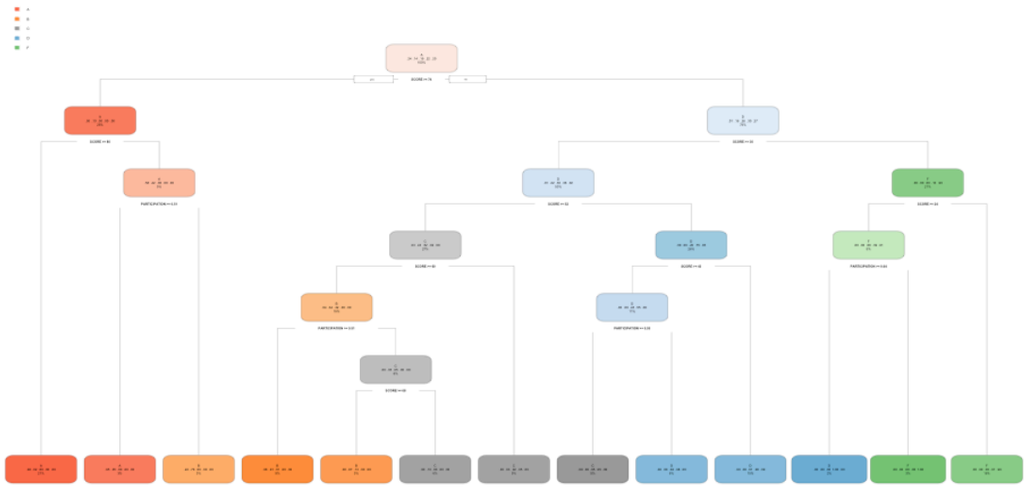
Prediction Challenge 02

Jack Lin

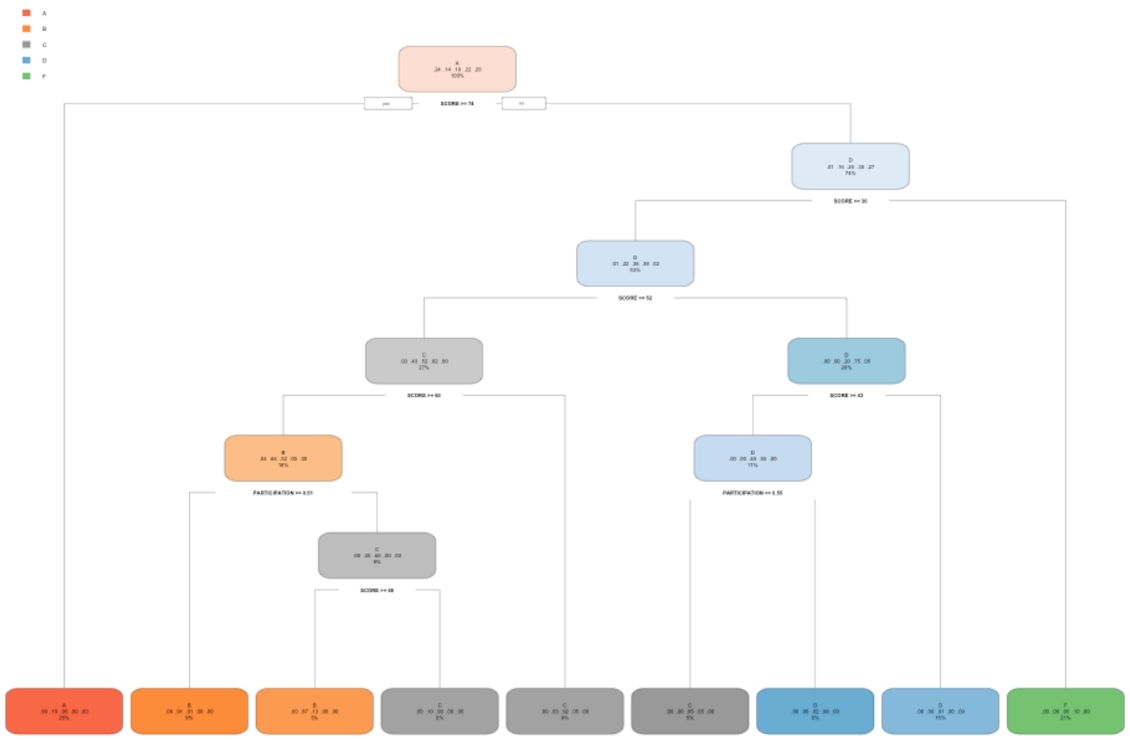
Model 1: Default minsplit and minbucket



R code:

```
> M2018_train <- read.csv("~/Documents/Rutgers/Data 101/Assignments/Prediction Challenge 1/M2018_train.csv", stringsAsFactors=FALSE)
>   View(M2018_train)
> moody.model1 <- rpart(GRADE~SCORE+ASKS_QUESTIONS+LEAVES_EARLY+PARTICIPATION, data=M2018_train)
> rpart.plot(moody.model1)
> predictedModel1 <- predict(moody.model1, newdata=M2018_train, type="class")
> model1.accuracy <- mean(predictedModel1 == M2018_train$GRADE)
> model1.accuracy
[1] 0.942789
> CrossValidation::cross_validate(M2018_train, moody.model1, 2, 0.8)
  accuracy_subset accuracy_all
1       0.9285714    0.9285714
2       0.9226190    0.9226190
```

Model 2: Minsplit = 50

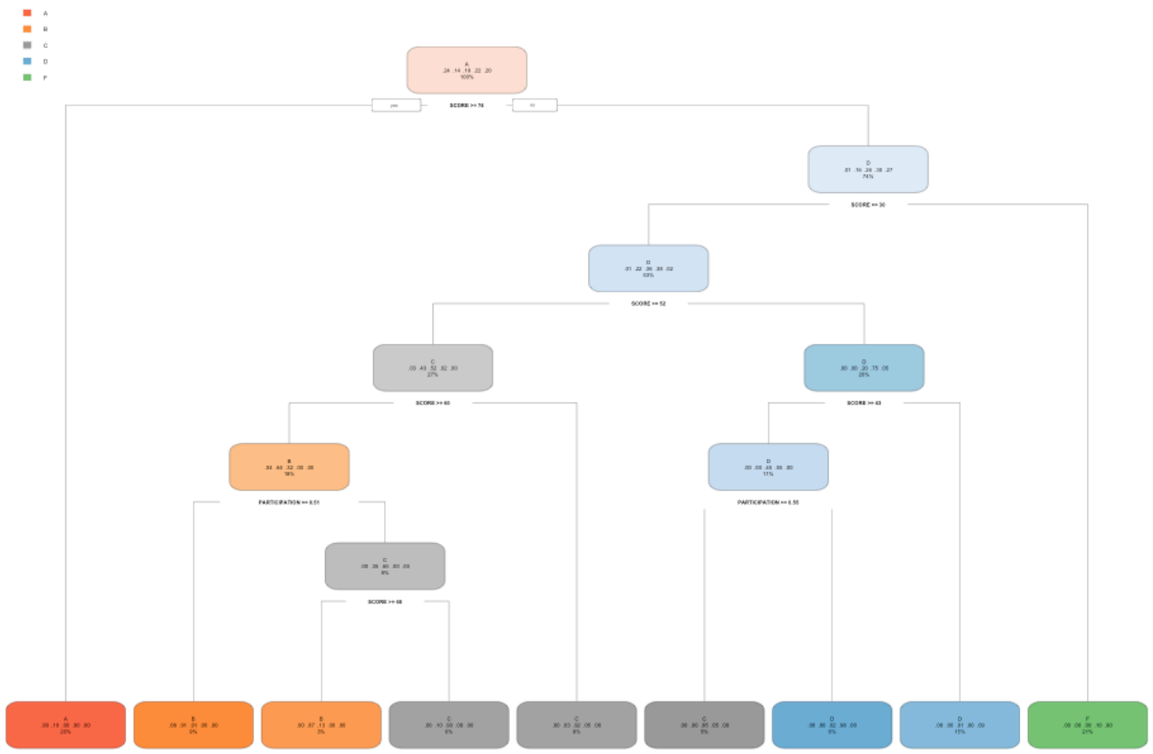


R code:

```
> moody.model2 <- rpart(GRADE~SCORE+ASKS_QUESTIONS+LEAVES_EARLY+PARTICIPATION, control = rpart.control(minsplit = 50), data =
M2018_train)
> rpart.plot(moody.model2)
> predictedModel2 <- predict(moody.model2, newdata=M2018_train, type="class")
> model2.accuracy <- mean(predictedModel2 == M2018_train$GRADE)
> model2.accuracy
[1] 0.9082241
> CrossValidation::cross_validate(M2018_train, moody.model2, 2, 0.8)
  accuracy_subset accuracy_all
1       0.8630952        0.875
2       0.8809524        0.875
```

Both accuracy test and cross validation demonstrate that Model 1 is favored so far.

Model 3: Minbucket = 20

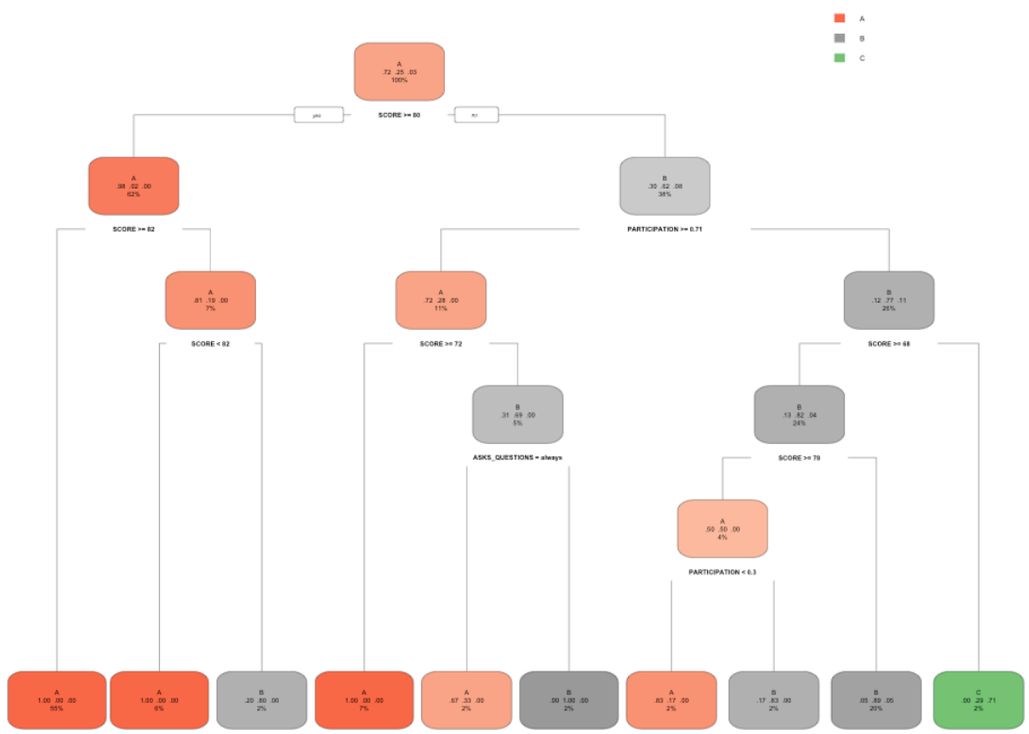


R code:

```
> moody.model3 <- rpart(GRADE~SCORE+ASKS_QUESTIONS+LEAVES_EARLY+PARTICIPATION, control = rpart.control(minbucket = 20), data =
M2018_train)
> rpart.plot(moody.model3)
> predictedModel3 <- predict(moody.model3, newdata=M2018_train, type="class")
> model3.accuracy <- mean(predictedModel3 == M2018_train$GRADE)
> model3.accuracy
[1] 0.9082241
> CrossValidation::cross_validate(M2018_train, moody.model3, 2, 0.8)
  accuracy_subset accuracy_all
1       0.8690476    0.8809524
2       0.8988095    0.9166667
```

Model 1 still seems the best so far…..What if we think outside of box and chop the data frame into two?

Top Half

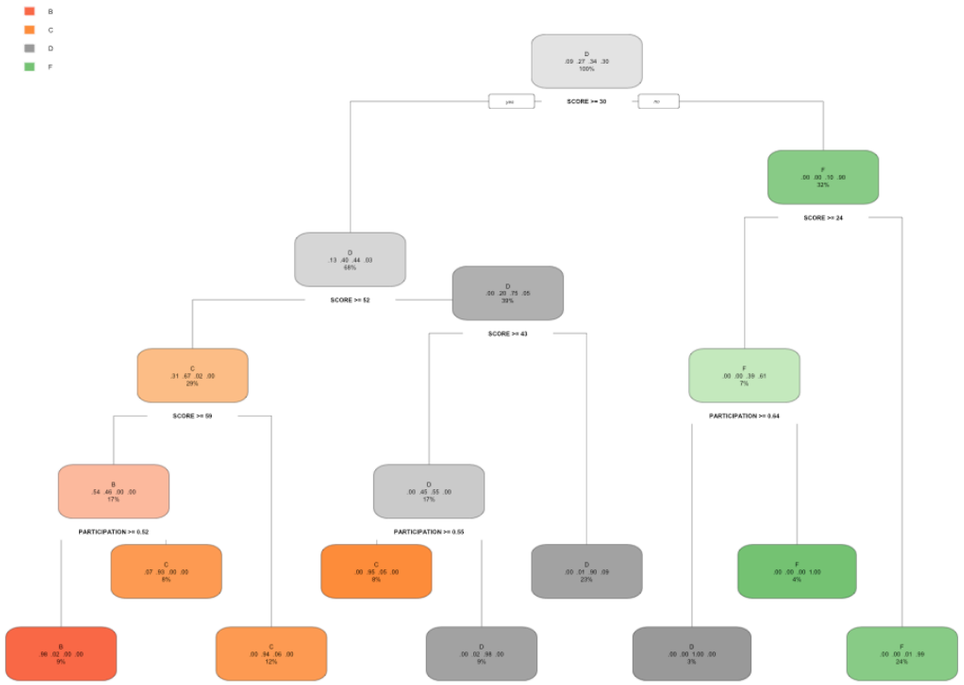


R code:

```
> TopHalf <- subset(M2018_train, M2018_train$SCORE>=66.99)
> TopHalf.tree <- rpart(GRADE~SCORE+ASKS_QUESTIONS+LEAVES_EARLY+PARTICIPATION, control = rpart.control(minbucket = 4), data =
TopHalf)
> rpart.plot(TopHalf.tree)
> predictedTopHalf <- predict(TopHalf.tree, newdata=TopHalf, type="class")
> TopHalf.accuracy <- mean(predictedTopHalf == TopHalf$GRADE)
> TopHalf.accuracy
[1] 0.9537367
> CrossValidation::cross_validate(TopHalf, TopHalf.tree, 2, 0.8)
  accuracy_subset accuracy_all
1       0.9122807    0.9122807
2       0.9473684    0.9298246
```

The cut-off point, 66.99, is the lowest score with an A in the training data set

Bottom Half



R code:

```
> BottomHalf <- subset(M2018_train, M2018_train$SCORE<66.99)
> BottomHalf.tree <- rpart(GRADE~SCORE+ASKS_QUESTIONS+LEAVES_EARLY+PARTICIPATION, control = rpart.control(minbucket = 4), data =
BottomHalf)
> rpart.plot(BottomHalf.tree)
> predictedBottomHalf <- predict(BottomHalf.tree, newdata=BottomHalf, type="class")
> BottomHalf.accuracy <- mean(predictedBottomHalf == BottomHalf$GRADE)
> BottomHalf.accuracy
[1] 0.9551971
> CrossValidation::cross_validate(BottomHalf, BottomHalf.tree, 2, 0.8)
  accuracy_subset accuracy_all
1       0.9553571    0.9553571
2       0.9642857    0.9642857
```

I decided to use this two-half approach in favor of Models 1–3 because of higher accuracy.

Implementation of My Prediction

R code:

```
> M2018_test_students <- read.csv("~/Documents/Rutgers/Data 101/Assignments/Prediction Challenge 1/M2018_test_students.csv",
stringsAsFactors=FALSE)
>   View(M2018_test_students)
> colnames(M2018_test_students)[3] <- “GRADE”
> Test.TopHalf <- subset(M2018_test_students, M2018_test_students$SCORE>=66.99)
> Test.BottomHalf <- subset(M2018_test_students, M2018_test_students$SCORE<66.99)
> predictedTopHalf.test <- predict(TopHalf.tree, newdata=Test.TopHalf, type=“class")
> Test.TopHalf$GRADE <- predictedTopHalf.test
> predictedBottomHalf.test <- predict(BottomHalf.tree, newdata=Test.BottomHalf, type="class")
> Test.BottomHalf$GRADE <- predictedBottomHalf.test
> Submission02 <- rbind(Test.TopHalf, Test.BottomHalf)
> Submission02 <- Submission02[,c(1,3)]
> write.csv(Submission02, ‘YuHonLinSubmission02.csv')
```

Results: 0.87596 on Kaggle public leaderboard. I might have overfit the testing data a little bit because I deliberately want “Ask Questions” to be a part of decision tree (hence I chopped my training data frame into two halves).