Mathematically Fixing NVSL Divisions

Ben Brandt

Introduction

The Northern Virginia Swimming League (NVSL) is the largest summer swim league in the USA boasting tens of thousands of athletes over the span of 102 teams scattered throughout three counties in Northern Virginia. The purpose of the NVSL is to teach children the fundamentals of swimming in a fun, nurturing, and competitive environment. While the NVSL has the goal of teaching the basics of swimming, it has also produced swimmers of all levels including Olympians and elite NCAA athletes. Many of these elite athletes’ careers started at the NVSL, learning the basics and developing their careers in their summer meets. The NVSL is a staple for all those who have come through it, fostering a strong community of swimmers and coaches spread throughout all Northern Virginia.

Because of how large the league is, it is impossible to have every team go against every team. Varying team sizes and skills makes the skill levels of teams imbalanced as some teams such as Tuckahoe have hundreds of swimmers and teams such as Edsall Park have roughly 30 swimmers’ total. Because of this, the league organizes itself into divisions somewhat like the way the NCAA organizes divisions. Every spring, a committee sends out a survey to all the team reps to get information about their expected team size, number of club swimmers, or any other major changes happening to the team. Given this information and the team’s performance from the previous season, the league organizes these 102 teams into 17 divisions with 6 teams in each division with 1 being the best and 17 being the worst.

Even with this effective system, there still can end up being errors with teams not being placed in the proper divisions. Teams such as Burke Station who were placed in Division 15, outscored the rest of their division by hundreds of points and teams such as Brandywine performing hundreds of points lower than their division. Clearly there are some errors with the way the league organizes divisions, making it less enjoyable for teams who were placed in the wrong division.

As such, it has become my personal project to make the divisions better. The NVSL website has a feature called Virtual Meet where the you can select two teams in the league along with the a list of meet dates. The Virtual Meet takes these two teams from a given meet date and simulates if the two teams went against each other based on the lineups from that given meet date. Using this feature, we can run every possible combination of teams and meet dates and attempt to rank them that way. The best way that I could come up with was assigning every team an ELO value similar to the way chess works and as teams go against each other, their ELO changes. When a team with a high ELO beats a team with a low ELO, not much changes but when a low ELO team beats a high ELO team, the changes are a lot more drastic. Running all these virtual meets gives a better idea of how the divisions should be organized based on teams ELOs.

The Code

In this folder, there are two different python files. The first file, NVSL\_Scraper, is designed to run all the virtual meets and store them in a spreadsheet. The second file, NVSL\_ELO\_Graph, is used for the data analysis such as the calculations of the ELOs and making proper divisions.

The NVSL\_Scraper file utilizes several libraries including Selenium, time, random, and csv. There are also two csv files linked to this code. One is progress.csv and the other is results.csv both included in this folder. Most of the starting code in this file is accessing the dropdown menu and selecting the necessary teams and weeks. The code then actually runs the virtual meet and scrapes the score of the resultant meet and puts it into the results.csv file referenced earlier. It then goes back to the virtual meet selection page to run another simulation. To be able to successfully run this code without getting banned from the website, the code needs to sleep briefly after making inputs for a random amount of time. Because they are all uniformly distributed, we can make an estimate of how long the code takes. With our 102 teams all going against each other for weeks we are left with the following formula to estimate the total running time of the program.

This gives us our estimated number of seconds total run in the program. The .05 represents the average time it takes to scroll to each option, and the 7 represents all other load times added together. These are just averages based on a uniform distribution so the times might not be exactly accurate but gives a good estimate for general time. Solving the above equation gives a total of 3,238,245 seconds or roughly 37 days and 11.5 hours of straight running the program. Now of course this code can’t be run whilst working on your computer so it will realistically take a lot longer than 37 days. I believe it took me two months to fully run all the simulations. Because of the extended amount of time, I implemented two more functions: A save\_progress and load\_progress function. The save\_progress function saves progress after each virtual meet is run in the progress.csv file and the load\_progress function reads the current step indices in the progress.csv and loads it into the loop to keep the progress running. As for the full loop with all of the functions running, while designing it I elected to run it in the following order:

team1 week1 team1 week1 🡪 team1 week1 team1 week2 🡪 t1 w1 t1 w3… 🡪 t1 w2 t1 w1 … 🡪 t1 w1 t2 w1 … 🡪 t2 w1 t1 w1. So, we have all the week combinations run for each team matchup before moving to the next team. Since there are 5 weeks for each team that leaves us with each team matching up 25 times. After all 25 meets are run, the second team updates and is now a new team with a new 25-week combinations. This process is repeated for all 102 teams giving us 102\*25 or 2,520 total meets run for one team. Repeat this process for all 102 teams and we are left with 102\*2,520 or 260,100 total virtual meets totaling 20 mB of data. Due to some early bugs in the code, there were 12 extra virtual meets run at some point which is why there are a few extra rows in the csv file. There also is the issue of the fact that a team is going against themselves on a number of occasions but those are ignored in the ELO calculations.

The second python file, NVSL\_ELO\_Graph utilizes the libraries matplotlib, numpy, pandas, mplcursors, and defaultdict. Because ELO is based on change over time, it isn’t the most accurate measure of true skill but rather improvement. But there are enough total virtual meets being run that hopefully it shouldn’t matter. Nonetheless, a shuffle function was added that takes the csv and shuffles it in its entirety to give us a semi-randomized ordering hopefully rendering any biases that exist useless. The next few functions in the code are responsible for parsing the score and assigning a winner and loser to each virtual meet. Every team was given an initial ELO of 1600 and changes are made based on the winner in conjunction with both teams expected outcomes based on their current ELO. As such, the following equation was used to calculate ELO changes:

for the expected ELO of team A and for the expected ELO of team B. If team A wins, they are assigned a value called actual of 1 and B is assigned 0, same goes for B, and in the event of a tie, both teams are assigned 0.5. Using these values and the expected values above the ELOs are calculated as follow:

where k is an arbitrary value used as a scale of separation. Essentially meaning if two teams have the same ELO, after the virtual meet, their values will both change by k. Again, because ELO is based on change over time and as such the order in which the virtual meets are simulated can have an effect on the results, part of this code not only does the ELO calculations once but rather does them 100 times totaling 26 million ELO calculations. Each iteration is stored on its own and then averaged together for the final results. The second half of the code after running all the ELO calculations is the analysis of all the data including a scatterplot of all the teams, a dictionary of a teams ranking, a dictionary of the teams ELOs, and a proper organization of the divisions.

Analysis and Findings of the Data

The second half of the NVSL\_ELO\_Graph file is all based on the analysis of the data that was obtained through the scraping and ELO calculations from earlier. Each team was given an expected ranking based on their placement last year. A team who places last in their division is by default better than the team who won the division one worse. Based on these expected rankings, we can determine if a team is overperforming or underperforming expectations. By using numpy and matplotlib together, a scatterplot can be obtained with a line of best fit provided comparing a teams ranking with their ELO as shown below:

A graph with red and green dots

AI-generated content may be incorrect.

A team with a green dot indicates they exceeded their expectations in that their actual ELO is higher than what it should be based off how they placed last year. From looking at the graph, there are very obviously some teams who exceed expectations by a lot and some teams who do not. Unfortunately, because of the large gap in skill between the top 3 teams and the surrounding teams, this creates a shift in the line of best fit where both ends of the extreme show significant over and underperformance respectively. This also explains why almost all teams ranking 20-40 are underperforming because teams at the top such as Chesterbrook and Tuckahoe bring up the line of best fit. There are multiple cases where a teams actual rank is better than their expected rank but are still red because of the top teams bringing up the expected ELO. Additionally, there are also many teams in middle rankings between roughly 20 and 60 that are all right about even with one another.

The goal of this project is to remedy some of those errors. Notice in the legend of the graph there is both an R-value and an R2-value. The R-value indicates the direction and strength of our linear relationship between -1 and 1. An R-value of -0.966 indicates that there is a very strong negative linear relationship between a team’s expected ranking and their ELO. Essentially this is saying as a team’s expected ranking goes up, their ELO goes down. There is also an R2-value in the legend which indicates the total percent of variability of ELO using the predictor of expected ranking. A value of 0.9332 is a good number given the circumstances and we know that most of the randomness in our data is accounted for. These two values being very high bode well for the NVSL because it means their current methodology is very thorough and well thought out. However, there are some teams, as seen in the graph who are far above or below expectations and this is where the leagues analysis suffers a bit and could benefit from this model.

Part of the analysis in the program is intended to highlight some of the teams who were most effected by the NVSLs division process. Now I am not privy to all the information the NVSL is so some of the information they receive might change what division they put a team in such as a new head coach or a large influx of swimmers. Nonetheless, I have outlined some teams that the NVSL did a poor job of evaluating and stuck them in the wrong division.

A graph with red and green dots

AI-generated content may be incorrect.

Looking at the and based on the code, I have identified a few teams who have either significantly overperformed or underperformed their expectations. The team circled in purple is Burke Station who had both the largest change in their expected ELO compared to their actual ELO as well as their expected rank compared to their actual rank. Burke Station managed to outperform their expected ELO by 520 and their actual rank was 21.15 places higher than where why should’ve been. The reason all the teams actual ranks are in decimals is because 100 iterations were run and as such, the actual rank is an average of all those 100 iterations hence the two decimal places. Now examining the two teams circled in blue who significantly underperformed their expectations. Firstly the one on the left is South Run who underperformed their ELO expectations by 427. The team circled on the right is Walden Glen who placed 21.03 places below where they were expected to place. Generally, though, most teams are significantly closer to their performances.

Additionally, the graph shows all the teams in relation to one another. There are also two dictionaries that are generated that show all the teams ELOs and their expected rankings.

Here are the true ratings of all the teams:

{'Chesterbrook': 1.26, 'Tuckahoe': 1.76, 'Overlee': 2.98, 'Highlands Swim': 4.0, 'Donaldson Run': 5.23, 'Kent Gardens': 5.77, 'Old Keene Mill': 7.0, 'Crosspointe': 8.26, 'Fair Oaks': 9.09, 'Langley': 10.43, 'Oakton': 10.87, 'Wakefield Chapel': 11.35, 'High Point Pool': 13.15, 'Mantua': 14.01, 'Hunt Valley': 14.87, 'Vienna Woods': 16.2, 'Little Rocky Run': 16.79, 'McLean': 18.08, 'Orange Hunt': 19.62, 'Little Hunting Park': 19.76, 'Hamlet': 20.89, 'Hiddenbrook': 21.84, 'Ravensworth Farm': 23.47, 'Vienna Aquatic': 23.76, 'Rolling Hills': 24.86, 'Lee Graham': 26.65, 'Fox Hunt': 27.11, 'Fairfax': 27.68, 'Pinecrest': 29.4, 'Cardinal Hill': 29.64, 'Parklawn': 30.83, 'Canterbury Woods': 32.66, 'Virginia Hills': 32.86, 'Springboard': 34.51, 'Dominion Hills': 34.73, 'Sleepy Hollow Rec': 35.89, 'Waynewood': 36.15, 'Cottontail': 39.27, 'Virginia Run': 39.76, 'Dunn Loring': 40.38, 'Mount Vernon Park': 40.88, 'Sleepy Hollow B & R': 41.68, 'Lakeview': 42.12, 'Poplar Heights': 42.77, 'Greenbriar': 45.52, 'Dowden Terrace': 45.77, 'Hunter Mill': 46.85, 'Fairfax Station': 47.85, 'South Run': 49.7, 'Camelot': 49.76, 'Kings Ridge': 50.9, 'Arlington Forest': 52.15, 'Lakevale Estates': 52.29, 'Villa Aquatic': 53.94, 'Shouse Village': 55.92, 'Truro': 56.18, 'Sully Station': 57.15, 'Forest Hollow': 57.57, 'Mansion House': 58.29, 'Daventry': 60.44, 'Hollin Meadows': 60.62, 'Country Club Hills': 62.2, 'Mosby Woods': 62.82, 'Burke Station': 63.85, 'Hayfield Farm': 65.61, 'Fairfax Club Estates': 66.99, 'Commonwealth': 67.55, 'Poplar Tree': 67.68, 'Parliament': 69.18, 'Somerset-Olde Creek': 70.15, 'Highland Park': 70.36, 'Rolling Valley': 72.4, 'Sideburn Run': 72.47, 'Stratford': 73.45, 'Riverside Gardens': 75.75, 'Holmes Run Acres': 76.06, 'Fox Mill Woods': 76.76, 'Lincolnia Park': 77.68, 'Great Falls': 79.43, 'Brookfield': 79.87, 'Walden Glen': 80.03, 'Rolling Forest': 81.59, 'Woodley': 82.95, 'Laurel Hill': 84.66, 'Fox Mill Estates': 84.97, 'Sully Station II': 85.37, 'Rutherford': 87.09, 'Hollin Hills': 88.49, 'Village West': 89.86, 'Brandywine': 90.44, 'Pleasant Valley': 91.03, 'Lake Braddock': 91.29, 'Herndon': 91.93, 'Newington Forest': 94.05, 'Annandale': 95.34, 'Long Branch': 95.63, 'Ilda Community': 96.97, 'North Springfield': 97.88, 'Broyhill Crest': 99.0, 'Springfield': 100.0, 'Edsall Park': 101.0, 'Pinewood Lake': 102.0}

There are some teams near the very top and bottom who end in a full number because of how consistently good or bad they are. For example, the bottom 4 teams are consistently getting 99th, 100th, 101st and 102nd without fail every time. Similarly, Highlands Swim ranked 4th in every single iteration. We can also get some insight into how closely two teams compare based on the rounding of their actual rankings. For example, a meet between Rolling Valley and Sideburn Run could be impossible to tell who is going to win as their actual rankings only differ by .07. On the other hand, Waynewood and Cottontail are right next to each other in the rankings but have a 3.12 actual rankings gap. This is partially why the ELO dictionary is also necessary as it allows us to see some variation that exists that might not be visible in the actual rankings dictionary.

Here are all the teams ELOS:

{'Chesterbrook': 3160.54, 'Tuckahoe': 3144.53, 'Overlee': 3102.45, 'Highlands Swim': 2967.45, 'Donaldson Run': 2902.89, 'Kent Gardens': 2885.09, 'Old Keene Mill': 2810.47, 'Crosspointe': 2716.72, 'Fair Oaks': 2692.69, 'Langley': 2669.62, 'Oakton': 2659.54, 'Wakefield Chapel': 2649.26, 'High Point Pool': 2529.77, 'Mantua': 2501.65, 'Hunt Valley': 2470.28, 'Vienna Woods': 2409.01, 'Little Rocky Run': 2385.4, 'McLean': 2304.35, 'Orange Hunt': 2253.49, 'Little Hunting Park': 2252.67, 'Hamlet': 2223.55, 'Hiddenbrook': 2205.59, 'Ravensworth Farm': 2158.4, 'Vienna Aquatic': 2153.37, 'Rolling Hills': 2128.02, 'Lee Graham': 2090.54, 'Fox Hunt': 2082.77, 'Fairfax': 2067.52, 'Pinecrest': 2038.99, 'Cardinal Hill': 2035.7, 'Parklawn': 2015.26, 'Canterbury Woods': 1986.16, 'Virginia Hills': 1979.71, 'Springboard': 1955.44, 'Dominion Hills': 1953.05, 'Sleepy Hollow Rec': 1936.21, 'Waynewood': 1932.82, 'Cottontail': 1886.49, 'Virginia Run': 1880.93, 'Dunn Loring': 1872.31, 'Mount Vernon Park': 1869.08, 'Sleepy Hollow B & R': 1858.99, 'Lakeview': 1852.16, 'Poplar Heights': 1844.84, 'Greenbriar': 1799.37, 'Dowden Terrace': 1794.38, 'Hunter Mill': 1768.93, 'Fairfax Station': 1746.44, 'South Run': 1705.61, 'Camelot': 1704.87, 'Kings Ridge': 1682.94, 'Arlington Forest': 1658.62, 'Lakevale Estates': 1656.78, 'Villa Aquatic': 1591.28, 'Shouse Village': 1483.95, 'Truro': 1475.49, 'Sully Station': 1462.81, 'Forest Hollow': 1451.05, 'Mansion House': 1441.02, 'Daventry': 1391.44, 'Hollin Meadows': 1382.68, 'Country Club Hills': 1343.94, 'Mosby Woods': 1325.84, 'Burke Station': 1290.35, 'Hayfield Farm': 1237.02, 'Fairfax Club Estates': 1211.37, 'Commonwealth': 1204.45, 'Poplar Tree': 1201.64, 'Parliament': 1183.2, 'Somerset-Olde Creek': 1171.99, 'Highland Park': 1169.62, 'Rolling Valley': 1143.54, 'Sideburn Run': 1142.67, 'Stratford': 1131.38, 'Riverside Gardens': 1098.23, 'Holmes Run Acres': 1093.79, 'Fox Mill Woods': 1082.83, 'Lincolnia Park': 1072.28, 'Great Falls': 1049.27, 'Brookfield': 1040.34, 'Walden Glen': 1038.68, 'Rolling Forest': 1013.03, 'Woodley': 966.97, 'Laurel Hill': 751.25, 'Fox Mill Estates': 741.19, 'Sully Station II': 730.7, 'Rutherford': 651.71, 'Hollin Hills': 612.99, 'Village West': 590.17, 'Brandywine': 583.31, 'Pleasant Valley': 575.48, 'Lake Braddock': 568.89, 'Herndon': 562.34, 'Newington Forest': 516.14, 'Annandale': 483.02, 'Long Branch': 476.21, 'Ilda Community': 442.03, 'North Springfield': 409.41, 'Broyhill Crest': 343.39, 'Springfield': 204.63, 'Edsall Park': 65.73, 'Pinewood Lake': 5.7}

Here provides slightly more insight into teams and their skills. Again looking at the top and bottom end, there are large gaps in skill where there is a 338 ELO gap between the 99th team and the 102nd team. Looking at the top end too, while the top 3 are close together, there is a 135 ELO gap between 3rd and 4th. The k-value and initial ELO mentioned earlier were chosen very strategically to ensure no team had a negative ELO and no team had a ridiculously high ELO. The ELOs all go to two decimal places for the same reason the actual rankings do: Because 100 iterations are run, it is simply an average of all the ELOs

Ideally if there weren’t specific numbers needed for each division, we could use a k-means cluster to make fair divisions that are all equal skill level. Because that is not the case however, we are forced to take each team average actual ranking and group them by 6 into 17 different divisions.

These are what the divisions should be for the upcoming season:

Division 1: Chesterbrook, Tuckahoe, Overlee, Highlands Swim, Donaldson Run, Kent Gardens

Division 2: Old Keene Mill, Crosspointe, Fair Oaks, Langley, Oakton, Wakefield Chapel

Division 3: High Point Pool, Mantua, Hunt Valley, Vienna Woods, Little Rocky Run, McLean

Division 4: Orange Hunt, Little Hunting Park, Hamlet, Hiddenbrook, Ravensworth Farm, Vienna Aquatic

Division 5: Rolling Hills, Lee Graham, Fox Hunt, Fairfax, Pinecrest, Cardinal Hill

Division 6: Parklawn, Canterbury Woods, Virginia Hills, Springboard, Dominion Hills, Sleepy Hollow Rec

Division 7: Waynewood, Cottontail, Virginia Run, Dunn Loring, Mount Vernon Park, Sleepy Hollow B & R

Division 8: Lakeview, Poplar Heights, Greenbriar, Dowden Terrace, Hunter Mill, Fairfax Station

Division 9: South Run, Camelot, Kings Ridge, Arlington Forest, Lakevale Estates, Villa Aquatic

Division 10: Shouse Village, Truro, Sully Station, Forest Hollow, Mansion House, Daventry

Division 11: Hollin Meadows, Country Club Hills, Mosby Woods, Burke Station, Hayfield Farm, Fairfax Club Estates

Division 12: Commonwealth, Poplar Tree, Parliament, Somerset-Olde Creek, Highland Park, Rolling Valley

Division 13: Sideburn Run, Stratford, Riverside Gardens, Holmes Run Acres, Fox Mill Woods, Lincolnia Park

Division 14: Great Falls, Brookfield, Walden Glen, Rolling Forest, Woodley, Laurel Hill

Division 15: Fox Mill Estates, Sully Station II, Rutherford, Hollin Hills, Village West, Brandywine

Division 16: Pleasant Valley, Lake Braddock, Herndon, Newington Forest, Annandale, Long Branch

Division 17: Ilda Community, North Springfield, Broyhill Crest, Springfield, Edsall Park, Pinewood Lake

The teams are in divisions based off their rankings that are calculated for their actual rankings. All the teams are highlighted based on whether or not the official divisions match up with the calculated predictions. Green indicates these calculations and the NVSL agree on a teams placement. Yellow indicates one division difference which in most cases is irrelevant. Red indicates a team who is 2+ divisions different between the calculations here and where the NVSL actually put them. Looking through the division list the first and last two divisions are fairly accurate this is because these are both ends of the extremes. This means that the gap is much larger and thus shifts around a lot less. Looking towards the middle however there are some clear errors in the methods between these calculations and the NVSL. Again, I do not have all the information the selection committee does so perhaps they know something I don’t. In division 4 we see Ravensworth Farm highlighted in red this is because the league put them in division 6 rather than 4. This is because last year they were in division 8 so the league likely didn’t think they should be moved up 4 divisions. Looking to division 6 we see Dominion Hills highlighted in red as the NVSL placed them in division 8. This is interesting because Dominion Hills placed last in division 7 last year and yet mathematically fall in division 6. Next in division 8 we have Poplar Heights and Greenbriar. Two teams who were placed in divisions 6 and 10 respectively. This goes to show some of the errors in the leagues ways. Greenbriar is one of the teams with the highest difference between expected ELO and actual ELO. We can see from here how the NVSL can throw a team in division 6 and another in 10 and in reality, they rank right next to one another. Looking to division 9 now with South Run who was placed in 7. South Run was the team with the worst comparison of expected ELO to actual ELO and the league underestimates how poor their performance was last season. In division 11, 3 teams are mislabeled. Hollin Meadows was placed into division 9, Burke Station was placed into 13, and Hayfield Farm was placed into 14. Very similarly to the situation with Greenbriar and Poplar Heights, we have 3 teams who are all right next to each other in skill and yet were placed 5 divisions apart from one another which should mean a dramatic gap in skill. Burke Station was the team that most outperformed expectations last season so putting them in division 13 rather than division 11 where they belong seems like a mistake. Moving to division 12, both Parliament and Somerset-Olde-Creek are very incorrectly placed and were put in division 14. So, both of these teams this season should have very impressive results in division 14 rather than in division 12 which is where they belong. Lastly, in division 14 we have Great Falls and Walden Glen which the league placed in 12 and 11 respectively. Walden Glen was the team with the largest drop in expected ranking compared to actual ranking so the league only moving them from 10 to 11 is a big mistake as they will likely have a bad season again. Great Falls is in a similar situation although not quite as extreme. Overall, however, the league generally does a good job of ensuring a fair, competitive meet a majority of the time.

Potential Errors and Improvements

With any project of this scale there are bound to be errors in the code as well as improvements that could be made. Admittedly, there was portions of this code that were generated by or improved by AI. While the core concepts and ideas of the code are originally my own, AI is responsible for a lot of the optimizations that occur and as such there are errors and inefficiencies in the code. In the NVSL\_Scraper, there can be some time adjustments made that make the code run slightly more optimally as well as some error detection in various places for debugging purposes. Additionally, figuring out a way to perhaps not run duplicate meets. For example, Annandale vs Broyhill Crest is the same as Broyhill Crest vs Annandale, so each meet is really run twice. While that is good for additional data, this could also be solved just by having more iterations in the code. Another issue with the code is that you can’t run other things on the computer while the code is running so figuring out a way to be able to work on the computer while the code is running would make it complete significantly faster. Moving on to the NVSL\_ELO\_Graph, many errors occurred while running both the score parser and the process results. The code is written in a way where if an there is an error in parsing the score, it is just ignored rather than generating an error. In the grand scheme, this isn’t a huge deal because of the sheer number of meets being run. Also a lot of the data analysis isn’t very well documented and can be difficult to understand for someone who didn’t write it. There could also be more done in terms of the analysis of the data such as getting the total changes in ELO and rankings for every team rather than the largest changes. As mentioned earlier, I am not privy to all the information that the selection committee is and as such potentially am not able to accurately assess a teams capabilities in the same way. There may also be other bugs or errors in the code that I am not aware of and can very easily be remedied.

Overall though, I think this project could give much more insight into the decisions the NVSL makes in regards to their division selection process and the results of this season will determine the validity of this project