1)

Predicting National Park Service Trail Popularity With Physical, Behavioral, and Reported Measures: Summary of Classification Techniques

Report by Sammy Rose Dobrozsi

December 11th, 2023

With over 21,000 miles of recreational trails managed by the United States National Parks Service, it is critical to understand how many trail users are going to different trails and for what reasons in order to allocate resources and direct future development. The present data was scraped from the website AllTrails roughly three years ago and includes behavioral measures of trail users (e.g. trail usage levels), reported measures by trail users (e.g. trail user reviews), and physical attributes of trails (length, elevation). AllTrails is a trail database, hosting GPS files of trails entered by users supported by metadata, focused on a phone app that can assist with trail selection and navigation. The different types of data are advantageous for gaining a more holistic view of what makes trails popular (and presumably more desirable to users).

Three types of classification analysis have been performed on this dataset: logistic regression, neural network modelling, and classification trees. In this report the results of these methods on this dataset are summarized for predicting trail whether a trail is popular or not.

2) Logistic Regression Analysis

Through logistic regression, the best model found utilized all available predictors with a cutoff value of 0.42:

```
Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -2.932e+01  1.112e+03  -0.026 0.978963
length                  -1.323e-06  4.824e-06  -0.274 0.783854
elevation_gain           5.825e-04  1.682e-04   3.463 0.000534 ***
difficulty_rating3      -1.678e-01  2.103e-01  -0.798 0.424816
difficulty_rating5      -6.507e-03  2.662e-01  -0.024 0.980498
difficulty_rating7      -2.990e-01  4.129e-01  -0.724 0.468903
route_typeout and back   9.946e-02  1.759e-01   0.565 0.571782
route_typepoint to point -2.440e-01  2.812e-01  -0.868 0.385496
visitor_usage2           8.296e-02  1.796e-01   0.462 0.644180
visitor_usage3           2.651e-01  3.404e-01   0.779 0.436189
visitor_usage4           1.403e+01  6.466e+02   0.022 0.982693
avg_rating1              9.341e+00  4.885e+03   0.002 0.998474
avg_rating1.5            8.742e+00  1.081e+04   0.001 0.999355
avg_rating2              9.190e+00  3.800e+03   0.002 0.998071
avg_rating2.5            8.878e+00  3.513e+03   0.003 0.997984
avg_rating3              8.088e+00  1.822e+03   0.004 0.996459
avg_rating3.5            2.377e+01  1.112e+03   0.021 0.982946
avg_rating4              2.538e+01  1.112e+03   0.023 0.981792
avg_rating4.5            2.610e+01  1.112e+03   0.023 0.981275
avg_rating5              2.624e+01  1.112e+03   0.024 0.981175
num_reviews              1.556e-01  8.643e-03  18.008  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2754.4  on 1986  degrees of freedom
Residual deviance:  992.0  on 1966  degrees of freedom
AIC: 1034
```

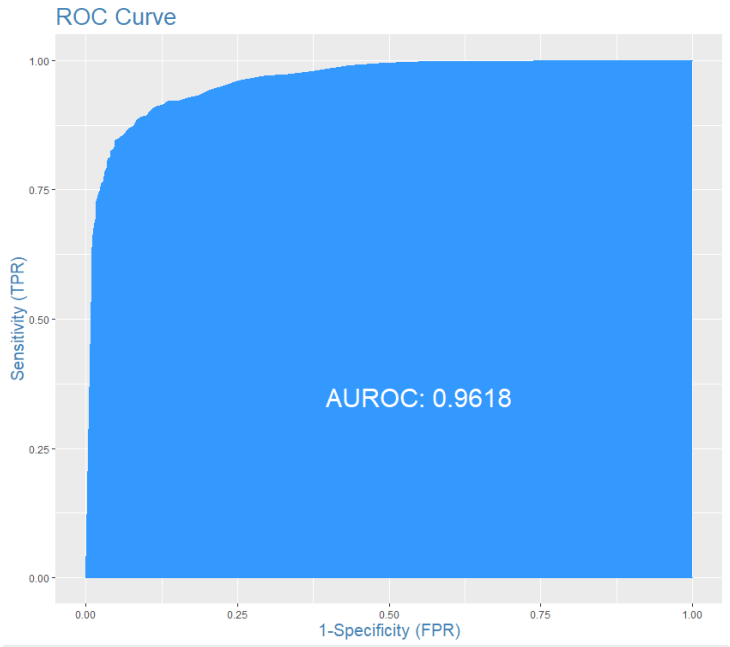
Full logistic regression model

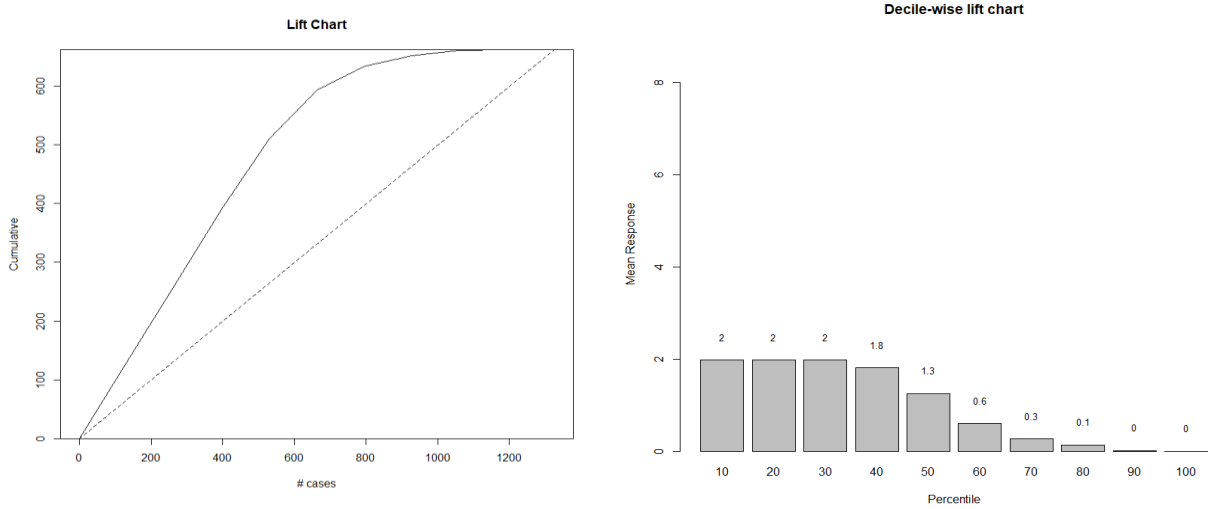
The full model was retained because removing predictors only marginally improved accuracy and performance metrics but was overall likely harmful to the model's applicability and practical usage to individuals in the trails industry. Other models attempted included removing Elevation Gain (reached via forward, backwards, and comprehensive variable selection), and using only Elevation Gain, Average Rating, and Number of Reviews (arrived at via stepwise AIC). A likely reason that only Elevation Gain and Number of Reviews are significant predictors in this model is probably multicollinearity between some of the predictors. Regardless, the difference between Null and Residual deviance produces a highly significant chi square test statistic and shows that the model itself is overwhelmingly significant (p-value approaching zero), so leaving all variables intact is theoretically sound.

```
> #Confusion Matrix
> confusionMatrix(valid.df[,1], pred, threshold = 0.42)
    0   1
0 604  72
1  60 590
> #Misclassification Error
> misClassError(valid.df[,1], pred, threshold = 0.42)
[1] 0.0995
> #Overall Model Specificity
> 1 - misClassError(valid.df[,1], pred, threshold = 0.42)
[1] 0.9005
```

Confusion Matrix, Misclassification Error, and Model Accuracy of Full Logistic Regression Model

This cutoff value balanced false positive and negative error rates, and only marginally improved overall model accuracy.





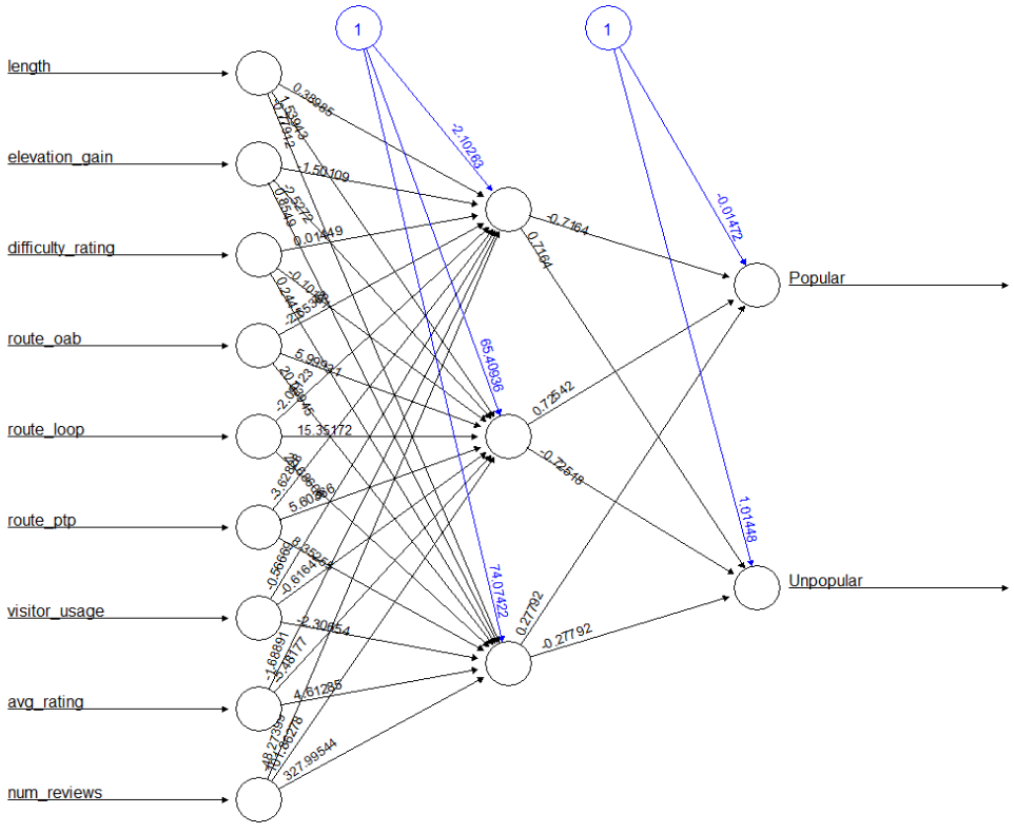
ROC Curve and Lift Charts for Full Logistic Regression Model

The ROC curve demonstrates excellent performance at a majority of cutoff values. The lift chart shows potentially slow convergence compared to a theoretically random model, but the model performs well in all other respects.

The model without elevation gain had an AIC of 1044, which was worse than the full model's, and the model with only Elevation Gain, Average Rating, and Number of Reviews had an AIC of 1023, an improvement over the full model but of the same magnitude as removing Elevation Gain hurt the AIC.

3) Neural Network Analysis

Through Neural Network Analysis, the best model identified was a neural network of the full model with all predictors and three hidden layers.



Full Neural Network Model w/Three Hidden Layers

```
     0   1
  0 926 110
  1  69 882
>
> sum(diag(c.mat1))/sum(c.mat1) #accuracy
[1] 0.9099144
> c.mat1[2,2]/sum(c.mat1[,2])   #sensitivity
[1] 0.8891129
> c.mat1[1,1]/sum(c.mat1[,1])   #specificity
[1] 0.9306533
```

```
     0   1
  0 604  86
  1  49 587
>
> sum(diag(c.mat11))/sum(c.mat11) #accuracy
[1] 0.89819
> c.mat11[2,2]/sum(c.mat11[,2])   #sensitivity
[1] 0.872214
> c.mat11[1,1]/sum(c.mat11[,1])   #specificity
[1] 0.9249617
```

Classification tables, model accuracy, sensitivity, and specificity of the full Neural Network model without with 3 layers. Left: Training, Right: Validation

The same model as was derived through stepwise AIC was also constructed as a neural network and tested and found to be nearly identically performing to the full model with 6 hidden layers. This produced a model that was more accurate and improved error rates.

```
     0   1
  0 926 110
  1  69 882
>
> sum(diag(c.mat1))/sum(c.mat1) #accuracy
[1] 0.9099144
> c.mat1[2,2]/sum(c.mat1[,2])   #sensitivity
[1] 0.8891129
> c.mat1[1,1]/sum(c.mat1[,1])   #specificity
[1] 0.9306533
```

```
     0   1
  0 926 110
  1  69 882
>
> sum(diag(c.mat33))/sum(c.mat33) #accuracy
[1] 0.9099144
> c.mat33[2,2]/sum(c.mat33[,2])   #sensitivity
[1] 0.8891129
> c.mat33[1,1]/sum(c.mat33[,1])   #specificity
[1] 0.9306533
```

Model performance metrics for two models on the training dataset. Left: Full model with 3 levels. Right: AIC model with 3 explanatory variables and 6 levels.

```
     0   1
  0 604  86
  1  49 587
>
> sum(diag(c.mat11))/sum(c.mat11) #accuracy
[1] 0.89819
> c.mat11[2,2]/sum(c.mat11[,2])   #sensitivity
[1] 0.872214
> c.mat11[1,1]/sum(c.mat11[,1])   #specificity
[1] 0.9249617
```

```
     0   1
  0 617  75
  1  36 598
>
> sum(diag(c.mat331))/sum(c.mat331) #accuracy
[1] 0.9162896
> c.mat331[2,2]/sum(c.mat331[,2])   #sensitivity
[1] 0.8885587
> c.mat331[1,1]/sum(c.mat331[,1])   #specificity
[1] 0.9448698
```

Model performance metrics for two models on the validation dataset. Left: Full model with 3 levels. Right: AIC model with 3 explanatory variables and 6 levels.

While there was not a great deal of variation in the confusion matrices when constructing different neural network models, the difference visible here is not insignificant. The model is almost 2% more accurate on the validation dataset with *fewer* predictors. In general adjusting the number of hidden layers did not have a drastic effect on the performance of the models, though that is with the limitation that no more than 9 layers were able to run on the equipment available.

Neural Networks especially did not have problems with overfitting of the data to the training dataset. Removing Elevation Gain tended to reduce sensitivity, although not necessarily overall accuracy. The tendency toward false positive bias existed in all neural network models just like in normal logistic regression. Ultimately, the "best" reduced model loses so much explanatory value in terms of what can actually be measured and used by trail and land management.

4) Classification Tree Analysis

While the full model using a basic classification tree method was very well performing once again, with Classification Tree methods the best model came from bagging. In general classification trees had the best performance out of any models produced in the analysis of this dataset; a random forest model had an overall accuracy of 97% on the training dataset, but degraded to 91% on the validation dataset (which was very good, but was indicative of overfitting). Boosting and bagging both showed virtually no signs of overfitting, and choosing between them came down to a discretionary choice based on how the different classification algorithms function.

```
predicted.class   0   1
              0 918 105
              1  82 882
> sum(diag(boostt2))/sum(boostt2) #accuracy
[1] 0.9058883
> boostt2[2,2]/sum(boostt2[,2])   #sensitivity
[1] 0.893617
> boostt2[1,1]/sum(boostt2[,1])   #specificity
[1] 0.918
```

```
predicted.class   0   1
              0 585  63
              1  63 615
> sum(diag(boostt1))/sum(boostt1) #accuracy
[1] 0.9049774
> boostt1[2,2]/sum(boostt1[,2])   #sensitivity
[1] 0.9070796
> boostt1[1,1]/sum(boostt1[,1])   #specificity
[1] 0.9027778
```

Confusion Matrix, Model Accuracy, Sensitivity, and Specificity of the Boosted Tree model for the full model. Left: Training, Right: Validation

```
                Observed Class
Predicted Class   0   1
              0 911 113
              1  89 874
> sum(diag(bagg1))/sum(bagg1) #accuracy
[1] 0.8983392
> bagg1[2,2]/sum(bagg1[,2])   #sensitivity
[1] 0.8855117
> bagg1[1,1]/sum(bagg1[,1])   #specificity
[1] 0.911
```

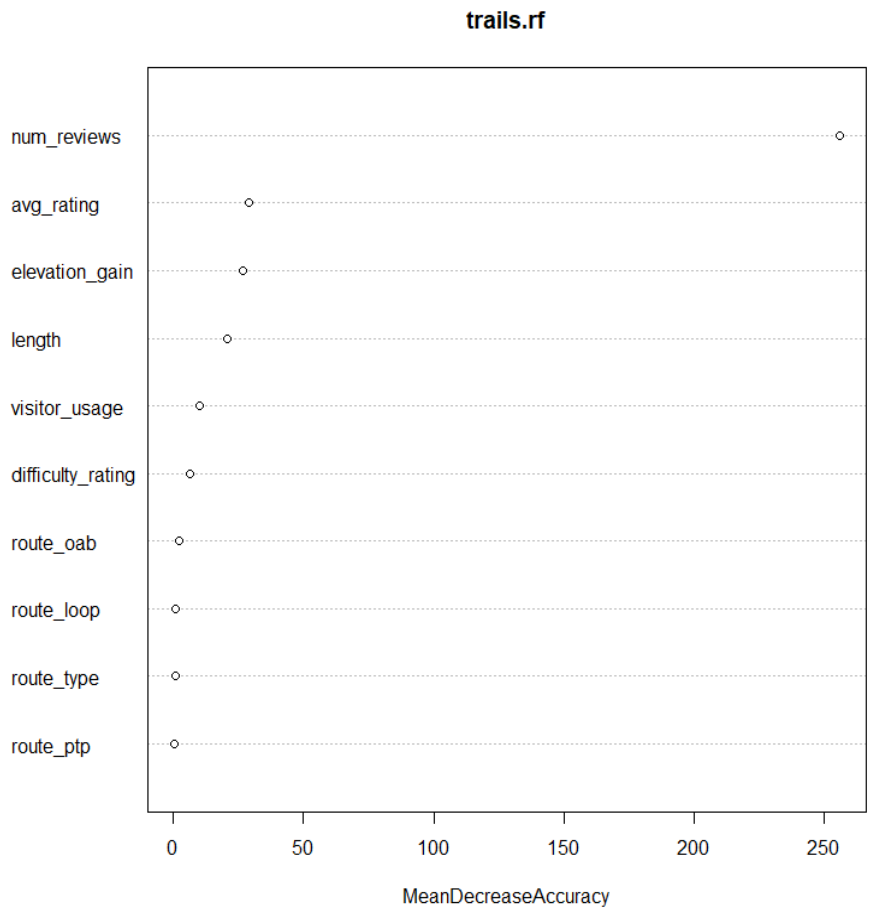
```
                Observed Class
Predicted Class   0   1
              0 586  67
              1  62 611
> sum(diag(bagg2))/sum(bagg2) #accuracy
[1] 0.9027149
> bagg2[2,2]/sum(bagg2[,2])   #sensitivity
[1] 0.9011799
> bagg2[1,1]/sum(bagg2[,1])   #specificity
[1] 0.904321
```

Confusion Matrix, Model Accuracy, Sensitivity, and Specificity of the Bagged Tree model for the full model. Left: Training, Right: Validation

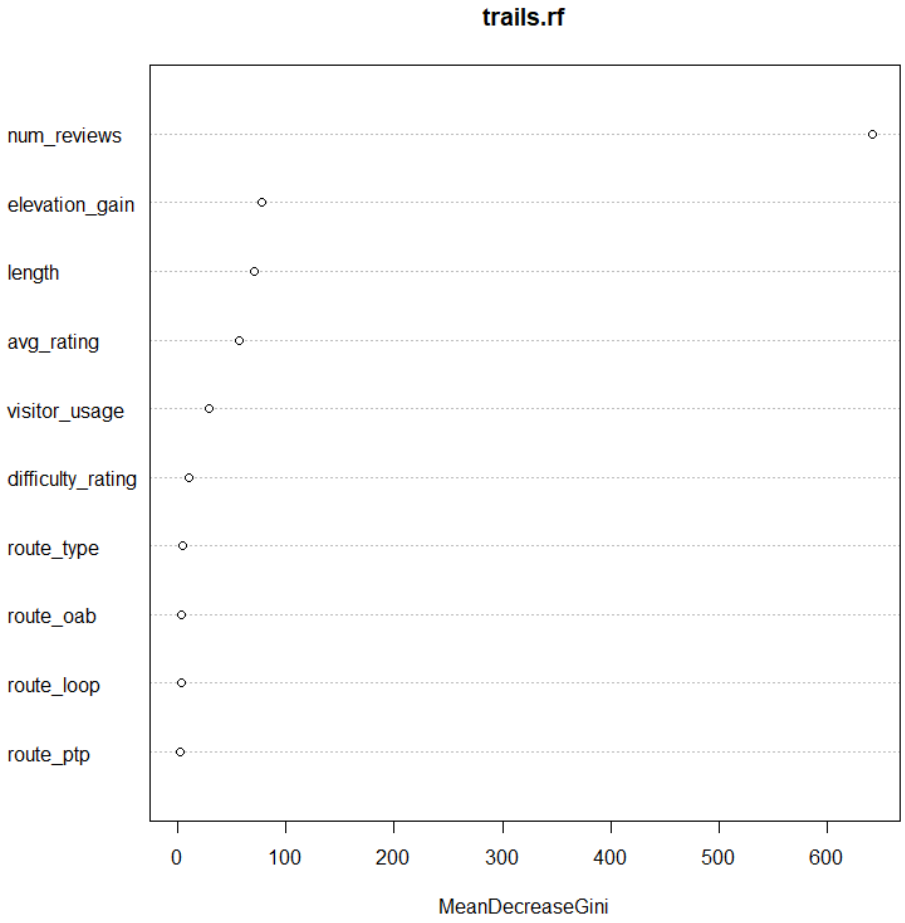
The performance of the two models is very similar and close, at least with the validation dataset. Inexplicably the bagging model performed slightly worse on the training dataset, but both are within half a percent of each other in every performance metric on the validation dataset. With 90% accuracy and virtually no reduction in accuracy between the training and validation datasets, the bootstrapped methods for classification trees perform the best of any other method explored, while retaining the full explanatory value of every variable in the original dataset.

5) Summary and Conclusion

The variable importance charts were particularly salient in the classification tree analysis when viewed through the lens of issues with multicollinearity encountered in logistic regression and neural network analysis.



Variable Important chart for Full Random Forest Model for Model Accuracy



Variable Important chart for Full Random Forest Model for Gini Index

In logistic regression analysis, removing all variables but elevation gain, average rating, and number of reviews had great model performance and an improved AIC to the full model. In Neural Network analysis, a model with the same variables performed better in all performance metrics than the boosted and bagged models. This was observed again in a classification tree model with only Average Rating and Number of Reviews, which had almost identical performance to the bagged and boosted models selected as a result of that analysis. In all forms of classification modelling, it was possible to make an either adequately or exceptionally performing model without the use of *any* physical attributes of trails as predictors. In the Accuracy variable importance chart, those two variables, Number of Reviews, and Average Rating, are the most important variables for training the overall model accuracy.

Especially in the case of classification tree analysis, without bootstrapping, including physical trail attributes increases explanatory value, but also massively increases the amount of error in the model and is detrimental to the model's ability to apply to novel datasets. The full model is still always preferable because conceptually and in application that makes sense. The class purity chart for variable importance from Random Forest shows that Elevation Gain *and Length* are both important in determining class purity, moreso than Average Rating. This is in line with a phenomenon observed in Logistic Regression: there are big differences in the range and means of the length and elevation gain between popular and unpopular trails:

```
> sum(trails.df$elevation_gain[trails.df$popularity==0])
[1] 1227548
> sum(trails.df$elevation_gain[trails.df$popularity==1])
[1] 898755.2
> sum(trails.df$elevation_gain[trails.df$popularity==1])-sum(trails.df$elevation_gain[trails.df$popularity==0])
[1] -328792.6

> mean(trails.df$elevation_gain[trails.df$popularity==0]) #1227548
[1] 744.8713
> mean(trails.df$elevation_gain[trails.df$popularity==1]) #898755.2
[1] 539.7929
> mean(trails.df$elevation_gain[trails.df$popularity==1])-mean(trails.df$elevation_gain[trails.df$popularity==0])
[1] -205.0783
```

The Sum of all elevation gain in Unpopular (0) and Popular (1) Trails followed by the difference (negative), and the mean elevation gain for each and their difference in the same order

Unpopular trails on average have about 205m more elevation, and there is a total of 328,793m of elevation more within that group, even though there are 7 fewer unpopular trails than there are popular. While the number of reviews on a given trail is an incredibly important predictor variable for whether or not a trail is popular, if a trail has high elevation gain it is more likely to be unpopular.

There are noticeable differences in at least those two physically measurable characteristics between popular and unpopular trails as defined by this analysis, and there are some immediate pragmatic explanations. The physical characteristics of a trail are not necessarily a reflection of how easy it is to get to that trail, and most people who use trails are probably not dedicated trail users (dedicated hikers, mountain bike riders, horseback riders). And there may be

differences between the type of users who leave reviews on AllTrails and those who do not that are not captured in this dataset.

A more parsimonious model is preferable, but parsimony is not the ultimate arbiter of what makes a "good" model. The models explored perform so well without so many predictors that ought to be important conceptually that, even though it would make sense to still include them and implement this model out in the world on new data, is probably a problem endemic with this dataset and the nature of the outcome variable being predicted. "Popularity" as we are modelling is not *actually Popularity*, i.e. the common concept that something can be popular. That actual concept would be very useful to natural resource and land managers; resources that see more use need to be constructed to withstand and accommodate that use, and require more and different kinds of maintenance, and planning and construction before that use ever occurs. Popularity is a calculated variable that AllTrails no longer measures or displays as information for trails on its website. While there seem to be strong, statistically significant relationships between popularity, some behaviors of trail users, and the attributes of the trails they use, in all likelihood "popularity" is mostly related to what trails AllTrails website/app users engage the most in, and is not what is idealized as "popularity" as would be useful to natural resource use stakeholders.

The fact of the matter is that some trails simply are more popular and subsequently receive more traffic and require more management. The amount of foot traffic to Old Faithful cannot be graphed by how long the distance is to the viewing point from the parking lot. The best views of the Grand Canyon are not the best because the trail they are seen from is a point to point trail or a loop trail. In the eyes of this dataset, it seems that popular trails are popular…because they are popular. While one does have to gain elevation to see magnificent views, or traverse longer trails to see more of something as large as the Grand Canyon, the presence of peak experiences, exceptional beauty, and notoriety are not measured in the variables. It is likely that those factors, along with

others such as seasons hiking is even possible (e.g. Glacier National Park has limited seasonality), typical weather conditions, how accessible a trail or trailhead is (e.g. a trail contained completely within Great Smoky Mountains National Park miles from any parking lot or access may be incredible, but unpopular because of accessibility).