# Machine Learning
# Finals Project


אריק טטייבסקי 208997056

רועי משולם 315635649

## Data Description

NBA Salaries: Hoops Fortune (2020-2025)

https://www.kaggle.com/datasets/omarsobhy14/nba-players-salaries?resource=download

The "NBA Player Salaries 2023-2025" dataset is a treasure trove of financial insights in the basketball world. It features detailed records of player earnings for each season. We've also manually added each player's NBA 2K rating

Each object in our data contains

Player ID - *unique* integer

Player Full Name - string

Salary 22-23 - integer

Salary 23-24 - integer

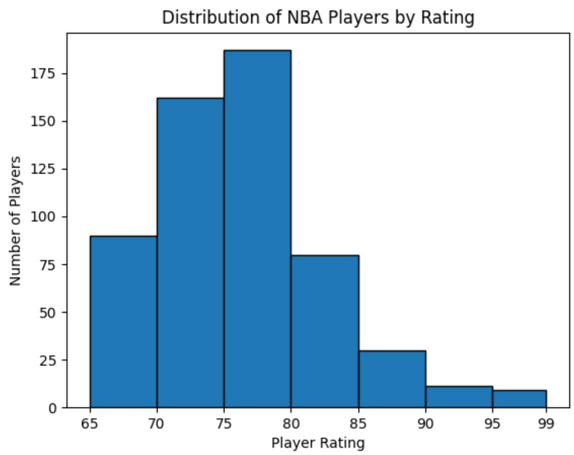
Salary 24-25 - integer

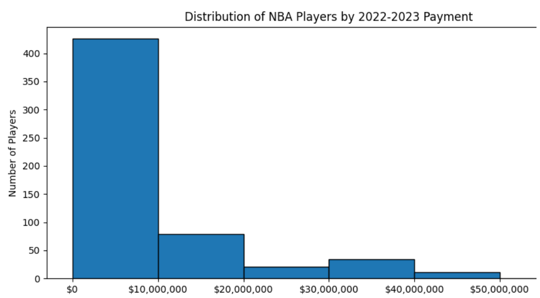
NBA 2K Rating - integer in the range of 60-99

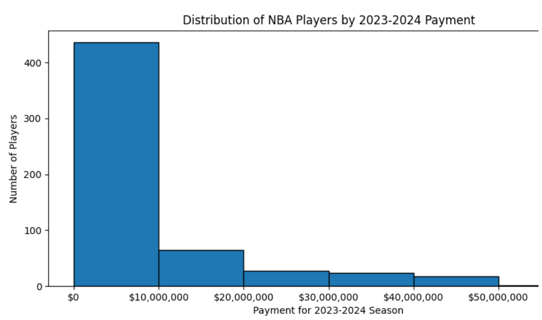
Partial screenshot from the csv file:

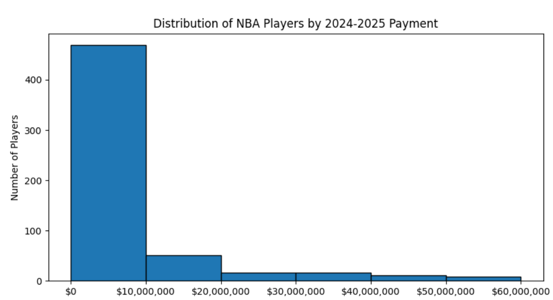
| Player Id | Player Name | 2022/2023 | 2023/2024 | 2024/2025 | 2K Rating |
|---|---|---|---|---|---|
| 1 | Stephen Curry | $48,070,014 | $51,915,615 | $55,761,217 | 96 |
| 2 | John Wall | $47,345,760 | $0 | $0 | 79 |
| 3 | Russell Westbrook | $47,080,179 | $0 | $0 | 81 |
| 4 | LeBron James | $44,474,988 | $46,698,737 | $50,434,636 | 97 |
| 5 | Kevin Durant | $44,119,845 | $47,649,433 | $51,179,020 | 96 |
| 6 | Bradley Beal | $43,279,250 | $46,741,590 | $50,203,930 | 87 |
| 7 | Paul George | $42,492,492 | $45,640,084 | $48,787,676 | 89 |
| 8 | Kawhi Leonard | $42,492,492 | $45,640,084 | $48,787,676 | 92 |
| 9 | Giannis Antetokounmpo | $42,492,492 | $45,640,084 | $48,787,676 | 97 |
| 10 | Damian Lillard | $42,492,492 | $45,640,084 | $48,787,676 | 95 |
| 11 | Klay Thompson | $40,600,080 | $43,219,440 | $0 | 86 |
| 12 | Kyrie Irving | $38,917,057 | $0 | $0 | 91 |
| 13 | Rudy Gobert | $38,172,414 | $41,000,000 | $43,827,586 | 84 |
| 14 | Khris Middleton | $37,984,276 | $40,396,552 | $0 | 86 |
| 15 | Anthony Davis | $37,980,720 | $40,600,080 | $43,219,440 | 94 |
| 16 | Jimmy Butler | $37,653,300 | $45,183,960 | $48,798,677 | 93 |
| 17 | Tobias Harris | $37,633,050 | $39,270,150 | $0 | 81 |
| 18 | Kemba Walker | $37,281,261 | $0 | $0 | 76 |
| 19 | Trae Young | $37,096,500 | $40,064,220 | $43,031,940 | 89 |
| 20 | Zach LaVine | $37,096,500 | $40,064,220 | $43,031,940 | 87 |
| 21 | Luka Doncic | $37,096,500 | $40,064,220 | $43,031,940 | 97 |
| 22 | Ben Simmons | $35,448,672 | $37,893,408 | $40,338,144 | 78 |
| 23 | Pascal Siakam | $35,448,672 | $37,893,408 | $0 | 87 |

# Data Distribution



## Question we would like to answer

We want to know if by learning the salaries of each year we can predict the rating of new incoming players to the NBA

## Our approach

We randomly split the data 50% to train and 50% to test and try to classify it using the following algorithms

Support Vector Machine (SVM)

Random Forest

Adaboost

K-nearest-neighbors (KNN)

## Problems we expect with the data

<u>High salary and poor performence players such as</u>

Ben Simmons

| | | | | | | |
|---|---|---|---|---|---|---|
| 22 | Ben Simmons | $35,448,672 | $37,893,4 | $40,338,1 | | 78 |

John Wall

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | John Wall | $47,345,760 | $0 | $0 | | 79 |

Kemba Walker

| | | | | | | |
|---|---|---|---|---|---|---|
| 18 | Kemba Walker | $37,281,261 | $0 | $0 | | 76 |

<u>Low salary and high performence players such as</u>

Desmond Bane

| | | | | | | |
|---|---|---|---|---|---|---|
| 369 | Desmond Bane | $2,130,240 | $3,845,08 | $5,767,62 | | 85 |

LaMelo Ball

| | | | | | | |
|---|---|---|---|---|---|---|
| 169 | LaMelo Ball | $8,623,920 | $10,900,6 | $14,301,6 | | 86 |

## Results - SVM

<u>Support Vector Machine (SVM)</u>

Mean Squared Error: 14.649878751887888

Sample screenshot

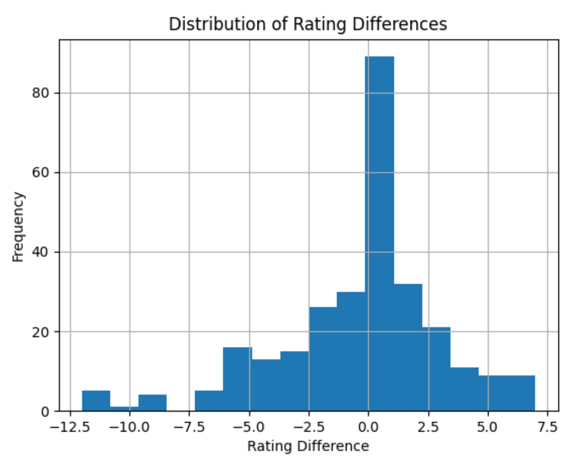
```
Player: Bojan Bogdanovic
  Predicted Rating = 81.97
  Actual Rating = 82.0

Player: RJ Barrett
  Predicted Rating = 83.42
  Actual Rating = 82.0

Player: Ron Harper Jr
  Predicted Rating = 71.12
  Actual Rating = 69.0

Player: Shaquille Harrison
  Predicted Rating = 71.00
  Actual Rating = 69.0

Player: Walker Kessler
  Predicted Rating = 73.00
  Actual Rating = 83.0
```

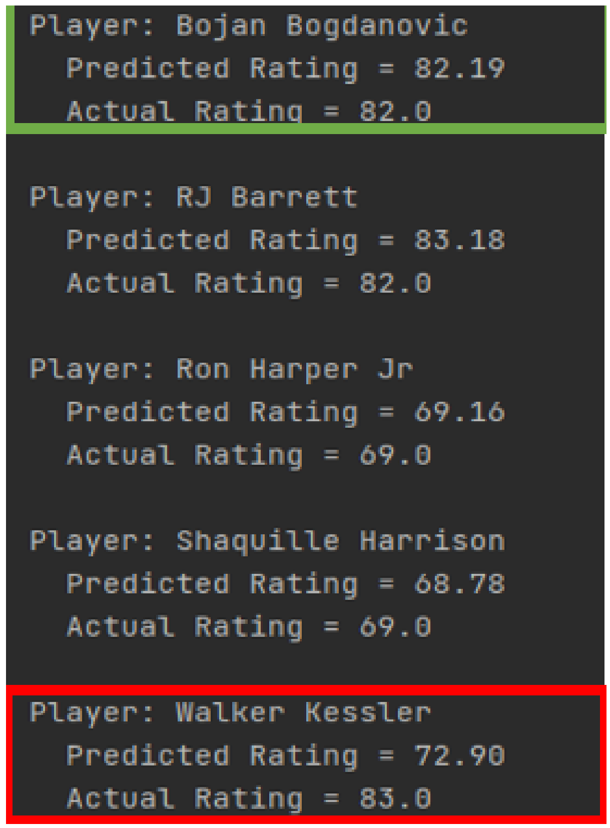


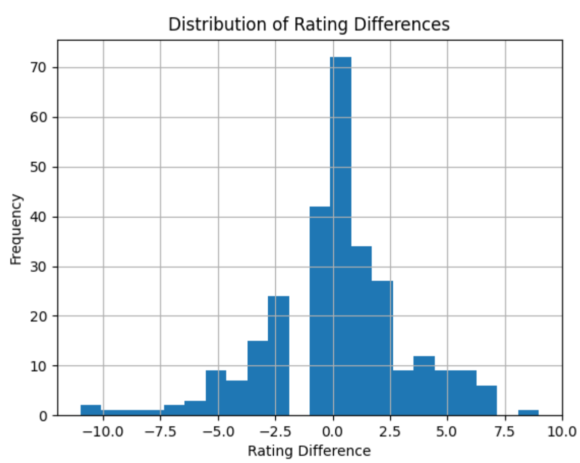
## Results - Random Forest

Random Forest

Mean Squared Error: 10.86627822690605

Sample screenshot



## Results - Adaboost

<u>Adaboost</u>

Mean Squared Error: 10.59777273779598

Sample screenshot

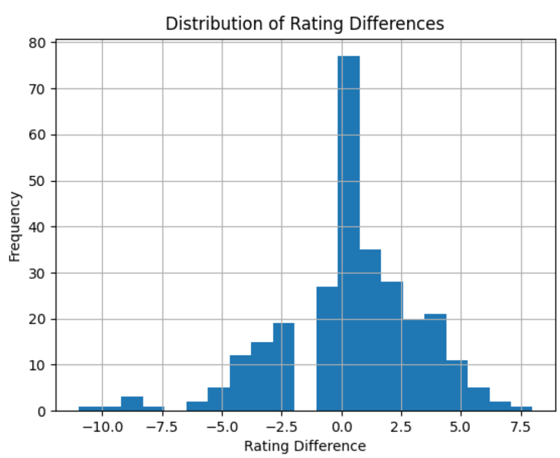
```
Player: Bojan Bogdanovic
   Predicted Rating = 80.66
   Actual Rating = 82.0

Player: RJ Barrett
   Predicted Rating = 86.11
   Actual Rating = 82.0

Player: Ron Harper Jr
   Predicted Rating = 69.14
   Actual Rating = 69.0

Player: Shaquille Harrison
   Predicted Rating = 69.34
   Actual Rating = 69.0

Player: Walker Kessler
   Predicted Rating = 73.90
   Actual Rating = 83.0
```



## Results - KNN

K-Nearest-Neighbors (KNN)

Mean Squared Error: 9.9244755244755524

Sample screenshot

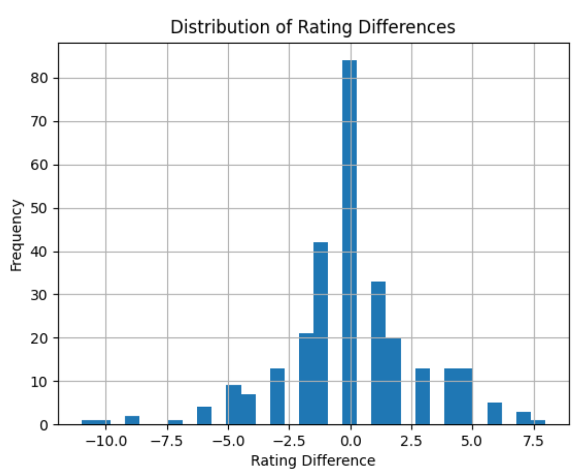
```
Player: Bojan Bogdanovic
  Predicted Rating = 83.00
  Actual Rating = 82.0

Player: RJ Barrett
  Predicted Rating = 83.40
  Actual Rating = 82.0

Player: Ron Harper Jr
  Predicted Rating = 68.80
  Actual Rating = 69.0

Player: Shaquille Harrison
  Predicted Rating = 69.00
  Actual Rating = 69.0

Player: Walker Kessler
  Predicted Rating = 73.60
  Actual Rating = 83.0
```



## Result - Analysis

We compared the MSE for the 4 different algorithms: SVM, Random Forest, AdaBoost, and KNN applied to the NBA dataset.

Mean Squared Error (MSE) Comparison:

SVM = 14.65, Random Forest = 10.87, AdaBoost = 10.60, KNN = 9.92

The MSE values indicate the average squared difference between the predicted and actual ratings. A lower MSE signifies better prediction accuracy.

**Difference in MSE between Algorithms:**

Algorithmic Approach: Each algorithm utilizes a distinct methodology for predicting the ratings. SVM aims to find a hyperplane that separates the data, Random Forest combines multiple decision trees, AdaBoost focuses on ensembling weak learners, and KNN relies on the proximity of data points.

Model Complexity: Different algorithms have varying levels of model complexity. SVM, Random Forest, AdaBoost, and KNN have different underlying assumptions, decision boundaries, and handling of non-linear relationships. These variances can impact their ability to capture the intricacies of the NBA dataset accurately.

Feature Importance: The selected features (salaries) used for prediction can influence the performance of the algorithms. While all algorithms used the same features, they might assign different levels of importance to each feature, resulting in varied MSE values.

**KNN and Best MSE:**

KNN achieved the lowest MSE among the four algorithms, indicating relatively better prediction accuracy. There are a few reasons why KNN might have outperformed the other algorithms:

Local Proximity: KNN utilizes the local proximity of data points to make predictions. In the NBA dataset, players with similar salaries might exhibit similar 2K ratings, making the local approach of KNN effective in capturing these relationships.

Parameter Tuning: The performance of KNN can be sensitive to the choice of the parameter k (the number of neighbors). By fine-tuning this parameter, KNN can adapt to the characteristics of the NBA dataset, leading to improved prediction accuracy.

**Improving Predictions:**

Additional Features: Factors such as player performance statistics, team dynamics, or player experience might provide valuable insights and improve the models' predictive power. Also we can take in other parameters such as height, weight and age.

Data Quality and Quantity: Ensure the dataset is clean and updated (new contracts).

## Experiment 2

We will try to improve each of the algorithms to generate a better mse:

<u>SVM</u>

We tried to change the kernel function to see what output it will generate and we noticed that the default function (rbf) is working the best

RBF - 14.649

Sigmoid - 267.163

Poly - 26.001

<u>Random Forest</u>

We changed the the n_estimators, max_depth variables to minimize the mse and we found out that when n_estimators=65, max_depth=3 we generate mse = 9.676 (original is 11.309).

n_estimators: This parameter specifies the number of trees in the random forest.

max_depth: The maximum depth of each tree in the random forest.

<u>Adaboost</u>

We changed the the learning_rate variable to minimize the mse and we found out that when learning_rate=0.16 we generate mse = 9.759 (original is 10.357).

learning_rate: This parameter is the weight applied to each regressor at each boosting iteration. A higher learning rate increases the contribution of each regressor.

<u>KNN</u>

We changed the the N-Neighbors variable to minimize the mse and we found out that when n_neighbors=7 we generate mse = 9.524 (original is 9.924).

n_neighbors: This parameter states the number of neighbors we want.

## Summary Experiment 2

<u>SVM:</u>

Original MSE: 14.649

Optimized MSE: 14.649

<u>Random-Forest:</u>

Original MSE: 11.309

Optimized MSE: 9.676

<u>AdaBoost:</u>

Original MSE: 10.357

Optimized MSE: 9.759

<u>KNN:</u>

Original MSE: 9.924

Optimized MSE: 9.524

## Experiment 3

We will use the improvments we found in experiment 2 and change the size of the test & train data to 80%-20%

## Summary Experiment 3

SVM:

Experiment 2 MSE: 14.649

Optimized MSE: 12.228

Random-Forest:

Experiment 2 MSE: 9.676

Optimized MSE: 9.522

AdaBoost:

Experiment 2 MSE: 9.759

Optimized MSE: 8.973

KNN:

Experiment 2 MSE: 9.524

Optimized MSE: 8.375

## Predict new incoming players to the NBA ratings:

Arik Tatievski, 15,000,000 , 20,000,000 , 25,000,000

Roi Meshulam, 2,500,000 , 4,000,000 , 10,000,000

<u>SVM</u>

```
Player: Arik Tatievski
  Predicted Rating = 82.85

Player: Roi Meshulam
  Predicted Rating = 76.30
```

<u>Random Forest</u>

```
Player: Arik Tatievski
  Predicted Rating = 81.50

Player: Roi Meshulam
  Predicted Rating = 78.16
```

<u>AdaBoost</u>

```
Player: Arik Tatievski
  Predicted Rating = 81.26

Player: Roi Meshulam
  Predicted Rating = 78.66
```

<u>KNN</u>

```
Player: Arik Tatievski
  Predicted Rating = 82.60

Player: Roi Meshulam
  Predicted Rating = 76.80
```