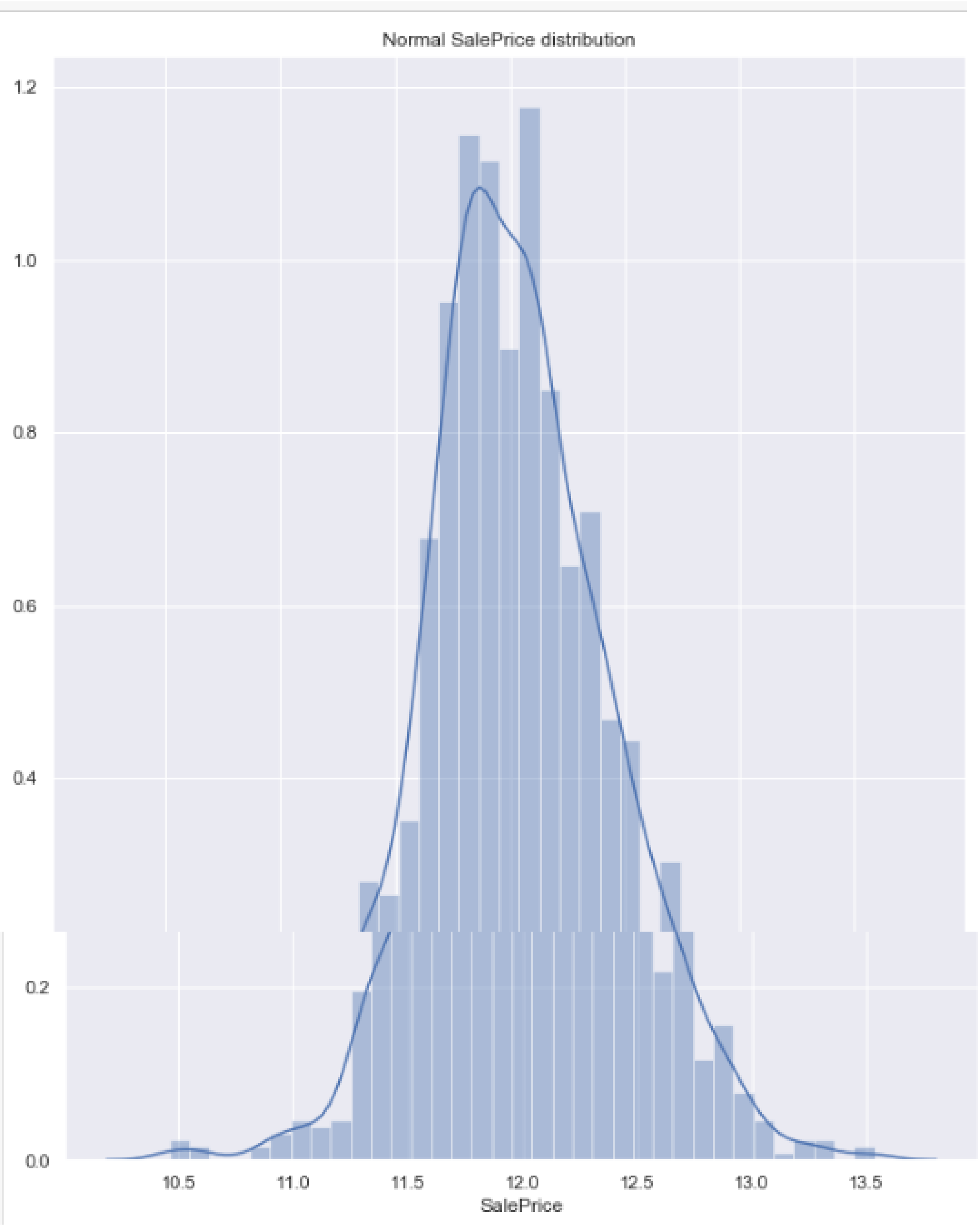
Data Preprocessing

- Distribution of ‘SalePrice’

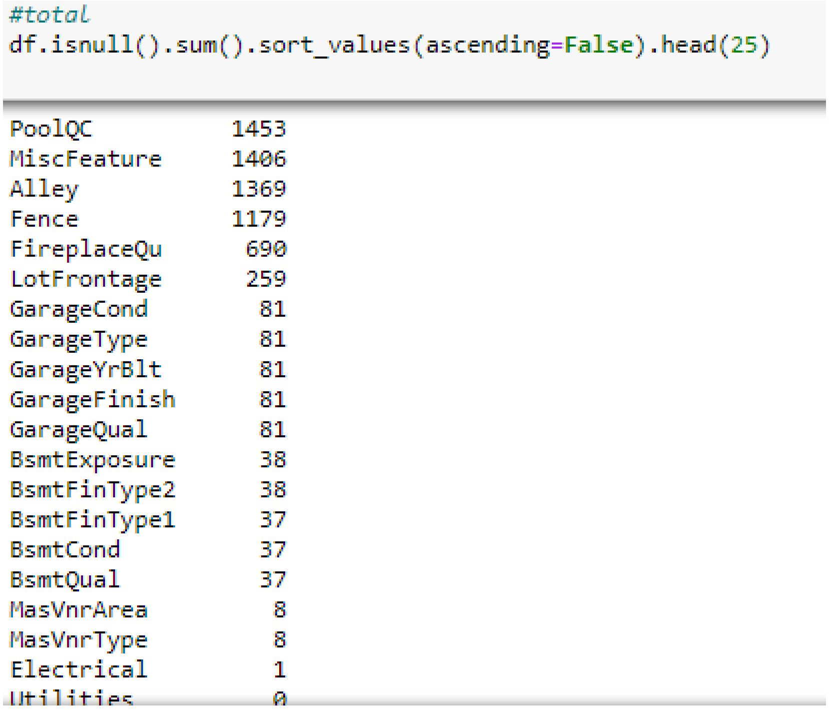


Deviate from the normal distribution.
Have appreciable positive skewness.

After Transformation:



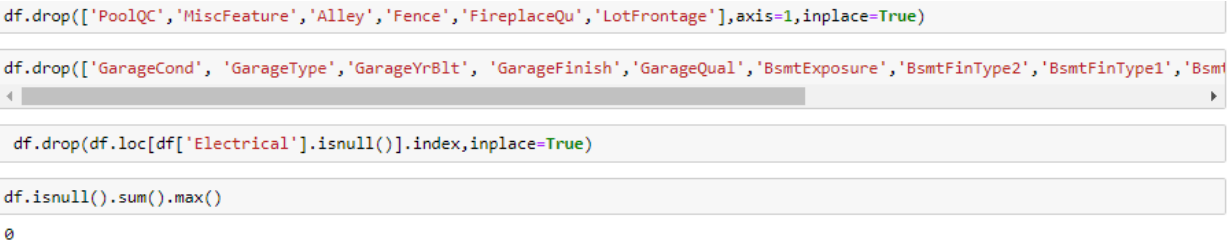
➢ Dealing with null values



```
#total
df.isnull().sum().sort_values(ascending=False).head(25)
```

```
PoolQC          1453
MiscFeature     1406
Alley           1369
Fence           1179
FireplaceQu      690
LotFrontage      259
GarageCond        81
GarageType        81
GarageYrBlt       81
GarageFinish      81
GarageQual        81
BsmtExposure      38
BsmtFinType2      38
BsmtFinType1      37
BsmtCond          37
BsmtQual          37
MasVnrArea         8
MasVnrType         8
Electrical         1
Utilities          0
```

Dropping columns with null values:



```
df.drop(['PoolQC','MiscFeature','Alley','Fence','FireplaceQu','LotFrontage'],axis=1,inplace=True)
```

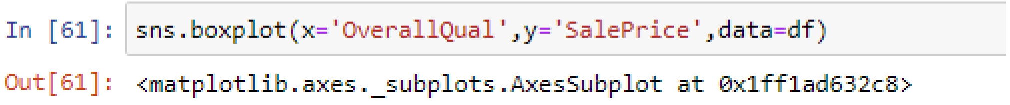
```
df.drop(['GarageCond', 'GarageType','GarageYrBlt', 'GarageFinish','GarageQual','BsmtExposure','BsmtFinType2','BsmtFinType1','Bsm
```

```
 df.drop(df.loc[df['Electrical'].isnull()].index,inplace=True)
```

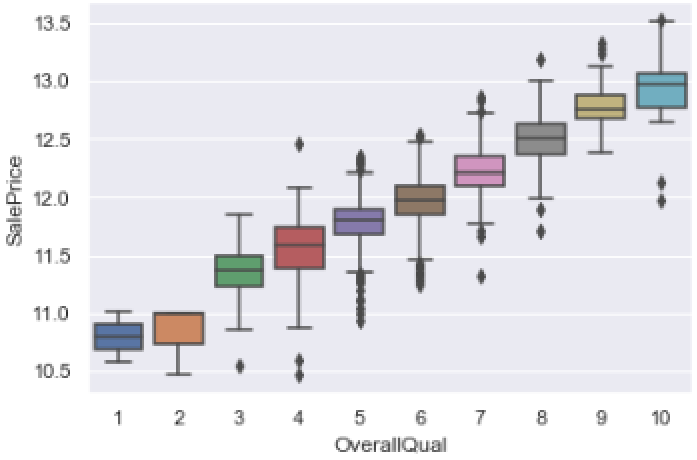
```
df.isnull().sum().max()
```

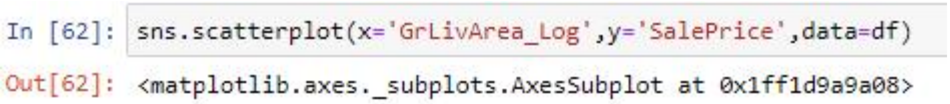
```
0
```

- Dealing with outliers

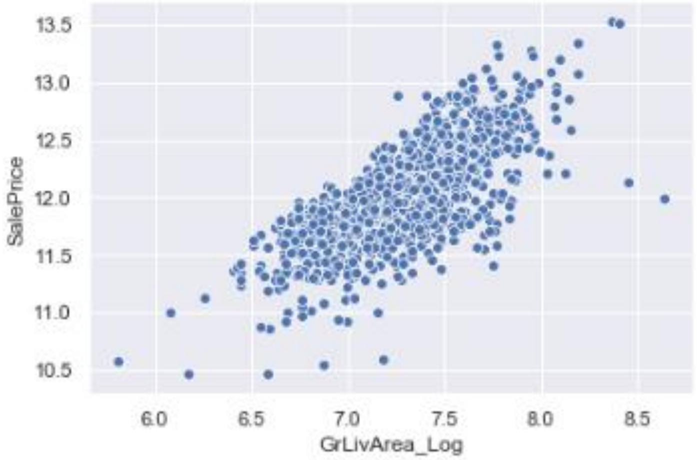


```
In [61]: sns.boxplot(x='OverallQual',y='SalePrice',data=df)
Out[61]: <matplotlib.axes._subplots.AxesSubplot at 0x1ff1ad632c8>
```

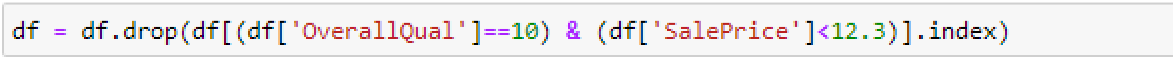




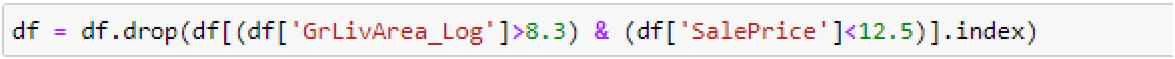
```
In [62]: sns.scatterplot(x='GrLivArea_Log',y='SalePrice',data=df)
Out[62]: <matplotlib.axes._subplots.AxesSubplot at 0x1ff1d9a9a08>
```



Removing outliers in ‘OverallQuall’ and ‘GrLivArea_Log’:

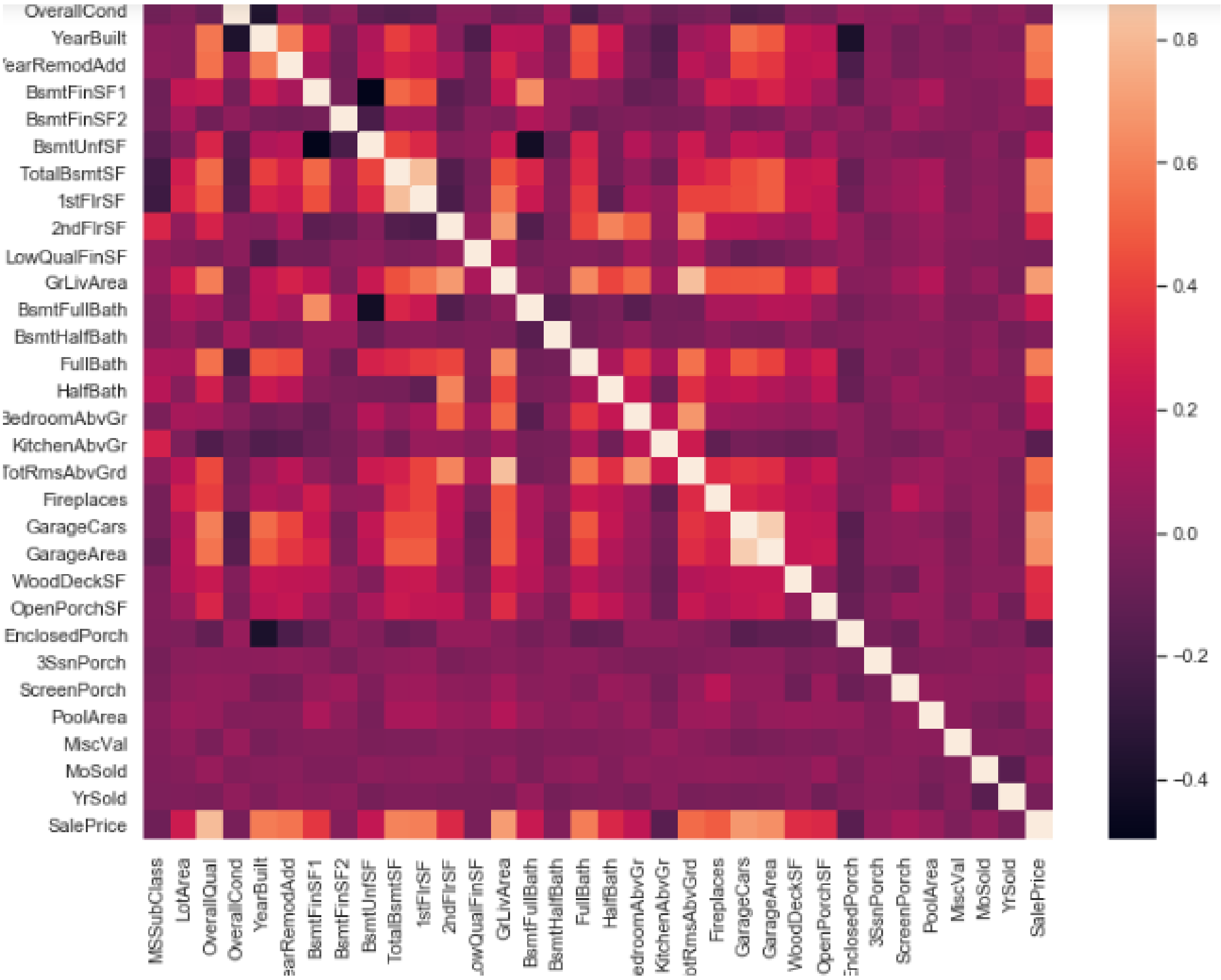


```
df = df.drop(df[(df['OverallQual']==10) & (df['SalePrice']<12.3)].index)
```



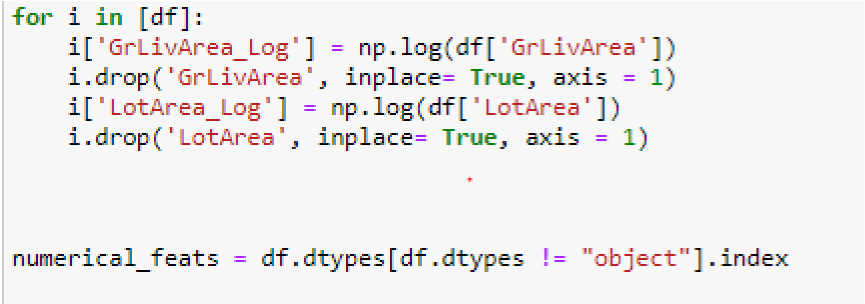
```
df = df.drop(df[(df['GrLivArea_Log']>8.3) & (df['SalePrice']<12.5)].index)
```

Feature Correlation with ‘SalePrice’

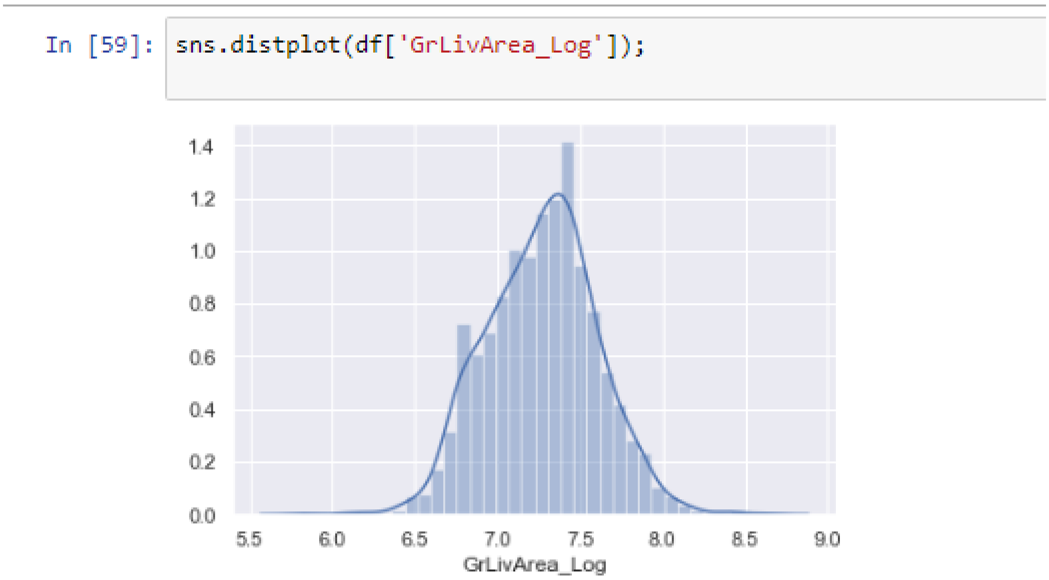


Data Transformation

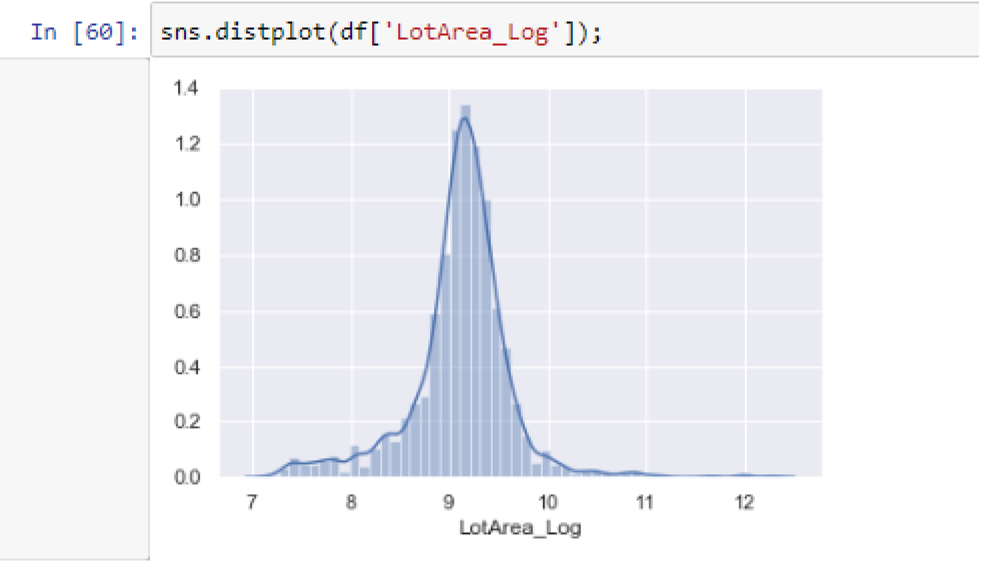
Log Transformation:



```
for i in [df]:
    i['GrLivArea_Log'] = np.log(df['GrLivArea'])
    i.drop('GrLivArea', inplace= True, axis = 1)
    i['LotArea_Log'] = np.log(df['LotArea'])
    i.drop('LotArea', inplace= True, axis = 1)


numerical_feats = df.dtypes[df.dtypes != "object"].index
```



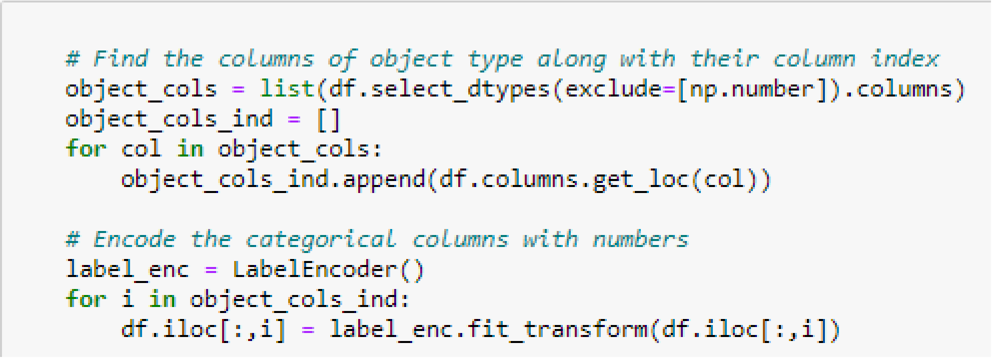
```
In [59]: sns.distplot(df['GrLivArea_Log']);
```



```
In [60]: sns.distplot(df['LotArea_Log']);
```

➢ Label Encoding

Encode categorical columns with numbers

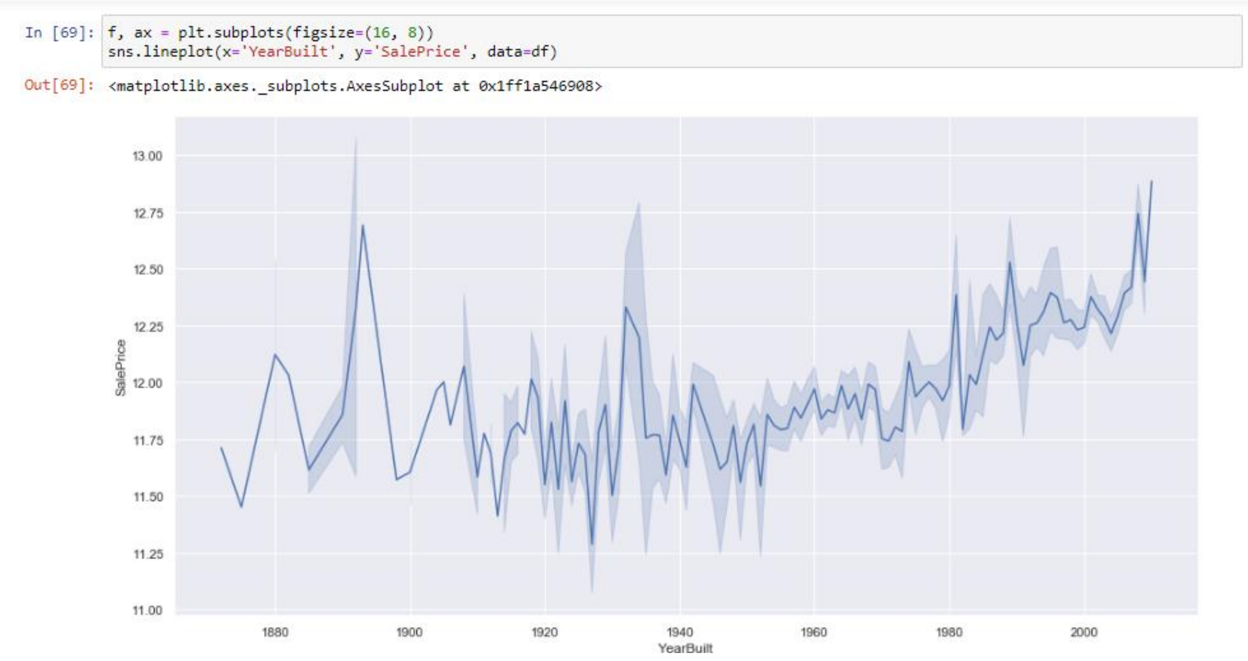


```
# Find the columns of object type along with their column index
object_cols = list(df.select_dtypes(exclude=[np.number]).columns)
object_cols_ind = []
for col in object_cols:
    object_cols_ind.append(df.columns.get_loc(col))

# Encode the categorical columns with numbers
label_enc = LabelEncoder()
for i in object_cols_ind:
    df.iloc[:,i] = label_enc.fit_transform(df.iloc[:,i])
```

Data Visualization

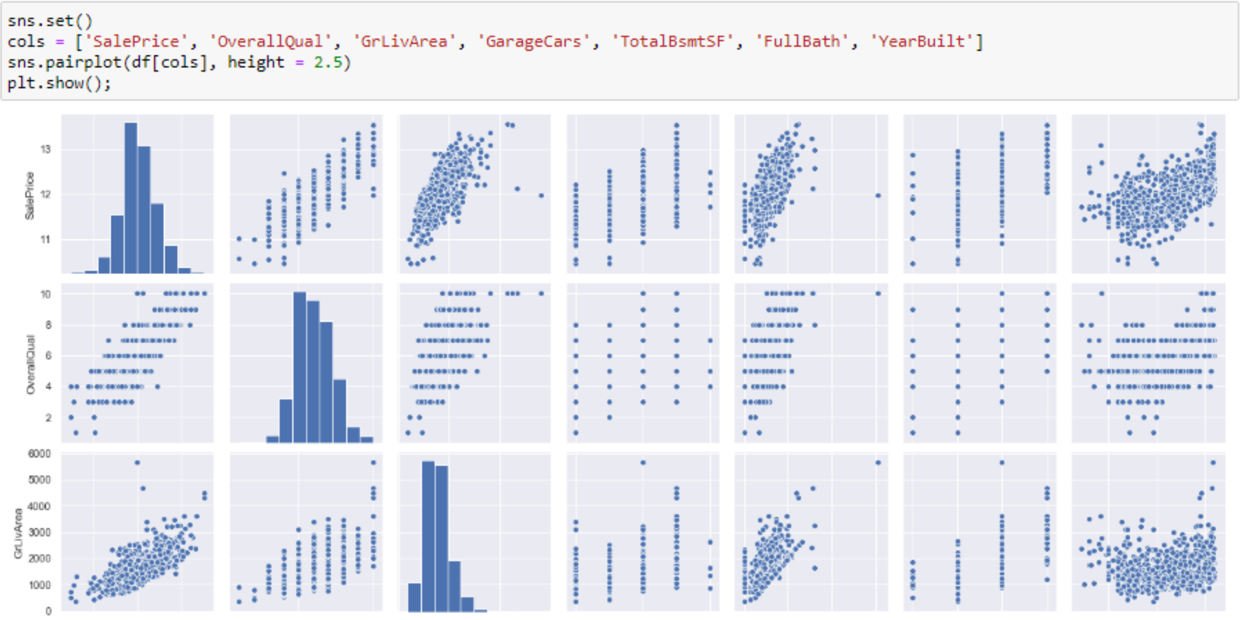
1. Relationship with numerical variables

SalePrice v/s YearBuilt

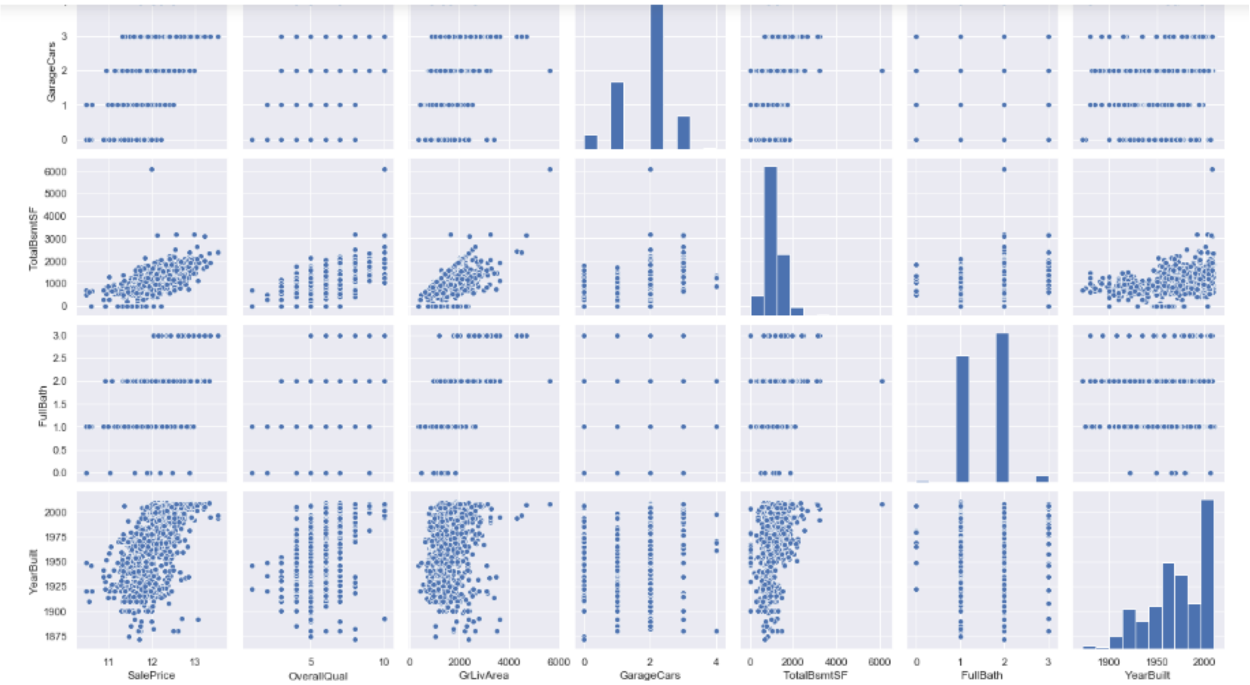


```
In [69]: f, ax = plt.subplots(figsize=(16, 8))
         sns.lineplot(x='YearBuilt', y='SalePrice', data=df)

Out[69]: <matplotlib.axes._subplots.AxesSubplot at 0x1ff1a546908>
```

Pair Plot

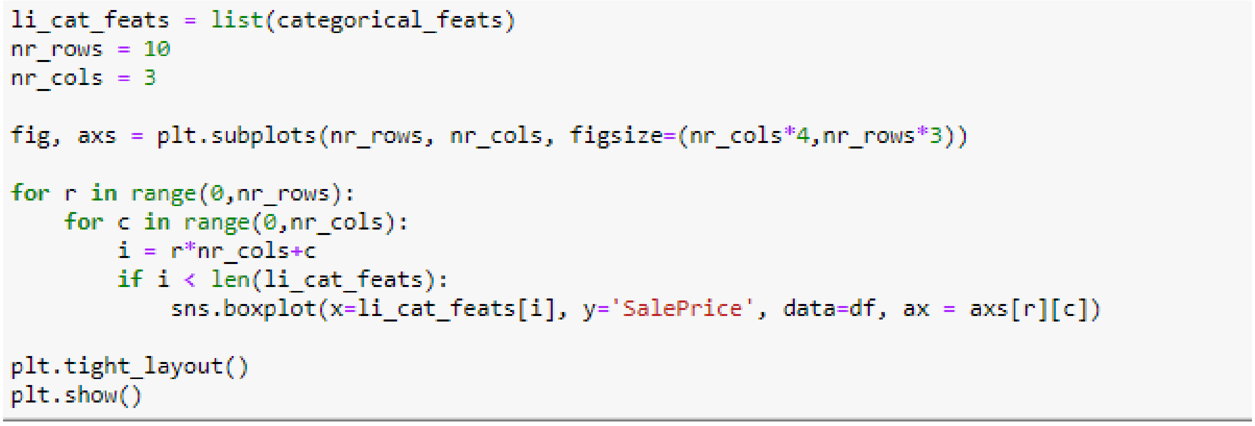


```
sns.set()
cols = ['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars', 'TotalBsmtSF', 'FullBath', 'YearBuilt']
sns.pairplot(df[cols], height = 2.5)
plt.show();
```

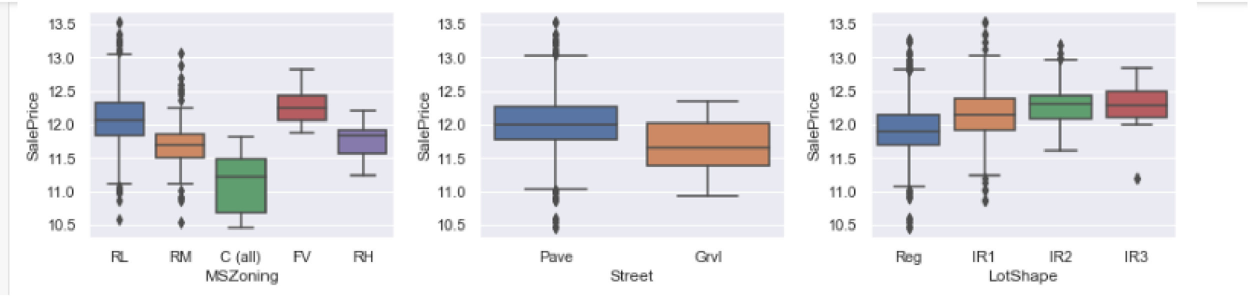


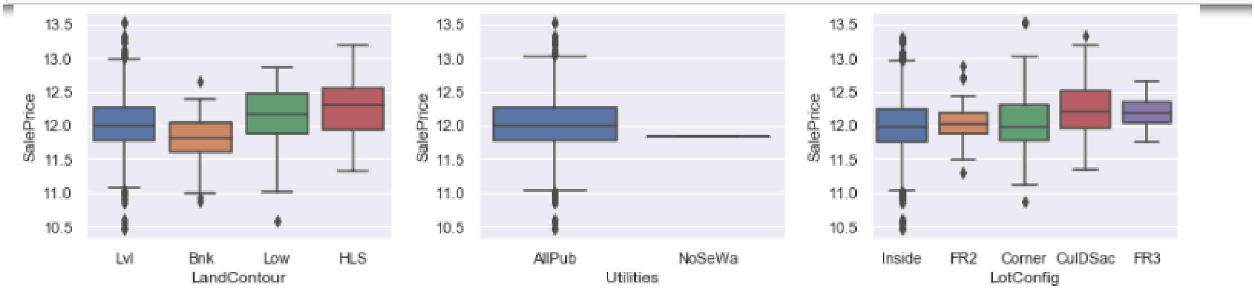
2. Relationship with categorical variables

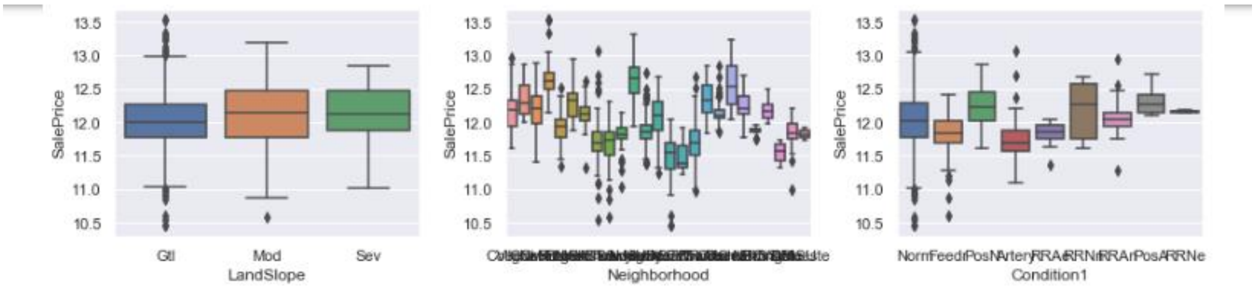
Box Plot

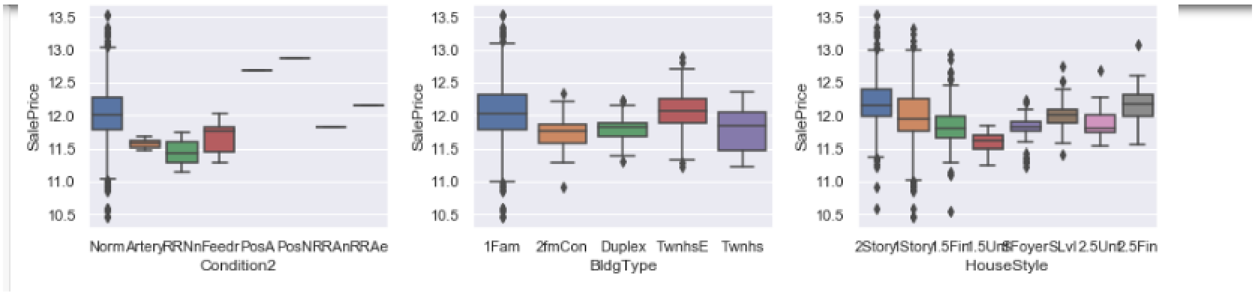


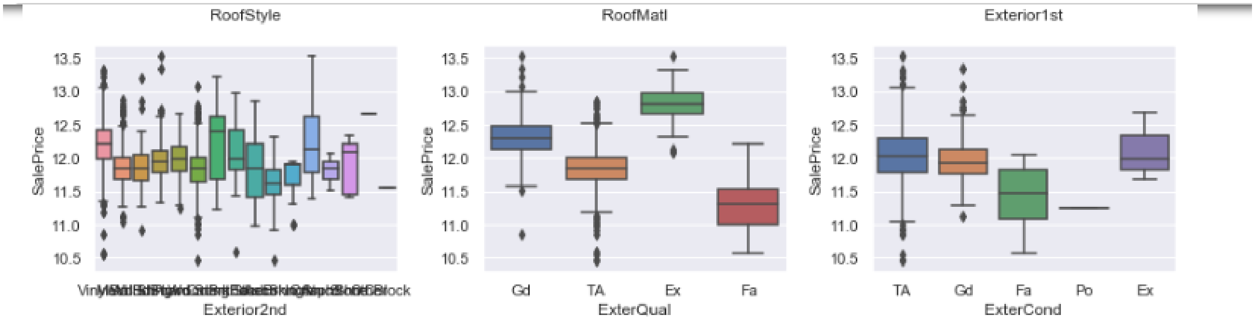
```
li_cat_feats = list(categorical_feats)
nr_rows = 10
nr_cols = 3

fig, axs = plt.subplots(nr_rows, nr_cols, figsize=(nr_cols*4,nr_rows*3))

for r in range(0,nr_rows):
    for c in range(0,nr_cols):
        i = r*nr_cols+c
        if i < len(li_cat_feats):
            sns.boxplot(x=li_cat_feats[i], y='SalePrice', data=df, ax = axs[r][c])

plt.tight_layout()
plt.show()
```

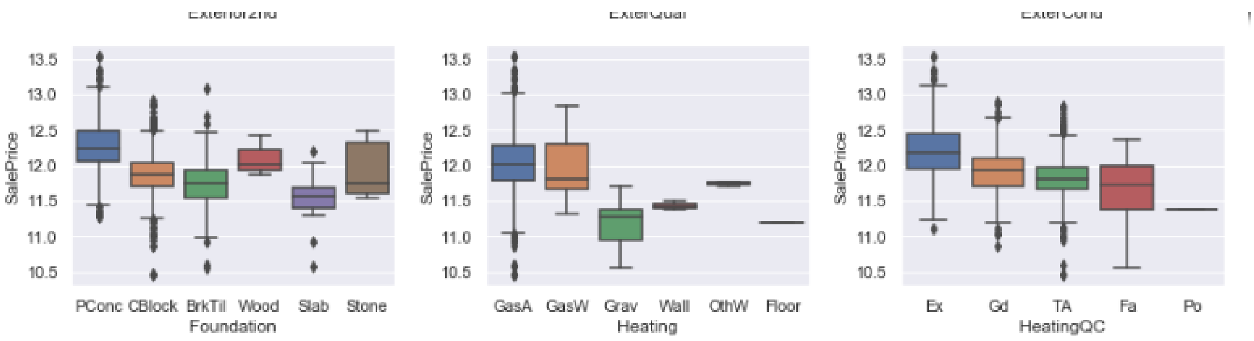


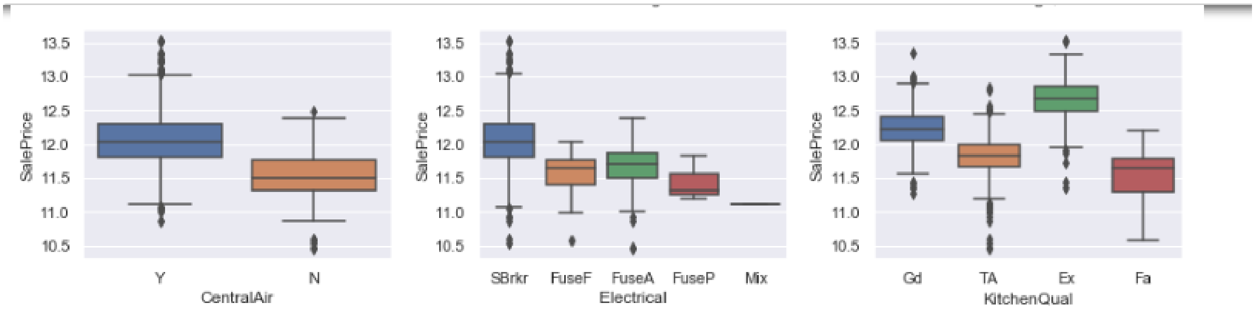

SalePrice
13.5
13.0
12.5
12.0
11.5
11.0
10.5
Lvl
Bnk
Low
HLS
LandContour
AllPub
NoSeWa
Utilities
Inside
FR2
Corner
CulDSac
FR3
LotConfig

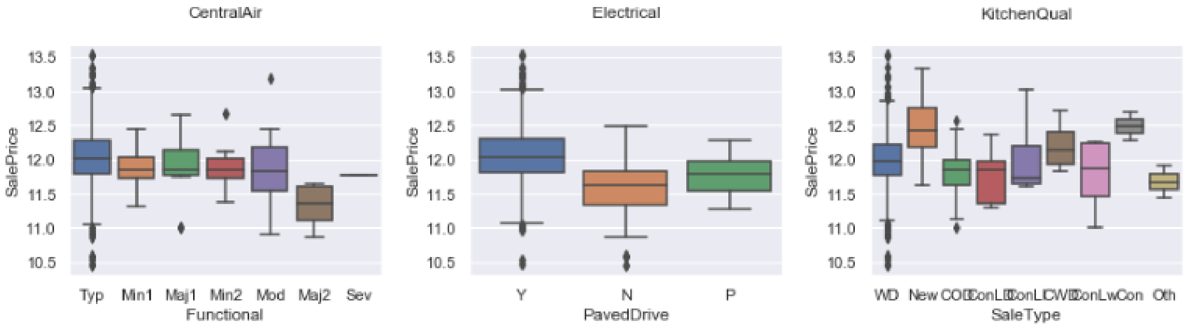

SalePrice
13.5
13.0
12.5
12.0
11.5
11.0
10.5
Gtl
Mod
Sev
LandSlope
Neighborhood
Norm
Feedr
PosN
Artery
RRAe
RRNn
RRAn
PosA
RRNe
Condition1


SalePrice
13.5
13.0
12.5
12.0
11.5
11.0
10.5
Norm
Artery
RRNn
Feedr
PosA
PosN
RRAn
RRAe
Condition2
1Fam
2fmCon
Duplex
TwnhsE
Twnhs
BldgType
2Story
1Story
1.5Fin
1.5Unf
SFoyer
SLvl
2.5Unf
2.5Fin
HouseStyle


RoofStyle
RoofMatl
Exterior1st
SalePrice
13.5
13.0
12.5
12.0
11.5
11.0
10.5
Exterior2nd
Gd
TA
Ex
Fa
ExterQual
TA
Gd
Fa
Po
Ex
ExterCond


Exterior2nd
ExterQual
ExterCond
SalePrice
13.5
13.0
12.5
12.0
11.5
11.0
10.5
PConc
CBlock
BrkTil
Wood
Slab
Stone
Foundation
GasA
GasW
Grav
Wall
OthW
Floor
Heating
Ex
Gd
TA
Fa
Po
HeatingQC


SalePrice
13.5
13.0
12.5
12.0
11.5
11.0
10.5
Y
N
CentralAir
SBrkr
FuseF
FuseA
FuseP
Mix
Electrical
Gd
TA
Ex
Fa
KitchenQual


CentralAir
Electrical
KitchenQual
SalePrice
13.5
13.0
12.5
12.0
11.5
11.0
10.5
Typ
Min1
Maj1
Min2
Mod
Maj2
Sev
Functional
Y
N
P
PavedDrive
WD
New
COD
ConLD
ConLI
CWD
ConLw
Con
Oth
SaleType

Model/s Development and Evaluation

Models Applied

Linear Regression:

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable *causes* the other (for example, higher SAT scores do not *cause* higher college grades), but that there is some significant association between the two variables. A scatterplot can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the correlation coefficient, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

A linear regression line has an equation of the form $Y = a + bX$, where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when $x = 0$).

Gradient Boosting Regression:

The Gradient Boosting Machine is a powerful ensemble machine learning algorithm that uses decision trees.

Boosting is a general ensemble technique that involves sequentially adding models to the ensemble where subsequent models correct the performance of prior models. AdaBoost was the first algorithm to deliver on the promise of boosting.

Gradient boosting is a generalization of AdaBoosting, improving the performance of the approach and introducing ideas from bootstrap aggregation to further improve the models, such as randomly sampling the samples and features when fitting ensemble members.

Random Forest Regression:

A random forest is an ensemble model that consists of many decision trees. Predictions are made by averaging the predictions of each decision tree. Or, to extend the analogy—much like a forest is a collection of trees, the random forest model is also a collection of decision tree models. This makes random forests a strong modeling technique that's much more powerful than a single decision tree.

Each tree in a random forest is trained on the subset of data provided. The subset is obtained both with respect to rows and columns. This means each random forest tree is trained on a random data point sample, while at each decision node, a random set of features is considered for splitting.

In the realm of machine learning, the random forest regression algorithm can be more suitable for regression problems than other common and popular algorithms.

Lasso Regression:

Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of muticollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

The acronym “LASSO” stands for Least Absolute Shrinkage and Selection Operator.

Model Evaluation

We will use stratified 10-fold cross validation to estimate model accuracy.
This will split our dataset into 10 parts, train on 9 and test on 1 and repeat for all combinations of train-test splits.
Stratified means that each fold or split of the dataset will aim to have the same distribution of example by class as exist in the whole training dataset.

k-Fold Cross-Validation

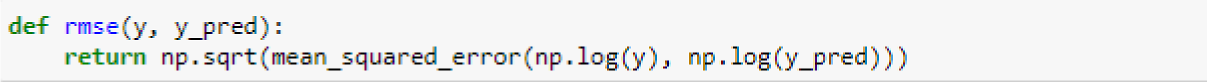
Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

Evaluation Metric: RMSE

RMSE is the standard deviation of the errors which occur when a prediction is made on a dataset. This is the same as MSE (Mean Squared Error) but the root of the value is considered while determining the accuracy of the model.



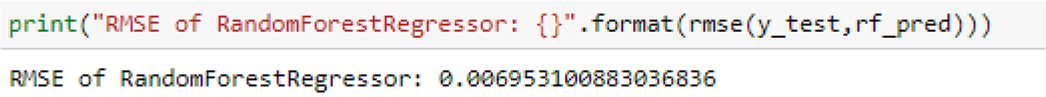
```
def rmse(y, y_pred):
    return np.sqrt(mean_squared_error(np.log(y), np.log(y_pred)))
```

We will be using the scoring variable when we run build and evaluate each model next.

We now have 4 models and rmse estimations for each. We need to compare the models to each other and select the most suitable model

Result:

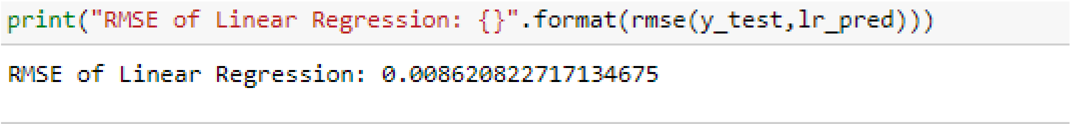
1. Random Forest Regressor



```
print("RMSE of RandomForestRegressor: {}".format(rmse(y_test,rf_pred)))
```

```
RMSE of RandomForestRegressor: 0.006953100883036836
```

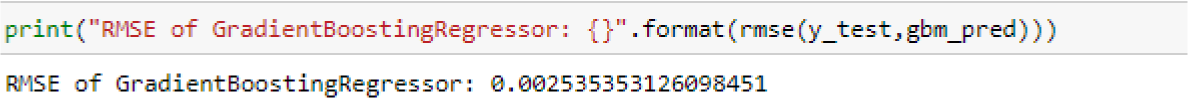
2. Linear Regression



```
print("RMSE of Linear Regression: {}".format(rmse(y_test,lr_pred)))
```

```
RMSE of Linear Regression: 0.008620822717134675
```

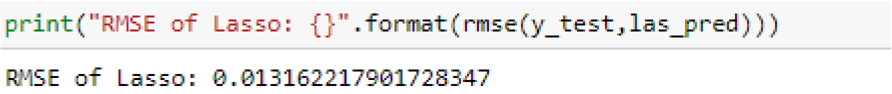
3. Gradient Boosting Regressor



```
print("RMSE of GradientBoostingRegressor: {}".format(rmse(y_test,gbm_pred)))
```

```
RMSE of GradientBoostingRegressor: 0.002535353126098451
```

4. Lasso Regression



```
print("RMSE of Lasso: {}".format(rmse(y_test,las_pred)))
```

```
RMSE of Lasso: 0.013162217901728347
```

Best Model

In this case, Gradient Boosting Regressor has the least estimated rmse value of around 0.0025.

Gradient Boosting Regressor is the best fit for this Dataset and can be used for deploying purposes.

CONCLUSION

In this study, Gradient Boosting Regressor was applied in order to predict the actual value of the prospective properties and decide whether to invest in them or not. Gradient Boosting Regressor was the best model as compared to the other four models with a rmse value of approximately 0.0025.

It model the price of houses with the available independent variables and can 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.