Homework 3

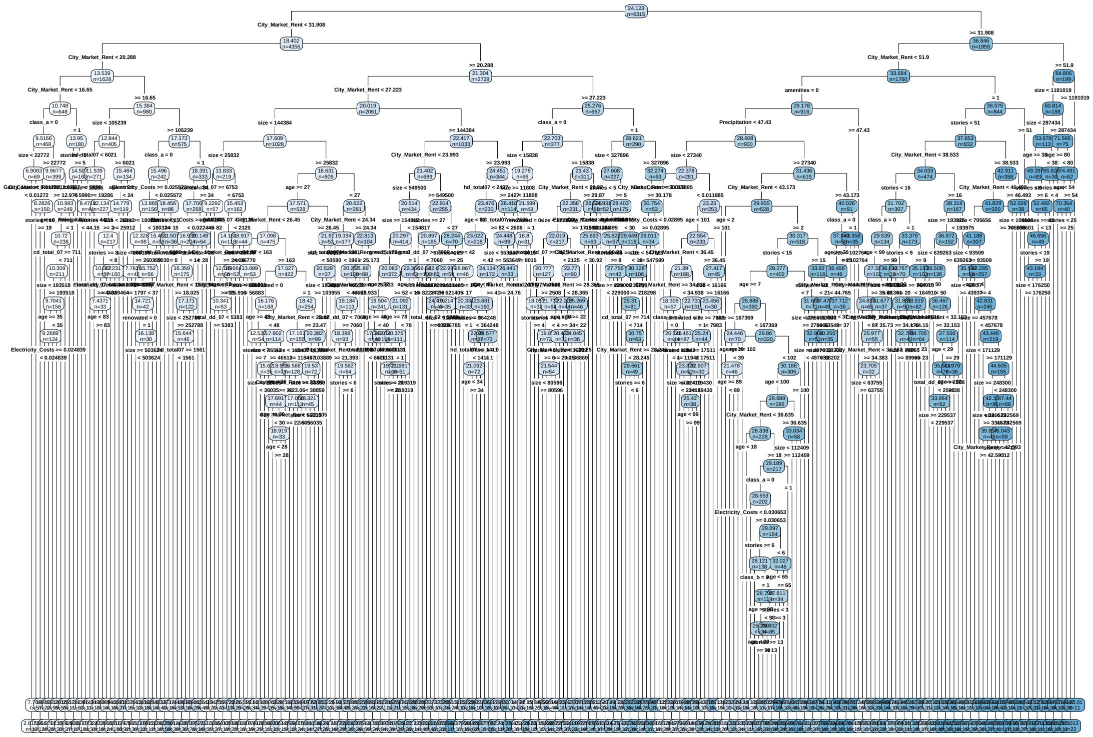
Patrick Massey

4/4/2022

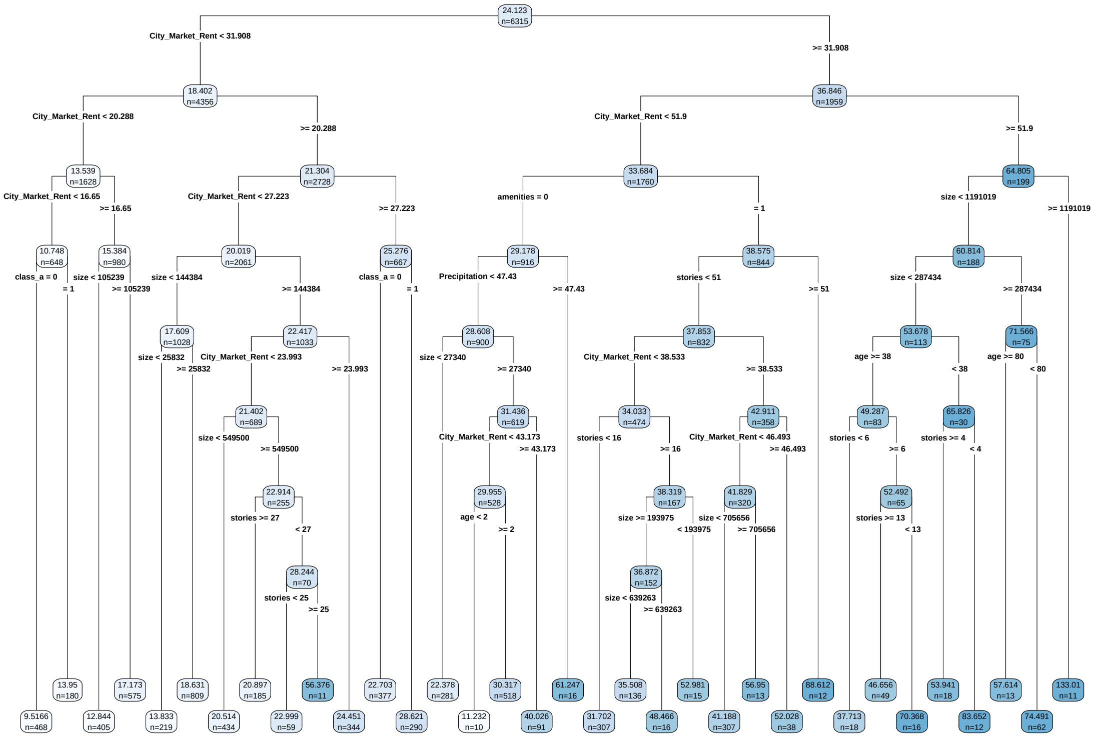
Before developing any models we first begin by performing some feature engineering. The first feature we engineer is the outcome variable of interest revenue which represents the revenue per square foot per calendar year. In order to create this feature we first scale down leasing_rate to a percentage by dividing by 100, and then multiplying that by the rent. We also create a new feature called utility_cost which is the sum of gas and electricity costs for rents that are quoted on a net contract basis. The purpose of this new feature is to capture the costs associated with a rental offered on a net contract basis. We then create a training set and a testing set with a split of 80/20. This gives us 6315 observations in our training set and 1579 observations in our testing set.

To begin developing our model we start with a linear model using all features of the data set excluding, CS_PropertyID, cluster, leasing_rate, Rent, LEED, and Energystar. We remove CS_PropertyID as it is just a unique building ID, and for similar reasons we remove cluster. We remove leasing_rate and Rent since these variables directly calculate our outcome variable. Lastly we remove LEED and Energystar because we are only concerned if a building is green certified or not, and not what kind of green certification a building may have. We capture this with the green_rating feature.

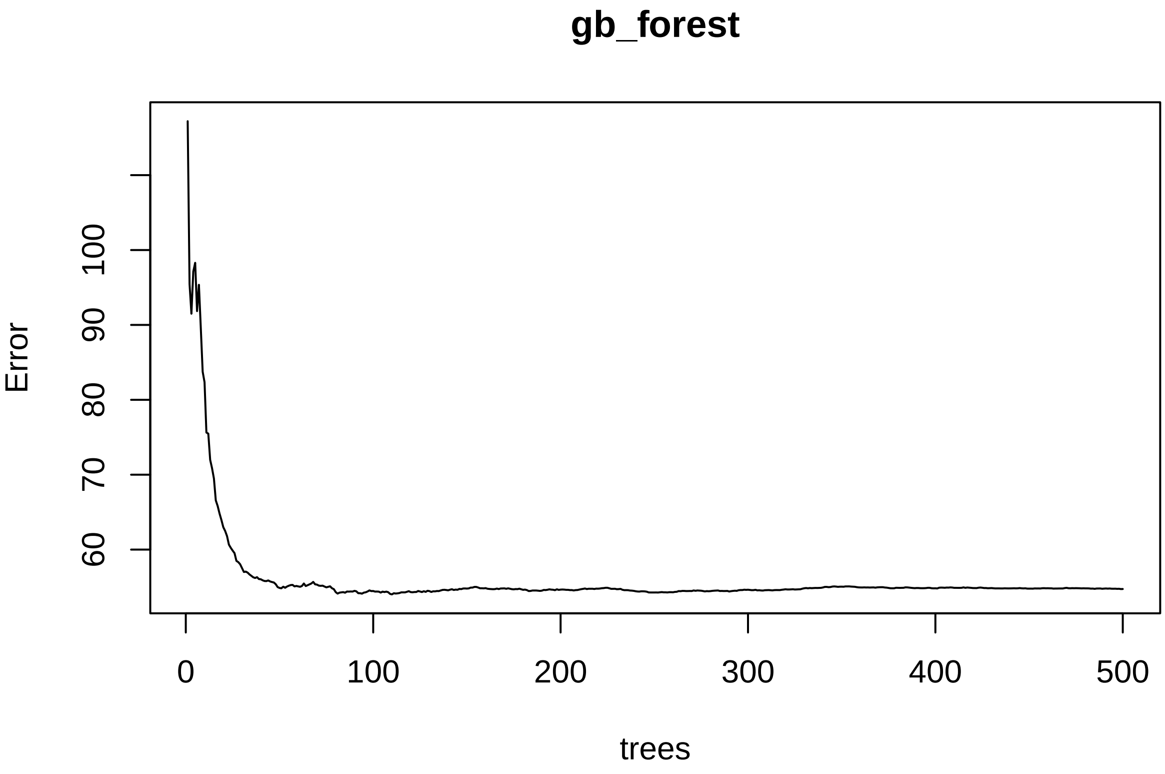
After getting a baseline model we then moved onto predicting using a tree model. The initial tree model generated, shown below, was extremely complex and not readable. This indicated that there might be some overfitting happening.



We then pruned our tree using the 1se method which generated the much simpler decision tree shown below. This tree sacrifices a marginal amount of performance for a much simpler tree.



The visualization of the tree really highlighted the interactions that were not included in our baseline linear model. Naturally after seeing the performance of the tree as compared to the linear model we wanted to see if it could be improved upon using a random forest.



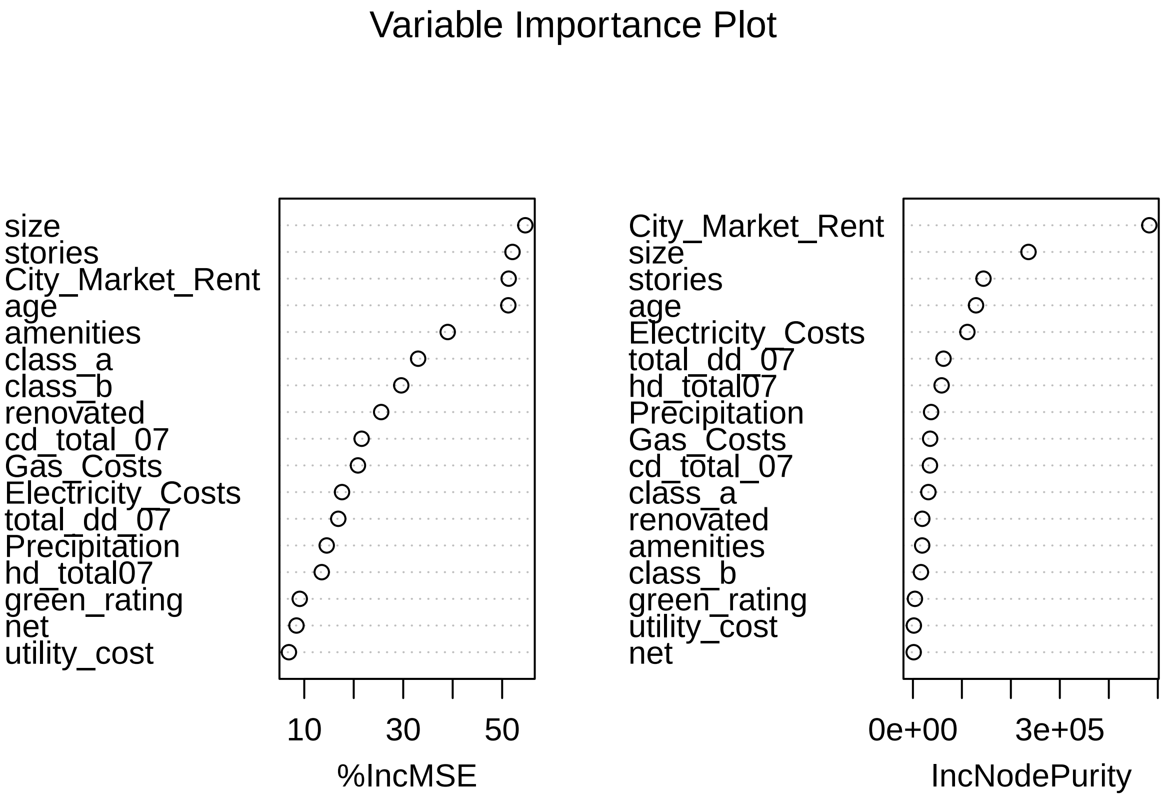
We see that our error really starts to bottom out around 100 trees. The performance of our models is shown below.

Model	RMSE
Linear	10.378612
CART	9.978718
Pruned Tree	10.392185
Random Forest	7.746483

The random forest provides a significant reduction in RMSE as compared to our baseline linear model.

Now that we have developed a model for predicting the revenue generated from an building we will look at the importance of the variables we have used in our model.

```r
varImpPlot(gb_forest, main = "Variable Importance Plot")
```


Variable Importance Plot
size
stories
City_Market_Rent
age
amenities
class_a
class_b
renovated
cd_total_07
Gas_Costs
Electricity_Costs
total_dd_07
Precipitation
hd_total07
green_rating
net
utility_cost
%IncMSE
City_Market_Rent
size
stories
age
Electricity_Costs
total_dd_07
hd_total07
Precipitation
Gas_Costs
cd_total_07
class_a
renovated
amenities
class_b
green_rating
utility_cost
net
0e+00
3e+05
IncNodePurity