HOUSING: PRICE PREDICTION

Submitted by:

Mitesh Verma

ACKNOWLEDGMENT

I would like to acknowledge Mr. Shwetank Mishra (FlipRobo) for his vital cooperation and help in ensuring the successful completion of my assignment. He deserves the utmost credit for the assignment's outcome.

Finally, I would want to convey my sincere thanks Datatrained Academy and their guidance without them, the task would not have been accomplished.

The website that I referred are:

https://www.kaggle.com

https://www.w3schools.com

https://www.freecodecamp.org

https://github.com

https://www.geeksforgeeks.org

INTRODUCTION

- ## Business Problem Framing

  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.

- ## Conceptual Background of the Domain Problem

  This project is all about predicting the House Price for the Australia.

  One of the domains for better understanding is:
  https://www.surpriseaz.gov/448/Housing-Programs

- ## Review of Literature

  Information of the dataset

  a. Range Index: 0 to 1167
  b. Total Columns: 81 columns
  c. dtypes: float64(3), int64(35), object (43)

- ## Motivation for the Problem Undertaken

  This project is on the data science and machine learning model, build the model to predict the house pricing based on some features.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

	count	mean	std	min	25%	50%	75%	max
Id	1168.0	724.136130	416.159877	1.0	360.50	714.5	1079.5	1460.0
MSSubClass	1168.0	56.767979	41.940650	20.0	20.00	50.0	70.0	190.0
LotFrontage	954.0	70.988470	24.828750	21.0	60.00	70.0	80.0	313.0
LotArea	1168.0	10484.749144	8957.442311	1300.0	7621.50	9522.5	11515.5	164660.0
OverallQual	1168.0	6.104452	1.390153	1.0	5.00	6.0	7.0	10.0
OverallCond	1168.0	5.595890	1.124343	1.0	5.00	5.0	6.0	9.0
YearBuilt	1168.0	1970.930651	30.145255	1875.0	1954.00	1972.0	2000.0	2010.0
YearRemodAdd	1168.0	1984.758562	20.785185	1950.0	1966.00	1993.0	2004.0	2010.0
MasVnrArea	1161.0	102.310078	182.595606	0.0	0.00	0.0	160.0	1600.0
BsmtFinSF1	1168.0	444.726027	462.664785	0.0	0.00	385.5	714.5	5644.0
BsmtFinSF2	1168.0	46.647260	163.520016	0.0	0.00	0.0	0.0	1474.0
BsmtUnfSF	1168.0	569.721747	449.375525	0.0	216.00	474.0	816.0	2336.0
TotalBsmtSF	1168.0	1061.095034	442.272249	0.0	799.00	1005.5	1291.5	6110.0
1stFlrSF	1168.0	1169.860445	391.161983	334.0	892.00	1096.5	1392.0	4692.0
2ndFlrSF	1168.0	348.826199	439.696370	0.0	0.00	0.0	729.0	2065.0
LowQualFinSF	1168.0	6.380137	50.892844	0.0	0.00	0.0	0.0	572.0
GrLivArea	1168.0	1525.066781	528.042957	334.0	1143.25	1468.5	1795.0	5642.0
BsmtFullBath	1168.0	0.425514	0.521615	0.0	0.00	0.0	1.0	3.0
BsmtHalfBath	1168.0	0.055651	0.236699	0.0	0.00	0.0	0.0	2.0
FullBath	1168.0	1.562500	0.551882	0.0	1.00	2.0	2.0	3.0
HalfBath	1168.0	0.388699	0.504929	0.0	0.00	0.0	1.0	2.0
BedroomAbvGr	1168.0	2.884418	0.817229	0.0	2.00	3.0	3.0	8.0
KitchenAbvGr	1168.0	1.045377	0.216292	0.0	1.00	1.0	1.0	3.0
TotRmsAbvGrd	1168.0	6.542808	1.598484	2.0	5.00	6.0	7.0	14.0
Fireplaces	1168.0	0.617295	0.650575	0.0	0.00	1.0	1.0	3.0
GarageYrBlt	1104.0	1978.193841	24.890704	1900.0	1961.00	1980.0	2002.0	2010.0
GarageCars	1168.0	1.776541	0.745554	0.0	1.00	2.0	2.0	4.0
GarageArea	1168.0	476.860445	214.466769	0.0	338.00	480.0	576.0	1418.0
WoodDeckSF	1168.0	96.206336	126.158988	0.0	0.00	0.0	171.0	857.0
OpenPorchSF	1168.0	46.559932	66.381023	0.0	0.00	24.0	70.0	547.0
EnclosedPorch	1168.0	23.015411	63.191089	0.0	0.00	0.0	0.0	552.0
3SsnPorch	1168.0	3.639555	29.088867	0.0	0.00	0.0	0.0	508.0
ScreenPorch	1168.0	15.051370	55.080816	0.0	0.00	0.0	0.0	480.0
PoolArea	1168.0	3.448630	44.896939	0.0	0.00	0.0	0.0	738.0
MiscVal	1168.0	47.315068	543.264432	0.0	0.00	0.0	0.0	15500.0
MoSold	1168.0	6.344178	2.686352	1.0	5.00	6.0	8.0	12.0
YrSold	1168.0	2007.804795	1.329738	2006.0	2007.00	2008.0	2009.0	2010.0
SalePrice	1168.0	181477.005993	79105.586863	34900.0	130375.00	163995.0	215000.0	755000.0

Description of the dataset.

- Data Sources and their formats
    1. Information of the dataset.
    2. Description of the dataset.
    3. Null Values are present in the dataset.
    4. Most of the column's dtype is object.
    5. Outliers are present in the dataset.

- Data Pre-processing Done
  1. Removing all the null values and dropping the column with 70% of null values - filled with mean (), median () & mode ().
  2. Encode the object column - LabelEncoder()
  3. Feature Selection - SelectKBest & f_classif.
  4. Correlation.
  5. Visualization - Normal Distribution & Outliers.

- Data Inputs- Logic- Output Relationships
  1. Select top 40 Features.
  2. Correlation -
     i. TotRmsAbvGrd & GrLivArea, has the correlation of 82%.
     ii. GarageArea & GarageCars, has the correlation of 88%.
     iii. TotalBsmtSF & 1stFlrSF, has the correlation of 81%.
  3. Removed the outliers.

- Hardware and Software Requirements and Tools Used

  Anaconda-navigator

  jupyter notebook

  matplotlib-inline==0.1.6

  numpy==1.23.2

  packaging==21.3

  pickleshare==0.7.5

  platformdirs==2.5.2

  prompt-toolkit==3.0.30

  pyparsing==3.0.9

  python-dateutil==2.8.2

  scikit-learn==1.1.2

  scipy==1.9.0

  sklearn==0.05

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  - EDA.
  - Null Values.
  - Encoding.
  - Feature Selection.
  - Correlation.
  - Visualization.
  - Model Building.
  - Hyperparameter Tuning.
  - Testing dataset.

- Testing of Identified Approaches (Algorithms)

  Algorithms used for the training and testing.

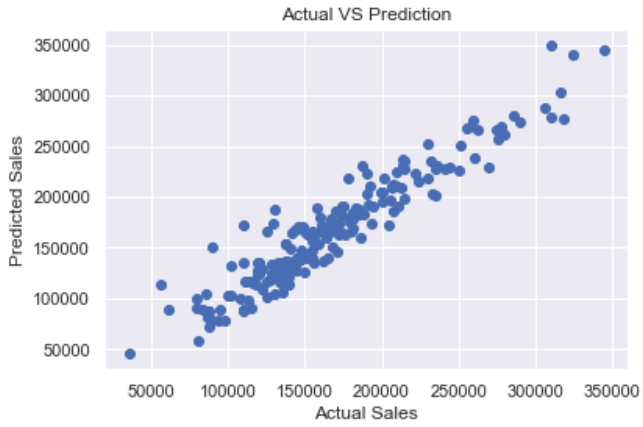
  - Linear Regression.
  - RandomForest Regressor.
  - AdaBoost Regressor.
  - GradientBoosting Regressor.
  - HistGradientBoosting Regressor.
  - XGB Regressor.
  - KNeighbors Regressor.

- Run and Evaluate selected models

- Linear Regression.

```
lr.fit(x_train,y_train)
score(lr, x_train,x_test,y_train,y_test,train = True)
score(lr, x_train,x_test,y_train,y_test,train = False)
```

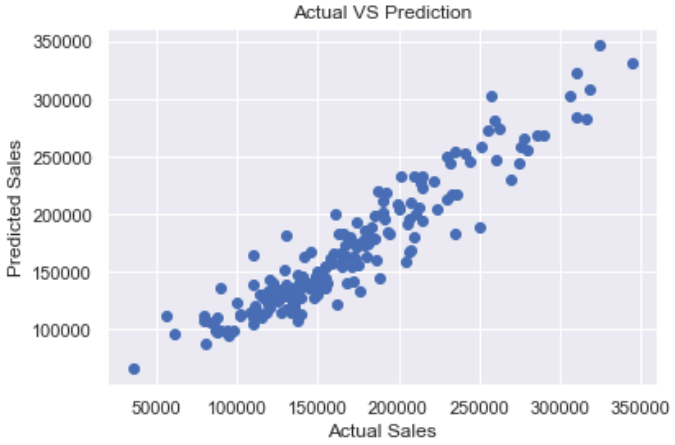
```
 ----- Train Result -----

R2 Score: 0.8760041136039864

 ----- Test Result -----

R2 Score: 0.8971125451644469

 ----- Model Evalution -----

Mean Absolute Error: 14211.539583153666

 Scatter Plot
```



- RandomForest Regressor.

```
rf.fit(x_train,y_train)
score(rf, x_train,x_test,y_train,y_test,train = True)
score(rf, x_train,x_test,y_train,y_test,train = False)
```

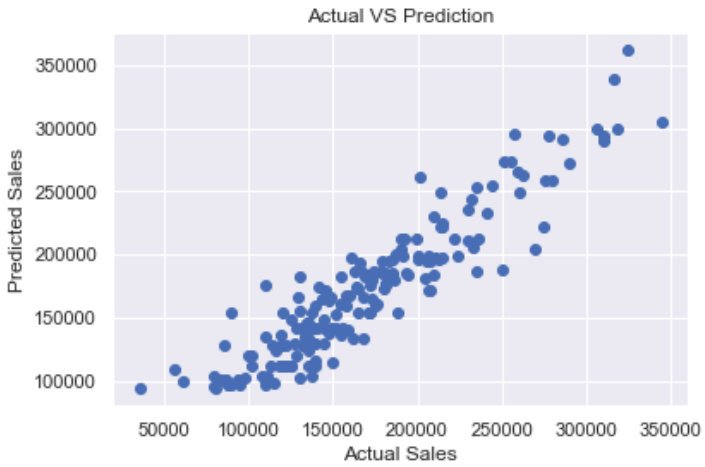
```
 ----- Train Result -----

R2 Score: 0.9815829559988335

 ----- Test Result -----

R2 Score: 0.8862419429705594

 ----- Model Evalution -----

Mean Absolute Error: 14824.485873873875

 Scatter Plot
```



- AdaBoost Regressor.

```
ada.fit(x_train,y_train)
score(ada, x_train,x_test,y_train,y_test,train = True)
score(ada, x_train,x_test,y_train,y_test,train = False)
```

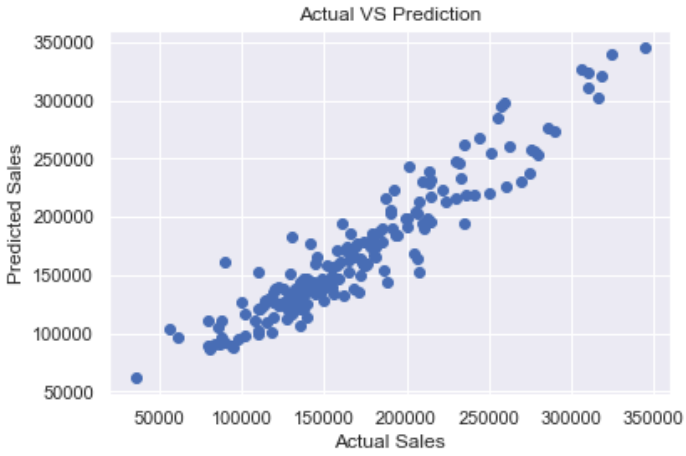
```
 ----- Train Result -----

R2 Score: 0.8892282646684366

 ----- Test Result -----

R2 Score: 0.860045413689696

 ----- Model Evalution -----

Mean Absolute Error: 16781.463879145893

 Scatter Plot
```



- GradientBoosting Regressor.

```
gb.fit(x_train,y_train)
score(gb, x_train,x_test,y_train,y_test,train = True)
score(gb, x_train,x_test,y_train,y_test,train = False)
```

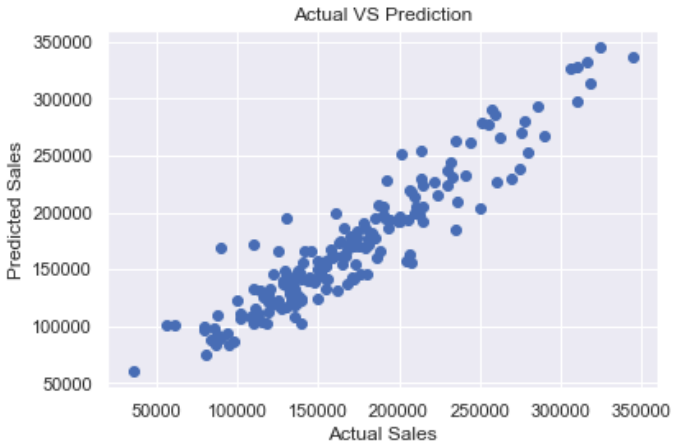
```
 ----- Train Result -----

R2 Score: 0.9595292656376133

 ----- Test Result -----

R2 Score: 0.892148595720675

 ----- Model Evalution -----

Mean Absolute Error: 14328.461379817269

 Scatter Plot
```



- HistGradientBoosting Regressor.

```
hgb.fit(x_train,y_train)
score(hgb, x_train,x_test,y_train,y_test,train = True)
score(hgb, x_train,x_test,y_train,y_test,train = False)
```

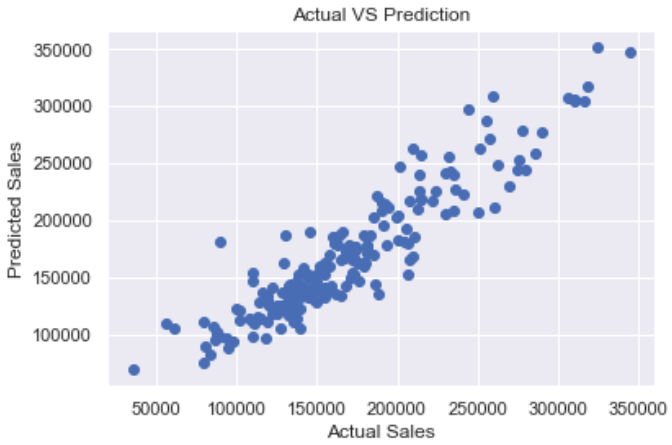
```
 ----- Train Result -----

R2 Score: 0.9870244410795318

 ----- Test Result -----

R2 Score: 0.8799940131761304

 ----- Model Evalution -----

Mean Absolute Error: 14616.643366379834

 Scatter Plot
```



- XGB Regressor.

```
xgb.fit(x_train,y_train)
score(xgb, x_train,x_test,y_train,y_test,train = True)
score(xgb, x_train,x_test,y_train,y_test,train = False)
```

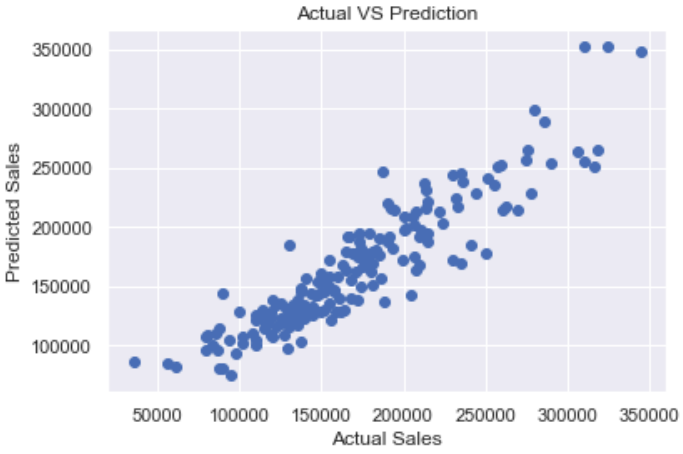
```
 ----- Train Result -----

R2 Score: 0.9999200464121489

 ----- Test Result -----

R2 Score: 0.8569479477466962

 ----- Model Evalution -----

Mean Absolute Error: 16353.914653716216

 Scatter Plot
```



- KNeighbors Regressor.

```
knn.fit(x_train,y_train)
score(knn, x_train,x_test,y_train,y_test,train = True)
score(knn, x_train,x_test,y_train,y_test,train = False)
```

```
 ----- Train Result -----

R2 Score: 0.8754474973001353

 ----- Test Result -----

R2 Score: 0.836770321726306

 ----- Model Evalution -----

Mean Absolute Error: 17267.794594594594

 Scatter Plot
```



- Interpretation of the Results

  GradientBoosting Regressor, is giving the best score among all other models.

CONCLUSION

- Key Findings and Conclusions of the Study

  Post Tuning score is best then the default parameters.