HOUSING PROJECT

Submitted by:

MIENGANDHA SINHA

ACKNOWLEDGMENT

I would like to express my gratitude to the Company FlipRoboTechnology to give this project to me. In making of this project I hereby used to take help from the references from some websites and also from which is given by the company as sample documentation and details related project and professionals guided me a lot in the project and the other previous projects helped me and guided me in completion of the project.

INTRODUCTION

- ## Housing Problem

In making of the project, there I realized that the nowadays to have the houses are very necessary need of each and every person around the globe. And those who are not have the houses it become the housing problem to the real world.

- ## Conceptual Background of the Domain Problem

It is a very large market housing problem. Many major contributors contribute in this project in the world’s economy. There are various companies working in the domain. As it is saying in the problem-statement that data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, importing 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. By doing some workout in the project there we can help others by model training and predicting things makes the previous background to work efficiently.

• Review of Project

As it is saying the company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. From this, there are some variables need to be predict are Sale Price, Sale Condition, Type of sale, Monthly Sold, Year Sold, Fence quality, Value of miscellaneous feature. The project is all about housing project that have to be predict and trained the data according to its required in the strategy.

• Motivation for the Problem Undertaken

Here, requirements of technically solved the problem in the two datasets having 1460 entries each having 81 variables, null values, contains numerical and categorical variable, extensive EDA has to be performed to gain relationships of important variable and price, to build Machine Learning models, apply regularization and determine the optimal values of Hyper Parameters. To modelling the price of houses with the available independent variables. It can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be good way for the management to understand the pricing dynamics of a new market.

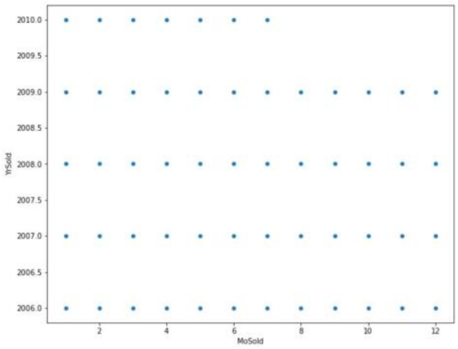
ANALYTICAL PROBLEM FRAMING

- Mathematical/Analytical Modeling of the Problem

Mathematical Modelling and Machine Learning Methods used in the Machine Learning that it is a modern advanced technology, which leads the anomalies of the machine in ground level. In this some mathematical/analytical problems are used. They are: data frame types has the length of 81 and data type is object, by using the data visualization subplot of LotFrontage, MSZoning, MSSubClass, YearBuilt, MiscVal, MoSold, MiscFeature, Fence, YrSold, SaleType, from distribution plot Monthly Sold, Yearly Sold and many more in the plot, from boxplot there are two outliers came in LotArea and SalePrice so here have to remove the outliers for ahead requirements, from scatterplot there are some target area where we have to focus on while we are doing some work on housing project:

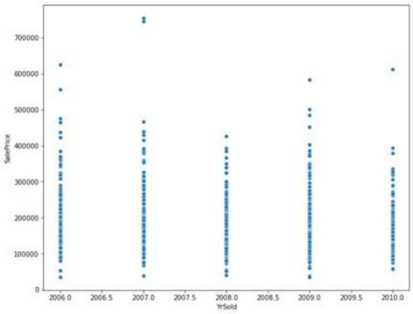
1. About Monthly sold and Yearly sold of house

```
          MoSold    YrSold
MoSold  1.000000 -0.144536
YrSold -0.144536  1.000000
```



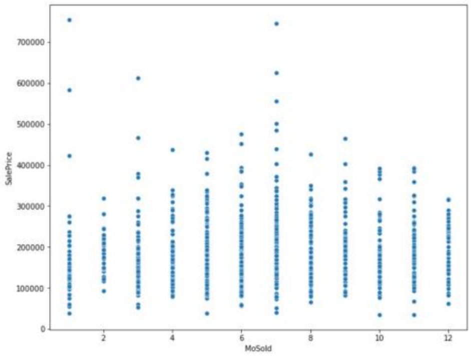
2. About Sale Price for the Yearly sold of the house

```
             YrSold  SalePrice
YrSold     1.000000  -0.045508
SalePrice -0.045508   1.000000
```



3. About Sale Price for monthly sold of the house

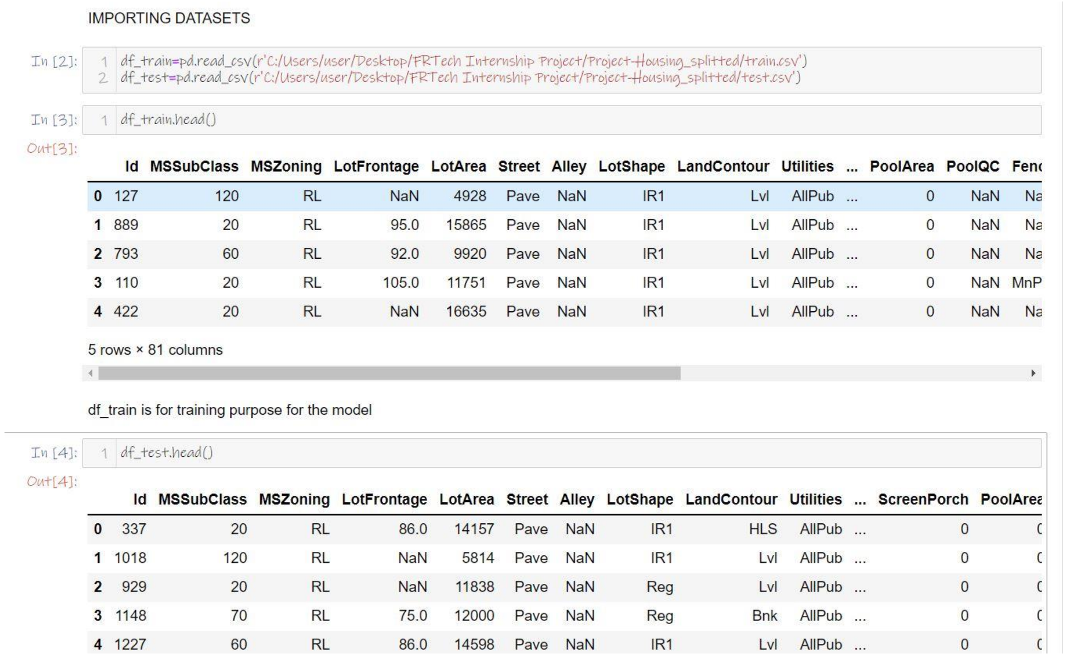
```
             MoSold  SalePrice
MoSold     1.000000   0.072764
SalePrice  0.072764   1.000000
```



From countplot of target columns: SalePrice, YrSold, MoSold, SaleType, YearBuilt, from boxplot, correlation heatmap among numeric attributes, Parse the data by using cyclic function from scatterplot, lineplot, relplot for target columns.

- Data Sources and their formats

The data sources and their formats are from .csv file (Comma-separated values)



IMPORTING DATASETS

In [2]:
```
df_train=pd.read_csv(r'C:/Users/user/Desktop/FRTech Internship Project/Project-Housing_splitted/train.csv')
df_test=pd.read_csv(r'C:/Users/user/Desktop/FRTech Internship Project/Project-Housing_splitted/test.csv')
```

In [3]:
```
df_train.head()
```

Out[3]:

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

5 rows × 81 columns

df_train is for training purpose for the model

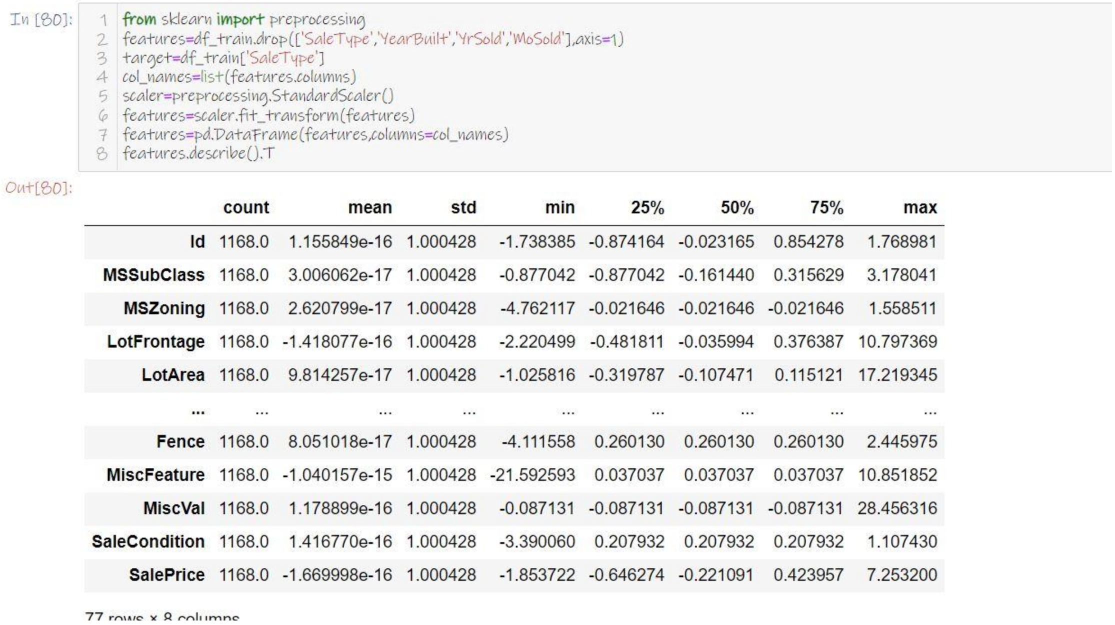
In [4]:
```
df_test.head()
```

Out[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	ScreenPorch	PoolAre
0	337	20	RL	86.0	14157	Pave	NaN	IR1	HLS	AllPub	...	0	0
1	1018	120	RL	NaN	5814	Pave	NaN	IR1	Lvl	AllPub	...	0	0
2	929	20	RL	NaN	11838	Pave	NaN	Reg	Lvl	AllPub	...	0	0
3	1148	70	RL	75.0	12000	Pave	NaN	Reg	Bnk	AllPub	...	0	0
4	1227	60	RL	86.0	14598	Pave	NaN	IR1	Lvl	AllPub	...	0	0

- ## Data Preprocessing Done

  The steps followed for the cleaning of the data is Label Encoder or Encoding after that dropping the attributes of data that target and extra columns show, then from importing preprocessing there transform the target columns into features then lastly it has to set/fit for the data frame.



In [80]:
```
from sklearn import preprocessing
features=df_train.drop(['SaleType','YearBuilt','YrSold','MoSold'],axis=1)
target=df_train['SaleType']
col_names=list(features.columns)
scaler=preprocessing.StandardScaler()
features=scaler.fit_transform(features)
features=pd.DataFrame(features,columns=col_names)
features.describe().T
```

Out[80]:

	count	mean	std	min	25%	50%	75%	max
Id	1168.0	1.155849e-16	1.000428	-1.738385	-0.874164	-0.023165	0.854278	1.768981
MSSubClass	1168.0	3.006062e-17	1.000428	-0.877042	-0.877042	-0.161440	0.315629	3.178041
MSZoning	1168.0	2.620799e-17	1.000428	-4.762117	-0.021646	-0.021646	-0.021646	1.558511
LotFrontage	1168.0	-1.418077e-16	1.000428	-2.220499	-0.481811	-0.035994	0.376387	10.797369
LotArea	1168.0	9.814257e-17	1.000428	-1.025816	-0.319787	-0.107471	0.115121	17.219345
...	...	...	...	...	...	...	...	...
Fence	1168.0	8.051018e-17	1.000428	-4.111558	0.260130	0.260130	0.260130	2.445975
MiscFeature	1168.0	-1.040157e-15	1.000428	-21.592593	0.037037	0.037037	0.037037	10.851852
MiscVal	1168.0	1.178899e-16	1.000428	-0.087131	-0.087131	-0.087131	-0.087131	28.456316
SaleCondition	1168.0	1.416770e-16	1.000428	-3.390060	0.207932	0.207932	0.207932	1.107430
SalePrice	1168.0	-1.669998e-16	1.000428	-1.853722	-0.646274	-0.221091	0.423957	7.253200

77 rows × 8 columns

- ## Data Inputs-Logic-Output Relationships

  The relationships between inputs and outputs can be studied extracting weights of the trained model. Regression is that relationships between them can be blocky or highly structured based on the training data. It requires the data scientist to train the algorithm with both labeled inputs and desired outputs.

- State the set of assumptions(if any) related to the problem under consideration

  Presumptions are by using regression label encoding, data scaling, that it means the relationship between the dependent and independent variables look fairly linear. Thus, our linearity assumption is satisfied.

- Hardware and Software Requirements and Tools Used

  By importing many important libraries are pandas, numpy, seaborn, matplotlib.pyplot linear regression, pickle, train_test_split, preprocessing, from sklearn.utlis import resample, label encoder enable_iterative_imputer, iterative_imputer, selectkbest, randomforestclassifier, mean_squared_error, mean_absolute_error, ridge, lasso, ridgecv, lassocv, qqplot, gradientboostingclassifier, standardscaler, confusion_matrix, classification_report, accuracy_score, gridsearchcv, kneighborsclassifier, logisticregression.

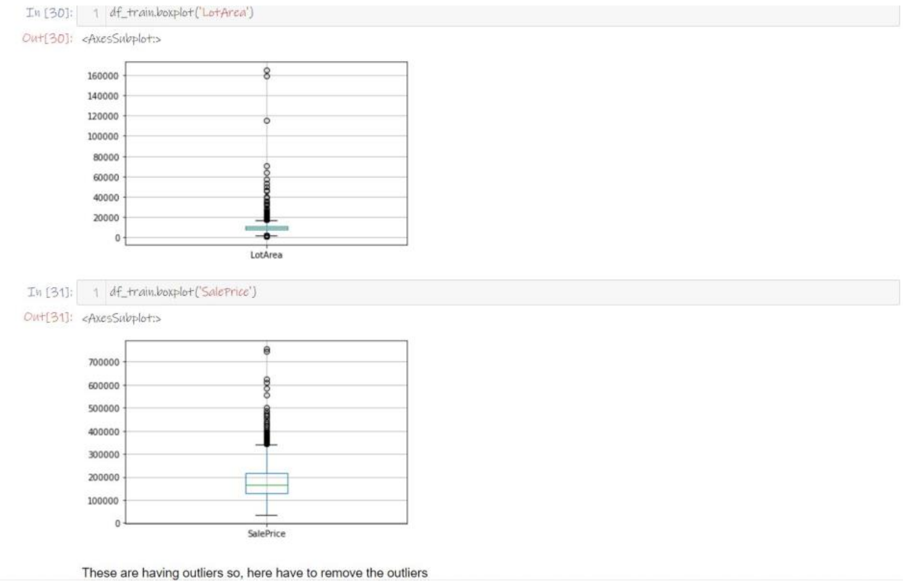
Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

The collection and interpretation of data in order to uncover patterns and trends. It is a component of data analytics. Statistical analysis can be used in situations like gathering research interpretations, statistical modelling or designing surveys and studies. The approaches/methods of identification are descriptive and inferential statistics which are describes as the properties of sample and population data, and inferential statistics which uses those properties to test hypotheses and draw efficient conclusions in terms of outputs.

- Testing of Identified Approaches (Algorithms)

OUTLIERS:

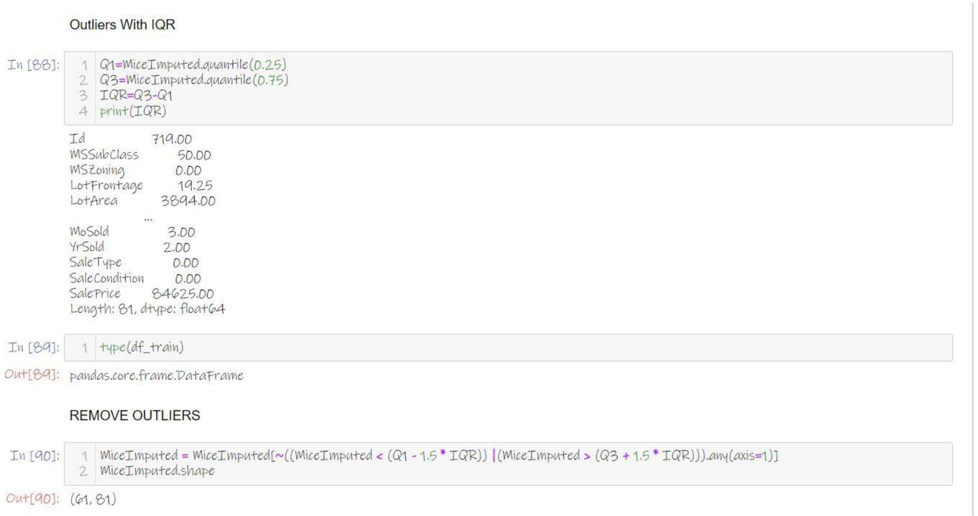


```
In [30]: df_train.boxplot('LotArea')
Out[30]: <AxesSubplot:>
```

```
In [31]: df_train.boxplot('SalePrice')
Out[31]: <AxesSubplot:>
```

These are having outliers so, here have to remove the outliers

REMOVE THE OUTLIERS:



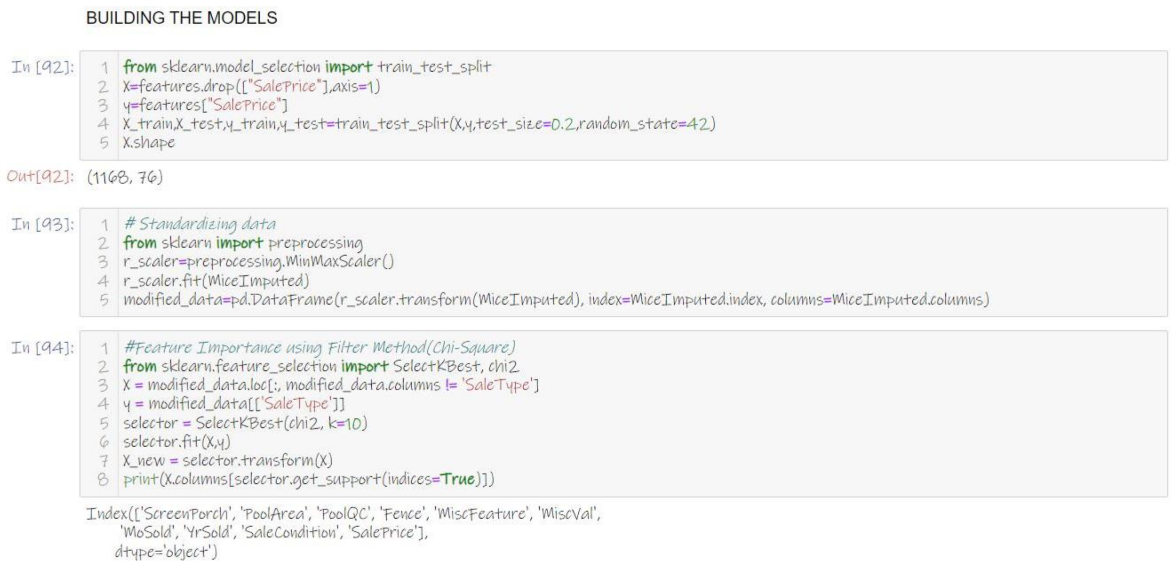
Outliers With IQR

```
In [88]: Q1=MiceImputed.quantile(0.25)
         Q3=MiceImputed.quantile(0.75)
         IQR=Q3-Q1
         print(IQR)

Id               719.00
MSSubClass        50.00
MSZoning           0.00
LotFrontage       19.25
LotArea         3894.00
                  ...
MoSold             3.00
YrSold             2.00
SaleType           0.00
SaleCondition      0.00
SalePrice      84625.00
Length: 81, dtype: float64

In [89]: type(df_train)
Out[89]: pandas.core.frame.DataFrame
```

REMOVE OUTLIERS

```
In [90]: MiceImputed = MiceImputed[~((MiceImputed < (Q1 - 1.5 * IQR)) |(MiceImputed > (Q3 + 1.5 * IQR))).any(axis=1)]
         MiceImputed.shape
Out[90]: (61, 81)
```

BULIDING THE MODELS:

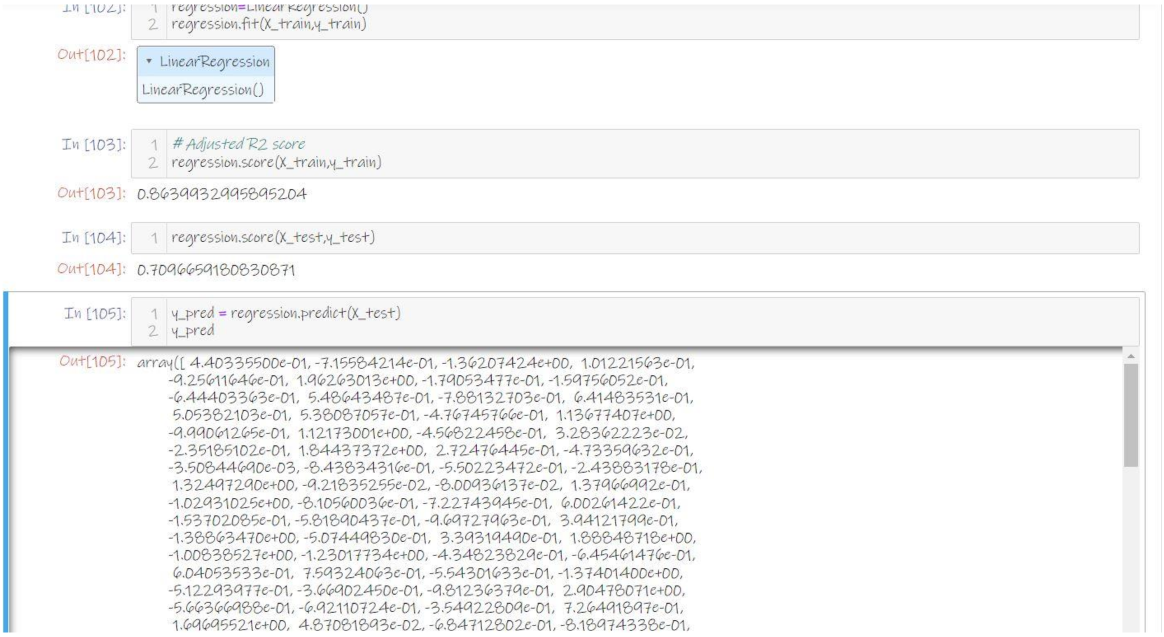


BUILDING THE MODELS

```
In [92]:
from sklearn.model_selection import train_test_split
X=features.drop(["SalePrice"],axis=1)
y=features["SalePrice"]
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
X.shape

Out[92]: (1168, 76)

In [93]:
# Standardizing data
from sklearn import preprocessing
r_scaler=preprocessing.MinMaxScaler()
r_scaler.fit(MiceImputed)
modified_data=pd.DataFrame(r_scaler.transform(MiceImputed), index=MiceImputed.index, columns=MiceImputed.columns)

In [94]:
#Feature Importance using Filter Method(Chi-Square)
from sklearn.feature_selection import SelectKBest, chi2
X = modified_data.loc[:, modified_data.columns != 'SaleType']
y = modified_data[['SaleType']]
selector = SelectKBest(chi2, k=10)
selector.fit(X,y)
X_new = selector.transform(X)
print(X.columns[selector.get_support(indices=True)])

Index(['ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVal',
       'MoSold', 'YrSold', 'SaleCondition', 'SalePrice'],
      dtype='object')
```

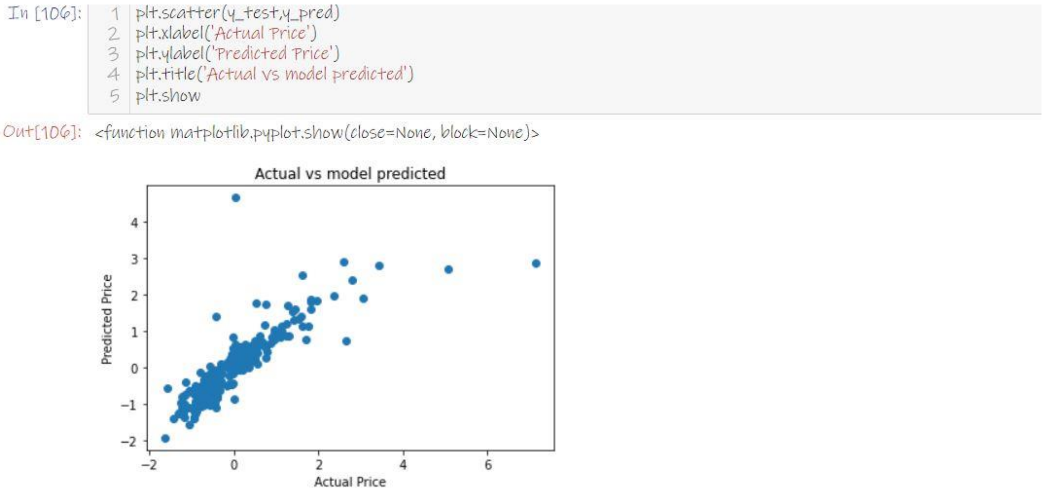
- Run and Evaluate selected models

LINEAR REGRESSION:



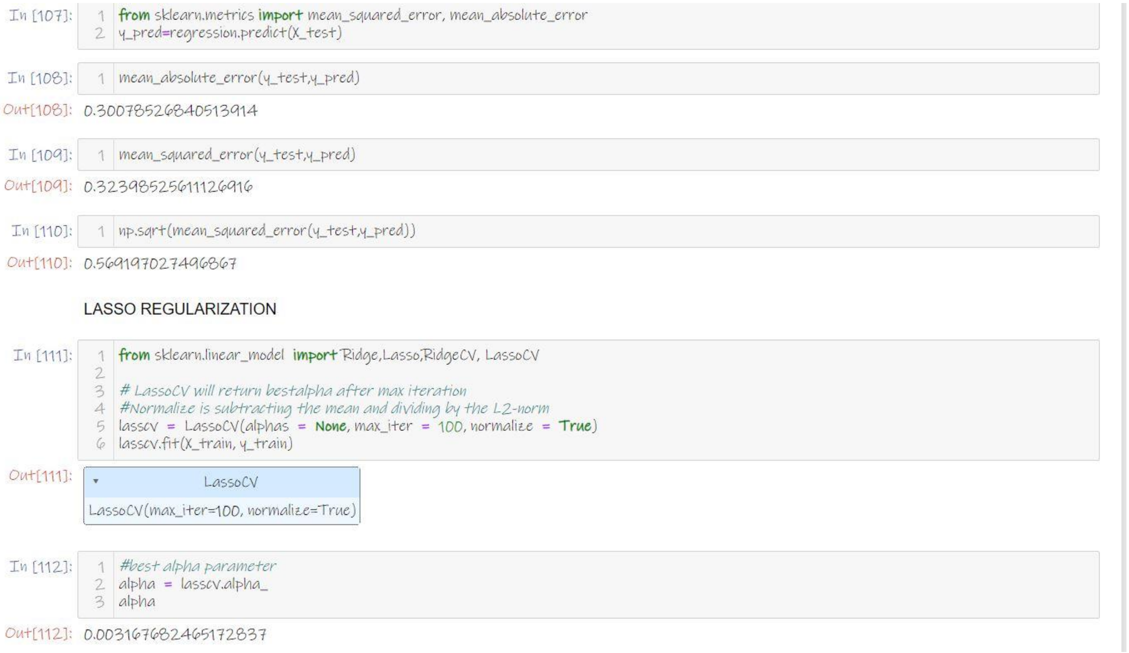
```
In [102]:
regression=LinearRegression()
regression.fit(X_train,y_train)

Out[102]:
▾ LinearRegression
LinearRegression()

In [103]:
# Adjusted R2 score
regression.score(X_train,y_train)

Out[103]: 0.8639932995895204

In [104]:
regression.score(X_test,y_test)

Out[104]: 0.7096659180830871

In [105]:
y_pred = regression.predict(X_test)
y_pred

Out[105]: array([ 4.40335500e-01, -7.15584214e-01, -1.36207424e+00,  1.01221563e-01,
       -9.25611646e-01,  1.96263013e+00, -1.79053477e-01, -1.59756052e-01,
       -6.44403363e-01,  5.48643487e-01, -7.88132703e-01,  6.41483531e-01,
        5.05382103e-01,  5.38087057e-01, -4.76745766e-01,  1.13677407e+00,
       -9.99061265e-01,  1.12173001e+00, -4.56822458e-01,  3.28362223e-02,
       -2.35185102e-01,  1.84437372e+00,  2.72476445e-01, -4.73359632e-01,
       -3.50844690e-03, -8.43834316e-01, -5.50223472e-01, -2.43883178e-01,
        1.32497290e+00, -9.21835255e-02, -8.00936137e-02,  1.37966992e-01,
       -1.02931025e+00, -8.10560036e-01, -7.22743945e-01,  6.00261422e-01,
       -1.53702085e-01, -5.81890437e-01, -9.69727963e-01,  3.94121799e-01,
       -1.38863470e+00, -5.07449830e-01,  3.39319490e-01,  1.88848718e+00,
       -1.00838527e+00, -1.23017734e+00, -4.34823829e-01, -6.45461476e-01,
        6.04053533e-01,  7.59324063e-01, -5.54301633e-01, -1.37401400e+00,
       -5.12293977e-01, -3.66902450e-01, -9.81236379e-01,  2.90478071e+00,
       -5.66366988e-01, -6.92110724e-01, -3.54922809e-01,  7.26491897e-01,
        1.69695521e+00,  4.87081893e-02, -6.84712802e-01, -8.18974338e-01,
```

ITS ACTUAL PRICE:



```
In [106]: plt.scatter(y_test,y_pred)
          plt.xlabel('Actual Price')
          plt.ylabel('Predicted Price')
          plt.title('Actual vs model predicted')
          plt.show

Out[106]: <function matplotlib.pyplot.show(close=None, block=None)>
```

MODEL EVALUATION:

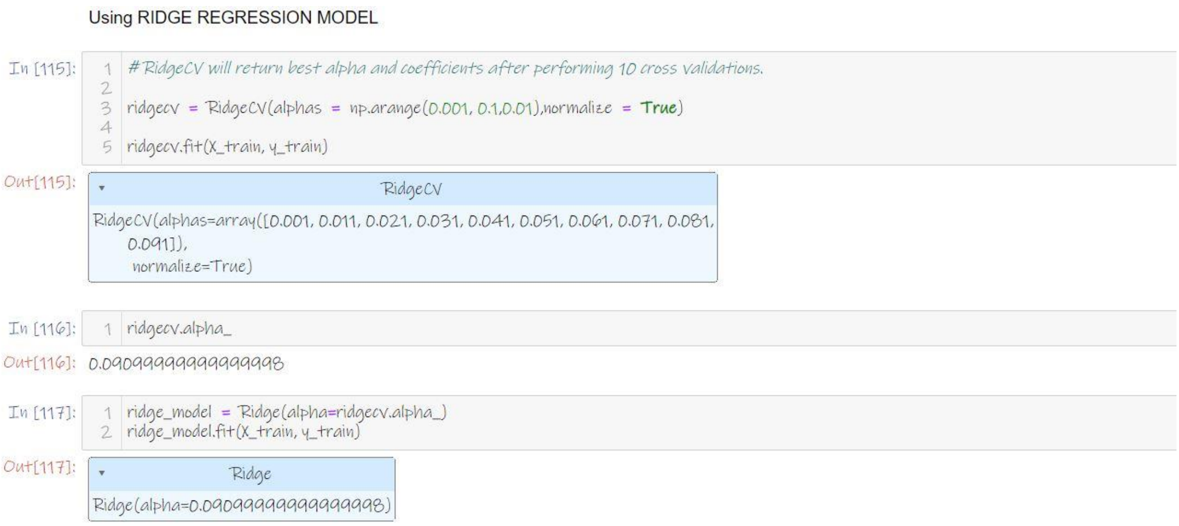


```
In [107]: from sklearn.metrics import mean_squared_error, mean_absolute_error
          y_pred=regression.predict(X_test)

In [108]: mean_absolute_error(y_test,y_pred)

Out[108]: 0.30078526840513914

In [109]: mean_squared_error(y_test,y_pred)

Out[109]: 0.32398525611126916

In [110]: np.sqrt(mean_squared_error(y_test,y_pred))

Out[110]: 0.569197027496867
```

LASSO REGULARIZATION

```
In [111]: from sklearn.linear_model  import Ridge,Lasso,RidgeCV, LassoCV

          # LassoCV will return bestalpha after max iteration
          #Normalize is subtracting the mean and dividing by the L2-norm
          lasscv = LassoCV(alphas = None, max_iter = 100, normalize = True)
          lasscv.fit(X_train, y_train)

Out[111]: ▾ LassoCV
          LassoCV(max_iter=100, normalize=True)

In [112]: #best alpha parameter
          alpha = lasscv.alpha_
          alpha

Out[112]: 0.0031676824651728 37
```

RIDGE REGRESSION MODEL:



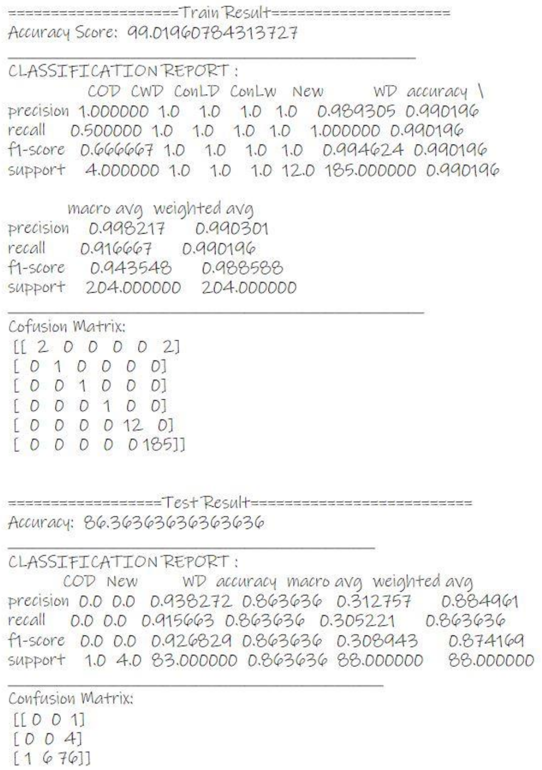
Using RIDGE REGRESSION MODEL

```
In [115]:
#RidgeCV will return best alpha and coefficients after performing 10 cross validations.

ridgecv = RidgeCV(alphas = np.arange(0.001, 0.1,0.01),normalize = True)

ridgecv.fit(X_train, y_train)

Out[115]:
RidgeCV
RidgeCV(alphas=array([0.001, 0.011, 0.021, 0.031, 0.041, 0.051, 0.061, 0.071, 0.081,
       0.091]),
        normalize=True)

In [116]:
ridgecv.alpha_

Out[116]: 0.09099999999999998

In [117]:
ridge_model = Ridge(alpha=ridgecv.alpha_)
ridge_model.fit(X_train, y_train)

Out[117]:
Ridge
Ridge(alpha=0.09099999999999998)
```

Ridge regression are the same as that of linear regression: linearity, constant variance, and independence. However, as ridge regression does not provide confidence limits, the distribution of errors to be normal need not be assumed. When the final regression coefficients are displayed, they are adjusted back into their original scale. However, the ridge trace is on a standardized scale.

GRADIENT BOOSTING CLASSIFIER: TRAIN AND TEST RESULT

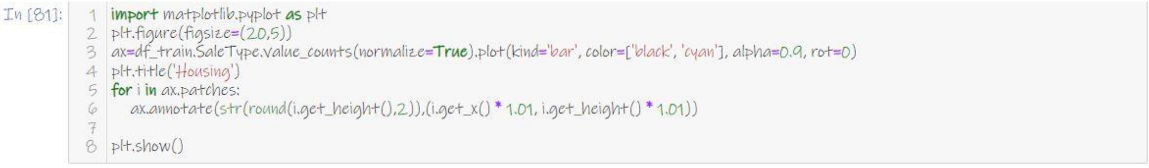
ACCUARCY SCORE 86%



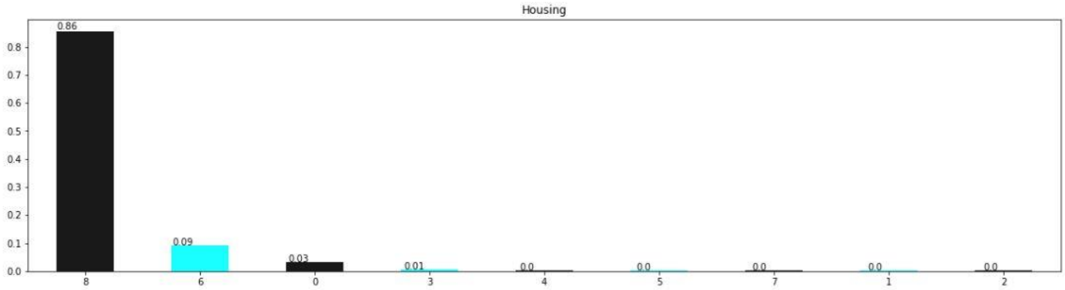
```
====================Train Result=====================
Accuracy Score:  99.01960784313727
_______________________________________________
CLASSIFICATION REPORT :
             COD  CWD  ConLD  ConLw   New        WD  accuracy  \
precision  1.000000  1.0   1.0   1.0   1.0   0.989305  0.990196
recall     0.500000  1.0   1.0   1.0   1.0   1.000000  0.990196
f1-score   0.666667  1.0   1.0   1.0   1.0   0.994624  0.990196
support    4.000000  1.0   1.0   1.0  12.0  185.000000  0.990196

          macro avg   weighted avg
precision    0.998217     0.990301
recall       0.916667     0.990196
f1-score     0.943548     0.988588
support    204.000000   204.000000
_______________________________________________
Cofusion Matrix:
 [[  2   0   0   0   0   2]
 [  0   1   0   0   0   0]
 [  0   0   1   0   0   0]
 [  0   0   0   1   0   0]
 [  0   0   0   0  12   0]
 [  0   0   0   0   0 185]]


==================Test Result==========================
Accuracy:  86.36363636363636
_______________________________________________
CLASSIFICATION REPORT :
          COD  New        WD  accuracy  macro avg  weighted avg
precision  0.0  0.0   0.938272  0.863636   0.312757      0.884961
recall     0.0  0.0   0.915663  0.863636   0.305221      0.863636
f1-score   0.0  0.0   0.926829  0.863636   0.308943      0.874169
support    1.0  4.0  83.000000  0.863636  88.000000     88.000000
_______________________________________________
Confusion Matrix:
 [[ 0  0  1]
 [ 0  0  4]
 [ 1  6 76]]
```

- Visualizations

SALE TYPE HOUSING:

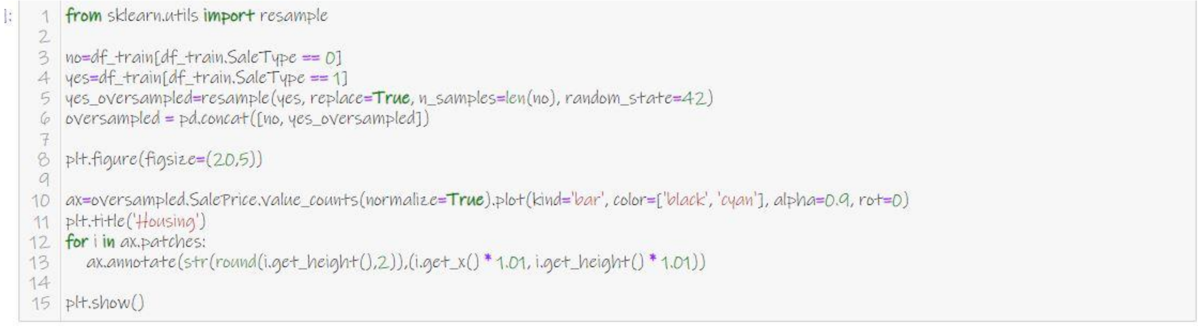


```
In [81]:
import matplotlib.pyplot as plt
plt.figure(figsize=(20,5))
ax=df_train.SaleType.value_counts(normalize=True).plot(kind='bar', color=['black', 'cyan'], alpha=0.9, rot=0)
plt.title('Housing')
for i in ax.patches:
    ax.annotate(str(round(i.get_height(),2)),(i.get_x() * 1.01, i.get_height() * 1.01))

plt.show()
```

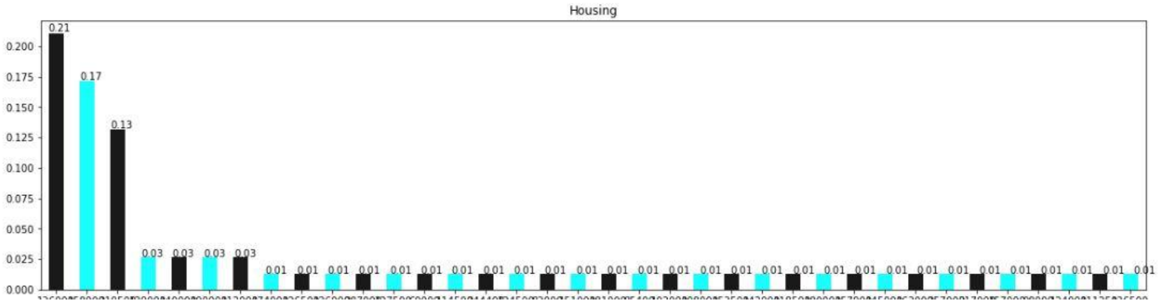


By using the matplotlib the target column is sale type which shows with the higher accuracy of 0.86 among them and it will titled as Housing.

RESAMPLE THE HOUSING SALE TYPE:

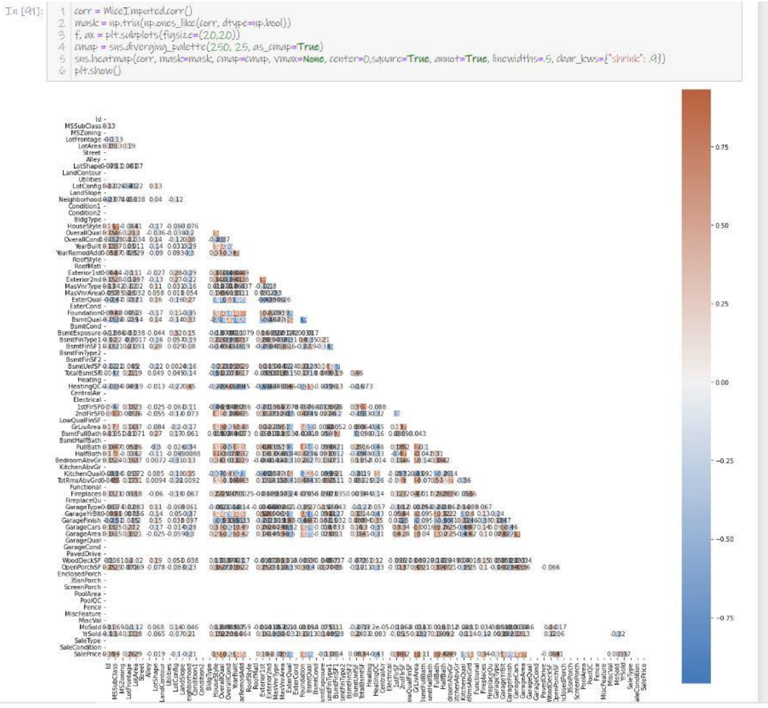


```
from sklearn.utils import resample

no=df_train[df_train.SaleType == 0]
yes=df_train[df_train.SaleType == 1]
yes_oversampled=resample(yes, replace=True, n_samples=len(no), random_state=42)
oversampled = pd.concat([no, yes_oversampled])

plt.figure(figsize=(20,5))

ax=oversampled.SalePrice.value_counts(normalize=True).plot(kind='bar', color=['black', 'cyan'], alpha=0.9, rot=0)
plt.title('Housing')
for i in ax.patches:
    ax.annotate(str(round(i.get_height(),2)),(i.get_x() * 1.01, i.get_height() * 1.01))

plt.show()
```



In this target column is sale type that it increases with 0.21 accuracy and it is oversampled the sale price by importing resample.

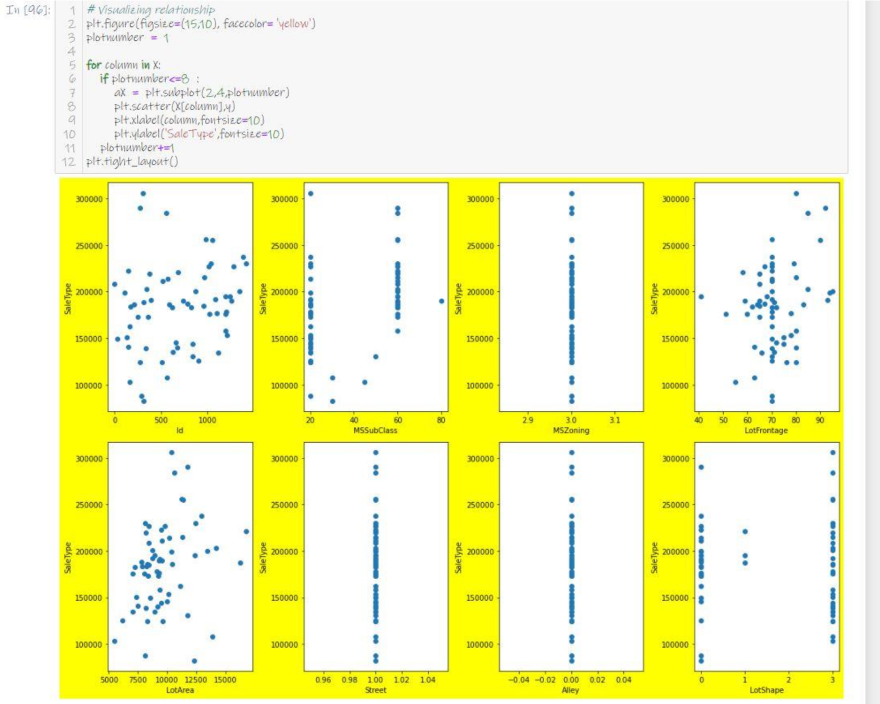
AFTER REMOVING THE OUTLIERS THE CORRELATION:



```
In [91]:
corr = MiceImputed.corr()
mask = np.triu(np.ones_like(corr, dtype=np.bool))
f, ax = plt.subplots(figsize=(20,20))
cmap = sns.diverging_palette(250, 25, as_cmap=True)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=None, center=0,square=True, annot=True, linewidths=.5, cbar_kws={"shrink": .9})
plt.show()
```

By using the mice imputed correlation this will be in the diverging from one side and it means it removes the outlier (np.ones) all the columns shows their correlation among them.

VISUALIZING THE RELATIONSHIP BETWEEN ALLTHE COLUMN AS SHOWN BELOW:



```
In [96]:
# Visualizing relationship
plt.figure(figsize=(15,10), facecolor= 'yellow')
plotnumber  = 1

for column in X:
   if plotnumber<=8  :
      aX  =  plt.subplot(2,4,plotnumber)
      plt.scatter(X[column],y)
      plt.xlabel(column,fontsize=10)
      plt.ylabel('SaleType',fontsize=10)
   plotnumber+=1
plt.tight_layout()
```

The relationship between the dependent and independent variables look fairly linear. Thus, our linearity assumption is satisfied.

- Interpretation of the Results

The results were interpreted from the visualizations, preprocessing and modelling:

1. The model of the house price with the available independent variables and exactly the prices are vary with the variables.
2. It can accordingly manipulating the strategy of the areas that will yield high returns as it make easier for the firm whoever take this opportunity in the housing project.
3. It will take a good way for the management to understand the pricing dynamics of a new project.
4. By doing visualization there are many things to be noted when it will according to work each other it means that from which aspect it is going to work either from sale type or condition as part from the sale price because it the main target of the housing case because according to this it will predicted whether it will rise in monthly or yearly.
5. By preprocessing the data it means that from the housing case label encoder helps the dataset column to transform to fit another column in to it.

CONCLUSION

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.

I would like to conclude here that this case is good way of predicting the data and its learning outcomes of the study in respect of Data Science from this it is very helpful in using visualization, data cleaning and various algorithms. By doing visualization there are many things to be noted when it will according to work each other it means that from which aspect it is going to work either from sale type or condition as part from the sale price because it the main target of the housing case because according to this it will predicted whether it will rise in monthly or yearly. By preprocessing the data it means that from the housing case label encoder helps the dataset column to transform to fit another column in to it. The model of the house price with the available independent variables and exactly the prices are vary with the variables.

THANK YOU