HOUSE PRICES: ADVANCED REGRESSION TECHNIQUES

Parakh Singhal 18csu152
Saksham Yadav 18csu187

INTRODUCTION:

Purchasing a home remains one of the biggest purchasing decision that individuals make in their lifetime. Our project aims to predict sale prices of houses by using various aspects of residential homes. The insights gathered could be used by individuals to complement their decision making process when purchasing a house. This helps to maximise the value users can gain while keeping to a budget. Findings could also be used by a property agent to improve the probability of a sale through proper marketing of key variables. The dataset that we are working on consists of mainly categorical variables stored as integers and factors. We would be focusing on descriptive and predictive analytics, with suggestions on how additional data could be obtained to conduct more experiments to optimise the purchase. The dataset used is obtained from Kaggle.

https://www.kaggle.com/c/house-prices-advanced-regression-techniques

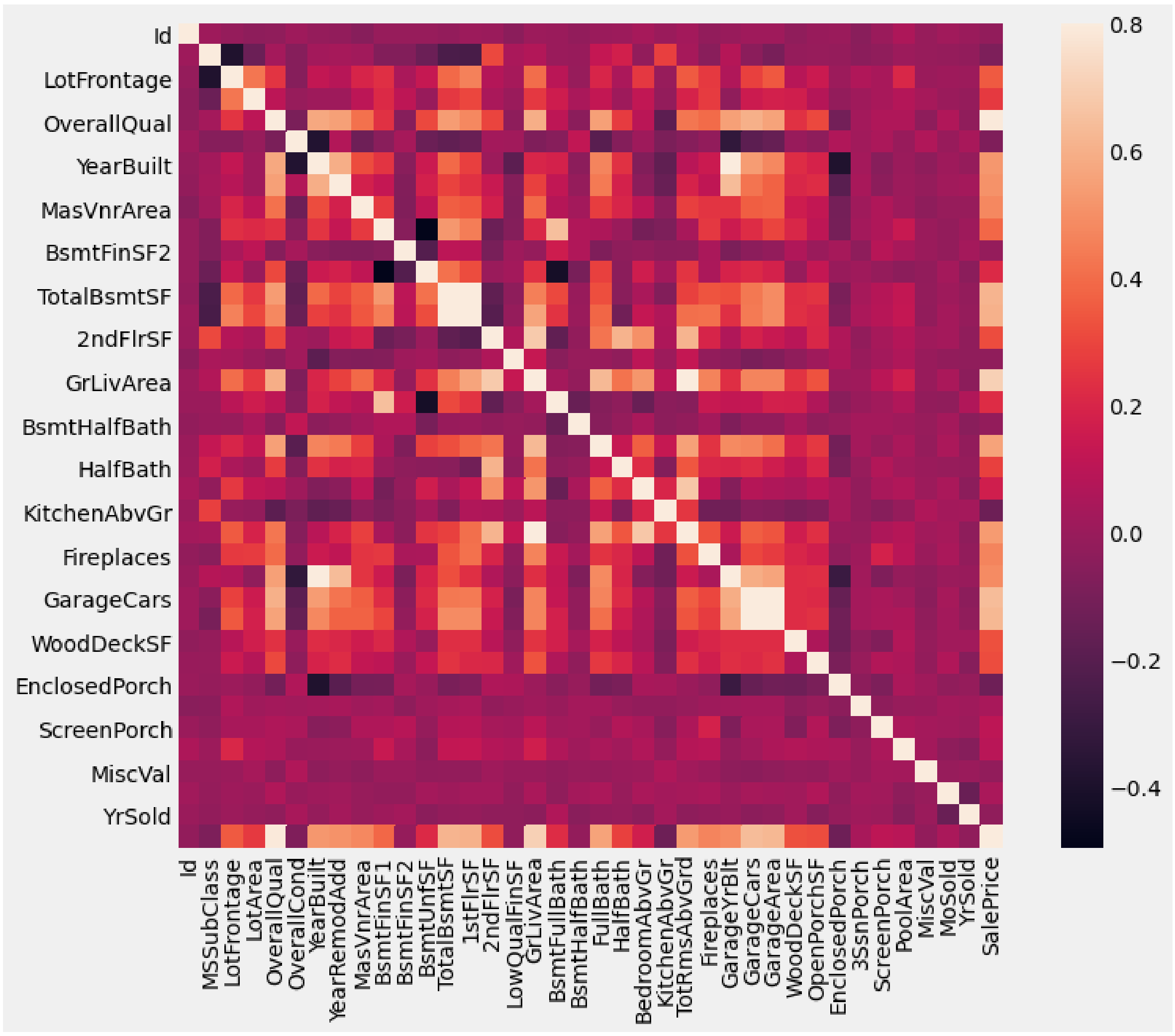
DATASET DESCRIPTION:

The training dataset has a total of 1460 observations with 81 variables while the test dataset has
one less variable. The dataset had information about the factors that affect the price of a house like the area space, garage availability ad it space, street type location, overall conditoi, year built, etc.

EXPLORATORY DATA ANALYSIS:

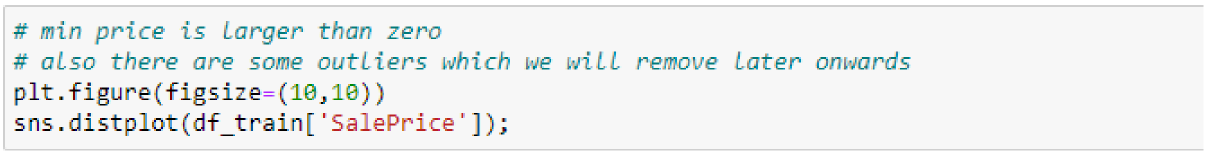
So, our main task was to find out the variables which were mostly responsible for the variation in the value of the variable SalePrice. For this we checked the correlation among the variable using the heatmap.

```
# correlation matrix
corrmat = df_train.corr()
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True);
```

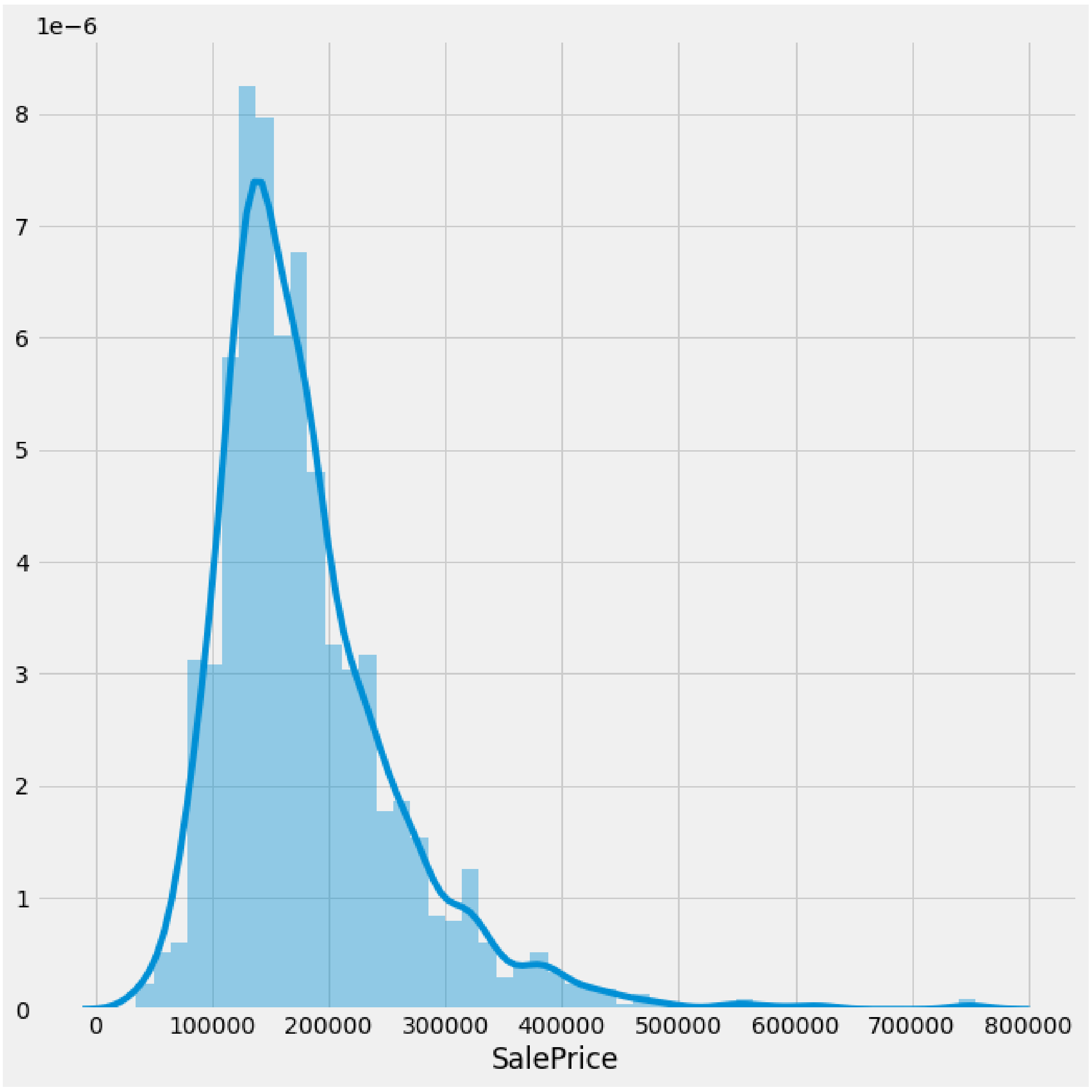


From this we got to know that the variable SalePrice has good relationship with 'OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea', 'TotalBsmtSF', '1stFlrSF', 'FullBath', 'TotRmsAbvGrd' and 'YearBuilt'

So, we chose these variables to predict the value of the target variable. And we used ‘seaborn’ to see the graphical representation of the chosen variable to see they are related with SalePrice.



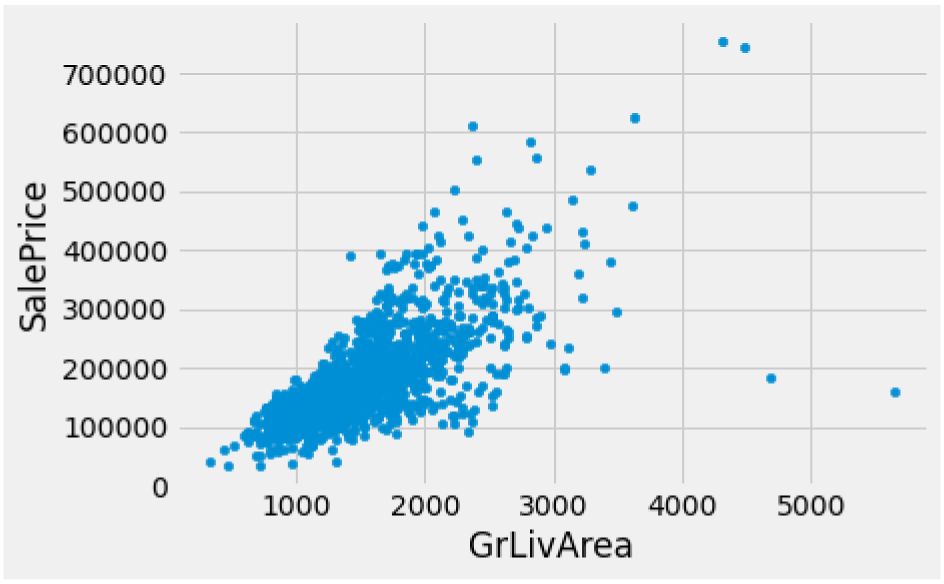
```
# min price is larger than zero
# also there are some outliers which we will remove later onwards
plt.figure(figsize=(10,10))
sns.distplot(df_train['SalePrice']);
```



From the above distplot we can see that the highest values are in the rage of 150,000 to 350,000 and also we can see that the lowest price is larger than zero. Therefore, there are some outlier on the right portion only.

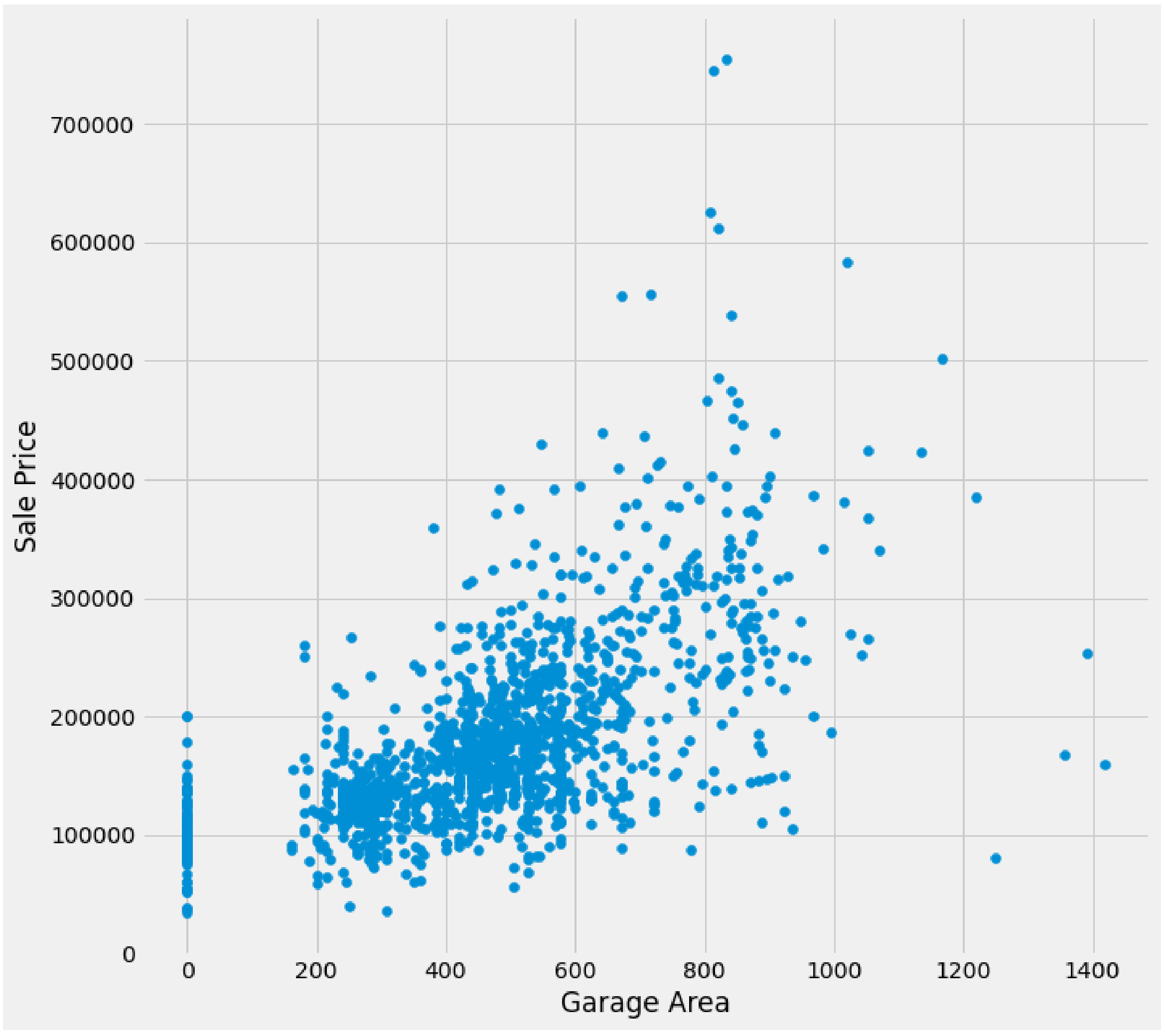
```
# scatter plot grlivarea/saleprice
plt.figure(figsize=(10,10))
plt.style.use('fivethirtyeight')
var = 'GrLivArea'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
data.plot.scatter(x=var, y='SalePrice')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a730935460>

<Figure size 720x720 with 0 Axes>
```



In this we can see that most of the houses has the above grade living area of about 1000 to 2000 sqft.

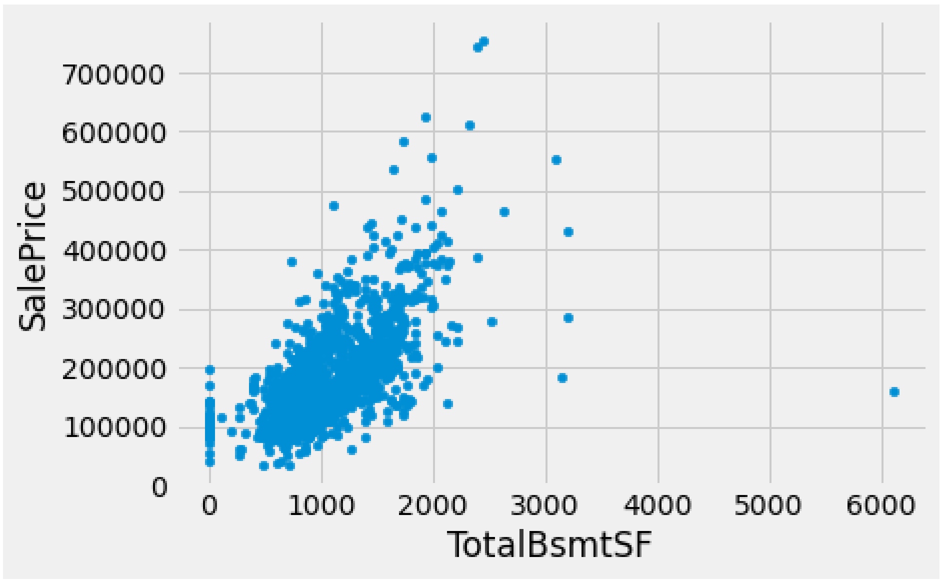
```python
# scatter plot garagearea/saleprice
plt.figure(figsize=(10,10))
plt.scatter(x=df_train['GarageArea'], y=df_train['SalePrice'])
plt.ylabel('Sale Price')
plt.xlabel('Garage Area')
plt.show()
```



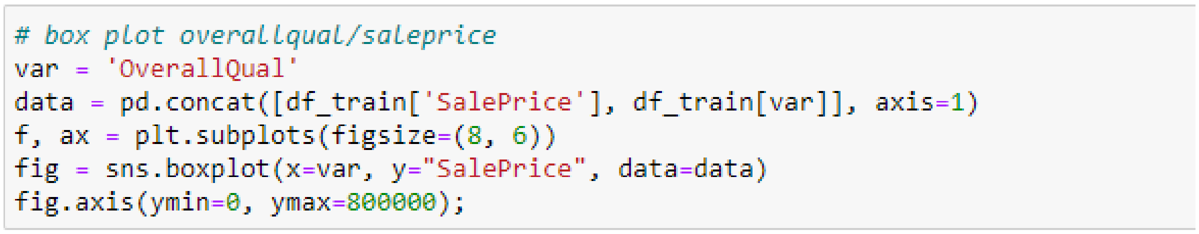
From this presentation we can see that there are many houses with NO garage or we can say that zero garage area and other than this most of the houses has garages of the area in the range of 200 to 600 sqft.

```
# scatter plot totalbsmtsf/saleprice
plt.figure(figsize=(10,10))
var = 'TotalBsmtSF'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
data.plot.scatter(x=var, y='SalePrice')
```

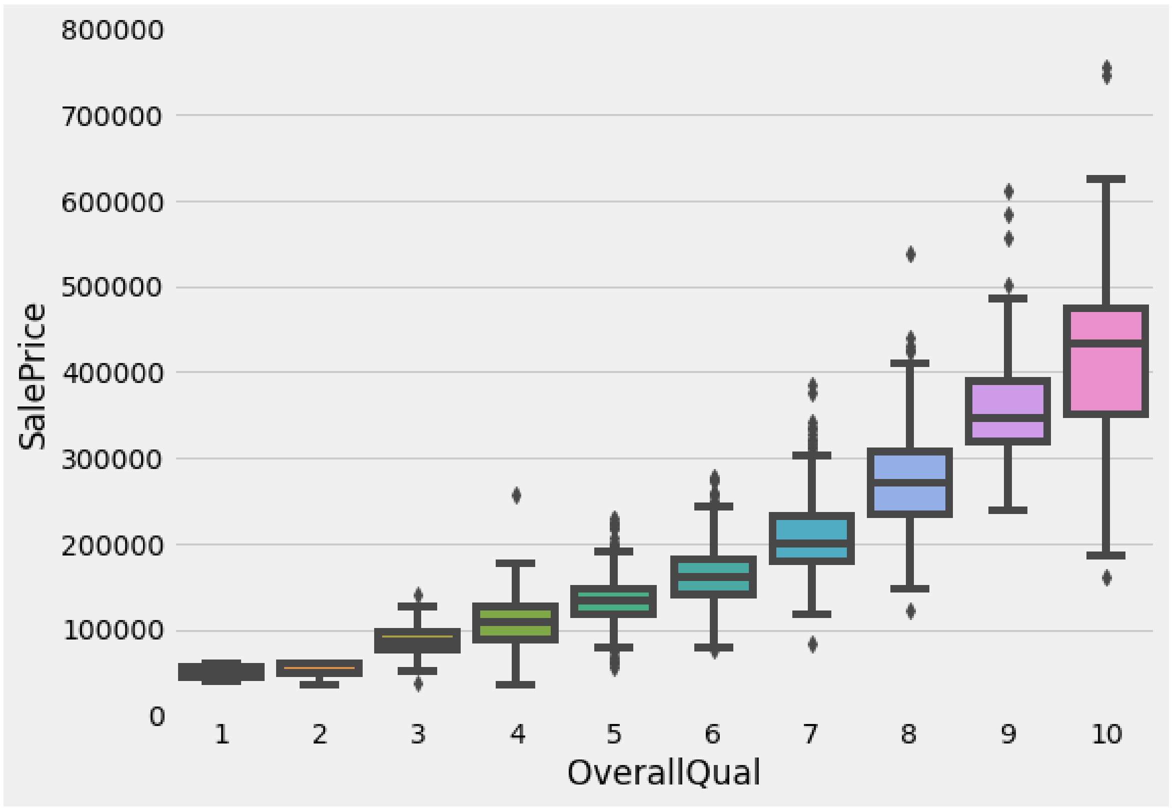
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a730a97d00>

<Figure size 720x720 with 0 Axes>
```



In this depiction we can observe that there are many houses with NO basement and houses with basement area of about 700 to 1700 sqft.

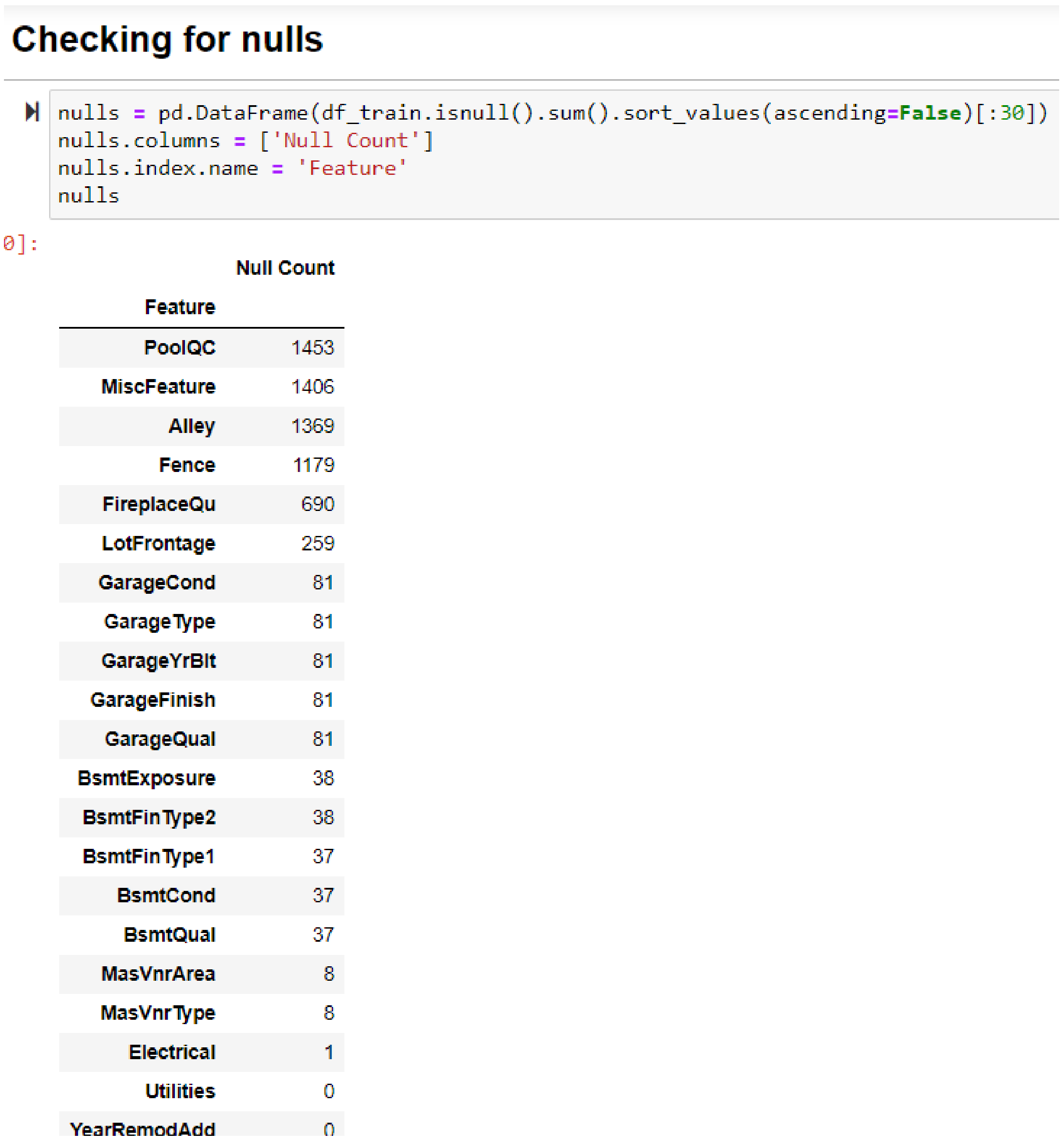


```
# box plot overallqual/saleprice
var = 'OverallQual'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
f, ax = plt.subplots(figsize=(8, 6))
fig = sns.boxplot(x=var, y="SalePrice", data=data)
fig.axis(ymin=0, ymax=800000);
```



From this we can have an intuition that with the increase in overall quality the SalePrice increases vigorously.

So, this was our EDA and now we move on to Preprocessing and applying our regression algorithms.



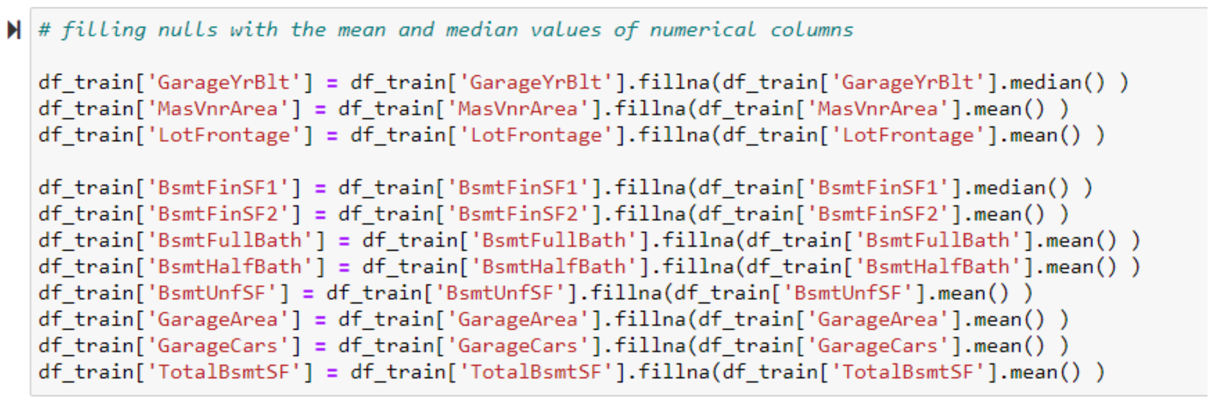
Checking for nulls

```python
nulls = pd.DataFrame(df_train.isnull().sum().sort_values(ascending=False)[:30])
nulls.columns = ['Null Count']
nulls.index.name = 'Feature'
nulls
```

0]:

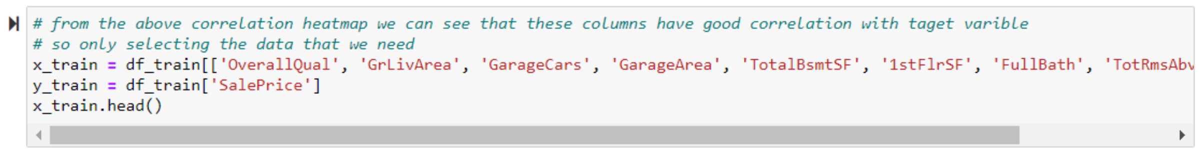
	Null Count
Feature	
PoolQC	1453
MiscFeature	1406
Alley	1369
Fence	1179
FireplaceQu	690
LotFrontage	259
GarageCond	81
GarageType	81
GarageYrBlt	81
GarageFinish	81
GarageQual	81
BsmtExposure	38
BsmtFinType2	38
BsmtFinType1	37
BsmtCond	37
BsmtQual	37
MasVnrArea	8
MasVnrType	8
Electrical	1
Utilities	0
YearRemodAdd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

Here we checked for null values in our dataset and from above we can see that there are many columns with a lot of null values. We filled the null values of the selected columns that had good relationship with our target variable SalePrice.



```
# filling nulls with the mean and median values of numerical columns

df_train['GarageYrBlt'] = df_train['GarageYrBlt'].fillna(df_train['GarageYrBlt'].median() )
df_train['MasVnrArea'] = df_train['MasVnrArea'].fillna(df_train['MasVnrArea'].mean() )
df_train['LotFrontage'] = df_train['LotFrontage'].fillna(df_train['LotFrontage'].mean() )

df_train['BsmtFinSF1'] = df_train['BsmtFinSF1'].fillna(df_train['BsmtFinSF1'].median() )
df_train['BsmtFinSF2'] = df_train['BsmtFinSF2'].fillna(df_train['BsmtFinSF2'].mean() )
df_train['BsmtFullBath'] = df_train['BsmtFullBath'].fillna(df_train['BsmtFullBath'].mean() )
df_train['BsmtHalfBath'] = df_train['BsmtHalfBath'].fillna(df_train['BsmtHalfBath'].mean() )
df_train['BsmtUnfSF'] = df_train['BsmtUnfSF'].fillna(df_train['BsmtUnfSF'].mean() )
df_train['GarageArea'] = df_train['GarageArea'].fillna(df_train['GarageArea'].mean() )
df_train['GarageCars'] = df_train['GarageCars'].fillna(df_train['GarageCars'].mean() )
df_train['TotalBsmtSF'] = df_train['TotalBsmtSF'].fillna(df_train['TotalBsmtSF'].mean() )
```

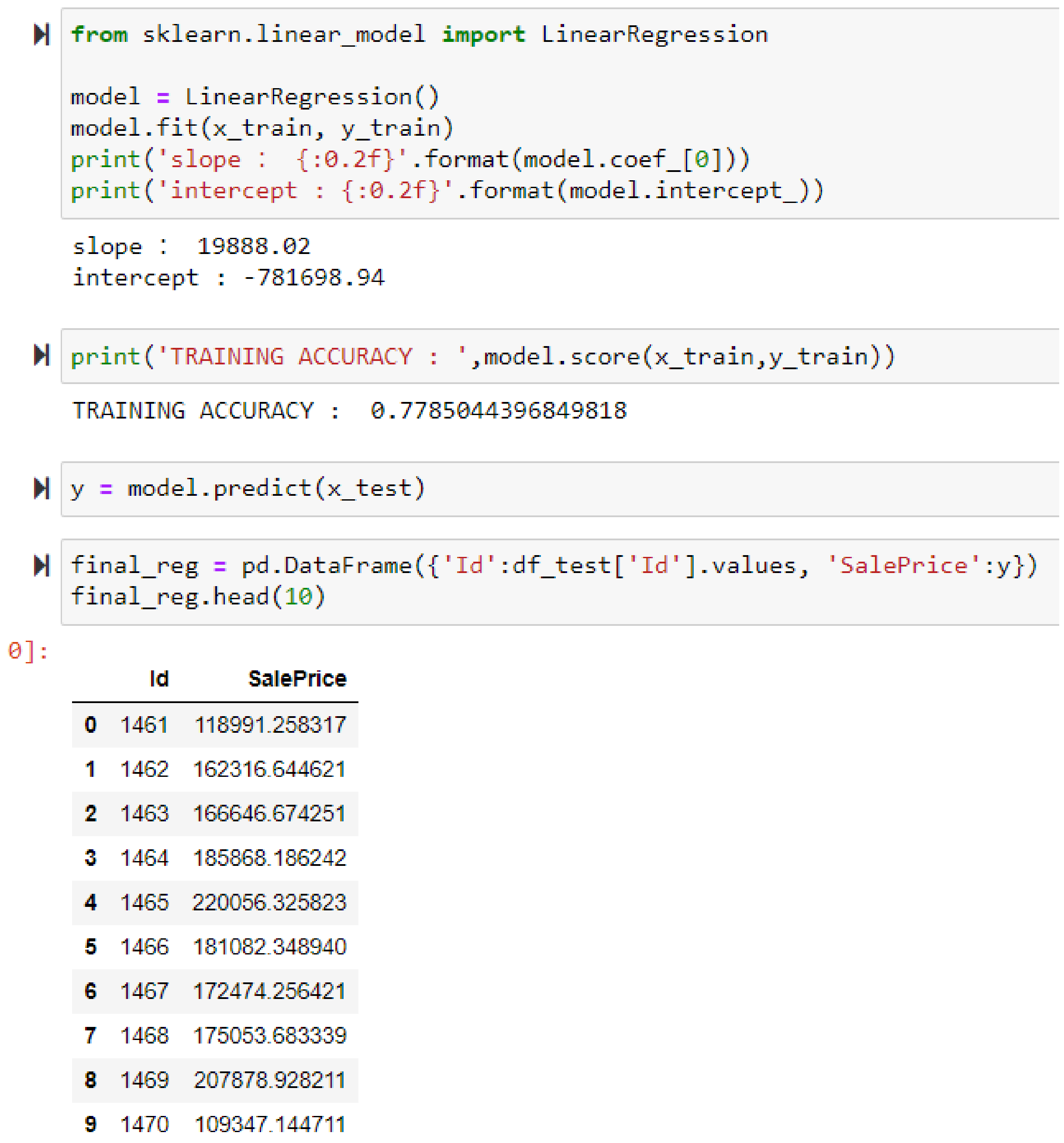
Here the columns were filled with their median and mean values to remove the null values.



```
# from the above correlation heatmap we can see that these columns have good correlation with taget varible
# so only selecting the data that we need
x_train = df_train[['OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea', 'TotalBsmtSF', '1stFlrSF', 'FullBath', 'TotRmsAbv
y_train = df_train['SalePrice']
x_train.head()
```

Here we selected the columns for our Regression algorithms, to make some good predictions.

Linear Regression



```python
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(x_train, y_train)
print('slope :  {:0.2f}'.format(model.coef_[0]))
print('intercept : {:0.2f}'.format(model.intercept_))
```

```
slope :  19888.02
intercept : -781698.94
```

```python
print('TRAINING ACCURACY : ',model.score(x_train,y_train))
```

```
TRAINING ACCURACY :  0.7785044396849818
```

```python
y = model.predict(x_test)
```

```python
final_reg = pd.DataFrame({'Id':df_test['Id'].values, 'SalePrice':y})
final_reg.head(10)
```

0]:

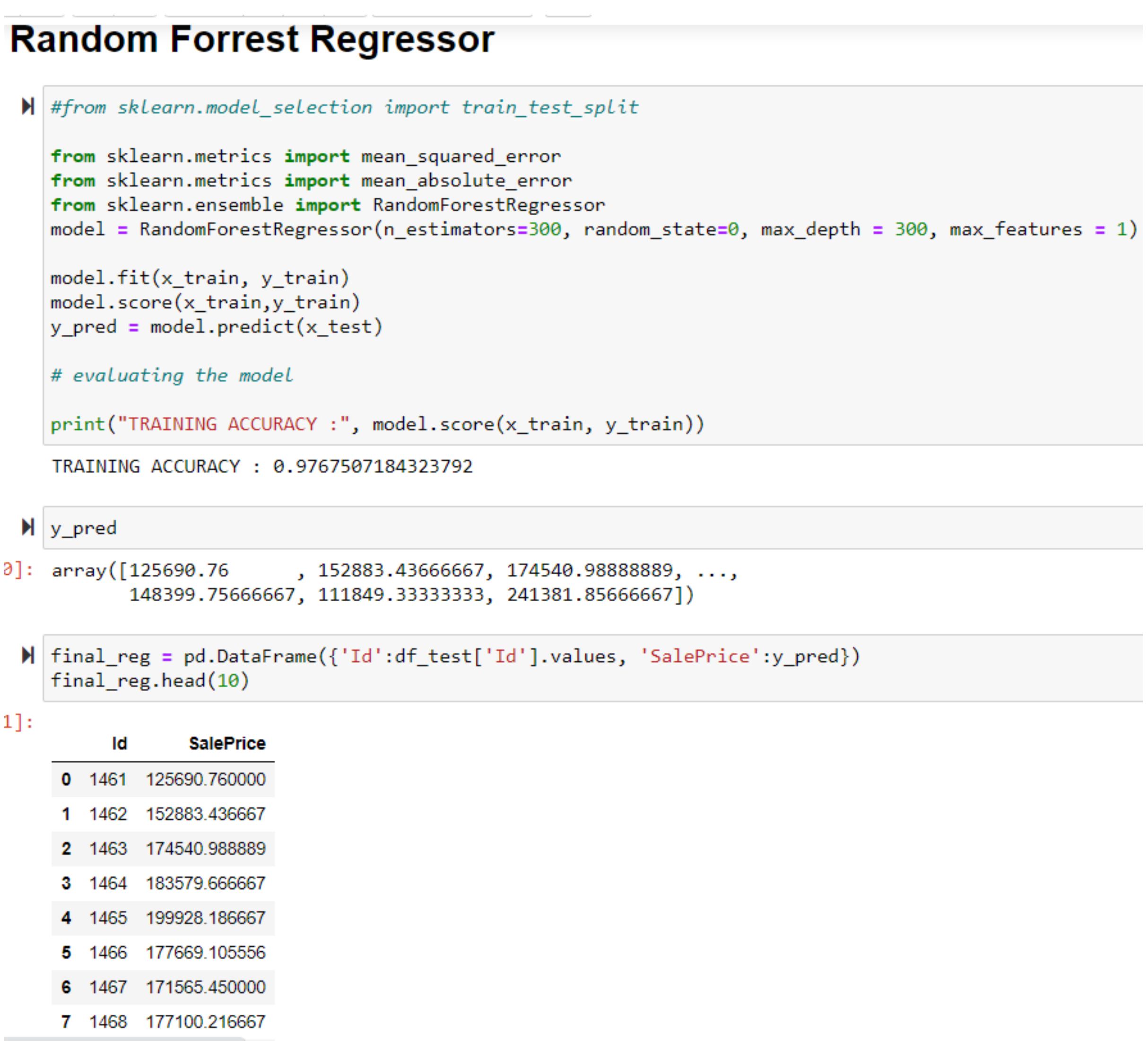
	Id	SalePrice
0	1461	118991.258317
1	1462	162316.644621
2	1463	166646.674251
3	1464	185868.186242
4	1465	220056.325823
5	1466	181082.348940
6	1467	172474.256421
7	1468	175053.683339
8	1469	207878.928211
9	1470	109347.144711

Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things:

1. does a set of predictor variables do a good job in predicting an outcome (dependent) variable?

2. Which variables in particular are significant predictors of the outcome variable, and in what way do they–indicated by the magnitude and sign of the beta estimates–impact the outcome variable? These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables.

In our model we used sklearn library to fit ad transform the data and predict the **SalePrice** of houses.
We got an accuracy of 77.85 % from linear regression.



Random Forrest Regressor

```
#from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_estimators=300, random_state=0, max_depth = 300, max_features = 1)

model.fit(x_train, y_train)
model.score(x_train,y_train)
y_pred = model.predict(x_test)

# evaluating the model

print("TRAINING ACCURACY :", model.score(x_train, y_train))
```

```
TRAINING ACCURACY : 0.9767507184323792
```

```
y_pred
```

```
0]:  array([125690.76      , 152883.43666667, 174540.98888889, ...,
            148399.75666667, 111849.33333333, 241381.85666667])
```

```
final_reg = pd.DataFrame({'Id':df_test['Id'].values, 'SalePrice':y_pred})
final_reg.head(10)
```

1]:

	Id	SalePrice
0	1461	125690.760000
1	1462	152883.436667
2	1463	174540.988889
3	1464	183579.666667
4	1465	199928.186667
5	1466	177669.105556
6	1467	171565.450000
7	1468	177100.216667

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

We used sklearn library to fit and transform our model and predict the SalePrice values of houses. From this method we got an accuracy of 97 % which is very high and can be can be result of overfitting.

Conclusion:

From the above predictions we conclude that Linear Regression was the best technique for predicting the <u>SalePrice</u> of the houses and can give a good selling value to them. This method, if used by the real estate property dealers, can help in increasing their income as they will know the reasonable value of the house and can negotiate accordingly. This will also increase their popularity as they will offer the best price in the market because of the help of *The Linear Regression*.