# ‘Moneyballing’ AFL Fantasy

Predicting Breakout Fantasy Seasons in AFL Players with PyTorch MLPs

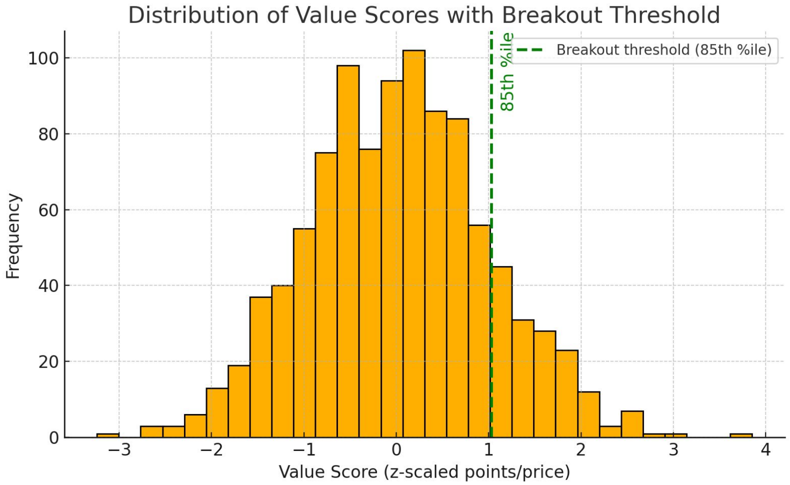
Rob Konstanty-Nicholson - 2/6/25

# Fantasy football in 60 seconds

- Buy team of players using set amount of in-game salary
- “Fantasy points” based on real-world stats
- The team with the most total points wins
- Player price is set by a formula based on recent performance
- Win by maximising points per dollar


# Applying ML - Why “Breakout” Player-Seasons?

- **Data volume:** ~800 players -> 22 player team
- **Mis-priced players:** Player pricing is a lagging indicator, formula based on previous season performance
- **“Breakout”** = Top 15% Ave. points per dollar
- Why classification not regression?



# Data Pipeline: 9 Yrs + 10k player seasons

## Web Scraping

- No APIs available
- afltables.com (requests + BeautifulSoup)
- dreamteamtalk.com (selenium)

## Raw CSVs

- File per season - fantasy stats
- File per player, each row a season of game stats

## Feature Build (.feather)

- Aggregate both datasets
  - Name inconsistencies
  - Repeated players
- Aggregate into features (next slide)
- Feather format - efficiency through columnar layout

## Train / Val Split

- Split by season to avoid look-ahead bias
- Set aside most recent full season (2024)

# Features

**Time Frame**

Split into one year or three year averages to best highlight long-term trends

- last_year_avg_
  Or
- 3yr_avg_

**Statistic**

23 statistics from in-game

Average per game over the time period

- _KI (kicks)
- _GO (goals) etc.

**Metadata**

- Number of games
- Average fantasy points
- Fantasy price
- Flag missing seasons
- Label: Breakout = 0 or 1

# Model Architecture and Training

## 4-Layer MLP: 256 → 128 → 64 → 32 → 1

**Input**
~50 input features

**FC 256**
BN ReLU Dropout 0.3

**FC 128**
BN ReLU Dropout 0.3

**FC 64**
BN ReLU Dropout 0.3

**FC 32**
BN ReLU Dropout 0.3

**Logit 1**

### Training Setup

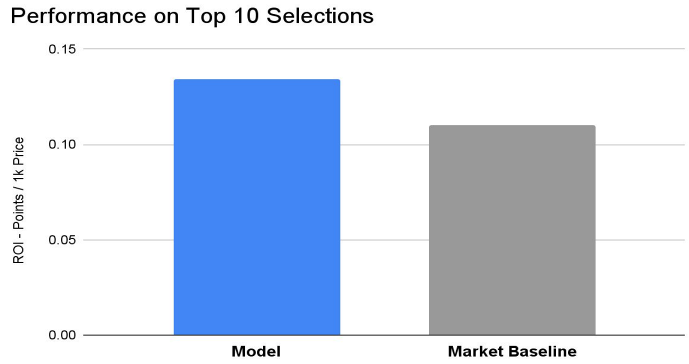
- Loss: BCEWithLogitsLoss
- Optimiser: Adam, LR0.001
- Batch = 64
- Epochs <= 1000, Early-stop 25

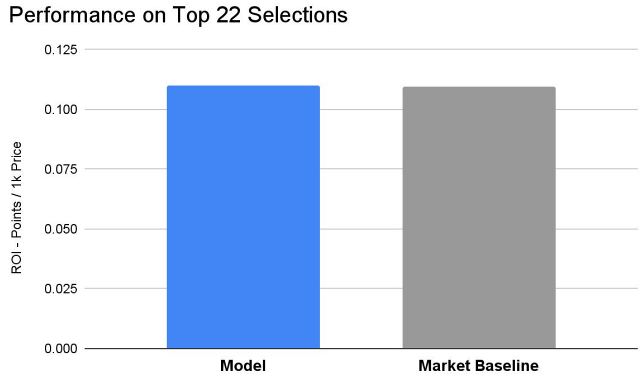
### Regularisation

- Dropout 0.3 for hidden layers
- BatchNorm

# Performance: Does the Model Pick Winners?

- ROC-AUC = 0.67 - beats random baseline
- Top 10 picks are strong (+22% on ave. market value)
- However model usefulness deteriorates with more picks
- Helpful assistant tool, but cannot be delegated full team selection responsibility



# Challenges & Next Steps

| Issue | Possible Improvement |
|---|---|
| Older Player Bias / Lack wider player context | Career-context features<br>Unsupervised classification of player types |
| Season-level granularity | Momentum feature<br>Game-by-game RNN |
| Single validation fold | Expand number of years in dataset<br>K-fold CV |
| Threshold value untested | Experiment labelling top 5%, 10%, 20% player seasons breakouts |

## Summary

- Fantasy context = Breakout framing provides practical value
- 50+ feature MLP lifts ROI + 22% on market ave.
- Useful, but not sufficient
- Ready for future hyperparameter and feature tuning, dataset expansion

# Thank you! - Questions?