SOUTHPORT FC – RECRUITMENT ASSIGNMENT

Francesco Trasforini, 2024-11-18

Introduction

The assignment requires creating a shortlist of 5 players based on 3-5 performance metrics. The database provided is an Excel file containing Wyscout data from the current season. The database contains players who could broadly fit the profile of a Right Wing Back (all players available have played some minutes on the right wing, either as a fullback, wingback, right winger or right attacking midfielder).

The shortlisted players should then be compared to Southport FC player S.Minihan on the selected metrics through charts, tables or visualizations.

Methodology

I chose to develop a web app to complete the assignment. The reasons are:

- A web app doesn't just offer a solution to the task at hand but can be reused and expanded to answer future research questions.
- Web apps offer easy interaction with the data, allowing less data-savvy people at the club to use the tool.
- Web apps can be accessed anytime from anywhere, eliminating the need for a local file copy.

Data Preparation

Before starting the development, I analysed and cleaned the data using a Jupiter notebook. I applied some data conversions, in particular:

- 'Contract Expires' was set to a date with format 'yyyy-mm-dd'.
- Columns of type 'object' that were better suited to type 'string' were changed accordingly (e.g. 'Player', 'Team'). The same applied to column of type 'object' which could contain a list of 'strings' (e.g. 'Passport country').
- After inspection, columns containing numeric values were changed to smaller data type (e.g. from int64 to int16) to save space in the database, since the numbers in these columns are going to be small by design. The only exception is market value, for which I used an int32 to allow for larger values.

I also ran a check for empty columns, and I found that there was an extra column 'Aerial duels per 90.1' that had only empty values. The database already has another column called 'Aerial duels per 90' which is for the most part correctly populated.

Other empty columns are 'Conceded goals per 90', 'Shots against per 90', 'Prevented goals', 'Prevented goals per 90', 'Exits per 90', but those are empty because they relate to goalkeeping attributes.

Since the database has no goalkeepers, I could have dropped these columns, but I chose to keep them. The reason is the following: my idea is to build a database that we can populate with all type of players, including goalkeepers. Then we can use the database and the connected app to scout players in any role.

Thus, the only column I dropped was 'Aerial duels per 90.1', which most likely was there either by error or to test if I will find it.

For issues related with missing data, I chose to keep the NaN values in the database as they were. Dropping the rows containing some NaN values didn't seem a suitable strategy, given the amount of missing data. Other strategies could have been to replace NaN with 0 or with the mean column value, but both these strategies introduce some bias in the data. Instead, keeping NaN values allowed me to retain the "real" information (in the end, a missing data point is a missing data point, not a 0 or an average). The way I handled the NaN was to simply exclude those data points from calculations when they occurred, as they truly are 'unknown'.

Note: on frontend, I choose to print "unavailable" instead of NaN for clarity (especially considering web app users may not know the meaning of NaN).

Once ready, the database was then deployed in sql (PostgreSQL).

Web app development

The Web app is made of 3 components:

- The backend, developed in Python
- The frontend, which combines HTML and Javascript
- The SQL database

The Web app was deployed through Railway and is available at the following link: Football Player Recruitment.

Functionalities

The Web App allows to filter player data by role. For now, there are only 3 roles available:

- Right Wing Back
- Center Forward
- Goalkeeper

Then the user can filter players using the 'Age' filter. This filter allows to select an option among 'Any', 'Under', 'Over', or 'Equal', before inserting an age between 18 and 40.

Another implemented filter is 'Minutes Played' which has options from 0 to 1500, spaced by 100 minutes each.

More filters could be implemented in the future, but for this task I opted to stick with those fundamental 2.

For each player in the retrieved table, there is a button 'Create Profile'. By clicking on it, the user will land on another html page, on which he can select another player to compare with the selected player, using a dropdown menu (default option: 'S. Minihan'). Clicking on 'Compare Players', will generate 2 percentile pizza plots, one per player, allowing easy comparisons on the 5 selected metrics (more on the selected metrics later).

Each value in the pizza plot is computed as the percentile score of the selected player in any given metric. Players who have NaN values in a metric are excluded from the computation for that given metric.

The user can then go back to the homepage by clicking on the 'Homepage' button.

In the Homepage, the user has the option 'Create Shortlist'. The shortlist will be generated on the filtered dataset. The size of the dataset will be displayed, to give context to the user (being shortlisted from a big sample is obviously more significant than being shortlisted from a small sample).

After creating the shortlist, it is then possible to click on 'Create Bar Chart', to generate a bar chart to compare the shortlisted players among the 5 selected metrics. It is also possible to add another player to the bar chart (by default, 'S. Minihan').

Shortlist logic

The shortlist will be generated on the filtered dataset in the following way:

- For each player, compute the percentile score in each of the 5 selected. Being a percentile, that score will be a number in the range 1-100.
- Each player will then have 5 scores, one per metric
- The scores are added to create a 'Total Score'
- The 5 players with the highest score are returned and create the shortlist. In case of ties for 5th place, all the players with the 5th highest score will be returned.

This approach works because:

- it creates a score that is relative to the other players in the dataset.
- Every score is going to be normalized as an integer from 1-100, allowing to get a significant score for metrics with different ranges.
- It offers an easy interpretation.

Limitation:

- This method does not indicate how much better a player is in a specific metric compared to others. For instance, a player who scores 100 in a metric might only be slightly better than their peers, or they might be significantly better, such as twice as good as the second-best player. This scoring system does not account for this variation.

However, this limitation is unlikely to happen in our context, as it is unlikely for players to be so dominant on the competition.

Selected metrics

Given that the task given is to shortlist Wing Backs, I selected the following 5 metrics as the most suitable:

- Successful Defensive Actions per 90
- PAdj Interceptions
- Progressive Runs per 90
- Accurate Crosses, %
- Progressive Passes per 90

While these metrics don't cover the full range of abilities required for a Wing Back, they cover the fundamentals to play the role: *Successful Defensive Actions per 90* and *PAdj Interceptions* control the defensive intensity and anticipation, *Progressive Runs per 90, Accurate Crosses,* % and *Progressive Passes per 90* control the level of creativity, bravery, accuracy, and attacking contribution necessary to be effective in the role.

Shortlisted players

Most teams in the Vanarama National North/South have played 15 to 17 matches. So, each player had the potential to play between 1350 and 1530 minutes.

Given that information, I chose to set the 'Minimum Played Minutes' filter to 700, to retain only players who have been involved roughly in at least half of the available playing time. Moreover, using 700 minutes as the minimum playing time filters out possible statistical anomalies.

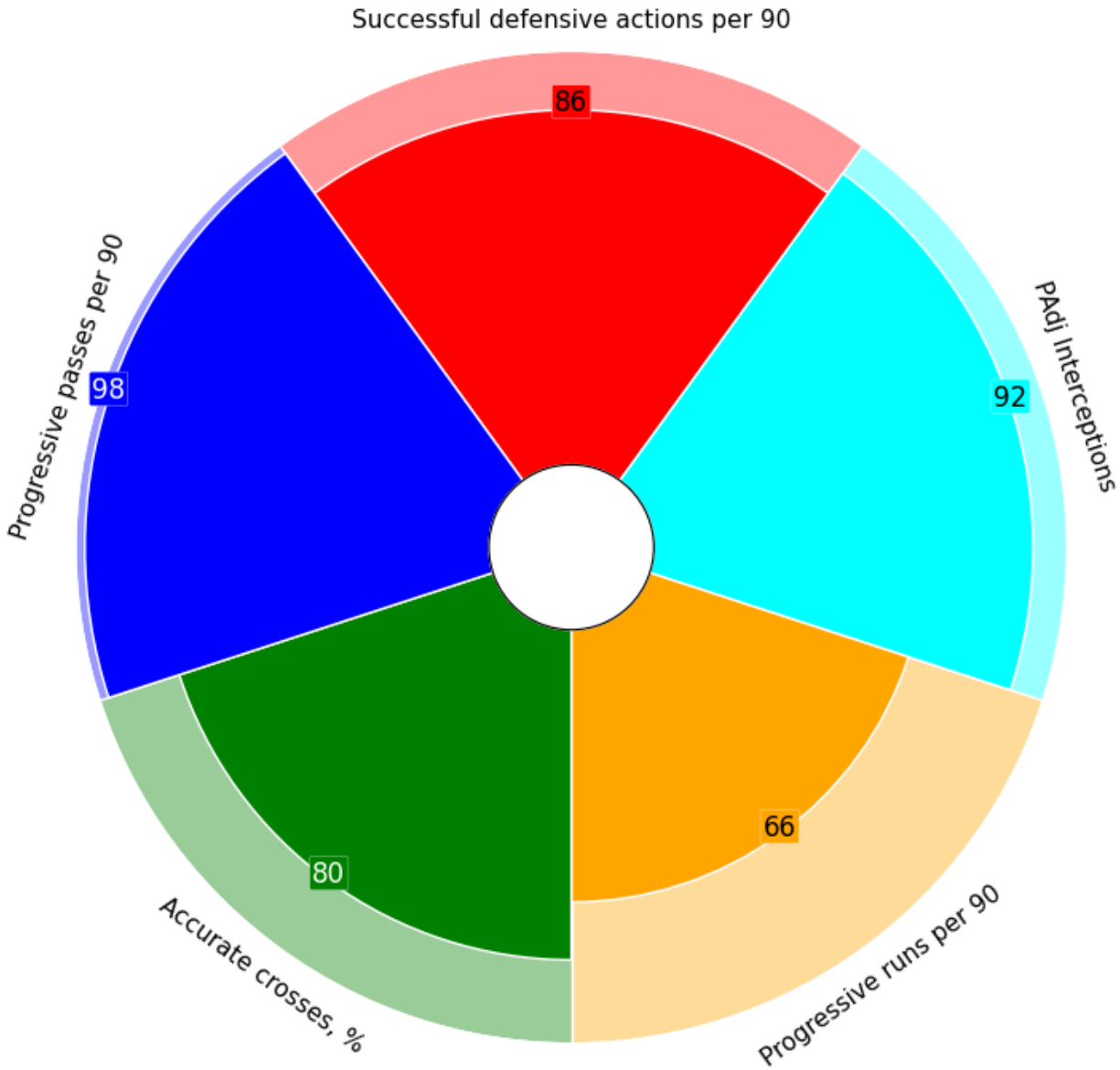
Filtering for 'Any' age and 700 minutes played would return a dataset of 136 players. I played around with the Age filter to see how many players would be left if I lowered the age to different values. In the end, I opted to filter for Under 30 players, obtaining a sample of 120 players. One reason I set the age filter relatively high is that most of the players shortlisted were already under 25. Thus, this choice allows for the score to be relative to a larger sample (more competition), showing that the selected players aren't just strong in their age group.

The **5 shortlisted players** are:

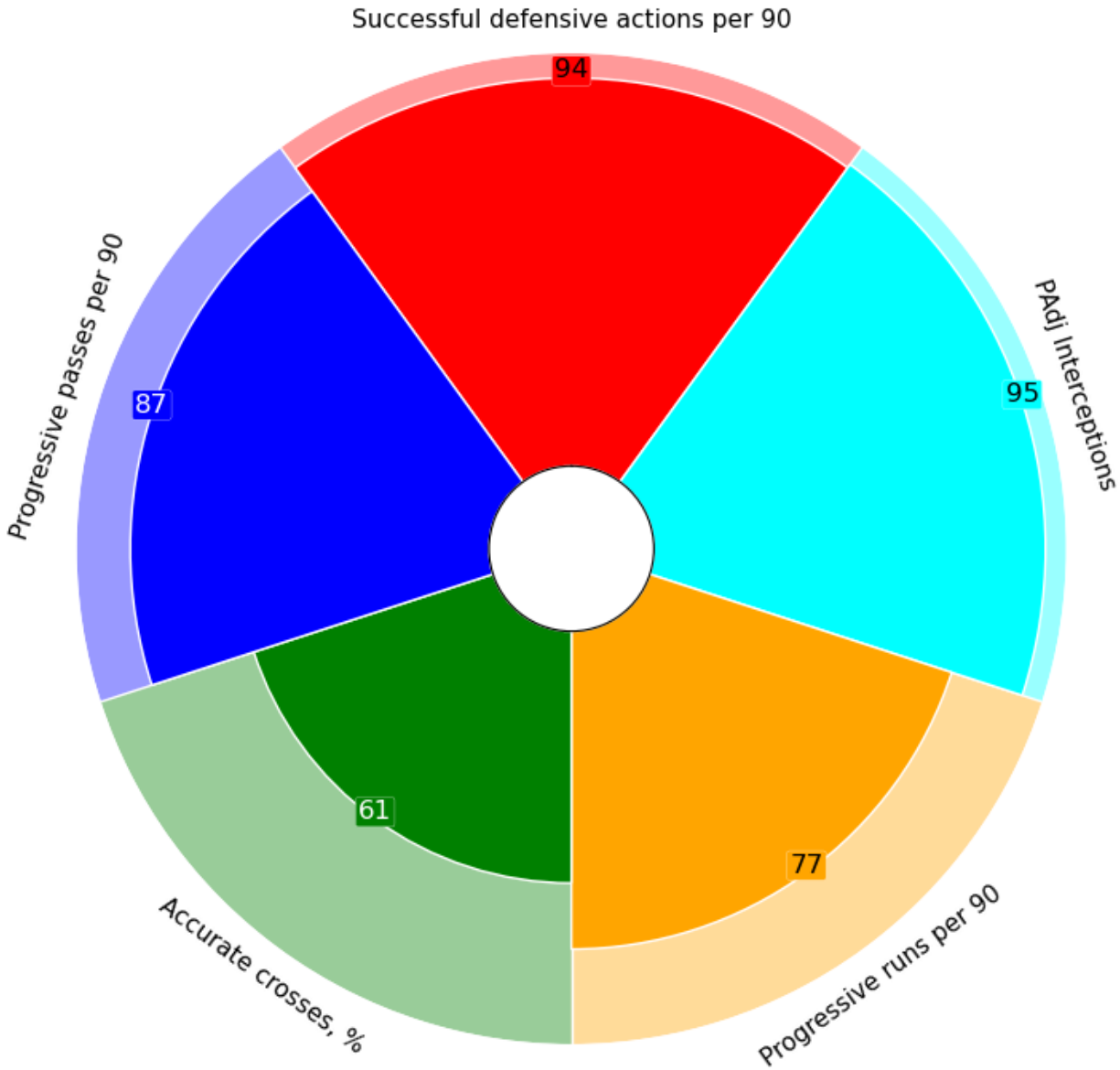
1. **D. Quick** (22) from Eastbourne Borough, who scored 431 points,
2. **L. Beeden** (24) from Slough Town, who scored 422 points,
3. **M. Parcell** (28) from Enfield Town, who scored 417 points,
4. **S. Robinson** (22) from Hereford FC, who scored 411 points,
5. **J. Hunter** (24) from Chester, who scored 408 points.

Note: while the scores are computed only against the players in the filtered dataset, the pizza plots show the percentile scores against the entire dataset. This was a design choice to allow a straightforward comparison among the charts.

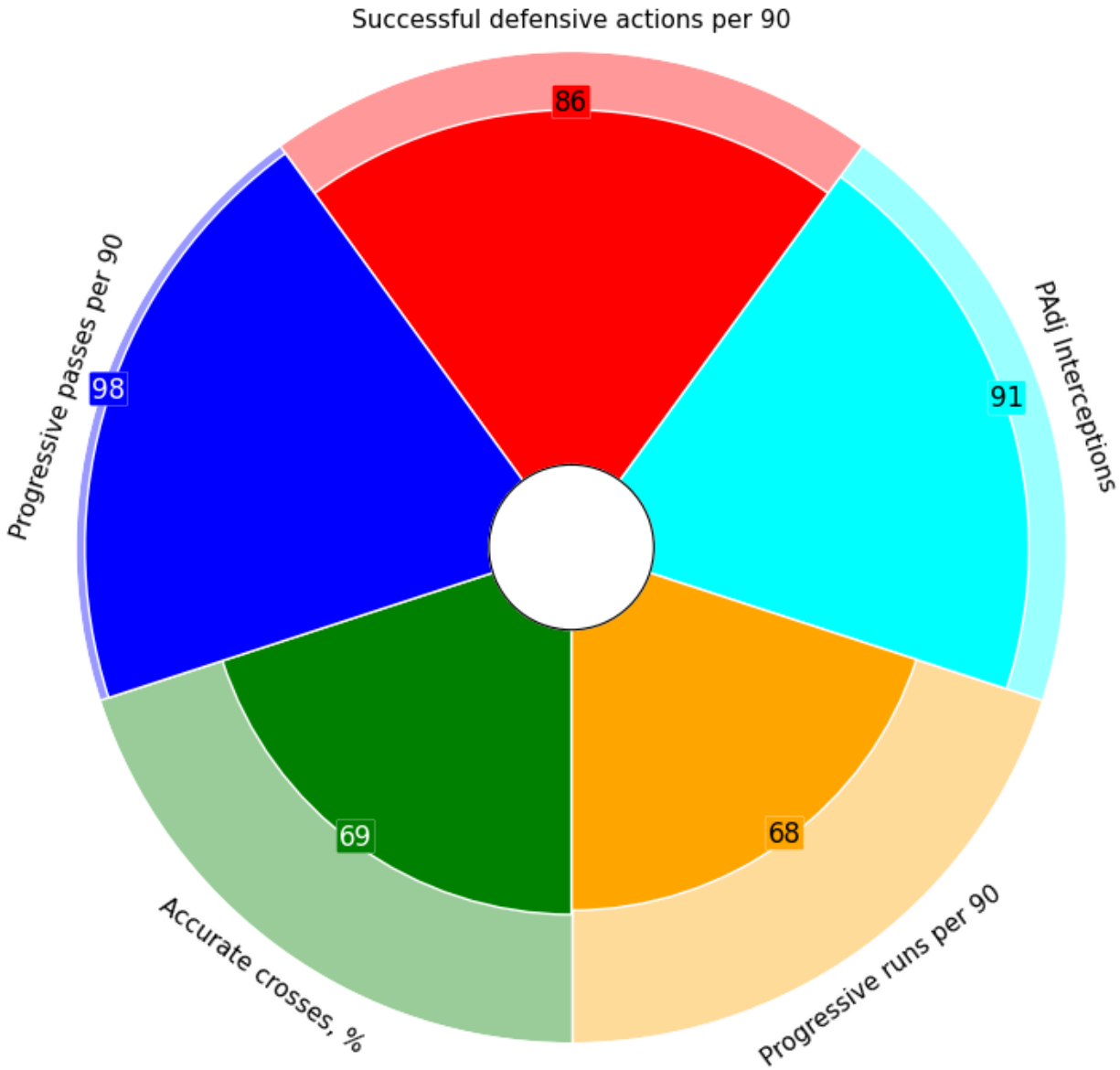
D. Quick pizza plot



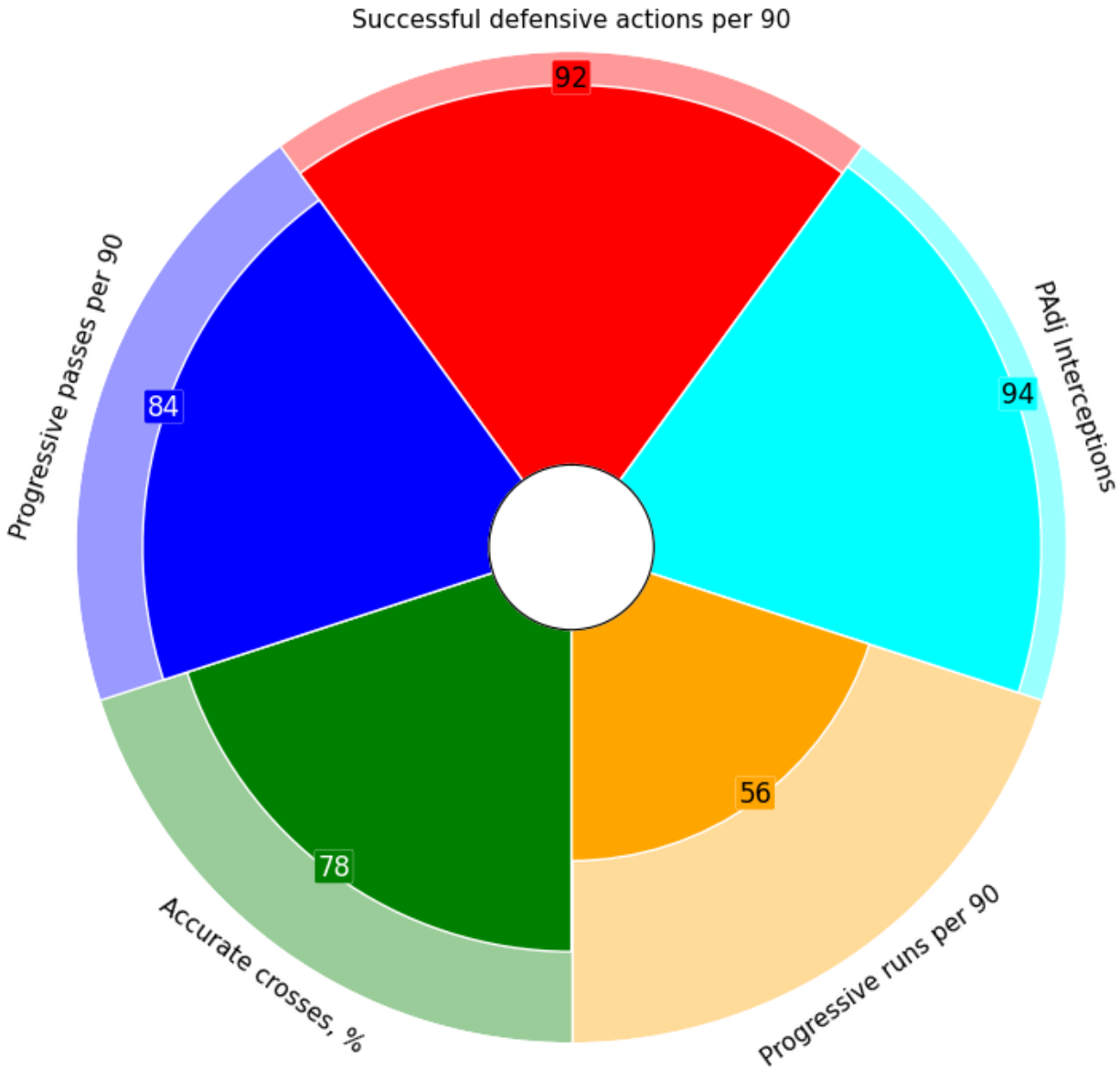
L. Beeden pizza plot



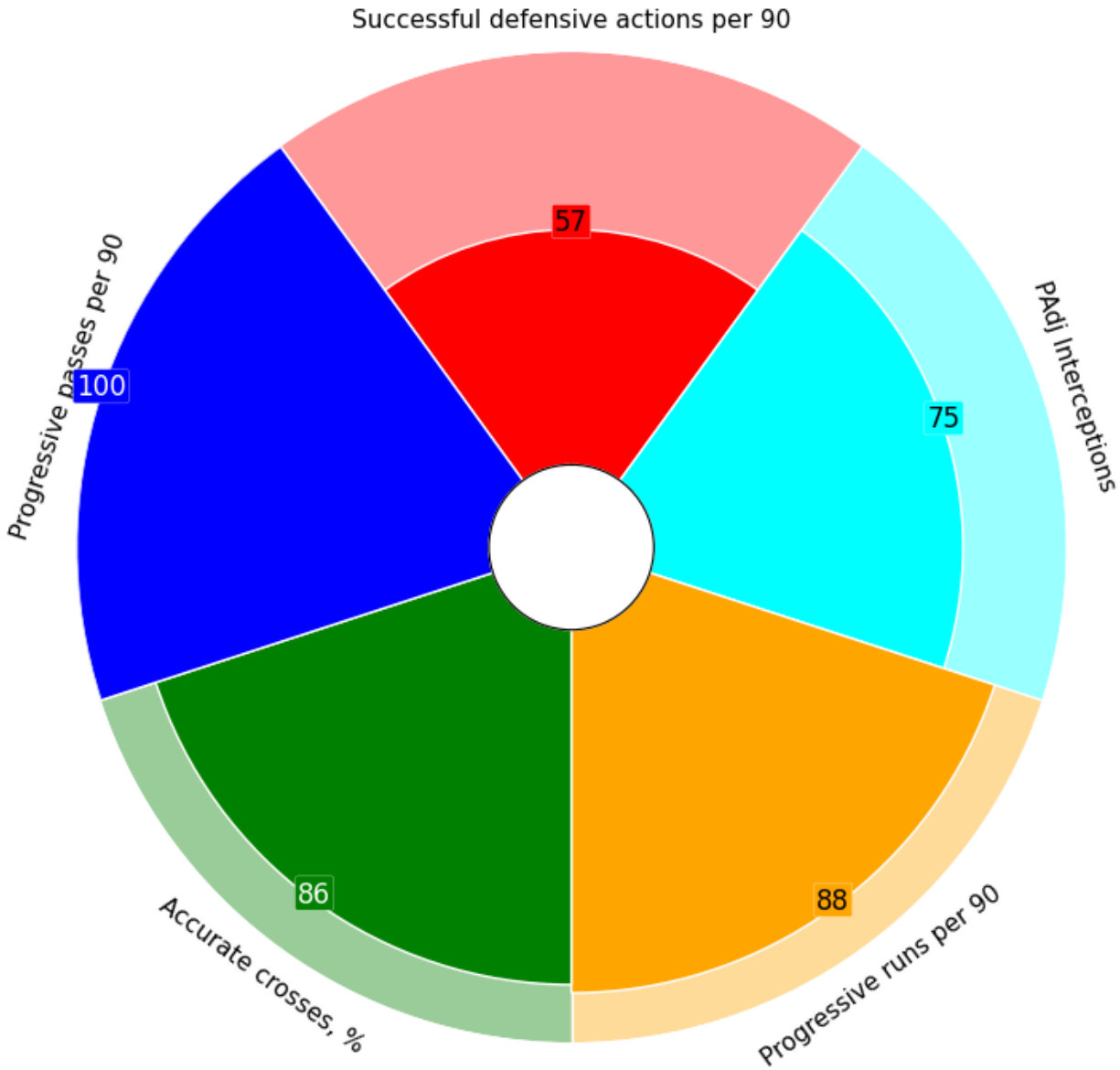
M. Parcell pizza plot



S. Robinson pizza plot

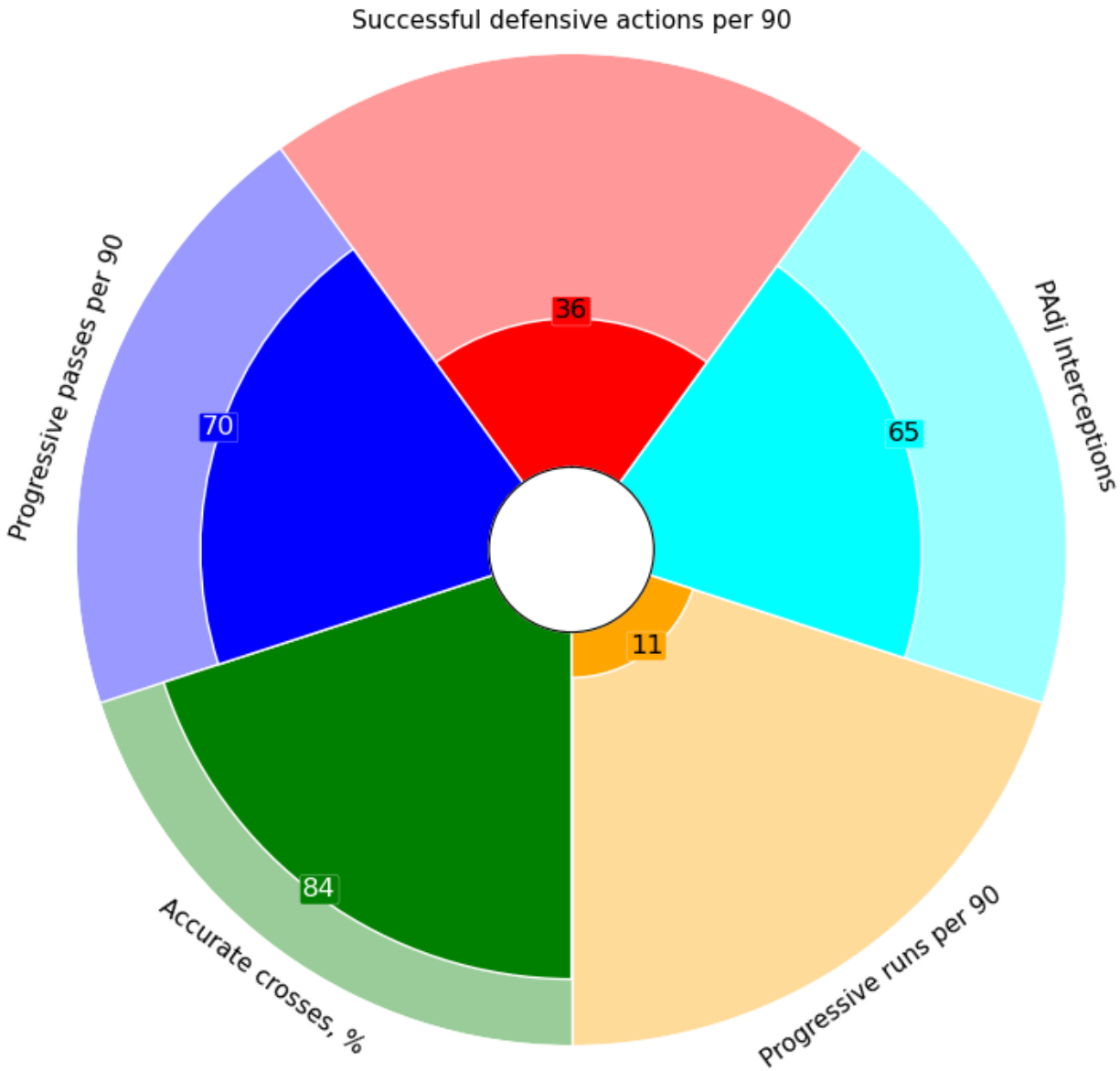


J. Hunter pizza plot

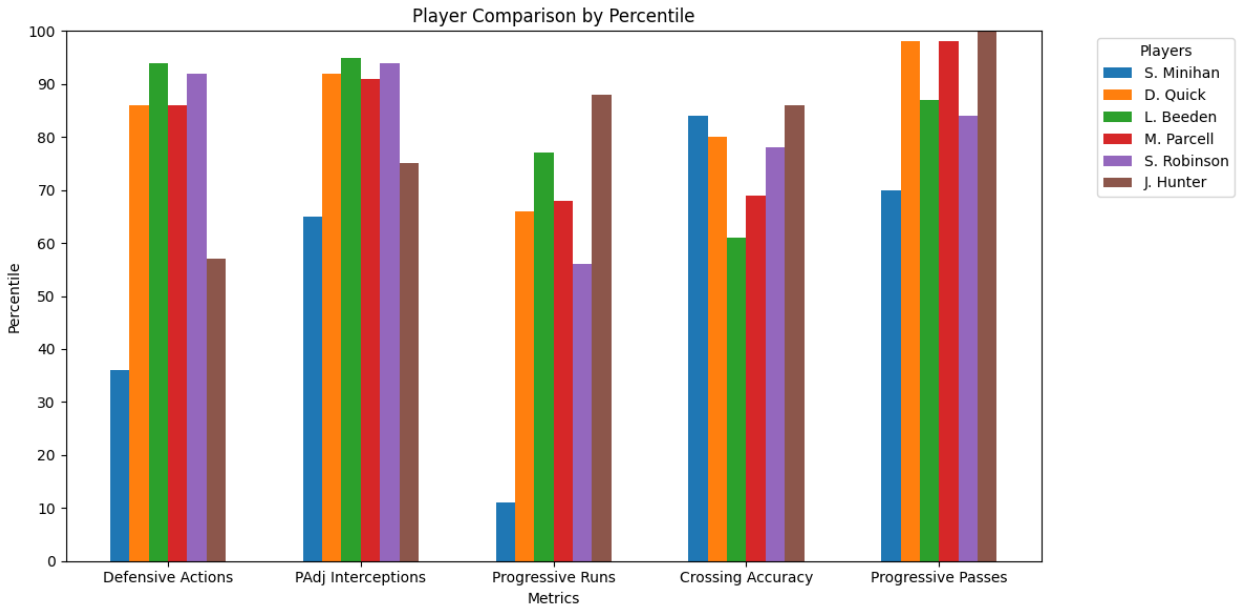


S. Minihan pizza plot

Here is S. Minihan pizza plot, which constitutes his “Player Profile” on the selected metrics.



Comparison between the shortlisted players and S. Minihan



I chose to use a bar chart for player comparisons because it clearly illustrates the scores and highlights which attributes each player excels in relative to the others.

From this plot, we can immediately infer that the selected players could be strong recruitment targets, since they're outperforming S. Minihan in most of the considered metrics. In fact, except for Crossing Accuracy, S. Minihan is the worst performer in every considered metrics.

Especially in Defensive Actions and Progressive Runs, S. Minihan output is worrying.

The numbers suggest that J. Hunter is a very talented attacking Wing Back, and he is still above average defensively.

S. Robinson has a more defensive minded profile but is also good going forward. His activity defensively seems to be quite high.

M. Parcell is elite defensively and for Progressive Passes.

L. Beeden has the strongest defensive game among the selected players, but he is also second for Progressive Runs. However, his crossing is the worst, even though still easily above the median.

D. Quick has the highest score of all but isn't the best in any category. His score for Progressive Passes should be highlighted, as well as the score in Possession Adjusted Interceptions.

It must be noted that S. Robinson, M. Parcell, L. Beeden and D. Quick have all recorded some minutes at RCB, which may have boosted their defensive metrics. However, the information on how many minutes they played in each role isn't available.

Conclusion

The used methodology offers an easy interpretation, at the same time the scoring system used is meaningful and able to rank player performances on the chosen metrics. The metrics selected capture the essential traits of the role.

The developed Web App is easy to use and can be easily accessed to anyone at the club anywhere and anytime, making research faster and easier. Moreover, it is a tool which can be expanded with new functionalities and become a general scouting tool.

The plots used are easy to interpret and allow for straightforward comparisons among players.

The 5 shortlisted players, Quick, Beeden, Parcell, Robinson and Hunter could all be interesting transfer targets for Southport FC.