Team Prediction and MVP Prediction of NBA

Stat4011 Group 13

01

INTRODUCTION

Objective

- When a new competition starts, we can predict which team wins
- Which players win the NBA MVP at end of season
- Find the best method of above

Point Overview

Winning team prediction

Based on the performance difference of two teams

The probability of home team or away team wins from different competition is predicted. (`pX_test`)

3 methods are used for finding a method with the highest performance

Logistic Regression, Random Forest, K nearest neighbors

Point Overview

MVP prediction

Based on the performance of each players

The share of the players is predicted.(`predictions`)

Share: The overall contribution of the players to their team

4 methods are used for finding a method with the highest performance

Ridge Regression, Random Forest, Standard Scaler, Extra Trees

02

HOME TEAM WIN PREDICTIONS

Data Collection

Web scraping from **ESPN**.com

about the 2025 NBA Season

Data Collection

Team Standing

Win-Losses,
Points per game
...

Game Result

Home_score,
Away_score,
For each match

Player Stats

Names,
Game played,
Offensive rebounds per game
...

web-scraping

Cleveland Cavaliers Schedule 2024-25

2024–25 ⌄ | Regular Season ⌄

Regular Season

DATE	OPPONENT	RESULT	W-L	HI POINTS	HI
Thu, 24 Oct	@ Toronto	W 136-106	1-0	Mobley 25	Mo
Sat, 26 Oct	vs Detroit	W 113-101	2-0	Wade 19	All
Sun, 27 Oct	@ Washington	W 135-116	3-0	Mitchell 30	Mo
Tue, 29 Oct	@ New York	W 110-104	4-0	Garland 34	
Thu, 31 Oct	vs Los Angeles	W 134-110	5-0	Mobley 25	All
Sat, 2 Nov	vs Orlando	W 120-109	6-0	Garland 25	Mo
Sun, 3 Nov	@ Milwaukee	W 114-113	7-0	Mitchell 30	All
Tue, 5 Nov	vs Milwaukee	W 116-114	8-0	Garland 39	All
Thu, 7 Nov	@ New Orleans	W 131-122	9-0	Mitchell 29	All
Sat, 9 Nov	vs Golden State	W 136-117	10-0	Garland 27	All
Sun, 10 Nov	vs Brooklyn	W 105-100	11-0	Mobley 23	Mo
Tue, 12 Nov	@ Chicago	W 119-113	12-0	Mitchell 36	Mo
Thu, 14 Nov	@ Philadelphia	W 114-106	13-0	Garland 25	Mi

```
game_results(result_links[0],result_links,club_table)
```

	home_team	away_team	home_score	away_score	record
0	Toronto	Cleveland Cavaliers	106	136	1–0
1	Cleveland Cavaliers	Detroit	113	101	2–0
2	Washington	Cleveland Cavaliers	116	135	3–0
3	New York	Cleveland Cavaliers	104	110	4–0
4	Cleveland Cavaliers	Los Angeles	134	110	5–0
5	Cleveland Cavaliers	Orlando	120	109	6–0
6	Milwaukee	Cleveland Cavaliers	113	114	7–0
7	Cleveland Cavaliers	Milwaukee	116	114	8–0
8	New Orleans	Cleveland Cavaliers	122	131	9–0
9	Cleveland Cavaliers	Golden State	136	117	10–0
10	Cleveland Cavaliers	Brooklyn	105	100	11–0
11	Chicago	Cleveland Cavaliers	113	119	12–0
12	Philadelphia	Cleveland Cavaliers	106	114	13–0

Memphis Grizzlies Stats 2024-25

2024–25 Regular Season | All Splits

Team Leaders

Points	Rebounds	Assists	Steals	Blocks
J. Jackson Jr. PF 22.9	S. Aldama PF 7.2	S. Pippen Jr. G 5.9	J. Jackson Jr. PF 1.3	J. 1

Player Stats - All Splits

NAME	GP	GS	MIN	PTS	OR	DR	REB	AST	STL	BLK	TO	PF	AST/TO
J. Jackson Jr. PF	15	15	28.7	22.9	1.3	4.3							
J. Morant PG	8	8	27.9	20.6	0.5	4.5							
D. Bane SG	10	9	28.2	16.2	1.0	5.6							
S. Aldama PF	17	11	28.2	12.7	1.6	5.6							
S. Pippen Jr. G	17	8	26.0	12.6	0.6	3.2							

```
s['PlayerStats'].head()
```

	Name	Game Played	Game Started	Minutes Per Game	Points Per Game	Offensive Rebounds Per Game	Defensive Rebounds Per Game	Rebounds Per Game	Assists Per Game	Steals Per Game	Blocks Per Game	Turnov Per G
0	Jaren Jackson Jr. PF	15	15.0	28.7	22.9	1.3	4.3	5.6	1.3	1.3	1.7	
1	Ja Morant PG	8	8.0	27.9	20.6	0.5	4.5	5.0	9.1	0.6	0.3	
2	Desmond Bane SG	10	9.0	28.2	16.2	1.0	5.6	6.6	3.6	0.9	0.6	
3	Santi Aldama PF	17	11.0	28.2	12.7	1.6	5.6	7.2	3.5	0.9	0.5	
4	Scotty Pippen Jr. G	17	8.0	26.0	12.6	0.6	3.2	3.9	5.9	1.1	0.4	

Data Cleaning & Preprocessing

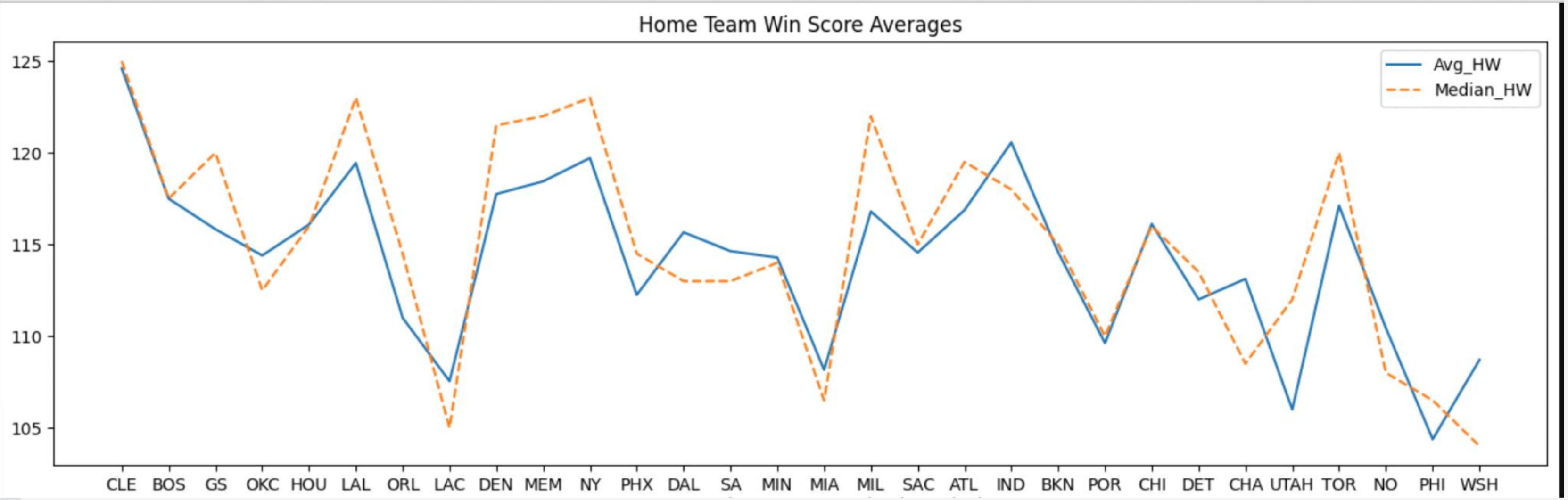
Home_stat

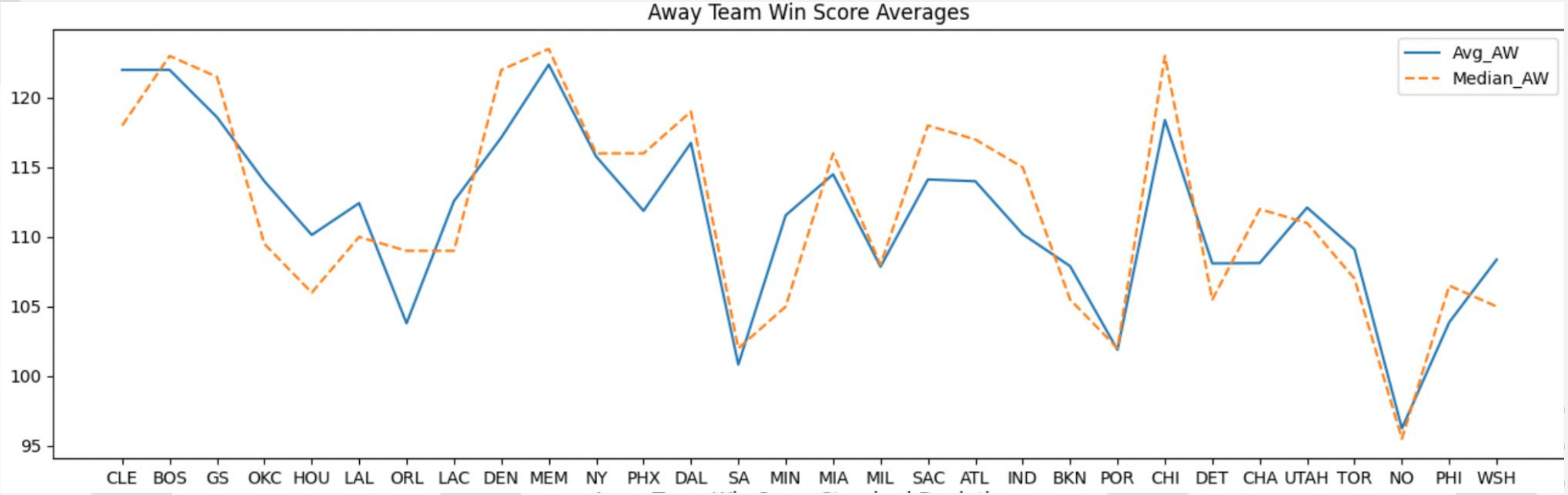
Avg home win / loss score
Median home win / loss score
SD of home win / loss score

Away_stat

Avg away win / loss score
Median away win / loss score
SD of away win / loss score

Club	Avg_Home_Win_Score	Median_Home_Win_Score	STD_Home_Win_Score	Avg_Home_Loss_Score	Median_Home_Loss_Score	STD_Home_Loss_Score
CLE	124.600000	125.0	11.128342	109.900000	109.5	7.892401
BOS	117.500000	117.5	8.093207	112.625000	113.0	6.537153
GS	115.833333	120.0	8.493462	106.500000	109.0	10.563301
OKC	114.400000	112.5	11.429786	103.600000	101.5	16.026229
HOU	116.090909	116.0	10.299643	104.181818	104.0	10.124841
LAL	119.444444	123.0	8.564930	115.777778	118.0	9.077173


Home Team Win Score Averages
Avg_HW
Median_HW
125
120
115
110
105
CLE BOS GS OKC HOU LAL ORL LAC DEN MEM NY PHX DAL SA MIN MIA MIL SAC ATL IND BKN POR CHI DET CHA UTAH TOR NO PHI WSH


Away Team Win Score Averages
Avg_AW
Median_AW
120
115
110
105
100
95
CLE BOS GS OKC HOU LAL ORL LAC DEN MEM NY PHX DAL SA MIN MIA MIL SAC ATL IND BKN POR CHI DET CHA UTAH TOR NO PHI WSH

Player & Shooting Statistics

Player Stats

‘Avg_Points' ,
'Avg_Rebounds',
'Avg_Assists',
'Avg_Steals',
'Avg_Blocks',
'Avg_Turnovers',

Shooting Stats

'Avg_FieldGoal_Percentage',
'Avg_ThreePoint_Percentage',
'Avg_FreeThrow_Percentage',
'Avg_TwoPoint_Percentage',
'Avg_Scoring_Efficiency',
'Avg_Shooting_Efficiency'

Player Stats - All Splits

NAME	GP	GS	MIN	PTS	OR	DR	REB	AST	STL	BLK	TO	PF	AST/TO
Jayson Tatum SF	17	17	36.7	28.9	0.4	7.8	8.2	5.9	1.5	0.5	2.8	2.7	2.1
Jaylen Brown SG	13	13	36.8	25.6	1.5	5.2	6.7	4.4	1.0	0.4	2.4	3.1	1.8
Derrick White PG	17	17	35.1	18.4	1.2	3.5	4.7	4.3	0.8	1.2	1.5	2.4	2.8
Payton Pritchard PG	17	0	27.4	14.5	0.8	1.8	2.6	2.9	1.1	0.1	1.0	1.9	2.9
Jrue Holiday PG	16	16	30.8	12.6	0.9	3.2	4.1	3.8	0.8	0.3	1.6	1.4	2.4

Team_Stat

Home_stats

+

Away_stats

+

Player_stats

+

Shooting_stats

Team Statistics → Player Statistics → Shooting Statistics → Team Statistics

Feature Difference

- calculates the **differences** in various **performance metrics between the home and away teams**
- creating new features that represent the **relative strengths and weaknesses** of the teams.

Home team features – Away team features

Why Feature Difference

Improved Model Interpretability

Capturing relative strengths and weaknesses

Focus on Relevant Information

Feature Reduction

Alignment with Predictive Goals

Improving model accuracy

Define Winning logic

Defined the **winning_logic** function to analyze game results for a specific team, determine whether the home team won each game based on the score difference

Home team score - Away team score >1
Return 1
Otherwise 0

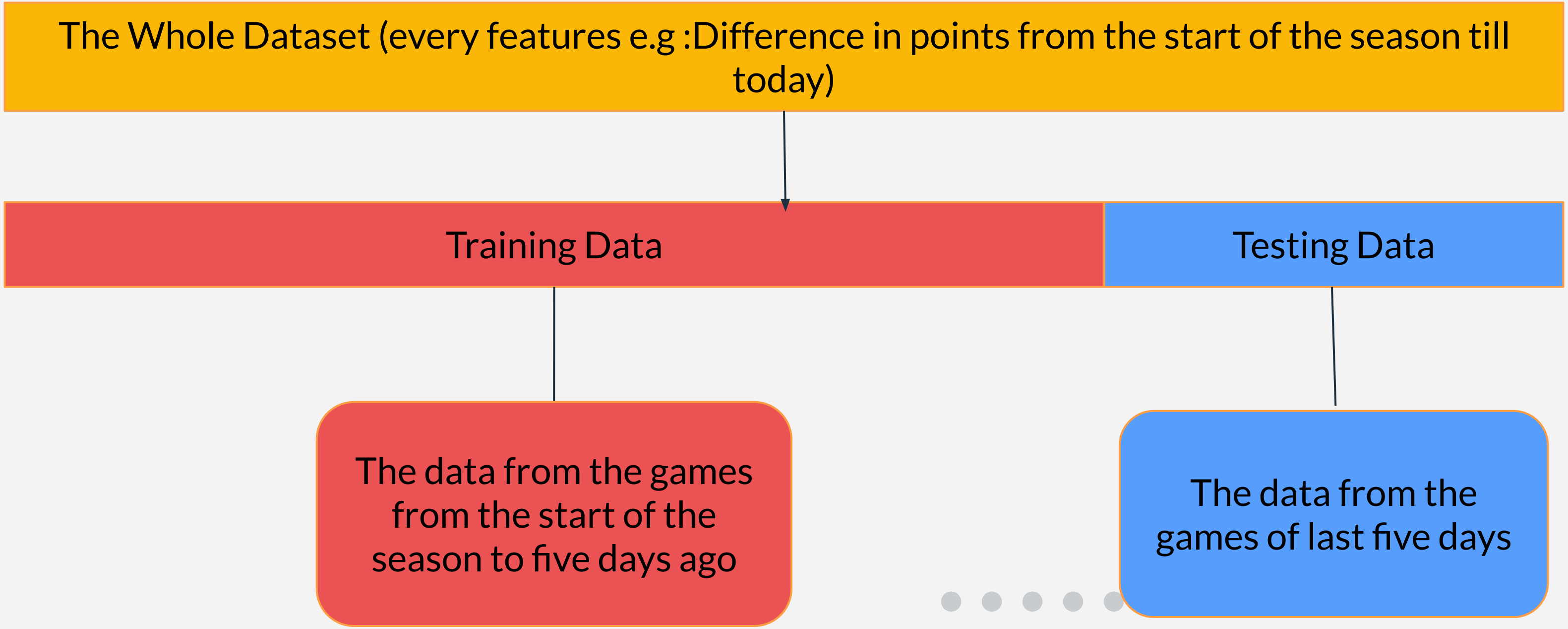
03

Performing prediction using various models

3.1 Preparing the training and testing data

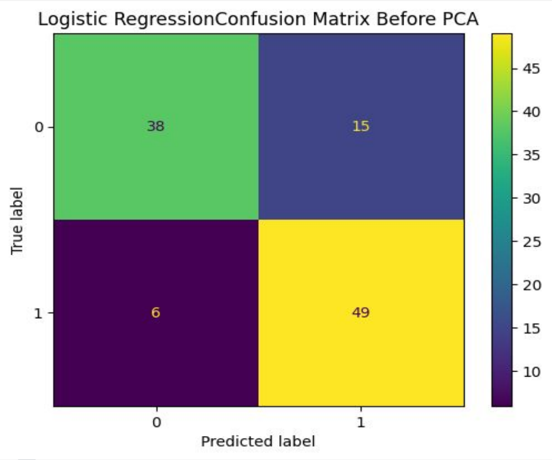
Defining Training and Testing Data

Rollover → Data is updated everyday



3.2 Performing Prediction with various models

1.Logistic Regression

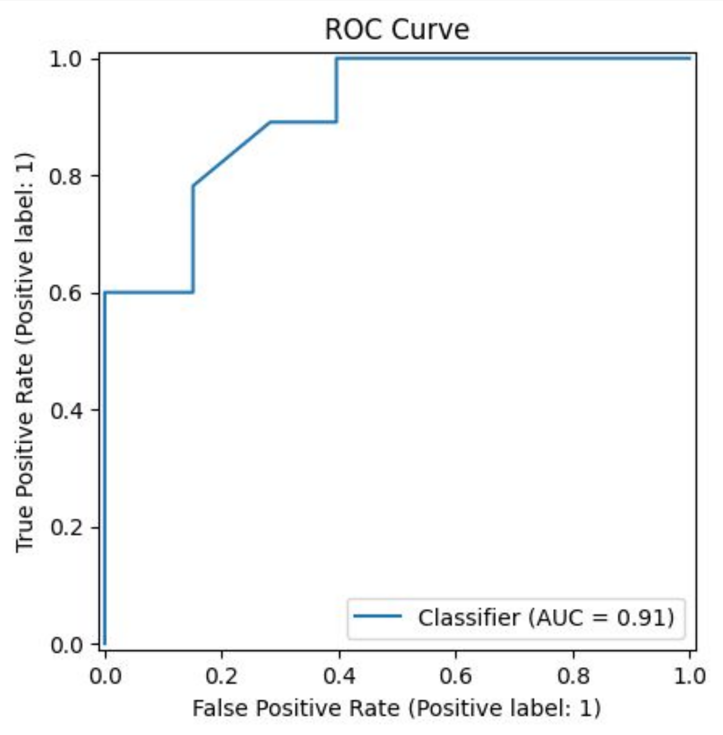



```
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.72      0.78        53
           1       0.77      0.89      0.82        55

    accuracy                           0.81       108
   macro avg       0.81      0.80      0.80       108
weighted avg       0.81      0.81      0.80       108

Accuracy of Logistic Regression: 0.8056
Precision of Logistic Regression: 0.7656
Recall of Logistic Regression: 0.8909
F1 Score of Logistic Regression: 0.8235
ROC AUC of Logistic Regression: 0.9057
```

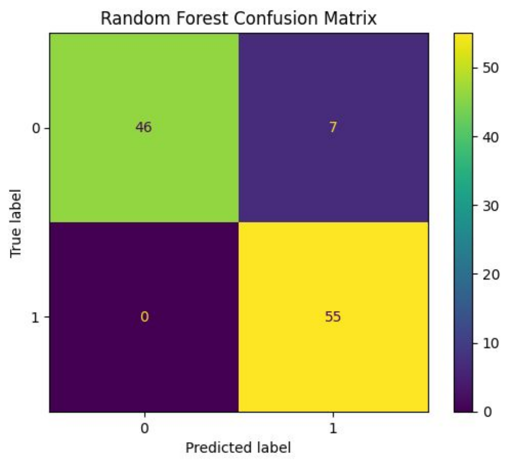
Accuracy	0.8056(80.56%)
Precision	0.7656(76.56%)
Recall	0.8909(89.09%)
F1 Score	0.8235(82.35%)
ROC AUC	0.9057(90.57%)

1.Logistic Regression

ROC:



2. Random Forest

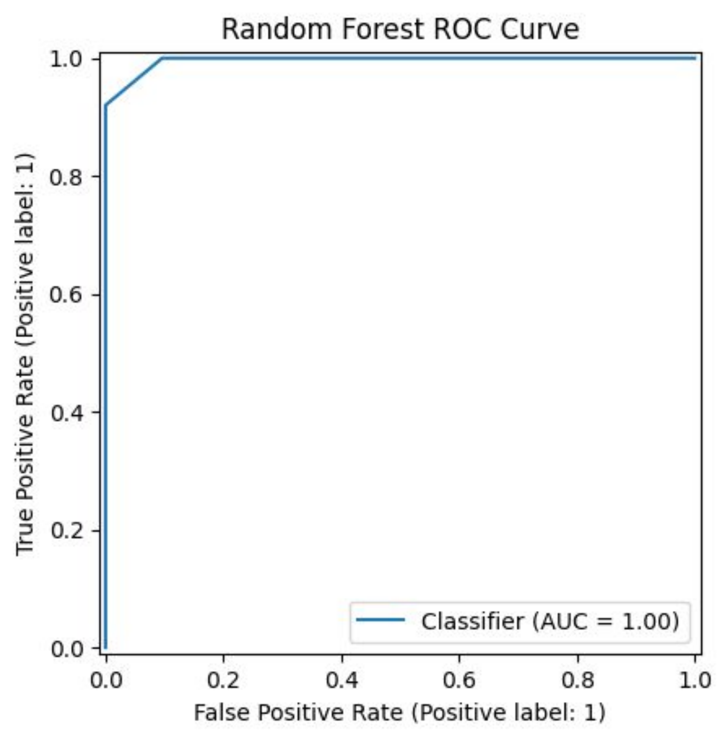



```
Random Forest Metrics:
Accuracy of Random Forest: 0.9352
Precision of Random Forest: 0.8871
Recall of Random Forest: 1.0000
F1 Score of Random Forest: 0.9402
ROC AUC of Random Forest: 0.9928

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.87      0.93        53
           1       0.89      1.00      0.94        55

    accuracy                           0.94       108
   macro avg       0.94      0.93      0.93       108
weighted avg       0.94      0.94      0.93       108
```

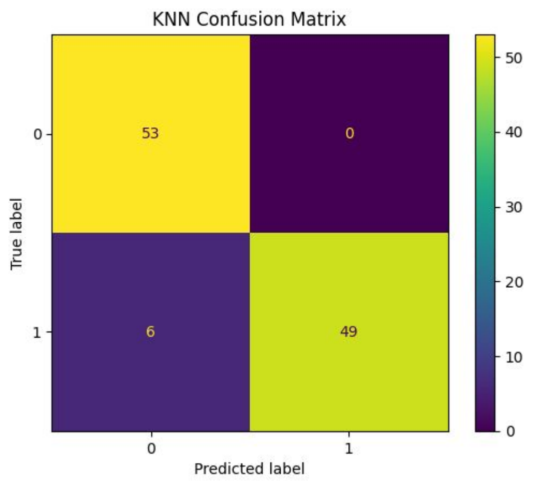
Accuracy	0.9352(93.52%)
Precision	0.8871(88.71%)
Recall	1.0000(100%)
F1 Score	0.9402(94.02%)
ROC AUC	0.9928(99.28%)

2. Random Forest

ROC:



3. K-Nearest Neighbour(KNN)

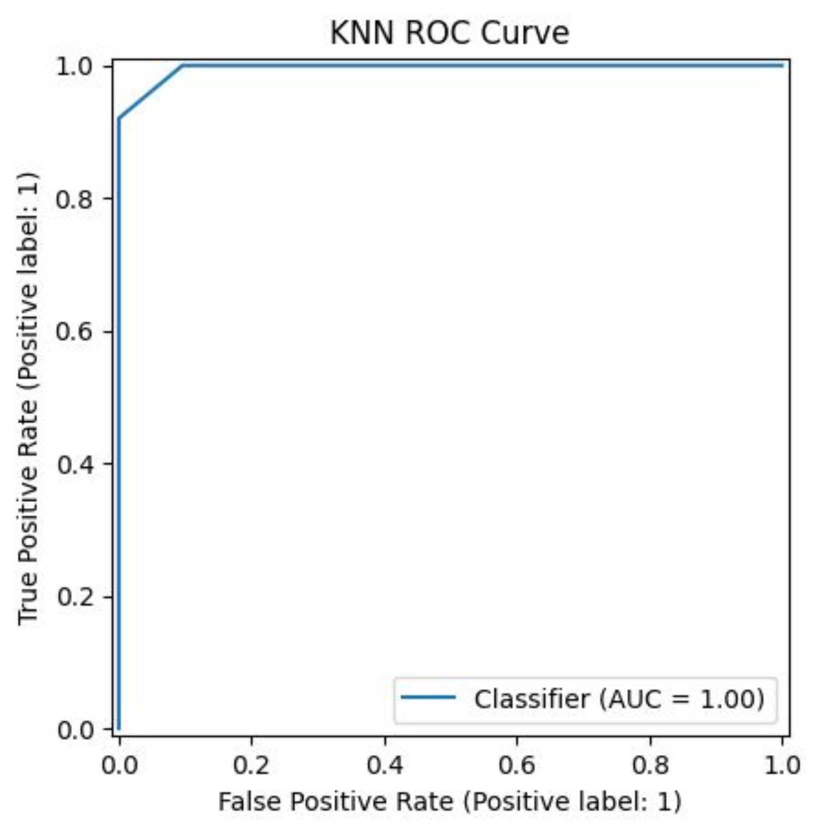



```
KNN Metrics:
Accuracy of KNN: 0.9444
Precision of KNN: 1.0000
Recall of KNN: 0.8909
F1 Score of KNN: 0.9423
ROC AUC of KNN: 0.9928

Classification Report:
              precision    recall  f1-score   support

           0       0.90      1.00      0.95        53
           1       1.00      0.89      0.94        55

    accuracy                           0.94       108
   macro avg       0.95      0.95      0.94       108
weighted avg       0.95      0.94      0.94       108
```

Accuracy	0.9444(94.44%)
Precision	1.0000(100%)
Recall	0.8909(89.09%)
F1 Score	0.9423(94.23%)
ROC AUC	0.9928(99.28%)

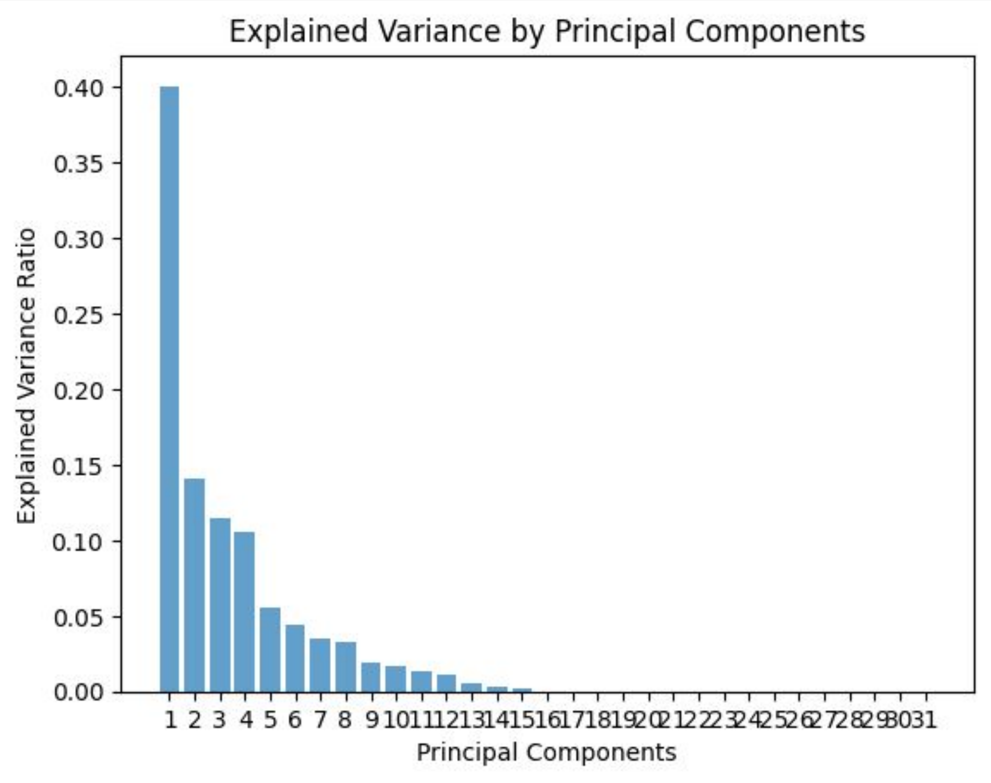
3. K-Nearest Neighbour(KNN)

ROC:



3.3 Reinforcement

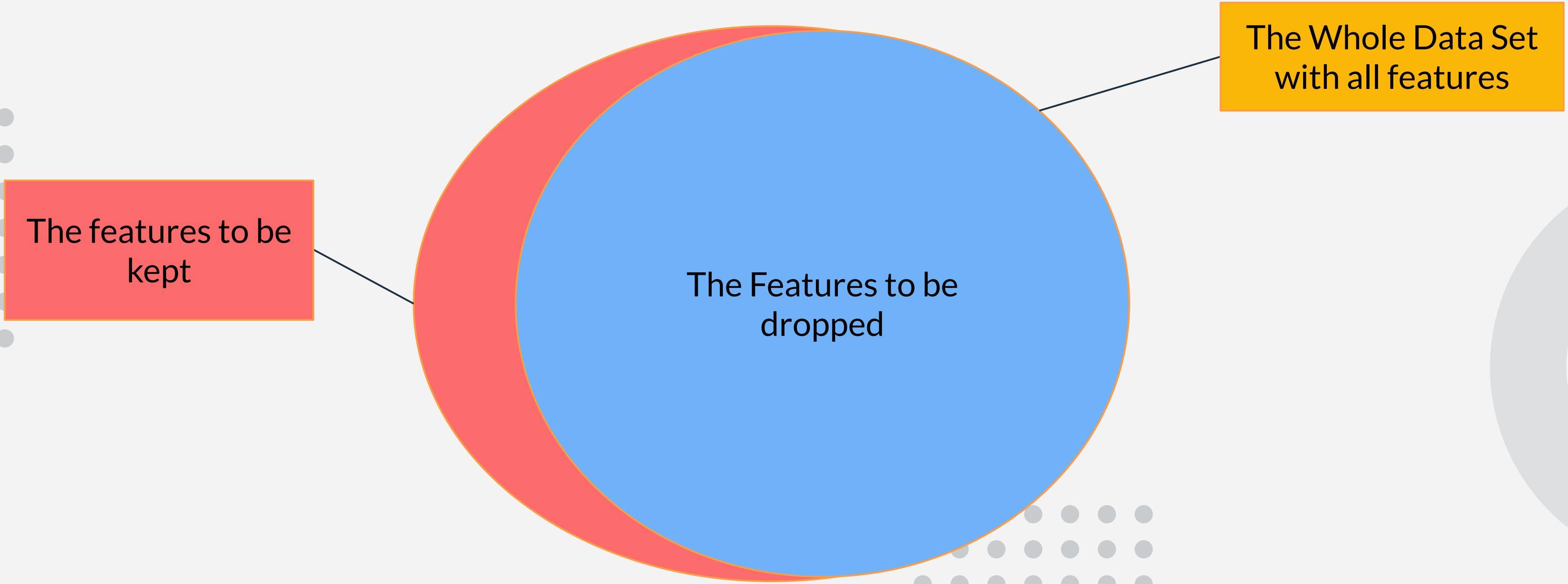
Principal Component Analysis(PCA)



	Principal Component	Explained Variance Ratio
0	PC1	4.008038e-01
1	PC2	1.403943e-01
2	PC3	1.150434e-01
3	PC4	1.051845e-01
4	PC5	5.526937e-02
5	PC6	4.400797e-02
6	PC7	3.501999e-02
7	PC8	3.279034e-02
8	PC9	1.873099e-02
9	PC10	1.731192e-02
10	PC11	1.304318e-02
11	PC12	1.093057e-02
12	PC13	5.874005e-03
13	PC14	3.649505e-03
14	PC15	1.946072e-03
15	PC16	9.674137e-17
16	PC17	8.655477e-17
17	PC18	7.968710e-17
18	PC19	5.635977e-17
19	PC20	3.894307e-17
20	PC21	1.306784e-17
21	PC22	1.042788e-17
22	PC23	0.000000e+00
23	PC24	0.000000e+00
24	PC25	0.000000e+00
25	PC26	0.000000e+00
26	PC27	0.000000e+00
27	PC28	0.000000e+00
28	PC29	0.000000e+00
29	PC30	0.000000e+00
30	PC31	0.000000e+00

Analysis Result

```
Features to keep:
['Avg_Home_Win_Score', 'Avg_Home_Loss_Score', 'Avg_Away_Loss_Score', 'Diff_Defensive_Rebounds', 'Diff_Assists', 'Diff_Turnovers', 'Diff_ThreePoint_Made', 'Dif
Features to drop:
['Diff_FreeThrow_Made', 'Diff_FieldGoal_Percentage', 'Diff_FreeThrow_Attempt', 'Diff_FreeThrow_Percentage', 'STD_Home_Win_Score', 'Diff_Steals', 'Diff_Fouls',
Number of features to keep: 12
```



1.Logistic Regression after PCA

Accuracy	0.8056(80.56%)
Precision	0.7656(76.56%)
Recall	0.8909(89.09%)
F1 Score	0.8235(82.35%)
ROC AUC	0.9057(90.57%)

Accuracy	0.8426(84.26%)
Precision	0.8909(89.09%)
Recall	0.8167(81.67%)
F1 Score	0.8522(85.22%)
ROC AUC	0.9498(94.98%)

2. Random Forest after PCA

Accuracy	0.9352(93.52%)
Precision	0.8871(88.71%)
Recall	1.0000(100%)
F1 Score	0.9402(94.02%)
ROC AUC	0.9928(99.28%)

Accuracy	0.9537(95.37%)
Precision	0.9231(92.31%)
Recall	1.0000(100%)
F1 Score	0.9600(96.00%)
ROC AUC	0.9957(99.57%)

3.KNN after PCA

Accuracy	0.9444(94.44%)
Precision	1.0000(100%)
Recall	0.8909(89.09%)
F1 Score	0.9423(94.23%)
ROC AUC	0.9928(99.28%)

Accuracy	0.9537(95.37%)
Precision	1.0000(100%)
Recall	0.9167(91.67%)
F1 Score	0.9565(95.65%)
ROC AUC	0.9583(95.83%)

MVP Prediction

What is MVP?

The **Most Valuable Player (MVP)** is an award given to the player with the highest impact on their team during the regular season of NBA.

This award is generally nominated at the end of the season in which the **whole data is accessible** to be utilized.

1 MVP Awarded Per Year

How MVP is Chosen

The MVP is chosen by sport journalists according to the **player performance statistic** during the season.

We will be using these statistics alongside several other advanced metrics in our data:

- **Player Efficiency Rating (PER):**
  The PER sums up all a player's positive accomplishments, subtracts the negative accomplishments, and returns a **per-minute rating of a player's performance**.

- **Box Plus/Minus (BPM):**
  BPM uses a player’s box score information, position, and the team’s overall performance to estimate the player’s contribution in **points above league average per 100 possessions played**.

- **Value Over Replacement Player (VORP):**
  A box score estimate of the points per 100 TEAM possessions that a player contributed above a replacement-level (-2.0) player, translated to an average team and prorated to an 82-game season.

Data Collection (MVP Prediction)

The data is scraped from “https://www.basketball-reference.com/” which contains:

- MVP rank and voting shares
- Average player statistics per game
- Advanced player statistics
- Team division standings

for every year from **1991 - 2024** joined into a single table.

Selenium chromedriver and javascript are used to load tables would otherwise be hidden in the website to ensure the completeness of the data.

Most Valuable Player (Michael Jordan Trophy) Share & Export ▼ Glossary

				Voting					Per Game						Shooting			Advanced	
Rank	Player	Age	Tm	First	Pts Won	Pts Max	Share	G	MP	PTS	TRB	AST	STL	BLK	FG%	3P%	FT%	WS	WS/48
1	Michael Jordan	27	CHI	77	891	960	0.928	82	37.0	31.5	6.0	5.5	2.7	1.0	.539	.312	.851	20.3	.321
2	Magic Johnson	31	LAL	10	497	960	0.518	79	37.1	19.4	7.0	12.5	1.3	0.2	.477	.320	.906	15.4	.251
3	David Robinson	25	SAS	6	476	960	0.496	82	37.7	25.6	13.0	2.5	1.5	3.9	.552	.143	.762	17.0	.264
4	Charles Barkley	27	PHI	2	222	960	0.231	67	37.3	27.6	10.1	4.2	1.6	0.5	.570	.284	.722	13.4	.258
5	Karl Malone	27	UTA	0	142	960	0.148	82	40.3	29.0	11.8	3.3	1.1	1.0	.527	.286	.770	15.5	.225

Sample of scraped data for MVP

Data Engineering

Ratios per Year:

- PTS_R = Total Points per Player / Total Points per *Average* Player

Categorical **Encoding**:

Milwaukee Bucks	1
Los Angeles Lakers	2
Dallas Mavericks	3

Remove **Redundant** Data:

- Remove ‘Total Points per Player’, replaced by PTS_R

… for multiple features across all years 1991-2024

Data Engineering

Add_ranks:

Sort Descending

Model

Name	MVP Shares	Rank	Predicted Shares	Predicted Rank
Giannis Antetokounmpo	0.952	1 (MVP)	0.225	1 (MVP)
Lebron James	0.746	2	0.165	3
James Harden	0.363	3	0.170	2
Luka Dončić	0.198	4	0.150	4
Kawhi Leonard	0.166	5	0.100	8

MVP Shares = Total Points Gained / Maximum Points Available

Training and Testing Data

Rollover

1. **Initialize** X = 2015
2. **Train** : All historical data before X year
3. **Test** : X Year
4. Fit the model
5. Predict MVP Shares
6. X = X+1
7. Go back to 1. and repeat until X = 2024

Train		Test	
<= 2021	2022	2023	2024

X= X+1

Train			Test
<= 2021	2022	2023	2024

Why no Cross-Validation?

- Does not respect the **temporal** nature of the data
- Data must be trained and tested in **year batches**

GridsearchCV and RandomsearchCV is difficult or impossible to apply

Evaluation Metrics

Mean Average Precision (MAP)

$$AP@K = \frac{1}{N}\sum_{k=1}^{K} Precision(k) \times rel(k)$$

Rank	Item	Precision@k
1		1/1 = 1
2		1/2 = 0.5
3		1/3 = 0.33
4		2/4 = 0.5
5		3/5 = 0.6
6		3/6 = 0.5

Average precision

$$AP@6 = \frac{1 + 0.5 + 0.6}{3} = 0.7$$

Not Relevant Relevant

Relevant = For all top 5 MVP players, predicted rank falls between 1-5

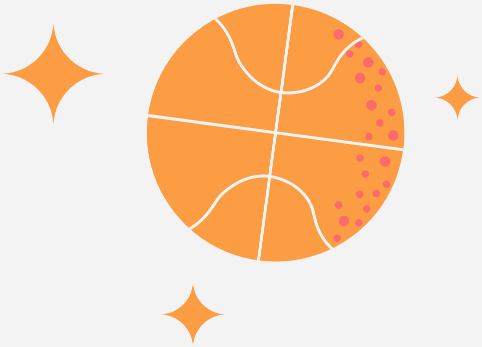
MAP is the mean of all AP values across all 10 years from 2015-2024

Evaluation Metrics

1. **Mean Average Precision (MAP) (for top k = 5 items)**
   A perfectly precise model will give precise matches for all k=5 relevant items, resulting in a value of 100%.

2. **Mean MVP Rank**
   The average predicted rank of the real MVP in our model.
   A perfect model will always predict MVP rank to be 1.

3. **MVP Accuracy**
   The number of times the model correctly predicted the MVP.
   A perfect model will score 10/10 correct guesses.



Methods

Random Forest

N = 500

min_samples = 3

Ridge Regression

Alpha = 0.1

Extra Trees

N = 500

min_samples = 3

Gradient Boosting

Estimators = 500

Alpha = 0.1

Results

For years **2015 - 2024** :

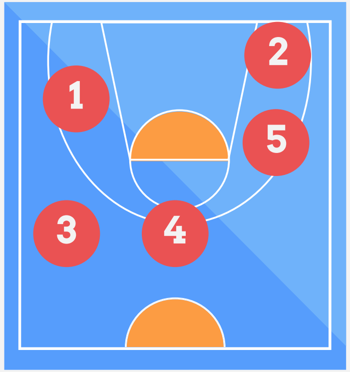
	Mean Average Precision	MVP Accuracy	Mean MVP Rank
Random Forest	90.32%	8/10	1.2
Ridge Regression	89.54%	5/10	1.5
Extra Trees	89.85%	8/10	1.5
Gradient Boosting	88.37%	8/10	1.3



Winning Team Prediction

- Probability of home team or away team wins in each competition is predicted.

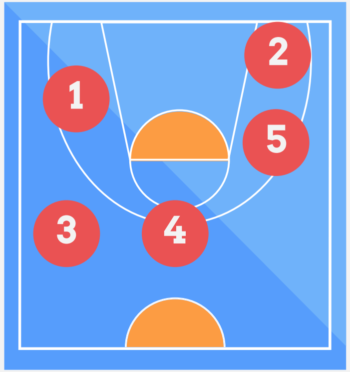
- After testing, KNN is the best method.



MVP Prediction

- The share of each players is predicted,it is then converted to rank.

- After testing, Random Forest is the best method.



Thanks!