# Predicting the MVP rank of NBA players based on their season performances

Aurélien Callens

# Context and Objectives

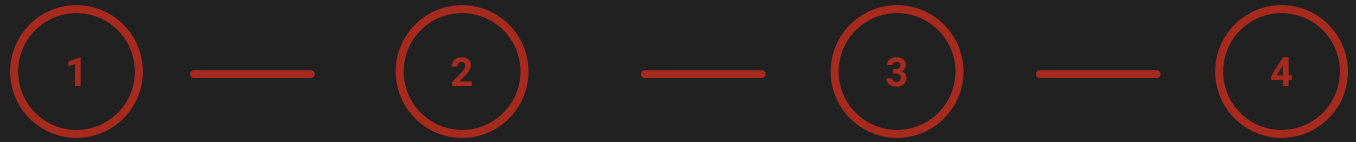
The NBA Most Valuable Player (MVP) trophy is an award given each year by the National Basketball Association (NBA) to the best player of the regular season.  To rank a player, we use the share. The higher the share is, the lower his rank will be. For a given year, the share of a player is given by:

$$Share = \frac{Number\ of\ MVP\ votes}{Maximum\ number\ of\ MVP\ votes}$$

**Objective: Build a model to predict the MVP rank of each NBA player based on data collected on players and their team**

# General workflow

Idea: Predict the share for each player with random forest (regression task) then perform the ranking based on the predicted values of the share




**Data cleaning**

Fill missing values + create rank variable based on true values of share + indicate correct data type for each variable

**Exploratory data analysis**

Visualization of the correlation matrix to see the relationships between shares and other features

**Modeling**

K-fold cross validation grouped on year to find the best hyperparameters for random forest algorithm

**Analyzing the results**

Evaluation of the performances of random forest

# Choices made for this analysis

Data cleaning:

- Fill missing values with 0: NA present when ratio were computed with 0 as numerator
- Remove the categorical variables: focus only on numerical variables

Modelling choices:

- Random forest: no preprocessing needed,  few hyperparameters to tune
- Grouped K-fold cross validation: for hyperparameter tuning, to evaluate objectively the performances, to account for the temporality of the data
- Regression task to predict share value: regression task ‘easier’ than ranking task
- Metrics: Normalized Discounted Cumulative Gain

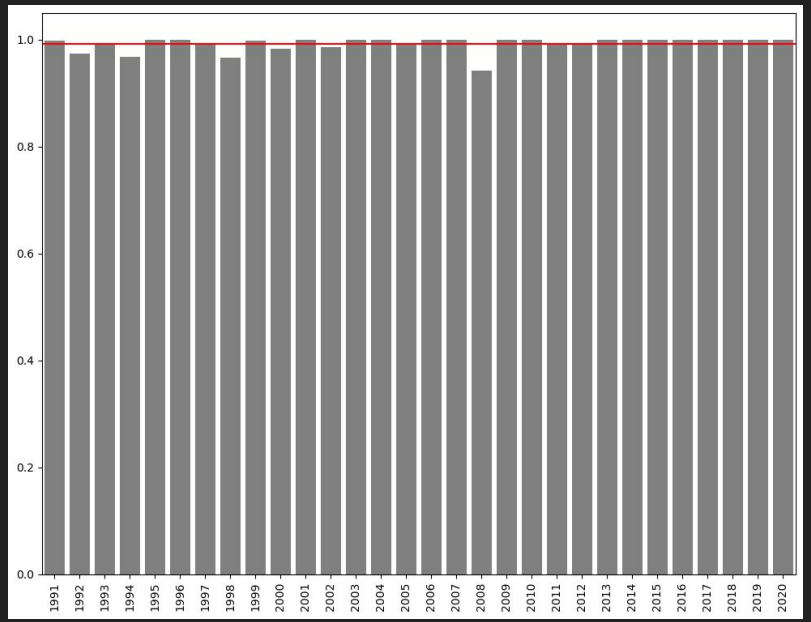
# Results

Ranking performances (top 3) on train data:

- NCDG averaged over all the years: 0.992

MVP prediction performance:

| | |
|---|---|
| 13520 | 2 |
| 2 | 28 |




Top 3 predicted for 2021:

# If I had more time

- Analyzing the errors of the models
- Change evaluation metric for the grid search
- Fine hyperparameter tuning
- Include more features (especially categorical variables)
- Test Xgboost model which are known to handle well ranking task (trade off between complexity and performances ?)