HOUSE SALE PRICE PRIDICTION

SUMMITED BY

NEETAL TIWARI

INTRODUCTION

Nowadays so many buildings are being built. It becomes difficult to decide the sale price of the house, so with the help of all the features the sale price of the house is decided by using the machine learning algorithms.

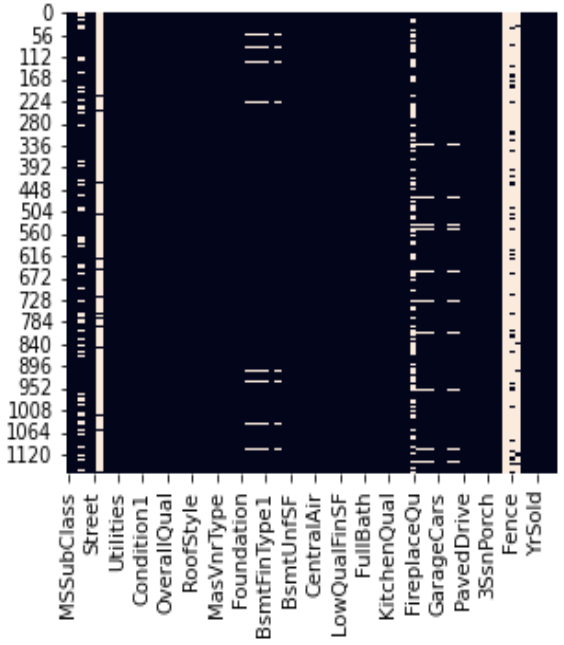
```
[3]: hp=pd.read_csv("housing.tsv",sep="\t")
[4]: hp
```

Out[4]:

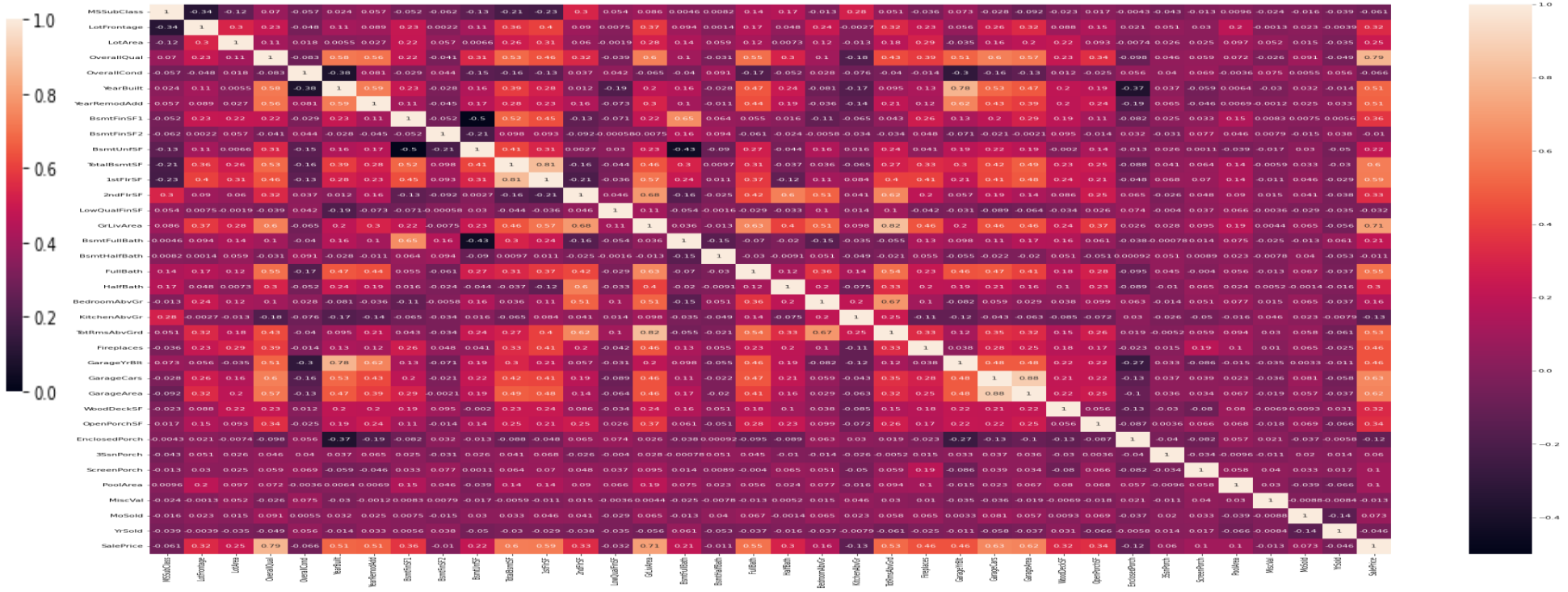
	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
2	793	60	RL	92.0	9920	Pave	NaN	IR1	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1163	289	20	RL	NaN	9819	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1164	554	20	RL	67.0	8777	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1165	196	160	RL	24.0	2280	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1166	31	70	C (all)	50.0	8500	Pave	Pave	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1167	617	60	RL	NaN	7861	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

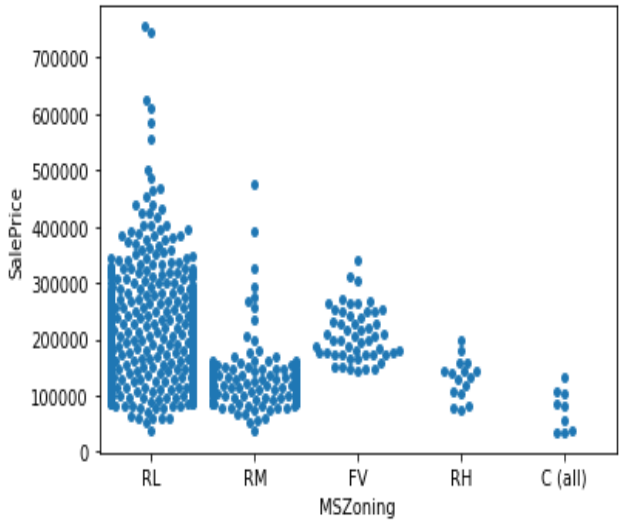


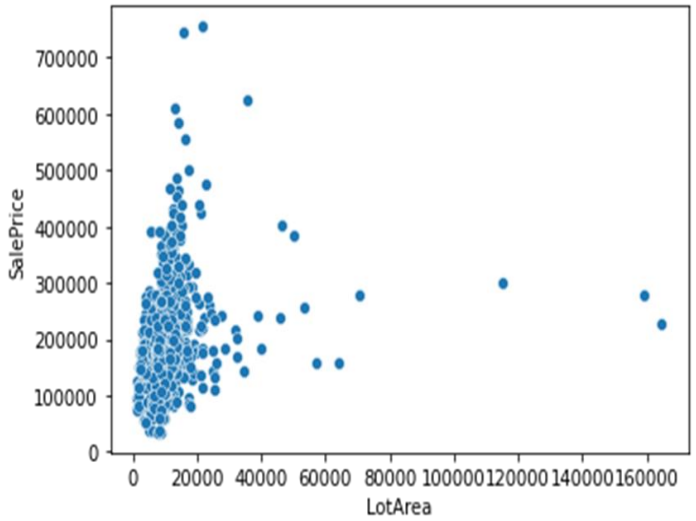
5558 missing values in housing Dataset.

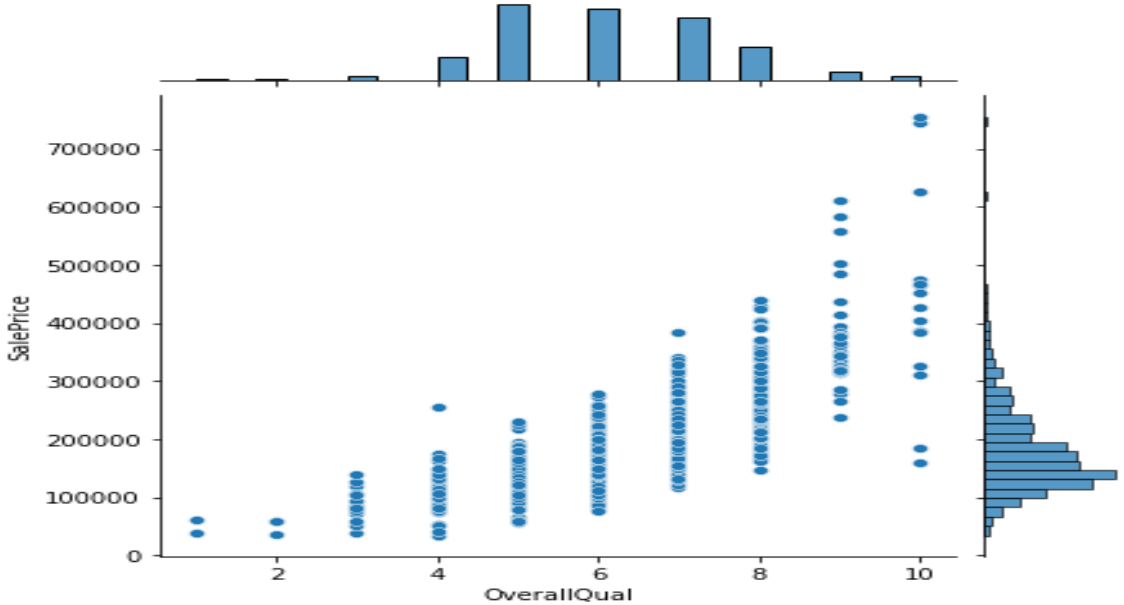


OverallCond(Rates the overall condition of the house) and EnclosedPorch(Enclosed porch area in square feet) are very less correlated and GrLivArea(Above grade (ground) living area square feet) and SalePrice are high correlated.

Data Visualizations or EDA





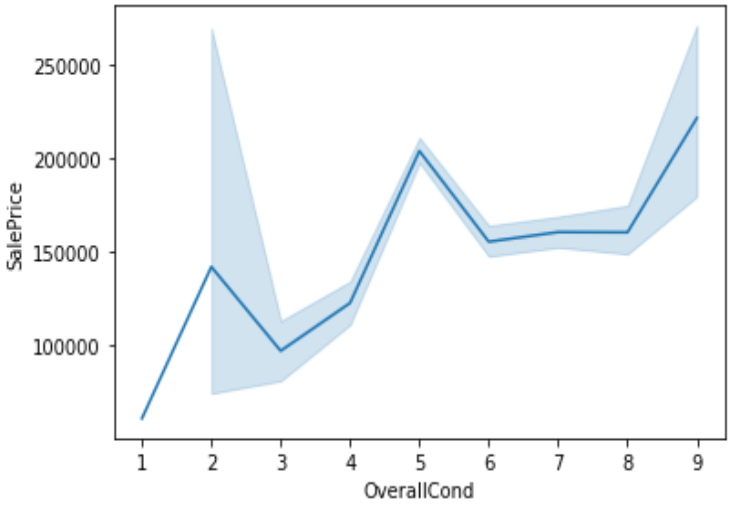


Sale Price is increasing with Lot Area (Lot size in square feet).

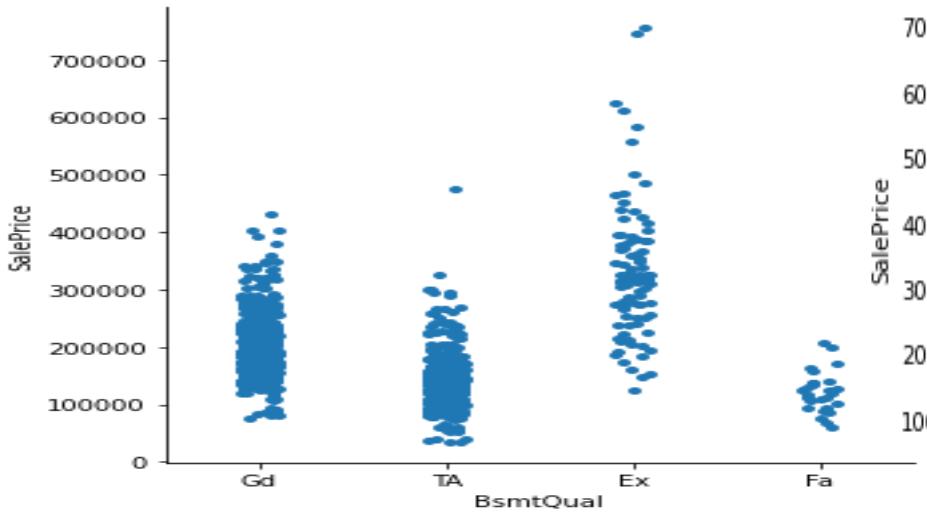
Sale Price of RL (Residential Low Density) is highest.

Sale Price is increasing With the OverallQual (Rates the overall material and finish of the house).

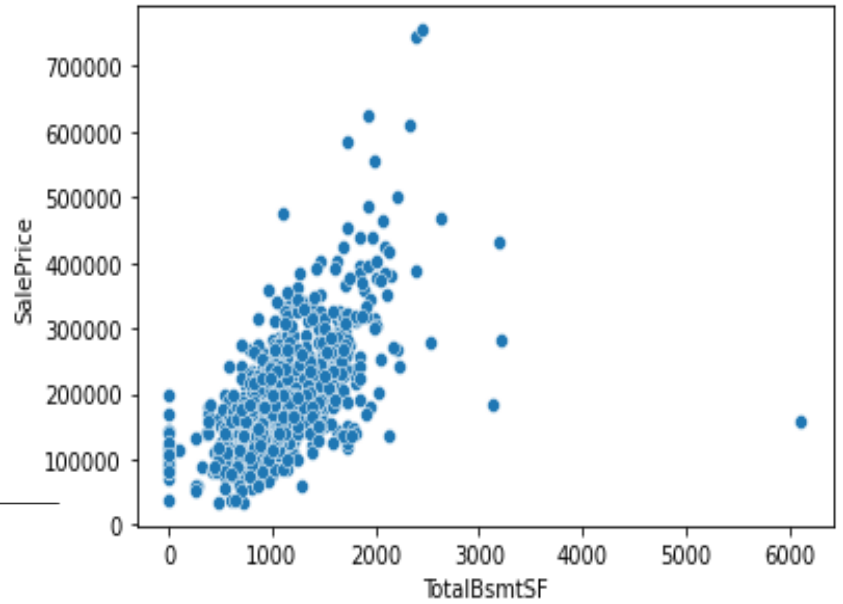
Data Visualizations or EDA



Sale Price is increasing with the OverallCond (Rates the overall condition of the house).



Sale Price is higher Ex (Excellent (100+ inches)) type of BsmtQual (Evaluates the height of the basement).



Sale Price is increasing with the TotalBsmtSF (Total square feet of basement area).

Data Pre-processing Done

In Data pre-processing I use Label Encoding method to encode the objects to int, there are 38 objects in housing dataset.
Removed the skewness of the data by using power transform, Power Transformer, most of the independents features such as poolarea, 3Ssnporch, lowQualFinSF are high skewed.
Then remove the outliers by using zscore method. In 680 rows outliers were present.
After that I scaled the Data by using StandardScaler method.

```
In [330]: le=LabelEncoder()
          objects=["MSZoning","Street","LotShape","LandContour","Utilities","LotConfig","LandSlope","Neighborhood","Condition1","Conditio
                  "BldgType","HouseStyle","RoofStyle","RoofMatl","Exterior1st","Exterior2nd","MasVnrType","ExterQual","BsmtCond",
                  "BsmtExposure","BsmtFinType1","BsmtFinType2","Heating","HeatingQC","CentralAir","Electrical","KitchenQual","Functional"
                  "GarageFinish","GarageQual","GarageCond","PavedDrive","SaleType","SaleCondition","Foundation","BsmtQual","ExterCond"]
          for i in objects:
              hp[i]=le.fit_transform(hp[i])

          objects

Out[330]: ['MSZoning',
           'Street',
           'LotShape',
           'LandContour',
           'Utilities',
           'LotConfig',
           'LandSlope',
           'Neighborhood',
           'Condition1',
           'Condition2',
           'BldgType',
           'HouseStyle',
           'RoofStyle',
           'RoofMatl',
           'Exterior1st',
           'Exterior2nd',
           'MasVnrType',
           'ExterQual',
           'BsmtCond',
           'BsmtExposure',
           'BsmtFinType1',
           'BsmtFinType2',
```

Removing the skewness

```
[350]: from sklearn.preprocessing import power_transform,PowerTransformer

[351]: PowerTransformer()

[351]: PowerTransformer()

[352]: x_new=power_transform(x)

[353]: x_new

[353]: array([[ 1.37043472, -0.16245555,  0.09365762, ..., -0.60480623,
                0.40906852,  0.02973497],
              [-1.16799937, -0.16245555,  1.11713521, ..., -0.60480623,
                0.40906852,  0.02973497],
              [ 0.4900471 , -0.16245555,  0.99880298, ..., -0.60480623,
                0.40906852,  0.02973497],
```

Model/s Development and Evaluation

LinearRegression

```
]: li=LinearRegression()
li.fit(x_train,y_train)
lipred=li.predict(x_test)

print('Mean absolute error:',mean_absolute_error(y_test,lipred))
print('Mean squared error:',mean_squared_error(y_test,lipred))
print('Root mean squraed Error:',np.sqrt(mean_squared_error(y_test,lipred)))
print(r2_score(y_test,lipred)*100)
```

```
Mean absolute error: 15268.645344478418
Mean squared error: 369463929.2027042
Root mean squraed Error: 19221.444513945986
89.19144303618182
```

RandomForestRegressor ¶

```
]: rf=RandomForestRegressor()
rf.fit(x_train,y_train)
rfpred=rf.predict(x_test)

print('Mean absolute error:',mean_absolute_error(y_test,rfpred))
print('Mean squared error:',mean_squared_error(y_test,rfpred))
print('Root mean squraed Error:',np.sqrt(mean_squared_error(y_test,rfpred)))
print(r2_score(y_test,rfpred)*100)
```

```
Mean absolute error: 12174.42857142857
Mean squared error: 262625027.68144077
Root mean squraed Error: 16205.709724706314
92.31698320876734
```

In LinearRegressin model r2 score was 89.19%.

In RandomforestRegressor I got 92.31% of accuracy which is very good.

Model/s Development and Evaluation

KNeighborsRegressor

```
6]: knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
knnpred=knn.predict(x_test)
print('Mean absolute error:',mean_absolute_error(y_test,knnpred))
print('Mean squared error:',mean_squared_error(y_test,knnpred))
print('Root mean squraed Error:',np.sqrt(mean_squared_error(y_test,knnpred)))
print(r2_score(y_test,knnpred)*100)
```

```
Mean absolute error: 16328.212244897957
Mean squared error: 475315694.7812245
Root mean squraed Error: 21801.736049709998
86.09478123094114
```

In KNeighbors Regressor I got 86.09 of accuracy.

DecisionTreeRegressor

```
7]: dtr=DecisionTreeRegressor()
dtr.fit(x_train,y_train)
dtrpred=dtr.predict(x_test)
print('Mean absolute error:',mean_absolute_error(y_test,dtrpred))
print('Mean squared error:',mean_squared_error(y_test,dtrpred))
print('Root mean squraed Error:',np.sqrt(mean_squared_error(y_test,dtrpred)))
print(r2_score(y_test,dtrpred)*100)
```

```
Mean absolute error: 18803.979591836734
Mean squared error: 919528250.244898
Root mean squraed Error: 30323.724214629343
73.09947551832813
```

In Decision Tree regressor I got 73.09% of accuracy.

CONCLUSION

In Housing sale price project 81 columns and 1168 were there, if we see by excel sheet it is very hard to set the price of house but by using ML it is easy to predict the sale price of house or buy price of the house as well…

It is very hard to analysis that which of the independent variable affect the house sale price more and which independent variable affect the sale price less, but by using all the visualization tools like matplotlib and seaborn we can easily see by the graph that how sale price increases and decreases.

Predict sell price is a regression problem that is why I used the all-regression algorithm,

Random forest regressor model works good for

housing project with 92.316% of accuracy, which is very good accuracy for the prediction.