
# NBA MVP Prediction

Joseph Sabaybay

# Most Valuable Player

One of the most prestigious award given to a player who displayed outstanding degree of skill in the regular season.

The Award is given during in the middle of the Playoff season around April/May

# How it is decided

*MVP Share = (MVP points for particular player) / (Total MVP points)*

| 2020-2021 NBA Awards Voting (Top 3 Candidates) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Player** | **1st Place Votes** (10 points) | **2nd Place Votes** (7 points) | **3rd Place Votes** (5 points) | **4th Place Votes** (3 points) | **5th Place Votes** (1 point) | **Total Points** | **MVP Share** |
| Nikola Jokic | 91 | 8 | 1 | 0 | 0 | 971 | 971 / 1010 = 0.961 |
| Joel Embiid | 1 | 62 | 23 | 8 | 3 | 586 | 586 / 1010 = 0.580 |
| Stephen Curry | 5 | 23 | 32 | 23 | 13 | 453 | 453 / 1010 = 0.449 |

Each member votes using a weighted voting system: 1st - 10 points, 2nd - 7 points, 3rd - 5 points

Before: Voting happens with “subject matter experts” who they think deserves the award

Current: 100 independent media members who are not affiliated with any team or player votes for the player

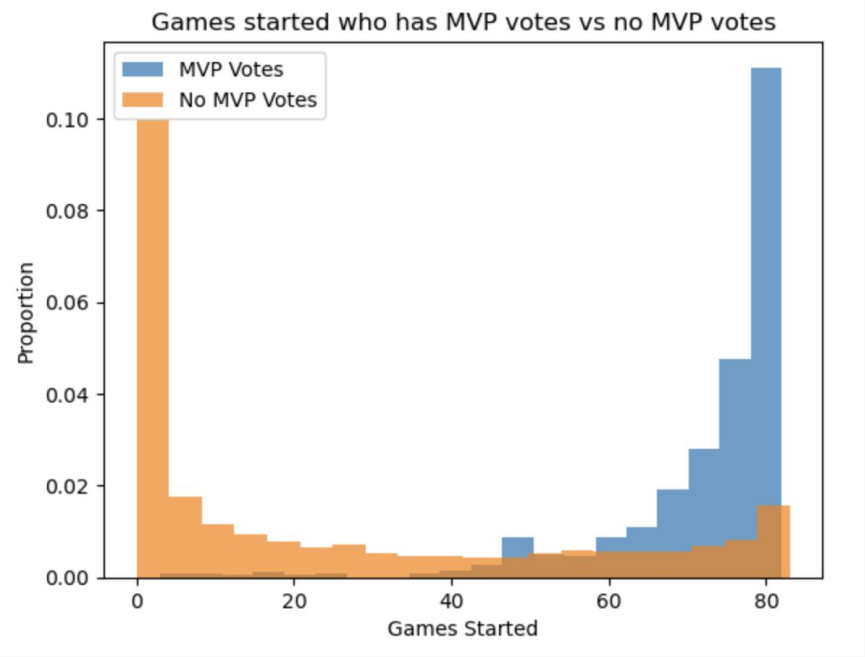
# EDA

There were null values in the percentages but as we look further, some player never attempted to score some field goals which resulted into zero dividing by itself giving it an error

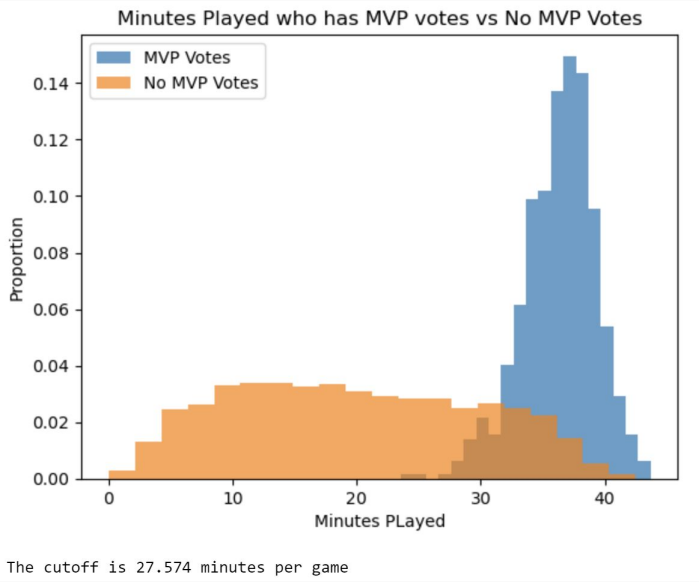
| | |
|---|---|
| fga_per_g | 0 |
| fg_pct | 63 |
| fg3_per_g | 0 |
| fg3a_per_g | 0 |
| fg3_pct | 2623 |
| fg2_per_g | 0 |
| fg2a_per_g | 0 |
| fg2_pct | 106 |
| efg_pct | 63 |
| ft_per_g | 0 |
| fta_per_g | 0 |
| ft_pct | 576 |

| | player | fga_per_g |
|---|---|---|
| **1407** | Yvon Joseph | 0.0 |
| **2077** | Jeff Lamp | 0.0 |
| **2602** | David Wood | 0.0 |
| **2956** | Mark Wade | 0.0 |
| **3183** | Gary Leonard | 0.0 |

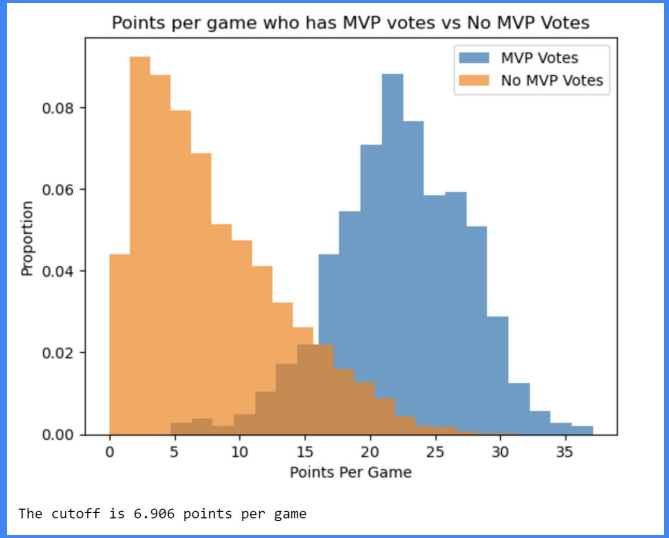
# Data Visualization



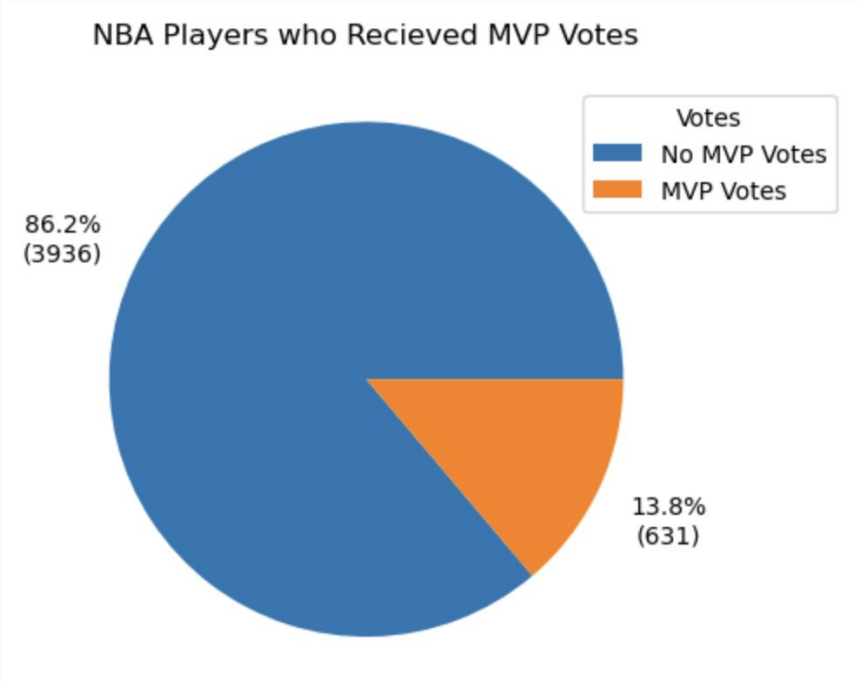

| | season | player | g | gs | pts_per_g | award_share |
|---|---|---|---|---|---|---|
| **0** | 1982 | Michael Cooper | 76 | 14 | 11.9 | 0.004 |
| **1** | 1991 | Kevin McHale | 68 | 10 | 18.4 | 0.001 |
| **2** | 1992 | Detlef Schrempf | 80 | 4 | 17.3 | 0.001 |
| **3** | 1995 | Michael Jordan | 17 | 17 | 26.9 | 0.011 |
| **4** | 1996 | Magic Johnson | 32 | 9 | 14.6 | 0.007 |
| **5** | 1999 | Darrell Armstrong | 50 | 15 | 13.8 | 0.002 |
| **6** | 1999 | Rasheed Wallace | 49 | 18 | 12.8 | 0.001 |
| **7** | 2021 | Derrick Rose | 50 | 3 | 14.7 | 0.010 |

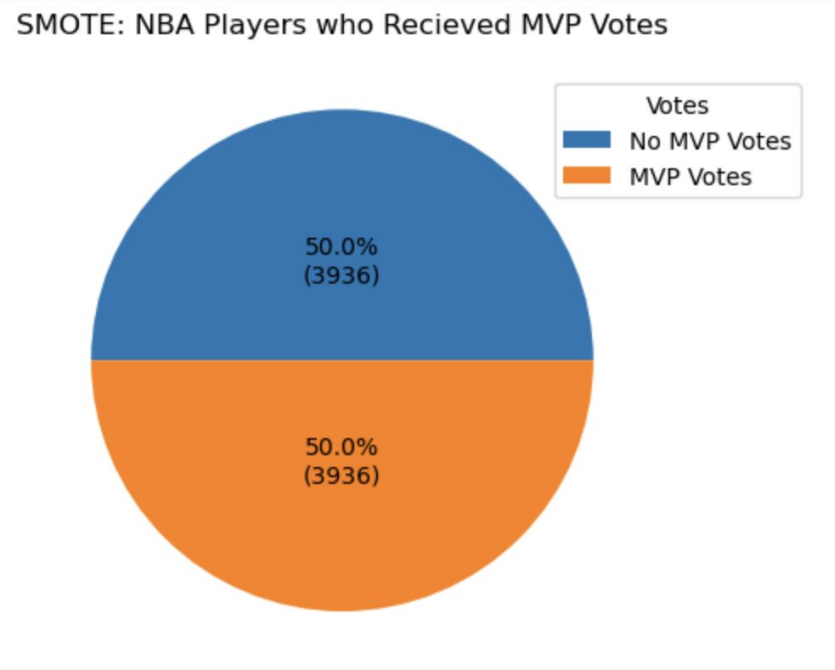



The cutoff is 27.574 minutes per game




The cutoff is 6.906 points per game


NBA Players who Recieved MVP Votes
Votes
No MVP Votes
MVP Votes
86.2%
(3936)
13.8%
(631)


SMOTE: NBA Players who Recieved MVP Votes
Votes
No MVP Votes
MVP Votes
50.0%
(3936)
50.0%
(3936)

# Modelling

# Random Forest Regressor

```
validation_scores = {"season" : [], "mae" : [], "is_mvp" : [], "was_top_two" : [], "was_top_three":[], "info" : []}
seasons = trainInfo.season.unique()
for season in seasons:
    print("\n")
    print(f"Season: {season}")

    trainFold = trainData[trainInfo["season"] != season]
    trainTarFold = trainTargets[trainInfo["season"] != season]
    valFold = trainData[(trainInfo["season"] == season) & (~trainInfo["is_smote"])]
    valTarFold = trainTargets[(trainInfo["season"] == season) & (~trainInfo["is_smote"])]
    valInfo = trainInfo[(trainInfo["season"] == season) & (~trainInfo["is_smote"])]

    regr = RandomForestRegressor(max_depth = 7, random_state=0)
    regr.fit(trainFold.to_numpy(), trainTarFold.to_numpy()[:,0])

    valPred = regr.predict(valFold)
    mae = np.mean(np.absolute(valPred - valTarFold.to_numpy()[:,0]))

    print(f"MAE: {mae}")

    topTwo = valInfo.iloc[np.argsort(valPred)[-2:]]
    was_top_two = sum(topTwo["is_mvp"]) > 0
    topThree = valInfo.iloc[np.argsort(valPred)[-3:]]
    was_top_three = sum(topThree["is_mvp"]) > 0
    print("Predicted top three players in MVP voting with their actual award_share:")
    print(topThree.iloc[::-1])


    mostVotes = np.argmax(valPred)
    score = np.amax(valPred)
    was_mvp = valInfo.iloc[[mostVotes]]["is_mvp"].values[0]
    player = valInfo.iloc[[mostVotes]]["player"].values[0]
    print(f"Predicted MVP: {player}")
    print(f"Predicted Win Share: {score}")
    print(f"Was MVP correct: {was_mvp}")

    validation_scores["season"].append(season)
    validation_scores["mae"].append(mae)
    validation_scores["is_mvp"].append(was_mvp)
    validation_scores["was_top_two"].append(was_top_three)
    validation_scores["was_top_three"].append(was_top_three)
    validation_scores["info"].append(valInfo.iloc[[mostVotes]])
```

```
Season: 2011
```

```
MAE: 0.016666480562666793
Predicted top three players in MVP voting with their actual award_share:
      season         player pos team_id  award_share is_mvp  is_smote
4215    2011   Derrick Rose  PG     CHI        0.977   True     False
7652    2011   LeBron James  SF     MIA        0.431  False     False
3505    2011  Dwight Howard   C     ORL        0.531  False     False
Predicted MVP: Derrick Rose
Predicted Win Share: 0.6733043942942362
Was MVP correct: True
```

```
Season: 2019
```

```
MAE: 0.01746455172160421
Predicted top three players in MVP voting with their actual award_share:
      season                 player pos team_id  award_share is_mvp  is_smote
970     2019  Giannis Antetokounmpo  PF     MIL        0.932   True     False
1586    2019           James Harden  PG     HOU        0.768  False     False
5846    2019           Nikola Jokić   C     DEN        0.210  False     False
Predicted MVP: Giannis Antetokounmpo
Predicted Win Share: 0.6925076747880279
Was MVP correct: True
```

# XGBoost Regressor

```
validation_scores = {"season" : [], "mae" : [], "is_mvp" : [], "was_top_two" : [], "was_top_three":[], "info" : []}
seasons = trainInfo.season.unique()

for season in seasons:
    print("\n")
    print(f"Season: {season}")

    trainFold = trainData[trainInfo["season"] != season]
    trainTarFold = trainTargets[trainInfo["season"] != season]
    valFold = trainData[(trainInfo["season"] == season) & (~trainInfo["is_smote"])]
    valTarFold = trainTargets[(trainInfo["season"] == season) & (~trainInfo["is_smote"])]
    valInfo = trainInfo[(trainInfo["season"] == season) & (~trainInfo["is_smote"])]

    regr = XGBRegressor(objective='reg:squarederror', random_state=0, max_depth = 6)
    regr.fit(trainFold, trainTarFold)

    valPred = regr.predict(valFold)
    mae = np.mean(np.absolute(valPred - valTarFold.to_numpy()[:,0]))
    print(f"MAE: {mae}")

    topTwo = valInfo.iloc[np.argsort(valPred)[-2:]]
    was_top_two = sum(topTwo["is_mvp"]) > 0
    topThree = valInfo.iloc[np.argsort(valPred)[-3:]]
    was_top_three = sum(topThree["is_mvp"]) > 0
    print("Predicted top three players in MVP voting with their actual award_share:")
    print(topThree.iloc[::-1])

    mostVotes = np.argmax(valPred)
    score = np.amax(valPred)
    was_mvp = valInfo.iloc[[mostVotes]]["is_mvp"].values[0]
    print(f"Predicted MVP: {player}")
    print(f"Predicted Win Share: {score}")
    print(f"Was MVP correct: {was_mvp}")

    validation_scores["season"].append(season)
    validation_scores["mae"].append(mae)
    validation_scores["is_mvp"].append(was_mvp)
    validation_scores["info"].append(valInfo.iloc[[mostVotes]])
    validation_scores["was_top_two"].append(was_top_two)
    validation_scores["was_top_three"].append(was_top_three)
```

```
Season: 2006
MAE: 0.017732246839433596
Predicted top three players in MVP voting with their actual award_share:
      season        player pos team_id  award_share is_mvp  is_smote
74      2006  LeBron James  SF     CLE        0.550  False     False
5904    2006   Kobe Bryant  SG     LAL        0.386  False     False
5649    2006    Steve Nash  PG     PHO        0.739   True     False
Predicted MVP: Giannis Antetokounmpo
Predicted Win Share: 0.4902097284793854
Was MVP correct: False
```

```
Season: 2014
MAE: 0.011046380895177984
Predicted top three players in MVP voting with their actual award_share:
      season         player pos team_id  award_share is_mvp  is_smote
1079    2014   Kevin Durant  SF     OKC        0.986   True     False
3772    2014   LeBron James  PF     MIA        0.713  False     False
3815    2014  Blake Griffin  PF     LAC        0.347  False     False
Predicted MVP: Giannis Antetokounmpo
Predicted Win Share: 0.9480108022689819
Was MVP correct: True
```

# Results

```
The Random Forest Model predicted the NBA MVP correctly 0.775% of the time
The Random Forest Model predicted the NBA MVP correctly in the top two 0.95% of the time
The Random Forest Model predicted the NBA MVP correctly in the top three 0.95% of the time
```

```
The XGBoost Model predicted the NBA MVP correctly 0.9% of the time
The XGBoost Model predicted the NBA MVP correctly in the top two 0.925% of the time
The XGBoost Model predicted the NBA MVP correctly in the top three 0.975% of the time
```


```
MAE: 0.018819785189378 14
[77]
Predicted top three players in MVP voting with their actual award_share:
     season                 player pos team_id  award_share  is_mvp
77     2022          Nikola Jokić   C     DEN        0.875    True
2      2022  Giannis Antetokounmpo  PF     MIL        0.595   False
40     2022           Luka Dončić  PG     DAL        0.146   False
45     2022           Joel Embiid   C     PHI        0.706   False
115    2022          Jayson Tatum  SF     BOS        0.043   False
100    2022            Chris Paul  PG     PHO        0.002   False
75     2022          LeBron James  PF     LAL        0.001   False
93     2022             Ja Morant  PG     MEM        0.010   False
96     2022       Dejounte Murray  PG     SAS        0.000   False
27     2022          Jimmy Butler  SF     MIA        0.000   False
Predicted MVP: Giannis Antetokounmpo
Predicted Win Share: 0.6058644056320 19
Was MVP correct: True
```

# Challenges and Next Steps

## Business Applications

- Sports Betting
- Player Improvement
- Team Development

## Challenges

- Degree of Difficulty
- Inexperience
- Time

## Next Steps

- Train more models
- More feature engineering
- Use the model in the 2023 season

# Thank You!