APPENDIX

A Raw Data

Tournament	Home Team	Away Team	Match Date	Phase	Final Score	Extra Periods	Home Team Points	Home Team Two Pointers Made	Home Team Two Pointers Attempted	...
Basket League	KAOD	Aris	2013-10-12	Regular Season	58-66	0.0	58.0	20.0	43.0	...
Basket League	PAOK	Rethymno	2014-10-12	Regular Season	88-79	1.0	88.0	15.0	33.0	...
Basket League	Koroivos	Trikala	2015-10-10	Regular Season	64-73	0.0	64.0	18.0	44.0	...
Basket League	Doxa Lefkadas	PAOK	2016-10-08	Regular Season	84-78	0.0	84.0	16.0	38.0	...
Basket League	Kolossos Rhodes	Rethymno	2017-10-07	Regular Season	64-63	0.0	64.0	19.0	43.0	...
Eurocup	Bonn	Alba Berlin	2013-10-15	Regular Season	65-86	0.0	65.0	14.0	35.0	...
Eurocup	Gran Canaria	Cantu	2014-10-14	Regular Season	101-81	0.0	101.0	35.0	66.0	...
Eurocup	Le Mans	Gran Canaria	2015-10-13	Regular Season	74-79	0.0	74.0	21.0	45.0	...
Eurocup	Alba Berlin	Fuenlabrada	2016-10-12	Regular Season	88-81	0.0	88.0	23.0	48.0	...
Eurocup	Ulm	Tofas Bursa	2017-10-10	Regular Season	83-73	0.0	83.0	26.0	45.0	...
Euroleague	Bamberg	Strasbourg	2013-10-16	Regular Season	84-70	0.0	84.0	21.0	34.0	...
Euroleague	Baskonia	Neptunas	2014-10-15	Regular Season	88-69	0.0	88.0	23.0	44.0	...
Euroleague	Unicaja Malaga	Bamberg	2015-10-15	Regular Season	76-71	0.0	76.0	23.0	36.0	...
Euroleague	Real Madrid	Olympiacos	2016-10-12	Regular Season	83-65	0.0	83.0	19.0	41.0	...
Euroleague	CSKA Moscow	Armani Milano	2017-10-12	Regular Season	93-84	0.0	93.0	26.0	50.0	...
Liga ACB	Bilbao	Zaragoza	2013-10-12	Regular Season	77-86	0.0	77.0	24.0	46.0	...
Liga ACB	Real Betis	Tenerife	2014-10-04	Regular Season	87-96	0.0	87.0	25.0	48.0	...
Liga ACB	Gran Canaria	Gipuzkoa	2015-10-10	Regular Season	97-64	0.0	97.0	26.0	41.0	...
Liga ACB	Estudiantes	Real Betis	2016-10-01	Regular Season	82-95	0.0	82.0	16.0	27.0	...
Liga ACB	Gran Canaria	Gipuzkoa	2017-10-01	Regular Season	84-76	0.0	84.0	20.0	37.0	...

Table A.1: Sample of raw data, first match of every season per tournament

Tournament/ League	2013-2014	2014-2015	2015-2016	2016-2017	2017-2018	Total
Euroleague	253	251	250	259	260	1273
Eurocup	366	305	304	146	184	1305
Greek League	205	207	207	206	204	1029
Liga ACB	329	328	328	295	327	1607
Total	1153	1091	1089	906	975	5214

Table A.2: Number of games per tournament and season

B Performance Indicators

FGA: Field goal attempts

$FGA = AP_2 + AP_3$

AP_k Attempted shots of k points $(k = 2, 3)$

FGM: Field goal shots made

$FGM = P_2 + P_3$

P_k Shots of k points made $(k = 2, 3)$

FGS: Field goal shots percentage

$FGS = FGM/FGA$

TREB%: Total rebound percentage

$$TREB\% = 100 \times \frac{TReb}{TReb + OTReb}$$

$TReb$: Total team rebounds
$OTReb$: Total opponent rebounds

ASST%: Assisted field goal percentage

$$ASST = 100 \times \frac{Assists}{FGM}$$

TS%: True shooting percentage

$$TS\% = 100 \times \frac{Points}{2(FGA + 0.44FTA)}$$

FTA = Free throws attempted

EFG%: Effective field goal percentage

$$EFG\% = 100 \times \frac{FGM + \frac{1}{2}P_3}{FGA}$$

OREB%: Offensive rebound percentage

$$OREB\% = \frac{OReb}{OReb + ODReb}$$

DREB%: Defensive rebound percentage

$$DREB\% = \frac{DReb}{DReb + OOReb}$$

$OReb$: Offensive team rebounds
$DReb$: Defensive team rebounds
$OOReb$: Offensive opponent rebounds
$ODReb$: Defensive opponent rebounds

TO%: Turnover percentage

$$TO\% = \frac{Turnovers}{FGA + 0.44FTA + Turnovers}$$

Poss: Possession

$Poss = FGA + 0.44FTA - OReb + Turnovers$

Continued on next page

STL%: Steal percentage	$STL\% = 100 \times \frac{Steals}{Poss}$
BLK%: Block percentage	$BLK\ = 100 \times \frac{Blocks}{Poss}$
BLKR: Block rate	$BLKR = 100 \times \frac{Blocks}{OAP_2}$ OAP_2: Opponent's attempted two pointers
PPS: Points per shot	$PPS = \frac{Points}{FGA}$
FIC: Floor impact counter	$FIC = Points + OReb + 0.75DReb + Assists + Steals + Blocks$ $-0.75FGA - 0.375FTA - Turnovers - 0.5Fouls$
AR: Assist rate	$AR = 100 \times \frac{Assists}{FGA - 0.44FTA + Assists + Turnovers}$
AST/TO: Assist to turnover ratio	$AST/TO = Assists/Turnovers$
STL/TO: Steal to turnover ratio	$STL/TO = Steals/Turnovers$
Play% : Play percentage	$Play\% = \frac{FGM}{FGA - OReb + Turnovers}$
Performance Index	Performance Index $= Points + Rebounds + Assists + Steals$ $+Blocks + Fouls\ \ Drawn$ $-(MFG + MFT + Turnovers + Blocks)$ $-Fouls\ \ Committed$ $MFG = FGA - FGM$: Missed field goals $MFT = FTA - FTM$: Missed free throws
GmSc: Hollinger Game Score	$GmSc = Points + 0.4FGM - 0.7FGA - 0.4MFT$ $+0.7OReb + 0.3DReb + Steals + 0.7Assists$ $+0.7Blocks - 0.4Fouls - Turnovers$
Ortg: Offensive rating	$Ortg = 100 \times \frac{Points}{Poss}$
Drtg : Defensive rating	$Drtg = 100 \times \frac{OpponentPoints}{OpponentPoss}$
EDiff : Efficiency differential	$EDiff = Ortg - Drtg$

Table B.1: Performance indicators

C Correlations with Point Difference

Performance Indicator	**Pearson Correlation**
Efficiency differential (Ediff=Offensive - Defensive rating)	0.98
Floor impact counter (FIC)	0.76
Performance Index	0.76
Hollinger game score	0.72
Defensive rating (Drtg)	0.66
Offensive rating (Ortg)	0.66
Play percentage indicator (Play%)	0.62
Shooting percentage (TS%)	0.60
Effective field goals percentage (EFG%)	0.59

Table C.1: Top-ten correlated Performance Indicator with points difference

D Rating Systems Explanation

D.1 Elo rating system

The Elo rating system is updated weekly value after every game-day using the formula

$$R_a^{(t)} = R_a^{(t-1)} + K\left(w_a^{(t)} - e_a^{(t)}\right), \tag{3}$$

where $R_a^{(t)}$ and $R_a^{(t-1)}$ are the ratings for game-day t and $t-1$ for team a, $w_a^{(t)}$ is the actual game outcome (win = 1; loss = 0;) for team a at game-day t and $e_a^{(t)}$ is the expected outcome (probability) for team a based on $R^{(t-1)}$ given by

$$e_a^{(t)} = \left\{1 + \exp\left(\tfrac{1}{400}\left(R_{O_a^{(t)}}^{(t-1)} - R_a^{(t-1)}\right)\log 10\right)\right\}^{-1}, \tag{4}$$

where $O_a^{(t)}$ is the opponent team of team a at game-day t and $R_{O_a^{(t)}}^{(t-1)}$ is its corresponding ELO rating as calculated from the data that were available before game-day t.

Parameter K is a multiplying factor controlling the sensitivity of the rating. This factor allows us to update the ratings, depending on the points difference, thus rewarding an 80-60 win more strongly than an 80-75. Following Hvattum & Arntzen (2010), one possibility is to specify K using the formula

$$K = k_0(1+\delta)^{\lambda}, \tag{5}$$

with δ being the absolute point difference, while $k_0 > 0$ and $\lambda > 0$ are tuning parameters.

D.2 PageRank approach

The PageRank approach for ranking teams is calculated by implementing the PageRank algorithm which was originally introduced by Page et al. (1999) for ranking websites. This is roughly based on a network representation where each team is a single node and two nodes are connected if they have played each other in the tournament we study. The weight of each directed link is important for the calculation of the final PageRank rating value. After an extensive study and comparison, Lazova & Basnarkov (2015) proposed to specify the weight using the function

$$f_{a,b} = \frac{l_{a,b}}{g_{a,b}} \times \frac{1}{G - g_{a,b} + 1}, \tag{6}$$

where

- $f_{a,b}$: weight of the link from node a to node b,
- $l_{a,b}$: number of games lost by team a amongst all games where a and b compete each other,
- $g_{a,b}$: number of games played between the two teams and
- G : maximum number of games played between any pair of teams/nodes.

In their publication, this weighting scheme was reported as the best among ten different alternatives. As input in the PageRank algorithm, we have used statistics based on the last year (365 days) in order to avoid obtaining outdated team ratings. For the implementation of the PageRank algorithm, we have used the python package "NetworkX".

D.3 Pi-rating

The Pi-rating is a dynamic approach for evaluating the strength of each team. In particular, this rating is re-calculated after each game (denoted here as time point t). Discrepancies between the predicted and the observed point difference determine the change (increase or decrease) of the rating. In this approach, each team is attached to two pi-rating values, one for the home and one for the away games of the team. An overall pi-rating for a team a is simply obtained by the mean of the two distinct values. Pi-ratings are considered highly informative rating measures which capture both the current form and the historical strength of each team. In this work, we have calculated all pi-ratings as in Constantinou & Fenton (2013) by using a translation to Python of the R package "piratings". The Pi-ratings learning rates γ which is the impact the home performances have on away ratings λ which is the change of old ratings with new ratings based on the recent results have been tuned separately for each tournament based on the mean square error of the expected and the observed point difference for each game (see Constantinou & Fenton (2013)) for data of seasons 2013/2014 - 2016/2017 for each tournament. To obtain the optimal values for (γ, λ) we have considered a grid of values from 0.01 to 0.25 with step 0.005 $\gamma = 0.01, 0.015, 0.2, ..., 0.25$ and a grid from 0.01 to 0.9 with step 0.005 $\lambda = 0.01, 0.015, 0.02, ..., 0.9$. The obtained mean square error and the optimal values are depicted in Figure D.3.1. Moreover, Table D.3.1 provides a summary of these values along with the mean square error values and the means of the tuning parameters across the four tournaments which can serve as a "good" default value for future implementations.

Tournament/ League	γ	λ	**Minimum mean square error**	**Mean square error of average parameters**	**Differences of mean square errors**
Euroleague	0.57	0.09	155.52	155.70	0.18
Eurocup	0.61	0.15	172.84	175.14	2.30
Greek League	0.55	0.09	142.16	142.76	0.60
Liga ACB	0.58	0.08	172.64	173.46	0.82
All leagues (Average)	0.58	0.10	160.79	161.76	0.94

Table D.3.1: Summary statistics of tuned Pi-rating parameters

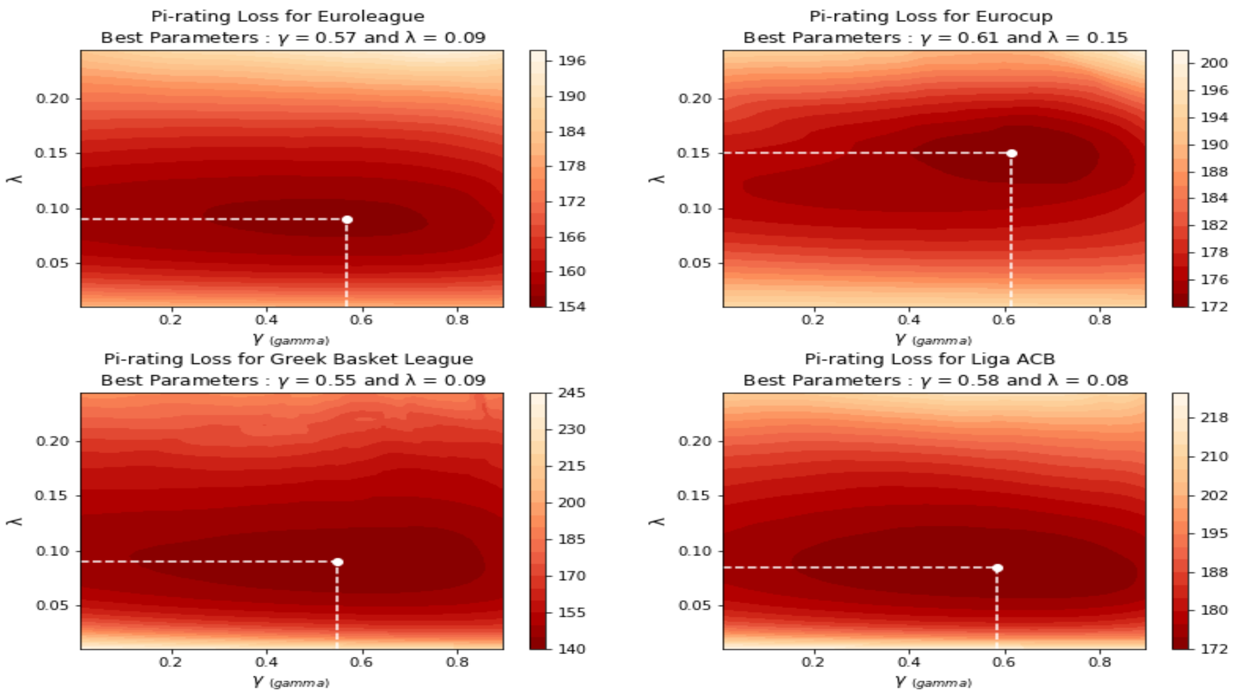


Figure D.3.1: Plot for finding the optimal mean square error (MSE) of points for a grid of γ and λ Pi-rating parameter values

E Final Features

Measures	**Home vs Away (Last Game)**	**Home vs Away (All Games)**	**All Games between the two teams**
Percentage of Wins	✗	✓$_{x_1}$	✓$_{x_2}$
Points Difference	✓$_{x_3}$	✓$_{x_4}$	✓$_{x_5}$
Ediff – Efficiency differential	✓$_{x_6}$	✓$_{x_7}$	✓$_{x_8}$
Winner[a]	✓$_{x_9}$	✗	✗

[a] 1 if the winner is the home team, -1 if the winner is the away team.

(a) Features of specific game records

Performance Measures (Games of one year period / Last 10 matches)	**Teams Performance Indices (Achieved Measures)**		**Opponent Team Indices (Conceded Measures)**	
	Mean	**Standard Deviation**	**Mean**	**Standard Deviation**
Wins	✓$_{x_{10}\ /\ x_{50}}$	✗	✗	✗
Points Difference	✓$_{x_{11}\ /\ x_{51}}$	✓$_{x_{12}\ /\ x_{52}}$	✗[b]	✗[b]
Ediff – Efficiency differential	✓$_{x_{13}\ /\ x_{53}}$	✓$_{x_{14}\ /\ x_{54}}$	✗[b]	✗[b]
FIC – Floor Impact Counter	✓$_{x_{15}\ /\ x_{55}}$	✓$_{x_{16}\ /\ x_{56}}$	✓$_{x_{17}\ /\ x_{57}}$	✓$_{x_{18}\ /\ x_{58}}$
Performance Index	✓$_{x_{19}\ /\ x_{59}}$	✓$_{x_{20}\ /\ x_{60}}$	✓$_{x_{21}\ /\ x_{61}}$	✓$_{x_{22}\ /\ x_{62}}$
Hollinger Game Score	✓$_{x_{23}\ /\ x_{63}}$	✓$_{x_{24}\ /\ x_{64}}$	✓$_{x_{25}\ /\ x_{65}}$	✓$_{x_{26}\ /\ x_{66}}$
Ortg – Offensive Rating	✓$_{x_{27}\ /\ x_{67}}$	✓$_{x_{28}\ /\ x_{68}}$	✗[c]	✗[c]
Drtg – Defensive Rating	✓$_{x_{29}\ /\ x_{69}}$	✓$_{x_{30}\ /\ x_{70}}$	✗[c]	✗[c]
Play%	✓$_{x_{31}\ /\ x_{71}}$	✓$_{x_{32}\ /\ x_{72}}$	✓$_{x_{33}\ /\ x_{73}}$	✓$_{x_{34}\ /\ x_{74}}$
Points	✓$_{x_{35}\ /\ x_{75}}$	✓$_{x_{36}\ /\ x_{76}}$	✓$_{x_{37}\ /\ x_{77}}$	✓$_{x_{38}\ /\ x_{78}}$
TS% – Shooting Percentage	✓$_{x_{39}\ /\ x_{79}}$	✓$_{x_{40}\ /\ x_{80}}$	✓$_{x_{41}\ /\ x_{81}}$	✓$_{x_{42}\ /\ x_{82}}$
EFG% – Effective Field Goal Percentage	✓$_{x_{43}\ /\ x_{83}}$	✓$_{x_{44}\ /\ x_{84}}$	✓$_{x_{45}\ /\ x_{85}}$	✓$_{x_{46}\ /\ x_{86}}$

[b] Same with the achieve measure with negative value.
[c] The opposite of the achieve measure (achieved Ortg = conceded Drtg).

(b) Box Score based performance indicators

All Games/ Last 10 Matches	**Elo Rating**	**PageRank**	**Pi-ratings**
Differences of Values	✓$_{x_{47}\ /\ x_{87}}$	✓$_{x_{48}}$[d]$_{/\ x_{88}}$	✓$_{x_{49}\ /\ x_{89}}$

[d] For the last one year (365 days).

(c) Team performance ratings

Measures of one year period	**Mean**	**Standard Deviation**	**Dummy variable**
Wins[e]	✓$_{x_{90}}$	✗	✗
Points Difference[e]	✓$_{x_{91}}$	✓$_{x_{92}}$	✗
Ediff[e]– Efficiency differential	✓$_{x_{93}}$	✓$_{x_{94}}$	✗
FIC – Floor Impact Counter	✓$_{x_{95}}$	✓$_{x_{96}}$	✗
Performance Index	✓$_{x_{97}}$	✓$_{x_{98}}$	✗
Hollinger Game Score	✓$_{x_{99}}$	✓$_{x_{100}}$	✗
Ortg – Offensive Rating	✗[f]	✗[f]	✗
Drtg – Defensive Rating	✓$_{x_{101}}$	✓$_{x_{102}}$	✗
Play%	✓$_{x_{103}}$	✓$_{x_{104}}$	✗
Points	✗[g]	✓$_{x_{105}}$	✗
TS% – Shooting Percentage	✓$_{x_{106}}$	✓$_{x_{107}}$	✗
EFG% – Effective Field Goal Percentage	✓$_{x_{108}}$	✓$_{x_{109}}$	✗
Phase	✗	✗	✓$_{x_{110}}$

[e] From the side of home teams
[f] Same with the Defensive Rating (Drtg)
[g] Same with the mean of points difference

(d) Tournament features

Table E.1: Description and labels of features used for predictive models and algorithms

Tournament	Home Team	Away Team	Match Date	Phase	tradition_pointsdiff_match	pi_ratings	history_FIC	Current_form_EFG	Tournament_Game_Score	...
Basket League	PAOK	Rethymno	2014-10-12	Regular Season	7.00	4.04	9.99	3.81	8.10	...
Basket League	Koroivos	Trikala	2015-10-10	Regular Season	22.00	4.04	-2.83	-2.89	7.51	...
Basket League	Doxa Lefkadas	PAOK	2016-10-08	Regular Season	0.00	-12.20	-49.40	-51.81	8.49	...
Basket League	Kolossos Rhodes	Rethymno	2017-10-07	Regular Season	2.80	2.83	-0.44	1.37	9.95	...
Eurocup	Gran Canaria	Cantu	2014-10-14	Regular Season	0.00	-11.40	-62.97	-55.96	8.71	...
Eurocup	Le Mans	Gran Canaria	2015-10-13	Regular Season	0.00	-2.38	-63.62	-5.35	6.29	...
Eurocup	Alba Berlin	Fuenlabrada	2016-10-12	Regular Season	0.00	11.80	45.17	50.67	7.28	...
Eurocup	Ulm	Tofas Bursa	2017-10-10	Regular Season	0.00	10.42	53.11	51.82	6.05	...
Euroleague	Baskonia	Neptunas	2014-10-15	Regular Season	0.00	9.37	53.19	51.22	7.73	...
Euroleague	Unicaja Malaga	Bamberg	2015-10-15	Regular Season	0.00	1.29	56.02	0.26	6.15	...
Euroleague	Real Madrid	Olympiacos	2016-10-12	Regular Season	13.40	0.82	7.62	1.28	8.46	...
Euroleague	CSKA Moscow	Armani Milano	2017-10-12	Regular Season	29.50	5.53	10.79	4.58	5.42	...
Liga ACB	Real Betis	Tenerife	2014-10-04	Regular Season	14.00	2.31	-2.73	2.56	6.58	...
Liga ACB	Gran Canaria	Gipuzkoa	2015-10-10	Regular Season	4.00	3.84	9.14	3.51	9.95	...
Liga ACB	Estudiantes	Real Betis	2016-10-01	Regular Season	0.67	0.56	-2.92	-9.43	7.51	...
Liga ACB	Gran Canaria	Gipuzkoa	2017-10-01	Regular Season	13.67	6.15	62.25	0.91	8.97	...

Table E.2: Sample of features, first match of every season per tournament

F Machine Learning Algorithms

F.1 Logistic Regression with Regularization

In logistic regression we model the probability of winning for the home team for a given set of values $\mathbf{x}$ of predictors/features $\mathbf{X} = (X_1, \ldots, X_p)$ (explained in Section 2.3). Specifically, a typical logistic regression model is summarized by

$$Y_i \sim \text{Bernoulli}(\pi_i) \text{ with } \log \frac{\pi_i}{1-\pi_i} = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}$$

where Y_i is a binary random variable that takes the value of one for the win of the home team in game i or the value of zero for its loss, π_i is the corresponding probability of a win for the home team, X_{ij} for $j = 1, \ldots, 110$ are the values of the predictors/features described in Section 2.3 for i game, β_0 is an overall constant parameter, and β_j for $j = 1, \ldots, 110$ are the corresponding coefficients measuring the effect of each predictor on the log odds of a win for the home team.

Typically, the model coefficients are estimated by taking the maximum likelihood estimates but here we considered the regularized versions of it by using either ridge or lasso regression estimates of them. The selection depends on the optimal solution suggested by K-fold CV for the tuning of the shrinkage parameter in the two approaches. Under this approach, the aim is to maximize the following penalized maximum log-likelihood function 7,

$$J(\boldsymbol{\beta}) = \beta_0 \sum_{i=1}^{n} y_i + \sum_{i=1}^{n} \sum_{j=1}^{p} y_i \beta_j X_{ij} + \sum_{i=1}^{n} \log \left\{ 1 + \exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} \right) \right\} + \lambda ||\boldsymbol{\beta}||_k, \tag{7}$$

where $||\boldsymbol{\beta}||_k = \sum_{j=1}^{p} |\beta_j|^k$; for $k = 1$ we have the l_1 norm and the lasso method and for $k = 2$ we have the l_2 norm and the ridge regression approach.

We maximize the penalized log-likelihood of the logistic regression with LIBLINEAR implementation (see Fan et al. (2008)) in scikit-learn library which apply a trust region Newton method (see Lin et al. (2007)).

Through the parameter λ we can control the impact of the regularization term. Higher values lead to smaller coefficients and lower model complexity. Careful tuning and specification of λ are needed since very high values may lead to under-fitted models while very small values may lead to over-fitted models.

F.2 Random Forest

Random Forest is essentially a method that combines inferences by multiple optimal decision trees obtained by bootstrap subsamples. Hence, a decision tree (see Li et al. (1984)) is the main ingredient of a Random Forest (see Breiman (2001)). Decision trees similarly make classifications to implementing a real life sequence of queries about the available data until we arrive at a final decision (or here prediction). The final form of the queries and the implied trees is specified using different mathematical algorithms. For the CART algorithm, which is the most popular one, a decision tree is built by determining a sequence of binary (yes/no) questions (called splits of nodes) that, when answered, lead to an improvement of a prediction measure such as the Gini Impurity. The Gini Impurity of a node is the probability of misclassification for a randomly chosen observation or individual in this node. In order to obtain the Gini Impurity, we first need to obtain the probability of winning of the home team for all games in $\mathcal{C}_m$ for any node m of a decision tree. The set $\mathcal{C}_m$ defines all the games with the characteristics/features splits defined by node m. Hence, the proportion of wins for node m is given by

$$P_i(m) = P(Y_i = 1|\mathbf{X} = \mathbf{x}_i : i \in \mathcal{C}_m) = \frac{1}{n_m}\sum_{i=1}^{n_m}\mathcal{I}(y_i = 1)\mathcal{I}(i \in \mathcal{C}_m) \text{ with } n_m = \sum_{i=1}^{n_m}\mathcal{I}(i \in \mathcal{C}_m)\ . \tag{8}$$

As a result, the Gini impurity of node m is given by:

$$I_G(m) = 2P_i(m)\big(1 - P_i(m)\big) \tag{9}$$

Finally, we classify each observation $i \in \mathcal{C}_m$ as a win for the home team if $P_i(m) > 0.5$ otherwise we classify it as a loss for the home team.

As we already mentioned, in random forests we consider different bootstrap sub-samples for training but in each sub-sample, we also consider a reduced number of features when looking for the best split, this reduced number of covariates which is usually set equal to the $\sqrt{p}$ or $\log_2(p)$ (reminder: p is the total number of features we consider – here $p = 110$). The selection of the different number of features in each tree aims in reducing the correlation between the optimal trees obtained by each sub-sample, resulting in variance reduction of the overall prediction (see Hastie et al. (2009)). The hyperparameters in Random Forests are

(a) the number of trees we consider,

(b) the number of features we consider (for the best split) in every sub-sample,

(c) the maximum number of levels in each tree (i.e. how many sequential splits we are going to impose on our features), and

(d) the minimum number of observations/individuals required at each node (essentially we need to specify two parameters here: one for the initial nodes and one for the terminal nodes/leafs; these parameters are labeled as `min_samples_split` and `min_samples_leaf`, respectively, in the scikit-learn implementation).

F.3 Extreme Gradient Boosting

Boosting is a method of converting weak learners into strong learners. The method combines the outputs of many "weak" classifiers to produce a powerful "committee". Extreme Gradient Boosting (XGBoost) Chen & Guestrin (2016), is a novel classifier based on an ensemble of classification trees (CART). In XGBoost, the trees are optimized using gradient boosting (see Friedman (2001)).

Let us consider a tree with a prediction score, for a set of covariates $\mathbf{x}$, given by $f(\mathbf{x}) = w_{z(\mathbf{x})}$; where $\mathbf{x}$ is the vector of features, z is a function assigning each data point to the corresponding leaf of a given tree, $z(\mathbf{x})$ is the specific leaf defined by $\mathbf{x}$ for this tree and $w_{z(\mathbf{x})}$ is the corresponding prediction score of the specific leaf $z(\mathbf{x})$ of the same tree. Generally $\mathbf{w} = (w_1, \ldots, w_T)$ is the set of tree weights (prediction scores) for leafs $1, \ldots, T$, respectively, for this specific tree under consideration. Here we consider an ensemble output of all $b = 1, \ldots, B$ trees hence we rewrite this expression as $f_b(\mathbf{x}) = w^{(b)}_{z_b(\mathbf{x})}$ in order to define the b specific tree quantities. Moreover, the prediction based on these B trees is given by

$$\hat{y_i} = \sum_{b=1}^{B} f_b(\mathbf{x}_i).\mathbf{F}, \tag{10}$$

The XGBoost algorithm tries to find the best vectors of weights $\mathbf{w}^{(b)}$ for each tree b by minimizing the loss function

$$\ell(t) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{b=1}^{B} \Omega(f_b) \tag{11}$$

where the first term contains the train loss function l which in our case is the logistic loss given by

$$l(y_i, \hat{y}_i) = -\ln f_{Bin}\left(y_i, \pi_i = \frac{e^{\hat{y}_i}}{1+e^{\hat{y}_i}}\right) = y_i \ln(1+e^{-\hat{y}_i}) + (1-y_i)\ln(1+e^{\hat{y}_i}), \tag{12}$$

between the observed score/class y_i and the predicted one $\hat{y}_i$ for each $i = 1, \ldots, n$ games. The second term in (11) is the regularization term, which controls the complexity of the model and helps to avoid overfitting. In XGBoost, the complexity is defined as:

$$\Omega(f_b) = \gamma T_b + \frac{1}{2}\lambda \sum_{t=1}^{T} \left\{w_t^{(b)}\right\}^2 \tag{13}$$

where T_b is the number of leaves of tree b, γ is the pseudo-regularization hyperparameter, depending on each data-set and λ is the shrinkage tuning parameter controlling the regularization of the ridge in the methodology.

The optimization/learning procedure is performed in an additive manner by optimizing the first tree in the first round of the algorithm, then optimizing the second added tree conditional on the values of the first and so on, each tree added modifies the overall model but the magnitude of the modification is controlled by a shrinkage parameter ν which is called the "learning rate". To simplify the procedure, a second order Taylor expansion is used in the loss function (11) in order to find the optimal weights (see Chen & Guestrin (2016)).

XGBoost identifies the shortcomings of weak learners (decision trees) by using high weight data points and gradients in the loss function. The loss function is a measure indicating how good is the model concerning the fit of the underlying data. The method requires a considerable number of parameters that must be tuned. The hyperparameters in this model that we need to tune are

(a) the learning rate that shrinks the contribution of each tree in order to prevent overfitting (ν),

(b) the number of trees (B),

(c) the maximum depth of each tree (and therefore the maximum number of leaves $T > T_b$ for all $b = 1, \ldots, B$),

(d) the percentage of data points taken to build each tree,

(e) number of features used by each tree,

(f) minimum loss reduction required to make a further partition on a leaf of the tree (γ the pseudo-regularization),

(g) L_2 regularization term on weights (parameter λ).

G Evaluation Metrics

G.1 Brier Score

Brier score (BS), Brier et al. (1950), which is a special case of Ranked Probability Score (RPS) (Epstein (1969)) when using binary outcomes. On our occasion, the BS (or RPS) is given by:

$$BS(\pi; \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}(\pi_i - y_i)^2 = \frac{1}{n}\sum_{i=1}^{n}\left[(1-\pi_i)^{y_i}(\pi_i)^{1-y_i}\right]^2 \tag{14}$$

for a given set of prediction probabilities $\pi = (\pi_1, \ldots, \pi_n)$ and observed binary data $\mathbf{y} = (y_1, \ldots, y_n)$; where π_i is the probability of a win for the home team in i game and y_i is the observed value for the event of win of the home in i game.

G.2 Accuracy

The accuracy is given by

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}(\pi_i > 0.5) \tag{15}$$

and is simply the proportion of correct predictions over the total number of games n we consider; where $\mathcal{I}(A)$ is the indicator function taking the value of one when condition A is true and zero otherwise. Here we classify our final predictions using the threshold of 0.5 for the prediction probability π_i.

G.3 F_1-score

The F_1-score (see Van Rijsbergen (1979)) is the harmonic mean of the precision (or positive predictive value) and the recall (or sensitivity) measures. Hence the F_1 is given by

$$F_1 = \frac{1}{\frac{1}{2}\left(\text{Precision}^{-1} + \text{Recall}^{-1}\right)} = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{16}$$

where Precision (or positive predictive value) is the proportion of games with correct predicted home wins over the sum of the total games with predicted home wins. Equivalently, the recall (or sensitivity) is the proportion of games with correct predicted home wins over the number of games of actual home wins. Hence, they are given by

$$\text{Precision} = \frac{\sum_{i=1}^{n} \mathcal{I}(\pi_i > 0.5)\mathcal{I}(y_i = 1)}{\sum_{i=1}^{n} \mathcal{I}(\pi_i > 0.5)} \tag{17}$$

$$\text{Recall} = \frac{\sum_{i=1}^{n} \mathcal{I}(\pi_i > 0.5)\mathcal{I}(y_i = 1)}{\sum_{i=1}^{n} \mathcal{I}(y_i = 1)}. \tag{18}$$

From the above equations, we can rewrite F_1 as:

$$F_1 = 2\,\frac{\sum_{i=1}^{n} \mathcal{I}(\pi_i > 0.5)\mathcal{I}(y_i = 1)}{\sum_{i=1}^{n} \mathcal{I}(\pi_i > 0.5) + \sum_{i=1}^{n} \mathcal{I}(y_i = 1)}. \tag{19}$$

H Benchmarks

H.1 Predictions based on Rating Systems

Tournament/League	**Accuracy**		
Rating Systems	**Pi-rating**	**PagaRank**	**Elo**
Euroleague	0.662	0.612	0.581
Eurocup	0.647	0.647	0.636
Greek League	0.745	0.755	0.725
Liga ACB	0.694	0.602	0.627

Results are obtained by using rating systems for prediction (team with the higher rating/rank wins) and they are evaluated in season 2017–2018

Table H.1.1: Accuracy of rating systems

H.2 Predictions based on Oliver's four factors

Tournament/League	**Accuracy**
Euroleague	0.627
Eurocup	0.620
Greek League	0.750
Liga ACB	0.645

Results are obtained by using the average of the last 10 matches of Oliver's four factors of both teams for prediction of winner and they are evaluated in accuracy for the season 2017–2018

Table H.2.1: Accuracy of Oliver's four factors

H.3 Climatology model home advantage 55-65%

Evaluating Full Information Model predictions with climatology model based on home advantage between 55-65%, for full season implementation by calculating the Brier Skill Score given by:

$$Brier\ Skill\ Score = 1 - \frac{Brier\ Score\ of\ full\ information\ model}{Brier\ Score\ of\ reference\ home\ advantage} \tag{20}$$

Tournament/League	55%				60%				65%			
Models	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**
Euroleague	0.080	0.126	0.130	0.124	0.056	0.103	0.107	0.101	0.053	0.100	0.104	0.097
Eurocup	0.111	0.154	0.137	0.138	0.102	0.146	0.128	0.129	0.112	0.156	0.138	0.139
Greek League	0.393	0.374	0.373	0.393	0.378	0.358	0.358	0.379	0.376	0.356	0.356	0.377
Liga ACB	0.180	0.175	0.171	0.185	0.170	0.164	0.160	0.175	0.176	0.171	0.167	0.182

Results are obtained using 2014–2017 data for training and 2017–2018 for validations

Table H.3.1: Climatology model for full season implementation

H.4 Baseline Vanilla Models

Home Effect	Euroleague	Eurocup	Greek League	Liga ACB
Intercept of LR / exp($Intercept$)	0.817 / 2.264	0.662 / 1.940	0.632 / 1.880	0.547 / 1.730

Table H.4.1: Estimated common home effect of the standard Baseline Vanilla Logistic Regression (LR) Model
(All baseline vanilla logistic regression models are regularised providing as a byproduct a group of teams serving as reference)

Tournament/League	Brier Score				Accuracy				F_1			
Models	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**
Euroleague	0.215	0.213	0.215	0.213	0.650	0.665	0.650	0.650	0.745	0.767	0.756	0.753
Eurocup	0.232	0.237	0.238	0.232	0.636	0.636	0.625	0.625	0.717	0.729	0.723	0.721
Greek League	0.167	0.171	0.181	0.171	0.735	0.745	0.735	0.745	0.806	0.814	0.804	0.814
Liga ACB	0.208	0.208	0.218	0.209	0.682	0.688	0.648	0.679	0.764	0.769	0.733	0.762

Results are obtained using 2014–2017 data for training and 2017–2018 for validations

(a) Full season implementation

Tournament/League	Brier Score				Accuracy				F_1			
Models	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**
Euroleague	0.212	0.214	0.204	0.207	0.667	0.650	0.692	0.675	0.762	0.753	0.776	0.766
Eurocup	0.216	0.235	0.245	0.225	0.607	0.667	0.560	0.607	0.723	0.754	0.718	0.732
Greek League	0.189	0.171	0.179	0.175	0.703	0.725	0.736	0.747	0.809	0.809	0.812	0.827
Liga ACB	0.214	0.214	0.220	0.210	0.647	0.673	0.634	0.686	0.745	0.750	0.723	0.774

Results are obtained by using the data in the middle of regular season as a training set in order to validate our results with the data of the rest of the regular season (for season 2017–2018).

(b) Mid-season implementation

Tournament/League	Brier Score				Accuracy				F_1			
Models	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**
Euroleague	0.221	0.283	0.260	0.246	0.700	0.600	0.650	0.650	0.800	0.692	0.759	0.759
Eurocup	0.206	0.168	0.213	0.189	0.688	0.750	0.688	0.750	0.762	0.833	0.762	0.818
Greek League	0.160	0.197	0.202	0.169	0.727	0.727	0.636	0.773	0.786	0.769	0.750	0.815
Liga ACB	0.195	0.221	0.194	0.190	0.714	0.714	0.714	0.714	0.786	0.786	0.769	0.786

Results are obtained by using the data in the regular season as a training set in order to validate our results with the data in the play-off phase (for season 2017–2018).

(c) Play-off implementations

(All baseline vanilla logistic regression models are regularised providing as a byproduct a group of teams serving as reference)
(Abbreviations: LR: Logistic Regression; RF: Random Forrest; XGB: Extreme gradient boosting; EL: Ensemble learning)

Table H.4.2: Evaluation metrics for the Baseline Vanilla Model

I Full Information Models

Tournament/League	Brier Score				Accuracy				F_1			
Models	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**
Euroleague	0.220	0.209	0.208	0.209	0.662	0.665	0.692	0.681	0.770	0.749	0.778	0.774
Eurocup	0.216	0.205	0.210	0.209	0.641	0.668	0.690	0.668	0.748	0.761	0.759	0.753
Greek League	0.145	0.150	0.150	0.145	0.770	0.755	0.779	0.784	0.833	0.818	0.833	0.839
Liga ACB	0.198	0.200	0.201	0.197	0.697	0.713	0.709	0.719	0.763	0.783	0.784	0.788

Results are obtained by using 2014-2017 data for training and 2017-2018 for validations.

(a) Full season implementation

Tournament/League	Brier Score				Accuracy				F_1			
Models	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**
Euroleague	0.215	0.238	0.221	0.219	0.683	0.600	0.658	0.692	0.793	0.733	0.781	0.804
Eurocup	0.235	0.206	0.203	0.210	0.571	0.726	0.679	0.667	0.723	0.793	0.765	0.763
Greek League	0.146	0.151	0.171	0.152	0.780	0.747	0.747	0.780	0.841	0.813	0.824	0.841
Liga ACB	0.205	0.220	0.223	0.209	0.712	0.686	0.614	0.693	0.798	0.769	0.751	0.789

Results are obtained by using the data in the middle of regular season as a training set in order to validate our results with the data of the rest of the regular season (for season 2017-2018).

(b) Mid-season implementation

Tournament/League	Brier Score				Accuracy				F_1			
Models	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**
Euroleague	0.226	0.192	0.206	0.204	0.650	0.700	0.750	0.750	0.774	0.813	0.839	0.839
Eurocup	0.191	0.165	0.179	0.174	0.750	0.750	0.688	0.750	0.846	0.818	0.783	0.833
Greek League	0.148	0.132	0.125	0.132	0.818	0.773	0.864	0.818	0.857	0.815	0.889	0.857
Liga ACB	0.249	0.232	0.217	0.229	0.571	0.524	0.667	0.571	0.710	0.668	0.759	0.710

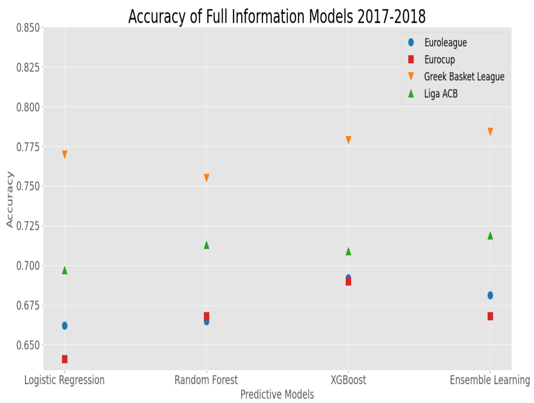
Results are obtained by using the data in the regular season as a training set in order to validate our results with the data in the play-off phase (for season 2017-2018).

(c) Play-off implementations

(Abbreviations: LR: Logistic Regression; RF: Random Forrest; XGB: Extreme gradient boosting; EL: Ensemble learning)

Table I.1: Evaluation metrics for the Full Information Model using all features

J Plots Full Information Models



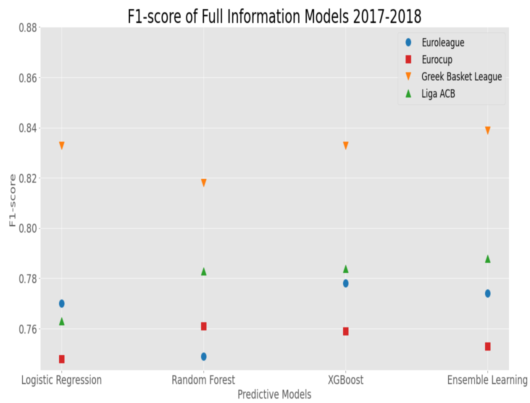


Figure J.1: Comparison of methods and algorithms in terms of accuracy and F_1 for Full Information Models for each tournament for the full season prediction scenario

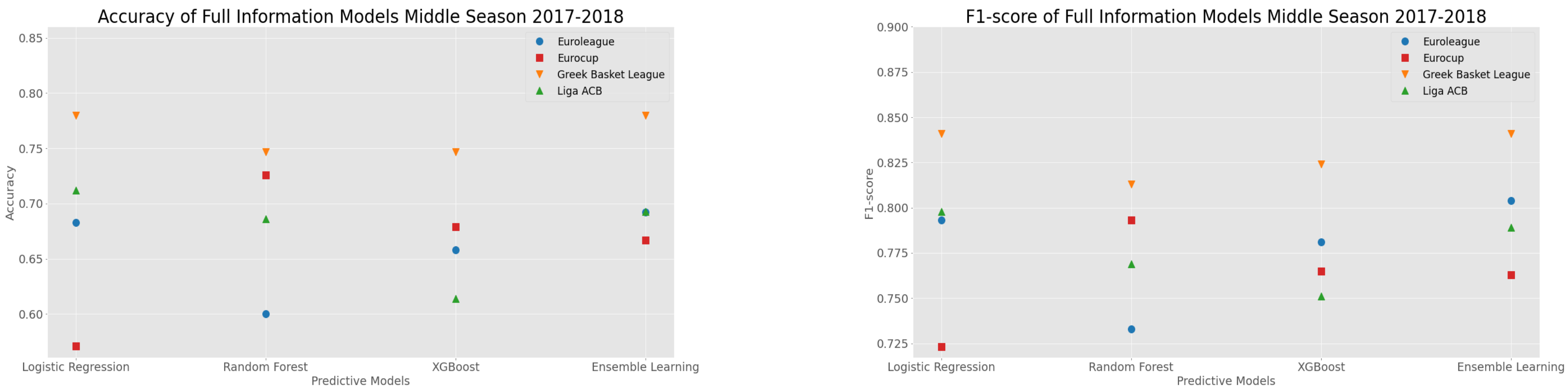


Figure J.2: Comparison of methods and algorithms in terms of accuracy and F_1 for Full Information Models for each tournament for the middle season prediction scenario

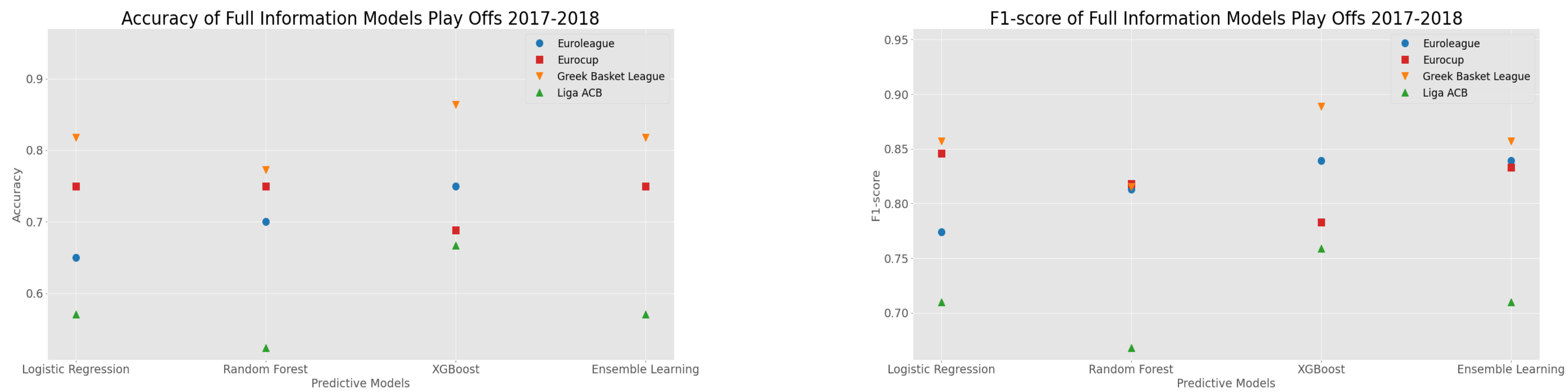


Figure J.3: Comparison of methods and algorithms in terms of accuracy and F_1 for Full Information Models for each tournament for the play-offs prediction scenario

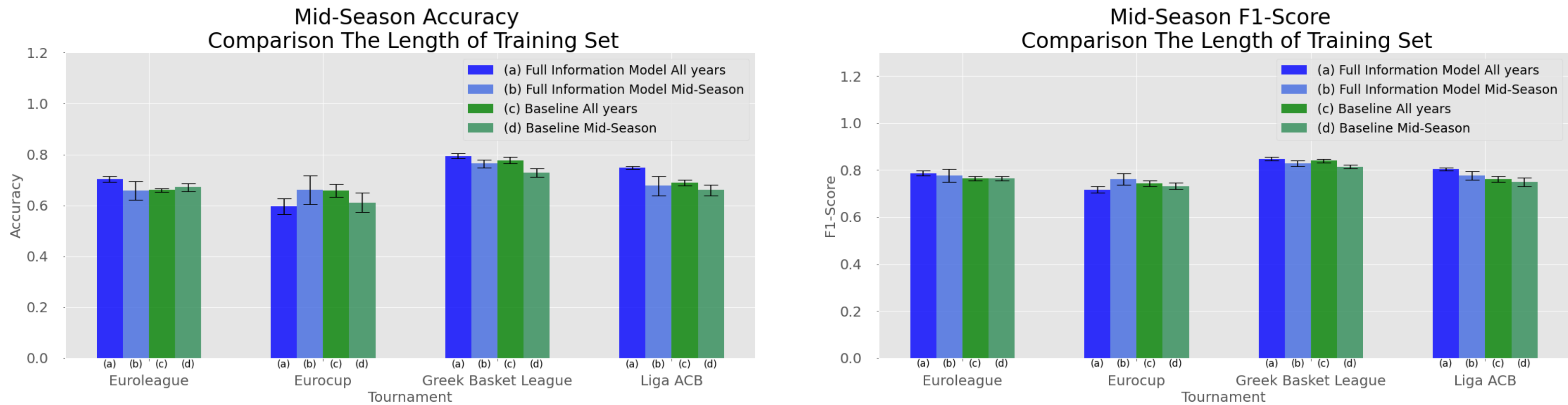


Figure J.4: Comparison of accuracy and F_1 performance in the mid-season scenario for the Full Information and Baseline Vanilla Models over different leagues and different set of training data-set (current mid-season vs. all previous games).

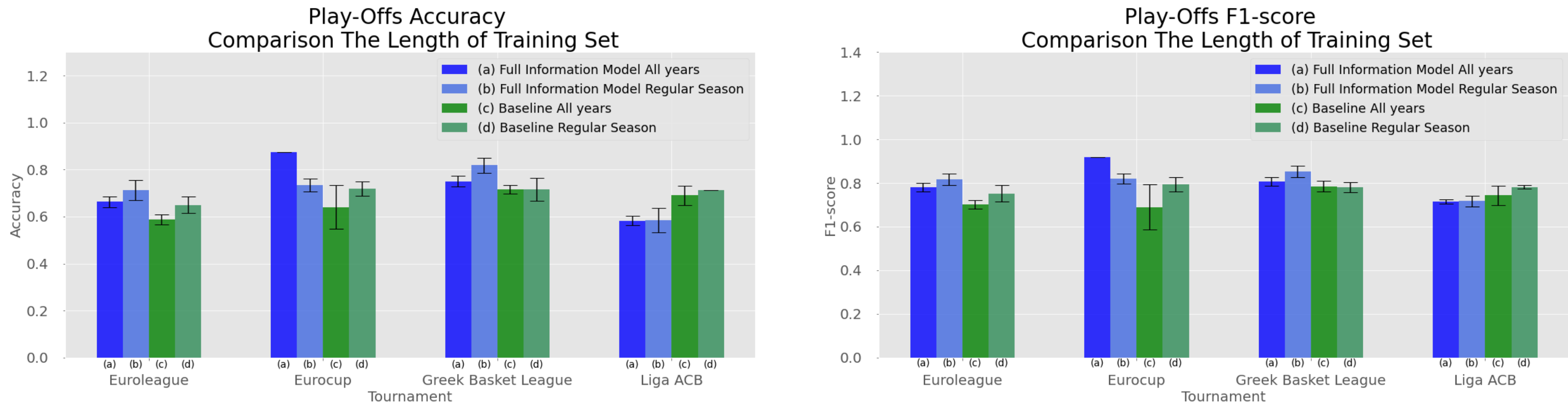


Figure J.5: Comparison of accuracy and F_1 performance in the play-offs scenario for the Full Information and Baseline Vanilla Models over different leagues and different set of training data-set (current mid-season vs. all previous games).

K Prediction of series of play-offs match-ups

Tournament/League	Baseline Vanilla Model				Full Information Model			
Models	**LR**	**RF**	**XGB**	**EL**	**LR**	**RF**	**XGB**	**EL**
Euroleague	0.750	0.500	0.625	0.625	0.625	0.750	0.625	0.750
Eurocup	0.714	0.571	0.714	0.714	0.429	0.714	0.571	0.571
Greek League	0.625	0.750	0.500	0.750	0.875	0.750	1.000	0.875
Liga ACB	0.714	0.857	0.714	0.714	0.571	0.429	0.714	0.571

Results are obtained by predicting the series of play-offs match-ups with Baseline Vanilla Model and Full Information Model over different leagues.

(Abbreviations: LR: Logistic Regression; RF: Random Forrest; XGB: Extreme gradient boosting; EL: Ensemble learning)

Table K.1: Accuracy of predictions of series of play-offs match-ups

L FEATURE IMPORTANCE

Features	Euroleague	Eurocup	Greek League	Liga ACB	Overall Mean
pi_ratings	0.56	1.0	0.78	0.89	0.81
PageRank	0.78	0.78	1.0	0.56	0.78
Current_form_EDiff	0.67	1.0	0.89	0.33	0.72
Current_form_Game_Score_received	0.56	1.0	1.0	0.22	0.69
Current_form_pointsdiff	0.44	1.0	0.78	0.56	0.69
Current_form_FIC	0.67	0.67	0.78	0.56	0.67
history_Game_Score	0.56	0.89	0.78	0.44	0.67
history_FIC	0.44	0.78	0.78	0.67	0.67
Current_form_Play	0.56	0.67	0.67	0.67	0.64
Current_form_Performance _Index _received	0.33	1.00	0.78	0.44	0.64
history_Play_received	0.67	0.89	0.78	0.22	0.64
Current_form_EFG_received_sd	0.44	0.89	0.67	0.56	0.64
history_Ediff	0.56	0.78	0.89	0.33	0.64
tradition_pointsdiff_general	0.33	0.67	0.78	0.67	0.61
history_pointsdiff	0.44	0.78	0.78	0.44	0.61
tradition_pointsdiff_match	0.67	0.56	0.67	0.56	0.61
Current_form_FIC_received_sd	0.67	0.44	0.78	0.56	0.61
Current_form_elo	0.33	0.89	0.78	0.44	0.61
elo	0.44	1.0	0.89	0.11	0.61
history_EFG	0.67	0.44	0.89	0.33	0.58
Current_form_EDiff_sd	0.44	0.89	0.78	0.22	0.58
Current_form_Play_sd	0.44	0.78	0.78	0.33	0.58
Current_form_Ortg	0.44	0.56	0.78	0.56	0.58
history_TS_received	0.44	0.78	0.78	0.33	0.58
history_Points	0.33	0.67	0.78	0.56	0.58
Current_form_FIC_received	0.33	1.0	0.67	0.33	0.58
history_Ortg	0.67	0.56	0.78	0.33	0.58
history_Drtg	0.44	0.67	0.78	0.44	0.58
Current_form_Points	0.44	0.78	0.67	0.44	0.58
history_winner	0.56	0.67	0.67	0.44	0.58
history_Ediff_sd	0.56	0.56	0.78	0.44	0.58
history_TS_sd	0.33	0.89	0.67	0.33	0.56
history_EFG_received	0.78	0.67	0.67	0.11	0.56
Current_form_winner	0.33	0.89	0.67	0.33	0.56
Current_form_TS_sd	0.44	0.89	0.67	0.22	0.56
Current_form_Drtg_sd	0.33	0.89	0.56	0.44	0.56
Current_form_Points_sd	0.56	0.89	0.67	0.11	0.56
tradition_winner_general	0.44	0.33	0.78	0.67	0.56
history_pointsdiff_sd	0.56	0.56	0.78	0.33	0.56
Current_form_Points_received	0.44	0.78	0.67	0.22	0.53
history_Play	0.44	0.56	0.78	0.33	0.53
history_Points_received	0.44	0.78	0.67	0.22	0.53
history_Play_received_sd	0.56	0.67	0.67	0.22	0.53
history_Performance_Index	0.67	0.78	0.67	0.0	0.53
history_Points_received_sd	0.44	0.67	0.67	0.33	0.53
Current_form_Performance _Index_sd	0.56	0.67	0.78	0.11	0.53
Current_form_TS	0.56	0.89	0.67	0.0	0.53
History_Performance _Index_received	0.44	0.78	0.78	0.11	0.53
Current_form_Drtg	0.44	0.78	0.78	0.11	0.53
tradition_Ediff_general	0.56	0.56	0.67	0.33	0.53
Current_form_pointsdiff_sd	0.44	0.89	0.67	0.11	0.53

Table L.1: Relative frequencies of feature importance across different prediction scenarios and implementations; Features are sorted according to the overall proportion of importance

M APPENDIX: DATA AND CODE

All data used in this article have been kindly provided to the authors by the Greek Organization of Football Prognostics (OPAP). Due to confidentiality reasons, we cannot publicly provide access to the actual data-set of this study. For this reason, we provide the code and an alternative data-set obtained via scrapping to the Git repository `https://tinyurl.com/Basketball-Machine-Learning` of the article. More specifically, in the Git repository, you can find two sets of code and files: one referring to the paper implementation (with no data available) and a second one with implementation to the crawled data obtained by `https://www.basketball-reference.com/`. For the crawled data-set we obtained results from eight tournaments including the ones presented in this work (Greek league, Liga ACB, Euroleague and Eurocup) for a period of five years: 2014/10/04-2020/06/30. The Git repository contains data, along with Python code and Jupyter notebooks for the pre-processing of the data and the tuning of the hyper-parameters for all algorithms. Moreover, two main modeling approaches have been implemented: one with Baseline Vanilla Model and a second one using the Full Information Model. For the analyses with the publicly available data, we have specified the training data-set by considering results from four seasons (2014–2018) while season 2018/19 was used for evaluating the prediction efficiency of the methods.

References

Ballı, S. & Özdemir, E. (2021), 'A novel method for prediction of euroleague game results using hybrid feature extraction and machine learning techniques', *Chaos, Solitons & Fractals* **150**, 111119.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**, 5–32.

Brier, G. W. et al. (1950), 'Verification of forecasts expressed in terms of probability', *Monthly weather review* **78**(1), 1–3.

Cai, W., Yu, D., Wu, Z., Du, X. & Zhou, T. (2019), 'A hybrid ensemble learning framework for basketball outcomes prediction', *Physica A: Statistical Mechanics and its Applications* **528**, 121461.

Carlin, B. P. (2005), Improved ncaa basketball tournament modeling via point spread and team strength information, *in* 'Anthology of Statistics in Sports', SIAM, pp. 149–153.

Chen, T. & Guestrin, C. (2016), Xgboost: A scalable tree boosting system, *in* 'Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining', pp. 785–794.

Constantinou, A. C. & Fenton, N. E. (2013), 'Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries', *Journal of Quantitative Analysis in Sports* **9**(1), 37–50.

Epstein, E. S. (1969), 'A scoring system for probability forecasts of ranked categories', *Journal of Applied Meteorology (1962-1982)* **8**(6), 985–987.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. & Lin, C.-J. (2008), 'Liblinear: A library for large linear classification', *the Journal of machine Learning research* **9**, 1871–1874.

Friedman, J. H. (2001), 'Greedy function approximation: a gradient boosting machine', *Annals of statistics* pp. 1189–1232.

García, J., Ibáñez, S. J., De Santos, R. M., Leite, N. & Sampaio, J. (2013), 'Identifying basketball performance indicators in regular season and playoff games', *Journal of human kinetics* **36**, 161.

Giasemidis, G. (2020), 'Descriptive and predictive analysis of euroleague basketball games and the wisdom of basketball crowds', *arXiv preprint arXiv:2002.08465* .

Gilovich, T., Vallone, R. & Tversky, A. (1985), 'The hot hand in basketball: On the misperception of random sequences', *Cognitive psychology* **17**(3), 295–314.

Harville, D. A. & Smith, M. H. (1994), 'The home-court advantage: How large is it, and does it vary from team to team?', *The American Statistician* **48**(1), 22–28.

Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Vol. 2, Springer.

Heit, E., Price, P. C. & Bower, G. H. (1994), 'A model for predicting the outcomes of basketball games', *Applied cognitive psychology* **8**(7), 621–639.

Hollinger, J. (2002), *Pro Basketball Prospectus*, Potomac Books.

Hollinger, J. (2005), *Pro Basketball Forecast*, Potomac Books.

Horvat, T., Job, J. & Medved, V. (2018), Prediction of euroleague games based on supervised classification algorithm k-nearest neighbours, *in* '6th International Congress on Support Sciences Research and Technology Support', Vol. 20, p. 21.

Hubáček, O., Šourek, G. & Železnỳ, F. (2019), 'Learning to predict soccer results from relational data with gradient boosted trees', *Machine Learning* **108**, 29–47.

Hvattum, L. M. & Arntzen, H. (2010), 'Using elo ratings for match result prediction in association football', *International Journal of forecasting* **26**(3), 460–470.

Kohavi, R. & John, G. H. (1997), 'Wrappers for feature subset selection', *Artificial intelligence* **97**(1-2), 273–324.

Kubatko, J., Oliver, D., Pelton, K. & Rosenbaum, D. T. (2007), 'A starting point for analyzing basketball statistics', *Journal of quantitative analysis in sports* **3**(3).

Lazova, V. & Basnarkov, L. (2015), 'Pagerank approach to ranking national football teams', *arXiv preprint arXiv:1503.01331* .

Li, B., Friedman, J., Olshen, R. & Stone, C. (1984), 'Classification and regression trees (cart)', *Biometrics* **40**(3), 358–361.

Lin, C.-J., Weng, R. C. & Keerthi, S. S. (2007), Trust region newton methods for large-scale logistic regression, *in* 'Proceedings of the 24th international conference on Machine learning', pp. 561–568.

Loeffelholz, B., Bednar, E. & Bauer, K. W. (2009), 'Predicting nba games using neural networks', *Journal of Quantitative Analysis in Sports* **5**(1).

Milanović, D., Selmanović, A. & Škegro, D. (2014), Characteristics and differences of basic types of offenses in european and american top-level basketball, *in* '7th International Scientific Conference on Kinesiology', p. 400.

Murphy, A. H. (1973), 'Hedging and skill scores for probability forecasts', *Journal of Applied Meteorology and Climatology* **12**(1), 215–223.

Naismith, J. (1941), 'Basketball: Its Origin and Development', *New York, Association Press* .

Oliver, D. (2004), *Basketball on paper: rules and tools for performance analysis*, Potomac Books, Inc.

Page, L., Brin, S., Motwani, R. & Winograd, T. (1999), The pagerank citation ranking: Bringing order to the web., Technical report, Stanford InfoLab.

Schwertman, N. C., McCready, T. A. & Howard, L. (1991), 'Probability models for the ncaa regional basketball tournaments', *The American Statistician* **45**(1), 35–38.

Shi, Z., Moorthy, S. & Zimmermann, A. (2013), Predicting ncaab match outcomes using ml techniques–some results and lessons learned, *in* 'ECML/PKDD 2013 Workshop on Machine Learning and Data Mining for Sports Analytics'.

Smith, T. & Schwertman, N. C. (1999), 'Can the ncaa basketball tournament seeding be used to predict margin of victory?', *The american statistician* **53**(2), 94–98.

Stefani, R. T. (1980), 'Improved least squares football, basketball, and soccer predictions', *IEEE transactions on systems, man, and cybernetics* **10**(2), 116–123.

Torres, R. A. (2013), 'Prediction of nba games based on machine learning methods', *University of Wisconsin, Madison* .

Van Rijsbergen, C. J. (1979), 'Information Retrieval, 2nd edition', *Butterworths* .

Zimmermann, A. (2016), 'Basketball predictions in the ncaab and nba: Similarities and differences', *Statistical Analysis and Data Mining: The ASA Data Science Journal* **9**(5), 350–364.