Indiana Hoosiers Consulting

Recommendation for Rarita's FSA League National Team

Henry Bobeck
Michael Dineen
Eric Herbst
Hariharan PV
Annie Renholzberger

Faculty Advisor
Russell Lyons

SOCIETY OF ACTUARIES

Table of Contents

Objectives and Executive Summary

Indiana Hoosiers Consulting has been tasked with selecting a new national team for the FSA League. This team should have a high probability of being ranked within the top 10 by the 2026-2027 season and winning the championship by the 2031-2032 season. We will first outline the steps taken to construct this team and then analyze how this will impact the economy and Rarita's brand.

This report details how Hammessi Bayes can leverage historical overall ratings from the FIFA videogame franchise to quantify the holistic skill of players based solely on their statistics. We have provided two potential rosters based on the commissioner's willingness and ability to secure additional funding for the team. The first roster assumes no additional funding is sought, and the second assumes unlimited additional funding. The model has also been included with this report to find a "goldilocks" solution by simply changing one assumption in the optimization model.

We have found that winning the FSA League will boost tourism revenues by approximately 1% the following year. More substantially, people's image of Rarita will improve because of football success. The attempt to quantify this shift in attitude is futile but leads to an improved destination brand and development as a target for foreign direct investment and export of goods.

We recommend that while football revenues are reinvested in the team, additional government revenues attributed to the football team be allocated towards experimenting with economic zones in the three Rarita provinces. These policies should be tailored to the specific needs of the three provinces, but revolve around environmental regulation, property rights, unemployment prevention, and monetary/fiscal policy, which are the areas most heavily correlated with increased economic productivity.

Team Selection

Leveraging FIFA Video Game Data to Rate Players

To assess an individual footballer's skill, a neural network (a type of machine learning algorithm) was built to determine an overall rating for each player. Overall ratings are a concept taken from the FIFA videogame franchise that estimate the ability of players. Overall ratings range from 47 to 99, with 99 being the best. The neural network was fed a dataset of statistics and FIFA 21 overall ratings of 1600 players from the English Premier League, German Bundesliga 1, Spanish LaLiga, French Ligue 1, and Italian Serie A. The model was able to train itself to predict overall ratings for each player based on these statistics. This model then computed the overall rating of FSA players. Another reason to choose FIFA 21 overall ratings is their correlation with team winning percentage, as seen below.

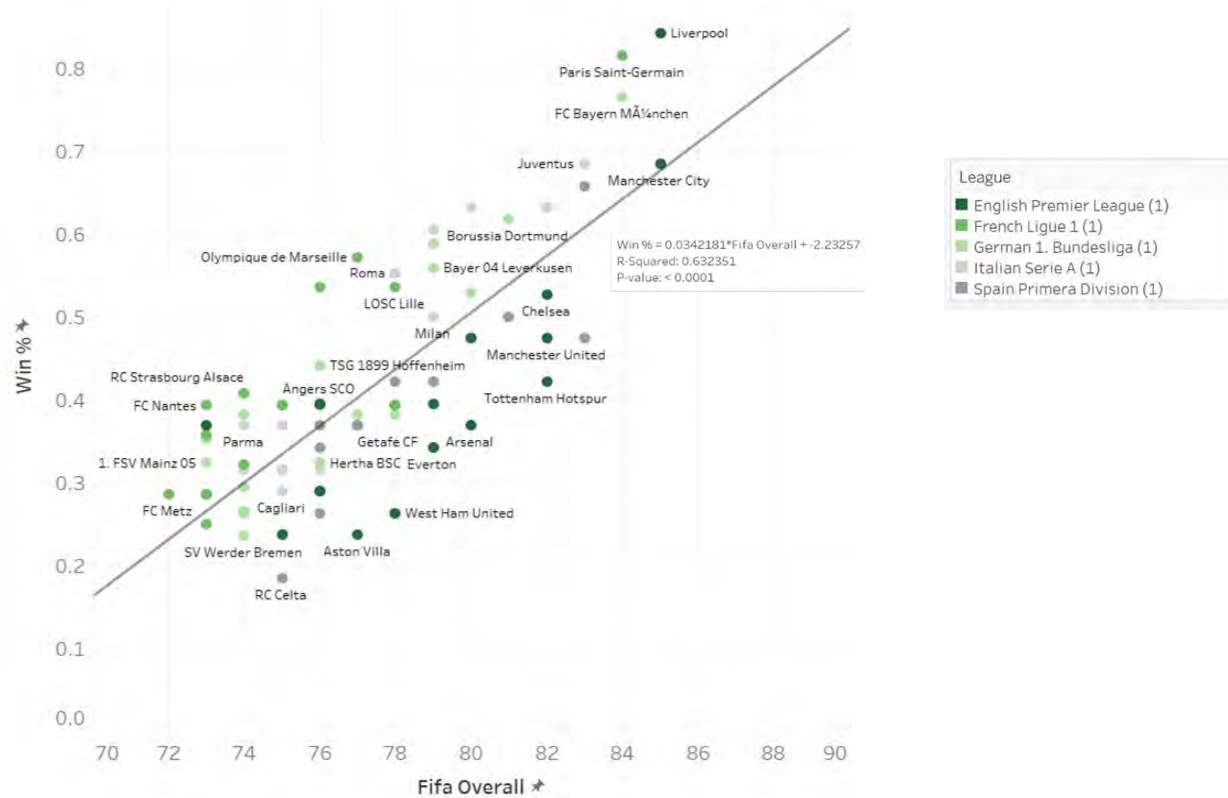

Figure 1: Correlation between overall ratings of teams on FIFA 21 and their subsequent winning percentages. The (1) in the legend indicates this league is the top league in its respective country.

The alternative to this approach was to handpick players with superior statistics in areas that are typically viewed as important by football experts. However, it is difficult to weight the relative importance of these statistics. Further, many of these statistics are influenced heavily by confounding variables such as the team's overall skill and style of play. For example, a team with a talented defense or a team that plays in a formation revolving around good defense prevents quality shots on goal, therefore bolstering the statistics of the goalkeeper. Moreover, a neural network was selected so that many statistics could be included without overfitting, allowing the model to find the statistics more attributable to individuals, thereby mitigating these confounds.

Football Metrics Used in Model

General	Shooting	Passing	Defense	Goalkeeping
Age	Goals / 90	Completion %	Tackles / 90	Goals Allowed / 90
90s	Shot on Target %	Short Pass Completion %	Tackle %	Shots on Target Allowed / 90
Salary	Shots / 90	Medium Pass Completion %	Tackles in Defensive 3rd / 90	Saves / 90
Position	Shot on Target / 90	Long Pass Completion %	Pressure %	Save %
	Goals / Shot	Expected Assists / 90	Pressures in Defensive 3rd / 90	Clean Sheet %
	Goals / Shot on Target	Assisted Shots (KP) / 90	Interceptions / 90	
	Expected Goals / 90	Progressive Passes / 90		

Figure 2: Football statistics used to predict FIFA 21 overall rating

All metrics used were standardized to adjust for volume (either percentages or per 90 minutes). With a limited amount of data to train the model, other statistics commonly viewed as less indicative of player skill were excluded.

Criterion

Once each player had been given an overall rating, an optimization model was run that selected the starting lineup. The first constraint was simply that players had to have played five or more matches to be considered. The second constraint was that at least seven of the eleven starters had to be from Rarita. If the lineup was comprised entirely of foreign players, national pride and the team's reputation would likely be damaged. The third constraint was that two or fewer players could be from the Rarita Football League (RFL). Most top national teams in the World Cup are comprised mostly of players from the top five leagues (Goal.com, 2014). When selecting bench players, age was a strong consideration, as Rarita's future success will hinge upon younger players gaining experience in the FSA League (Fifield, 2018). These players are also RFL players, as it minimized total player expense and would satisfy Rarita football fans. The constraints for player positions were to have greater than or equal to two forwards, six midfielders, four defenders, and one goalkeeper (eleven total players). Having players that can play midfielder along with another position is beneficial for team strategy and versatility in times of injury (Irish Times, 2022). Lastly, one of the two possible lineups used only the ∂995,000,000 initial investment without using any other capital.

Roster Recommendation

Bronze in the lineups below indicate an RFL player and black outlines indicate foreign players.

Starters without Additional Funding – 95 Overall

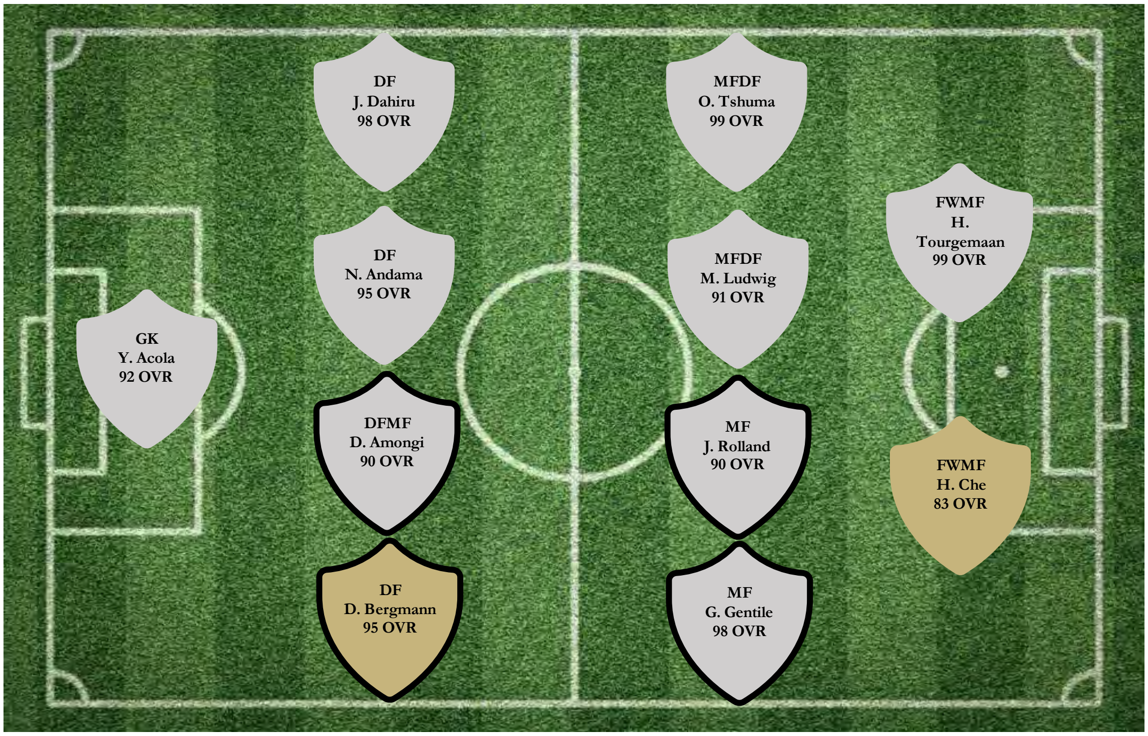

Figure 3: Rarita's optimized starters without additional funding. OVR is abbreviation for overall rating

Starters with Additional Funding – 98 Overall

Figure 4: Rarita's optimized starters without additional funding. OVR is abbreviation for overall rating

Bench Players

Figure 5: Rarita's recommended bench players (regardless of additional funding)

Funding and Revenues

If the lineup requiring additional funding is selected, Rarita can pursue corporate sponsorships for its national football team. These sponsorships would need to total approximately ∂2.3 billion in additional revenue over ten years for the team to break even. Typically, national teams will have an array of sponsors including transportation companies, telecommunication companies, food/beverage companies, financial services, and gambling services (Chilean Men's National Team Sponsors, 2021). Rarita should aim to obtain five or more core sponsorships across these target industries. Our model conservatively assumed that sponsorship revenues will be no larger than average RFL sponsorship revenues, but well-negotiated national team deals could be significantly larger. Supplier sponsorships for competitive teams often have the most lucrative contracts while mitigating supply costs for the team. Germany's national team, which performs similarly internationally to Rarita's expected performance, secured an eight-year deal with Nike worth ∂500 million in 2007, which is nearly ∂800 million today adjusted to inflation (Times of Malta, 2007).

Expected Results

	Additional Funding	No Additional Funding
Probability Top-24 (Qualifying)	100%	99.9%
Probability Top-10	95.3%	69.3%
Probability 1st of 57 teams	24.6%	4.5%
Expected Finish	4	8
97.5th Percentile Year Finish Place	1	1
2.5th Percentile Year Finish Place	12	19

Figure 6: Rarita's expected results and probability ranges of success

Rank	Country	Expected Winning %
1	Dosqaly	0.814
2	Esia	0.812
3	Sobianitedrucy	0.790
4	Nganion	0.788
5	Giumle Lizeibon	0.787
6	Quewenia	0.763
7	**Rarita**	**0.739**
8	Southern Ristan	0.725
9	People's Land of Maneau	0.725
10	Manlisgamncent	0.725
11	Rosvi	0.725
12	Greri Landmoslands	0.696
13	Mico	0.696
14	Varijitri Isles	0.682
15	Redohrainbri	0.667
16	Djipines	0.652
17	Byasier Pujan	0.643
18	Ledian	0.642
19	Nkasland Cronestan	0.637
20	Ngoque Blicri	0.608
21	Moaithe	0.592
22	Xikong	0.589
23	Eastern Niasland	0.561
24	Southslands	0.561

Figure 7: Simulated league standings without additional funding

Rank	Country	Expected Winning %
1	**Rarita**	**0.815**
2	Esia	0.814
3	Dosqaly	0.814
4	Nganion	0.790
5	Giumle Lizeibon	0.789
6	Sobianitedrucy	0.789
7	Quewenia	0.764
8	People's Land of Maneau	0.739
9	Rosvi	0.739
10	Southern Ristan	0.736
11	Manlisgamncent	0.726
12	Greri Landmoslands	0.698
13	Mico	0.698
14	Varijitri Isles	0.684
15	Redohrainbri	0.671
16	Ledian	0.655
17	Byasier Pujan	0.655
18	Djipines	0.645
19	Nkasland Cronestan	0.641
20	Ngoque Blicri	0.625
21	Moaithe	0.594
22	Xikong	0.591
23	Galamily	0.564
24	Eastern Niasland	0.564

Figure 8: Simulated league standings with additional funding

Economic Impact

Expected Government Revenues

Analysis determined that whether teams played in the FSA League and whether they performed well did not impact the team's revenue, and therefore did not affect the government's tax collections. However, a 2012 study found that winning the World Cup was associated with a 1% increase in the value of that country's tourism companies (Nicolau, 2018). Since a company's valuation is calculated by its cash flows, it can be assumed that there is similarly 1% more tourism revenue on average in the year after a World Cup win. This results in a GDP boost of approximately 0.1% since tourism is approximately 10% of GDP (Statista, 2022). When also considering 2nd, 3rd, and 4th place finishes having smaller but still positive effects on tourism revenue, it was calculated that Rarita's government can expect to earn over ∂3 million over the course of ten years under the team that has additional funding. Without additional funding, this number is just over ∂1 million.

Although these numbers are immaterial when viewed beside Rarita's federal budget, the "halo effect" of winning this league would likely be much more economically significant. In addition to becoming more of a destination brand, businesses contemplating foreign direct investment and importing countries seeking to buy goods would view Rarita more favorably compared to previous years (Gholipour, 2020). However, this halo effect cannot be calculated with any certainty.

Leveraging Additional Government Revenues to Further Accelerate Growth

The best way to bolster an economy is by having strong incentives for creativity and innovation fostered by thoughtful, enforced governmental policy. Studies have shown that economically outperforming countries tend to experiment more than their counterparts with new policies that influence markets (Madgavkar, 2021). Many of these countries use economic zones to test these experimental policies before broadly implementing them. These policies can be tailored to certain zones representative of the holistic economies of the provinces (West, East, and Central Rarita) (OECD, 2010).

Rarita should use additional government revenues attributed to FSA success to experiment with policies in economic zones. We recommend that Rarita experiment with varying monetary and fiscal policies, environmental policies, unemployment-preventative policies, and property-rights policies that stimulate economic growth. Rarita should measure GDP and Gini-coefficients (an index measuring the disparity between rich and poor) to evaluate the success of these policies. Once these policies are implemented on a wider scale, enforcement is crucial. Rarita should ensure it has well-funded regulatory agencies and a strong judiciary system that can give streamlined, consistent decisions.

Implementation Plan

Timeline

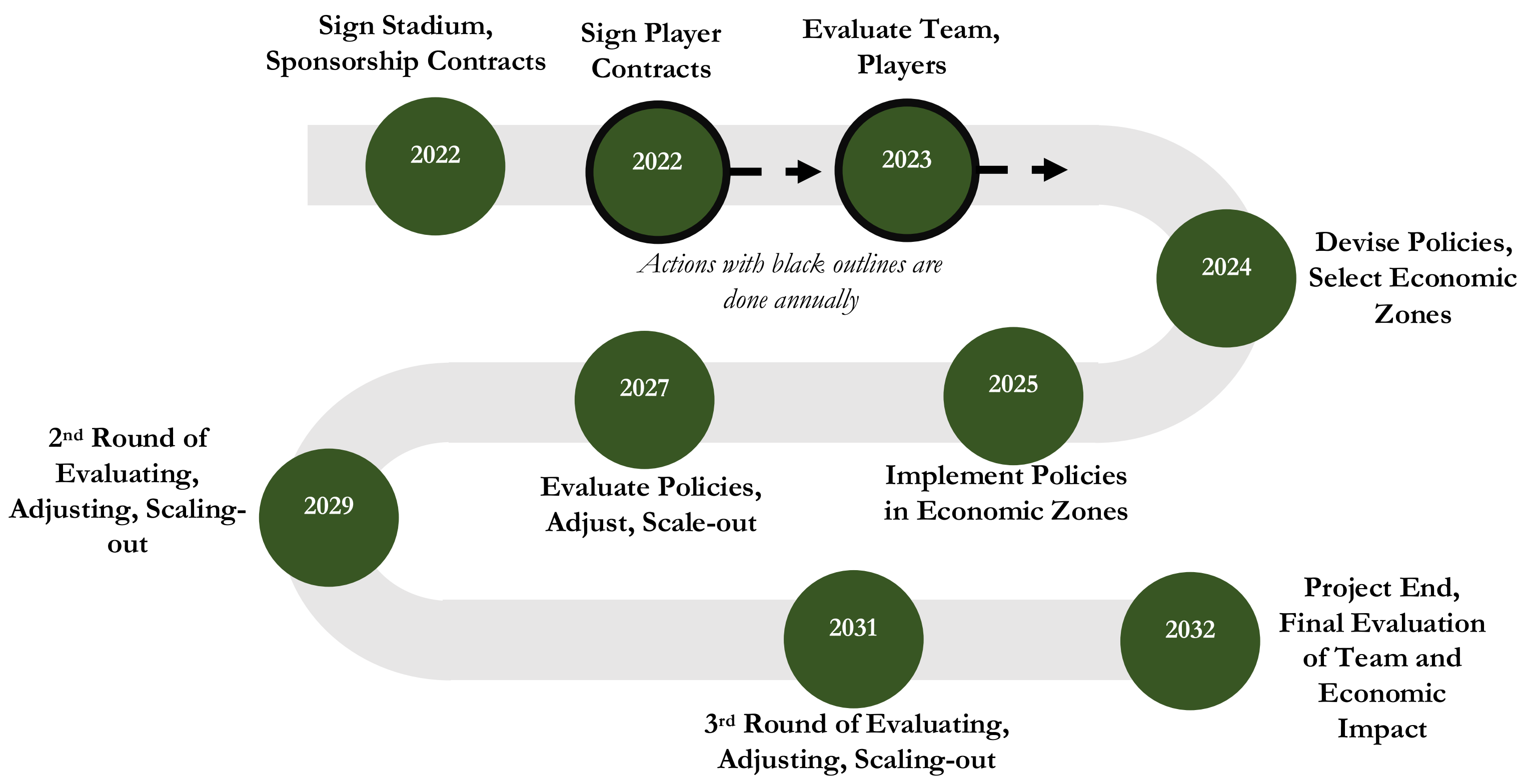

Figure 9: 10-year timeline depicting implementation plan

Football Implementation

All football players can be signed immediately in preparation for the 2022-2023 season. Rarita's National Team should also seek in the near-term to secure deals with sponsors and a large, centralized stadium with the capacity and infrastructure to handle crowds larger than typical RFL matches. This stadium should also be within driving distance of an airport to cater to tourists and other national team fans.

At the end of each season, the success of the team and players should be evaluated. The team should be evaluated based on winning percentage. Players should be evaluated by both qualitative, observed factors and their overall rating from the past year calculated by the algorithm. Finally, the optimization model should be run to help pick the new team, with an additional constraint that most of the team should return to maintain team chemistry.

Economic Implementation

Policies should be devised and economic zones should be identified within Central, West, and East Rarita by the end of the second season. Three years after the team's construction, when additional government revenues have been recognized, Rarita should implement these policies in the economic zones. After two years of observing how these policies affect GDP and the Gini-coefficient, they should either be adjusted or scaled out to the rest of the province.

Assumptions

Economic Assumptions

No clear pattern emerged from a time-series analysis of inflation rates or one-year risk free rates. Further, both assumptions were flatlined and assumed to follow a normal distribution. The risk free rate was assumed to be 1.12% with a standard deviation of 1.16% and the inflation rate was assumed to be 3.08% with a standard deviation of 1.21%.

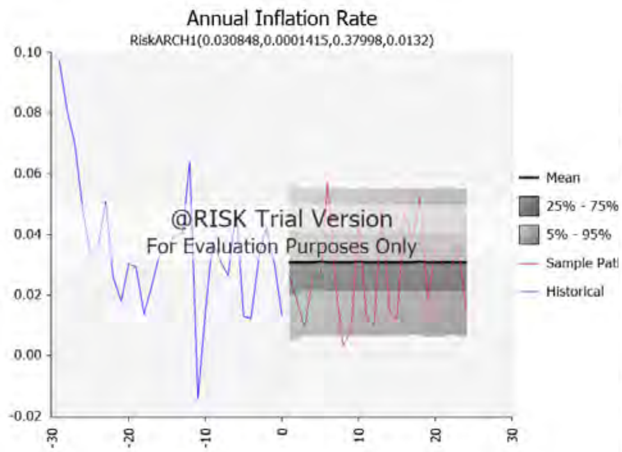

Figure 10: Time-series analysis of inflation rate

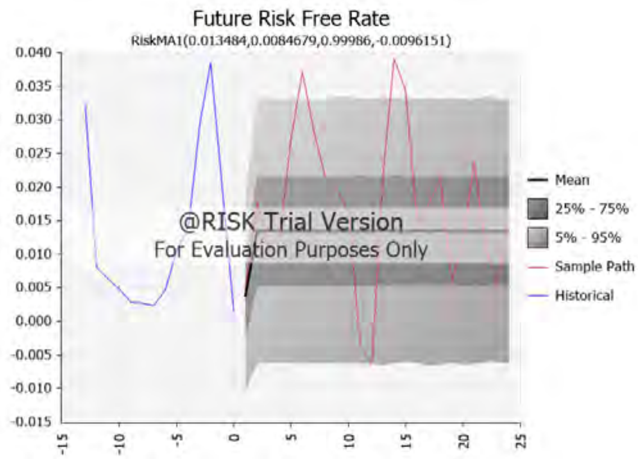

Figure 11: Time-series analysis of risk-free rate

Football Environment Assumptions

Our analysis found that team revenues minus costs other than player salaries per capita should be estimated at ∂7.83 with a standard deviation of ∂0.55. We assumed that salaries will grow at the inflation rate, and it was conservatively assumed that football revenues will not grow since the football revenue tables supplied did not show evidence of growth. It was also assumed that more funds will come in from the government after the ten-year project has ended.

We assumed that the overall team rating Rarita will be able to achieve with its inflation-adjusted salary in future years will not change. We also assumed that players that played in the FSA Tournament but were not included in the dataset for the A, B, C, D, E, and RFL leagues were in a lower league with a similar level of competition to the RFL. Their overall ratings were assumed to be the median of RFL players.

Key Risks, Risk Mitigation, Sensitivity Analysis

Risk Matrix

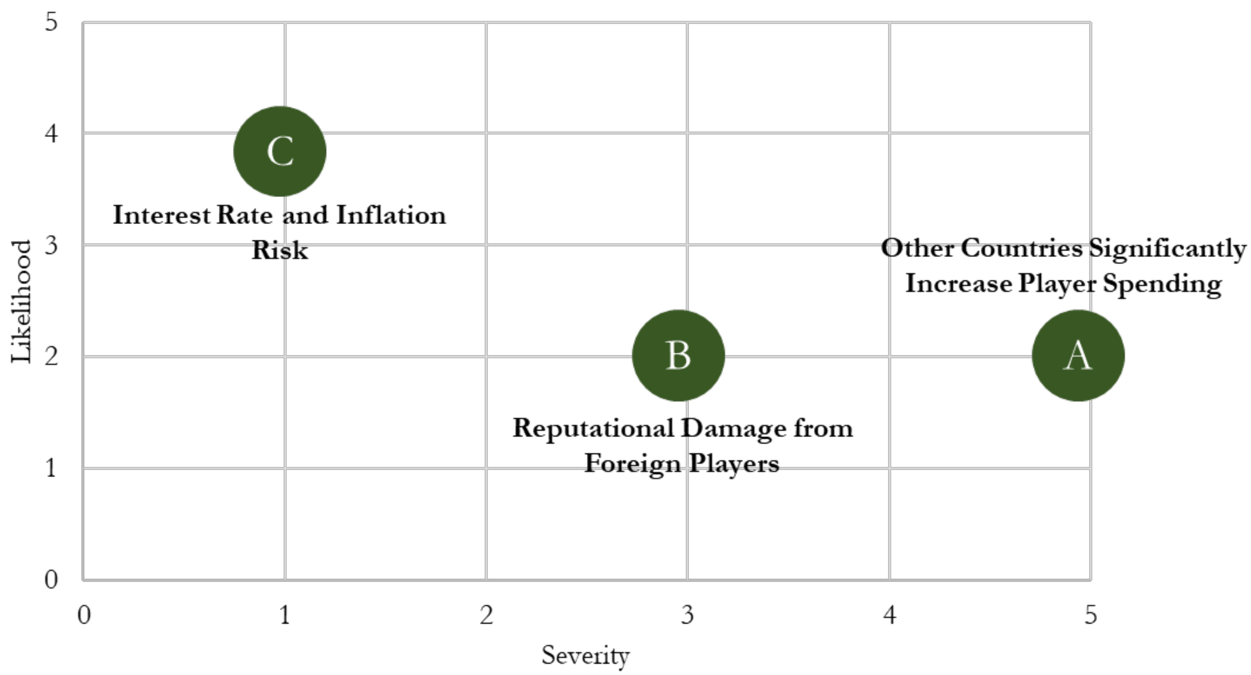

Figure 12: Risk matrix

Competition Spending (A)

Other countries could together decide to invest more in football players to build their own brand image and boost their economy. This would push up the price of players, harming Rarita's ability to construct a competitive team while staying within the budget of the initial investment. To mitigate this risk, Rarita should seek corporate sponsorships if salary growth begins to outpace expectations.

Reputational Damage Caused from Foreign Players (B)

If other teams did not pursue foreign players to the degree that Rarita does, fans could lose national pride and the halo effect mentioned earlier could become an adverse factor to the economy where businesses, importers, and tourists associate a bad image with Rarita. This is mitigated through only having four foreign players out of eleven starters and by re-evaluating lineups annually.

Interest Rate and Inflation Risk (C)

Higher than expected inflation rates would decrease the purchasing power of the initial investment in later years of the project. Lower risk free rates would decrease the team's return on these funds, leaving less money for salaries. To mitigate this risk for the roster without additional funding, a ∂50 million margin of safety was built into the model. The probability of the full ∂995 million being used within ten years is 1.5%.

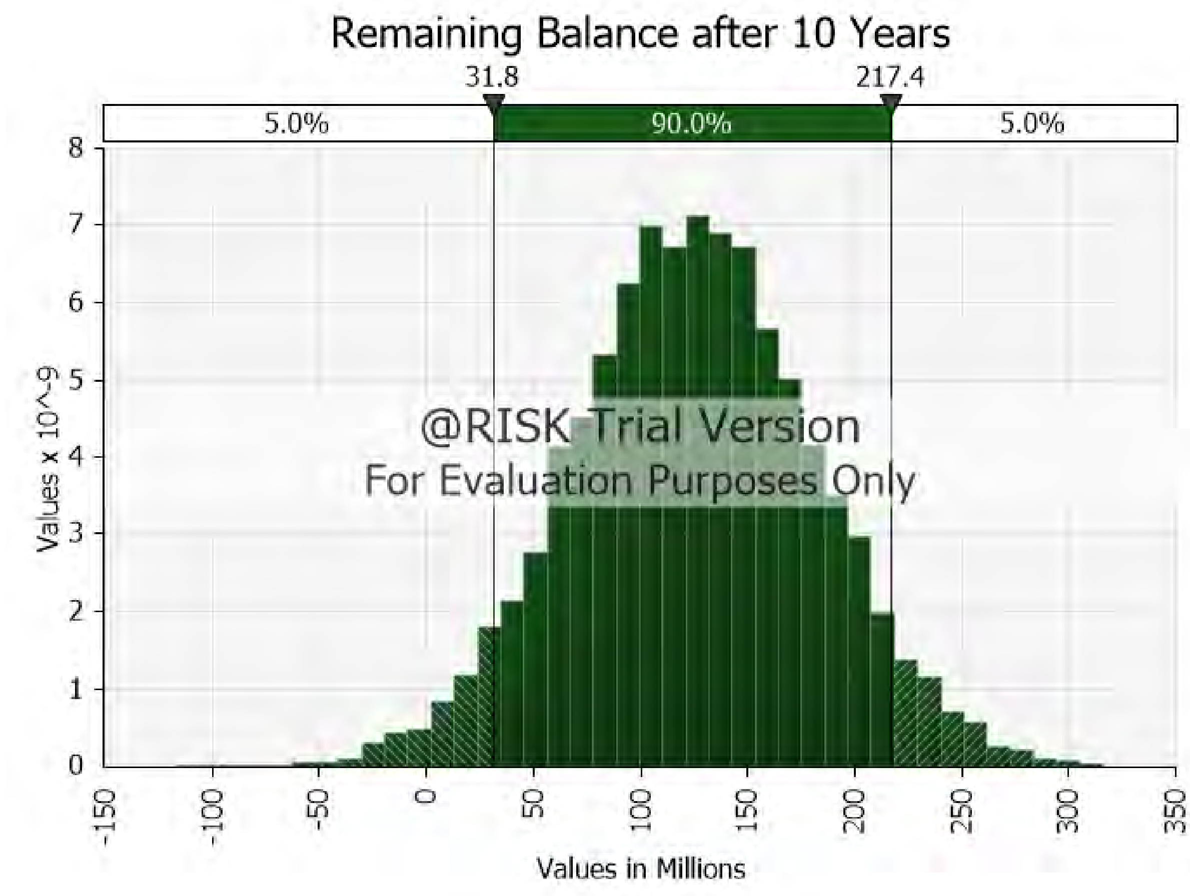

Figure 13: Sensitivity analysis for the final remaining balance of initial investment after ten years with varying risk free rate and inflation.

Data and Data Limitations

External Data Sources

FBREF is an online database for football statistics. This database was used to find the statistics of 1600 players from the English Premier League, German Bundesliga 1, Spanish LaLiga, French Ligue 1, and Italian Serie A.

Kaggle is a data repository that has been scraped and cleaned by a community of users and professional data scientists. The FIFA 21 overall ratings for each player came from this dataset and were matched to their 2019-2020 statistics from FBREF to create the dataset fed to the neural network.

Contributors to Football Success Go Beyond the Data

One qualitative but crucially important limitation is that statistics do not tell the entire story in football. Although one player may have great statistics, they may play in a manner detrimental to their teammates. Players may also create friction or drama in the locker room that lowers morale and performance. On the other hand, some players may have mediocre stats but be paramount to a team's culture, leadership, and youth development. In this case, replacing this player with somebody who has a higher overall rating would cause the team to lose its identity and likely lose more games.

Lastly, given this data, there was not an opportunity to select a coach, which is perhaps more important than any player. Coaches historically select players based on the plays and formations they employ. How well these strategies work and how well a coach can adapt to the other team's strategy can compensate for less talent. Further, even with confidence in the recommended team's skill, it is difficult to know if they will play well as a team or fit under the scheme of Rarita's national team coach.

Inaccuracies in Supplied Data

The supplied data for players had some data entry errors. For example, shots on target percentage for some players was negative, which is not possible. In these cases, we filled the value with a zero. Null values were also filled with a zero. This was preferable to throwing out records, as this would have meant potentially losing out on a skilled player who was not given an overall rating.

Limitations of Neural Network

There are two main limitations to the neural network. First, 1600 rows of data is simply not that much data on which to train the model. More rows would make the model more accurate. Secondly, the algorithm was not able to factor in that RFL players play against inferior competition. Salary was the only variable that likely indicated an RFL player to the algorithm.

Appendix

I. Calculating Team's Overall Rating

The *initial* overall FIFA 21 team rating is calculated by averaging the overall ratings of the eleven starters. However, there is an adjustment that must be made to find the *final* overall. If a player's overall is above the team's initial overall, that difference (called the correction factor) is added onto the sum of all eleven of the player overall ratings and then divided by eleven. No adjustment is made if the player is below the team's initial overall rating.

II. Calculating Winning Probability Based on Overall Team Difference

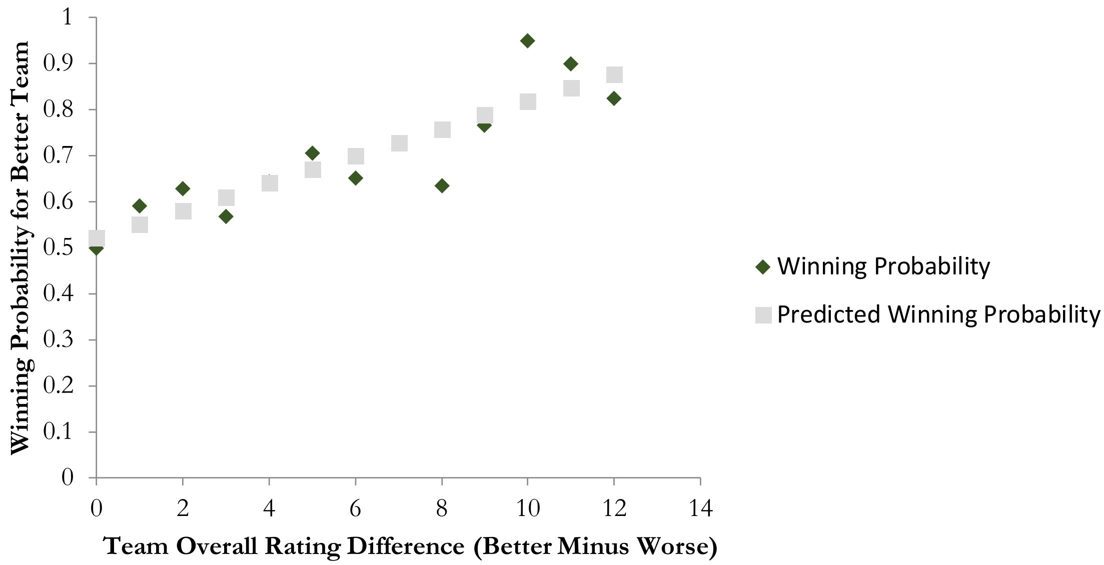

Figure 14: Winning probability of better team as a function of FIFA 21 overall rating difference

This plot was constructed using 533 fixtures from the English Premier League and German Bundesliga. This equation was used to predict the standings of the 2023 FSA League:

Winning Probability of Better Team = 0.5 + 0.030 * Absolute Overall Difference

Standard Error: 0.0665 | p-value on slope: < 0.0001

III. Neural Network Logical Diagram

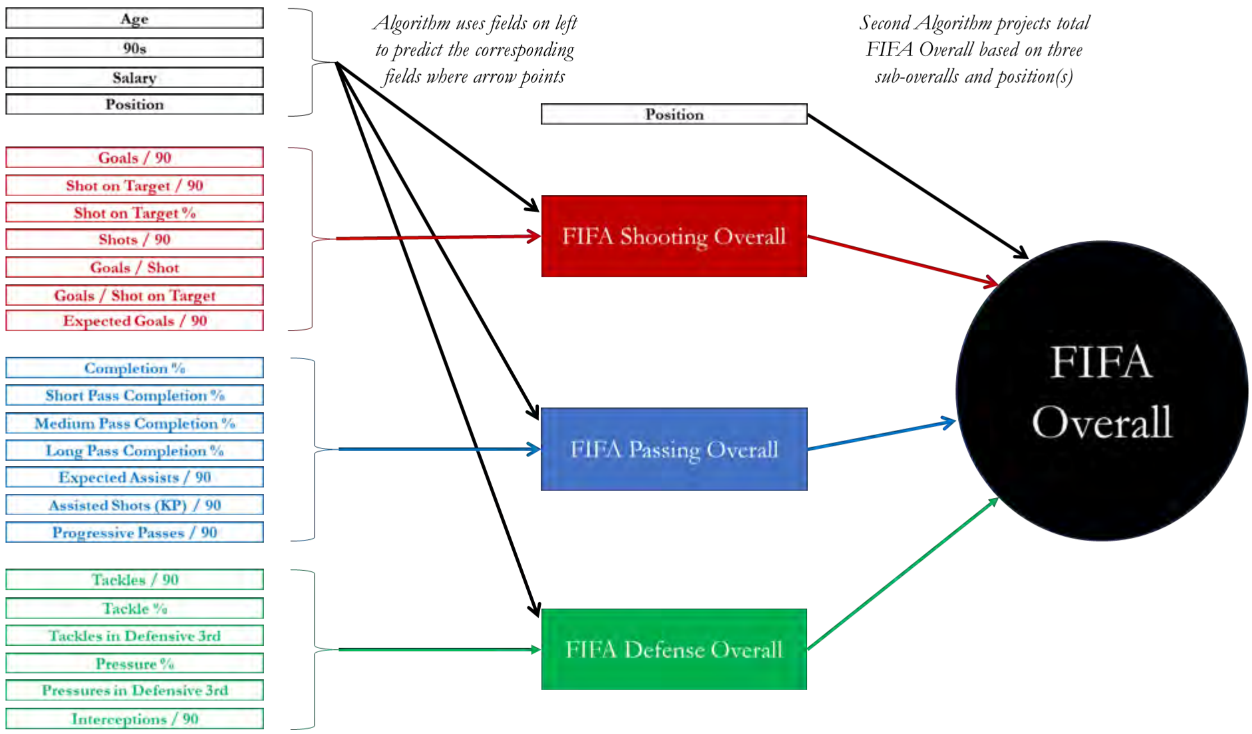

Figure 15: Neural network diagram

IV. Full FSA League Standings (38 games)

Projected League Table (Standings) – Team Pursues Additional Funding

Rank	Country	Expected Winning %	2.5th Winning %	97.5th Winning %	Probability Top-24	Probabilit y Top-10	Probability 1st
1	**Rarita**	**0.815**	**0.711**	**0.921**	**100.00%**	**95.25%**	**24.58%**
2	Esia	0.814	0.684	0.921	100.00%	94.65%	24.73%
3	Dosqaly	0.814	0.684	0.921	100.00%	94.71%	24.09%
4	Nganion	0.790	0.658	0.895	99.97%	87.83%	14.13%
5	Giumle Lizeibon	0.789	0.658	0.895	99.99%	88.11%	13.97%
6	Sobianitedrucy	0.789	0.658	0.895	100.00%	87.81%	13.97%
7	Quewenia	0.764	0.632	0.895	99.96%	77.92%	7.33%
8	People's Land of Maneau	0.739	0.605	0.868	99.81%	65.02%	3.52%
9	Rosvi	0.739	0.605	0.868	99.88%	63.85%	3.61%
10	Southern Ristan	0.736	0.605	0.868	99.75%	62.52%	3.16%
11	Manlisgamncent	0.726	0.605	0.842	99.78%	55.73%	2.63%
12	Greri Landmoslands	0.698	0.553	0.816	99.00%	39.43%	1.04%
13	Mico	0.698	0.553	0.816	99.01%	39.27%	0.90%
14	Varijitri Isles	0.684	0.553	0.816	98.46%	32.15%	0.63%
15	Redohrainbri	0.671	0.526	0.816	97.22%	25.82%	0.53%
16	Ledian	0.655	0.500	0.789	95.49%	19.52%	0.31%

17	Byasier Pujan	0.655	0.500	0.789	95.39%	19.17%	0.15%
18	Djipines	0.645	0.500	0.789	93.77%	16.37%	0.20%
19	Nkasland Cronestan	0.641	0.500	0.789	93.50%	14.57%	0.11%
20	Ngoque Blicri	0.625	0.474	0.763	90.06%	10.12%	0.08%
21	Moaithe	0.594	0.447	0.737	80.09%	4.88%	0.04%
22	Xikong	0.591	0.447	0.737	78.94%	4.78%	0.01%
23	Galamily	0.564	0.421	0.711	67.63%	1.92%	0.00%
24	Eastern Niasland	0.564	0.421	0.711	66.78%	2.15%	0.01%
25	Southslands	0.563	0.421	0.711	66.70%	2.13%	0.00%
26	Loco Phirema	0.475	0.316	0.632	22.05%	0.14%	0.00%
27	Saintu	0.474	0.316	0.632	21.59%	0.08%	0.00%
28	Unicorporated Tiagascar	0.470	0.316	0.605	20.45%	0.06%	0.00%
29	Dastatesne	0.445	0.289	0.605	12.28%	0.05%	0.00%
30	Highhlaands	0.445	0.289	0.579	12.36%	0.05%	0.00%
31	Leoneku Guidisia	0.445	0.289	0.605	12.41%	0.03%	0.00%
32	Cabral Retrea	0.442	0.289	0.579	11.57%	0.04%	0.00%
33	Isle of Lababwe	0.442	0.289	0.579	11.64%	0.02%	0.00%
34	Pahon	0.442	0.289	0.579	11.06%	0.02%	0.00%
35	New Uwi	0.442	0.289	0.579	11.34%	0.02%	0.00%
36	Eastern Sleboube	0.413	0.263	0.553	5.81%	0.01%	0.00%
37	Cuandbo	0.409	0.263	0.553	4.95%	0.00%	0.00%
38	Bernepamar	0.383	0.237	0.526	2.55%	0.00%	0.00%
39	Liacra	0.327	0.184	0.474	0.31%	0.00%	0.00%
40	Ingre	0.325	0.184	0.474	0.16%	0.00%	0.00%
41	Ili Siaco	0.301	0.184	0.447	0.10%	0.00%	0.00%
42	Kesternsri	0.298	0.158	0.447	0.05%	0.00%	0.00%
43	Humberstonia	0.273	0.158	0.421	0.03%	0.00%	0.00%
44	Prometricia	0.270	0.132	0.421	0.03%	0.00%	0.00%
45	West Iyan	0.270	0.158	0.395	0.00%	0.00%	0.00%
46	Cabballi	0.270	0.158	0.395	0.03%	0.00%	0.00%
47	Tiagascar Westlands	0.270	0.132	0.395	0.01%	0.00%	0.00%
48	Iyan	0.270	0.132	0.395	0.00%	0.00%	0.00%
49	Tiliqoiuy	0.270	0.132	0.395	0.01%	0.00%	0.00%
50	Naguayli	0.269	0.132	0.395	0.01%	0.00%	0.00%
51	Deshslands Landdenhai	0.269	0.132	0.395	0.01%	0.00%	0.00%
52	Klausterton	0.269	0.132	0.395	0.01%	0.00%	0.00%
53	Isle of Jabber	0.269	0.132	0.395	0.01%	0.00%	0.00%
54	Kani	0.269	0.132	0.395	0.04%	0.00%	0.00%
55	Mandlestan	0.268	0.132	0.395	0.01%	0.00%	0.00%
56	Bernoullist	0.265	0.132	0.395	0.00%	0.00%	0.00%

Figure 16: Projected league standings with additional funding

Projected League Table (Standings) – No Additional Funding

Rank	Country	Expected Winning %	2.5th Winning %	97.5th Winning %	Probability Top-24	Probability Top-10	Probability 1st
1	Dosqaly	0.814	0.684	0.921	100.00%	95.84%	28.81%
2	Esia	0.812	0.684	0.921	100.00%	95.66%	27.01%
3	Sobianitedrucy	0.790	0.658	0.895	99.99%	90.50%	16.95%
4	Nganion	0.788	0.658	0.895	100.00%	89.84%	16.95%
5	Giumle Lizeibon	0.787	0.658	0.895	100.00%	90.13%	16.17%
6	Quewenia	0.763	0.632	0.895	99.98%	81.05%	9.29%
7	**Rarita**	**0.739**	**0.605**	**0.868**	**99.85%**	**69.30%**	**4.51%**
8	Southern Ristan	0.725	0.579	0.842	99.77%	61.07%	3.34%
9	People's Land of Maneau	0.725	0.579	0.842	99.73%	60.41%	3.17%
10	Manlisgamncent	0.725	0.579	0.842	99.70%	61.06%	3.05%
11	Rosvi	0.725	0.579	0.842	99.71%	60.34%	3.19%
12	Greri Landmoslands	0.696	0.553	0.816	98.97%	43.72%	1.40%
13	Mico	0.696	0.553	0.816	99.00%	43.59%	1.38%
14	Varijitri Isles	0.682	0.553	0.816	98.27%	35.99%	0.84%
15	Redohrainbri	0.667	0.526	0.789	97.62%	27.76%	0.49%
16	Djipines	0.652	0.500	0.789	95.61%	21.56%	0.32%
17	Byasier Pujan	0.643	0.500	0.789	94.58%	17.91%	0.26%
18	Ledian	0.642	0.500	0.789	94.14%	18.31%	0.23%
19	Nkasland Cronestan	0.637	0.500	0.763	93.46%	16.67%	0.13%
20	Ngoque Blicri	0.608	0.474	0.737	86.68%	8.23%	0.07%
21	Moaithe	0.592	0.447	0.737	81.39%	5.45%	0.02%
22	Xikong	0.589	0.447	0.737	80.34%	5.62%	0.02%
23	Eastern Niasland	0.561	0.421	0.711	68.11%	2.74%	0.00%
24	Southslands	0.561	0.421	0.711	68.52%	2.47%	0.01%
25	Galamily	0.552	0.395	0.711	63.28%	1.86%	0.01%
26	Saintu	0.470	0.316	0.632	21.86%	0.10%	0.00%
27	Loco Phirema	0.470	0.316	0.605	21.63%	0.04%	0.00%
28	Unicorporated Tiagascar	0.467	0.316	0.605	20.99%	0.09%	0.00%
29	Dastatesne	0.440	0.289	0.579	12.47%	0.04%	0.00%
30	Highhlaands	0.440	0.289	0.579	12.49%	0.03%	0.00%
31	New Uwi	0.440	0.289	0.579	11.85%	0.03%	0.00%
32	Leoneku Guidisia	0.440	0.289	0.579	12.66%	0.00%	0.00%
33	Isle of Lababwe	0.440	0.289	0.579	12.59%	0.02%	0.00%
34	Pahon	0.440	0.289	0.579	11.82%	0.03%	0.00%
35	Cabral Retrea	0.436	0.289	0.579	11.21%	0.02%	0.00%
36	Eastern Sleboube	0.409	0.263	0.553	5.52%	0.00%	0.00%
37	Cuandbo	0.407	0.263	0.553	5.33%	0.01%	0.00%
38	Bernepamar	0.380	0.237	0.526	2.55%	0.00%	0.00%
39	Ingre	0.323	0.184	0.474	0.21%	0.00%	0.00%
40	Liacra	0.320	0.184	0.474	0.28%	0.00%	0.00%

41	Kesternsri	0.294	0.158	0.421	0.09%	0.00%	0.00%
42	Ili Siaco	0.291	0.158	0.421	0.06%	0.00%	0.00%
43	Bernoullist	0.271	0.132	0.395	0.01%	0.00%	0.00%
44	Cabballi	0.271	0.132	0.395	0.02%	0.00%	0.00%
45	Prometricia	0.271	0.132	0.395	0.03%	0.00%	0.00%
46	Iyan	0.271	0.132	0.395	0.02%	0.00%	0.00%
47	Isle of Jabber	0.271	0.132	0.421	0.06%	0.00%	0.00%
48	Tiagascar Westlands	0.270	0.158	0.395	0.01%	0.00%	0.00%
49	Tiliqoiuy	0.270	0.158	0.395	0.03%	0.00%	0.00%
50	Kani	0.270	0.158	0.395	0.02%	0.00%	0.00%
51	Deshslands Landdenhai	0.270	0.132	0.395	0.02%	0.00%	0.00%
52	Mandlestan	0.270	0.132	0.395	0.02%	0.00%	0.00%
53	Klausterton	0.270	0.132	0.395	0.05%	0.00%	0.00%
54	West Iyan	0.269	0.158	0.395	0.00%	0.00%	0.00%
55	Humberstonia	0.258	0.132	0.395	0.00%	0.00%	0.00%
56	Naguayli	0.258	0.132	0.395	0.00%	0.00%	0.00%

Figure 17: Projected league standings without additional funding

V. Solver Model Setups

Note: OpenSolver had to be installed to solve these problems, as there were too many decision variables for the built-in Excel solver tool.

Each player has a decision variable that takes on a zero or one. A zero means that player is not selected and a one means that player is selected. The solver to the right maximizes the team's overall rating while constraining position counts, total player counts, RFL counts, and the remaining balance on the government's initial investment.

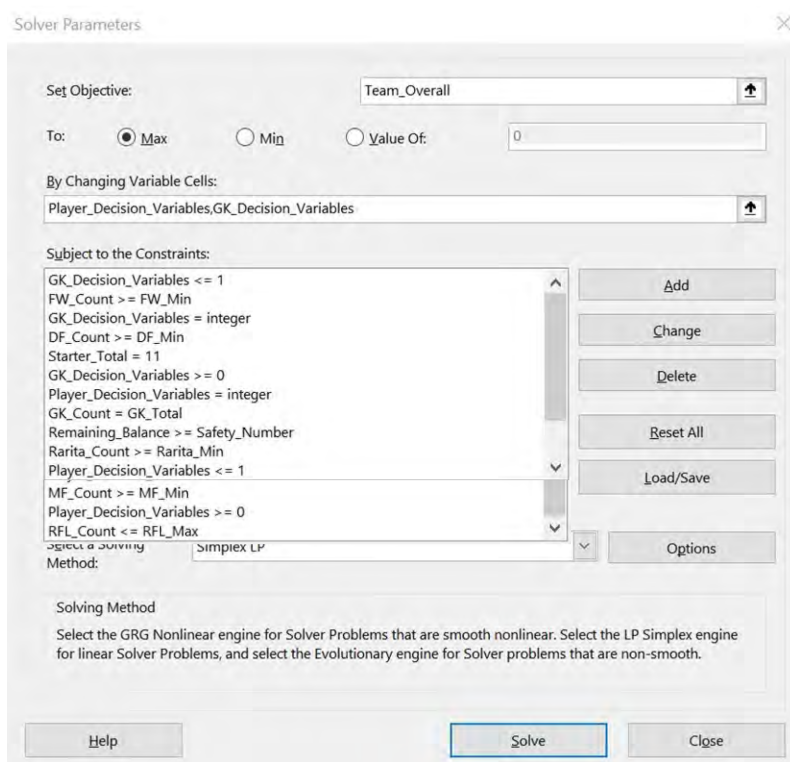

Figure 17: Optimization model setup for Rarita Lineup

Each potential matchup was given either a zero or one. A zero means the matchup was not played and a one means the matchup is played. Since there are 56 teams total, teams do not play each other more than once. This solver looks for any solution where the "strength of schedule" is comparable and every team plays 38 total games. This was needed to simulate the range of probability of success for Rarita's possible national league teams.

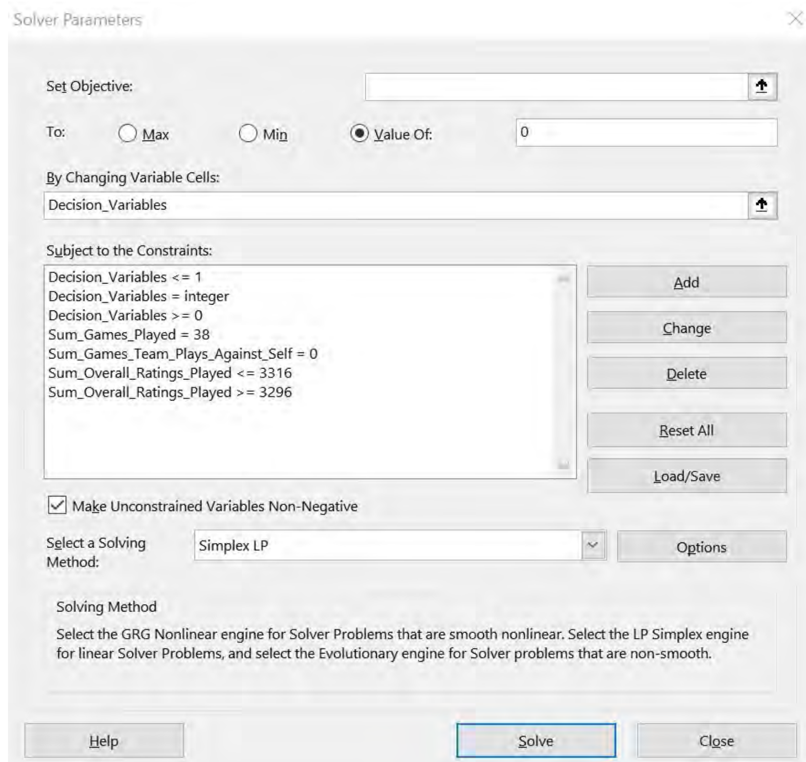

Figure 18: Optimization model setup for FSA League schedule

VI. Risk Function Setups

- To randomize inflation/risk free rate:=@RiskNormal (Mean, Standard Deviation)
- To randomize whether team wins a game: =@RiskBinomial(1, Winning Probability)
- To find percentiles: =@RiskPercentile(Cell that changes each iteration, percentile)
- To find means: =@RiskMean(Cell that changes each iteration)
- *Every simulation was run with 10,000 iterations

VII. Neural Network Code

training set: X, y

X: real named player gametime statistics

y: player's real fifa rating

RESULTING MODEL:

input: any player's gametime statistics

output: predicted fifa rating

In [38]:
```
import numpy as np
import pandas as pd
```

Import training data (real player data + real fifa ratings)

In [39]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remou
nt=True).
```

In [40]:
```
file = '/content/drive/MyDrive/School/SoccerData/SoFrickinNeural.csv'
file_goalies = '/content/drive/MyDrive/School/SoccerData/RealGoalieData.csv'
df = pd.read_csv(file, encoding='latin1')
df_g = pd.read_csv(file_goalies, encoding='latin1')
df_g.head()
```

Out[40]:

	long_name	age	club_name	league_name	overall	Salary	90s	GA90	SoTA/90	Saves/90	Save%	CS%
0	Jan Oblak	27	Atlético Madrid	Spain Primera Division	91	6500000	37.7	0.72	2.785146	2.175066	77.1	44.7
1	Marc-André ter Stegen	28	FC Barcelona	Spain Primera Division	90	13520000	36.0	1.00	3.055556	2.166667	72.7	38.9
2	Alisson	27	Liverpool	English Premier League	90	8320000	28.3	0.81	2.791519	2.049470	72.2	44.8
3	Thibaut Courtois	28	Real Madrid	Spain Primera Division	89	13000000	34.0	0.59	2.794118	2.176471	78.9	52.9
4	Manuel Neuer	34	FC Bayern München	German 1. Bundesliga	89	6500000	33.0	0.94	3.363636	2.545455	74.8	45.5

Data cleaning

In [41]:
```
df["Salary"] = df["Salary"].str.replace(',', '')
df["Salary"] = df["Salary"].str.replace('$', '')
df["Salary"] = pd.to_numeric(df["Salary"])
```

```
df.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600 entries, 0 to 1599
Data columns (total 75 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   long_name        1600 non-null   object 
 1   age              1600 non-null   int64  
 2   club_name        1600 non-null   object 
 3   league_name      1600 non-null   object 
 4   overall          1600 non-null   int64  
 5   potential        1600 non-null   int64  
 6   Salary           1600 non-null   float64
 7   Position_SOA     1600 non-null   object 
 8   MF               1600 non-null   int64  
 9   DF               1600 non-null   int64  
 10  FW               1600 non-null   int64  
 11  real_face        1600 non-null   int64  
 12  pace             1599 non-null   float64
 13  shooting         1599 non-null   float64
 14  passing          1599 non-null   float64
 15  defending        1599 non-null   float64
 16  90s              1600 non-null   float64
 17  Gls              1600 non-null   int64  
 18  Sh               1600 non-null   int64  
 19  SoT              1600 non-null   int64  
 20  SoT%             1600 non-null   float64
 21  Sh/90            1600 non-null   float64
 22  SoT/90           1600 non-null   float64
 23  G/Sh             1600 non-null   float64
 24  G/SoT            1598 non-null   float64
 25  xG               1600 non-null   float64
 26  npxG             1600 non-null   float64
 27  npxG/Sh          1600 non-null   float64
 28  G-xG             1600 non-null   float64
 29  np:G-xG          1600 non-null   float64
 30  Tkl              1600 non-null   int64  
 31  TklW             1600 non-null   int64  
 32  Def 3rd          1600 non-null   int64  
 33  Mid 3rd          1600 non-null   int64  
 34  Att 3rd          1600 non-null   int64  
 35  Tkl_1            1600 non-null   int64  
 36  Tkl_Att          1600 non-null   int64  
 37  Tkl%             1599 non-null   float64
 38  Dribbles_Past    1600 non-null   int64  
 39  Press            1600 non-null   int64  
 40  Press_Succ       1600 non-null   int64  
 41  Press_Succ_%     1599 non-null   float64
 42  Def 3rd_2        1600 non-null   int64  
 43  Mid 3rd_3        1600 non-null   int64  
 44  Att 3rd_4        1600 non-null   int64  
 45  Blocks           1600 non-null   int64  
 46  Blocks_Shooting  1600 non-null   int64  
 47  Blocks_SoT       1600 non-null   int64  
 48  Blocks_Pass      1600 non-null   int64  
 49  Int              1600 non-null   int64  
 50  Tkl+Int          1600 non-null   int64  
 51  Clr              1600 non-null   int64  
 52  Err              1600 non-null   int64  
 53  Cmp              1600 non-null   int64  
 54  Att              1600 non-null   int64  
 55  Cmp%             1598 non-null   float64
 56  TotDist          1600 non-null   int64  
 57  PrgDist          1600 non-null   int64  
 58  Short_Cmps       1600 non-null   int64  
 59  Short_Att        1600 non-null   int64  
 60  Short_Cmp%       1598 non-null   float64
 61  Med_Cmps         1600 non-null   int64  
 62  Medium_Att       1600 non-null   int64  
 63  Medium_Cmp_%2    1593 non-null   float64
 64  Long_Cmps        1600 non-null   int64  
 65  Long_Cmp_Att     1600 non-null   int64  
 66  Long_Cmp%        1579 non-null   float64
 67  Ast              1600 non-null   int64  
 68  xA               1600 non-null   float64
 69  A-xA             1600 non-null   float64
 70  KP               1600 non-null   int64  
 71  1/3              1600 non-null   int64  
 72  PPA              1600 non-null   int64  
 73  CrsPA            1600 non-null   int64  
 74  Prog             1600 non-null   int64  
dtypes: float64(24), int64(47), object(4)
memory usage: 937.6+ KB
```

Define helper function to trim NaN rows

In [43]:
```
def trim_na(df):
  return df.replace([np.inf, -np.inf], np.nan).dropna(axis=0)
```

Normalize the counting-style stats by 90s played

In [44]:
```
counting_stats_labels = ['Gls', 'KP', 'Prog', 'Int', 'Tkl', 'Def 3rd', 'Def 3rd_2']
```

In [45]:
```
df = trim_na(df[df['90s'] != 0]) # remove any players who haven't played a full 90
df[counting_stats_labels] = df[counting_stats_labels].div(df['90s'], axis=0)
```

Deciding which columns to use for each specific attribute we want to predict

In [46]:
```
other_labels = ['long_name', 'overall', 'shooting', 'passing', 'defending']
other_labels_gol = ['long_name', 'overall']
# labels by individual fifa stat (shooting, passing, defending)
sho_stats_labels = ['90s', 'age', 'Salary', 'Gls', 'SoT%', 'Sh/90', 'SoT/90', 'G/Sh', 'G/SoT', 'xG']
pas_stats_labels = ['90s', 'age', 'Salary', 'Cmp%', 'Short_Cmp%', 'Medium_Cmp_%2', 'Long_Cmp%', 'xA', 'KP', 'Prog']
dfn_stats_labels = ['90s', 'age', 'Salary', 'Tkl', 'Tkl%', 'Press_Succ_%', 'Int', 'Def 3rd', 'Def 3rd_2']
# labels for predicting overall from position + shooting, passing, defending
ove_stats_labels = ['shooting', 'passing', 'defending', 'FW', 'MF', 'DF']
# labels for predicting overall of a goalie
gol_stats_labels = ['90s', 'age', 'Salary', 'GA90', 'SoTA/90', 'Saves/90', 'Save%', 'CS%']

all_input_stats = sho_stats_labels + pas_stats_labels + dfn_stats_labels + ['FW', 'MF', 'DF']
all_input_stats = list(set(all_input_stats))
```

In [47]:
```
sho = df[other_labels+sho_stats_labels].copy()
pas = df[other_labels+pas_stats_labels].copy()
dfn = df[other_labels+dfn_stats_labels].copy()
ove = df[other_labels+ove_stats_labels].copy()
gol = df_g[other_labels_gol+gol_stats_labels].copy()
```

Feature scaling

In [48]:
```
from sklearn.preprocessing import StandardScaler
```

In [49]:
```
# returns feature scaled input, then the scaler object
def feature_scaling(input_X):
  sc = StandardScaler()
  X_s = input_X.copy()
  X_s = sc.fit_transform(X_s)
  return X_s, sc # X_s is the feature-scaled version of the input data, sc is the scaler object
```

Testing different ML algorithms

MLP Regressor

In [50]:
```
from sklearn.model_selection import GridSearchCV
import warnings
from sklearn.neural_network import MLPRegressor
```

Finding best parameters:

In [51]:
```
# mlp = MLPRegressor(max_iter=200)
# parameter_space = {
# 'hidden_layer_sizes': [(5), (10), (10,10)],
# 'activation': ['relu'],
# 'solver': ['sgd', 'lbfgs']
# }
# grid = GridSearchCV(mlp, parameter_space, cv=5)
# warnings.filterwarnings("ignore")
# grid.fit(X, y)
# # print the results
# print('Best final score:\n', grid.best_score_)
# print('Best parameters found:\n', grid.best_params_)
```

In [52]:
```
mlp = MLPRegressor(hidden_layer_sizes=(20, 20), solver='sgd', activation='relu', max_iter=200)
```

Ridge Regression

Finding best-performing alpha parameter

In [53]:
```
from sklearn.linear_model import Ridge
```

In [54]:
```
# ridreg = Ridge()
# parameter_space = {
# 'alpha': [.001, .002, .01, .025, .05, .1]
# }
# grid = GridSearchCV(ridreg, parameter_space, cv=5)
# warnings.filterwarnings("ignore")
# grid.fit(X, y)
# # print the results
# print('Best final score:\n', grid.best_score_)
# print('Best parameters found:\n', grid.best_params_)
```

In [55]:
```
ridreg = Ridge(alpha=0.1)
```

Random Forest

In [56]:
```
from sklearn.ensemble import RandomForestRegressor
```

In [57]:
```
# rand_forest = RandomForestRegressor()
# parameter_space = {
# 'n_estimators': [100, 200],
# 'max_features': ['auto', 'sqrt', 'log2'],
# 'max_depth' : [4,5,6,7],
# 'criterion' :['mse', 'mae']
# }
# grid = GridSearchCV(rand_forest, parameter_space, cv=5)
# warnings.filterwarnings("ignore")
# grid.fit(X, y)
# # print the results
# print('Best final score:\n', grid.best_score_)
# print('Best parameters found:\n', grid.best_params_)
```

In [58]:
```
rand_forest = RandomForestRegressor(n_estimators=200, max_features='sqrt', max_depth=4, criterion='mae')
```

Based on testing error results, we will go ahead with the MLP Regressor.

Function to automate a bit of the overhead in getting the appropriate X, y from dataset

In [59]:
```
# returns (X, y, sc), where sc is the feature scaling object
def get_X_y(input, other_labels, output_label):
  input_X = input.iloc[:,len(other_labels):]
  input_y = input[output_label]
  X_s, sc = feature_scaling(input_X)
  X = X_s
  y = input_y.values
  return X, y, sc
```

Intermediate models for fifa shooting, fifa passing, fifa defending, + Final model for goalies

In [60]:
```
# intermediate model for players (predicts shooting)
X, y, sc_shooting = get_X_y(sho, other_labels, 'shooting')
model_shooting = MLPRegressor(hidden_layer_sizes=(20, 20), solver='sgd', activation='relu', max_iter=200)
model_shooting.fit(X,y)
shooting_p = model_shooting.predict(X)

# intermediate model for players (predicts passing)
X, y, sc_passing = get_X_y(pas, other_labels, 'passing')
model_passing = MLPRegressor(hidden_layer_sizes=(20, 20), solver='sgd', activation='relu', max_iter=200)
model_passing.fit(X,y)
passing_p = model_passing.predict(X)

# intermediate model for players (predicts defending)
X, y, sc_defending = get_X_y(dfn, other_labels, 'defending')
model_defending = MLPRegressor(hidden_layer_sizes=(20, 20), solver='sgd', activation='relu', max_iter=200)
model_defending.fit(X,y)
defending_p = model_defending.predict(X)

# final model for goalies (predicts overall)
X, y, sc_goalies = get_X_y(gol, other_labels_gol, 'overall')
model_goalies = MLPRegressor(hidden_layer_sizes=(20, 20), solver='sgd', activation='relu', max_iter=200)
model_goalies.fit(X,y)
gol_overall_p = model_goalies.predict(X)
```

Results on real players:

In [61]:
```
results = df.copy()
results = results[other_labels]
results['shooting_p'] = shooting_p
results['passing_p'] = passing_p
results['defending_p'] = defending_p
results['FW'] = df['FW']
results['MF'] = df['MF']
results['DF'] = df['DF']
results.head()
```

Out[61]:

	long_name	overall	shooting	passing	defending	shooting_p	passing_p	defending_p	FW	MF	DF
0	Lionel Messi	93	92.0	91.0	38.0	99.137569	91.118253	50.767958	1	1	0
1	Cristiano Ronaldo	92	93.0	81.0	35.0	95.085076	77.224841	46.795830	1	0	0
2	Robert Lewandowski	91	91.0	78.0	43.0	92.696843	73.996729	40.326543	1	0	0
3	Neymar	91	85.0	86.0	36.0	87.643365	89.938154	45.109066	1	1	0
4	Kevin De Bruyne	91	86.0	93.0	64.0	87.164279	89.465329	62.868394	0	1	0

In [62]:
```
# the following code will save these results to csv:
#results_sorted = results.sort_values('prediction', ascending=False)[['long_name', 'overall', 'prediction']]
#results_sorted.to_csv('sho_results_sorted.csv', encoding='latin1')
```

Results on real goalies:

In [63]:
```
gol_results = df_g.copy()
gol_results['overall_p'] = gol_overall_p
gol_results
#gol_results.to_csv('/content/drive/MyDrive/School/SoccerData/RealGoalieData_Results.csv', encoding='latin1') # save re
```

Out[63]:

	long_name	age	club_name	league_name	overall	Salary	90s	GA90	SoTA/90	Saves/90	Save%	CS%	overall_p
0	Jan Oblak	27	Atlético Madrid	Spain Primera Division	91	6500000	37.7	0.72	2.785146	2.175066	77.1	44.7	88.231794
1	Marc-André ter Stegen	28	FC Barcelona	Spain Primera Division	90	13520000	36.0	1.00	3.055556	2.166667	72.7	38.9	89.636573
2	Alisson	27	Liverpool	English Premier League	90	8320000	28.3	0.81	2.791519	2.049470	72.2	44.8	87.769431
3	Thibaut Courtois	28	Real Madrid	Spain Primera Division	89	13000000	34.0	0.59	2.794118	2.176471	78.9	52.9	90.750628
4	Manuel Neuer	34	FC Bayern München	German 1. Bundesliga	89	6500000	33.0	0.94	3.363636	2.545455	74.8	45.5	88.313370
...	...	...	...	...	...	...	...	...	...	...	...	...	...
100	Baptiste Reynet	29	Nîmes Olympique	French Ligue 1	75	936000	22.6	2.17	4.955752	2.964602	58.9	9.1	76.848795
101	Angus Gunn	24	Southampton	English Premier League	74	1352000	10.0	2.50	5.300000	3.000000	56.6	20.0	75.749117
102	Andreas Luthe	33	1. FC Union Berlin	German 1. Bundesliga	74	988000	10.0	1.20	3.500000	2.300000	65.7	30.0	74.821852
103	Pavao Pervan	32	VfL Wolfsburg	German 1. Bundesliga	74	1352000	8.0	1.00	4.500000	3.625000	80.6	37.5	76.931288
104	Ludovic Butelle	37	Angers SCO	French Ligue 1	73	468000	26.0	1.23	3.000000	1.923077	62.8	42.3	75.315312

105 rows × 13 columns

Final model for players (predicting overall)

fitting the model

In [64]:
```
model_overall = MLPRegressor(hidden_layer_sizes=5, solver='lbfgs', activation='relu', max_iter=200)
X, y, sc_overall = get_X_y(ove, other_labels, 'overall')
model_overall.fit(X,y)
```

```
Out[64]: MLPRegressor(hidden_layer_sizes=5, solver='lbfgs')
```

Predicting based on generated stats from previous models

```
In [65]: X, y, m = get_X_y(results, other_labels, 'overall')
```

```
overall_p = model_overall.predict(X)
```

```
In [66]: results['overall_p'] = overall_p
```

```
In [67]: results
```

Out[67]:

	long_name	overall	shooting	passing	defending	shooting_p	passing_p	defending_p	FW	MF	DF	overall_p
0	Lionel Messi	93	92.0	91.0	38.0	99.137569	91.118253	50.767958	1	1	0	100.116498
1	Cristiano Ronaldo	92	93.0	81.0	35.0	95.085076	77.224841	46.795830	1	0	0	93.232249
2	Robert Lewandowski	91	91.0	78.0	43.0	92.696843	73.996729	40.326543	1	0	0	90.700682
3	Neymar	91	85.0	86.0	36.0	87.643365	89.938154	45.109066	1	1	0	93.629728
4	Kevin De Bruyne	91	86.0	93.0	64.0	87.164279	89.465329	62.868394	0	1	0	92.259947
...	...	...	...	...	...	...	...	...	...	...	...	...
1595	Tiago Djaló	68	41.0	57.0	67.0	40.200675	54.288621	64.993619	0	0	1	65.161892
1596	Mattias Svanberg	68	65.0	69.0	52.0	68.234625	66.050998	51.059256	0	1	0	72.747943
1597	Timothy Weah	68	68.0	59.0	30.0	57.322885	57.098555	27.949496	0	1	0	64.505196
1598	Ahmed Kutucu	68	69.0	55.0	41.0	68.547812	59.117182	41.253400	1	1	0	70.908425
1599	Andrea Pinamonti	68	70.0	45.0	22.0	70.118257	57.641644	37.073918	1	0	0	72.104105

1575 rows × 12 columns

Using new (fake player) data with these trained models

```
In [68]: file = '/content/drive/MyDrive/School/SoccerData/ABCDE.csv'
file_goalies = '/content/drive/MyDrive/School/SoccerData/GK_ABCDE.csv'
df = pd.read_csv(file, encoding='latin1')
df_g = pd.read_csv(file_goalies, encoding='latin1')
df_g.head()
```

Out[68]:

	Player	Nation	Squad	age	90s	GA90	SoTA/90	Saves/90	Save%	CS%	League	Year	Salary
0	F. Dauda	Nganion	Fanatical Outlaws	33	7.62	2.27	6.62	4.80	68.69	0.10	A	2020	18470000
1	S. Kiconco	Nkasland Cronestan	Fanatical Outlaws	34	23.47	1.42	4.26	2.97	68.39	20.06	A	2020	16720000
2	Z. Marini	Central Republic of Boekrainego	Fighting Cougars	27	38.05	1.48	4.32	3.08	68.33	23.63	A	2020	22680000
3	P. Kabugo	People's Land of Maneau	Great Galactic Gorgons	22	36.05	0.84	3.52	2.72	75.22	36.17	A	2020	21600000
4	G. Mwebaze	Nganion	Green Fleet	28	37.94	0.96	3.22	2.46	73.54	34.16	A	2020	17220000

```
In [69]: sho = df[sho_stats_labels].copy()
pas = df[pas_stats_labels].copy()
dfn = df[dfn_stats_labels].copy()
gol = df_g[gol_stats_labels].copy()
```

```
In [70]: # Use same feature scaling model used in training before inputting data
sho_s = sc_shooting.transform(sho)
pas_s = sc_passing.transform(pas)
dfn_s = sc_defending.transform(dfn)
gol_s = sc_goalies.transform(gol)

# outfield players
shooting_p = model_shooting.predict(sho_s)
passing_p = model_shooting.predict(pas_s)
defending_p = model_defending.predict(dfn_s)
close = pd.DataFrame()
close['shooting_p'] = shooting_p
close['passing_p'] = passing_p
close['defending_p'] = defending_p
fw = np.asarray(df['FW'])
close['FW'] = np.asarray(df['FW'])
close['MF'] = np.asarray(df['MF'])
close['DF'] = np.asarray(df['DF'])
close[close.isna().any(axis=1)]
close_s = sc_overall.transform(close)
overall_p = model_overall.predict(close_s)
df['shooting_p'] = shooting_p
df['passing_p'] = passing_p
df['defending_p'] = defending_p
df['overall_p'] = overall_p
```

```python
# goalies
gol_overall_p = model_goalies.predict(gol_s)
df_g['overall_p'] = gol_overall_p
```

Previewing the results:

In [71]:
```python
df.head()
```

Out[71]:

	Player	Nation	MF	DF	FW	Squad	age	Born	90s	Gls	...	Def 3rd	Press_Succ_%	Def 3rd_2	Int	Salary	League	shooting_
0	I. Winter	Danan Seekeeling	0	1	0	Fanatical Outlaws	27	1991	13.00	0.07	...	1.01	34.47	7.02	1.09	25480000	A	78.22427
1	P. Nakubulwa	Dosqaly	0	1	0	Fanatical Outlaws	22	1997	30.77	0.14	...	0.80	34.51	4.99	1.11	9950000	A	63.47091
2	M. Mahlangu	Imeer Vircoand	0	1	0	Fanatical Outlaws	34	1985	6.53	0.04	...	1.34	28.90	8.03	1.29	26500000	A	74.78786
3	I. Huber	Lenia Gerdanho	0	1	0	Fanatical Outlaws	25	1993	10.49	0.00	...	2.61	29.11	12.32	1.18	25530000	A	74.81830
4	A. Kabusingye	People's Land of Maneau	0	1	0	Fanatical Outlaws	18	2000	4.93	0.03	...	3.28	30.41	16.19	0.46	9510000	A	64.83013

5 rows × 35 columns

In [72]:
```python
df_g.head()
```

Out[72]:

	Player	Nation	Squad	age	90s	GA90	SoTA/90	Saves/90	Save%	CS%	League	Year	Salary	overall_p
0	F. Dauda	Nganion	Fanatical Outlaws	33	7.62	2.27	6.62	4.80	68.69	0.10	A	2020	18470000	103.898006
1	S. Kiconco	Nkasland Cronestan	Fanatical Outlaws	34	23.47	1.42	4.26	2.97	68.39	20.06	A	2020	16720000	94.878902
2	Z. Marini	Central Republic of Boekrainego	Fighting Cougars	27	38.05	1.48	4.32	3.08	68.33	23.63	A	2020	22680000	100.568048
3	P. Kabuqa	People's Land of Maneau	Great Galactic Gorgons	22	36.05	0.84	3.52	2.72	75.22	36.17	A	2020	21600000	96.419112
4	G. Mwebaze	Nganion	Green Fleet	28	37.94	0.96	3.22	2.46	73.54	34.16	A	2020	17220000	92.350356

Save these final predicted stats for the fake players to csv

In [73]:
```python
# outfield players
#df.to_csv('/content/drive/MyDrive/School/SoccerData/ABCDE_results.csv', encoding='latin1')
# goalies
#df_g.to_csv('/content/drive/MyDrive/School/SoccerData/GK_ABCDE_results.csv', encoding='latin1')
```

Works Consulted

Alex. (2021, September). FIFA 21 players & teams full database / UPD 09.21. Kaggle. https://www.kaggle.com/datasets/cashncarry/fifa-21-players-teams-full-database

Chilean men's National Team Sponsors. Chilean Men's National Team Sponsors. (2021, June 14)., https://sportskhabri.com/chilean-mens-national-team-sponsors/

Organisation for Economic Co-operation and Development [OECD]. Designing economic zones For Effective Investment (2010, February 16), https://www.oecd.org/mena/competitiveness/44866506.pdf

Fifield, D. (2018, May 17). Gareth Southgate defends picking young players in World Cup squad. *The Irish Times*.

https://www.irishtimes.com/sport/soccer/international/gareth-southgate-defends-picking-young-players-in-world-cup-squad-1.3499492

Fizia, C. (2018, June 14). National Team sponsors for the 2018 FIFA World Cup: Coca-Cola and Adidas lead the charge.

https://www.sportcal.com/Insight/features/118618

Football news - top stories, Videos & Results. Eurosport. (n.d.). Retrieved March 24, 2022, from https://www.eurosport.com/football/

Football statistics and history. Football Stats and History Statistics, scores and history for 100+ men's and women's club and national team competitions.

https://fbref.com/en/

German FA says Nike offers deal amid row with Adidas. (2007, January 30). *Times of Malta*

https://timesofmalta.com/articles/view/german-fa-says-nike-offers-deal-amid-row-with-addas.27972

Guillaume Vandenbroucke, G., & Peake, M. (2021, December 9). Poor countries see better education, not income: St. Louis Fed. *Saint Louis Fed Eagle*.

https://www.stlouisfed.org/publications/regional-economist/second-quarter-2020/poor-co untries-catching-rich-education-not-income

Ireland, J. (2022, January 6). Being a utility player in soccer is awesome: Here's 7 reasons why. Expand Your Game. https://expandyourgame.com/utility-player-in-soccer/

Leone, S. (2020, October 8). FIFA 21 complete player dataset. https://www.kaggle.com/datasets/stefanoleone992/fifa-21-complete-player-dataset

Madgavkar, A., Seong, J., & Woetzel, J. (2021, June 23). How governments in emerging economies can help boost and sustain growth. McKinsey & Company. https://www.mckinsey.com/industries/public-and-social-sector/our-insights/how-governments-in-emerging-economies-can-help-boost-and-sustain-growth

Nicolau, J. (2018, June 28). Does winning the FIFA World Cup boost the national economy? Howard Feiertag Department of Hospitality and Tourism Management | Virginia Tech. https://htm.pamplin.vt.edu/news/2018/06/pamplin-worldcup.html

Published by Statista Research Department, & 3, F. (2022, February 3). *Travel and tourism: Share of global GDP 2020*. statista . https://www.statista.com/statistics/1099933/travel-and-tourism-share-of-gdp/

Skoutelas, P. P. (n.d.). Sponsored content: Is a lack of investment in public transit infrastructure failing our economy?POLITICO. https://www.politico.com/sponsor-content/2018/06/when-public-transit

Which league and clubs have the most players at the 2014 World Cup? Which league and clubs have the most players at the 2014 World Cup? | Goal.com. (n.d.). Retrieved March 24, 2022, from https://www.goal.com/en/news/1717/editorial/2014/06/05/4861228/which-league-and clubs-have-the-most-players-at-the-2014#:~:text=Discover-,Which%20league%20and%20clubs%20have%20the,at%20the%202014%20World%20Cup%3F&text=The%20World%2

0Cup%20sees%20football,Brazil%20than%20any%20other%20club