**Basketball**

**Data Analytics Battles**

**Sponsored by**

**Drake University**

**&**

**The Iowa Center**

**for**

**Interdisciplinary Training**

**Competition Overview**

The Basketball Data Analytics Battles is a competition for individual teams of High School or Middle School students that will introduce them to the exciting world of Data Science. Prior to the announcement of the 2020 NCAA Basketball Tournament field, competitors must submit an algorithm that will choose ten teams from the tournament field most likely to win games and number them one through ten. Each time a team in the list wins a tournament game, points are earned equivalent to the team’s assigned number in the list. In addition, if a team in the list that is seeded 9-16 wins a game in the tournament, that team will earn additional five bonus points as a “Cinderella” team. For a first round win, a Cinderella team will earn five bonus points, for a second round win the bonus earned will be ten points. For each additional round game won, add on five more bonus points.

For example, Villanova was the champion of the 2018 tournament. They won 6 games as a #1 seed. If a submitted algorithm for the 2018 tournament chose Villanova and assigned it the number 9, the choice will earn a total of (9 x 6) = 54 points. If the same algorithm chose #11 seed Loyola of Chicago and assigned the team the number 3, the Cinderella team that won four games in the tournament earns a total of (4 x 3) + (5+10+15+20) = 62 points.

**NOTE: The largest point total one can earn is 695 points. However, this score is next to impossible since it will require Cinderella teams to completely dominate the tournament. Any total score of 180 points or more can be considered good performance.**

To assist in the development of high performing algorithms, competitors will be provided the number of tournament games won and the statistics on approximately 30 variables for each qualifying team for the 2013 through 2019 tournaments. This will allow for competitors to test the performance of algorithms they develop over the previous seven tournaments.

**Rules for the Competition**

1. Each team must have an adult sponsor that registers the team. This adult can be a teacher, parent or guardian, club advisor, or a responsible person age at least 21. The adult sponsor will be responsible for making sure the students comply with the rules and meet all the requirements and deadlines for the algorithm submission.
2. As an individual team, you are only allowed to submit one algorithm for the competition. A team consists of 2 to 4 students, grades 6 through 12.
3. A template will be provided to the sponsors for algorithm submission.
4. In terms of guidance, sponsors can teach their students any concepts or train them on any tools that they can use that may help them in the development of algorithms. For example, sponsors can coach the students on how to use tools found in Excel or statistical software such as R and SAS or teach them statistical concepts that can be useful such as calculating z-scores or correlations. Sponsors are not allowed to provide examples or assist in the development of algorithms. Sponsors can answer questions students may have as they develop algorithms. This competition will rely on the honor system. Some example algorithms are provided later in this document and are not eligible for submission.

**Algorithm Submission**

1. On March 13th, 2020 before the tournament field is announced, sponsors will collect all the algorithms submitted.
2. Once the tournament field is announced Sunday March 15th, 2020, the 2020 Excel spread sheet will be constructed and made available to the sponsors as soon as possible.
3. Once in their possession, the sponsors make the 2020 Excel spread sheet available to their student teams so they can evaluate their algorithms and determine their team lists for eventual submission.
4. For accuracy purposes, sponsors can either check themselves or have teams exchange their algorithms to check for accuracy. We advise having teams exchange their algorithms as a component of the learning experience.
5. Once checked, the sponsors collect all algorithm submissions and submit them electronically to the Iowa Center for Interdisciplinary (ICIT) Director Rick Spellerberg at rickspellerberg@yahoo.com.
6. Sponsors will be provided team scoring updates several times during the competition to let their teams know where they stand as well at the end of the competition.
7. This competition is run on the Honor System to reduce operational costs in the hopes of never having to require a registration fee. All questions that come up in the course of the competition can be directed to the ICIT Director at any time.

**Important Competition** **Dates**

November 1, 2019 – March 13, 2020: Algorithm development phase.

**March 13, 2020: Deadline for sponsors first collection of algorithms**.

March 15, 2020: Selection Sunday

March 16 or 17, 20120: Sponsors are provided the 2020 Excel spread sheet.

March 19-22, 2020: Rounds 1 and 2 of the NCAA tournament.

**March 25, 2020: Deadline for sponsors to submit evaluated algorithms to the ICIT.**

March 26-29, 2020: Rounds 3 and 4 of the NCAA tournament.

April 4, 2019: The Final Four

April 6, 2019: The Championship Game

NOTE: Teams will be provided updates on competition results between the 2nd and 3rd rounds, and after the 4th round. The final competition results will be announced as soon as possible after the championship game.

**Basketball Data Analytics Battles Conference**

As a culmination to the competition, Drake University and the Iowa Center for Interdisciplinary Training and will host a conference to celebrate the efforts of all the participants. All interested schools and individual teams are invited to present their algorithms at a poster board session. Schools and individual teams will be invited to give oral presentations. Professionals in the industry and representatives from related undergraduate programs will be represented and will give presentations. The date for this conference will be as soon as possible after the championship game and made available to all the participants as soon as the plans are finalized.

**Example Algorithms**

Consider the following thought. Any team that has qualified for the tournament and managed to defeat many of their opponents during the season by a large margin are candidates for doing very well in the tournament. If you believe this fact is true, then the variable representing the average point differential per game may be a wonderful way of measuring how many games a team will win in the tournament.

**Proposed Algorithm**: Order the teams in the tournament by largest to smallest average point differential per game. Assign the team with the highest value the number ten, the second highest the number 9 and continue this process until you get to the tenth highest value and assign that team the number 1. If in a step of the process there are two or more teams with the same average point differential per game, assign the values using alphabetical order via the school’s name.

Now we must test this algorithm for performance. We will first consider the 2016 tournament results. Assuming you analyze the Excel spreadsheet for the 2016 season correctly, you will get the following results. The teams starred were seeded 9-16 making them eligible for bonus points.

TEAM Rating Games Won Points Earned Region

\*Stephen F Austin 10 1 15 E

Michigan State 9 0 0 MW

\*Wichita State 8 1 13 S

Kansas 7 3 21 S

\*Gonzaga 6 2 27 MW

Stoney Brook 5 0 0 E

Indiana 4 2 8 E

Villanova 3 6 18 S

Purdue 2 0 0 MW

North Carolina 1 5 5 E

**Total Points Earned = 107 Maximum Points Possible = 695**

Now if you test the algorithm on the 2013 through 2015 tournaments and the 2017 through 2018 tournaments, the algorithm will perform at the following levels.

**2013 = 112 points 2014 = 97 points 2015 = 170 points 2017 = 129 points 2018 = 111**

The average points earned for this algorithm over the five tournaments is 121.0. Considering a good average will be in the 170+ range, this algorithm appears to be a weak performer. There is one year 2015 where the algorithm performs much better than the others, but this is offset by five other years where the algorithm poorly performs. Analysis of the algorithm is now important. If you look at the results produced by the algorithm for the 2016 tournament and focus on the regions, you will notice that no team in the West Region was selected, this is a potential problem. At no point in the tournament will this algorithm earn points from teams in the west region for the 2016 tournament. This may be the problem.

**Adjusted Algorithm**: First order the teams from the highest to lowest average point differential per game. We now create the list of ten teams. Now find the team from each region with the highest average point differential per game and order them from largest to smallest average point differential. Number these teams in reverse order 10, 9, 8 and 7. Now find the team in each region with the second highest average point differential and order them from largest to smallest average point differential. Number these teams in reverse order 6, 5, 4, and 3. Finally, find the team with the third highest average point differential in each region and order them from highest to lowest average point differential. Take the top two teams and number them in reverse order 1 and 2. You now have your list of ten teams chosen taking the regional assignments into account.

If we test the adjusted algorithm that takes Regional assignments of the teams into consideration for the 2016 tournament, we end up with the following results.

TEAM Rating Games Won Points Earned Region

\*Stephen F Austin 10 1 15 E

Michigan State 9 0 0 MW

\*Wichita State 8 1 13 S

\*Yale 7 1 12 W

Kansas 6 3 18 S

\*Gonzaga 5 2 25 MW

Stoney Brook 4 0 0 E

Texas A&M 3 2 6 W

Indiana 2 2 4 E

Villanova 1 6 6 S

**Total Points Earned = 99 Maximum Points Possible = 695**

The first thing you will notice is that the total points earned with the adjusted algorithm is slightly better, so it seems the adjusted algorithm performed better. Also, if you total the number of games won listed in the third column, you will notice the adjusted algorithm outperformed the original algorithm by one game.

Now if you test the algorithm on the 2013 through 2015 tournaments and the 2017 through 2018 tournaments, the algorithm will perform at the following levels.

**2013 = 112 points 2014 = 97 points 2015 = 166 points 2017 = 138 points 2018 = 113**

The average points earned for this algorithm over the five tournaments is 120.83. By going with the adjusted algorithm there is a small decrease in average performance. In the real world, this level of decrease, while appearing small, can mean millions of dollars oflost revenue for a company.

The adjusted average point differential per game algorithm that takes regional assignments into consideration performed slightly worse than the original algorithm, this indicates creating algorithms that chose teams across all the regions may not be that important. However, when you reach the final four teams in the tournament, you hope all four teams appear in your list earning you points along the way. This will not be possible if any region is not represented by a team in your list.

Developing an algorithm that only takes in consideration one variable is leaving out too many factors that contribute to success or failure and this fact may explain why the previous algorithms do not perform well. One way of incorporating several variables into an algorithm is by taking a weighted average over several variables. If y1 , y2, …. , yn are a list of n-variables, then a weighted average of these variables looks like the following expression:

(A1y1 + A2y2 +…. + Anyn) / (A1 + A2 +…. + An)

Where each Ai is a real number greater than zero. By calculating a weighted average, algorithms for the competition can be created that take in account several variables.

**Weighted Average Algorithm:** We will calculate the following weighted average for each qualifying team where y1 = the three-point percentage, y2 = the free throw percentage and y3 = the assist to turnover ratio.

(y1 + 2y2 +3y3) / 6

First order the teams from the highest to lowest average rating using the weighted average described above. We now create the list of ten teams. Now find the team from each region with the highest rating and order them from largest to smallest rating. Number these teams in reverse order 10, 9, 8 and 7. Now find the team in each region with the second highest rating and order them from largest to smallest rating. Number these teams in reverse order 6, 5, 4, and 3. Finally, find the team with the third highest rating in each region and order them from highest to lowest average point differential. Take the top two teams and number them in reverse order 1 and 2. You now have your list of ten teams chosen taking the regional assignments into account.

Now we must test this algorithm for performance. We will first consider the 2016 tournament results. Assuming you analyze the Excel spreadsheet for the 2016 season correctly, you will get the following results.

TEAM Rating Games Won Points Earned Region

Michigan State 10 0 0 MW

North Carolina 9 5 45 E

Iowa 8 1 8 S

Saint Joe 7 1 7 W

Virginia 6 3 18 MW

\*Michigan 5 0 0 E

Villanova 4 6 24 S

Texas A & M 3 2 6 W

\*SFA 2 1 7 E

Purdue 1 0 0 MW

**Total Points Earned = 115 Maximum Points Possible = 695**

Now if you test the algorithm on the 2013 through 2015 tournaments and the 2017 through 2018 tournaments, the algorithm will perform at the following levels.

**2013 = 38 points 2014 = 115 points 2015 = 104 points 2017 = 69 points 2018 = 109**

The average points earned for this algorithm over the five tournaments is 91.67. By going with the weighted average algorithm described it appears the performance level will not be as strong as the previous algorithms tested. Two of the variables are percentages in decimal form and the third, the assist to turnover ratio when good can be 2 or higher. The algorithm may perform much better by significantly lowering the weight used for the assist to turnover ratio. To potentially improve the performance of this algorithm, it will be necessary to adjust the weights on the three variables and conduct additional testing.

**Scaling a Data Set to the Interval [0 , 1]**

The example algorithm previously discussed that uses a weighted average of several variables to establish a “rating”. In our example, two of the variables involved percentages, (free throw and 3-point) and one was a ratio of counts, (assists to turnovers). In a sense, we are comparing apples to oranges. This can make it difficult to choose appropriate weights. By scaling data sets to the interval [0 , 1], we can avoid this problem.

Let

Y = any data value in the list Ymin = minimum value in the data set

 Ymax = maximum value in the data set Ys = the corresponding scaled data vale

then,

Ys = ( Y – Ymin ) / ( Ymax – Ymin )

Notice that this scaling function takes Ymin to 0 and Ymax to 1 in the scaling. The function also preserves the ordering of the data. It also produces a list of “unitless” values, i.e., we are no longer comparing apples to oranges.

**Correlations & Ratings**

We saw that a weighted average can be used to develop a “rating” value for each team in terms of predicting the number of games a team will win in the tournament. In our example the variables chosen each had a positive correlation in terms of games won in the tournament, i.e., the larger the data value, the better. Some variables have a negative correlation. For instance, the number of turnovers a team had in the season or whether a team is a Cinderella or not. For each of these variables we expect the lower the number the more games that will be won. When developing a rating using a mixture of correlated variables, it makes sense to subtract out terms involving variables that have a negative correlation. Depending on the number of each type of correlations are used, you may need to order the rating from largest to smallest or vice versa before constructing your team list. Testing your algorithm for performance will help you determine which. This subject can be a great topic for teachers to discuss with their students to assist them in algorithm development.

**Cinderella Bonus System**

One last thing the students may want to consider is to be sure the algorithms they develop take advantage of the Cinderella bonus system. One thing that sponsors can suggest to their students is to build a two step algorithm, one of which is devoted exclusively to picking a set number of Cinderella teams. For example, the algorithm can be constructed to pick non-Cinderella teams first and assign them the values 4 through 10. It then picks three Cinderella teams and assigns then the values 1 through 3. The options here are endless.