Basketball season

A data mining project

Group T05G03
(33 %) Guilherme Matos
(33 %) João Ferreira
(33 %) Luís Arruda

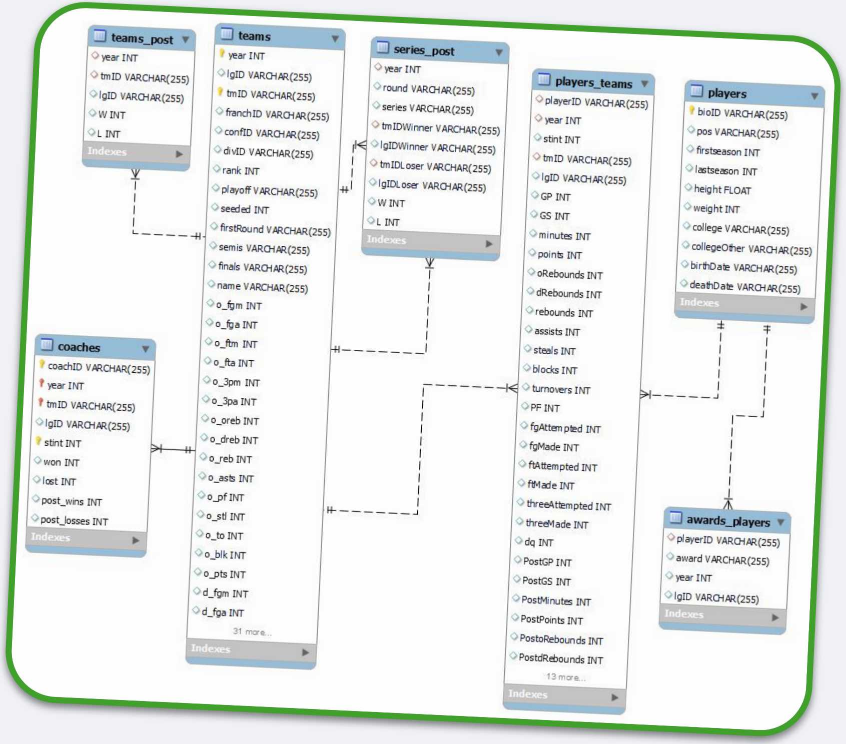
AC (M.EIC) 2025/2026

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

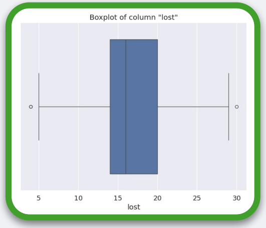
Domain description

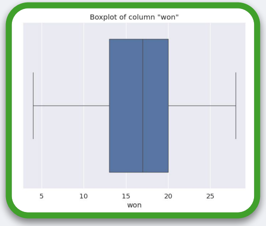
10 years of basketball tournaments:

- 2 different **conferences** per year;
- Each tournament is made of a **regular season** (round-robin) and a **post season** (playoffs);
- Each **team** is made of a coach (that can change anytime) and a list of players;
- For each **player** is specified their position, team, and stats for each year;
- There is a set of **awards** that may be given each year to players, teams, and coaches.



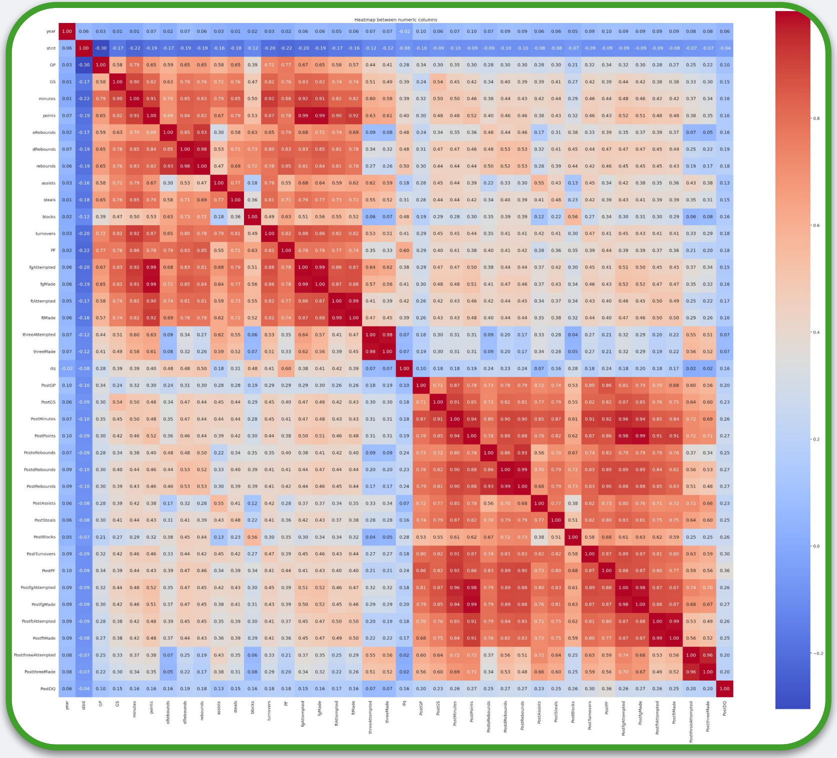
Exploratory data analysis





Approach:

- **Heatmaps** of tables to see the correlation of variables;
- **Box plots** (median, standard deviation, quarters, minimum and maximum) of columns to detect outliers;
- **Python types** of each column to detect strings in number fields;
- Automatic checks for **nulls** and **missing data**;
- **Manual analysis** of the database.



Data cleaning

Actions taken from the data analysis:

- Removed **empty** columns
  (e.g. `divID` in `teams.csv`);
- Removed columns with always the **same** values
  (e.g. `lgID` is always `WNBA`);
- Removed **redundant** columns
  (e.g. `rebounds` = `oRebounds` + `dRebounds`);
- Corrected **bad formatted** data
  (e.g. `Kim Perrot Sportsmanship` should be `Kim Perrot Sportsmanship Award`);
- Removed **duplicate data**
  (e.g. `Most improved player` in year 5 has two players).

cleaner.py

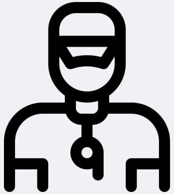
```python
columns_to_remove: dict[str, Any] = {
    "players_teams": ["lgID"],
    "players": [
        "collegeOther", "firstseason", "lastseason",
        "deathDate", "birthDate"
    ],
    "awards_players": ["lgID"],
    "series_post": ["lgIDWinner", "lgIDLoser"],
    "teams_post": ["lgID"],
    "teams": [
        "lgID", "divID", "seeded",
        "tmORB", "tmDRB", "tmTRB",
        "opptmORB", "opptmDRB", "opptmTRB"
    ],
    "coaches": []
}

columns_to_remove_redundancy: dict[str, list[str]] = {
    "players_teams": [
    "rebounds", "threeAttempted", "PostRebounds",
    "PostfgAttempted", "PostfgMade", "PostMinutes",
    "PostthreeMade", "PostftMade",
    ],
    "teams": [
    "d_fta", "o_3pa", "min", "d_fgm", "d_reb"
    ],
}

values_to_rename: dict[str, dict[str, dict[str, str]]] = {
    "awards_players": {
        "award": {
            "Kim Perrot Sportsmanship": "Kim Perrot Sportsmanship Award"
        }
    }
}
```

Data mining problems

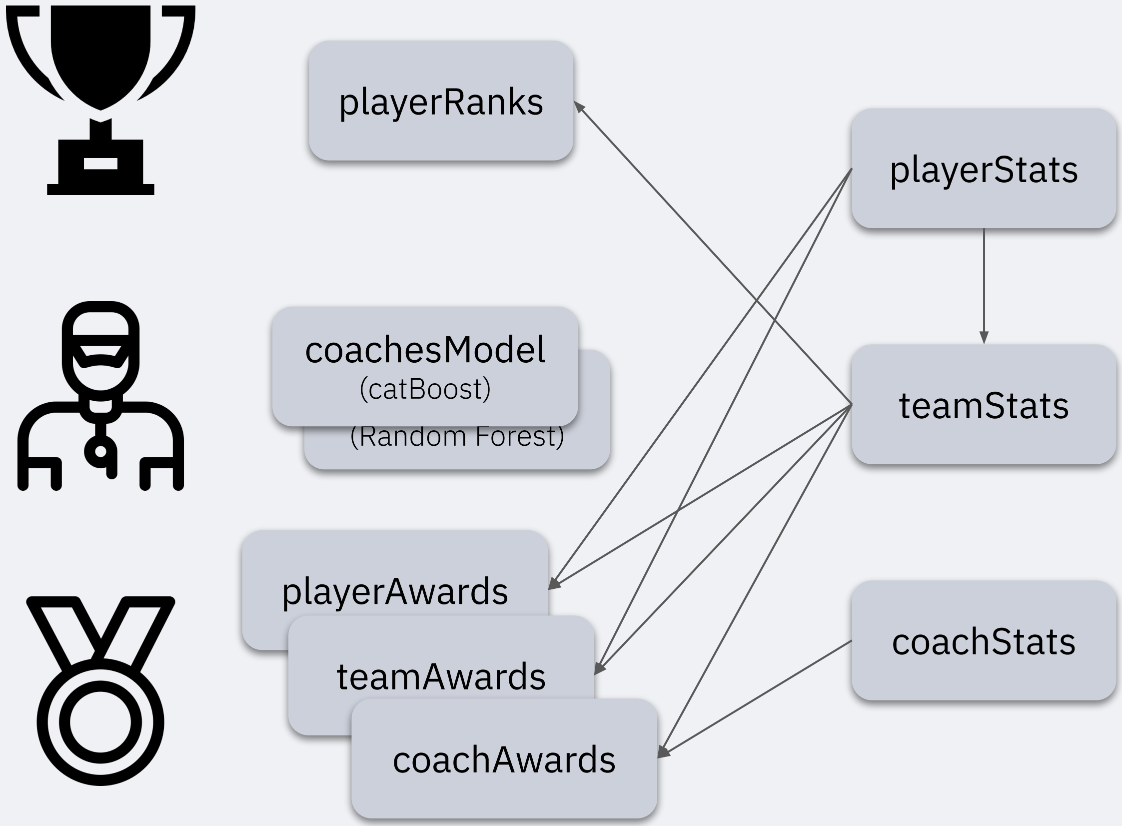
1. The ranking of the regular season for each conference



2. Set of teams that will change coaches

3. Who won each of the individual awards

Model architecture



Before solving the problems, **important statistics** are **forecasted** for the players, teams and coaches for the test year, so that those data points can be **used as input** for the remaining models.

Is this a bad idea? Yes.

PlayerStats model

The playerStats model was designed to predict a certain player's performance based on past contributions to a team, based on their teammates (team synergy) and their individual contributions.

Pros:

- The model allows for player synergy detection; meaning it can theoretically identify subsections of teams that contribute more to a team's performance;
- It is independent of the team name or other factors that do not reflect a player's potential;
- It allows for the deduction of various player parameters from small data inputs, making a trained model of this useful for various other models that require the aforementioned data;

Cons:

- It depends heavily on data abundance, meaning, if the training dataset is small, it is not better than pure function transformations on the training data (pure functions such as averages and means);
- It adds one more layer of statistical inference of data, meaning later usage of the data by other models will accrue errors from this model as well;
- The way it is constructed leads to data sparsity on teammate identification, which reduces the performance of the model while simultaneously making it less scalable and more computationally expensive;
- It will misbehave when encountering players for which there is no data;

#1: Problem definition

Problem:
The ranking of the regular season for each conference; this is a multi-label classification problem.

Input:

- `teams.csv` with statistical columns.

Output:

- `rank` column of `teams.csv` for the test year.

```
year,tmID,franchID,confID,rank,playoff,firstRound,semis,finals,name,o_fgm,o_fga,
9,ATL,ATL,EA,7,N,,,,Atlanta Dream,895,2258,542,725,202,340,737,1077,492,796,285,
10,ATL,ATL,EA,2,Y,L,,,Atlanta Dream,1089,2428,569,755,114,404,855,1259,547,741,3
1,CHA,CHA,EA,8,N,,,,Charlotte Sting,812,1903,431,577,131,305,630,935,551,713,222
2,CHA,CHA,EA,4,Y,W,W,L,Charlotte Sting,746,1780,410,528,153,309,639,948,467,605,
3,CHA,CHA,EA,2,Y,L,,,Charlotte Sting,770,1790,490,663,211,302,653,955,496,647,24
4,CHA,CHA,EA,2,Y,L,,,Charlotte Sting,787,1881,456,590,187,342,629,971,499,697,27
5,CHA,CHA,EA,5,N,,,,Charlotte Sting,745,1744,436,590,166,256,616,872,426,648,210
6,CHA,CHA,EA,6,N,,,,Charlotte Sting,772,1913,447,624,104,316,609,925,493,727,284
7,CHA,CHA,EA,6,N,,,,Charlotte Sting,864,2178,552,777,176,347,747,1094,527,734,31
7,CHI,CHI,EA,7,N,,,,Chicago Sky,858,2175,449,643,157,357,680,1037,509,674,277,52
8,CHI,CHI,EA,6,N,,,,Chicago Sky,925,2277,489,723,186,383,782,1165,597,634,262,51
9,CHI,CHI,EA,5,N,,,,Chicago Sky,899,2100,564,812,110,357,770,1127,548,639,269,52
10,CHI,CHI,EA,5,N,,,,Chicago Sky,930,2136,527,693,186,307,776,1083,532,605,249,5
1,CLE,CLE,EA,2,Y,W,L,,Cleveland Rockers,809,1828,426,570,141,331,603,934,539,64
```

Relevant data:

- Team composition (players);
- Team synergy (wins, losses and other statistics of a particular team composition);
- Players’ defensive and offensive contributions (See this);

#1: Data preparation

LabelEncoder:

Encode labels `tmID` and `confID`;

The strings are converted into an integer, so that the model can use those entries. After testing, those integers and converted back to the original string.

Filtering specific features:

Only the following statistic-related columns are used as features:

```
"o_fgm",
"o_fga",
"o_ftm",
"o_fta",
"o_3pm",
"o_oreb",
"o_dreb",
"o_reb",
"o_asts",
"o_pf",
"o_stl",
"o_to",
"o_blk",
"o_pts",
```

```
"d_fgm",
"d_ftm",
"d_3pm",
"d_3pa",
"d_oreb",
"d_dreb",
"d_asts",
"d_pf",
"d_stl",
"d_to",
"d_blk",
"d_pts",
```

#1: Experimental setup

Approach caveats:

Despite being a classification problem, our approach used a **regression mechanism followed by a sorting stage**, given the labels are not independent but describe an ordered ranking. As such, the model attempts to derive a “score” that can be ordered, with a value that closely resembles the rank the team would be classified as.

NuSVR

The NuSVR model is a **support vector regressor** that allows the control of the Nu parameter, which is “An upper bound on the fraction of training errors and a lower bound of the fraction of support vectors” (from scikit learn).

K-fold cross validation

For testing, the k-fold cross validation method was used to maximize the rentability of the training data.

In each iteration, 9 years are used for training and 1 year is used for validation.

For each year, a new model is tested with that year, training a new model on the remaining ones.

#1: Results

With statistics from **teamStats**:

Rank	Accuracy	# Hits	# Total
1	0.10	2	20
2	0.10	2	20
3	0	0	20
4	0	0	20
5	0.10	2	20
6	0.05	1	20
7	0.12	2	16
8	0.17	1	6

With statistics from **original data**:

Rank	Accuracy	# Hits	# Total
1	0.70	14	20
2	0.40	8	20
3	0.45	9	20
4	0.20	4	20
5	0.20	4	20
6	0.45	9	20
7	0.62	10	16
8	0.50	3	6

#1: Year 11 Predictions

Conference **EA**:

Rank	Team
1	NYL
2	WAS
3	CHI
4	CON
5	ATL
6	IND

Conference **WE**:

Rank	Team
1	TUL
2	PHO
3	SEA
4	LAS
5	MIN
6	SAS

#2: Problem definition

- The objective is to predict **which teams will undergo a coaching change during the same year of a given season.**
- The model should indicate whether a team will perform a coaching change (1) or not (0).
- This is a **binary classification problem**.

coaches.csv

coachID	year	tmID	lgID	stint	won	lost	post_wins	post_losses
adamsmi01w	5	WAS	WNBA	0	17	17	1	2
adubari99w	1	NYL	WNBA	0	20	12	4	3
adubari99w	2	NYL	WNBA	0	21	11	3	3
adubari99w	3	NYL	WNBA	0	18	14	4	4
adubari99w	4	NYL	WNBA	0	16	18	0	0
adubari99w	5	NYL	WNBA	1	7	9	0	0
adubari99w	6	WAS	WNBA	0	16	18	0	0
adubari99w	7	WAS	WNBA	0	18	16	0	2
adubari99w	8	WAS	WNBA	1	0	4	0	0
aglerbr99w	1	MIN	WNBA	0	15	17	0	0

#2: Data preparation

- **Used coaches.csv** and **teams.csv** for **RandomForest** and **CatBoost**
- Merged tables on (teamID, year)
- **Used players_teams.csv** as additional input for **CatBoost**
- **Removed rows where stint = 2**, as they are not used in the prediction
- Used **LabelEncoder** to transform categorical variables for the **RandomForest**
- **CatBoost handles categorical data internally**, requiring only the list of categorical feature indices

#2: Data preparation

1. **computeInheritedTalent**

   Calculates each player's efficiency for last year and sums the values per team/year.

2. **computeCoachTenure**

   Sorts each coach's historical record and increments tenure by +1 whenever the coach remains with the same team in the next season; resets to 1 when the coach changes teams.

3. **computePrevCoachMadePlayoffs**

   Checks whether the coach reached the playoffs last year

4. **computeCoachPrevWin**

   Computes the coach's historical win percentage for last year

5. **bumpPastStats**

   Retrieves the coach's previous-season statistics and the team's previous-season statistics.

#2: Experimental setup

Walk-forward validation was used, **training on all past seasons and testing on the following one**, which better reflects real-world conditions since future seasons should never use information from the future.

Confusion matrix used to assess predictions for each test year.

Hyperparameter tuning for the RandomForest was attempted using **GridSearchCV** with **ROC_AUC** as the scoring metric to better handle the rare positive cases, but the small dataset caused failures in early seasons; even with **StratifiedKFold** some splits still lacked positive samples, so in those cases the model reverted to a default RandomForest without tuning.

The **CatBoost** model was trained using **Logloss**, which **penalizes confident wrong predictions** and encourages well-calibrated probability estimates.

#2: Results

rfCoaches.py

YEAR	PRECISION %	PRECISION	RECALL %	RECALL
2	--	0/0	0%	0/2
3	0%	0/4	0%	0/3
4	14%	1/7	50%	1/2
5	33%	2/6	67%	2/3
6	20%	1/5	50%	1/2
7	14%	1/7	100%	1/1
8	0%	0/9	0%	0/1
9	25%	1/4	100%	1/1
10	0%	0/3	0%	0/3

```
tmID,year,stint,stint_pred,prob_change
```

```
WAS,7,0,1,86
CHA,7,0,1,62
LAS,7,0,1,68
HOU,7,0,0,38
CHI,7,0,1,50
NYL,7,0,0,21
SEA,7,0,0,35
SAS,7,0,1,52
DET,7,0,0,45
MIN,7,1,1,84
CON,7,0,0,42
PHO,7,0,1,50
SAC,7,0,0,42
IND,7,0,0,31
WAS,8,1,0,11
SAC,8,0,1,51
LAS,8,0,1,51
NYL,8,0,1,61
SEA,8,0,0,36
SAS,8,0,1,82
DET,8,0,1,52
CHI,8,0,1,51
CON,8,0,0,35
HOU,8,0,1,51
PHO,8,0,1,79
IND,8,0,0,27
MIN,8,0,1,51
```

```
SEA,9,0,0,48
SAC,9,0,0,34
LAS,9,0,0,44
NYL,9,0,1,51
IND,9,0,0,48
PHO,9,0,0,48
SAS,9,0,0,17
CHI,9,0,0,48
DET,9,0,1,53
ATL,9,0,0,48
WAS,9,1,1,80
CON,9,0,0,46
HOU,9,0,1,70
MIN,9,0,0,44
SEA,10,0,1,65
SAC,10,1,0,46
LAS,10,0,0,3
NYL,10,1,0,6
IND,10,0,0,7
PHO,10,0,0,38
MIN,10,0,0,41
SAS,10,0,1,58
CHI,10,0,1,53
DET,10,1,0,20
ATL,10,0,0,4
WAS,10,0,0,41
CON,10,0,0,6
```

#2: Results

catBoostCoaches.py

YEAR	PRECISION %	PRECISION	RECALL %	RECALL
2	12%	2/16	100%	2/2
3	0%	0/1	0%	0/3
4	50%	2/4	100%	2/2
5	0%	0/2	0%	0/3
6	20%	1/5	50%	1/2
7	20%	1/5	100%	1/1
8	0%	0/3	0%	0/1
9	50%	1/2	100%	1/1
10	--	0/0	0%	0/3

```
tmID,year,stint,stint_pred,prob_change
```

```
WAS,7,0,0,20
CHA,7,0,0,40
LAS,7,0,0,26
HOU,7,0,0,36
CHI,7,0,1,71
NYL,7,0,0,22
SEA,7,0,1,64
SAS,7,0,0,3
DET,7,0,1,59
MIN,7,1,1,91
CON,7,0,1,58
PHO,7,0,0,23
SAC,7,0,0,31
IND,7,0,0,11
WAS,8,1,0,2
SAC,8,0,1,63
LAS,8,0,0,4
NYL,8,0,0,27
SEA,8,0,0,3
SAS,8,0,0,28
DET,8,0,0,18
CHI,8,0,0,6
CON,8,0,0,12
HOU,8,0,1,67
PHO,8,0,1,92
IND,8,0,0,7
MIN,8,0,0,4
```

```
SEA,9,0,0,2
SAC,9,0,0,10
LAS,9,0,0,3
NYL,9,0,0,23
IND,9,0,0,1
PHO,9,0,0,5
SAS,9,0,0,12
CHI,9,0,0,5
DET,9,0,0,2
ATL,9,0,0,3
WAS,9,1,1,59
CON,9,0,0,3
HOU,9,0,1,86
MIN,9,0,0,16
SEA,10,0,0,2
SAC,10,1,0,3
LAS,10,0,0,2
NYL,10,1,0,4
IND,10,0,0,0
PHO,10,0,0,7
MIN,10,0,0,5
SAS,10,0,0,40
CHI,10,0,0,23
DET,10,1,0,5
ATL,10,0,0,2
WAS,10,0,0,1
CON,10,0,0,5
```

#2: Year 11 Predictions

catBoost:

Team	SEA	CHI	IND	PHO	LAS	SAS	WAS	ATL	MIN	TUL	CON	NYL
Stint	0	0	0	0	1	0	0	0	0	0	0	1
Prob.	5	1	3	8	64	10	0	4	3	1	5	67

randomForest:

Team	SEA	CHI	IND	PHO	LAS	SAS	WAS	ATL	MIN	TUL	CON	NYL
Stint	0	0	0	0	1	1	0	1	0	0	1	1
Prob.	25	41	49	24	66	62	41	72	41	41	87	66

#3: Problem definition

Problem:

Who won each of the individual awards?

- A multi-label classification problem.

Input:

- `players_teams.csv` with statistical columns;
- `teams.csv` with statistical columns;
- `coaches.csv` with statistical columns;
- `players.csv`;
- `awards.csv` with historical data.

Output:

- `awards.csv` for the test year.

```
playerID,award,year
thompti01w,All-Star Game Most Valuable Player,1
leslili01w,All-Star Game Most Valuable Player,2
leslili01w,All-Star Game Most Valuable Player,3
teaslni01w,All-Star Game Most Valuable Player,4
swoopsh01w,All-Star Game Most Valuable Player,6
douglka01w,All-Star Game Most Valuable Player,7
```

Types of awards:

- Players:
  - "All-Star Game Most Valuable Player"
  - "Defensive Player of the Year"
  - "Kim Perrot Sportsmanship Award"
  - "Most Improved Player"
  - "Most Valuable Player"
  - "Rookie of the Year"
  - "Sixth Woman of the Year"
  - "WNBA Finals Most Valuable Player"
- Teams:
  - "WNBA All-Decade Team"
  - "WNBA All Decade Team Honorable Mention"

  These awards are for each decade and only one year is available for testing!
- Coaches:
  - "Coach of the Year"

This will require, at least, three different models with their tailored features.

#3: Data preparation

- **Feature Engineering:**

Players	Teams	Coaches
• Per-game statistics (PPG, RPG, APG, etc.); • Shooting percentages; • Efficiency metrics; • Team performance context; • Year-over-year improvements; • League rankings.	• Career totals and averages; • Peak season performance; • Seasons played; • Playoff appearances and championships; • Individual awards won; • Team success metrics.	• Win percentage and season record; • Playoff performance; • **Team improvement metrics;** • Conference standings; • Home/away performance; • Historical coaching record.

- Standardizing features;
- Handling infinite values.

#3: Experimental setup

Model:
GradientBoostingClassifier

A simple choice for a complex dataset with noise and large variance.

Testing:
Walk-forward validation

- bpg_rank: Player percentile rank of blocks per game
- rpg_rank: Player percentile rank of rebounds per game
- point_differential: avg_o_pts - avg_d_pts
- carrer_ppg: Points per game, with all games in account
- carrer_apg: Assists per game, with all games in account
- carrer_rpg: Rebounds per game, with all games in account

Feature analysis:

All-Star Game Most Valuable Player	dRebounds: 0.3529
Defensive Player of the Year	steals: 0.4354
Kim Perrot Sportsmanship Award	bpg_rank: 0.2027; PF: 0.1948
Most Improved Player	PostoRebounds: 0.3974
Most Valuable Player	PostAssists: 0.3274; dRebounds: 0.2942
Rookie of the Year	dq: 0.2572
Sixth Woman of the Year	PostPF: 0.3318; rpg_rank: 0.2983
WNBA Finals Most Valuable Player	PostfgMade: 0.2150
Coach of the Year	point_differential: 0.1479
WNBA All-Decade Team:	career_ppg: 0.3427
WNBA All Decade Team H.M.:	career_apg: 0.2243; peak_rpg: 0.1956

#3: Results

Test with year 10:

```
✗ All-Star Game Most Valuable Player
  Predicted: powelni01w, Actual: cashsw01w
√ Defensive Player of the Year
  Predicted: catchta01w, Actual: catchta01w
✗ Kim Perrot Sportsmanship Award
  Predicted: castriz01w, Actual: lawsoka01w
✗ Most Improved Player
  Predicted: catchta01w, Actual: langhcr01w
✗ Most Valuable Player
  Predicted: pondeca01w, Actual: tauradi01w
✗ Rookie of the Year
  Predicted: augusse01w, Actual: mccouan01w
✗ Sixth Woman of the Year
  Predicted: wrighta01w, Actual: bonnede01w
✗ WNBA Finals Most Valuable Player
  Predicted: pondeca01w, Actual: tauradi01w
✗ Coach of the Year
  Predicted: gaineco01w, Actual: meadoma99w
```

```
WNBA All-Decade Team:
  coopecy01w
  leslili01w
  jacksla01w
  birdsu01w
  griffyo01w
  thompti01w
  mcwilta01w
  catchta01w
  melvich01w
  smithka01w
```

```
WNBA All Decade Team
Honorable Mention:
  weathte01w
  boltoru01w
  holdsch01w
  tauradi01w
  penicti01w
```

Test with year 9:

```
✗ All-Star Game Most Valuable Player
  Predicted: leslili01w, Actual: cashsw01w
✗ Defensive Player of the Year
  Predicted: tauradi01w, Actual: leslili01w
✗ Kim Perrot Sportsmanship Award
  Predicted: wiggica01w, Actual: johnsvi01w
✗ Most Improved Player
  Predicted: wiggica01w, Actual: hoffmeb01w
√ Most Valuable Player
  Predicted: parkeca01w, Actual: parkeca01w
✗ Rookie of the Year
  Predicted: wiggica01w, Actual: parkeca01w
- Sixth Woman of the Year
  N/A: Insufficient positive samples
✗ WNBA Finals Most Valuable Player
  Predicted: nolande01w, Actual: smithka01w
✗ Coach of the Year
  Predicted: laimbbi01w, Actual: thibami99w
```

Note that the model is trained on all known years before the test year, so the year 10 is not used in this test.

#3: Year 11 Predictions

Award	Person
All-Star Game Most Valuable Player	adairje01w
Defensive Player of the Year	adairje01w
Kim Perrot Sportsmanship Award	swanike01w
Most Improved Player	adairje01w
Most Valuable Player	youngta01w
Rookie of the Year	youngta01w
Sixth Woman of the Year	adairje01w
WNBA Finals Most Valuable Player	adairje01w
Coach of the Year	dunnli99wc

WNBA All-Decade Team:
coopecy01w
leslili01w
thompti01w
jacksla01w
smithka01w
catchta01w
swoopsh01w
birdsu01w
griffyo01w
mcwilta01w

WNBA All Decade Team Honorable Mention:
weathte01w
boltoru01w
penicti01w
tauradi01w
johnssh01w

Note that the team award predictions:

- *Assume that the year 11 is when the awards are given, which is not true;*
- *Do not have enough data to correctly predict an accurate result.*

Conclusions

The models did **not achieve high performance**, which was expected given the **limited and irregular** nature of the **data**.

Even without strong predictive results, the project successfully: cleaned and organized complex datasets, built robust training and time-aware validation pipelines, explored different modeling approaches and techniques.

Limitations

The data we have contains a **high level of noise**.

The dataset is relatively small, and this amplifies inconsistencies.

It becomes difficult to predict player statistic when some players with existing history suddenly play only two minutes, or when rookies enter the league and only log a couple of minutes with the rest of their stats remaining close to zero.

Future work

- Experiment additional hyperparameters;
- Explore alternative ways of generating player statistics;
- Try new modeling strategies.