PROJECT REPORT(HW2)

The model that gave me the best results is XGBRegressor

The basic idea behind this model is :

1. Fit a model to the data:
   $F_1(x) = y$
2. Fit a model to the residuals:
   $h_1(x) = y - F_1(x)$
3. Create a new model:
   $F_2(x) = F_1(x) + h_1(x)$

The working of this model is based on the idea of inserting more models that correct the errors of the previous model.

Specifically,

$$F(x) = F_1(x) \mapsto F_2(x) = F_1(x) + h_1(x) \ldots \mapsto F_M(x) = F_{M-1}(x) + h_{M-1}(x)$$

where $F_1(x)$ is an initial model fit to y

Since we initialize the procedure by fitting $F_1(x)$, our task at each step is to find $h_m(x) = y - F_m(x)$, where we can interpret h_m as a regression tree.

The performance of XGBRegressor:

I developed three models based on the dataset supplied to us

1) Simple Linear Regressor

2) Random Forest Regressor

3) XGBRegressor

Out of these, XGBRegressor produced the best results:

- Mean squared error (Value should be close to 0) : 0.026937869863
- r2 score (variance value should be close to 1) : 0.001093094631

The results of the Linear Regression Model:

- Mean squared error: 0.028183615432
- r2 score: 0.001091589370

The results of the Random Forest Regressor:

- Mean squared error: 0.031580246669
- r2 score: -0.14807825011

Based on these two metrics (Mean Squared Error and r2 score), we can say that XGBRegressor performed the best out of these three models.

Experiences

1) I have plotted a scatter plot in my Jupyter notebook using the 'Latitude' and 'Longitude' features in the dataset. After I had plotted the graph, including a parameter 'alpha' in the df.plot.scatter() function later, changed my deduction completely. The 'alpha' parameter creates the chart with transparent points. Therefore, if a point overlaps another point, the point will get darker. While I was trying to estimate the geoghaphic location of the majority of houses in the data set, I picked a dense area but to my surprise when I added the 'alpha' parameter to my function, the sparse area had more houses which were overlapping in the plot and therefore were not visible earlier. Seeing how a single parameter could make such a difference to the final deduction is very intriguing.

2) While I was trying to estimate how the 'logerror' value varied according to the transaction dates. I was bogged down trying to understand the line chart of 'logerror' vs 'transactiondates'. The plot was a messy description of the relationship of the X and Y fields. What I didn't realize was that each transaction date had multiple logerror values in the dataset. When I figured this out I calculated the mean of logerror values for each transaction date. Then I went further and calculated the mean logerror for

each month of the year. The final line chart between ‘month’ and ‘logerror’ showed a result that could be easily analyzed by the reader. Therefore, giving more specific data to the plots helps us make better deductions.