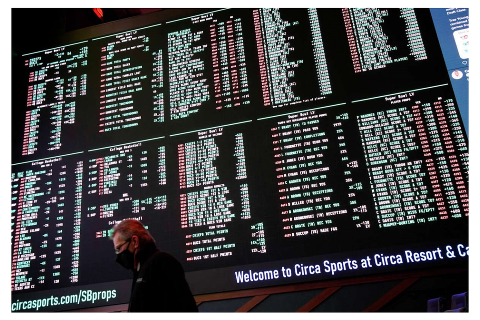
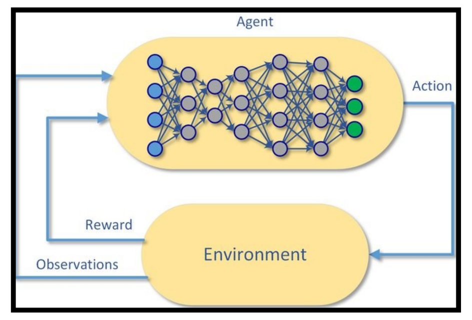
IF (A)I WERE A BETTING MAN: PROFITABLE SPORTS BETTING WITH DEEP REINFORCEMENT LEARNING

Team members: Anthony Weng {ad2weng@stanford.edu}

Predicting

- With the rise of digital sportsbook platform, sports betting is more popular than ever!
- Professional bettors will often employ complex systems which operate off of proprietary knowledge to place their profitable bets.
- We want to see if an ML approach can recover a profitable betting strategy using only public information!
- To accomplish this, we use deep reinforcement learning (DRL) to teach an agent to bet using an event's odds/moneyline history.


+


=

?

Basically: public odds data + machine learning = jackpot?

- Ultimately, our DRL agent doesn't beat a zero-profit strategy (beating favorites-only), but it shows promise for an ML approach!

Data

- The original dataset used in this work was collected by Lisandro Kaunitz, Shenjun Zhong and Javier Kreiner, as a part of their work in **"Beating the bookies with their own numbers - and how the online sports betting market is rigged."**
- Broadly, the dataset contains hourly-sampled odds figures on 92,647 soccer matches from 1005 leagues around the world.

Raw dataset attributes:		Cleaned dataset attributes:
Hourly odds by match.		Hourly odds by match.
Odds for home win, away win, and tie.	→	Odds for **only home win and away win kept**.
Odds figures from up to 32 bookmakers for each match.	→	Odds figures are the **median** numbers from the up to 32 bookmakers.
Details about match included: date and time, league, teams, score, etc.	→	Odds converted to moneyline figures.
		No match details kept.

Results

Table 1. General test set statistics.

Strategy	Test set bets made	Profitable bet rate (%), # profitable bets made	Overall profit ($)	Avg. amount won ($)	Avg. amount lost ($)	Largest amount won ($)
Favorite-only	1000	68.1 (681)	43,040.00	110.03	100.00	170.00
Random	125	55.2 (69)	-7,980.00	166.96	348.21	750.00
Logistic Regression	1000	65.4 (654)	17,652.90	40.44	25.41	108.24
Deep Q-learning	1000	58.6 (586)	24,592.08	126.91	120.24	660.70

Table 2. Performance on match types of interest.

	Underdog upset		Closely contested		Clear favorite to win	
Strategy	Hit rate (%)	Avg. winnings ($)	Hit rate (%)	Avg. winnings ($)	Hit rate (%)	Avg. winnings ($)
Favorite-only	NA	NA	60	119.85	77	100.00
Random	44	257.41	57	156.25	57	177.50
Logistic Regression	36	14.44	56	17.81	76	59.57
Deep Q-learning	34	230.27	51	134.42	68	119.55

- The deep reinforcement learning model **did not** outperform the favorites-only strategy for many metrics.
- It also performed poorly on match types of interest.
- The DRL approach did perform better than the logistic regression baseline in many desirable criterion.

Train/dev/test splits:

- Train: ~6K entries.
- Dev: ~3K entries.
- Test: ~1K entries.

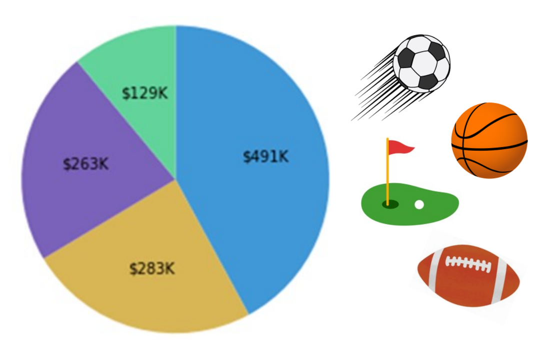
Discussion & Future Work

We observe two principal findings regarding the performance of the DRL agent relative to logistic regression and favorites-only agents:

- The DRL agent outperformed the logistic regression agent in terms of total profit, despite it achieving a significantly lower profitable bet rate (-6.8%).
- This suggests that, while the DRL agent was correct less often, it was "more correct" when it was.
- This was expected since a DNN is much more flexible than a logistic regression, but it's still promising as to the potential of DRL techniques for this domain.

- The DRL agent performed worse than the favorites-only strategy, which in reality should be a near-zero profit strategy.
- This was unexpected, especially since the DRL agent could (in theory) learn to replicate the favorites-only strategy.
- We found our DRL agent was only betting on the home team, suggesting that our DNN architecture was too simplistic to learn from the already limited training data.

Future works in this space could:



Bet on multiple events to create a risk-minimizing bet portfolio!

Incorporate public, sport-specific information!

Teach agents to bet on non-binary outcomes for realism!

Key References

[1] Lisandro Kaunitz, Shenjun Zhong, and Javier Kreiner. Beating the bookies with their own numbers - and how the online sports betting market is rigged, 2017.

[2] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[3] Andre Cornman, Grant Spellman, and Daniel Wright. Machine learning for professional tennis match prediction and betting. *Stanford Unverisity*, 1(2):4, 2017.

[4] Fredrik A Dahl. A reinforcement learning algorithm applied to simplified two-player texas hold'em poker. In *Machine Learning: ECML 2001: 12th European Conference on Machine Learning Freiburg, Germany, September 5–7, 2001 Proceedings 12*, pages 85–96. Springer, 2001.

[5] Sascha Wilkens. Sports prediction and betting models in the machine learning age: The case of tennis. *Journal of Sports Analytics*, 7(2):99–117, 2021.

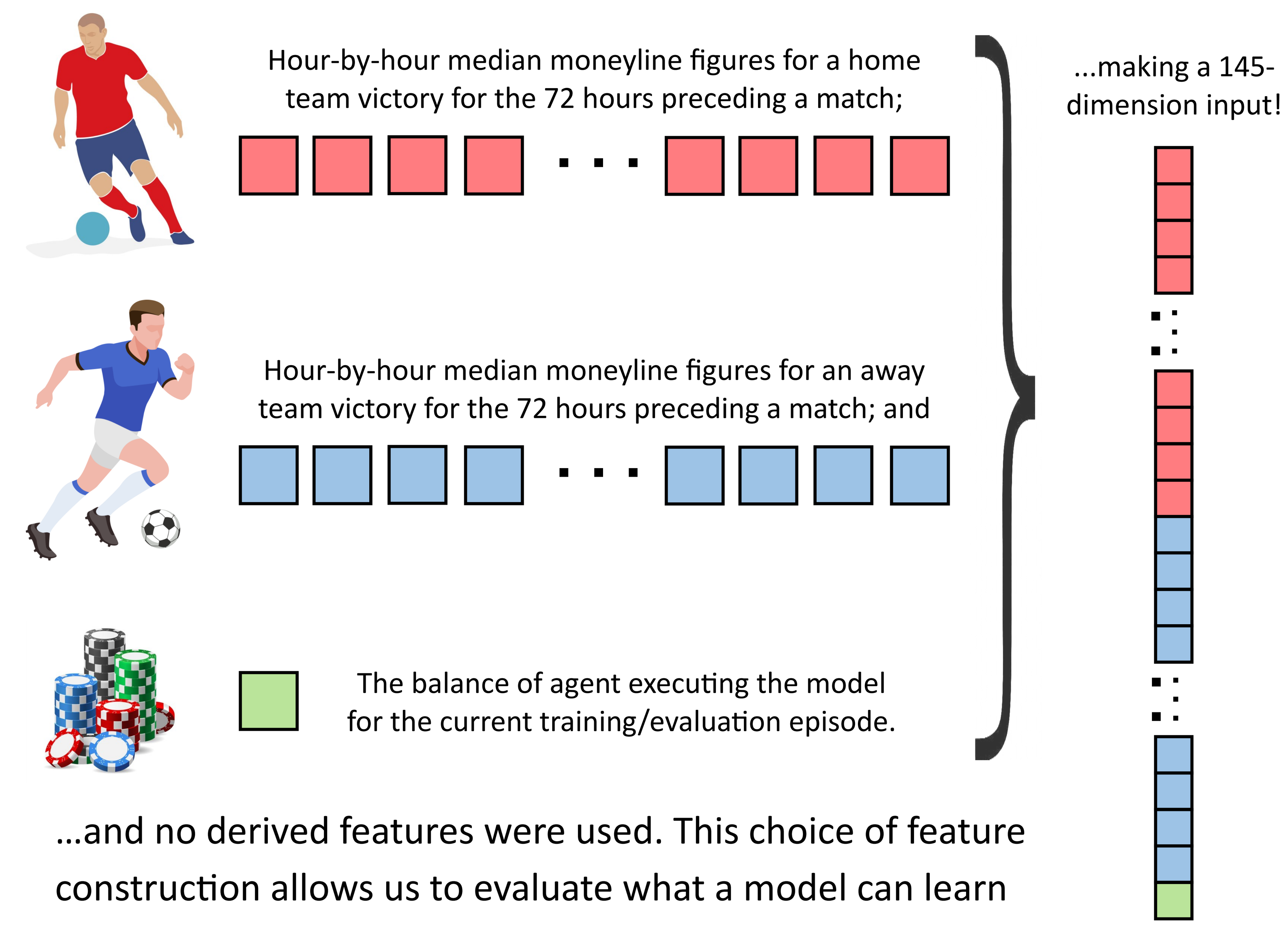
[6] Conor Walsh and Alok Joshi. Machine learning for sports betting: should forecasting models be optimised for accuracy or calibration? *arXiv preprint arXiv:2303.06021*, 2023.

Image Credits

- Deep Q (models, top): https://www.novatec-gmbh.de/wp-content/uploads/reinforcement_learning_loop-650x294.png
- Deep Q (models, bottom): https://cdn.analy-ticsvidhya.com/wp-content/uploads/2019/04/Screenshot-2019-04-16-at-5.46.01 PM.png
- Logistic regression: https://pimages.toolbox.com/wp-content/uploads/2022/04/11040522/46-4.png
- Favorites-only betting: https://kiowacasino.com/wp-content/uploads/2018/07/KIO_Casino_TableGames_09_gaming-1.jpg
- Random betting: https://dnycf48t040dh.cloudfront.net/Random-Module-Python.jpeg
- Soccer player (red): https://www.uidownload.com/files/92/449/320/soccer-player-vector-image-2-thumb.jpg
- Soccer player (blue): https://static.vecteezy.com/system/resources/previews/019/012/064/original/soccer-football-player-png.png
- Poker chips: https://media1.thehungryjpeg.com/thumbs2/ori_3903623_613u1k5975xzlxujwcrjcmkssvbqsz4t4o48s8xo_stack-of-casino-chips.jpg
- Odds board: https://s.hdnux.com/photos/01/30/56/03/23257313/3/1200x0.jpg
- DRL diagram (predicting): https://i.imgur.com/gsXfI91.jpg
- Jackpot: https://cdn.mos.cms.futurecdn.net/PdEx865YaM4EJVcNbHF4ng-415-80.jpg
- Pi graph: https://chartio.com/learn/charts/pie-chart-complete-guide/
- Soccer ball: https://illustoon.com/photo/thum/4581.png
- Basketball: https://upload.wikimedia.org/wikipedia/commons/thumb/7/72/Basketball_Clipart.svg/1200px-Basketball_Clipart.svg.png
- Golf: https://us.123rf.com/450wm/martialred/martialred1809/martialred180900005/108396845-golf-course-green-with-flag-or-flagstick-and-golf-ball-flat-vector-color-icon-for-sports-apps-and.jpg?ver=6
- Football: https://static.vecteezy.com/system/resources/previews/011/064/977/original/american-football-clipart-simple-american-football-watercolor-style-illustration-isolated-on-white-background-rugby-ball-cartoon-hand-drawn-style-american-football-design-vector.jpg
- Jamal Murray stat line: https://preview.redd.it/cxuckhy0smi61.jpg?auto=webp&s=f52069388b7c336032e9928957cf7128311e85f7
- Tim Duncan stat line: https://pbs.twimg.com/media/DCXuFVeW0AApvzx.jpg
- Horse race: https://horserookie.com/wp-content/uploads/2022/04/Racehorses-Canva.jpg

Features

For our trainable models/strategies, the raw features consist of:



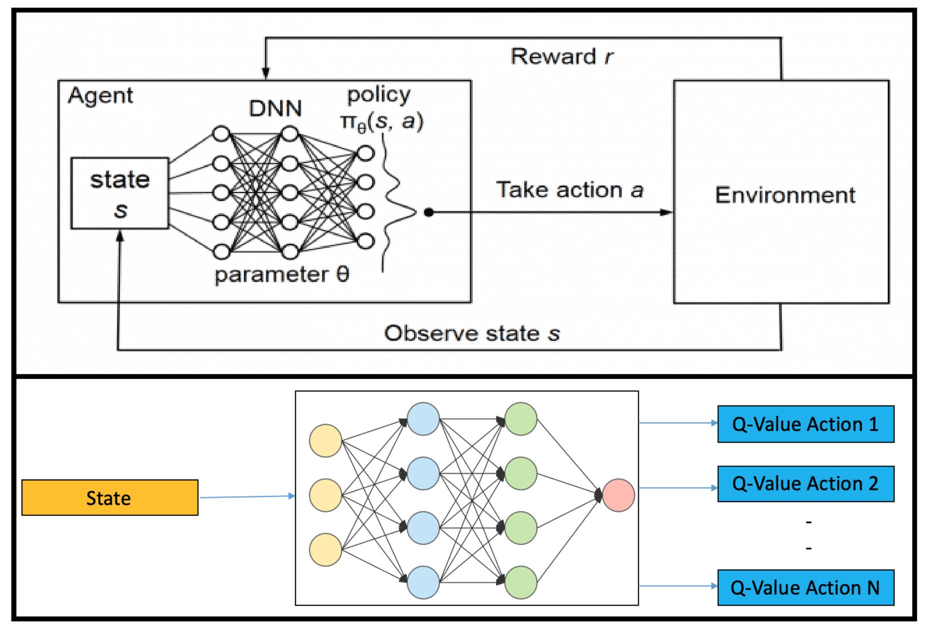
...and no derived features were used. This choice of feature construction allows us to evaluate what a model can learn from only **publicly-available**, **sport-agnostic** information.

Models

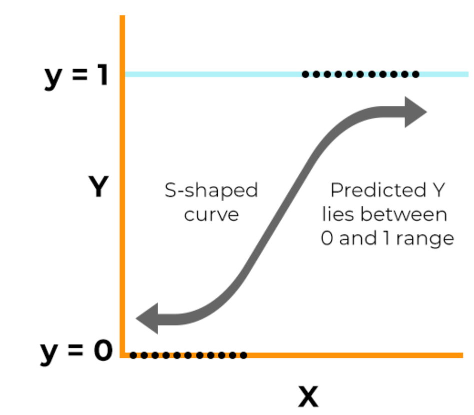
Our primary model/approach for DRL is a **deep Q-network.**

Two key techniques we use in our implementation are:

1. Use of training and target networks.
2. An **ε**-greedy action selection regime

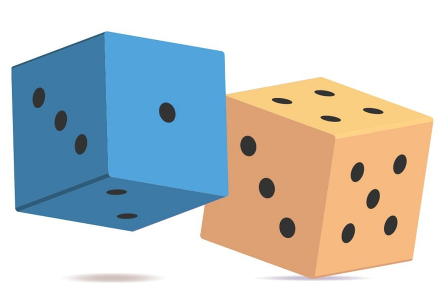


We also implement three baseline models/strategies:



1. Logistic Regression

2. Favorites-only betting



3. Random betting