A comps-based approach for interpreting tree-based predictions with an application to the NFL draft

Elisabeth Millington[1], Scott Powers[2]

[1]Rice University, Department of Kinesiology, [2]Rice University, Department of Sport Management

RICE UNIVERSITY Sport Analytics

Introduction

- Random Forests are a powerful machine learning prediction model, but like other "black box" algorithms, their results are difficult to interpret.
- In sports, interpretability is crucial because it builds trust for the user in the predictions, and front offices need to consider non-quantifiable information in their decision-making.
- We aim to use the *k*-NN properties of random forests to add interpretability to the QBR predictions produced for quarterback prospects in the form of "player comps".

Data

- The data used in this project consists of publicly available data from Sports Reference
- Our dataset includes 2,099 quarterbacks. The data for each player-season includes team, conference, strength of schedule, games played, passing statistics (attempts, completions, touchdowns, interceptions), rushing statistics (attempts, yards, touchdowns), and awards (All-America designation, Heisman voting).
- In addition to college statistics, we obtained each player's NFL passing attempts and Total Quarterback Rating (QBR) (Burke, 2016). QBR is based on expected points added (EPA), and considers each quarterback's share of their team's EPA and accounts for home-field advantage, defensive strength, and garbage time.
- We regressed each player's QBR/season to zero to mitigate the effects of players who put up very high QBR numbers in very small samples.

Software

- R package treecomp.
- Extracts similarity scores from random forests.
- Open-source: github.com/elisabethmill/treecomp

Methods

Random Forest Modeling

1. We used a random forest model to predict average NFL QBR based on college quarterback statistics.
   - Response Variable: Average NFL QBR across seasons (0 for players who never played in the NFL)

Random Forest as Adaptive Nearest Neighbors

1. Building on the work of Lin & Jeon (2006) we interpret the random forest model as a data-adaptive weighted k-nearest neighbors (k-NN) algorithm.
   - Let $\vec{x}_1, \dots, \vec{x}_n \in \mathbb{R}^p$ be the training feature vectors and $y_1, \dots, y_n \in \mathbb{R}$ the corresponding outcomes (NFL QBR). We draw B bootstrap samples and train a decision tree on each sample $b \in \{1, \dots, B\}$.
   - Let $n_{b,i}$ be the number of times observation i appears in bootstrap sample b
   - Let $\mathcal{T}_{b,i}$ denote the terminal node of observation i in tree b
   - Let $|\mathcal{T}_{b,i}|$ be the number of observations in that node (with repetition)
2. Given a new query point $\vec{x}_0$, the prediction of tree b is:
   1. $\hat{y}_b(\vec{x}_0) = \sum_{i=1}^{n} w_{b,0,i} \cdot y_i$ *where* $w_{b,0,i} = \frac{n_{b,i} \cdot \mathbb{I}\{\mathcal{T}_{b,0} = \mathcal{T}_{b,i}\}}{|\mathcal{T}_{b,0}|}$
3. The random forest prediction is the average across trees:
   - $\hat{y}(\vec{x}_0) = \frac{1}{B}\sum_{b=1}^{B} \hat{y}_b(\vec{x}_0) = \sum_{i=1}^{n} \bar{w}_{0,i} \cdot y_i$ *with* $\bar{w}_{0,i} = \frac{1}{B}\sum_{b=1}^{B} w_{b,0,i}$
4. This formulation shows that a random forest defines a custom neighborhood for $\vec{x}_0$, where observations that frequently land in the same leaf across trees are assigned higher weights.

Similarity Score

1. We interpret $\bar{w}_{0,i}$ as a similarity score between the query point $\vec{x}_0$ and a training point $\vec{x}_i$:
   - $\bar{w}_{0,i} = \frac{1}{B}\sum_{b=1}^{B} \frac{n_{b,i}}{|\mathcal{T}_{b,0}|} \cdot \mathbb{I}\{\mathcal{T}_{b,0} = \mathcal{T}_{b,i}\}$
2. Higher similarity implies more frequent co-occurrence in terminal nodes

Results

The random forest model achieved a test RMSE of 8.61, explaining 43.7% of the variance in regressed NFL QBR.

Key predictors of NFL success:

- Final-season Heisman voting
- Passing stats per season (yards, TDs, completions)
- Final-season strength of schedule

High multicollinearity exists between some variables (e.g., yards, touchdowns, and completions per season are all pairwise correlated at ≥ 0.94).

Application to 2025 NFL Draft

The predictions for the top four NFL prospects pre-draft were

- Cam Ward: 37.0
- Dillon Gabriel: 34.6
- Shedeur Sanders: 33.2
- Jaxson Dart: 24.0

Cam Ward		Dillon Gabriel		Shedeur Sanders		Jaxson Dart	
Comp	Score	Comp	Score	Comp	Score	Comp	Score
Johnny Manziel	2.3%	Mason Rudolph	2.0%	Mason Rudolph	1.9%	Teddy Bridgewater	1.9%
Marcus Mariota	2.3%	Dwayne Haskins	1.9%	Kenny Pickett	1.9%	Tajh Boyd	1.6%
Baker Mayfield	2.2%	Kenny Pickett	1.9%	Philip Rivers	1.9%	Russell Wilson	1.5%
Philip Rivers	2.1%	C.J. Stroud	1.9%	Ben Roethlisberger	1.8%	John Beck	1.5%
Trevor Lawrence	2.0%	Andrew Luck	1.9%	Case Keenum	1.8%	Drake Maye	1.4%
Matt Leinart	2.0%	Bo Nix	1.9%	Russell Wilson	1.8%	Zach Terrell	1.4%
Russell Wilson	2.0%	Trevor Lawrence	1.9%	Trevone Boykin	1.8%	Kevin Hogan	1.2%
Lamar Jackson	1.9%	Philip Rivers	1.8%	Kellen Moore	1.7%	Blake Bortles	1.2%
Deshaun Watson	1.9%	Aaron Rodgers	1.7%	Andrew Luck	1.7%	Sam Howell	1.2%
Case Keenum	1.9%	Bryce Young	1.6%	C.J. Stroud	1.7%	Patrick Mahomes	1.1%

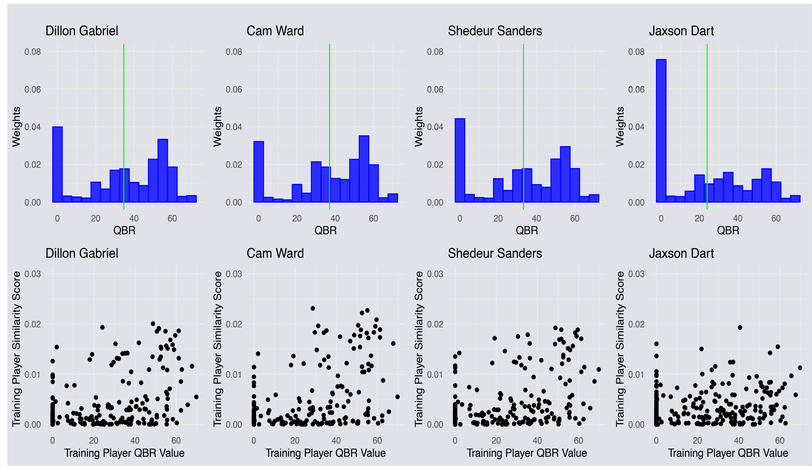

Figure 1: Top: Histogram of training players' regressed QBR values weighted by similarity to the prospect being predicted. The vertical green line annotates each prospect's predicted QBR, which matches the mean of the weighted distribution. Bottom: Scatter plot showing individual similarity scores and regressed QBR for historical prospects.

References:

- Burke, B. (2016). How is total QBR calculated? We explain our (improved) QB rating [September 27, 2016]. ESPN.com. https://www.espn.com/nfl/story/_/id/17653521/how-total-qbr-calculated-explain-our-improved-qb-rating
- Lin, Y., & Jeon, Y. (2006). Random forests and adaptive nearest neighbors. Journal of the American Statistical Association, 101 (474), 578–590. https://doi.org/10.1198/016214505000001230
- Probst, P. (2024). tuneRanger: Tune random forest of the 'ranger' package (Version 0.7). https://cran.r-project.org/web/packages/tuneRanger/index.html
- Wright, M. N. (2024). Ranger: A fast implementation of random forests (Version 0.17.0). https://cran.r-project.org/web/packages/ranger/index.html

Acknowledgement: The authors thank Kevin Meers for suggestions that led to improvements in the random forest model for predicting NFL quarterback prospect success.