PROPERTY PRICE PREDICTION WITH REGRESSION ANALYSIS

*

Md. Sayed Hasan Emon(20201021)
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
sayed.hasan.emon@g.bracu.ac.bd

Md. Israk Hossain (20201150)
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
israk@gmail.com

Raya Subah(20201132)
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
raya@gmail.com

Nutan Chaudhury
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh

Mr. Syed Zamil Hasan Shoumo
Dept of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
shoumo.hasan@bracu.ac.bd

Mr. Shayekh Bin Islam
Dept of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
shayekh.bin.islam@g.bracu.ac.bd

I. Introduction

The project aims to predict property prices based on certain features(e.g., square feet, number of bedrooms, number of bathrooms, location, etc.). Here, we give some input as Area(Square Feet), BHK, Bath, and Location and our model gives us the estimated price of the property.

***Index Terms*—Artifical Intelligence; Regression; Decision Tree; XGB Regressor; encoding; error; Machine Learning**

II. DATASET DESCRIPTION

Initially, there were 8 features in our dataset. It is a regression problem. Because the nature of our output is a continuous value. Precisely, we are predicting continuous numerical value. In other words, we are trying to estimate a relationship between input variables and a continuous target variable. Again, the type of our data(input features) is numerical and we are trying to predict a numerical value, so regression is likely the right choice. On the other hand for the categorical problem, the dataset (input feature) is categorical (more likely textual) or it is used to make a decision or assign a label. Note that the boundary between regression and classification can be a bit blurry. For example, predicting the probability of an event occurring(a value between 0 and 1), might seem like a regression problem. However, this can also be a binary classification problem, where it classifies instances into two classes based on a threshold (e.g., if the probability is greater than 0.5, classify as class A; otherwise, classify as class B). Finally, the choice between regression and classification depends on the nature of the data, the addressing problem, and the type of output we need from our machine learning model.

Identify applicable funding agency here. If none, delete this.

There are 13000 data points. Our dataset has both Quantitative features and Categorical features.

III. DATA VISUALIZATION

A. HEATMAP

Initially, our dataset holds categorical values as well as numerical values. So In this state, we are trying to create a heatmap to visualize our data including categorical values.

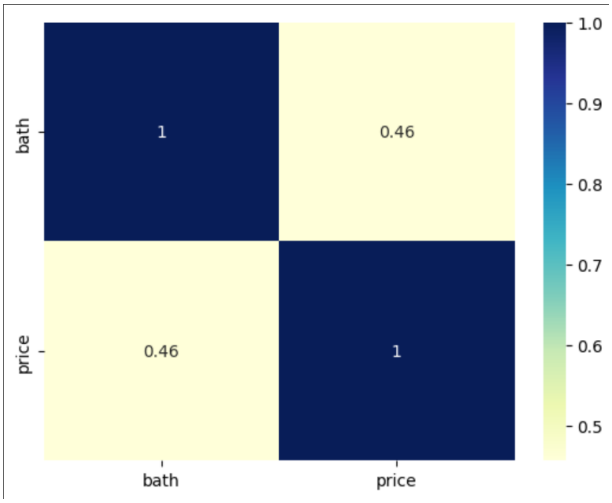

Fig. 1. Heatmap with categorical values

B. SCATTER PLOT

We are working with a regression problem, however, imbalanced datasets are more commonly associated with classification problems where there is a significant disparity in the number of instances among different classes. So, class

imbalance doesn't directly apply in the same way as it does for classification and we can not represent our continuous numerical value with bar chart as well. However, we can still encounter the distribution of our target variable(property price) for our regression model. So, we can similarly plot scatter plot of our target variable in a linear regression context

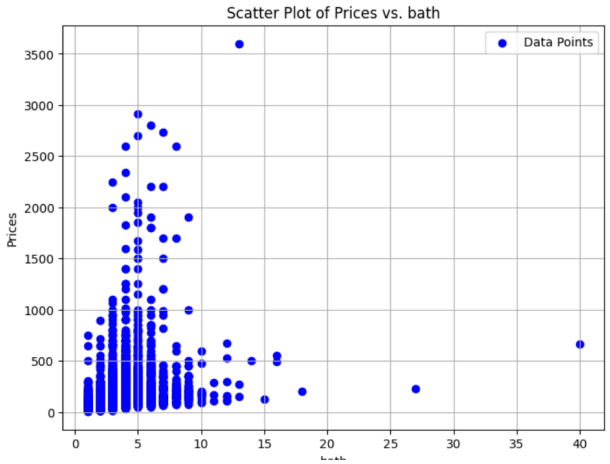

Fig. 2. Price vs Bath

IV. Data Preprocessing

A. Null Values

1) Dropping null values for location and size: After exploring our dataset, we can see that there are a very small number of null values for the features' location (1) and size (16). So removing these little values will not significantly impact the accuracy of our model. Finally, we can drop the sample's null values for location and size using the dropna() function.

2) Handling Null values for bath : From the dataset, we can see that almost half of the sample (6908) has 2 baths, so we set the null values for 'bath' with the median of bath, which is 2. Similarly, for 'balcony', the median is used to fill the null values.

B. Categorical value

1) Encoding 'location': Location is a categorical variable. We need to apply dimensionality reduction techniques here to reduce the number of locations. Dimensionality Reduction: Any location with fewer than 10 data points will be labeled as an "other" location. This way, the number of categories can be reduced by a huge amount. When we do one hot encoding, it will help us have fewer dummy columns.

C. Multicollinearity

occurs when two or more independent variables are highly correlated. This can cause problems for regression analysis because it becomes difficult to distinguish the individual effects of these correlated variables. By dropping one column (dropfirst = True) from the one-hot encoded variables, we can eliminate this correlation between them and reduce the multicollinearity issue. Example: Let's say we have a categorical variable "Color" with three categories: "Red," "Green," and "Blue." After one hot encoding, we will create three binary columns: "Color-Red," "Color-Green," and "Color-Blue." However, we only need two columns to represent this information because if "Color-Red" and "Color-Green" are both 0, then it's automatically understood that the color is "Blue." Including all three columns would lead to multicollinearity.

V. Feature Engineering

A. Size

From the feature 'size', we notice that the values contain numerical values and strings. In order to remove the string portion, we will split each piece of data and take only the numerical value and store it in another feature called 'bhk'. After that,t the size column is dropped.

B. Total Sqft

Some of the data in the feature 'total-sqft' contains ranges e.g., 1000-2500). To eliminate the range of values, we split the range and took the median of the range as the value for the data.

C. Price Per Sqft

Using 'price' and 'total-sqft', we create another column named 'price-per-sqft'. Here, the price is multiplied by 100,000 to get the actual price per square foot. After that, the feature 'total-sqft' is dropped. Also, the null values of 'price-per-sqft' are removed.

VI. Feature Scaling

In order to apply feature scaling, we calculate the standard deviation of each feature. As the standard deviations are not large, we decided not to scale any features.

VII. Splitting

We are using "Random Splitting" to split the dataset because the imbalanced dataset problem has already been handled. Otherwise, the model would have become biased towards the dominant class.

VIII. Model

We have used the following models: 1. Linear Regression 2. Decision Tree 3. Extreme Gradient Boosting Regressor

IX. Comparison

Since we are trying to calculate precision and recall for a regression problem, which is not applicable since precision and recall are classification-specific metrics. For regression problems, we can use different evaluation metrics, such as Mean Squared Error (MSE) and Mean Absolute Error (MAE).

TABLE I
MSE AND MAE

variable	Linear Regression	Decision Tree	XGB Regressor
MSE	15150.1741	1667.1722	657.8952
MAE	44.9039006074321	4.475504022121669	6.65415058523881

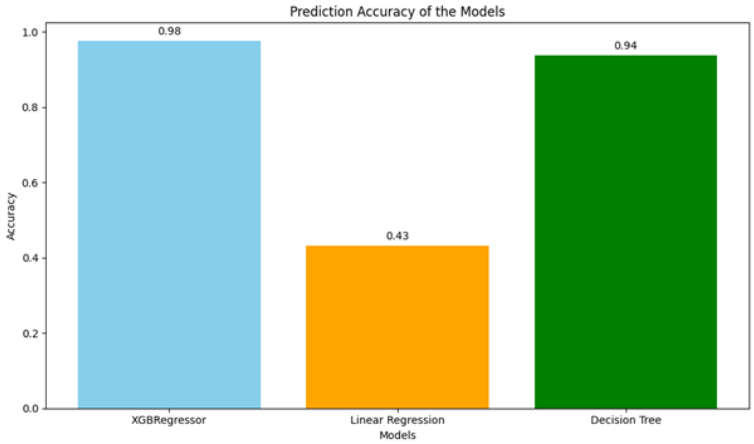

Fig. 3. Accurcay of the Models

X. CONCLUSION

To sum up, we can conclude that using XGBRegressor yields higher accuracy than linear regression and Decision tree. So XGB Regressor is the best model for this scenario.