Michael Er Jun Li

260869379

MAIS 202

February 18th 2019

Iteration 2

1. Problem statement

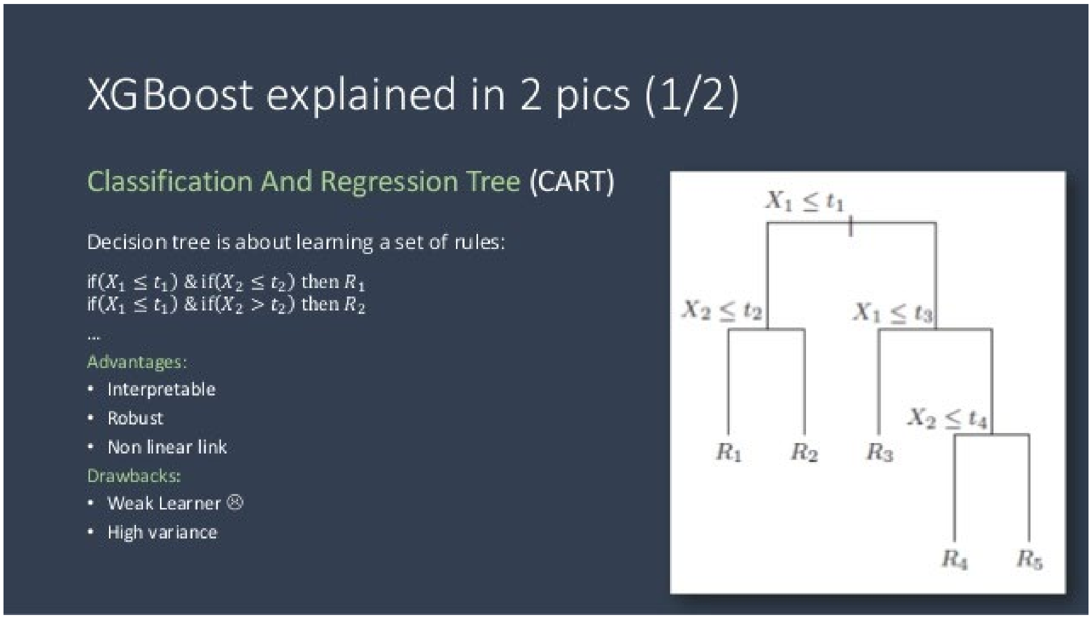
My project idea is to create an English Premier League (soccer) prediction bot that will bet using its prediction of the outcome of each game which are based on the results of previous games. We have one target variable which is FTR (stands for full-time result). We want to predict FTR given all the features in the dataset.

2. Data Preprocessing

I am working with the datasets from http://www.football-data.co.uk/englandm.php instead of https://www.soccerstats.com/trends.asp?league=england. Although the second dataset gives important statistics on each team i.e. the number of matches a team has kept clean sheets or the number of games a team failed to score at home or away (%), the first dataset is better since it gives us the final result of each match which is the most important information in my opinion. In fact, this dataset contains insightful information about each game from the beginning of the season such as full-time result, the number of shots on target, the number of corners and free kicks obtained by each team and the number of yellow and red cards given to each team. Every column of my dataset is represented by an acronym which will be explained in more detail by a legend contained in another document. FTR which stands for full-time result is the most important feature in my dataset, however it is a string which represent either “H”, “A” or “D”. The first letter means that the home team won. “A” means that the away team has won and “D” means that the game is a draw. In addition, I will take in consideration the number of corners and free kicks obtained by each team as well as shots on target, because theses are real goal opportunities. The dataset’s size is (262,62). We can see that it contains a lot of features, therefore I will perform a dimension reduction on the dataset and only keep the most important features. This means that there are 61 features and 261 samples (games). It is important to note that the dataset is updated weekly by the website to include the latest fixtures of the BPL.

3. Machine learning model

Originally, I was hesitant between decision tree and SVM. However, I read about XGBoost which is an implementation of gradient boosted decision trees designed for speed and performance. With that being said, gradient boosting is basically a combination of gradient descent and boosting. Boosting is a machine learning algorithm primarily for reducing bias, and also variance in supervised learning. It's an algorithm that converts weak learners to strong ones. A 'weak' learner (classifer, predictor, etc) is just one which performs relatively poorly--its accuracy is above chance, but just barely. In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification. The idea is given a weak learner, we run it multiple times on (reweighted) training data, then let learned classifier class. We reduce the error at each iteration using gradient descent.



XGBoost explained in 2 pics (1/2)

Classification And Regression Tree (CART)

Decision tree is about learning a set of rules:

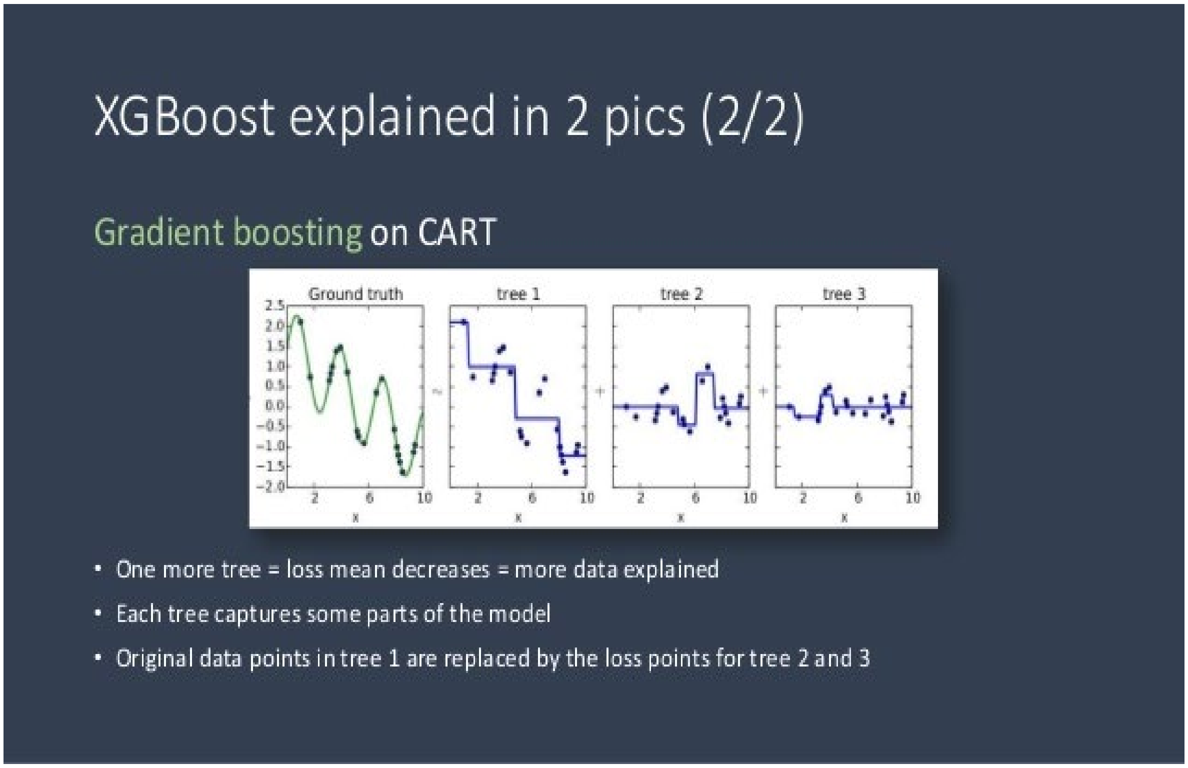
if$(X_1 \leq t_1)$ & if$(X_2 \leq t_2)$ then R_1
if$(X_1 \leq t_1)$ & if$(X_2 > t_2)$ then R_2
...

Advantages:

- Interpretable
- Robust
- Non linear link

Drawbacks:

- Weak Learner ☹
- High variance



XGBoost explained in 2 pics (2/2)

Gradient boosting on CART

- One more tree = loss mean decreases = more data explained
- Each tree captures some parts of the model
- Original data points in tree 1 are replaced by the loss points for tree 2 and 3

4. Preliminary results

Results are printed in the Iteration 2. ipynb file. My preliminary results are around 50% which is very low. This means that my model can correctly predict one game for each 2 features. For the moment, I'm predicting the outcome of each match based on the data collected on the actual game (number of shots on target, number of free kicks and corner kicks, etc.) to give an insight on the importance of the selected features. However, the end goal would be to predict the outcome of two randomly picked teams.

5. Next steps

The next steps would be to tune my hyperparameters as well as working on the presentation. I would like to program a bot that will automatically trade for me. Most importantly, I will have to work on my dataset and combine some of the features in order to give more insightful information on the game. I want to group data by team to make a profil for each one of them. In addition, I would like to get the number of points collected by each team when they are playing at home or away. These features would be more appropriate for my analysis and would allow my model to increase its accuracy.