Intelligent Systems TC2011.1
March Madness Prediction

Rodolfo Adrián Beltrán Nájera – A1561890
PhD. Benjamín Valdés Aguirre
10/10/2022

Abstract

The following document have as purpose to code in python using dataframe to develop a decision tree.

Introduction

Basketball is one of the most influential sports today, many fans not only follow professional leagues around the world, but the also follow college basketball. United States having one of the best infrastructures around sports focus on developing young superstars and one great way to gain visibility to the media is through March Madness.

The NCAA Division 1 men's basketball tournament is commonly known as March Madness, which is a single elimination tournament played each spring in the United States. It features 68 college basketball teams from NCAA Division 1 level and the winner is crown as the national champion, which not only receives the media attention, but the possibility become a professional basketball player in the NBA or overseas.

During the next sections, an implementation of a Decision Tree will be shown with the objective if a team is good enough to make it to the biggest D1 college basketball tournament of the year.

Data Set

This project uses "College Basketball Dataset" which includes datasets from season 2013 through 2021 season, only 2021 data will be use in this project. The dataset contains 21 features and 1 outcome (MM: March Madness), but only the 5 most important features will be used in this project.

Link to dataset:
https://www.kaggle.com/datasets/andrewsundberg/college-basketball-dataset

Features:

1. G: Number of games played
2. W: Number of games won
3. ADJOE: Adjusted Offensive Efficiency (An estimate of the offensive efficiency (points scored per 100 possessions) a team would have against the average Division I defense)

4. ADJDE: Adjusted Defensive Efficiency (An estimate of the defensive efficiency (points allowed per 100 possessions) a team would have against the average Division I offense)
5. BARTHAG: Power Rating (Chance of beating an average Division I team)

To understand how the dataset is built 5 cases will be displayed.

X feature inputs					y results
G	W	ADJOE	ADJDE	BARTHAG	MM
24	20	118.1	81.1	0.9521	1
24	22	123.2	94.5	0.9548	1
29	21	123.5	95.7	0.9491	1
15	6	98.6	108.5	0.2493	0
25	8	102.2	109.2	0.3185	0

y results have only 1 and 0, 1 stand for reaching the playoffs and 0 to miss them.

Implementation

The first step was to import the dataset and name each column to know with what we are working and simplify the information taking out the data we don't consider relevant for this project.

With y (result) and X (features) well define we can start to feed the algorithm so it can determine an outcome with the data given by the user. Sklearn was utilized and it worked well since it is very friendly to any user, giving the opportunity to set the maximum depth and teaching the algorithm with the function fit.

```
#hyper parameter max depth
tree_clf = DecisionTreeClassifier(max_depth = 4)
tree_clf.fit(X.values,y)
```

Now it is time to select the data of one team to determine if it's going to make it to the playoffs, we need to give the 5 feature values. With the following values, a test run will be done.

G	W	ADJOE	ADJDE	BARTHAG
25	20	121.18	102.9	0.8245

```
How many games did they played? 25

How many games did they win? 20

What is the average of points scored by the team per game? 121.18

What is the average of points allowed by the teamper game? 102.9

What are the chances of beating a D1 team? (0 - 1) 0.8245

Better luck next year :(
```

Unfortunately, the team is not ready to go to March Madness, even though they have great statistics the volume of teams is incredibly high a many of those teams have more than 0.85 BARTHAG being a crucial statistic to take into consideration.

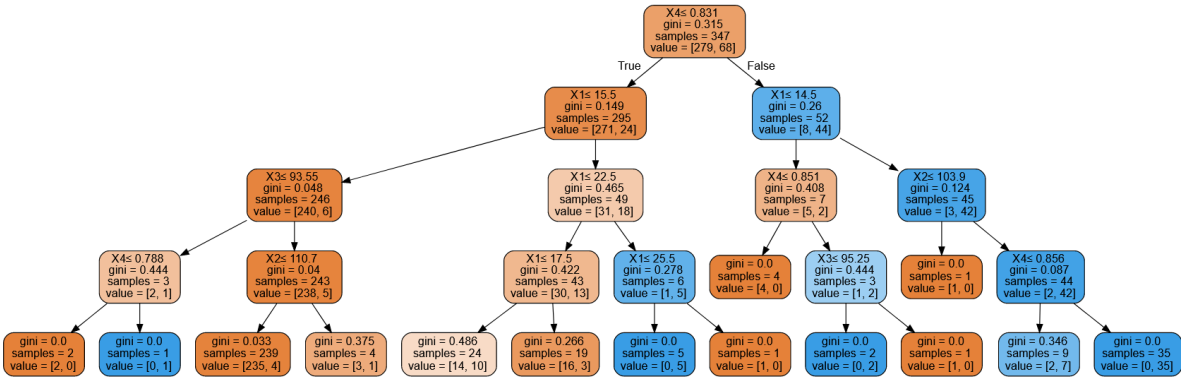
Now that we have the data, we can do a Decision Tree with the following code. It is important to remark that the code uses Gini Impurity to determine how well a decision tree was split.

```
from sklearn import tree
import graphviz

dot = tree.export_graphviz(tree_clf,
                           out_file=None,
                           filled=True,
                           rounded=True,
                           special_characters=True,
                           fontname='helvetica')

#Display a graph of the Decision Tree
graph = graphviz.Source(dot)
graph
```

It displays the following Decision Tree.



Conclusion

The code work to a low scale, since basketball is a have many statistics (basic and advance) a more complex code can be done, in the decision tree the depth is set to 4 and we can see that it needs more than 4 iterations to have a more precise outcome. It may be overfitting the amount of information this code can in order to give the best possible answer and for a it can be improve by using more features and use one decision tree per conference to create a random forest and try to predict is a team can be a national champion or not.