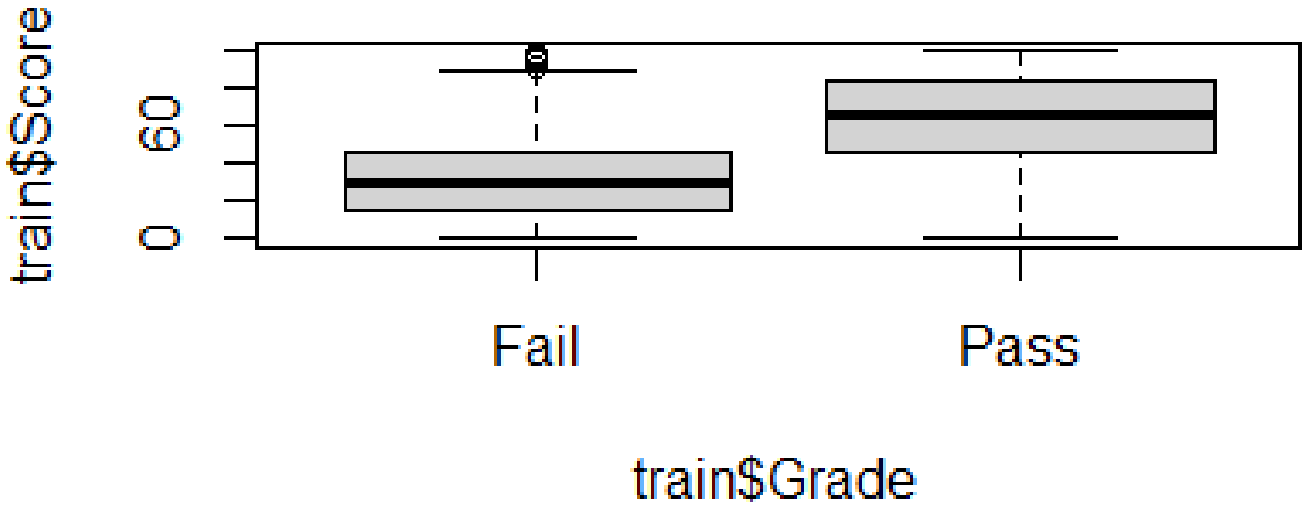
Algorithm & Prediction Model

By Jeremy Prasad

What are we dealing with?

First, I wanted to see exactly how much overlap there was concerning Grade and Score. There was a lot, but still with a very simple, basic prediction model, I got an error of 22%.

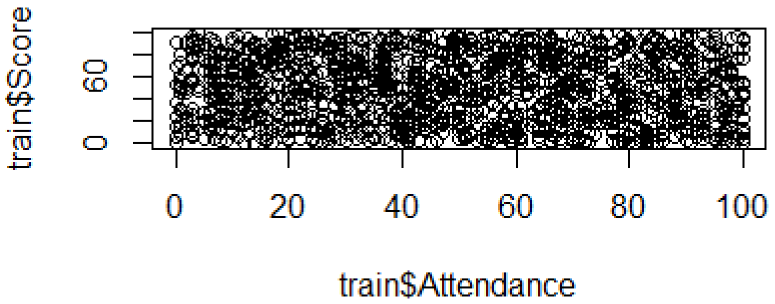
```
#Basic Algorithm
Initialize all grades to "Fail"
Set all grades >= 45 to
"Pass"

#Error
22.71%
```

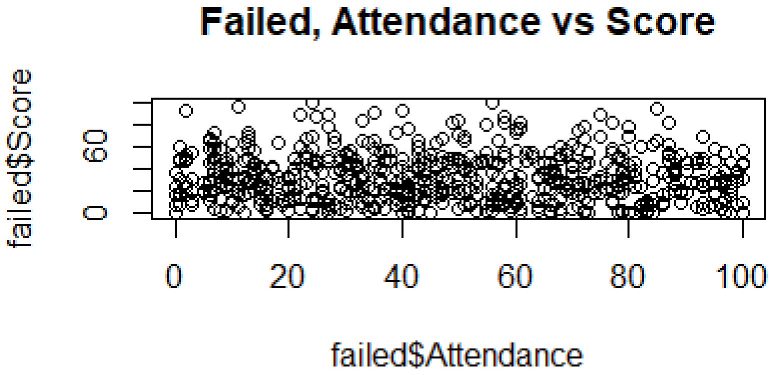


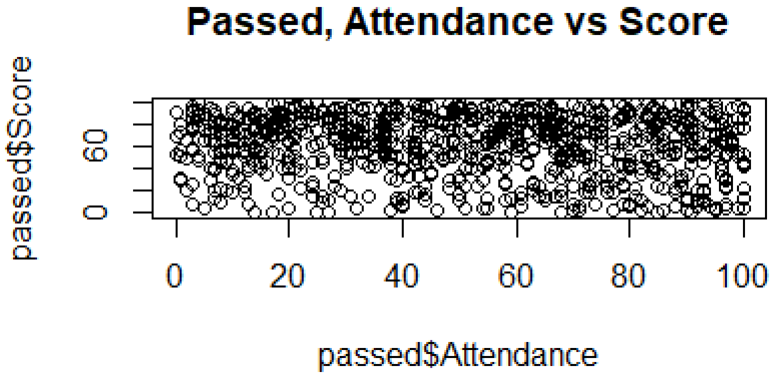
When testing our models, we want the set we're looking at to have boxplots as close as possible to this!

What matters?

On the surface, it seemed that Attendance doesn't matter. Still, it provides interesting graphs to help visualize score. This is the reason why I chose 45 as the cutoff score for the basic model (it worked pretty well for such a basic model!)



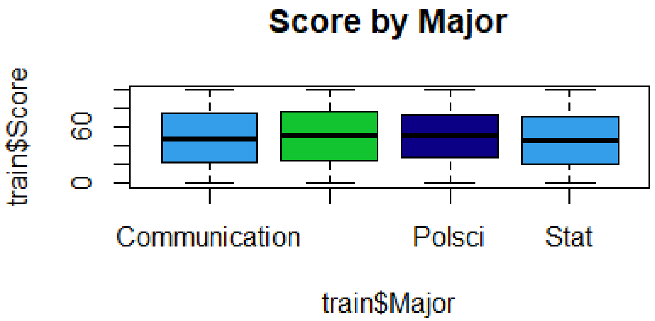


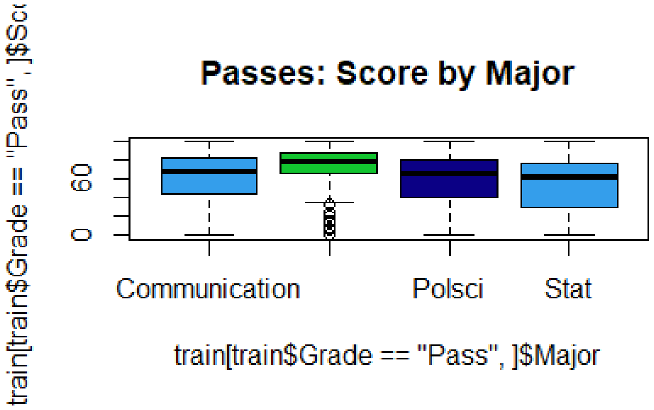


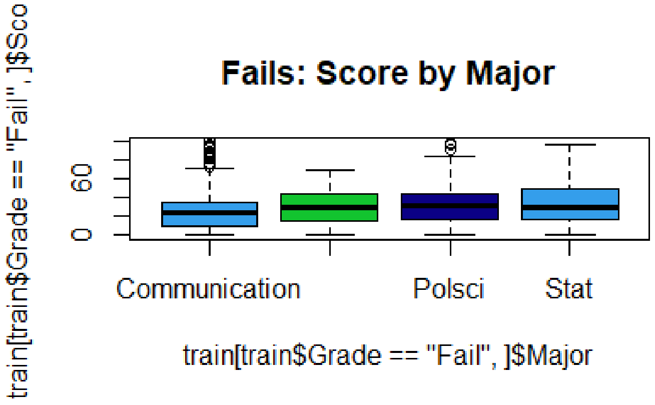
However, after I ran a permutation test on attendance (for Communication Majors), it was VERY LOW… To me, this indicates that Attendance definitely plays a role somewhere, but it is just hiding itself in these plots…

Organize the model?

This is one of the most important things in designing the model. After playing around for a while, for my final model, I decided that categorizing the data by Major would be the most effective way to predict as accurately as possible. Why? Every student is broken into one of four Majors. But one could argue the same for Seniority. However, I chose to group my model based off Major because each major has different levels of seniority, which might be easier to understand than the other way around (though still true). For example, you might find more correlations between Statistics majors than All Sophomores.



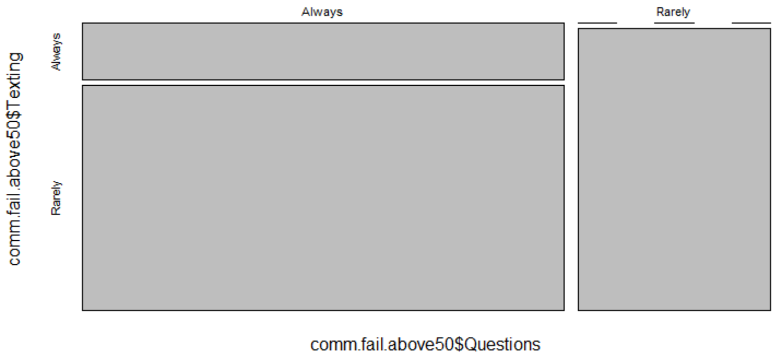


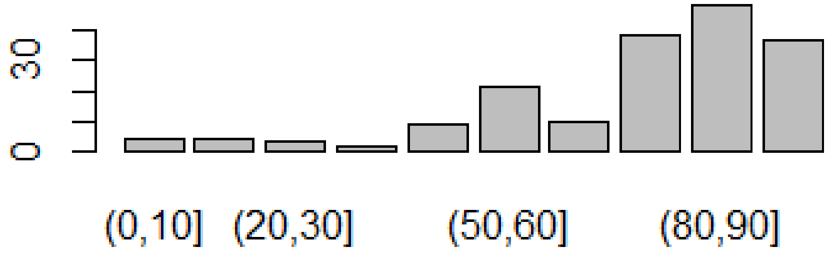


Finding Correlations

This is the hardest part of the assignment—especially in the early stages when I didn't formally organize my data by major. After hours of searching, trial, and error, one can begin to see meaningful patterns. For example, subsetting on CS majors allows us to see that nobody failed with a score >=70.

I think the key is to start general and narrow down by attribute. So, one might start by looking by Major, then by Major & Seniority, then by Major & Seniority & Questions, etc. After doing this, you are bound to find something!

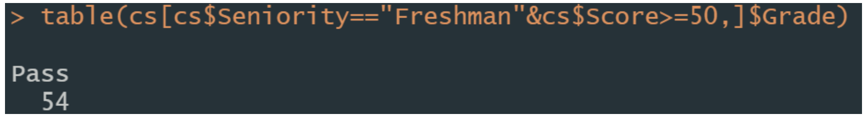




Examples of Findings

Because I'm a CS Major, I'll show you some findings I came across during my analysis on CS students from the dataset :)

By using the `table` function repeatedly on CS students of different Seniority and certain scores, we can see that



```
> table(cs[cs$Seniority=="Freshman"&cs$Score>=50,]$Grade)

Pass
  54
```

Freshmen CS students with a score >= 50 will pass

Sophomore CS students with score >= 50 will pass

Junior CS students with a score >=70 will pass

Senior CS students with a score >=70 will pass.

By narrowing down CS -> Seniority -> Score, I was able to find powerful correlations! A similar approach can be applied to other majors (obviously using different conditions).

This is also the method I used to determine that Attendance was actually important for other majors, despite how misleading the original scatter plot was!

Early Models

I won’t get too much into the details of how I arrived at my model (it was a lot of work & 1,000+ lines of code), but I will say I failed quite a lot. My initial model had an error rate of >32%. That was so hard to look at–it was even worse than the basic model. It seemed like every time I added more stuff, it got worse! So, after refining it, I was able to match the 22% given by the simple model, but I was hoping for more accuracy.

Around this time is when I formalized my search by looking exclusively at majors. This was key for this assignment because it is how I arrived at my final model.

Final Model

```
myprediction<-test
decision <- rep("Fail",nrow(myprediction))
decision[test$Score>=50] <- "Pass"
decision[test$Major=="Cs"&test$Seniority=="Freshman"&test$Score>=50] <- "Pass"
decision[test$Major=="Cs"&test$Seniority=="Freshman"&test$Score<50] <-"Fail"
decision[test$Major=="Cs"&test$Seniority=="Sophomore"&test$Score>=50]<-"Pass"
decision[test$Major=="Cs"&test$Seniority=="Sophomore"&test$Score<50]<-"Fail"
decision[test$Major=="Cs"&test$Seniority=="Junior"&test$Score>=70]<- "Pass"
decision[test$Major=="Cs"&test$Seniority=="Junior"&test$Score>=50&test$Score<70]<-"Fail"
decision[test$Major=="Cs"&test$Seniority=="Senior"&test$Score>=70]<- "Pass"
decision[test$Major=="Cs"&test$Seniority=="Senior"&test$Score>=50&test$Score<70]<-"Fail"
decision[test$Major=="Polsci"&Score>=50]<-"Pass"
decision[test$Major=="Polsci"&test$Score<50&test$Questions=="Rarely"] <- "Fail"
decision[test$Major=="Polsci"&test$Score<50&test$Questions=="Always"] <- "Pass"
decision[test$Major=="Communication"&test$Attendance<70&test$Score<40] <- "Fail"
decision[test$Major=="Communication"&test$Score>=40] <- "Pass"
decision[test$Major=="Communication"&test$Attendance>=70&test$Score<40&test$Texting=="Always"] <- "Pass"
decision[test$Major=="Communication"&test$Attendance>=70&test$Score<40&test$Texting=="Rarely"] <- "Fail"
decision[test$Major=="Stat"&test$Questions=="Always"&test$Score<60] <- "Fail"
decision[test$Major=="Stat"&test$Questions=="Always"&test$Score>=60] <- "Pass"
decision[test$Major=="Stat"&test$Questions=="Always"&test$Score<60] <- "Fail"
decision[test$Major=="Stat"&test$Questions=="Rarely"&test$Score>=35] <- "Pass"
decision[test$Major=="Stat"&test$Questions=="Rarely"&test$Score<35] <- "Fail"

test$Projection <-decision
```

Personal Testing error rate: ~14%
Kaggle error rate: ~14%

Conclusion

As you can see, the final model ended up being a bit complex, but it did pretty well.

Also, something to note is that continually shuffling the training & testing data set is important. You always want to make sure your data is fresh and accurately reflects the entire set.

```
boxplot(original$Score ~ original$Grade)
boxplot(train$Score ~ train$Grade)
boxplot(test$Score ~ test$MyProjection)
```

Running the first 2 commands will show the similarity of the original and training data sets, and running the last command shows how accurate your prediction model was.