

# Beating Vegas: NBA

▀ John Berry, David Mitre, Avi Hashash

# Refresher on Problem & Interest

- Our goal is to use machine learning algorithms to predict the outcomes of various NBA betting scenarios.
- It's no secret that Vegas usually comes out on top, but we aim to beat the system with the odds stacked against us.
- Bets that are more likely to hit are by nature less profitable.
- Basketball is a game of inches often!
- Many factors to go into losses, wins, and stat lines.
- This year is particularly competitive, and many confounding factors like covid restriction and roster shuffle that puts uncertainty on the evaluation of bets.




## Example: Celtics - Nets Playoff Series



- Low sample size of truly healthy Nets team.
- Kyrie played 29 regular season games this year, 54 last – regular season has 82 games a year.
- KD 55 games this year, 35 last.
- February 10<sup>th</sup> trade Harden for Seth Curry, Andre Drummond.

# Outlook on Series

- Caesars opened Celtics as underdogs despite seeding and bettors heavily went the other way.
- Other sports books opened similarly, and bettors took Boston at plus money.
- Very uncertain!



 **Caesars Sportsbook**   
@CaesarsSports 


96% of the tickets and 98% of the money so far has been placed on the Celtics to beat the Nets in the first round of the [#NBAPlayoffs](#) 🤯




The Celtics opened at +115 and are now favorites in the series.

**BOSTON CELTICS VS BROOKLYN NETS** >

APR 17 | 12:00PM SERIES BETTING

 <b>Boston Celtics</b>	-130
 <b>Brooklyn Nets</b>	+110

10:38 AM · Apr 13, 2022 

 20  Reply  Share

[Read 5 replies](#)

## Related Work

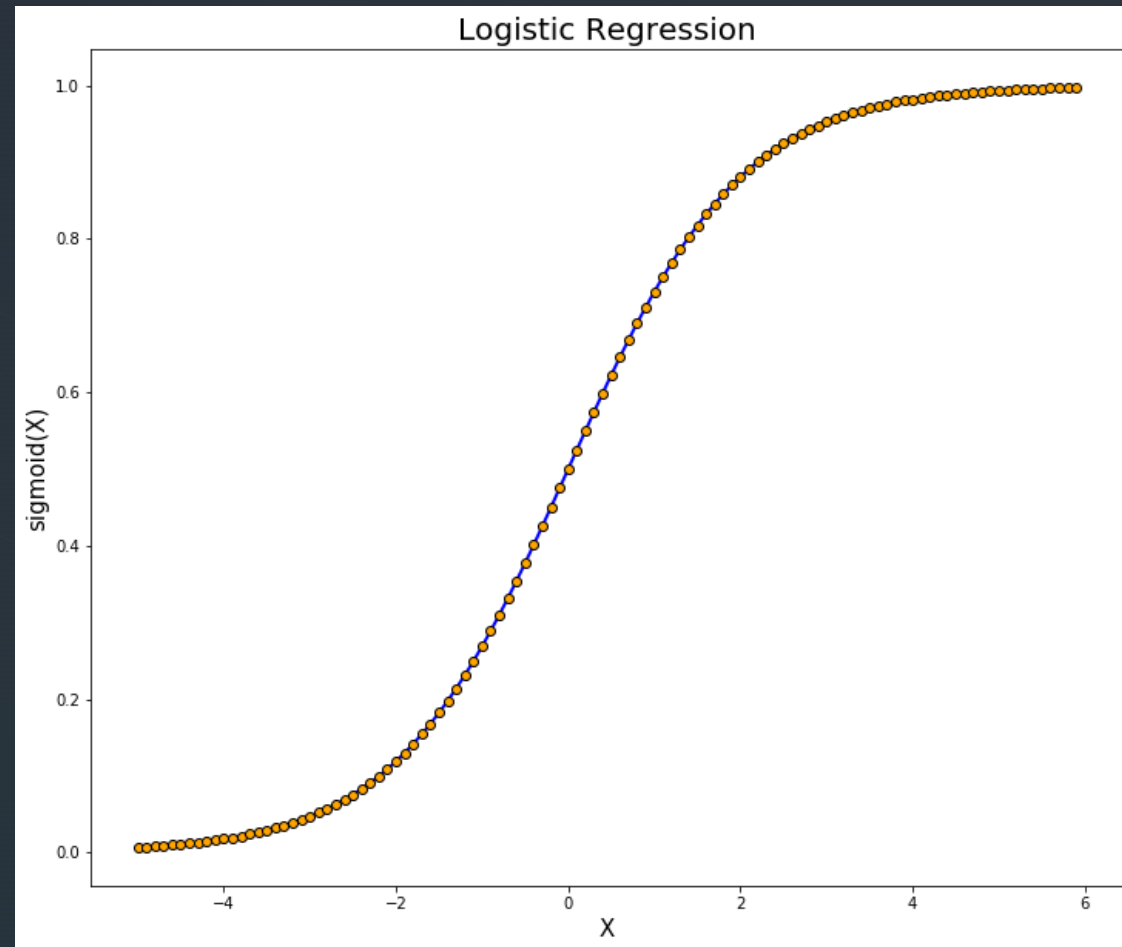
- Can generate a model with high accuracy, it is nowhere near as profitable.
- <https://towardsdatascience.com/machine-learning-for-sports-betting-not-a-basic-classification-problem-b42ae4900782>
- This work focused on a Neural Network approach and shows some interesting data surrounding accuracy and payout strategies which we are hoping to incorporate.
- <https://towardsdatascience.com/predicting-the-outcome-of-nba-games-with-machine-learning-a810bb768f20>
- Best published model had a prediction accuracy of 74.1% (for playoff outcomes), with most others achieving an upper bound between 66–72% accuracy.
- The upset rate across the entire season in the NBA averages 32.1% In the playoffs, the upset rate — as defined by teams with a lower regular season record winning— drops to 22% (which actually means most NBA-playoff prediction models underperform).



# Approach to Problem

## Logistic Regression

- Binary Classifier  
 $P(<0.5) = \text{False}$   
 $P(>0.5) = \text{True}$
- Fits data to sigmoid function  
 $\frac{1}{1+e^{-1(a_n * x_n)}}$
- Data is fitted using complex loss function:
- Test-train split, 75/25
- Used sklearn library



# Gathering Data

- Used NBA api for datasource, and Pandas for cleaning and storing data
- Gathered 25000 games from 2000-2021, limited by API timeouts
- L.R. Assumes gaussian distribution, so we scaled (standardized) data
- Independent variables should have minimal covariance to avoid overfitting

	WL_x	AST_x	BLK_x	DREB_x	FG3_PCT_x	FG_PCT_x	FT_PCT_x	FTA_x	OREB_x	STL_x	TOV_x
WL_x	1.000000	0.155784	0.115995	0.213477	0.215614	0.277687	0.073141	0.046981	-0.073407	0.056115	-0.152668
AST_x	0.155784	1.000000	0.168657	0.238589	0.294753	0.532329	0.103110	-0.119583	-0.169835	0.224997	-0.020236
BLK_x	0.115995	0.168657	1.000000	0.257729	0.050425	0.218304	-0.066410	0.158538	0.067682	0.069628	0.086349
DREB_x	0.213477	0.238589	0.257729	1.000000	0.272993	0.293473	0.036625	0.092212	-0.114150	-0.237396	-0.019249
FG3_PCT_x	0.215614	0.294753	0.050425	0.272993	1.000000	0.530398	0.246618	-0.176668	-0.371147	-0.094857	-0.268670
FG_PCT_x	0.277687	0.532329	0.218304	0.293473	0.530398	1.000000	0.133609	0.081331	-0.335767	0.123195	-0.082234
FT_PCT_x	0.073141	0.103110	-0.066410	0.036625	0.246618	0.133609	1.000000	-0.200609	-0.271647	-0.075060	-0.261923
FTA_x	0.046981	-0.119583	0.158538	0.092212	-0.176668	0.081331	-0.200609	1.000000	0.252697	0.175915	0.286051
OREB_x	-0.073407	-0.169835	0.067682	-0.114150	-0.371147	-0.335767	-0.271647	0.252697	1.000000	0.053227	0.183405
STL_x	0.056115	0.224997	0.069628	-0.237396	-0.094857	0.123195	-0.075060	0.175915	0.053227	1.000000	0.182769
TOV_x	-0.152668	-0.020236	0.086349	-0.019249	-0.268670	-0.082234	-0.261923	0.286051	0.183405	0.182769	1.000000

# Playoff Correlative Matrix

	WL_x	AST_x	BLK_x	DREB_x	FG3_PCT_x	FG_PCT_x	FT_PCT_x	FTA_x	OREB_x	STL_x	TOV_x	PLUS_MINUS_x
WL_x	1.000000	0.169813	0.073992	0.285347	0.197222	0.286583	0.148750	-0.177306	0.036004	-0.207725	0.099764	0.794647
AST_x	0.169813	1.000000	0.441108	0.745891	0.673895	0.723168	0.199383	-0.674583	-0.518286	0.329926	0.187090	0.258418
BLK_x	0.073992	0.441108	1.000000	0.503472	0.379269	0.728018	0.076814	-0.076794	-0.289449	0.439896	0.431860	0.137537
DREB_x	0.285347	0.745891	0.503472	1.000000	0.622781	0.727702	0.334007	-0.511703	-0.143442	-0.017653	0.313176	0.391127
FG3_PCT_x	0.197222	0.673895	0.379269	0.622781	1.000000	0.486693	0.471583	-0.399424	-0.495661	0.297469	-0.198038	0.181205
FG_PCT_x	0.286583	0.723168	0.728018	0.727702	0.486693	1.000000	0.032592	-0.483723	-0.308922	0.091349	0.352883	0.434009
FT_PCT_x	0.148750	0.199383	0.076814	0.334007	0.471583	0.032592	1.000000	-0.311999	-0.133935	-0.048430	-0.135900	0.054773
FTA_x	-0.177306	-0.674583	-0.076794	-0.511703	-0.399424	-0.483723	-0.311999	1.000000	0.546486	0.173809	-0.004572	-0.290715
OREB_x	0.036004	-0.518286	-0.289449	-0.143442	-0.495661	-0.308922	-0.133935	0.546486	1.000000	-0.377782	0.409437	0.023828
STL_x	-0.207725	0.329926	0.439896	-0.017653	0.297469	0.091349	-0.048430	0.173809	-0.377782	1.000000	0.055564	-0.213802
TOV_x	0.099764	0.187090	0.431860	0.313176	-0.198038	0.352883	-0.135900	-0.004572	0.409437	0.055564	1.000000	0.154874
PLUS_MINUS_x	0.794647	0.258418	0.137537	0.391127	0.181205	0.434009	0.054773	-0.290715	0.023828	-0.213802	0.154874	1.000000



# Evaluation and Effectiveness

- Split dataset into train and test sets, fit the Logistic Regression model to the train set, then find accuracy on the test set
- Tried a lot of different iterations of the features, using all in the set was most effective
- Improved to 66.2% current accuracy
- Next, we trained the model using only regular season games, and tested it on the postseason
- Correctly predicted the postseason winners 61.6% of the time

```
[ [2055  509 ]  
  [ 976  887 ] ]
```

```
precision on test set: 0.6215294690696541  
recall on test set: 0.43969676085458304  
F1 on test set: 0.5150353178607466
```

# Discussion and Improvement

- Next steps:
  - Incorporate betting lines and predict the outcomes of postseason games. Use our model's guidance to "bet" on games where the ML agrees with our model's prediction
  - Observe our wins and losses cumulatively, see if our model beats Vegas this off season
  - Using `predict_proba`, we yielded a 56% chance of Celtics loss in this matchup, using both teams stats from this year
  - Only calculates for one game, would need to refine to calculate for entire 7 game series