NBA Prediction Lineup

Group 8: Andy Dai, Ashad Ahmed, Kevin Jacob

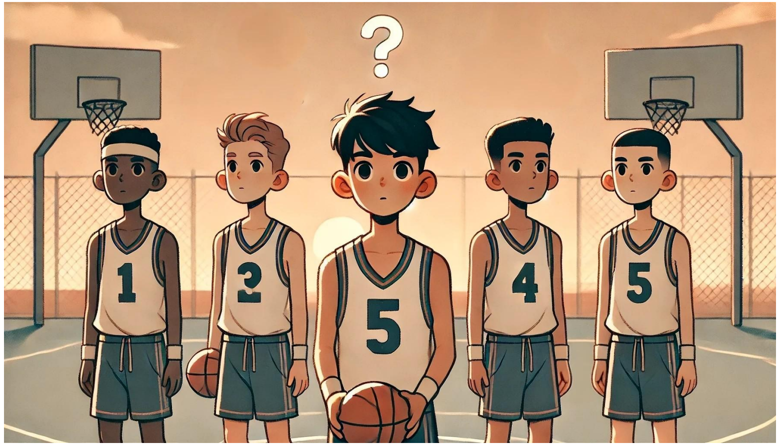
TOPICS

INTRODUCTION

01

Problem Statement

Our problem involves going through historical NBA data, using it to build a machine learning model for predicting the best player for a certain lineup where the given lineup is missing one player.



PROJECT APPROACH

02

PROJECT APPROACH

For the model, we needed to make a few assumptions to improve the performance of the model

- Only players on home team currently playing will be chosen

- Players in future season are not allowed to play in the previous seasons

- If a player is required to be reassigned to a new team, it will be removed from the previous team they were on

FUNCTIONALITY

03

MODEL FEATURES

Lineup Embedding

The most basic part of the model where it takes all lineups and embeds it where we use cosine to find the closest distance for players

Team Cohesion

A feature which is used to detect how often players play with each other, giving a higher weight to those who play with each other more often

Categorizing Start Time

Some players play more often during earlier start times than others, so this helps categorizing whether a player is more likely to play near the beginning, middle, or end of a game based on the start time

Limiting to players for that home team and 3 seasons before if available

This helps reduce the search space and prevents any player that otherwise be impossible to play during that game in that season

EVALUATION AND FINDINGS

04

RESULTS EXAMINATION

Results

- Highest Overall Accuracy = 26%
- However when predicting for 2016 lineups, only scored a maximum of 11% accuracy

Loss Function

```
Training samples: 947650
Epoch 1/30, Loss: 0.3680
Epoch 2/30, Loss: 0.3410
Epoch 3/30, Loss: 0.3340
Epoch 4/30, Loss: 0.3301
Epoch 5/30, Loss: 0.3275
Epoch 6/30, Loss: 0.3256
Epoch 7/30, Loss: 0.3241
Epoch 8/30, Loss: 0.3229
Epoch 9/30, Loss: 0.3218
Epoch 10/30, Loss: 0.3210
Epoch 11/30, Loss: 0.3202
Epoch 12/30, Loss: 0.3195
Epoch 13/30, Loss: 0.3189
Epoch 14/30, Loss: 0.3183
Epoch 15/30, Loss: 0.3179
Epoch 16/30, Loss: 0.3174
Epoch 17/30, Loss: 0.3169
Epoch 18/30, Loss: 0.3165
Epoch 19/30, Loss: 0.3161
Epoch 20/30, Loss: 0.3158
Epoch 21/30, Loss: 0.3154
Epoch 22/30, Loss: 0.3150
Epoch 23/30, Loss: 0.3149
Epoch 24/30, Loss: 0.3145
Epoch 25/30, Loss: 0.3142
Epoch 26/30, Loss: 0.3140
Epoch 27/30, Loss: 0.3137
Epoch 28/30, Loss: 0.3135
Epoch 29/30, Loss: 0.3133
Epoch 30/30, Loss: 0.3130
Training complete. Model and embeddings saved.

Process finished with exit code 0
```

Accuracy test

```
Model loaded successfully!
Prepared 1000 test lineups for predicti
Predictions saved to 'NBA_predictions.c

Overall Accuracy: 21.70%  (217/1000)

2007 TESTS - accuracy: 12.00%
There are 12/100 correct results

2008 TESTS - accuracy: 31.00%
There are 31/100 correct results

2009 TESTS - accuracy: 22.00%
There are 22/100 correct results

2010 TESTS - accuracy: 28.00%
There are 28/100 correct results

2011 TESTS - accuracy: 40.00%
There are 40/100 correct results

2012 TESTS - accuracy: 32.00%
There are 32/100 correct results

2013 TESTS - accuracy: 35.00%
There are 35/100 correct results

2014 TESTS - accuracy: 13.00%
There are 13/100 correct results

2015 TESTS - accuracy: 2.00%
There are 2/100 correct results

2016 TESTS - accuracy: 2.00%
There are 2/100 correct results

Done!

Process finished with exit code 0
```

CHALLENGES

05

For the main testing for the NBA Test, we used the model with 50 epochs to ensure that we achieved the highest accuracy, though overfitting may occur

Number of matches per year within the test dataset

- There are 100 matches per year/season

Average number of matches across the entire dataset.

- 2007: 1230 games
- 2008: 1230 games
- 2010: 1230 games
- 2011: 1229 games
- 2012: 990 games
- 2013: 1229 games
- 2014: 1230 games
- 2015: 1230 games

```
predictions were saved into NBA predictions.c

overall accuracy: 18.90%  (189 out of 1000)

2007 Test accuracy: 0.00%
0 out of 100 are correct

2008 Test accuracy: 27.00%
27 out of 100 are correct

2009 Test accuracy: 18.00%
18 out of 100 are correct

2010 Test accuracy: 28.00%
28 out of 100 are correct

2011 Test accuracy: 33.00%
33 out of 100 are correct

2012 Test accuracy: 35.00%
35 out of 100 are correct

2013 Test accuracy: 31.00%
31 out of 100 are correct

2014 Test accuracy: 13.00%
13 out of 100 are correct

2015 Test accuracy: 2.00%
2 out of 100 are correct

2016 Test accuracy: 2.00%
2 out of 100 are correct

Process finished with exit code 0
```

```
Total training samples: 670245
Total Players: 627
epoch [1 out of 50], loss: 0.349735
epoch [2 out of 50], loss: 0.314686
epoch [3 out of 50], loss: 0.305646
epoch [4 out of 50], loss: 0.300585
epoch [5 out of 50], loss: 0.296997
epoch [6 out of 50], loss: 0.294470
epoch [7 out of 50], loss: 0.292439
epoch [8 out of 50], loss: 0.290806
epoch [9 out of 50], loss: 0.289370
epoch [10 out of 50], loss: 0.288206
epoch [11 out of 50], loss: 0.287171
epoch [12 out of 50], loss: 0.286264
epoch [13 out of 50], loss: 0.285502
epoch [14 out of 50], loss: 0.284760
epoch [15 out of 50], loss: 0.284128
epoch [16 out of 50], loss: 0.283475
epoch [17 out of 50], loss: 0.283011
epoch [18 out of 50], loss: 0.282479
epoch [19 out of 50], loss: 0.281964
epoch [20 out of 50], loss: 0.281552
epoch [21 out of 50], loss: 0.281097
epoch [22 out of 50], loss: 0.280721
epoch [23 out of 50], loss: 0.280373
epoch [24 out of 50], loss: 0.280005
epoch [25 out of 50], loss: 0.279683
epoch [26 out of 50], loss: 0.279306
epoch [27 out of 50], loss: 0.279020
epoch [28 out of 50], loss: 0.278751
epoch [29 out of 50], loss: 0.278539
epoch [30 out of 50], loss: 0.278245
epoch [31 out of 50], loss: 0.278013
epoch [32 out of 50], loss: 0.277801
epoch [33 out of 50], loss: 0.277555
epoch [34 out of 50], loss: 0.277217
epoch [35 out of 50], loss: 0.277063
epoch [36 out of 50], loss: 0.276869
epoch [37 out of 50], loss: 0.276688
epoch [38 out of 50], loss: 0.276490
epoch [39 out of 50], loss: 0.276307
epoch [40 out of 50], loss: 0.276067
epoch [41 out of 50], loss: 0.275954
epoch [42 out of 50], loss: 0.275732
epoch [43 out of 50], loss: 0.275553
epoch [44 out of 50], loss: 0.275400
epoch [45 out of 50], loss: 0.275236
epoch [46 out of 50], loss: 0.275091
epoch [47 out of 50], loss: 0.274970
epoch [48 out of 50], loss: 0.274833
epoch [49 out of 50], loss: 0.274678
epoch [50 out of 50], loss: 0.274524
model saved
```

CHALLENGES

CHALLENGES	SOLUTION
Many different ways of splitting the data and testing it which required significant changes to the code each time our group wanted to try a different method.	Tested many different ways of splitting the data such as random 80/20 split, leave one out cross validation, stratified split. Ultimately went with 80/20 split
Introducing more features to reduce the search space of the model to improve its accuracy also introduced greater performance overhead when training the model.	Optimizing each feature to reduce the complexity level to ensure that it is able to train within a reasonable amount of time.
Optimizing the model to improve accuracy. There were many times where adding features or small parameter adjustments to the model would lower the accuracy	Trial and error, eventually our data did improve in terms of accuracy, however a lot more trial and error is required for our model

CONCLUSION

06

Q&A

07

THANK YOU!

From:
Andy Dai,
Ashad Ahmed,
Kevin Jacob