Daniel Rees

5/7/2022

STP494

Final Project

Roller Coaster Analysis

Roller coasters play a pivotable role in the success of entertainment within the states. The thrilling nature of these attractions allow for a fun experience with friends through fast speeds and fanatical heights. In this project, I will be evaluating the factors within a roller coaster dataset recorded in 2015. This dataset contains categorical variables: Park, Track and Inversions, and continuous variables: Speed, Height, Drop, Length and Duration. Many believe that the tallest coaster in the park is the most thrilling due to its high velocities. However, could shorter attractions produce similar speed? For this project, I will be creating models to determine how the maximum height of a roller coaster can predict the maximum speed.

The dataset contains 242 entries recorded for the independent variable: height, and dependent variable: speed. For accuracy of the models, we will use a train/test split in order to reduce model error. Understanding the variables is important before performing the analysis in order to understand the different characteristics each factor holds:

Speed	
Mean	55.35062
Standard I	1.202307
Median	55
Mode	55
Standard I	18.66483
Sample Va	348.3759
Kurtosis	2.442264
Skewness	0.526582
Range	144.6
Minimum	4.5
Maximum	149.1
Sum	13339.5
Count	241

Height	
Mean	123.7149
Standard I	4.365444
Median	115
Mode	108.3
Standard I	67.76992
Sample Va	4592.762
Kurtosis	2.991587
Skewness	1.254487
Range	412
Minimum	8
Maximum	420
Sum	29815.28
Count	241

As the descriptive statistics show, height contains a larger range with a maximum of 420 ft. Speed on the other hand has a maximum of 149 mph. We can also observe that height has much more variation within its data which could account for outliers within the process. Without observing the numbers, we can assume that height plays an essential role in speed due to gravitational pull. Since gravity is measured in $\frac{m}{s^2}$ we can confirm that height contains exponential characteristics that will determine speed. Therefore, as height increases, speed will increase in exponentially.

For this analysis, I will be using kNN, single trees and boosting methods to find the best fit. The k-nearest neighbor test (kNN) will utilize different surrounding data points to predict the most common value. We will find the most accurate k-value for this model in order to determine the lowest mean squared error of the graph. Although higher k-value lead to complex results, we want to avoid this to prevent over fitting. The next test will be using decision trees to determine accurate prediction by splitting the data into most-likelihood values. The number of times the decision is split will constitute the size of the tree. Same with the kNN method, we want to minimize the tree as much as we can to prevent overfitting. An extension of this method will be boosting which we will also run on the dataset. Boosting combines the strategy of trees in order to estimate a final prediction based on tree decisions. At first, the graph will be split twice and then reveal a prediction line. Then, the data will split again and alter this line to create a more accurate prediction, we will run this as many times necessary until we find the line of best fit, with the minimum RMSE.

KNN:

The KNN model will be a useful tool in predicting the value of height based on speed. This non-linear test utilizes relevant neighboring points for every value of the independent factor. As a result, a prediction will be plotted for every specific value of speed, giving us a jagged line that can accurately predict the data with minimum error. In this test, we well try different values of k-nearest neighbors to see which one gives the most accurate model. Although a higher k-values might yield a more complex model with solution, it is important to find the k-value with an average bias and variance. First, we will split the dataset into a train and test dataset where the train is 75% of the data and the test is 25%. This way we can predict how accurate the two datasets are compared to one another.

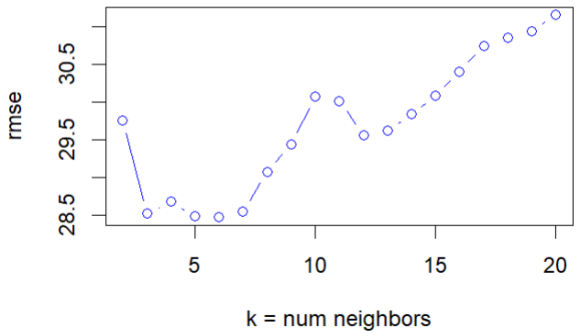


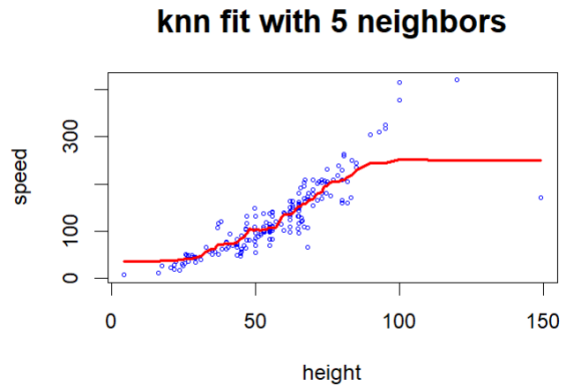
Figure 1.

Figure 1 evaluates the root mean squared error for k= (0,20) neighbors in the model. RMSE appears to increase as more neighbors are being accounted for. The sample above is an example of how high variance can increase error. Based on the graph, we can expect the smallest error to be between 2-7 nearest neighbors.

```
> print(rmseres)
 [1] 29.75019 28.52021 28.67873 28.48219 28.47487 28.54579 29.06819 29.43654 30.07350
[10] 30.01535 29.55965 29.61759 29.83268 30.08288 30.40139 30.74605 30.85508 30.93652
[19] 31.15102
```

Figure 2.

As seen in figure 2, the lowest mean-squared error is 28.4787 at 5 nearest neighbors. This was predicted by the first chart as neighbors increased with low bias. The final graph representing the kNN fit is:



Single tree:

For the single trees test, we will find the probable values through a series of branches and nodes that predict the most accurate value for each given speed. For the parameters in our decision tree, we have ‘minsplit’ and ‘cp’ (complexity parameter). For the minsplit, this value represents the smallest number of times the decision tree should be split. The complexity parameter represents the minimum number of times the model will be improved.

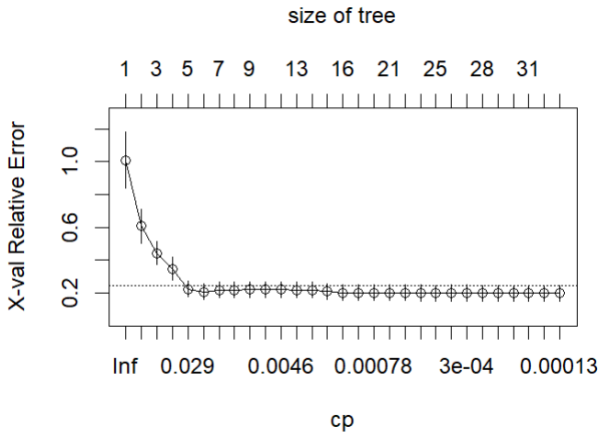


Figure 4

For this graph. The parameters set are (minsplit=2, cp=0.001) We can see that the lowest RMSE begins at a complexity parameter of around 0.029, as well as the size of a tree beginning to minimize are after the size of tree is 5. However, regardless of the change in the complexity

parameter, graph will follow the same rules. Following this test, we will record the different RMSE for different 'minsplit' values to see how each test performs with a cp=0.001

Min. Split	cp	RMSE
2	0.005	23.56674
3	0.005	25.68448
4	0.005	25.68448
5	0.005	27.52846
6	0.005	27.52846
7	0.005	25.56394
8	0.005	25.56394
9	0.005	25.56394
10	0.005	25.56394
11	0.005	25.56394
12	0.005	25.56394

Figure 5

The decision tree with only 2 splits yielded the lowest error at 23.56674. However, given the complexity of a size=2 tree, we will choose the next biggest tree with the lowest RMSE. This value would be cp=7. This means that a single tree can contain 7 observations in the parent node that could be split further giving a final RMSE=25.56394.

Boosting:

In conjunction with the single tree method, the boosting method combines a series of different size trees to create a predicting line. The test is measured by different iterations or tests that correspondently refine the final fit. For this test, we will be testing a sample of 30 iterations starting at the two-tree mark. The parameters which we will be testing will be the n.trees value which constitutes the number of iterations used in making concluding fit.

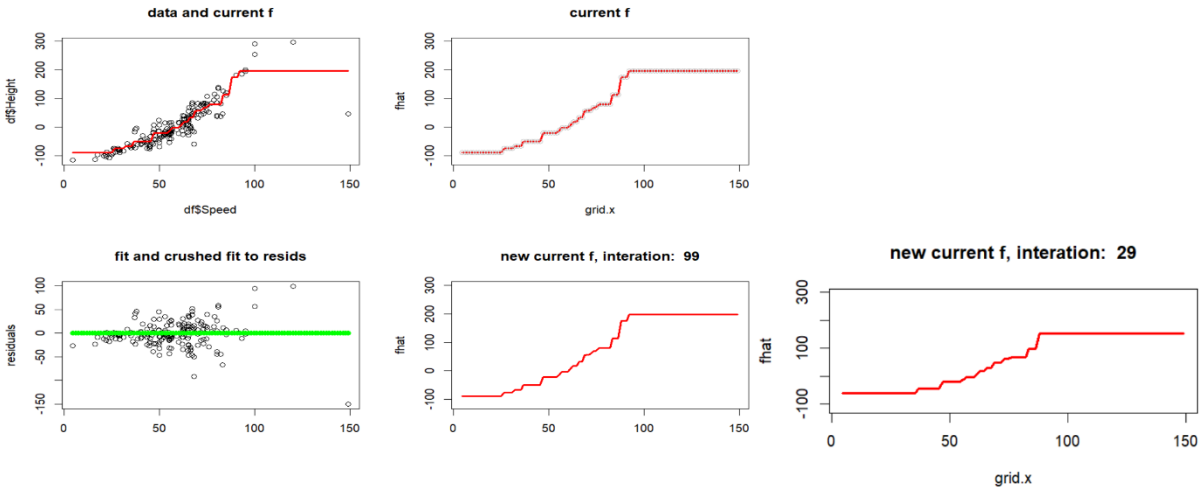


Figure 6

Figure 6 include a timelapse of iteration from 99 to 29. The comparison between the two is to show how the fit becomes less deformed after too many n.trees. Increasing the number of trees in the boosting plot will help minimize error, but iteration too high will lead to over-fitting. So, it is crucial to choose a number minimizes the RMSE without becoming too complex.

n.trees	RMSE
2	40.4556
4	31.696
6	27.984
8	26.1545
10	25.1942
12	25.0015
14	24.6844
16	24.5163
18	24.4249
20	24.9938
22	25.0386
24	25.0305
26	25.4345
28	26.0931
30	26.2453

Figure 7

Figure 7 concludes that at iteration 18, we will have the smallest RMSE of 24.4249. Since we want to choose the minimum number of iterations use to prevent overfitting, we will conclude that iteration 18 yields the best results.

Conclusion:

The prediction of speed through height provided an interesting way to test different methods. The most effective method was boosting which provided the smallest RMSE of 24.4249. Although the nearest neighbor method gave us a small RMSE, the outliers that were caused by the maximum height made it hard to predict values with a high independent value. Since the kNN test relies heavily on clustered values with low variances, the outlying values prevented prediction from becoming accurate due to lack of data. The overall results concluded that lower height values do not produce higher speeds.