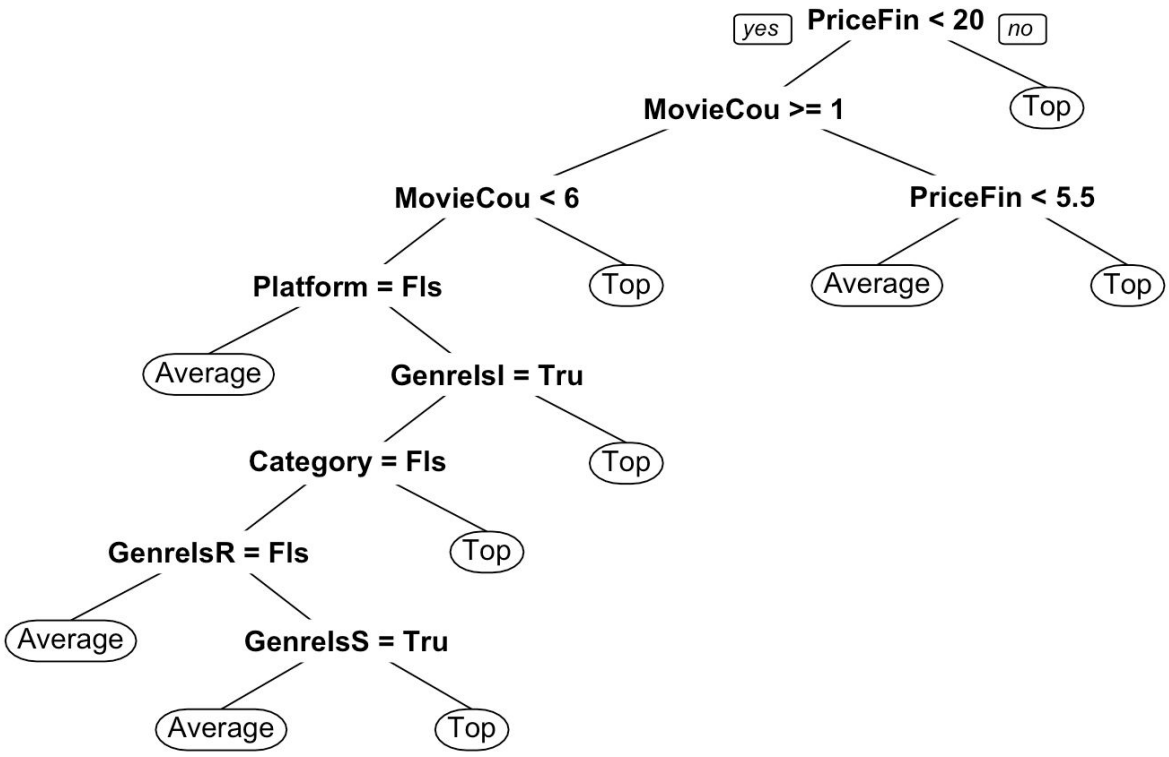
James Valles
Homework 2: DSC 441

Part 1

A.



B.

PriceFinal was the most important/predictive variable in my model.

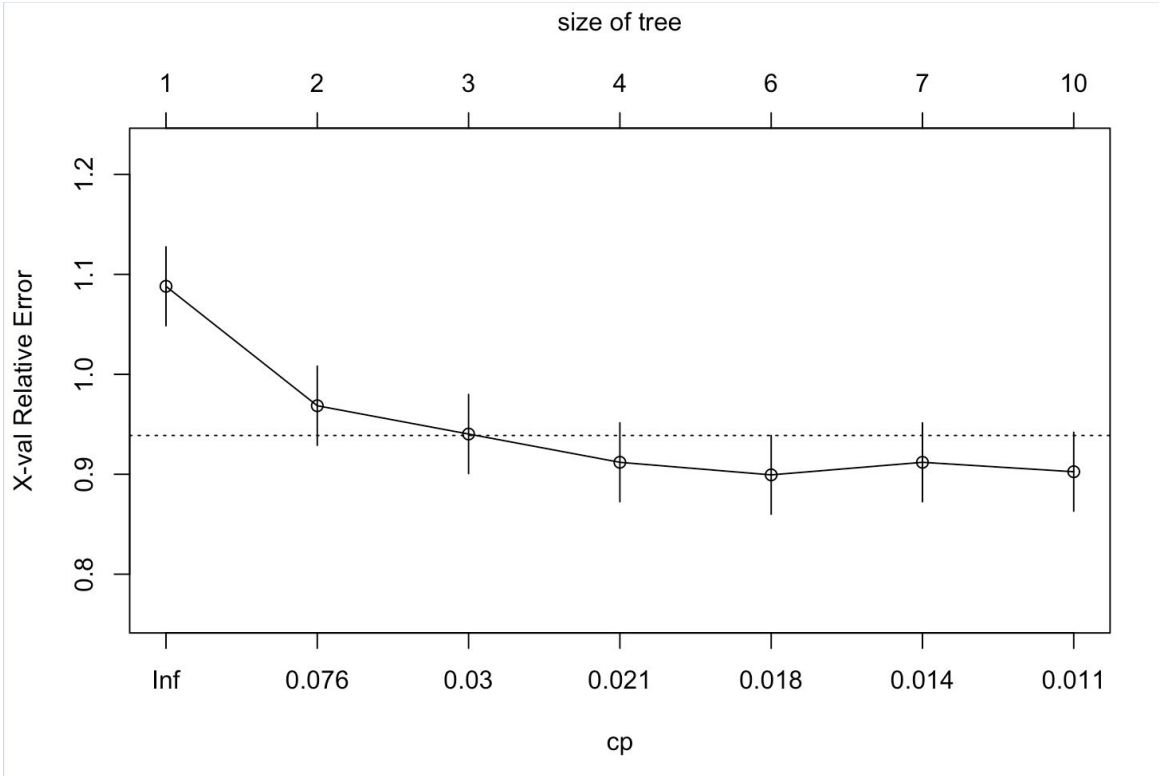
```
> tree$variable.importance
                PriceFinal                 MovieCount               GenreIsIndie
                10.6025912                  7.5222707                  5.3305541
              PlatformMac              MacReqsHaveMin           GenreIsStrategy
                 3.3753650                  3.2411175                  3.2088889
       LinuxReqsHaveMin CategoryIncludeLevelEditor              PlatformLinux
                 2.4661463                  2.1725338                  2.0904249
              GenreIsRPG           ScreenshotCount             MacReqsHaveRec
                 1.7780791                  1.5134488                  1.0548016
                  IsFree             PublisherCount        GenreIsFreeToPlay
                 0.8323040                  0.7586616                  0.7356454
         AchievementCount       CategorySinglePlayer             GenreIsSports
                 0.6768997                  0.5939116                  0.4330653
           DeveloperCount              CategoryCoop
                 0.3794024                  0.3169524
```

C.

The ideal complexity hyperparameter based on the table and graph would be **0.020440**.

```
> printcp(tree)

Classification tree:
rpart(formula = score ~ ., data = training, method = "class")

Variables actually used in tree construction:
[1] CategoryIncludeLevelEditor GenreIsIndie              GenreIsRPG
[4] GenreIsStrategy            MovieCount                PlatformMac
[7] PriceFinal

Root node error: 318/636 = 0.5

n= 636

        CP nsplit rel error  xerror     xstd
1 0.141509      0   1.00000 1.08805 0.039499
2 0.040881      1   0.85849 0.96855 0.039633
3 0.022013      2   0.81761 0.94025 0.039582
4 0.020440      3   0.79560 0.91195 0.039499
5 0.015723      5   0.75472 0.89937 0.039451
6 0.012579      6   0.73899 0.91195 0.039499
7 0.010000      9   0.70126 0.90252 0.039464
>
```

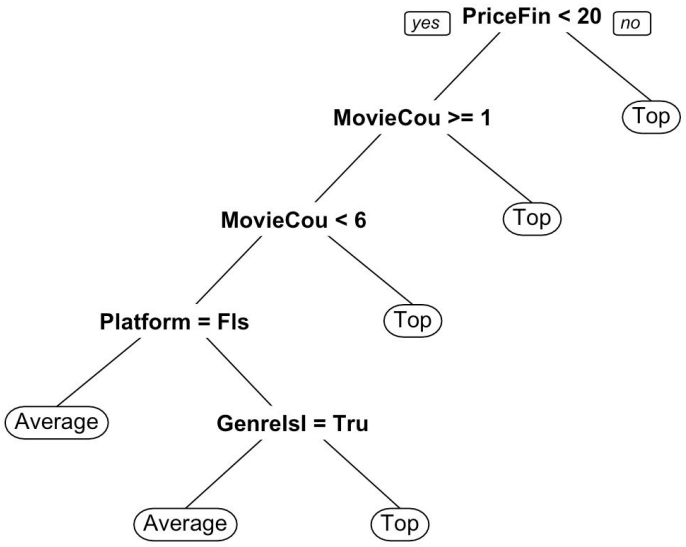
Leftmost is size of tree 4. So 4-1 is 3 nsplit and CP is 0.020440 according to table.



D.

```
> confusionMatrix(tree.pred, testing$score)
Confusion Matrix and Statistics

          Reference
Prediction Average Top
   Average      95  63
   Top          61  93

               Accuracy : 0.6026
                 95% CI : (0.5459, 0.6573)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.0001731

                  Kappa : 0.2051

 Mcnemar's Test P-Value : 0.9284440

            Sensitivity : 0.6090
            Specificity : 0.5962
         Pos Pred Value : 0.6013
         Neg Pred Value : 0.6039
             Prevalence : 0.5000
         Detection Rate : 0.3045
   Detection Prevalence : 0.5064
      Balanced Accuracy : 0.6026

       'Positive' Class : Average
```

E. Generate pruned tree using optimal complexity parameter of **0.020440** from question 1C.



F.

```
> confusionMatrix(tree.pruned, testing$score)
Confusion Matrix and Statistics

          Reference
Prediction Average Top
   Average      92  63
   Top          64  93

               Accuracy : 0.5929
                 95% CI : (0.5362, 0.648)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.0006078

                  Kappa : 0.1859

 Mcnemar's Test P-Value : 1.0000000

            Sensitivity : 0.5897
            Specificity : 0.5962
         Pos Pred Value : 0.5935
         Neg Pred Value : 0.5924
             Prevalence : 0.5000
         Detection Rate : 0.2949
   Detection Prevalence : 0.4968
      Balanced Accuracy : 0.5929

       'Positive' Class : Average
```

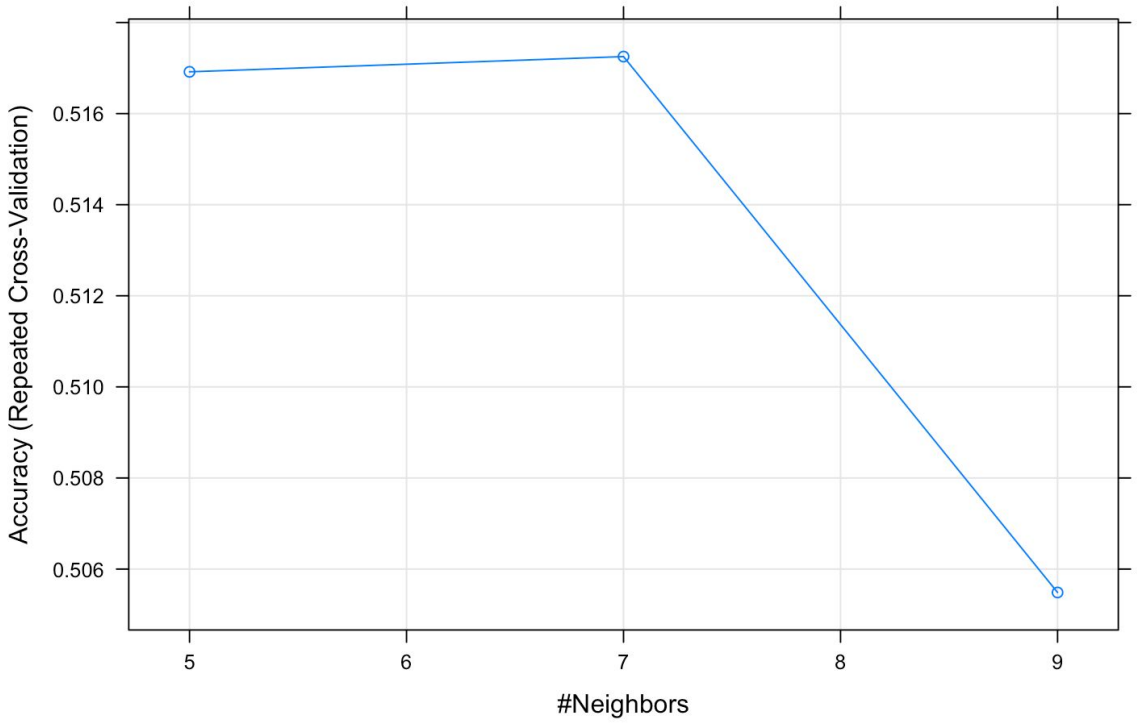
The unpruned tree has a depth of 7, while the pruned tree has a depth of 4. Because we are making the pruned tree more flexible by making it simpler, the pruned tree is slightly less accurate. The pruned tree had an accuracy score of 0.5929 compared to the unpruned tree 0.6026.

Part 2: KNN

A.

```
> varImp(knn1)
ROC curve variable importance

  only 20 most important variables shown (out of 58)

                                Importance
PriceFinal                          100.00
GenreIsIndie.False                   84.15
GenreIsIndie.True                    84.15
PlatformMac.False                    61.41
PlatformMac.True                     61.41
DeveloperCount                       56.59
MacReqsHaveMin.True                  47.76
MacReqsHaveMin.False                 47.76
PublisherCount                       47.41
CategoryMultiplayer.True             43.21
CategoryMultiplayer.False            43.21
PlatformLinux.False                  36.39
PlatformLinux.True                   36.39
GenreIsAdventure.True                31.84
GenreIsCasual.False                  31.84
GenreIsCasual.True                   31.84
GenreIsAdventure.False               31.84
MacReqsHaveRec.True                  27.29
MacReqsHaveRec.False                 27.29
CategoryIncludeLevelEditor.True      22.74
> 
```

B.



Build a model with the maximum training accuracy would take 7 neighbors as seen in the graph.

C.

```
> confusionMatrix(pred, testing$score)
Confusion Matrix and Statistics

          Reference
Prediction Average Top
   Average      82  58
   Top          60  84

               Accuracy : 0.5845
                 95% CI : (0.5248, 0.6424)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.002599

                  Kappa : 0.169

 Mcnemar's Test P-Value : 0.926652

            Sensitivity : 0.5775
            Specificity : 0.5915
         Pos Pred Value : 0.5857
         Neg Pred Value : 0.5833
             Prevalence : 0.5000
         Detection Rate : 0.2887
   Detection Prevalence : 0.4930
      Balanced Accuracy : 0.5845

       'Positive' Class : Average
```

D.

I would want to optimize my model for sensitivity because it would be worse to miss games that were top games, then it would be to make a bad guess and, in this case, write about a game that does not become a top hit. As discussed in the lecture, sensitivity is used when it costs more to miss something than to guess incorrectly. Whereas, specificity is concerned with the classifier correctly identifying when something is not the target class.