Football Dream Team using ML

Identifying player archetypes and predicting team success from football data

Presented by *Nicholas Posborg, Pelle Rechnagel, Sebastian Hjaltelin, Christian Bardram and Søren Gundesen*

UNIVERSITY OF COPENHAGEN

Motivation: why football + ML?

- Football clubs increasingly collect event data: passes, shots, duels, tackles, locations, minutes.
- Raw event data is too detailed to interpret directly.
- ML can compress thousands of actions into player roles, team profiles, and predictive models.
- The project is not only prediction: it also tries to make the prediction interpretable.

Goal of the project

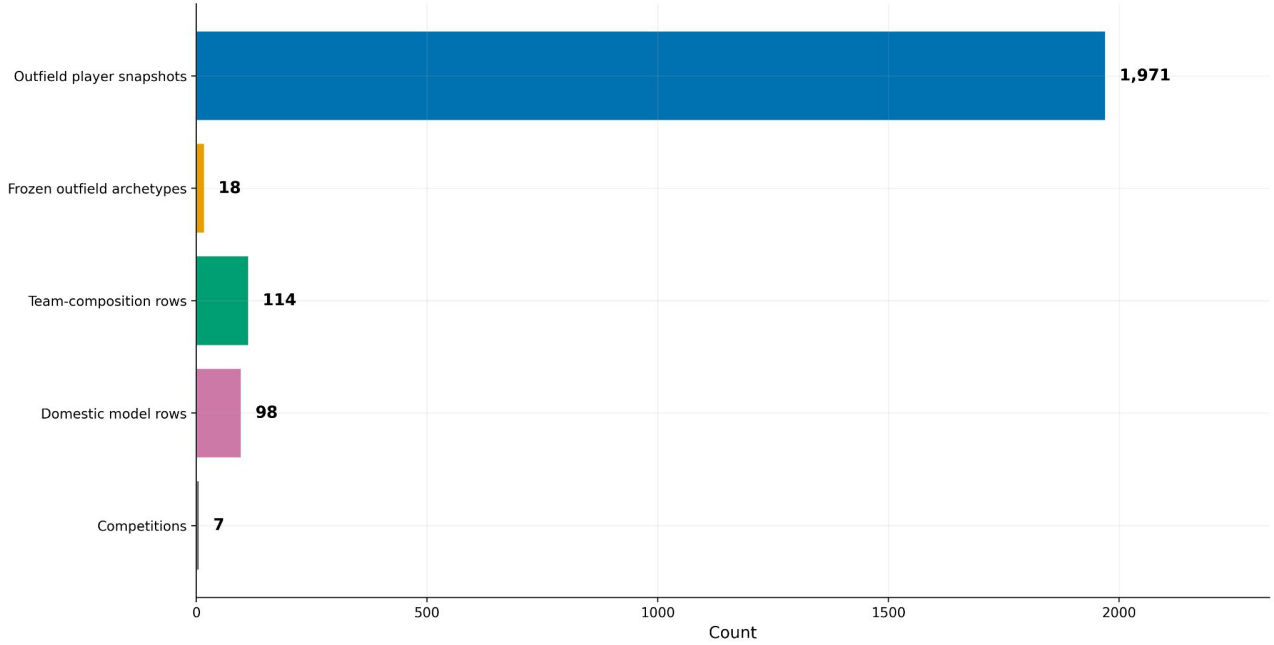
- Build meaningful player archetypes from event data.
- Convert individual player profiles into team-level composition features.
- Predict team success using only team-composition information.
- Evaluate whether the model learns real signal beyond a simple baseline.
- Interpret which archetypes are associated with stronger teams.

Data: what one raw row looks like

- Each raw Wyscout row describes one football action.
- Examples: pass, shot, duel, foul, throw-in.
- Important fields: team, player, action type, minute, start x/y position.
- This is event-level data, not yet ML-ready.
- First challenge: convert many event rows into stable player-level statistics.

Team	Player	Action	Minute	Start x/y
Russia	Y. Gazinskiy	Pass / Simple pass	8	(61, 97)
Senegal	M. Niang	Shot / Shot	18	(88, 34)
Russia	F. Smolov	Duel / Ground loose ball duel	28	(70, 24)
Portugal	Cédric Soares	Free Kick / Throw in	38	(69, 100)
Costa Rica	D. Guzmán	Foul / Foul	52	(50, 42)
Russia	S. Ignashevich	Others on the ball / Touch	64	(9, 41)

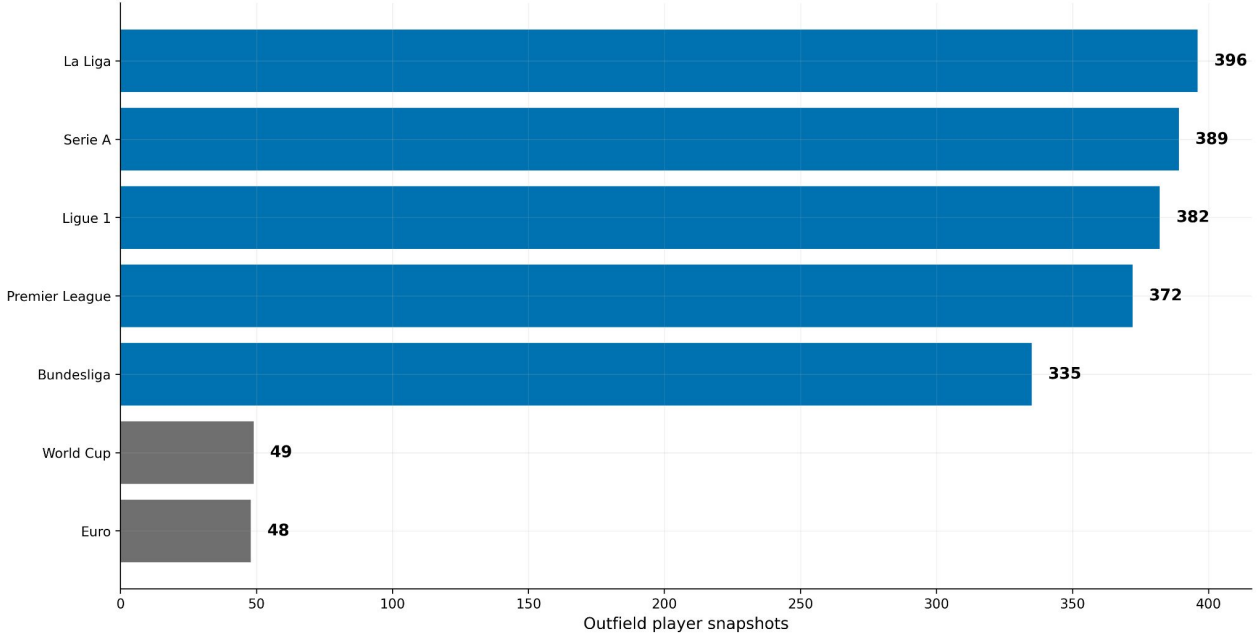
Dataset overview



- Final data contains **1,971 outfield player snapshots**.
- A snapshot is a player-team-competition profile, not necessarily a whole-career identity.
- Players are clustered into **18 frozen outfield archetypes**.
- Team modelling uses **114 team-composition rows**, with **98 domestic model rows**.
- This gives a sizable but still limited supervised dataset.

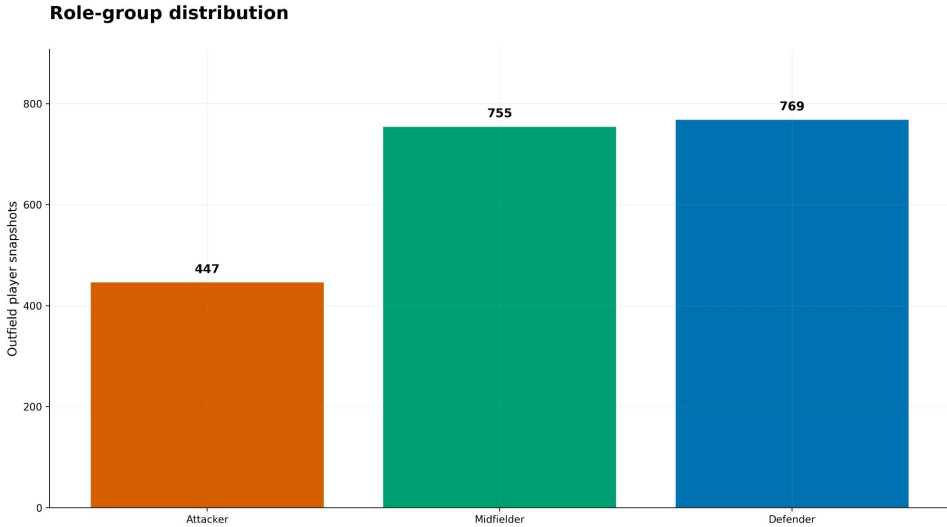
Competition and role coverage

- Data covers **7 competitions**.
- Main domestic leagues are much larger than World Cup / Euro samples.
- Outfield roles are split into attackers, midfielders, and defenders.
- Role balance matters because comparing centre-backs and strikers directly would create misleading clusters.
- Therefore clustering is done in a role-aware way.



Pelle, Sebastian, Søren, Nicholas, Christian

Feature engineering



- Event counts are converted to player features.
- Main features are per-90 rates: passes per90, shots per90, tackles per90, interceptions per90, goals per90.
- Also ratio features: pass accuracy, duel win rate, goal conversion, crossing intensity, shooting intensity.
- Features are standardized before clustering.
- Metadata leakage is avoided: names, teams, nationalities, positions, and success labels are not clustering features.

Method overview

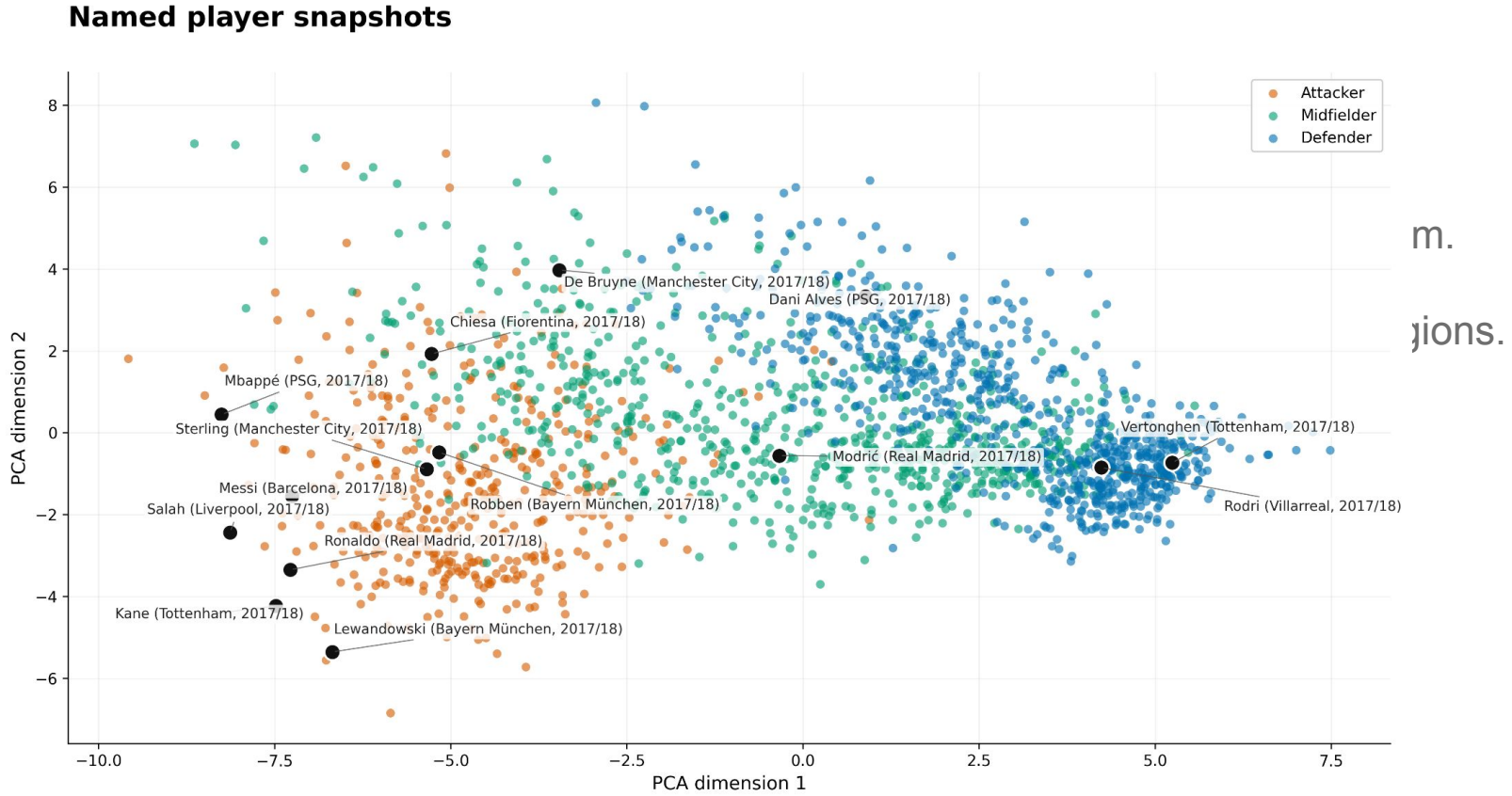
- **Step 1:** Aggregate raw events into player snapshots.
- **Step 2:** Cluster players into archetypes using unsupervised learning.
- **Step 3:** Label clusters using high/low z-score features and player examples.
- **Step 4:** Aggregate teams into archetype minute shares.
- **Step 5:** Predict points per match using supervised regression.
- Some of the topics we used in these steps: preprocessing, clustering, PCA, cross-validation, regularization, ensemble models, and feature importance.

Why clustering?

- Football positions are too coarse: two “midfielders” can have very different roles.
- Clustering finds statistical player profiles from data rather than manually defining them.
- PCA projection shows that attackers, midfielders, and defenders occupy different regions.
- Named examples make the abstract clusters interpretable.
- PCA is only for visualization; clustering uses the full feature space.

Why clustering?

- Foot
- Clus
- PCA
- Nam
- PCA



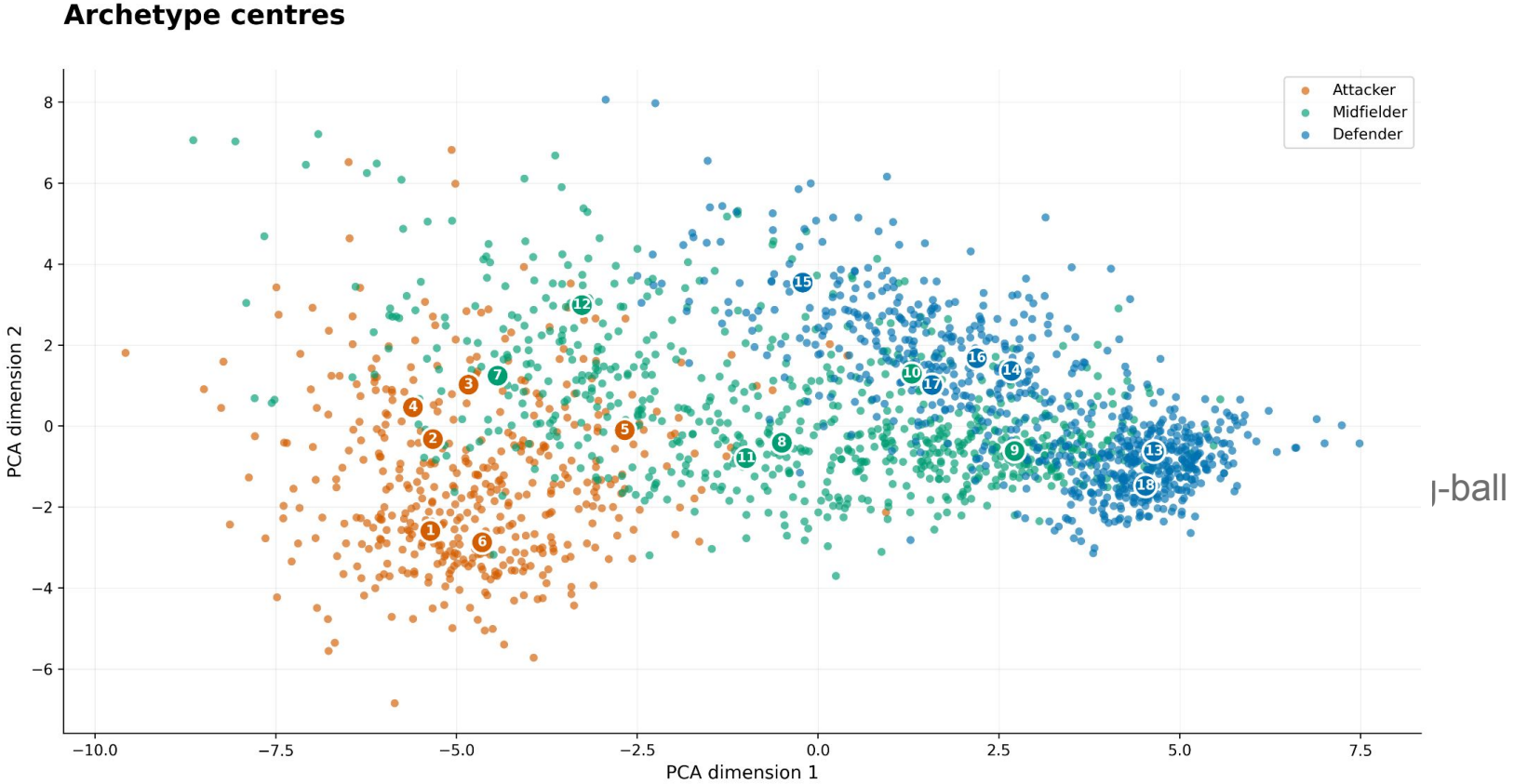
Pelle, Sebastian, Søren, Nicholas, Christian

18 archetypes centres

- The model freezes **18 final archetypes**.
- Archetypes are role-aware: attacking, midfield, and defensive profiles.
- Each centre represents a typical statistical profile.
- This creates a vocabulary for team construction: not just “players”, but “types of players”.
- Examples: possession fullback, creative inside forward, secure possession centre back, long-ball aerial stopper.

18 archetypes centres

- The
- Archi
- Each
- This
- Exan
- aeria



-ball

Pelle, Sebastian, Søren, Nicholas, Christian

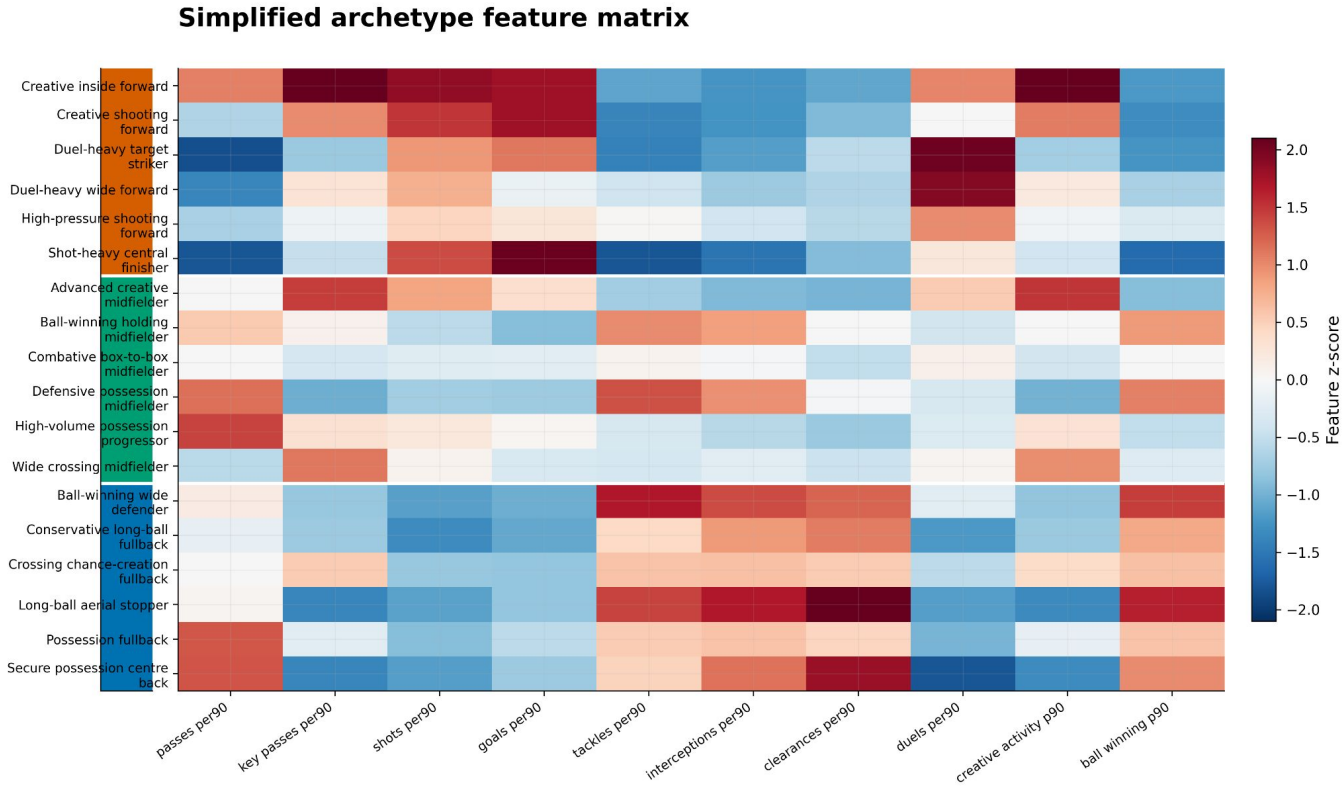
Archetype feature matrix

- Rows are archetypes; columns are football features.
- Red means above-average; blue means below-average, measured as standardized z-scores.
- This explains why each archetype received its name.
- Example: creative forwards have high shots/goals/creative activity; centre-backs have high clearances/interceptions/ball-winning.

Archetype feature matrix

- Rows are
- Red mean
- This expla
- Example:

clearance



ed z-scores.

ave high

From players to teams

- Each team is represented by archetype composition.
- Main team features are minute shares: how much playing time goes to each archetype.
- This is better than raw player names because it generalizes across teams and leagues.
- Example feature: “possession fullback minutes share”.
- Target variable: team success, mainly **points per match**.

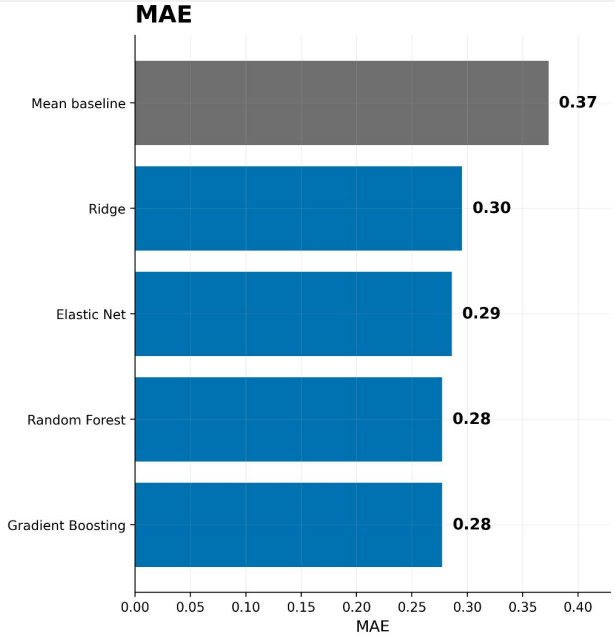
Supervised models used

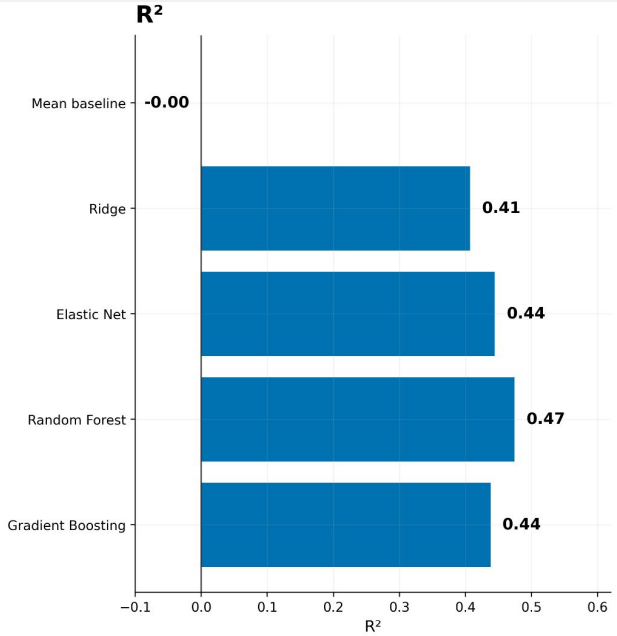
- Baseline: predict mean points per match.
- Ridge: linear model with L2 regularization.
- Elastic Net: linear model with L1 + L2 regularization.
- Random Forest: nonlinear ensemble of decision trees.
- Gradient Boosting: sequential tree ensemble focused on reducing errors.
- Validation: Leave-One-Competition-Out to test transfer across leagues.

Why these methods?

- Ridge/Elastic Net are interpretable and useful with many correlated features.
- Random Forest can capture nonlinear interactions between team-composition features.
- Gradient Boosting is a strong tabular-data method and often performs well on structured data.
- Mean baseline is essential: the ML model must beat “just predict average team success”.
- Leave-One-Competition-Out is stricter than random K-Fold because it tests generalization to unseen competitions.

Main ML performance





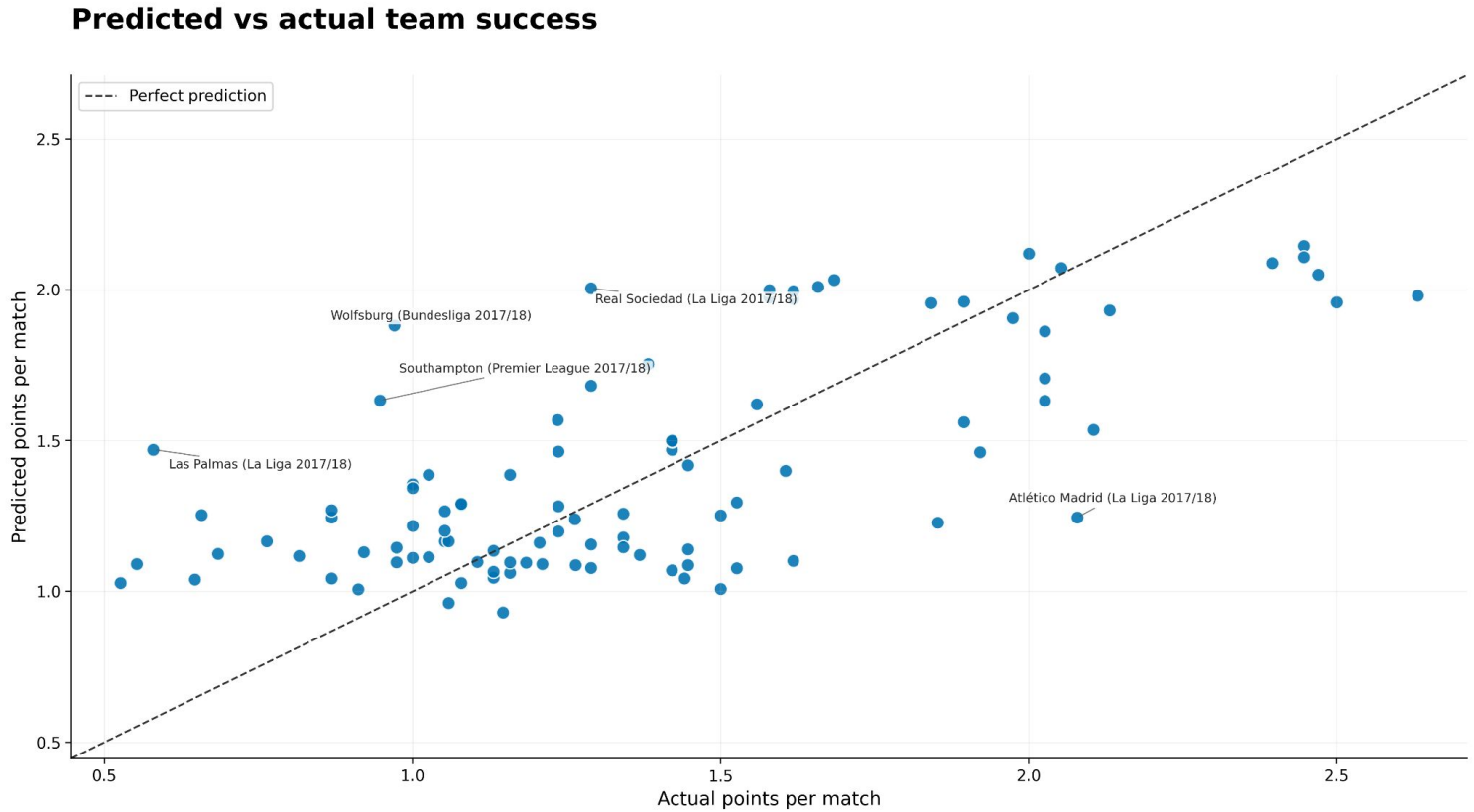
- All ML models beat the mean baseline on MAE.
- Baseline MAE is about **0.37 points per match**.
- Best models reach about **0.28–0.29 MAE**.
- Random Forest gives the strongest R^2, around **0.47**.
- Interpretation: archetype composition explains meaningful variation, but far from all team success.

Predicted vs. actual success

- Each point is a team-competition row.
- Dashed line = perfect prediction.
- The model captures the broad trend: stronger teams often receive higher predicted points.
- But there are large residuals for some teams.
- This shows both the strength and limitation of the model.

Predicted vs. actual success

- Each
- Dash
- The
- But t
- This

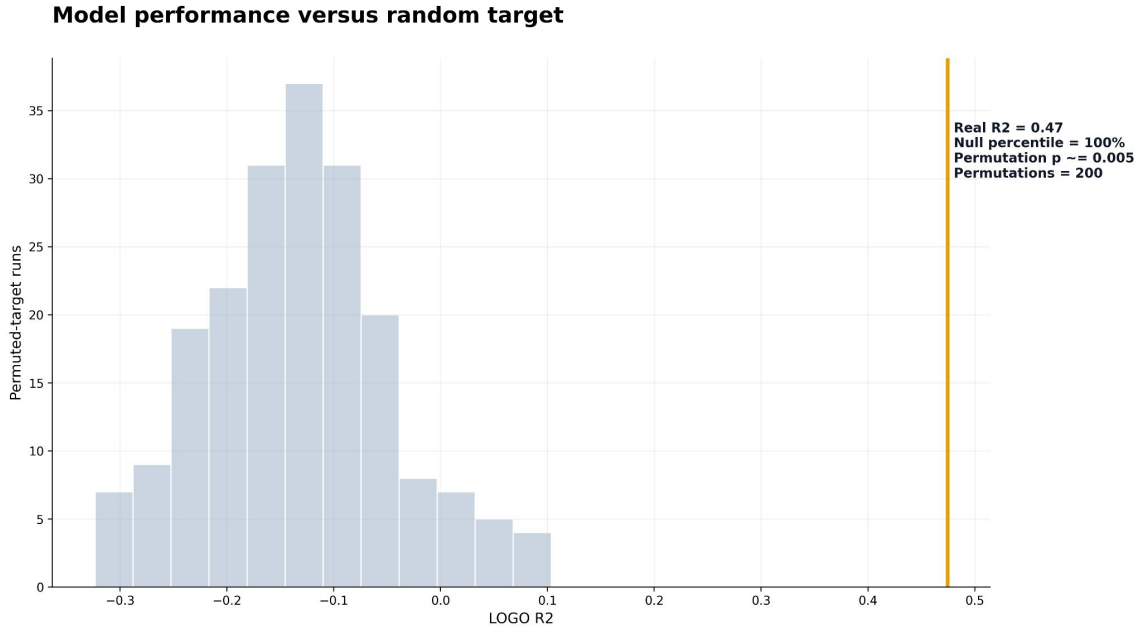


points.

Pelle, Sebastian, Søren, Nicholas, Christian

Validation against random target

- To check if the model learned real structure, the target was permuted many times.
- Random-target models produce much worse R^2.
- Real model: **$R^2 \approx 0.47$**.
- Permutation p-value is about **0.005**.
- This supports that the model learns non-random signal, not just noise.



What player types are associated with success?

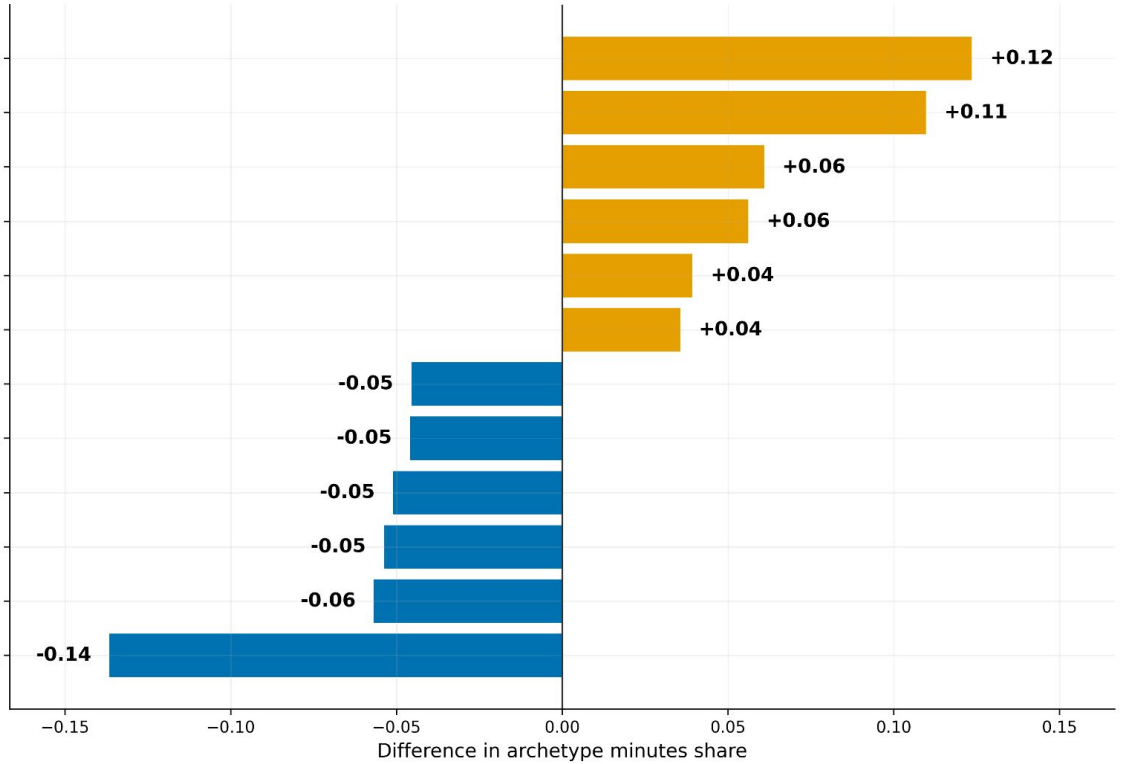
- Stronger teams have higher shares of possession-oriented and creative archetypes.
- Positive differences: secure possession centre backs, possession fullbacks, possession progressors, creative forwards.
- Weaker teams have relatively more long-ball aerial stoppers and conservative long-ball fullbacks.
- This is descriptive, not causal.
- The result suggests successful teams tend to control possession and progress the ball more.

What player types are associated with success?

- St
- Pc
- cr
- W
- Th
- Th

essors,

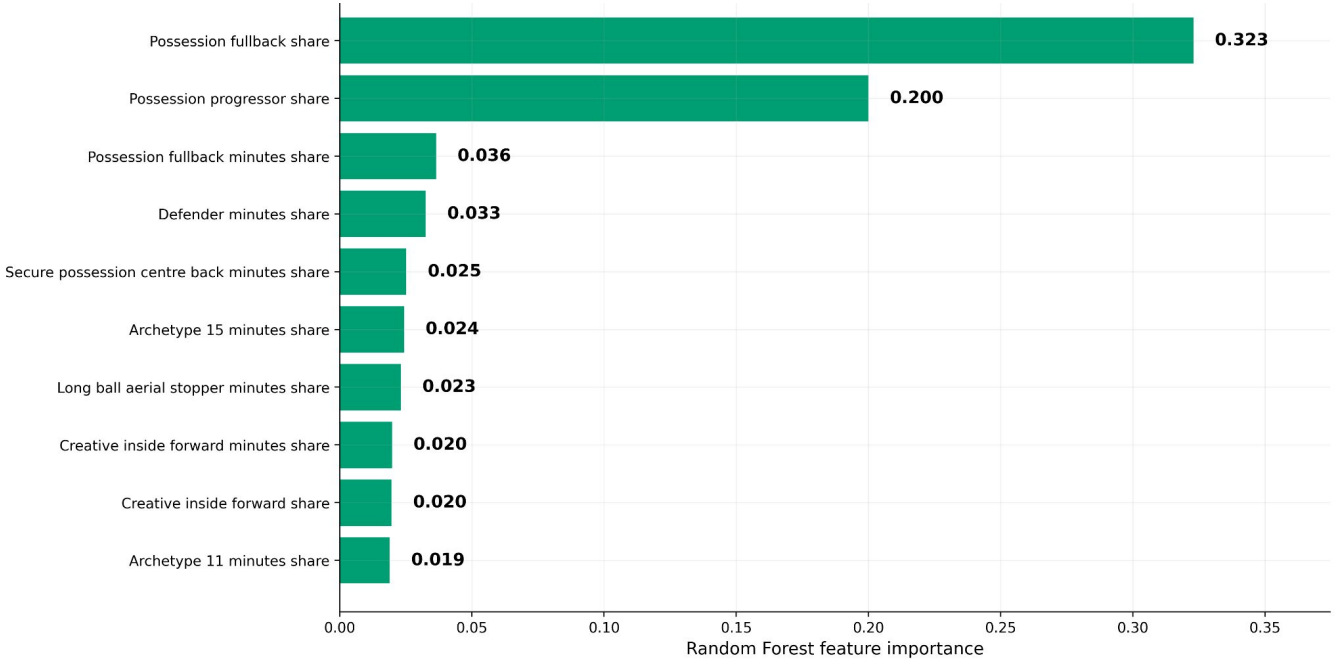
ks.



Pelle, Sebastian, Søren, Nicholas, Christian

Feature importance and interpretation

- Random Forest feature importance shows which composition features matter most for prediction.
- Possession fullback share is the largest feature.
- Possession progressor share is also highly important.
- This supports the pattern from the top-vs-bottom comparison.
- But feature importance is model sensitivity, not proof that adding such players would cause success.



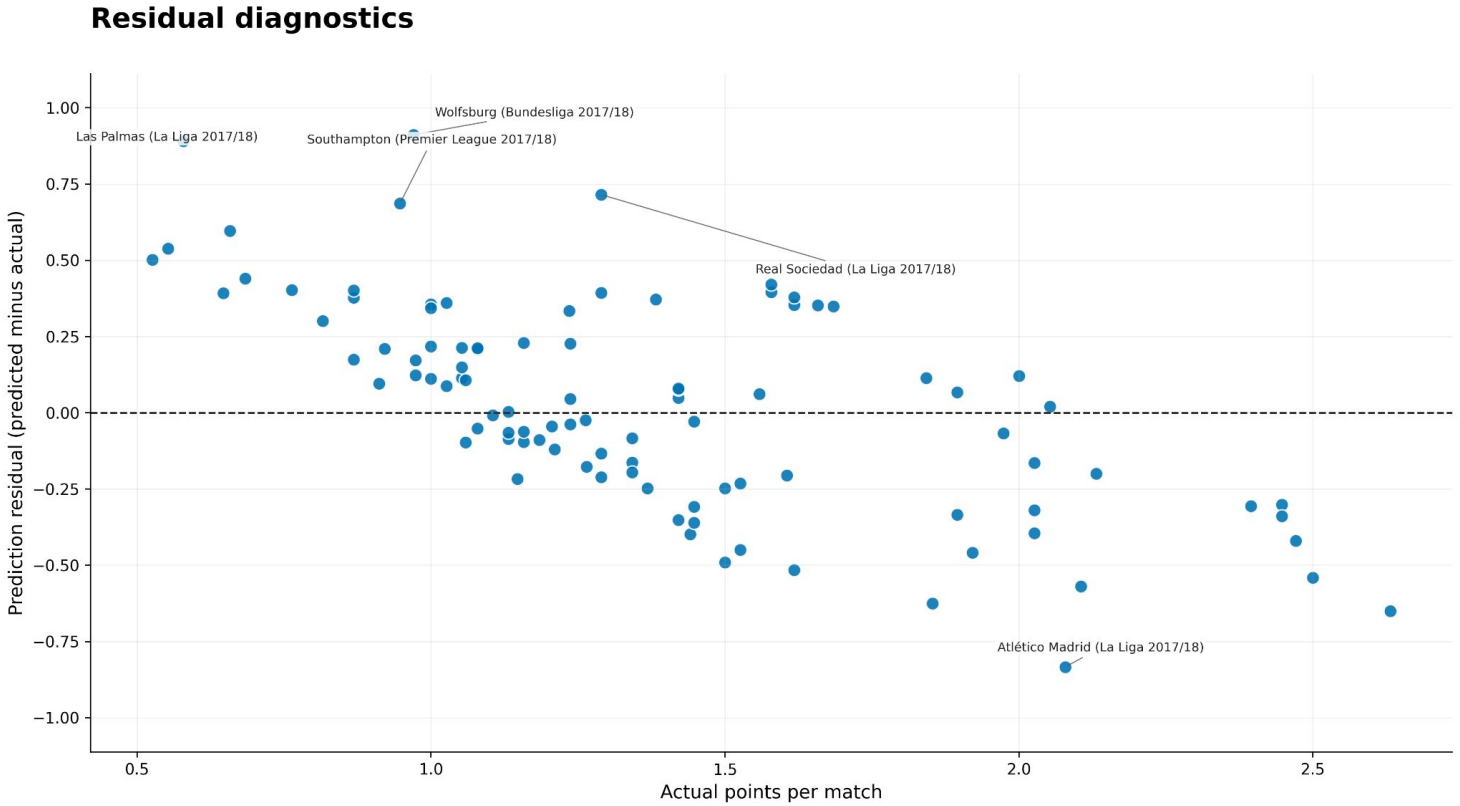
Pelle, Sebastian, Søren, Nicholas, Christian

Discussion, limitations, conclusion

- Main result: player archetypes can be learned from event data and used to predict team success.
- Best model beats baseline: MAE about **0.28** vs **0.37**, R^2 about **0.47**.
- Strength: interpretable ML pipeline from raw events to team-level predictions.
- Limitation: association, not causation; sample size is limited; no injuries, wages, coaching, schedule strength, or physical tracking data.
- Conclusion: ML gives a useful scouting/team-composition lens, but should support - not replace - football expertise.

Discussion, limitations, conclusion

- Main success.
- Best
- Strer
- Limit ıg, schedule stren
- Conc replace - footb



Appendix

A1 Full repository structure

- `01*`: feature construction
- `02*`: clustering and archetype freezing
- `03*`: team success modelling
- `04*`: squad templates / player matching
- `05*`: goalkeeper and formation extensions
- `15*–21*`: final plots and deliverables
- `99*`: reproduction and audits

A2 Full data lineage

Stage	What it contains	Example row / unit	Used for
Raw Wyscout events	Individual actions: passes, shots, duels, tackles, locations, minutes	One row = one event/action	Starting point
Match/team/player metadata	IDs, names, teams, competitions, minutes, role groups	One row = player/team/match metadata	Joining and filtering
Player-team-competition rows	Aggregated player statistics per 90 minutes	One row = one player snapshot	Clustering
Final labelled archetype table	Each player snapshot assigned to one of 18 archetypes	Player snapshot → archetype label	Interpreting player types
Team-composition table	Share of minutes played by each archetype for each team	One row = one team-season/competition	Supervised ML input
Team-success table	Points per match / team performance target	One row = team success outcome	Regression target

A3 Feature engineering details

- **Per-90 normalization:** event counts were converted into rates per 90 minutes, so players with different playing time could be compared fairly.
- **Rate features:** examples include passes per90, shots per90, tackles per90, interceptions per90, goals per90, and clearances per90.
- **Derived profile features:** extra features such as pass accuracy, duel win rate, goal conversion, shooting intensity, crossing intensity, and creative activity were constructed.
- **Missing values:** missing or undefined values were filled using simple imputation, e.g. median values or zeros depending on the feature type.
- **Minimum minutes filter:** very low-minute player snapshots were removed to avoid unstable statistics from tiny samples.
- **Outfield-only main story:** goalkeepers were excluded from the main archetype model because their event profiles are structurally different from outfield players.

A4 Leakage prevention

- No player names in ML features
- No team success variables in input features
- No physical metadata in action-only clustering
- Outcome columns removed before supervised modelling

A5 Clustering technical details

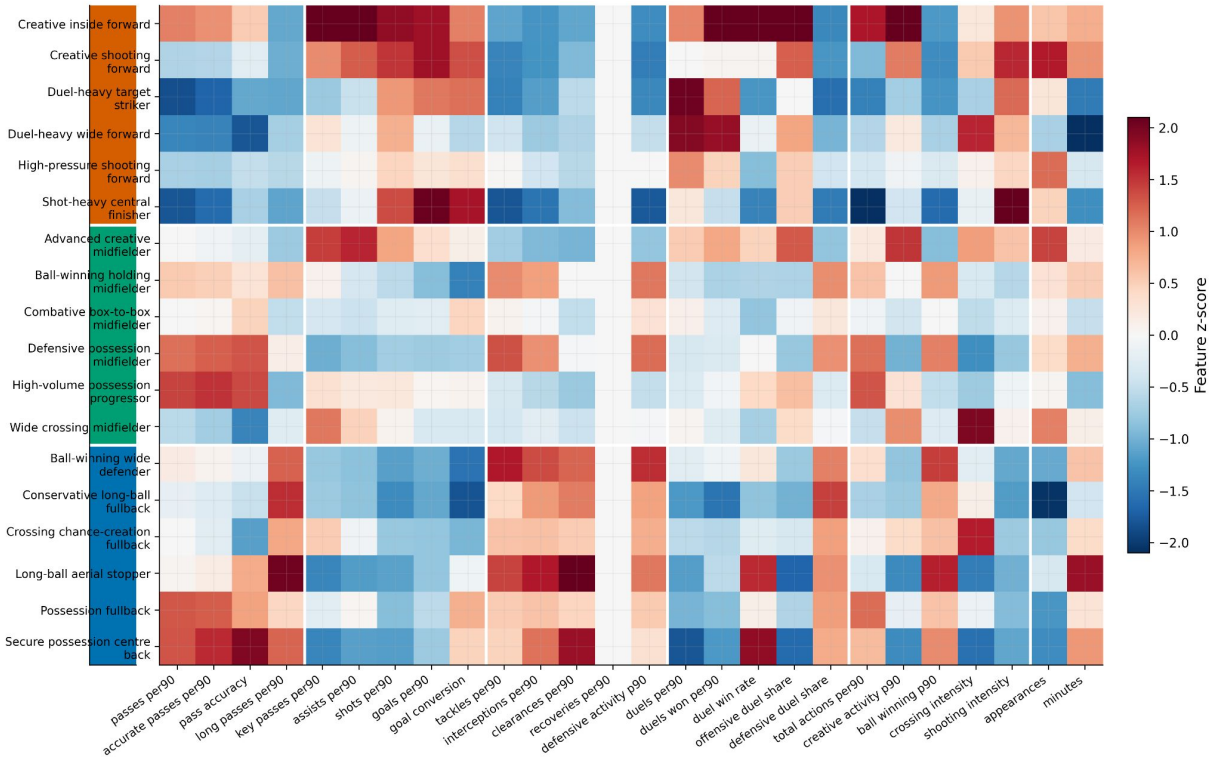
- StandardScaler before clustering
- KMeans for attackers/midfielders
- GMM for defenders
- Subclusters selected using silhouette score and minimum-size constraints
- Final result: 18 frozen archetypes

A6 PCA and dimensionality reduction

- PCA is unsupervised dimensionality reduction: high-dimensional data = fewer informative axes.
- We first standardized the player-feature matrix X so all features contributed on comparable scales.
- PCA computes the covariance matrix of X and diagonalizes it into eigenvectors and eigenvalues.
- Eigenvectors define new orthogonal feature directions; eigenvalues rank how much variance each direction explains.
- PC1 and PC2 were used to plot player snapshots and archetype centres in 2D.
- The purpose was model inspection: checking whether learned archetypes correspond to meaningful football roles.
- Caveat: PCA is linear and only visualizes part of the information; clustering used the full feature representation.

A7 Full archetype matrix

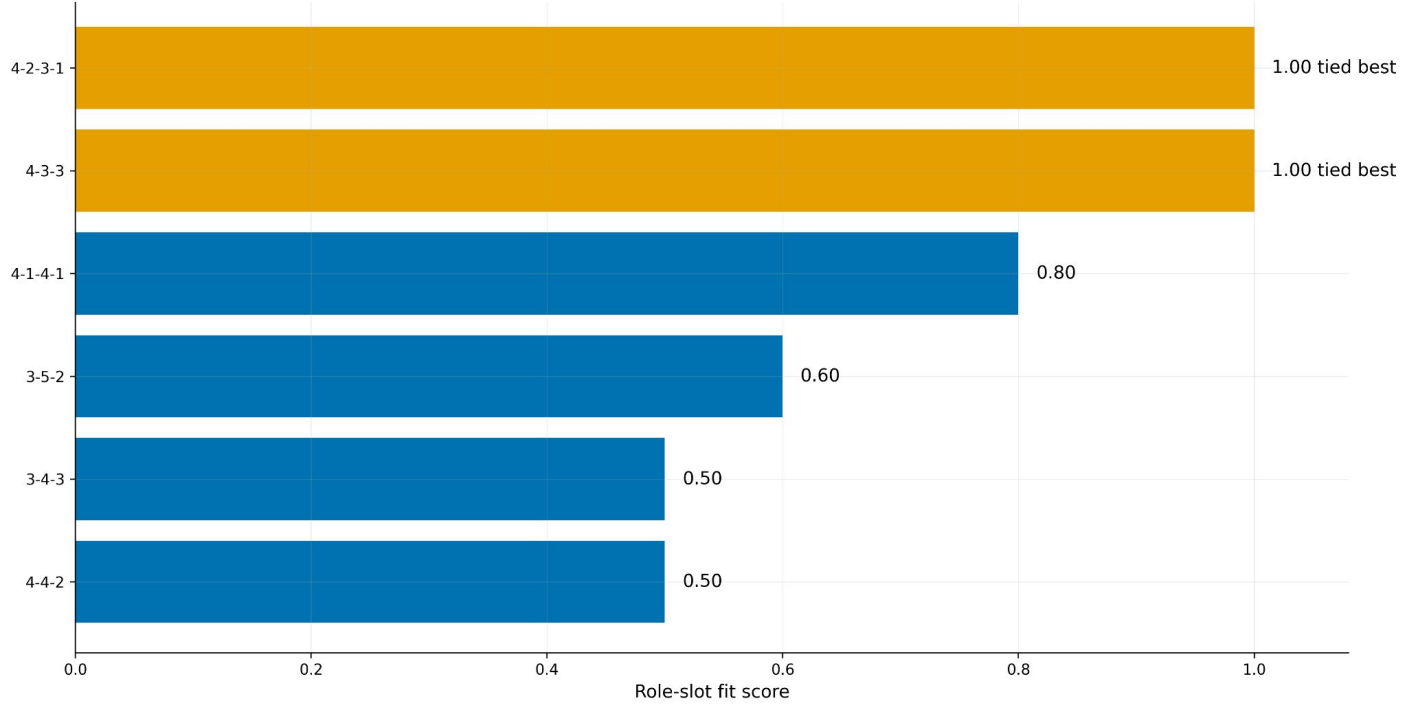
- Same as earlier plot, but the other was a simplified version because this was a bit dense for presentation.



Pelle, Sebastian, Søren, Nicholas, Christian

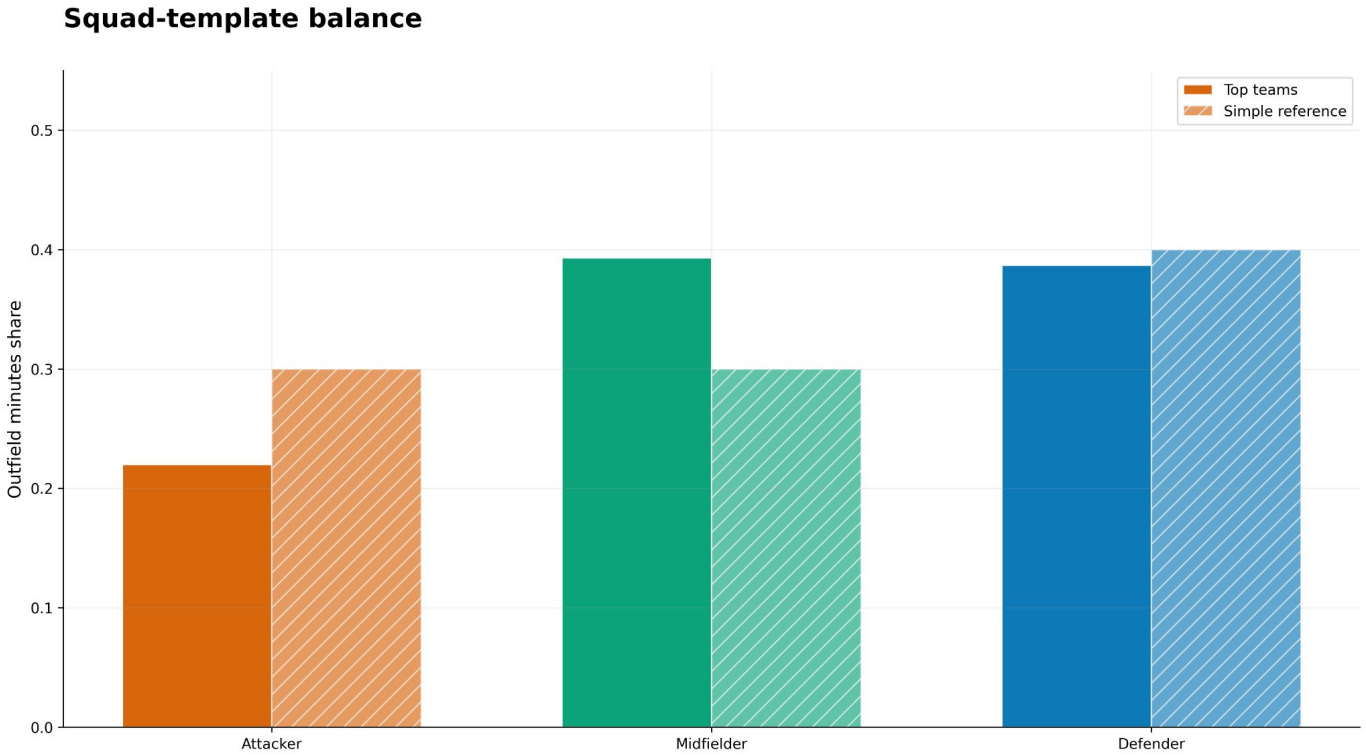
A8 Formation role-slot fit

- Here, a role-slot means a tactical position inside a formation, such as fullback, centre-back, defensive midfielder, winger or striker. The role-slot fit score measures how well our learned archetypes can fill the required roles in common formations. The 4-2-3-1 and 4-3-3 fit best, but this is a tactical diagnostic, not a main prediction result



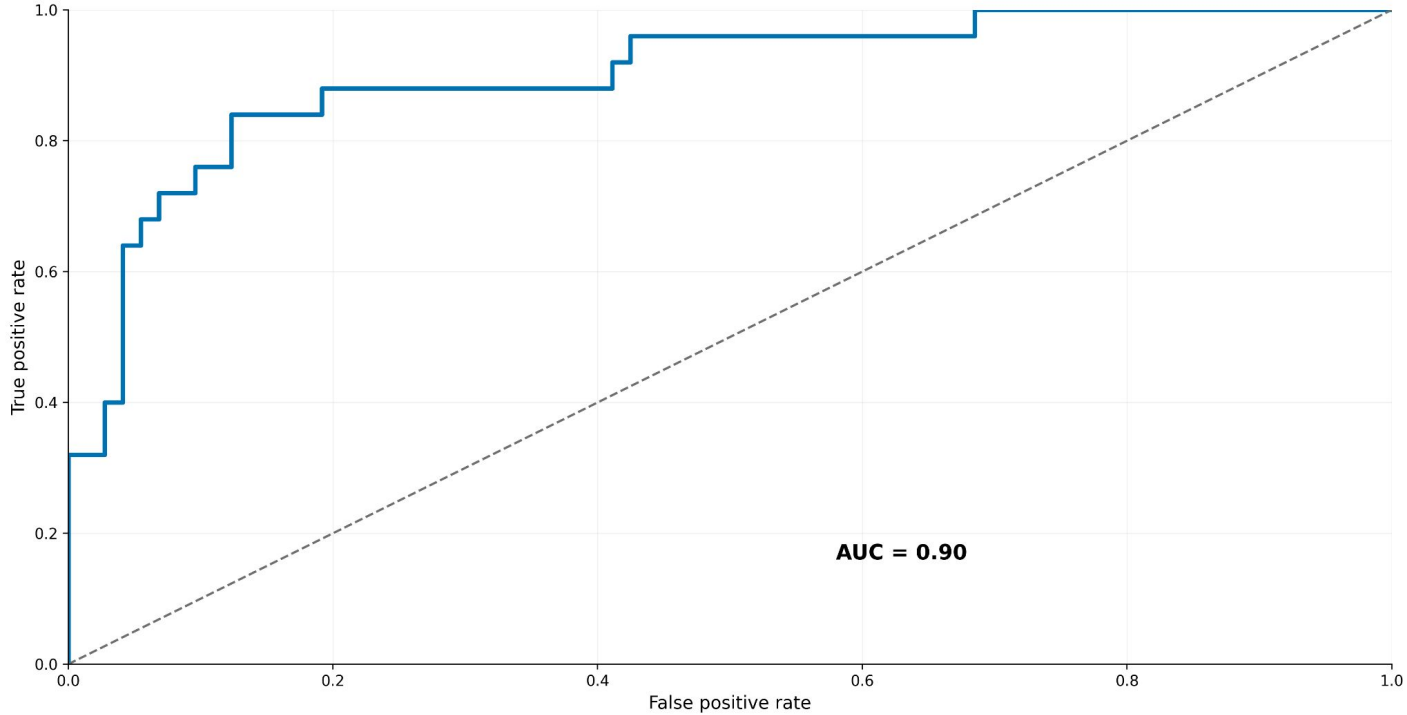
A9 Squad template balance

- Top teams compared with simple reference (the simple reference was just our assumption).



A10 Auxiliary ROC check

- AUC ≈ 0.90.
- But this is auxiliary because the main task is regression.
- It converts points per match into top-quartile classification.



Pelle, Sebastian, Søren, Nicholas, Christian

A11 The models/theory we used in short and evaluation metrics

- Ridge: linear regression with L2 penalty.
- Elastic Net: L1 + L2 regularization.
- Random Forest: bagging of decorrelated decision trees.
- Gradient Boosting: additive trees trained sequentially.
- Cross-validation estimates generalization error.

- MAE: average absolute prediction error in points per match.
- RMSE: punishes large errors more.
- R^2: fraction of variance explained relative to baseline.
- Permutation p-value: checks whether real target performs better than shuffled target.

A12 Selected hyperparameters/reproducibility

- Ridge alpha = 10.
- Elastic Net alpha = 0.02, l1_ratio = 0.35.
- Random Forest: 400 trees, min_samples_leaf = 3.
- Gradient Boosting: depth 2, learning rate 0.04, 150 estimators.
- Fixed random seed = 42 (of course!).

A13 Challenges

- Raw event data is messy and huge.
- Player roles are not fixed labels.
- Team success depends on many missing variables.
- Small supervised sample compared with number of features.
- Need to balance prediction and interpretability.

A14 If we had more time

- Add tracking data: speed, distance, pressing, spacing.
- Add multiple seasons.
- Add player market value / salary / age curves.
- Test causal questions with stronger design.
- Try graph neural networks for player networks.
- Use SHAP values for more robust interpretability.

A14.2 Extra project

- **Our first attempt.**
- **Superliga project using match data to evaluate player in roles on the pitch and comparing to a team's success.**

Data Used

- SuperLiga training data for 2021/2022, 2022/2023, 2023/2024, 2024/2025
  - Train matches 772
- Testing data for 2025/2026
  - Test matches 192
- Features

Feature example

- Underneath is an example of what the data looks like for a single match(fixture_id) for all the players participating in that match.

	fixture_id	season_id	starting_at	team	is_home	team_formation	player_id	player_name	formation_position	position_id	type_id	player_stat_passes	player_stat_accurate_passes	player_stat_long
0	19130290	23584	2024-09-01 16:00:00	Aalborg BK	0	4-3-3	3861877.0	Vincent Müller	1.0	24	11	37.0	36.0	
1	19130290	23584	2024-09-01 16:00:00	Aalborg BK	0	4-3-3	159505.0	Oumar Diakhite	4.0	25	11	64.0	60.0	
2	19130290	23584	2024-09-01 16:00:00	Aalborg BK	0	4-3-3	37657515.0	Sebastian Otoa	3.0	25	11	58.0	50.0	
3	19130290	23584	2024-09-01 16:00:00	Aalborg BK	0	4-3-3	6642507.0	Lars Kramer	NaN	25	12	NaN	NaN	
4	19130290	23584	2024-09-01 16:00:00	Aalborg BK	0	4-3-3	37565350.0	Mylian Jimenez	NaN	26	12	17.0	15.0	

Features

- 772 Matches:
  - 2 teams 3 outcomes: HomeWin, Draw and AwayWin
  - 2 formations: fx: 4-3-3, 4-4-2, 3-5-2……
    - Positions. fx: GK (Goalkeeper) with assigned number 1. ST: 9, RW: 11
  - Individual player statistics
    - Passes, Goals, duels, tackles, dripple attempts….
  - Weakness in data
    - Physical data is missing: sprints, total distance ran, max speed…etc

The Machine Learning in the data

- Using random forrest we average the last 5 games and predict a team succes given the formula:

$$Y^{\text{success}} = P + 0.35(P - b).$$

row	crosses_roll5	shots_roll5	Y^{success}
1	0.5	0.8	0.7
2	1.0	1.0	1.1
3	2.0	1.4	1.4
4	3.2	1.9	2.1
5	4.0	2.4	2.7
6	4.8	2.8	3.0
7	5.5	3.0	3.3
8	6.0	3.4	3.5

- Then it converts the features importance into contribution and then g_r becomes a percentile C_i,r and the same is done for purely whether a player looks like certain player role F_i,r and gives them a final score of S_i,r, which is between 0 and 1

role	$g_r(x_i)$
RW	3.10
LW	2.95
ST	2.20
CM	1.80
GK	0.40

$$C_{i,r} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}\left[g_r(x_j) \leq g_r(x_i)\right].$$

$$S_{i,r} = F_{i,r}^{0.5} \cdot C_{i,r}$$

Pelle, Sebastian, Søren, Nicholas, Christian

On all players in the superliga

Top CB candidates

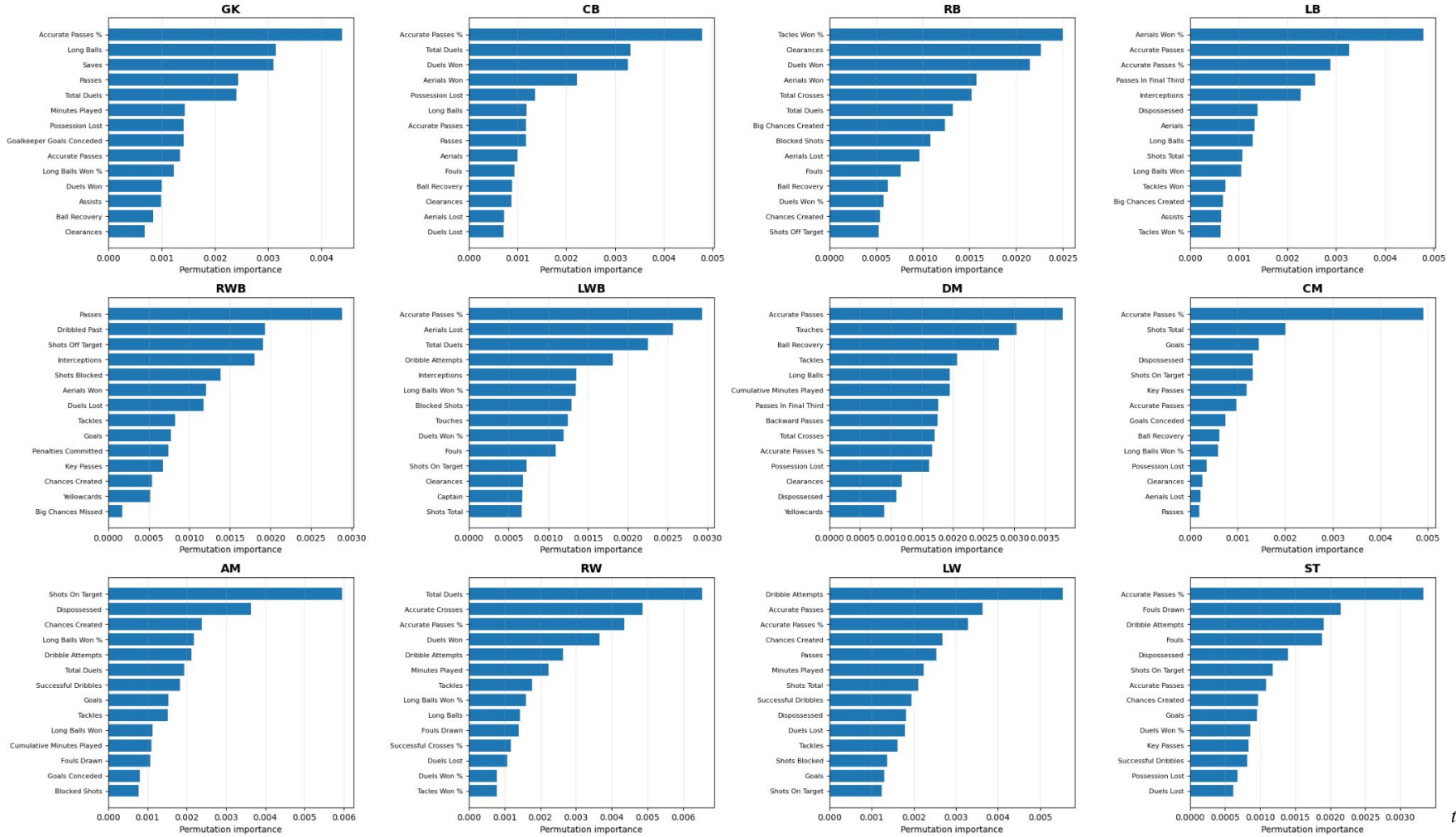
	player_name	team	ML_CB_score	ML_CB_contribution_pct	ML_CB_fit_pct	ML_CB_contribution_pred	best_learned_position	best_learned_score
864	Pantelis Hatzidiakos	FC København	0.956	0.968	0.977	1.792	CB	0.956
789	Gabriel Pereira Magalhães dos Santos	FC København	0.934	0.984	0.901	1.863	CB	0.934
1033	Peter Ankersen	Nordsjælland	0.903	0.992	0.828	1.960	CB	0.903
536	Pontus Rödin	Silkeborg IF	0.900	0.920	0.958	1.686	CB	0.900
961	Felix Beijmo	AGF	0.887	0.982	0.814	1.860	CB	0.887
952	Luis Binks	Brøndby IF	0.882	0.911	0.938	1.675	CB	0.882
719	Frederik Alves Ibsen	Brøndby IF	0.882	0.900	0.960	1.660	CB	0.882
1002	Caleb Marfo Yirenkyi	Nordsjælland	0.874	0.969	0.813	1.794	CB	0.874
937	Eric Kahl	AGF	0.868	0.951	0.832	1.745	CB	0.868
854	Mathias Jattah-Njie Jørgensen	FC København	0.863	0.888	0.945	1.645	CB	0.863
951	Tobias Salquist	Nordsjælland	0.860	0.901	0.912	1.661	CB	0.860
582	Marc Dal Hende	Sønderjyske Fodbold	0.846	0.921	0.843	1.688	CB	0.846

Top RWB candidates

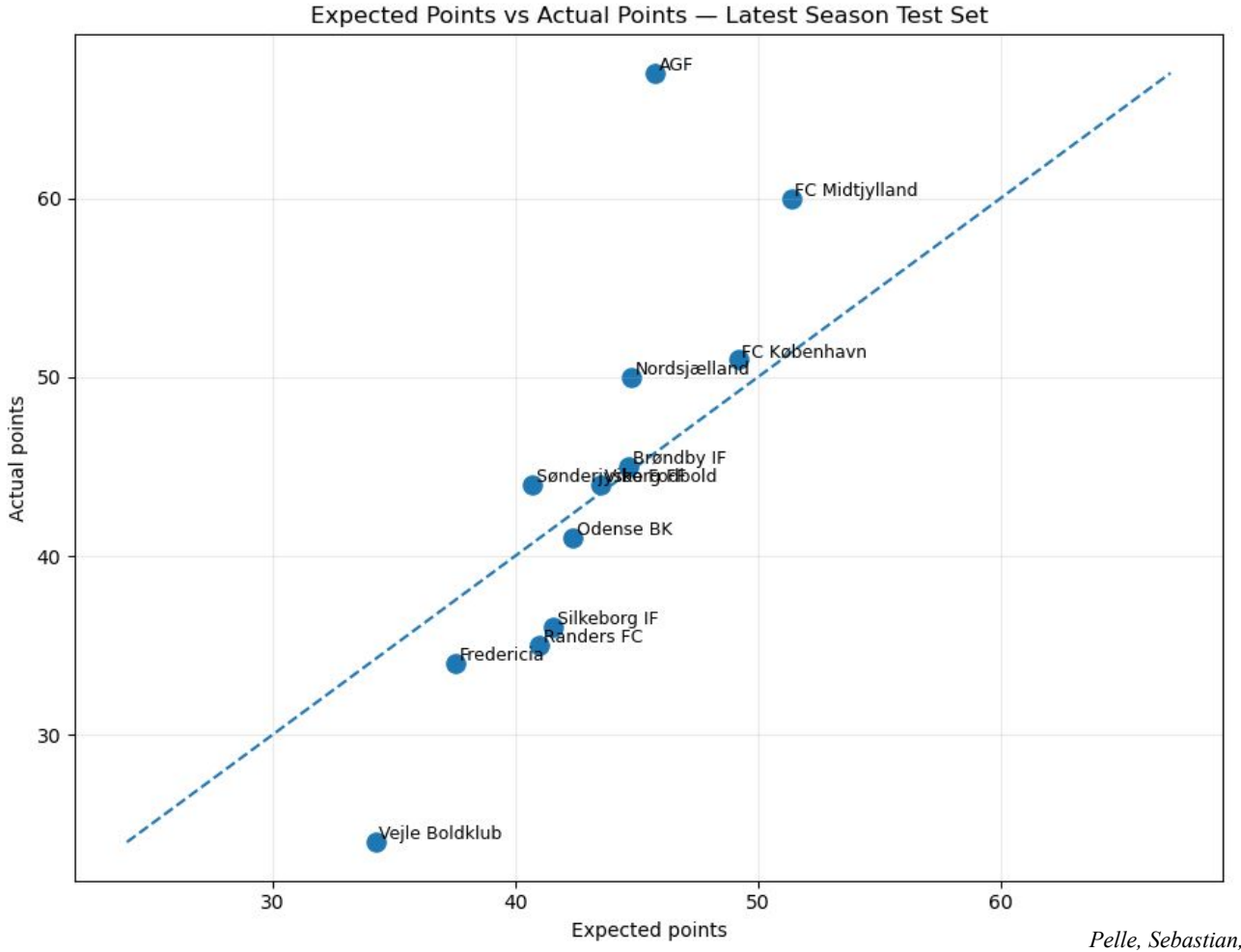
	player_name	team	ML_RWB_score	ML_RWB_contribution_pct	ML_RWB_fit_pct	ML_RWB_contribution_pred	best_learned_position	best_learned_score
964	Gift Links	AGF	0.909	0.950	0.917	2.162	RWB	0.909
1028	Juho Lähteenmäki	Nordsjælland	0.864	0.877	0.970	1.926	RWB	0.864
954	Marko Divković	Brøndby IF	0.837	0.940	0.792	2.111	RWB	0.837
785	Mads Enggård	Vejle Boldklub	0.809	0.860	0.884	1.882	RWB	0.809
855	Nikolas Langberg Dyhr	Randers FC	0.777	0.791	0.965	1.785	RWB	0.777
465	Kolbeinn Birgir Finnsson	Lyngby Boldklub	0.746	0.761	0.962	1.719	RWB	0.746
620	Sebastian Søraas Sebulonsen	Brøndby IF	0.745	0.748	0.994	1.699	RWB	0.745
517	Oliver Sonne	Silkeborg IF	0.741	0.893	0.688	1.979	RWB	0.741
1019	Dario Esteban Osorio Osorio	FC Midtjylland	0.737	0.875	0.710	1.926	LW	0.792
558	Giorgi Gocholeishvili	FC København	0.736	0.751	0.958	1.707	RB	0.809
658	Clement Bischoff	Brøndby IF	0.734	0.903	0.661	1.996	LW	0.770
572	Leon Klassen	Lyngby Boldklub	0.731	0.787	0.863	1.780	RWB	0.731

Pelle, Sebastian, Søren, Nicholas, Christian

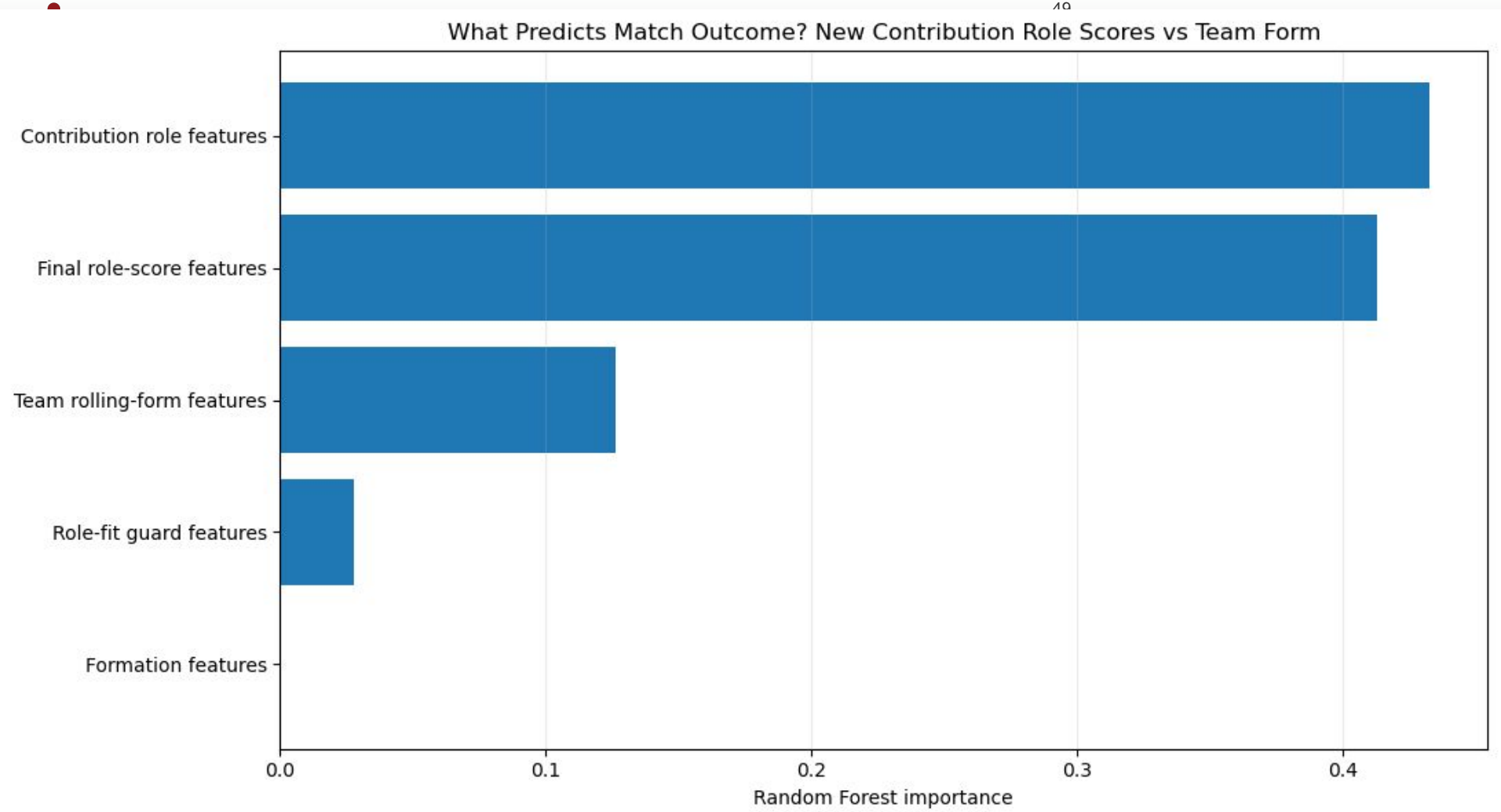
Outcome-Based Raw Player Feature Importances by Pitch Position



tian



Pelle, Sebastian, Søren, Nicholas, Christian


What Predicts Match Outcome? New Contribution Role Scores vs Team Form
Contribution role features
Final role-score features
Team rolling-form features
Role-fit guard features
Formation features
0.0
0.1
0.2
0.3
0.4
Random Forest importance

The dream team from the Superliga

	slot_index	assigned_role	player_name	team	role_score	best_learned_position	best_learned_score	ML_GK_score	ML_CB_score	ML_RB_score	ML_LB_score	ML_RWB_score	ML_LWB_sco
0	0	GK	Diant Ramaj	FC København	0.980	GK	0.980	0.980	0.071	0.027	0.050	0.026	0.0
1	1	RB	Rodrigo Huescas	FC København	0.955	RB	0.955	0.694	0.549	0.955	0.545	0.512	0.5
2	2	CB	Pantelis Hatzidiakos	FC København	0.956	CB	0.956	0.493	0.956	0.124	0.429	0.134	0.3
3	3	CB	Gabriel Pereira Magalhães dos Santos	FC København	0.934	CB	0.934	0.664	0.934	0.435	0.841	0.180	0.2
4	4	LB	Christian Nikolaj Sørensen	Vejle Boldklub	0.907	LB	0.907	0.765	0.281	0.582	0.907	0.514	0.2
5	5	RW	Jordan Larsson	FC København	0.977	RW	0.977	0.650	0.231	0.283	0.302	0.682	0.5
6	6	CM	Jeppe Tverskov	Nordsjælland	0.953	CM	0.953	0.669	0.633	0.463	0.519	0.331	0.3
7	7	CM	Kristoffer Olsson	FC Midtjylland	0.951	CM	0.951	0.719	0.536	0.210	0.625	0.612	0.6
8	8	LW	Ernest Appiah Nuamah	Nordsjælland	0.920	LW	0.920	0.771	0.095	0.176	0.071	0.243	0.3
9	9	AM	Mikkel Duelund	Vejle Boldklub	0.949	AM	0.949	0.581	0.227	0.190	0.207	0.332	0.3
10	10	ST	Alan Goncalves Sousa	Aalborg BK	0.944	ST	0.944	0.780	0.106	0.389	0.231	0.485	0.4

Pelle, Sebastian, Søren, Nicholas, Christian

A15 Optimisation

- **I was a bit unsure whether our appendix is graded on future or already done optimisation.**

So I will also give you the latter:

Feature filtering and cleaning

- Removed non-informative or leakage-prone columns such as player names, team names, labels, and target-related variables.
- Converted raw event counts into more comparable per-90 and rate features.

Standardisation before clustering

- Features were scaled before clustering so high-count variables like passes did not dominate smaller-scale variables like rates.

Role-aware clustering

- Players were separated into broad role groups before final archetype construction.
- This avoids clustering strikers directly against centre-backs when their statistical tasks are fundamentally different.

Choice of number of archetypes

- Tested different cluster configurations and selected a final interpretable set of **18 frozen archetypes**.
- The final number was chosen as a balance between detail and interpretability.

Model comparison

- Compared simple baseline, Ridge, Elastic Net, Random Forest, and Gradient Boosting.
- This makes it possible to see whether complexity actually improves performance.

Regularisation

- Ridge and Elastic Net were used to reduce overfitting with correlated team-composition features.

Cross-validation strategy

- Leave-One-Competition-Out validation tests whether the model generalizes across competitions, not just random splits.

Random Forest settings

- Used many trees and restrictions such as minimum leaf size to reduce variance and avoid overfitting.

Permutation/null validation

- Compared the real model to models trained on shuffled targets.
- This checks whether the performance is better than what could be achieved by random structure.

A16 Project statement: Contribution balance for this project

- **We all contributed equally.**

Pelle, Sebastian, Søren, Nicholas, Christian