In this part of the course project, you will build classification trees. This part continues the scenario from Parts One and Two, as it uses the same modified version of the human resources data set available on the Kaggle website. Use the HRdata4groups.csv data set to predict each individual's performance (Performance Score ID) using classification trees. Complete your analysis in R using the RStudio instance on the previous page in Canvas, then return to this page to answer project part questions and carefully explain what you discovered.

**Predict Individual Performance**

The data set is a modification of the data set available at https://www.kaggle.com/rhuebner/human-resources-data-set. We have synthesized values for Mechanical Aptitude and Verbal Aptitude to create the analyses in this project.

Below is an explanation of the data:

* **Performance Score ID** — A scale from 1 to 4, where 1 = Performance Improvement Plan (PIP), 2 = Needs Improvement, 3 = Fully Meets Expectation, 4 = Exceeds Expectation.
* **EmpSatisfaction** — A basic satisfaction score between 1 and 5 reported on a recent employee satisfaction survey integer.
* **Mechanical Aptitude** — A continuous value between 0 and 100, with larger numbers indicating higher aptitude.
* **Verbal Aptitude** — A continuous value between 0 and 100, with larger numbers indicating higher aptitude.
* **CollapseScore**— Reduction of Performance Score to two categories: high that equals 1 and low that equals 0.
* **PayRate** — Salary of the employee.
* **Age** — Age of employee.
* **JobTenure** — Years employed at the company.
* **EngagementSurvey** — Level of engagement of the employee.

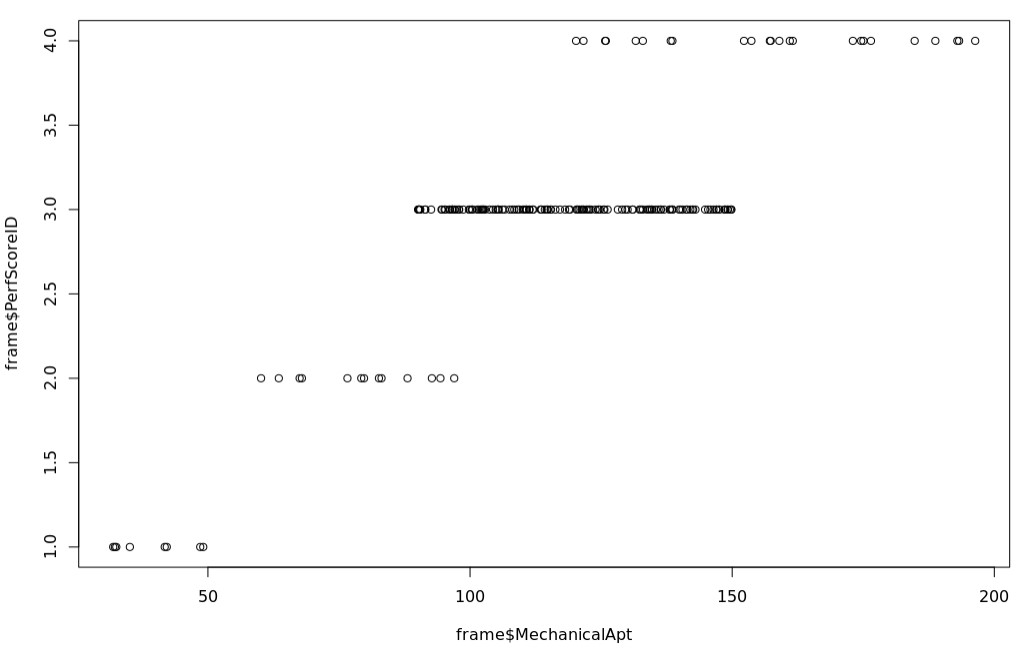
1. Explain the model you developed. It is sufficient to use the function ctree() in R to accomplish this in the style of the Codio exercise Practice: Building a Classification Tree in R — Small Example.

Model Formula:

PerfScoreID ~ MechanicalApt

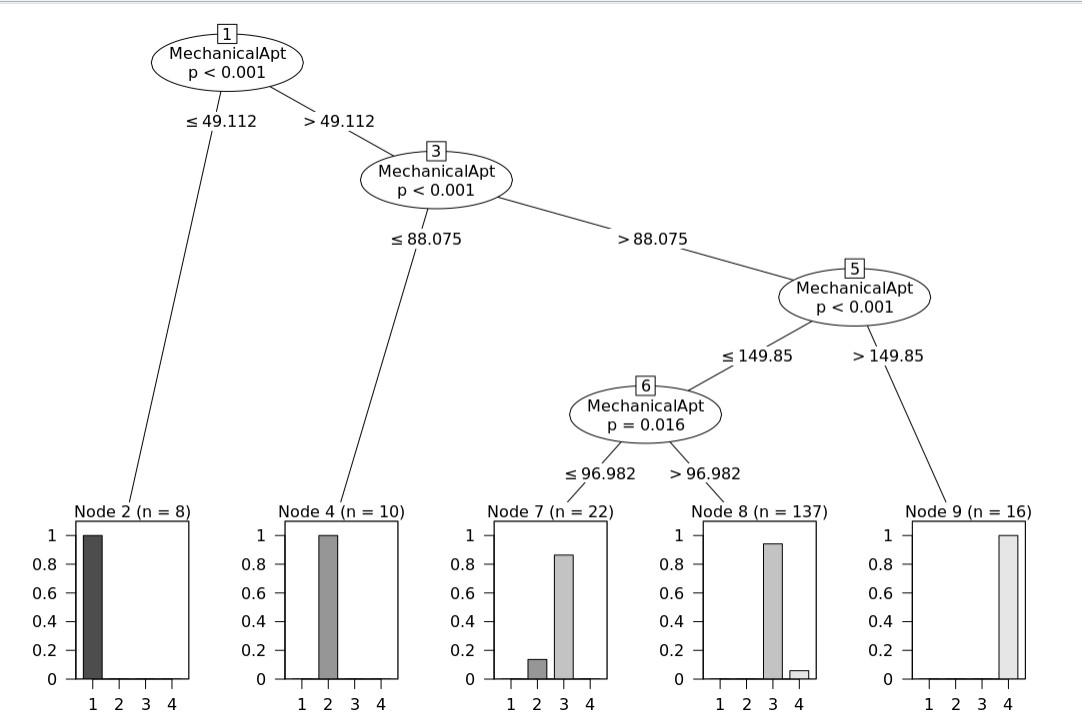
The model I developed uses only Mechanical Aptitude as the independent variable for classification tree predictions of Performance Score ID for each individual. This is because every other variable had a very weak correlation with Performance Score ID or would have caused collinearity with Mechanical Aptitude if included (Verbal Aptitude). Even further when trying to run with all variables included the classification tree only used Mechanical Aptitude. To have the model run as a classification tree not a regression tree the Performance Score ID was made into a factor.

Scatter Plot:



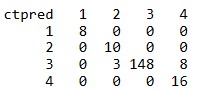
Scatter plot shows that Mechanical Aptitude allows for good separation between performance score IDs.

Classification Tree:



This shows how the model is able to predict performance score ID based upon 4 levels of classification splits happening at Mechanical Aptitude values of 49.12, 88.075, 149.85, and 96.982. The last split at 96.982 is arguably not required because both sides of that split still give the same classification of 3.

Confusion Matrix:



Confusion matrix model predictions are represented in each row. Actual Performance Score ID are represented in each column. This shows that 3 individuals with a performance score ID of two are misclassified as three. This also shows that 8 individuals with a performance score ID of four are misclassified as three.

Code:

library(partykit)

frame<-read.csv("HRdata4groups.csv")

frame$PerfScoreID<- as.factor(frame$PerfScoreID)

ctout <- ctree(PerfScoreID ~ MechanicalApt, data=frame)

ctpred <- predict(ctout,frame)

mean(ctpred == frame$PerfScoreID) # 0.943

table <- table(ctpred, frame$PerfScoreID)

print(table)

plot(ctout)

1. Describe how well your model performs.

The model performs very well, accurately predicting the proper performance score ID with a 94.3% success rate. The model has perfect precision predicting Performance Score ID for classifications of one and three. Also three of the 5 leaf nodes are perfectly homogenous. The other two only have a 13% and an 8% chance of misclassification so they also have good purity. With the use of this classification tree there was a total information gain of 0.225, with a remaining Entropy of 0.896. However, a future user of this model needs to keep in mind the dataset is very unbalanced with the performance score ID of three making up 77% of the data. This unbalance tends to have classification trees focus their models onto these majority classifications. So, this model despite being 94.3% accurate overall, correctly classifies only 77% of performance score ID two and 66.67% of performance score ID four as seen in the confusion matrix. A next step that could happen, if there was more than one useful independent variable, would be mindfully sampling the Performance Score ID data down to a scale that is similar to the other Performance Score IDs in an effort to correct the imbalance and improve accuracy for these two classification groups. That said this model has an excellent recall and performs very well overall and can be used with the precision weaknesses in mind.

Separation of each performance score

Could have just had a depth of three

Level of purity

Drop in entropy