|  |
| --- |
|  |
| Mining Madness  Predicting the winner of the NCCAA Division I Men’s Basketball Tournament using performance data from the regular season. |
| |  |  |  | | --- | --- | --- | | Owen McCarron | 4/13/17 | CS699 – Spring 2017 | |

Table of Contents

[Introduction 4](#_Toc479860342)

[NCAA Division I Men's Basketball Tournament 4](#_Toc479860343)

[Tournament Participation and Seeding 4](#_Toc479860344)

[Tournament Schedule 4](#_Toc479860345)

[Data Mining Task and Data Set 5](#_Toc479860346)

[Data Set 5](#_Toc479860347)

[Data Source 5](#_Toc479860348)

[Format 5](#_Toc479860349)

[Data Preparation and Preprocessing 6](#_Toc479860350)

[Files Used 6](#_Toc479860351)

[Step 1: Create a single row per team per game in the regular season 6](#_Toc479860352)

[Step 2: Create new variables 6](#_Toc479860353)

[Step 3: Average the regular season game statistics by team and season 6](#_Toc479860354)

[Step 4: Aggregate Wins and Losses by team and season, create W/L percentage variables, and merge to regular season means 6](#_Toc479860355)

[Step 5: Parse the seed column to create individual ranking number and region assigned. Merge to regular season data. 7](#_Toc479860356)

[Step 6: Aggregate wins in the tournament to derive the champion each season. 7](#_Toc479860357)

[Step 7: Merge the tournament results with the regular season averages, 1 row per team per season. 7](#_Toc479860358)

[Attribute Selection 7](#_Toc479860359)

[1. CFS Subset evaluator 7](#_Toc479860360)

[2. OneRAttributeEval 7](#_Toc479860361)

[3. InfoGainAttributeEval 7](#_Toc479860362)

[4. Correlation Attribute Evaluation 7](#_Toc479860363)

[5. Self Selected 7](#_Toc479860364)

[Selection Algorithm Results 8](#_Toc479860365)

[Classification 9](#_Toc479860366)

[Classification Algorithm 9](#_Toc479860367)

[1. Naïve Bayes 9](#_Toc479860368)

[2. J48 - Decision Tree 9](#_Toc479860369)

[3. Logistic - Regression 9](#_Toc479860370)

[4. IBK - k-Nearest Neighbors 9](#_Toc479860371)

[Classification Results 10](#_Toc479860372)

[**1.** **CfsSubset** 10](#_Toc479860373)

[**2.** **OneRAttributeEval** 10](#_Toc479860374)

[**3.** **InfoGained** 11](#_Toc479860375)

[**4.** **Correlation** 11](#_Toc479860376)

[**5.** **Self-selected** 12](#_Toc479860377)

[Performance comparison 13](#_Toc479860378)

[Classifying Tournament Non-Winners (Positive) 13](#_Toc479860379)

[True Positives 14](#_Toc479860380)

[False Positives 14](#_Toc479860381)

[Classifying Tournament Winners (Negative) 15](#_Toc479860382)

[True Negatives 16](#_Toc479860383)

[False Negatives 16](#_Toc479860384)

[Accuracy and ROC Area 17](#_Toc479860385)

[Accuracy 18](#_Toc479860386)

[ROC Area 18](#_Toc479860387)

[Conclusion 19](#_Toc479860388)

[Appendix A – 2017 Bracket 20](#_Toc479860389)

[Appendix B: Kaggle Data File Descriptions 21](#_Toc479860390)

[Teams 21](#_Toc479860391)

[Seasons 21](#_Toc479860392)

[RegularSeasonCompactResults 22](#_Toc479860393)

[RegularSeasonDetailedResults 22](#_Toc479860394)

[TourneyCompactResults 23](#_Toc479860395)

[TourneyDetailedResults 23](#_Toc479860396)

[TourneySeeds 23](#_Toc479860397)

[TourneySlots 24](#_Toc479860398)

[References 25](#_Toc479860399)

# Introduction

## NCAA Division I Men's Basketball Tournament

Each year, the NCAA holds a tournament where sixty-eight men’s college basketball teams from across the country are invited to participate. The tournament, which takes place in March and April, results in the national champion for college basketball in that year.

### Tournament Participation and Seeding

As of the 2016/2017 season, there were 351 teams across 32 conference in Division I Basketball.

The 32 teams who win their conference in the regular season are automatically included in the tournament. The remaining 36 teams are determined by a selection committee made up of athletic directors and conference commissioners. The committee decides on these participants by assessing their performance during the regular season.

Once the 68 participants are determined, each team is ranked from 1-68 by the selection committee. The teams are then divided into four regions depending on their ranking; East, West, South, and Midwest. For instance, teams ranked 1-4 are spread across the four regions and ranked as number 1 in the region. Teams ranked 5-8 are spread across the four regions, ranked as number 2 in each region.

### Tournament Schedule

Seeding is performed at the start of the tournament only. The first round of the regional games is scheduled with the highest seeded team facing the lowest seeded team. Subsequent rounds are scheduled as if the higher seeds will win and the highest seed again faces the lowest seed. If a lower seed beats a higher seed, the schedule and seeding remains unchanged. For an example of the tournament schedule, review [Appendix A](#_Appendix_A_–).

# Data Mining Task and Data Set

The goal of this exercise is to predict the winner of the tournament by using both regular season game statistics and pre-tournament seeding.

The idea for this task was adapted from a 2017 Kaggle Competition[[1]](#footnote-1).

## Data Set

### Data Source

The data used in this analysis was made available as part of the Kaggle Competition. Kaggle references that Kenneth Massey[[2]](#footnote-2) provided most of the historical data.

### Format

There are six CSV files available for analysis which cover over 30 seasons worth of data (1985 – 2016). The information includes game statistics for each individual game that took place in those season (regular season and in the tournament). The files also define how teams are ranked pre-tournament. The files included are as follows:

|  |  |  |
| --- | --- | --- |
| File Name | Description | Details |
| Teams | Identifies the 364 college teams involved in the tournament between 1985-2016. | Tuples: 364  Attributes: 2 |
| Seasons | Identifies each of the four regions the tournament bracket is separated into in each season. | Tuples: 33  Attributes: 6 |
| Regular Season Results | Identifies each game played in each regular season (pre-tournament).  Attributes include:   * the teams that participated in each game * the date game played * the final score * general basketball statistics (i.e. Field Goals Made, Fouls, Rebounds) | Tuples: 71,241  Attributes: 34 |
| Tourney Results | Identifies each game played in the tournament and includes all attributes available in the Regular Season Results data set. | Tuples: 914  Attributes: 34 |
| Tourney Seeds | Identifies the pre-tournament seeding (or ranking) for each team. | Tuples: 2,084  Attributes: 3 |
| Tourney Slots | Defines the logic for how games are scheduled based on seeding. | Tuples: 2,050  Attributes: 4 |

The detailed descriptions of the fields that were provided by Kaggle are listed in [Appendix B](#_Appendix_B:_Kaggle).

# Data Preparation and Preprocessing

Several preprocessing steps were required to generate a single file that addressed the classifier, winning the tournament. The regular season details were combined with the tournament seeding and outcome of the tournament through R programming.

### Files Used

RS.data <- read.csv("RegularSeasonDetailedResults.csv")

TS.data <- read.csv("TourneySeeds.csv")

TDR.data <- read.csv("TourneyDetailedResults.csv")

### Step 1: Create a single row per team per game in the regular season

Both Result tables (Regular Season and Tourney) have one row per game. The statistics for both teams (Winner and Loser) are encompassed in the same row. For this exercise, we need to represent each team on it’s on row with their associated statistics and the result of the game (Win or Loss).

### Step 2: Create new variables

Based on the raw basketball statics, new variables for potential inclusion in classifier algorithms were created:

|  |  |
| --- | --- |
| Variable | Logic |
| Total Rebounds | RS.WL$tr <- RS.WL$or+RS.WL$dr |
| Point Differential | RS.WL$pt.diff <- RS.WL$score-RS.WL$score.opp |
| Field Goal Percentage | RS.WL$fg.per <- RS.WL$fgm/RS.WL$fga |
| 3 pt Field Goal Percentage | RS.WL$fg3.per <- RS.WL$fgm3/RS.WL$fga3 |
| Free Throw Percentage | RS.WL$ft.per <- RS.WL$ftm/RS.WL$fta |
| 2 pt Field Goals Made | RS.WL$fgm2 <- RS.WL$fgm - RS.WL$fgm3 |
| 2 pt Field Goals Attempted | RS.WL$fga2 <- RS.WL$fga - RS.WL$fga3 |
| 2 pt Field Goals Percentage | RS.WL$fg2.per <- RS.WL$fgm2/RS.WL$fga2 |

### Step 3: Average the regular season game statistics by team and season

To compare a team’s performance during the regular season to their results in the tournament that year, the basketball statistics for each game were aggregated by their means. The result is one row per team per season that has an average of each statistic.

### Step 4: Aggregate Wins and Losses by team and season, create W/L percentage variables, and merge to regular season means

In step 4, the Wins and Losses are counted by team and season. Through those counts, Win and Loss percentages were created. These percentages will be used instead of actual counts since some teams played more games than others during the season.

### Step 5: Parse the seed column to create individual ranking number and region assigned. Merge to regular season data.

Pull in the seed and region assigned for that team and merge to the appropriate season

### Step 6: Aggregate wins in the tournament to derive the champion each season.

Each team with 6 wins in the tournament is the winner.

### Step 7: Merge the tournament results with the regular season averages, 1 row per team per season.

Only teams selected for the tournament are represented. This file will be loaded into Weka

The data provided were complete so there were no issues with missing or invalid data.

# Attribute Selection

For attribute selection, the following algorithms were used in Weka:

### CFS Subset evaluator

Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.

### OneRAttributeEval

Evaluates the worth of an attribute by using the OneR classifier.

Criteria: Only those attributes with the highest ranking 98.491 were selected.

### InfoGainAttributeEval

Evaluates the worth of an attribute by measuring the information gain with respect to the class.

Criteria: Only those attributes ranked greater than zero were selected.

### Correlation Attribute Evaluation

Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class.

Criteria: Only those attributes ranked greater than 0.1 were selected.

### Self Selected

Attributes selected based on a review of the attribute selection algorithm and knowledge of the game of basketball.

Criteria: Selected those attributes that were common among the results of the first four algorithms. Additionally, selected free throw percentage and the rebounding statistics to understand their influence.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Selection Algorithm Results | | | | | | | | |
|  | CfsSubset | OneR | | InfoGained | | Correlation | | Self Selected |
| Criteria | All | 98.491 | | > 0 | | > 0.1 | |  |
| Season |  | 98.491 | Season | 0 |  | 0.00238 |  |  |
| team |  | 98.491 | team | 0 |  | 0.0553 |  |  |
| pts.RS |  | 98.491 | pts.RS | 0 |  | 0.14554 | pts.RS | pts.RS |
| pts.opp.RS |  | 98.491 | pts.opp.RS | 0 |  | 0.02333 |  |  |
| pt.diff.RS | pt.diff.RS | 98.276 |  | 0.03147 | pt.diff.RS | 0.20194 | pt.diff.RS | pt.diff.RS |
| fgm.RS | fgm.RS | 98.491 | fgm.RS | 0.01496 | fgm.RS | 0.14758 | fgm.RS | fgm.RS |
| fga.RS |  | 98.384 |  | 0 |  | 0.09409 |  |  |
| fg.per.RS |  | 98.384 |  | 0 |  | 0.10426 | fg.per.RS |  |
| fgm2.RS |  | 98.491 | fgm2.RS | 0 |  | 0.1297 | fgm2.RS |  |
| fga2.RS |  | 98.491 | fga2.RS | 0 |  | 0.09008 |  |  |
| fg2.per.RS |  | 98.491 | fg2.per.RS | 0 |  | 0.07856 |  |  |
| fgm3.RS |  | 98.491 | fgm3.RS | 0 |  | 0.01418 |  |  |
| fga3.RS |  | 98.491 | fga3.RS | 0 |  | 0.01018 |  |  |
| fg3.per.RS |  | 98.491 | fg3.per.RS | 0 |  | 0.0512 |  |  |
| ftm.RS |  | 98.491 | ftm.RS | 0 |  | 0.06664 |  |  |
| fta.RS |  | 98.491 | fta.RS | 0 |  | 0.04226 |  |  |
| ft.per.RS |  | 98.491 | ft.per.RS | 0 |  | 0.07042 |  | ft.per.RS |
| or.RS |  | 98.491 | or.RS | 0 |  | 0.08217 |  | or.RS |
| dr.RS |  | 98.491 | dr.RS | 0 |  | 0.10746 | dr.RS | dr.RS |
| tr.RS |  | 98.491 | tr.RS | 0 |  | 0.12279 | tr.RS | tr.RS |
| ast.RS |  | 98.491 | ast.RS | 0 |  | 0.09727 |  |  |
| to.RS |  | 98.491 | to.RS | 0 |  | 0.00756 |  |  |
| stl.RS |  | 98.491 | stl.RS | 0 |  | 0.06941 |  |  |
| blk.RS |  | 98.276 |  | 0 |  | 0.16385 | blk.RS |  |
| pf.RS |  | 98.491 | pf.RS | 0 |  | 0.07447 |  |  |
| wperc.RS |  | 98.491 | wperc.RS | 0.02323 | wperc.RS | 0.14956 | wperc.RS | wperc.RS |
| lperc.RS |  | 98.491 | lperc.RS | 0.02323 | lperc.RS | 0.14956 | lperc.RS |  |
| region.TS | region.TS | 98.491 | region.TS | 0.00416 | region.TS | 0.04065 |  |  |
| seed.num.TS | seed.num.TS | 98.491 | seed.num.TS | 0.02989 | seed.num.TS | 0.17755 | seed.num.TS | seed.num.TS |

# Classification

## Classification Algorithm

The following are the four classification algorithms selected for this exercise and run using Weka. The test options selected was “Cross-Validation” with Folds = 10.

### Naïve Bayes

Class for a Naive Bayes classifier using estimator classes.

### 

### J48 - Decision Tree

Class for generating a pruned or unpruned C4.

### Logistic - Regression

Class for building and using a multinomial logistic regression model with a ridge estimator.

### IBK - k-Nearest Neighbors

K-nearest neighbors classifier.

## Classification Results

### **CfsSubset**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naïve Bayes |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 877 | 37 | Instances | 885 | 43 |
| Actual (W) | 6 | 8 | % | 95.37 | 4.63 |
|  | | | | | | | |
| J48 |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 914 | 0 | Instances | 914 | 14 |
| Actual (W) | 14 | 0 | % | 98.4914 | 1.5086 |
|  | | | | | | | |
| Logistic |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 913 | 1 | Instances | 913 | 14 |
| Actual (W) | 14 | 0 | % | 98.3836 | 1.6164 |
|  | | | | | | | |
| IBK |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 904 | 10 | Instances | 904 | 24 |
| Actual (W) | 14 | 0 | % | 97.4138 | 2.5862 |

### **OneRAttributeEval**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naïve Bayes |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 848 | 66 | Instances | 858 | 70 |
| Actual (W) | 4 | 10 | % | 92.4569 | 7.5431 |
|  | | | | | | | |
| J48 |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 914 | 0 | Instances | 914 | 14 |
| Actual (W) | 14 | 0 | % | 98.4914 | 1.5086 |
|  | | | | | | | |
| Logistic |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 904 | 10 | Instances | 907 | 21 |
| Actual (W) | 11 | 3 | % | 97.7371 | 2.2629 |
|  | | | | | | | |
| IBK |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 901 | 13 | Instances | 903 | 25 |
| Actual (W) | 12 | 2 | % | 97.306 | 2.694 |

### **InfoGained**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naïve Bayes |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 848 | 66 | Instances | 858 | 70 |
| Actual (W) | 4 | 10 | % | 92.4569 | 7.5431 |
|  | | | | | | | |
| J48 |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 914 | 0 | Instances | 914 | 14 |
| Actual (W) | 14 | 0 | % | 98.4914 | 1.5086 |
|  | | | | | | | |
| Logistic |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 904 | 10 | Instances | 907 | 21 |
| Actual (W) | 11 | 3 | % | 97.7371 | 2.2629 |
|  | | | | | | | |
| IBK |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 910 | 13 | Instances | 903 | 25 |
| Actual (W) | 12 | 2 | % | 97.306 | 2.694 |

### **Correlation**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naïve Bayes |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 835 | 79 | Instances | 845 | 83 |
| Actual (W) | 4 | 10 | % | 91.056 | 8.944 |
|  | | | | | | | |
| J48 |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 908 | 6 | Instances | 908 | 20 |
| Actual (W) | 14 | 0 | % | 97.8448 | 2.1552 |
|  | | | | | | | |
| Logistic |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 909 | 5 | Instances | 909 | 19 |
| Actual (W) | 14 | 0 | % | 97.9526 | 2.0474 |
|  | | | | | | | |
| IBK |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 903 | 11 | Instances | 906 | 22 |
| Actual (W) | 11 | 3 | % | 97.6293 | 2.3707 |

### **Self-selected**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naïve Bayes |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 848 | 66 | Instances | 857 | 71 |
| Actual (W) | 5 | 9 | % | 92.3491 | 7.6509 |
|  | | | | | | | |
| J48 |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 908 | 6 | Instances | 908 | 20 |
| Actual (W) | 14 | 0 | % | 97.8448 | 2.1552 |
|  | | | | | | | |
| Logistic |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 911 | 3 | Instances | 911 | 17 |
| Actual (W) | 14 | 0 | % | 98.1681 | 1.8319 |
|  | | | | | | | |
| IBK |  | Predicted (L) | Predicted (W) |  |  | Classified Correctly | Classified Incorrectly |
| Actual (L) | 903 | 11 | Instances | 904 | 24 |
| Actual (W) | 13 | 1 | % | 97.4138 | 2.5862 |

# Performance comparison

The goal of this exercise is to predict whether or not a team will win the NCAA tournament based on their performance in the regular season and based on their seeding.

## Classifying Tournament Non-Winners (Positive)

The following are the performance information related to classifying teams that did not win the tournament.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | TP Rate | FP Rate |
| CFS | NaiveBays | 0.96 | 0.429 |
| CFS | Logistics | 0.999 | 1 |
| CFS | IBK | 0.989 | 1 |
| CFS | J48 | 1 | 1 |
|  |  |  |  |
| OneRAttributeEval | NaiveBays | 0.928 | 0.286 |
| OneRAttributeEval | Logistics | 0.989 | 0.786 |
| OneRAttributeEval | IBK | 0.986 | 0.857 |
| OneRAttributeEval | J48 | 1 | 1 |
|  |  |  |  |
| InfoGained | NaiveBays | 0.907 | 0.214 |
| InfoGained | Logistics | 0.998 | 1 |
| InfoGained | IBK | 0.98 | 1 |
| InfoGained | J48 | 1 | 1 |
|  |  |  |  |
| Correlation | NaiveBays | 0.914 | 0.286 |
| Correlation | Logistics | 0.995 | 1 |
| Correlation | IBK | 0.988 | 0.786 |
| Correlation | J48 | 0.993 | 1 |
|  |  |  |  |
| Self-selected | NaiveBays | 0.928 | 0.357 |
| Self-selected | Logistics | 0.997 | 1 |
| Self-selected | IBK | 0.988 | 0.929 |
| Self-selected | J48 | 0.993 | 1 |

## True Positives

The algorithms were all relatively accurate in predicting whether a team would not win the tournament. The J48 decision tree algorithm was the most effective in predicting losing teams in the tournament, averaging 99.7% TP rate across the attribute selections. The J48 algorithm was especially effective when combined with the CFS, OneRAttributeEval, and InfoGained attribute selections where it was able to identify 100% of the losing teams.

False Positives

Most of the algorithms incorrectly classified winning teams as losers. The algorithm that performed best in not falsely categorizing a Winner as a Loser was Naïve Bayes. Across the attribute selections, it categorized an average of 31.4% of the winning teams as losers.

## Classifying Tournament Winners (Negative)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | TP Rate | FP Rate |
| CFS | NaiveBays | 0.571 | 0.04 |
| CFS | Logistics | 0 | 0.001 |
| CFS | IBK | 0 | 0.011 |
| CFS | J48 | 0 | 0 |
|  |  |  |  |
| OneRAttributeEval | NaiveBays | 0.714 | 0.072 |
| OneRAttributeEval | Logistics | 0.214 | 0.011 |
| OneRAttributeEval | IBK | 0.143 | 0.014 |
| OneRAttributeEval | J48 | 0 | 0 |
|  |  |  |  |
| InfoGained | NaiveBays | 0.786 | 0.093 |
| InfoGained | Logistics | 0 | 0.002 |
| InfoGained | IBK | 0 | 0.02 |
| InfoGained | J48 | 0 | 0 |
|  |  |  |  |
| Correlation | NaiveBays | 0.714 | 0.086 |
| Correlation | Logistics | 0 | 0.005 |
| Correlation | IBK | 0.214 | 0.012 |
| Correlation | J48 | 0 | 0.007 |
|  |  |  |  |
| Self-selected | NaiveBays | 0.643 | 0.072 |
| Self-selected | Logistics | 0 | 0.003 |
| Self-selected | IBK | 0.071 | 0.012 |
| Self-selected | J48 | 0 | 0.007 |

## True Negatives

Only the Naïve Bays algorithm was effective in identifying True Negative or those teams that won the tournament. It was able to identify an average of 68.6% of winning teams correctly across the attribute selection. It was most effective in the InfoGained attribute selection with a 77% True Negative Rate.

False Negatives

While it was the most effective in identifying teams that won the tournament, the Naïve Bayes algorithm also had the highest rate of tournament losers incorrectly classified as winners at an average of 7% across the attribute selections.

## Accuracy and ROC Area

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Accuracy | ROC Area |
| CFS | NaiveBays | 95.37% | 90.30% |
| CFS | Logistics | 98.38% | 83.50% |
| CFS | IBK | 97.41% | 52.10% |
| CFS | J48 | 98.49% | 41.30% |
|  |  |  |  |
| OneRAttributeEval | NaiveBays | 92.46% | 89.10% |
| OneRAttributeEval | Logistics | 97.74% | 77.40% |
| OneRAttributeEval | IBK | 97.31% | 57.90% |
| OneRAttributeEval | J48 | 98.49% | 41.30% |
|  |  |  |  |
| InfoGained | NaiveBays | 92.46% | 90.90% |
| InfoGained | Logistics | 97.74% | 82.60% |
| InfoGained | IBK | 97.31% | 51.70% |
| InfoGained | J48 | 98.49% | 41.30% |
|  |  |  |  |
| Correlation | NaiveBays | 91.06% | 90.60% |
| Correlation | Logistics | 97.95% | 88.90% |
| Correlation | IBK | 97.63% | 60.80% |
| Correlation | J48 | 97.84% | 56.20% |
|  |  |  |  |
| Self-selected | NaiveBays | 92.35% | 90.40% |
| Self-selected | Logistics | 98.17% | 89.70% |
| Self-selected | IBK | 97.41% | 55.00% |
| Self-selected | J48 | 97.84% | 43.20% |

## Accuracy

The highest rate of accuracy belongs to the J48 Decision Tree algorithm, averaging 98.23% across the different attribute selections. The other algorithm also performed well in regard to accuracy. The worst performer for accuracy was the Naïve Bays algorithm.

## ROC Area

Comparing the ROC area for each algorithm, NaïveBays was the best performer averaging 90.26% across the attribute selections. Both the IBK and J48 algorithms are close to random guessing.

# Conclusion

Each season, there is only one winner among the 68 teams that participate in the NCAA tournament. That means 98.5% of teams will lose the tournament. If we randomly guessed that a team lost, chances are we would be correct. For that reason, it is difficult to read much into the success of the True Positive and accuracy rates of the classification algorithms.

This concept is illustrated by the ROC curve. While the K-Nearest Neighbor and Decision Tree algorithms have high accuracy rates, the ROC area show that these classifications are really close to random guessing.

The True Negative Rate and the ROC are better indicators of the algorithm performance for our stated goal, identifying a winner of the tournament. It is clear when analyzing these two metrics that Naïve Bayes is the most successful of the four algorithms at identifying a winner. It has vastly outperformed the other algorithms in identifying True Negatives (67%). The other algorithms combine to average 4%.

Now that we have identified the best classification algorithm, we can assess its performance among the different attribute selections:

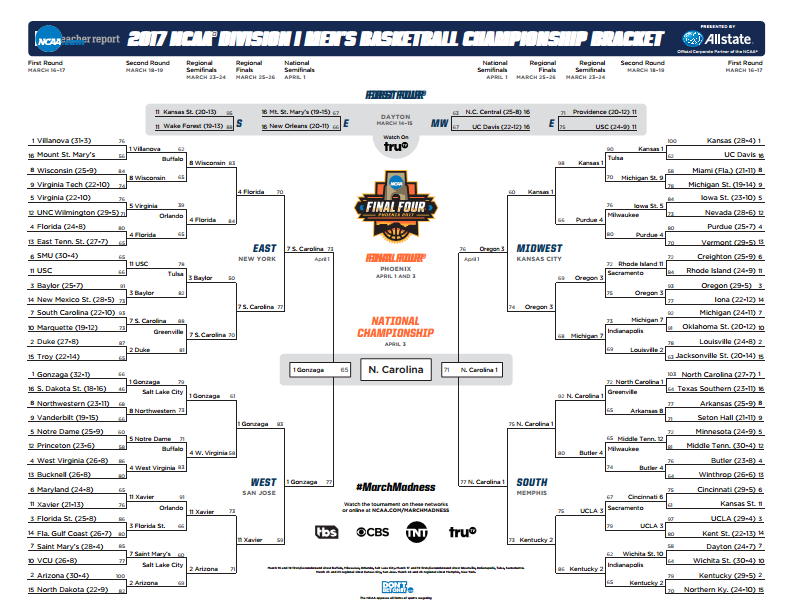
|  |  |  |
| --- | --- | --- |
| Attribute Selection | TP Rate | FP Rate |
| CFS | 0.571 | 0.04 |
| OneRAttributeEval | 0.714 | 0.072 |
| InfoGained | 0.786 | 0.093 |
| Correlation | 0.714 | 0.086 |
| Self-selected | 0.643 | 0.072 |

The best performer was the InfoGained attribute selection at 79% TP Rate. That selection consists of only 6 attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | OneRAttribute | Correlation |
| pt.diff.RS | Average of the difference in points scored to the points allowed to the opponent | No | Yes |
| fgm.RS | The number of shots made (2 and 3pts) | Yes | Yes |
| wperc.RS | The percent of games won during the season | Yes | Yes |
| lperc.RS | The percent of games lost during the season | Yes | Yes |
| region.TS | The region the team is assigned to pre-tournament | Yes | No |
| seed.num.TS | The seed the team is assigned to pre-tournament | Yes | Yes |

Many of these 6 attributes overlap with the other two best performing selections, OneRAttributeEval and Correlation. Reviewing these attributes, it’s evident that the two best indicators of whether a team may win the tournament is their scoring which leads to wins. This is represented by the point differential, field goals made, and win/loss percentage. Seeding is also important. The better seed you have, the easier your competition should be throughout the tournament.

# Appendix A – 2017 Bracket[[3]](#footnote-3)



# Appendix B: Kaggle Data File Descriptions

Below we describe the format and fields of the "essential" data files. Optional files may be added to the data while the competition is running. You can assume that we will provide the essential files for the current season. You should not assume that we will provide optional files for the current season.

## **Teams**

This file identifies the different college teams present in the dataset. Each team has a 4 digit id number.

## **Seasons**

This file identifies the different seasons included in the historical data, along with certain season-level properties.

* "season" - indicates the year in which the tournament was played
* "dayzero" - tells you the date corresponding to daynum=0 during that season. All game dates have been aligned upon a common scale so that the championship game of the final tournament is on daynum=154. Working backward, the national semifinals are always on daynum=152, the "play-in" games are on days 134/135, Selection Sunday is on day 132, and so on. All game data includes the day number in order to make it easier to perform date calculations. If you really want to know the exact date a game was played on, you can combine the game's "daynum" with the season's "dayzero". For instance, since day zero during the 2011-2012 season was 10/31/2011, if we know that the earliest regular season games that year were played on daynum=7, they were therefore played on 11/07/2011.
* "regionW/X/Y/Z" - by convention, the four regions in the final tournament are always named W, X, Y, and Z. Whichever region's name comes first alphabetically, that region will be Region W. And whichever Region plays against Region W in the national semifinals, that will be Region X. For the other two regions, whichever region's name comes first alphabetically, that region will be Region Y, and the other will be Region Z. This allows us to identify the regions and brackets in a standardized way in other files. For instance, during the 2012 tournament, the four regions were East, Midwest, South, and West. Being the first alphabetically, East becomes W. Since the East regional champion (Ohio State) played against the Midwest regional champion (Kansas) in the national semifinals, that makes Midwest be region X. For the other two (South and West), since South comes first alphabetically, that makes South Y and therefore West is Z. So for this season, the W/X/Y/Z are East,Midwest,South,West.

## **RegularSeasonCompactResults**

This file identifies the game-by-game results for 32 seasons of historical data, from 1985 to 2015. Each year, it includes all games played from daynum 0 through 132 (which by definition is "Selection Sunday," the day that tournament pairings are announced). Each row in the file represents a single game played.

* "season" - this is the year of the associated entry in seasons.csv (the year in which the final tournament occurs)
* "daynum" - this integer always ranges from 0 to 132, and tells you what day the game was played on. It represents an offset from the "dayzero" date in the "seasons.csv" file. For example, the first game in the file was daynum=20. Combined with the fact from the "season.csv" file that day zero was 10/29/1984, that means the first game was played 20 days later, or 11/18/1984. There are no teams that ever played more than one game on a given date, so you can use this fact if you need a unique key. In order to accomplish this uniqueness, we had to adjust one game's date. In March 2008, the SEC postseason tournament had to reschedule one game (Georgia-Kentucky) to a subsequent day, so Georgia had to actually play two games on the same day. In order to enforce this uniqueness, we moved the game date for the Georgia-Kentucky game back to its original date.
* "wteam" - this identifies the id number of the team that won the game, as listed in the "teams.csv" file. No matter whether the game was won by the home team or visiting team, "wteam" always identifies the winning team.
* "wscore" - this identifies the number of points scored by the winning team.
* "lteam" - this identifies the id number of the team that lost the game.
* "lscore" - this identifies the number of points scored by the losing team.
* "numot" - this indicates the number of overtime periods in the game, an integer 0 or higher.
* "wloc" - this identifies the "location" of the winning team. If the winning team was the home team, this value will be "H". If the winning team was the visiting team, this value will be "A". If it was played on a neutral court, then this value will be "N". Sometimes it is unclear whether the site should be considered neutral, since it is near one team's home court, or even on their court during a tournament, but for this determination we have simply used the Kenneth Massey data in its current state, where the "@" sign is either listed with the winning team, the losing team, or neither team.

## **RegularSeasonDetailedResults**

This file is a more detailed set of game results, covering seasons 2003-2016. This includes team-level total statistics for each game (total field goals attempted, offensive rebounds, etc.) The column names should be self-explanatory to basketball fans (as above, "w" or "l" refers to the winning or losing team):

* wfgm - field goals made
* wfga - field goals attempted
* wfgm3 - three pointers made
* wfga3 - three pointers attempted
* wftm - free throws made
* wfta - free throws attempted
* wor - offensive rebounds
* wdr - defensive rebounds
* wast - assists
* wto - turnovers
* wstl - steals
* wblk - blocks
* wpf - personal fouls

## **TourneyCompactResults**

This file identifies the game-by-game NCAA tournament results for all seasons of historical data. The data is formatted exactly like the regular\_season\_compact\_results.csv data. Note that these games also include the play-in games (which always occurred on day 134/135) for those years that had play-in games.

## **TourneyDetailedResults**

This file contains the more detailed results for tournament games from 2003 onward.

## **TourneySeeds**

This file identifies the seeds for all teams in each NCAA tournament, for all seasons of historical data. Thus, there are between 64-68 rows for each year, depending on the bracket structure.

* "season" - the year
* "seed" - this is a 3/4-character identifier of the seed, where the first character is either W, X, Y, or Z (identifying the region the team was in) and the next two digits (either 01, 02, ..., 15, or 16) tells you the seed within the region. For play-in teams, there is a fourth character (a or b) to further distinguish the seeds, since teams that face each other in the play-in games will have the same first three characters. For example, the first record in the file is seed W01, which means we are looking at the #1 seed in the W region (which we can see from the "seasons.csv" file was the East region). This seed is also referenced in the "tourney\_slots.csv" file that tells us which bracket slots face which other bracket slots in which rounds.
* "team" - this identifies the id number of the team, as specified in the teams.csv file

## **TourneySlots**

This file identifies the mechanism by which teams are paired against each other, depending upon their seeds. Because of the existence of play-in games for particular seed numbers, the pairings have small differences from year to year. If there were N teams in the tournament during a particular year, there were N-1 teams eliminated (leaving one champion) and therefore N-1 games played, as well as N-1 slots in the tournament bracket, and thus there will be N-1 records in this file for that season.

* "season" - the year
* "slot" - this uniquely identifies one of the tournament games. For play-in games, it is a three-character string identifying the seed fulfilled by the winning team, such as W16 or Z13. For regular tournament games, it is a four-character string, where the first two characters tell you which round the game is (R1, R2, R3, R4, R5, or R6) and the second two characters tell you the expected seed of the favored team. Thus the first row is R1W1, identifying the Round 1 game played in the W bracket, where the favored team is the 1 seed. As a further example, the R2W1 slot indicates the Round 2 game that would have the 1 seed from the W bracket, assuming that all favored teams have won up to that point. The slot names are different for the final two rounds, where R5WX identifies the national semifinal game between the winners of regions W and X, and R5YZ identifies the national semifinal game between the winners of regions Y and Z, and R6CH identifies the championship game. The "slot" value is used in other columns in order to represent the advancement and pairings of winners of previous games.
* "strongseed" - this indicates the expected stronger-seeded team that plays in this game. For Round 1 games, a team seed is identified in this column (as listed in the "seed" column in the tourney\_seeds.csv file), whereas for subsequent games, a slot is identified in this column. In the first record of this file (slot R1W1), we see that seed W01 is the "strongseed", which during the 1985 tournament would have been Georgetown. Whereas for games from Round 2 or later, rather than a team seed, we will see a "slot" referenced in this column. So in the 33rd record of this file (slot R2W1), it tells us that the winners of slots R1W1 and R1W8 will face each other in Round 2. Of course, in the last few games of the tournament - the national semifinals and finals - it's not really meaningful to talk about a "strong seed" or "weak seed", but those games are represented in the same format for the sake of uniformity.
* "weakseed" - this indicates the expected weaker-seeded team that plays in this game, assuming all favored teams have won so far. For Round 1 games, a team seed is identified in this column (as listed in the "seed" column in the tourney\_seeds.csv file), whereas for subsequent games, a slot is identified in this column.

# References

**There are no sources in the current document.**

<https://en.wikipedia.org/wiki/NCAA_Division_I_Men%27s_Basketball_Tournament>

<https://en.wikipedia.org/wiki/NCAA_basketball_tournament_selection_process>

<https://en.wikipedia.org/wiki/List_of_NCAA_Division_I_men%27s_basketball_programs>

<http://i.turner.ncaa.com/sites/default/files/external/printable-bracket/bracket-ncaa.pdf>

<https://www.kaggle.com/c/march-machine-learning-mania-2017/data>

http://machinelearningmastery.com/perform-feature-selection-machine-learning-data-weka/

## 

1. <https://www.kaggle.com/c/march-machine-learning-mania-2017/data> [↑](#footnote-ref-1)
2. <http://www.masseyratings.com/> [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)