Sports Strategy Analytics using **Machine Learning** & **Probabilistic Model Checking**

NUS National University of Singapore | School of Computing

NUS Master of Computing

Capstone Project Report

Prepared by: **AARON L H CHAN (A0117352M)**

Advised by: **Dr. JIN-SONG DONG**

December 2022

SPORTS STRATEGY ANALYTICS USING MACHINE LEARNING AND PROBABALISTIC MODEL CHECKING

by

AARON LOK HIN CHAN
(Bachelor of Commerce in Management Information Systems and Marketing, University of British Columbia)

A CAPSTONE PROJECT SUBMITTED FOR THE DEGREE OF

MASTER OF COMPUTING

in

COMPUTER SCIENCE

in the

GRADUATE DIVISION

of the

NATIONAL UNIVERSITY OF SINGAPORE

2022

Supervisor:
Dr. Jin-Song Dong

Declaration

I hereby declare that this project report is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in this report.

This report has also not been submitted for any degree in any university previously.

Aaron Lok Hin Chan, A0117352M

01 December 2022

Abstract

Tennis has consistently been ranked as one of the world's most popular sports, but efforts to push the sport into the new age of advanced data analytics have seemingly lagged far behind other popular sports, such as basketball, football, and baseball. Although many recent works have pushed the envelope in the right direction, lack of data availability for lower-tiered tennis players remains a major roadblock for generating accurate match outcome predictions. Building on top of previous research work done by Dong et al., our goal is to take the existing concepts of probabilistic communicating sequential processes (PCSP) and the process analysis toolkit (PAT) model checker, and expand its usage to lower-tiered tennis players through a concept we call data generalization by clustering. To test the effectiveness of our approach, we first cluster players based on play style using three distinct game components: service game, return game, and rally game. Next, we employ a 3-step iterative experimental process to select the most optimal feature sets and machine learning algorithms to use for player clustering. Lastly, we evaluate the cluster results using metrics such as intra-inter cluster distance, evaluation against a controlled feature set, and finally, player-cluster data replacement in the original PCSP model to determine how accurately our clustered representation resembles the players themselves.

Acknowledgements

I would like to acknowledge and give my special thanks to my supervisors and mentors, Dr. Jin-Song Dong and Michael Jiang Kan, for their relentless support and insights throughout all stages of this project. This project would not be possible without their advice and guidance.

Table of Contents

1. Introduction

In the cross-section of sports and analytics, enthusiasts often think of the likes of baseball's success story of Billy Beane's Oakland A's using data to pick out undervalued players to fit in a lineup [1]; basketball's Daryl Morey, former general manager of the Houston Rockets, revolutionizing the way we value the three-pointer in the offensive gameplan; or SAP's use of historical penalty shoot-out data in soccer to guide Germany to a championship in the 2014 World Cup [2]. With the steady growth in computing power and explosion of big data in the last decade, there is no surprise that stakeholders are extensively seeking better ways to utilize data to make crucial decisions in all facets of the game. This includes choosing which players to add to the roster to optimize a certain winning metric, predicting ticket sales from a revenue perspective, and pinpointing exact movements in a player's actions to dictate how a player should practice to improve and maximize success. In-game strategy is no exception. Having the ability to predict match outcomes and to then surgically make small improvements to a team's strategy has become the holy grail in the world of modern sports.

1.1. Background & Motivation

Over the years, tennis has evolved from a past-time of primarily the upper-class in the late 1800's to a sport that people of all ages and backgrounds love to play and watch fervently. Although recognized as one of the most popular sports, with about 87 million people playing tennis worldwide in 2021, it is still somehow lagging behind its peers in its use of data [3]. For instance, the idea of recording the length of rallies, and then bucketing them into groups such as 0-4, 5-8, and 9+ shots, was done by IBM only as recently as 2015 [4]. In recent years, various attempts have been made to jumpstart the data revolution among tennis enthusiasts and data practitioners. Efforts include using linear regression, neural networks and deep learning techniques to predict match outcomes of Wimbledon in 2019 [5], as well as using Markov chain models to predict match winners [6].

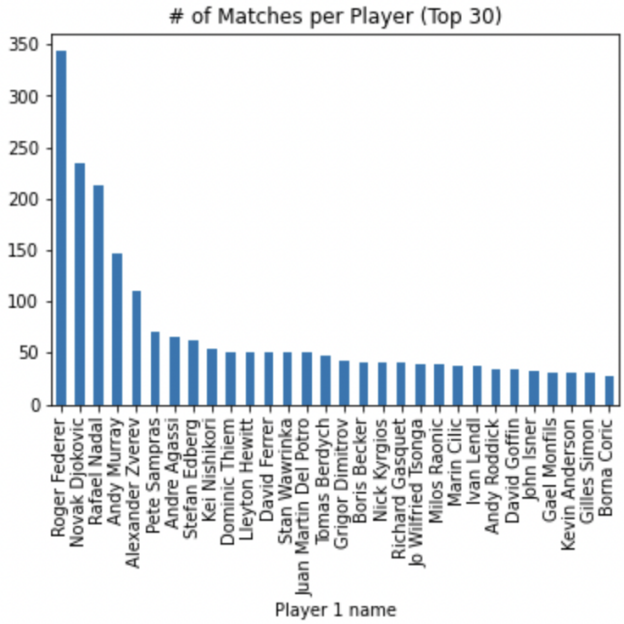
Regardless of how advanced machine learning techniques have become, the bottleneck for accuracy seems predicated on the quality and intricacy of the data itself, since most recent attempts have resorted to using "high-level data" to be used as model features. In this paper, "high-level data" refers to a category of data that is summarized from a match-level or higher, such as player win-loss records, ATP rankings, rating systems, head-to-head records, and match

scores. Although these attempts are unquestionably useful in its own right, they still leave a large gap from a more granular, strategic standpoint. Firstly, high-level data often does not show the full picture of what is really happening inside of a match, let alone inside a single set or a single point. For example, simply looking at win-loss ratios by themselves will neglect any luck factor that undoubtedly comes into play in any sport, where tennis is no exception. They will also not tell us what aspects of the game the player excelled in or struggled in, in order for them to focus in on a particular skill or tactic given a specific situation. Furthermore, high-level data tend to neglect specific play styles that may be advantageous or disadvantageous in certain player matchups. Lastly, traditional techniques also do not have the ability to dissect parts of the game at a more granular level. For example, if we wanted to analyze specific sequential data such as shot type combinations that may be especially effective from one player towards another, high-level data will not be able to provide this level of insight. The ability to make any useful inferences from analytics declines significantly from a decrease of available data on players of interest. In short, past attempts have largely fallen short of the type of in-depth analysis we aspire to achieve in today's game.

One of the main reasons is simply a lack of open-sourced data; there is a wide gap of data that is available to the public which can be used for more in-depth analysis. One of the relatively recent advances in sports technology include Hawk-Eye, a vision-based ball-tracking system used to improve umpiring decisions and provide interactive visuals for the audience [7]. But even with cutting-edge technology like Hawk-Eye, tennis organizers have refused to open source the collected data to the public. As a result, a treasure trove of accurate spatial-temporal tennis data has been locked up from the general public, and unable to be made use of by enthusiasts of both sports and data.

One bright spot has been Jeff Sackmann's Tennis-Abstract website, which may be single-handedly pushing forward use of data in tennis to the public [8]. Over the years, Tennis-Abstract has provided crowd-sourced data for thousands of matches, which includes data for over 8,000 matches recorded from as early as 1970 to today. The site provides high-level data such as rankings, Elo ratings and head-to-head records, and also low-level data such as shot-level data, which can all be filtered and dissected in a variety of different ways directly using the tools built on top of the website. However, as they are the sole provider of this data, there are still challenges in getting all of it to mainstream consumption. First, the site relies heavily on volunteers to keep the site running with current match statistics, which may often result in missing data points and

inconsistencies. Also, since the collection of data is sourced from volunteer's efforts, the availability of certain game's data depends entirely on the popularity of the match-up, which results in a heavy skew of data. To illustrate this point, as of 2019, out of the total 885 players that have been recorded on the site, Roger Federer's matches account for about 4.2% of the entire dataset, while the big three of Roger Federer, Rafael Nadal, and Novak Djokovic take up about 10% of the dataset (cite - my analysis from tennis-abstract). Aside from the apparent lack of availability of certain matches, the low-level data is typically collected in a very raw format, which means users need to have the technical know-how and knowledge of Tennis-Abstract's specific formatting standards to extract, process, and convert it to meaningful insights. Lastly, as mentioned, the quantity and quality of data greatly depends on the popularity of the players of interest. As such, players ranked outside the top 50 currently will have less than 17 matches recorded for them on this site, which means any players ranked lower will have even less data available. The quality of insights data analytics can produce drastically drops as data on matches decreases. For many low-tier players, there are simply not enough data points to drill down to the level of detail that is needed for useful strategic insights.



This chart was produced using data extracted from Tennis-Abstract. We can see Federer sits atop with almost 350 matches recorded, while the 30th top-recorded player, Borna Coric, has 28 matches recorded. Additionally, even with the wealth of data available for top players on this site, there is still a lack of certain spatial and temporal characteristics of ball and player movement. This includes metrics such as ball speed, ball rotation direction, ball rotation speed, player position, player speed and more. Although a site like Tennis-Abstract at times

provides approximate location of players during a shot, or approximate location of ball bounce location, it is often mis-labeled, or simply not at the pin-point level needed to base decisions off of. This type of data could potentially be gained from a computer-vision driven program such as Hawk-Eye, or a custom-made program, which requires extremely high technical proficiency in machine learning and data manipulation to develop. Ultimately, this type of granular data is crucial for the type of assessment desired by coaches and players, especially for a sport that focuses heavily on player positioning and ball speed and rotation.

Consequently, without this information readily available for both high and low-tier players, predictive analysis and strategy formulation faces many challenges to meet ideal expectations in a variety of fields such as strategy coaching and performance assessments, which will in turn make an impact on how a player should prepare for their next match.

1.2. Process Analysis Toolkit (PAT) & PCSP Module

The Process Analysis Toolkit (PAT), developed by Dong et al., is the model checker software program used to create simulations and verifications from scenarios in concurrent systems. It is written in the PCSP language, which is a superset of the CSP language. CSP, which stands for "communicating sequential processes", is a formal modeling language used to describe interactions in concurrent systems [9]. To adapt to the landscape of predicting tennis match outcomes, Dong et al. have created the probabilistic-CSP module (PCSP), which is an extension of CSP that adds a probabilistic element to it [10]. In essence, it combines CSP with the essential element of non-deterministic outcomes that is needed to properly model a game like tennis. An example for this is the probability of directions in which an opponent may return a ball, which can be denoted as “DownMid”, “DownLine”, and “CrossCourt” in the existing PCSP model.

```
Ply1_de_stroke = pcase{
    13: FH_Crosscourt { ball = 6} -> Ply2_de_stroke
    13: FH_Downline { ball = 4} -> Ply2_ad_stroke
    13: FH_DownMid { ball = 5} -> Ply2_mid_stroke
    13: BH_InsideIn {ball = 4} -> Ply2_ad_stroke
    12: BH_InsideOut { ball = 6} -> Ply2_de_stroke
    12: BH_DownMid { ball = 5} -> Ply2_mid_stroke
```

Essentially, Dong et al. have created a novel approach that uses probabilistic communicating sequential processes (PCSP) and the probabilistic model checker (PAT) system to model and calculate the probability of specific scenarios occurring in a tennis match, such as the chance of a specific player winning against another player. The approach adopts Markov Decision Process (MDP) techniques and PAT’s verification capabilities to calculate shot type

distribution probability by either player at any point of a match. By gathering data on the successful probabilities of each type of shot performed by a player under a given situation, and using probabilistic reasoning techniques, PCSP and PAT present a much more comprehensive and novel way to analyze and predict match outcomes compared with traditional attempts in the field of sports analytics.

1.3. Objective

This paper builds on top of the work done by Dong et al. as we aim to uncover the patterns and generalizations in habits, shot preferences, and other unique strengths and weaknesses of players based purely on the same underlying set of data that was aggregated online and used in the PCSP study. While our overarching goal is similar to the study of Dong et al's., we will tackle the problem from another angle. The area of focus in this paper will be on clustering tennis players based on play styles, which can be broken down further into individual game components, which are service game, return game, and rally game.

There are also several synergies that can be taken from our goal of clustering players, which can be used by Dong et al.'s original study.

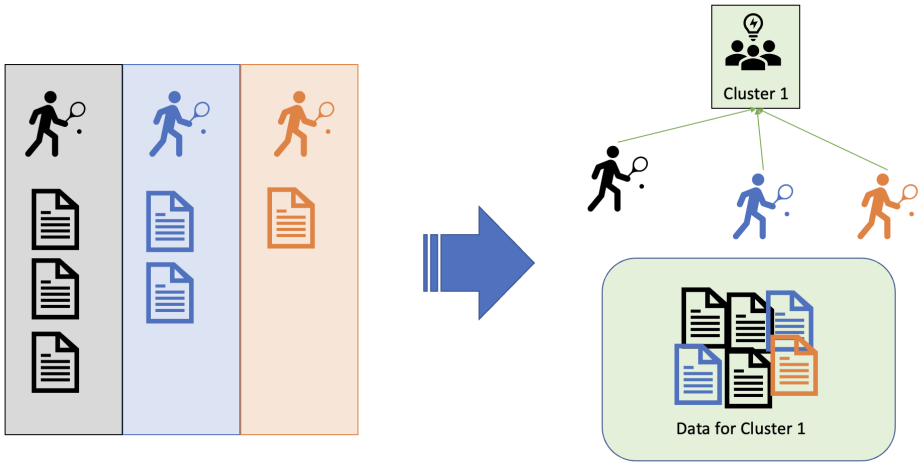
1. **Generate Cluster Labels:**

   The ability to attribute an accurate cluster label to a player has many potential benefits. Cluster labels obtained from this study can act as an additional feature to be fed into the original PCSP model to give even more accurate predictions which can be tailored to playstyle.

2. **Increase Data Access for Players through Cluster Generalization:**

   Data availability is an obvious roadblock when it comes to modeling players that are outside the top 50 players in the world. Without a vast amount of data, there is simply no easy way to uncover useful inferences and generate prediction results. On top of this, we often need historical match data that was played between two specific players, which is much more rare than data on one player alone. This is also the approach that the PCSP model takes, and in effect, creates a significant data barrier for low-tier players to use this technique. To counter this, there are two ways to do this: simply gather as much raw data as we can for each sets of players of interest in a brute-force manner; or find a way to generalize a player into a player type (or cluster), granting access to all the data from the player type which can be used to describe the player himself. Evidently, the second

approach requires significantly less data to do than the first type. To do this however, we will need to cluster a player into a certain player profile based on a number of factors like shot type preference and other features. Granted, player clustering also requires a not insignificant amount of data to produce useful results, the goal of clustering a player into a cluster type means that player can have access to all the data related to the entire cluster, which may include 10's to hundreds of other players, depending on the scale of the dataset. The challenge is to fit players into distinct cluster types that are different enough to each other but also form a good representation of the individual players' style as well.



3. **Evaluation using PCSP:**

To evaluate the results of our player clustering, we will input the clustered data back into the PCSP model to replace individual players' data, in order to gauge the "closeness" of results when running queries such as winning probability in a tie-break game, which is used in the original PCSP study.

As described in the second point above, the main goal of this paper is to create a technique to help make the PCSP approach accessible to even the lowest-level tennis players, by granting them access to a treasure trove of data associated with the player's particular cluster. As with the study by Dong et al., real-world applications of the PCSP model are tremendous in the world of sports. With the ability to accurately predict not only winning probability against an opponent, but also predict next shot type and location means we can analyze a player's performance on a much more granular level. Improvements can be made in fields such as sports betting, or high performance sport coaching, where the coach can pinpoint the exact shot type the player needs

to improve in order to gain a certain percentage advantage over another player, which can be quantified through data. This extends to individual practice sessions as well. As an example, from Dong et al.'s original study, it was shown that based on PCSP's analysis, if Federer improved his "backhand down the line" stroke by 2%, and reduced his error rates on ad-court by the same amount, he would improve his winning probability against Rafael Nadal to a range of 32-38.6%, from a 29.9-36% range [11].

1.4. Approach

1.4.1. Game Components

In order to capture a player's playstyle in an encapsulated and generalized way, there are a number of key factors that should be considered. First, we need to decide how many and what game components we wish to emulate. A "game component", can be thought of as a "cluster type", which correlates to a particular sub-component of the whole game of tennis. For example, Player A may hit "rally" shots just like Player B, but may have a completely different "serve" shot style, which may be similar to Player C's. If we simply group Player A into the same general cluster as Player B, we will neglect that Player A has a similar serve style as Player C, which will not be a good representation of overall playstyle. Hence, we should cluster each player multiple times, with each time relating to a different game component.

For this, there are a few considerations. We cannot split the clusters into too many subcategories, as we need sufficient data in every component. On the other hand, the clusters cannot be too general, as this may not be able to represent players well enough to make useful inferences. As mentioned, having only one game component (eg. overall playstyle) may not be sufficient to capture nuances in a player's overall style against another's. Lastly, the abstractions of each subcomponent need to be largely orthogonal or mutually exclusive from other subcomponents in order to justify having separate game components. Basically, we should break down game components into small enough components that can fairly represent different types of game scenarios, but large enough to be easy to generalize among many players.

Tennis can be described as a relatively "repetitive" type of game, in the sense that players will only ever be in certain areas of the court, and that there is a known predictable sequence to the game which must be followed to be in line with the rules. To breakdown the game into game

components that satisfy the above factors, we can break the game down into the following three game components:

1. **Service Game:** this component is related to the serve shots (first and second). Every point is started with a serve shot from the player who acts as the "server" for that tennis game. Players then alternate to serve between each game. A player that missed their first serve gets a second serve right afterwards. Both of these shots are captured in this game component.
2. **Return Game:** this component is related to the first return shot after the opponent has served. This can be thought of as the first shot of the point by the opponent player.
3. **Rally Game:** this component is related to any shot after the return shot, or the third shot and after.

Sequentially, every point will start with a **serve shot → return shot → rally shot → rally shot…**. In other words, every shot after the second shot (return shot) onwards is included in the "rally" game component. It is important to note that these three game components also emulate the exact three game components used in the original PCSP model by Dong et al., which is convenient for our final evaluation since we will need to substitute individual player's data with its assigned cluster's data inside each PCSP function.

```
De_Ply1Serve = pcase {                              // all probability is base
              231: ServeT_in{ball= 6} -> Ply2_BackHandR  // T will have opponer
              119: ServeT_err{ball=9} -> De_Ply1Serve_2nd   // Federer tries to
              337: ServeWide_in{ball =6} -> Ply2_ForeHandR
              173: ServeWide_err{ball=9} -> De_Ply1Serve_2nd
              92: ServeBody_in{ball=6} -> (Ply2_BackHandR [] Ply2_ForeHandR)
              48: ServeBody_err{ball=9} -> De_Ply1Serve_2nd};
```

PCSP function for the 1st serve sub-component

1.4.2. Feature Selection

Next, we need to decide on the features to use to represent each game component. This is arguably the most difficult step, as the quality of cluster groups depend almost entirely on the quality of data that is fed in. For feature selection, we take on a multi-pronged approach that each has its own pro's and con's, and will be evaluated individually at the end. These approaches include using:

1. **High-level Summary data from Tennis-Abstract website:**

These are high-level summary data for each player that is already organized for us by the Tennis-Abstract website creators. For the Service game component, examples of features include “Unreturned %”, “Service impact”, and “Percent of points won when return was put in play”. The complete data set can be found in the Appendices. The advantages that come with this is the ease of extractability and interpretability of summary metrics.

2. **Low-level Data that Emulates PCSP Functions:**

This is data transformed from lower-level, also from Tennis-Abstract, and emulates the features used in the original PCSP model. For example, for the “De_Ply1Serve” function shown in the below PCSP function, there are 231 occurrences where the shot fell into the category called “`ServeT_in{ball = 6}`”. In simple terms, this means that during this player’s 1st serve from the deuce court side, there are 231 times where he served towards the T (middle line), and it was served successfully inbounds. The following action is “Ply2_BackHandR”, but in the PCSP model, this is the only possible occurrence in its chosen abstraction. The features used for clustering will be the probability distribution of each of these low-level actions.

```
De_Ply1Serve = pcase {                              // all probability is base
              231: ServeT_in{ball= 6} -> Ply2_BackHandR  // T will have opponer
              119: ServeT_err{ball=9} -> De_Ply1Serve_2nd   // Federer tries t
              337: ServeWide_in{ball =6} -> Ply2_ForeHandR
              173: ServeWide_err{ball=9} -> De_Ply1Serve_2nd
              92: ServeBody_in{ball=6} -> (Ply2_BackHandR [] Ply2_ForeHandR)
              48: ServeBody_err{ball=9} -> De_Ply1Serve_2nd};
```

3. **Experimentation with Feature Engineering:**

Building off of the previous two feature selection methods, we can experiment with using only sub-fields or conditional data, which can be based on more human-understandable patterns and heuristics. For example, if we feel that each player’s “return game” cluster can be fully based off on the player’s tendency to return a ball “down the line” from the “ad-court” side with the “backhand”, we can specifically drill-down to the probability of this particular occurrence happening, and only use these features for clustering. Note that while approach #2 may also cover this scenario as a whole, it also has all other data as well, whereas this approach may only have data relating to this scenario. The preference of having a more “narrow” slice of data can be apparent since

feeding in too much “useless” data into a clustering model can sometimes overload or blur the algorithm’s inference ability. For example, if every player has the exact same tendencies when serving from the ad-court side, there might not be any purpose in feeding in this information since it may only serve to take its focus away from other data points that are more of a distinguishing factor. The benefit of this approach is it can confirm or refute what experts see from the game, and can make use of domain knowledge of the game.

From a quantitative side, we can also experiment with factors like feature importance by adding or dropping features based on clustering evaluation results. Since we have established a standard evaluation metric, we can take a more empirical approach, such as first selecting a number of initial features, run our clustering algorithm, evaluate, and then experimentally add or drop features based on evaluation metrics. By repeating this process, we can iteratively improve our clustering results, and choose the best set of features based on our metrics. We can also look at metrics such as feature importance or variance of a field in a dataset to determine which field to add or drop. The downside to this approach is the final feature set obtained may be less “human-interpretable” compared to the previous approaches.

As there are advantages and disadvantages with each technique, we will discuss them in more detail in the “Methodology - Feature selection” section.

1.4.3. Clustering & Machine Learning Techniques

Similar to how we empirically tested our feature sets, we also tested a variety of machine learning techniques to determine which clustering technique to use. In this paper, we experimented with three well known machine learning techniques to cluster players into separate clusters for each of the mentioned game components, including:

- K-Means clustering
- Agglomerative clustering
- DBScan

What is Clustering?

Clustering is the task of dividing a number of data points into a number of groups, or "clusters", in a way that data points within each group are more similar than they are with data points outside their group. In machine learning, this is known as an unsupervised problem, since we do not have the "labels" for each data point before feeding into the algorithm. Because of this, the algorithm must decide by itself how to label each data point. The number of clusters may or may not be given by the user, depending on the specific algorithm chosen [12].

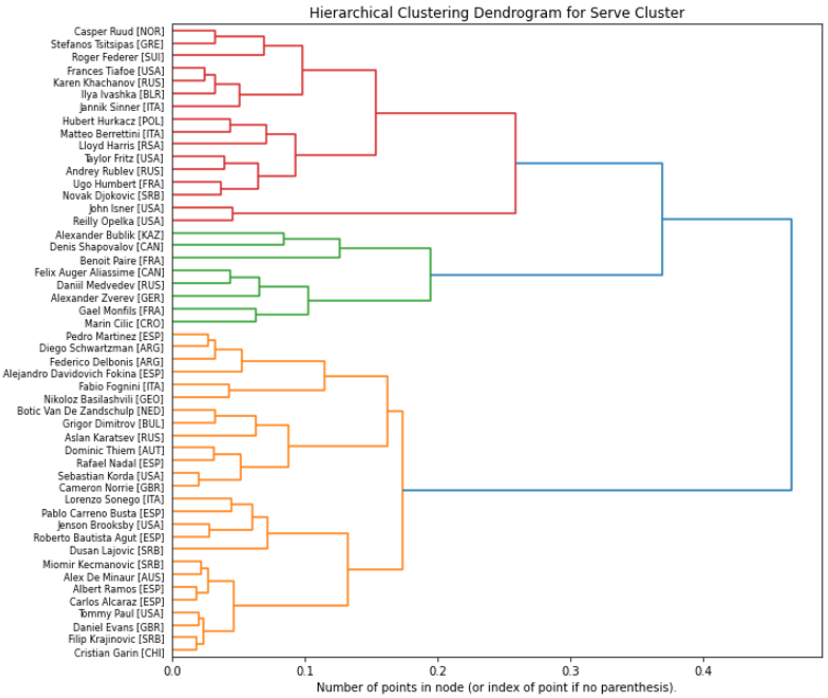
K-Means clustering is the most popular clustering algorithm used by practitioners, in which the user needs to choose "k", the number of clusters, before initiating the algorithm. As with any other clustering algorithm, the user must also have a data set, which has a set of features, which can be thought of as the number of columns in a dataset [13].

Let's assume we set "k" as 3, which means 3 clusters, and our data set has only 2 features, "x" and "y". By having only 2 features, we can easily visualize how the K-Means algorithm works. However, in practice, and in our experiments, there will often be many more features used. We can follow the below steps to understand how it works:

1. Randomly initialize k=3 clusters by placing them randomly along the X and Y axis, which represents the 2 features we are using to cluster. Each cluster location will also be referred to as "centroids".
2. During each iteration of K-Means, the algorithm will loop through each data point and calculate the "distance" between the data point and each cluster centroid. It then assigns the cluster label to that data point which is the nearest to it based on the lowest distance. The "distance" metric most often used is Euclidean distance.
3. After assigning every data point, it will then need to recalculate each clusters' centroid position by taking the average location of every data point that is assigned into its cluster.
4. Repeat steps 2 and 3 until each cluster centroid has "stabilized" and does not move much between iterations. The user can also choose to only run K-Means for a set number of iterations, for example, 50 times, in which K-Means will stop and return the cluster labels for each data point.

Agglomerative Hierarchical Clustering is another popular clustering algorithm, which is also known as a "bottom-up" approach. Agglomerative clustering starts by treating each data point

as its own cluster, and then successively merges pairs of clusters as it moves up the distance hierarchy [14]. This procedure can be visualized with a dendrogram:



As seen above, as we move from left to right in this dendrogram, the “distance” between points increases, as it is analogous to moving up the distance hierarchy. Clusters that were merged “earlier” (left-most side) indicate that the data points inside these clusters were more similar, which again, is determined by a “distance” metric. The distance metric can be Euclidean distance, Manhattan distance, or any other user-defined metric that is based on the feature set inputted into the algorithm.

After we have created this structure shown in the diagram above, we can choose at what point to “cut” the chart, which will output a final number of distinct clusters. For example, if we want to end up with 3 clusters, then the algorithm will simply stop “merging” clusters when it has reached 3 distinct clusters, which can be visualized as stopping around 0.3 in the horizontal axis in the dendrogram above.

DBScan is a clustering algorithm that is quite unlike the two mentioned above. First, it does not require the user to input the desired number of clusters. It is most suited for data that has a “density” attribute to it that cannot be captured by simpler metrics such as Euclidean distance [15].

For each experiment, we obtained results using all three cluster algorithms, and evaluated the results based on the “Intra-inter Cluster Distance” metrics mentioned. By recording these results, we can objectively arrive at the conclusion of which machine learning technique is best suited for our purpose.

1.4.4. Evaluation Metrics

Evaluating output accuracy in an unsupervised learning setting like clustering is inherently a challenging task. Since there are no “correct” labels to compare with, there isn’t really one objectively correct way to evaluate clusters. Having said that, most clustering problems generally rely on a set of metrics called intra-cluster distance and inter-cluster distance to evaluate cluster results [16]. In our study, “intra-cluster distance” is defined as the sum of squared differences (euclidean distance) between a data point and its own cluster’s centroid values, while “inter-cluster distance” is the sum of squared differences between a cluster’s centroid and a data point outside the cluster.

Intra-Cluster Distance: The average of sum of squared differences between the cluster’s centroid values (or cluster’s averages) vs. all of the features for every player inside this cluster.

$$\frac{\sum_{p \in P_{C1}} \sum_{f \in F} (f_{C1} - f_p)^2}{|P_{C1}|}$$

Formula for **Intra-cluster Distance** for Cluster #1:
(p: player, P_{C1}: all players in Cluster 1, f: feature, F: all features, C_1: cluster 1)

Inter-cluster Distance: The average of sum of squared differences between the cluster’s centroid values (or cluster’s averages) vs. all of the features for every player that is **not in its cluster**.

Formula for **Inter-cluster Distance** for Cluster #1: $$\frac{\sum_{p \in P_{\neg C1}} \sum_{f \in F} (f_{C1} - f_p)^2}{|P_{\neg C1}|}$$

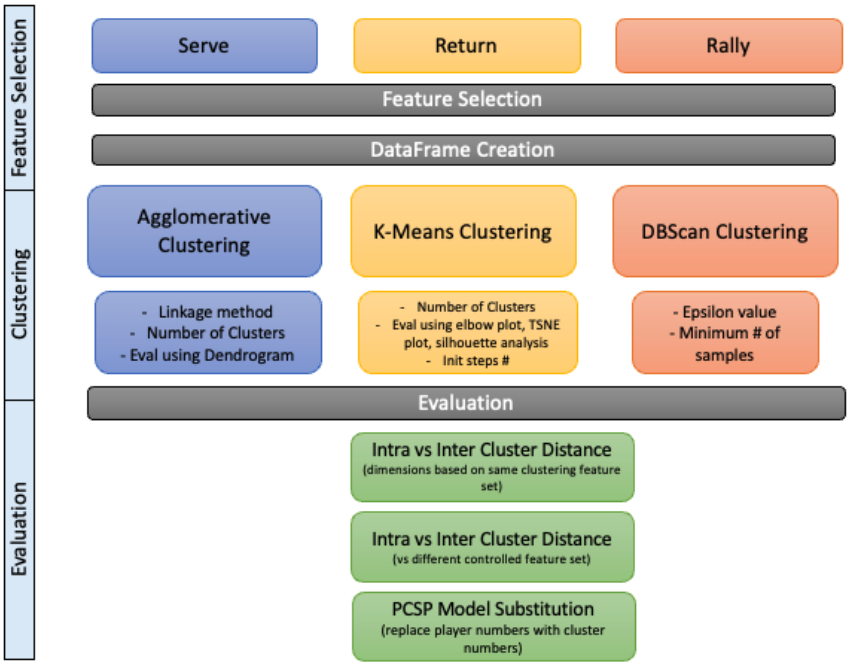
The only difference here is that we take all players that are **not in Cluster 1**, denoted by $P_{\neg C_1}$. Intuitively, for each cluster, we want to have a **low** intra-cluster distance, while having a **high** inter-cluster distance, since this would indicate members of each cluster are "close together", while members from different clusters are farther apart.

Next, we can simply take the difference between inter-cluster distance and intra-cluster distance as an indicator of how well divided each cluster is, and then average out this value for all clusters to evaluate the performance of a particular feature set and cluster algorithm.

After each clustering experiment, we will conduct a 3-part evaluation process:

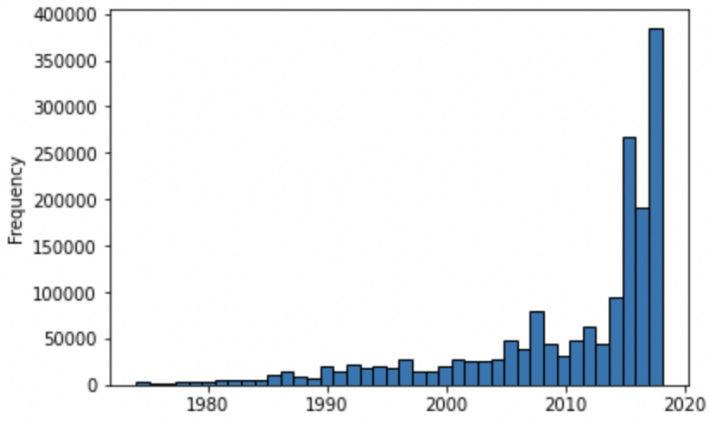
- **Evaluation #1 - Intra-Inter Cluster Distance Difference using Same Feature Set:**
    - This relates to using the same feature set as clustering to do the Intra-inter Cluster Distance evaluation.
- **Evaluation #2 - Intra-Inter Cluster Distance Difference against a Control Feature Set:**
    - To be more rigorous, we will try to use a completely different "controlled" feature set to evaluate against our cluster result that was clustered using a variable feature set.
- **Evaluation #3: PCSP Player-Cluster Replacement Difference:**
    - Finally, we will take the cluster results with the best Intra-inter Cluster distance scores from the previous evaluation steps, and input them directly into the original PCSP model. Ultimately, we will replace each player's individual data, with each of their game component's cluster's data. We will run the PAT model's verification functions to obtain a game winning % for each player, which we can compare with the baseline numbers, as well as several other test cases to complete our experiment.

Summary diagram of our end-to-end experiment workflow (going from top→bottom):



1.5. Scope

The scope of our analysis will cover the top 133 ATP men's tennis players, extracted and processed from the Tennis-Abstract website. The results can easily be extended to other players, such as more amateur, lower-tier players, or players from other tennis leagues, such as the WTP.



This chart shows the distribution of years in which the matches took place from the datasets available on the Tennis-Abstract website.

1.6. Project Scripts

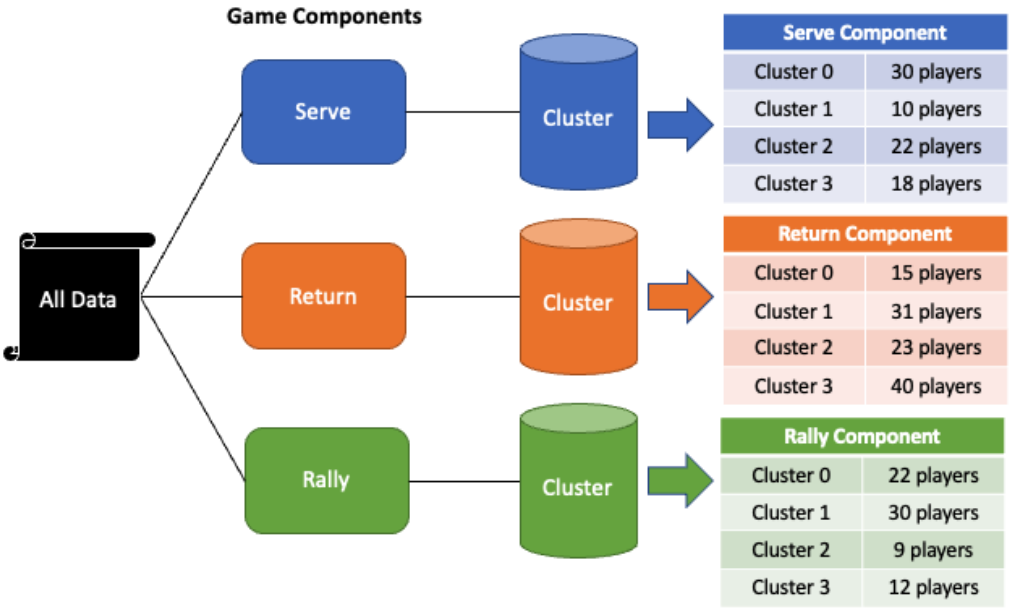
If interested, the experiments and Python scripts mentioned throughout this paper can be found on the project GitHub repository [17]. They encompass features and functionalities such as initial feature engineering, interactive visualization charts, main script to conduct end-to-end experiments, and the automated PCSP data populator tool.

2. Technical Research Contribution

This paper has six primary technical research contributions:

1. **New Player Clustering Method using Concept of Game Components:**

This study makes use of three game components - serve, return, and rally - data to create a new way to cluster tennis players based on different components of their game. Since each player can have a different cluster label for each component, it creates a more unique combination of overall play style than simply having one general cluster. For example, if we have 4 clusters per game component, there can be a total combination of 4 * 4 * 4 = 64 unique cluster combinations for each player, allowing room to properly classify a player into an overall play style that is sufficiently nuanced but also has similar components with other players. A variety of cluster algorithms were tested in our clustering pipeline as well, including K-Means, Agglomerative, and DBScan, as well as each of their hyperparameters, such as number of clusters, linkage method, and epsilon.



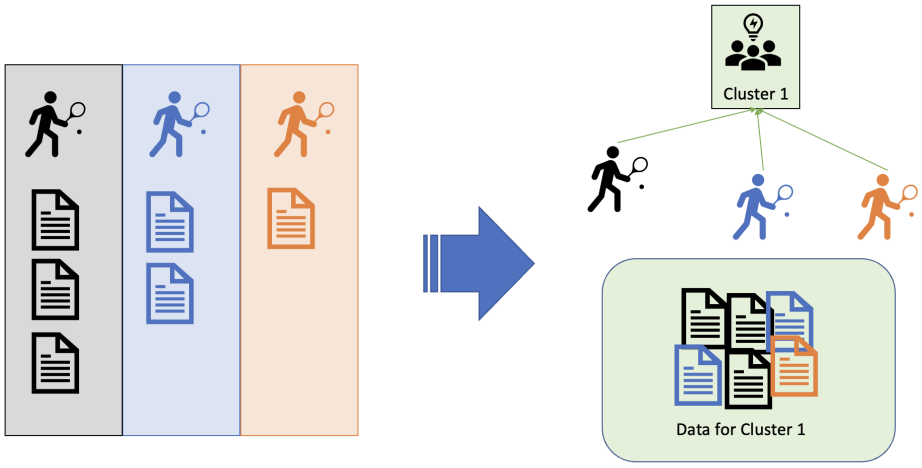
2. **3-Part Evaluation Process:**

We developed an automated 3-part evaluation process to thoroughly evaluate each cluster result. Metrics include the Intra-Inter Cluster Distance score against the same feature set, **Intra-Inter Cluster Distance** score against a control feature set, and lastly, the difference

score obtained when replacing player numbers with cluster numbers in the original PCSP model. The project script also has a feature to export the results summary for each cluster result into an Excel file.

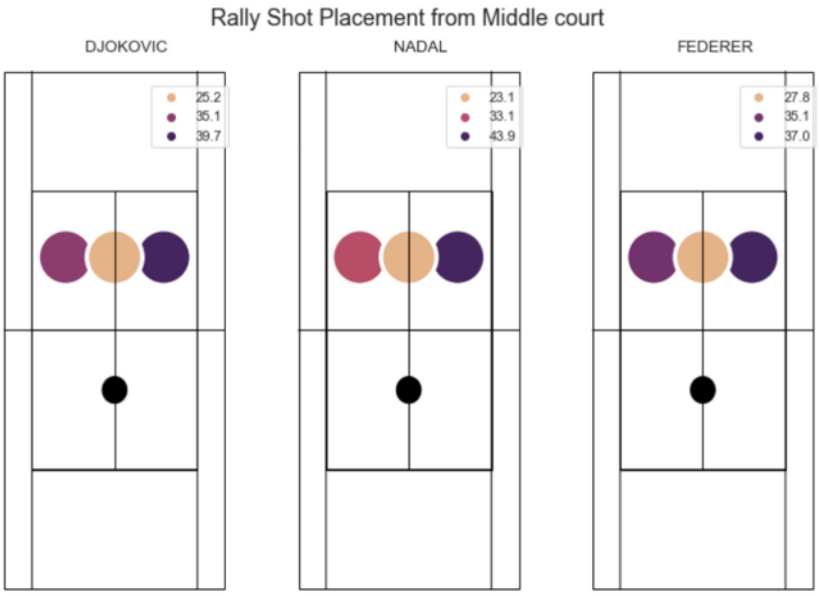
3. **Data Generalization through Player Clustering:**

This paper introduces the concept of data generalization through player clustering, which refers to allowing a player to access all of the cluster's data, after we place the player into the correct cluster. This way, we only need enough data to cluster a player, in order to enable that player to use the PCSP model using the cluster's dataset. This approach generalizes the simulation by simulating a cluster-vs-cluster match instead of a player-vs-player match.



4. **Interactive Shot Location Preference Chart:**

Using data from Tennis-Abstract, we developed an interactive and filterable data exploration chart to visually display where players prefer targeting shots, which can be filtered by shot type, player, player's cluster, and "from-court" location.



Filterable chart parameters using Python:

```
#Set 3 variables, and then run the cell to get the shot charts

players = ["Djokovic","Nadal","Federer"]
shot_type = "return"
from_court = "ad court"
cluster=True
```

5. **Automated PCSP Cluster Data Populator Script:**

Using the results from our cluster experiments, an automated script was created in Python to generate all the necessary PCSP data relating to each player's cluster for each game component to populate the original PCSP functions in the PAT model checker.

6. **Experiment Results:**

This study will present concrete empirical experiment results from the feature selection and evaluation phases mentioned above, and will be covered in more detail in the "Experiment Results" section.

3. Related Works

The phenomenon of using data to analyze sports and predict outcomes is not a new one. For as long as sports have been played, people at all levels have attempted to use data to solve age-old questions like “what are the chances of Team A winning over Team B?”

3.1. Sorting Strokes: Classifying Tennis Players Based on Style (HSAC)

Harvard Sports Analysis Collective (HSAC) published a blog detailing how they would use statistics to cluster tennis players into different playstyles, and also studied the impact on win probability based on different playstyles [18]. First, they believed there are four prominent style groups that we can qualitatively see and thus group together. These four styles are:

- Aggressive baseliners
- Serve-and-volleyers
- Counter-punchers
- All-court players

HSAC’s methodology included the use of K-Means clustering algorithm, and input features such as average ace percentage, serve speed, points played at net, net points won, unforced error rate, and frequency of forehand and backhand strokes. From their clustering results, they then proceeded to attribute a description and label name, such as “Left Counter-puncher”, to each cluster qualitatively based on the players they observed in each cluster.

Classification	Description	Examples
0: Left Counter-puncher	Consistent, low unforced error rate, mostly left-handed	Rafael Nadal, Feliciano Lopez, Fernando Verdasco
1: Right Counter-puncher	Right-handed, two-handed backhands, weaker net plays	Kei Nishikori, David Goffin, Alex Zverev
2: All-court	One-handed backhand, second highest serve speed group, median across other stats	Dominic Thiem, Grigor Dimitrov, Richard Gasquet
3: Big Server	Fastest serve group, high number of net approaches, aggressive play style	Roger Federer, Novak Djokovic, Andy Murray

[18]

Next, the authors added each player’s ELO rating to the model to indicate the skill difference between players. ELO is a popular player rating mechanism used in many zero-sum games, such as chess, which can also be adapted to sports [19]. Since their objective is to

contrast playstyle, ELO rating provides a way to scale and control the skill level, so they can focus on playstyle.

Elo Adv. (P1)	Elo Disadv. (P2)	Multiplied Effect on Win Odds Elo Adv.
Left Counter-puncher	Big Server	1.95
Right Counter-puncher	Left Counter-puncher	1.21
Right Counter-puncher	All-court	0.84
All-court	Big Server	0.4
Big Server	Right Counter-puncher	1.4

[18]

While controlling for skill level, they were able to calculate the historical winning likelihood of one cluster over another, which gives a good indication of which playstyle would match up well against another one. For example, from their study, they concluded that the cluster labeled “Left-counter-puncher” had a 1.95 times higher likelihood of winning against the cluster labeled “Big Server”, while the cluster labeled “All-court” had a 0.4 likelihood of winning against the “Big Server” cluster.

This study certainly shone an interesting perspective on clustering by play style, and its conclusion would certainly confirm what many fans would perceive while watching the game without the knowledge of data - that players perform and play differently while playing against opponents of different types. It is hard to argue against the fact that a strong “left counter-puncher” such as Nadal would play a certain way against an aggressive “All-court” player like Federer, but adjust his game accordingly against a renowned “big server” like John Isner.

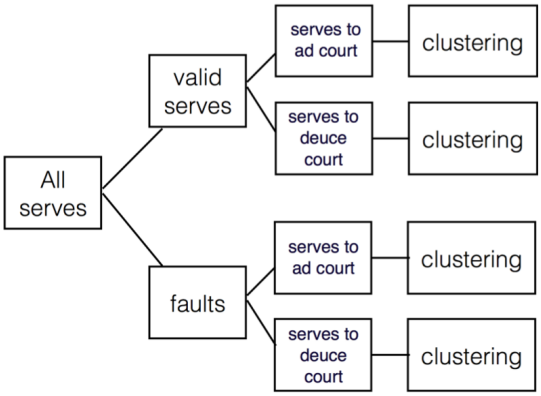
Although this study does a valiant job to address some of its issues, there are still a few shortcomings about the experiments conducted. First, as the authors pointed out themselves, it is quite difficult to separate play style from skill level, and ELO rating is not a perfect tool. Thus, skill level will still leak into the stats regarding win percentage between two clusters. In other words, as long as a cluster is able to house the best players, such as Federer, Djokovic, or Nadal, they will inherently always have a large winning percentage against a cluster that does not have these players. Next, perhaps due to the brief nature of the study, it did not bother to conduct any form of evaluation on the clustering results. In other words, the author simply ran the K-Means algorithm based on an arbitrary feature set, and took the results as they are given,

without questioning or evaluating them based on metrics, or trying the algorithm with another set of features. Although they did attempt to group the players in each cluster under a common description, it is done so after the fact, and in more of a qualitative way. Qualitative descriptions can often act as a useful heuristic, but it is also very subjective, so we should also have a more quantitative way to evaluate a cluster's quality. In our study, we employ quantitative metrics such as "Intra-inter Cluster Distance" in order to empirically determine what feature set works best in our cluster algorithms.

Lastly, HSAC's study concluded with historical winning odds between each cluster. While it is interesting to see the probability of players from one cluster winning over another, it suffers from some issues such as the skill-level issue mentioned earlier. The historical stats used were extracted from a match-level standpoint, which can often be an over-simplistic way of simulating matchups between two players. With our study, using the PCSP and PAT modeling functionality, we will conduct winning chances using shot-level data between two players, which includes much more nuance in a game, such as probabilities of a player making a certain shot type against another player, and the sequences afterwards.

3.2. Predicting Serves in Tennis using Style Priors (Wei et al.)

In 2015, Wei et al. published a paper, "Predicting Serves in Tennis using Style Priors", which used computer vision and spatiotemporal data to predict a player's serve shot locations. They utilized data from the Hawk-Eye technology from three years of the Australian Open Men's draw, and created "style priors" to predict a player's serve shot locations. Because they had the full x, y, z coordinates of every serve shot at every time point, they were able to recreate the entire spatial trajectory of each shot. Note that this was not a luxury that our study had, as we did not have access to fine-granted spatiotemporal data. In their study, they then incrementally grouped all the serve shots into layered sub-groups before performing clustering. The subgroups were "valid serves" and "faults" in the first level, and then "serves to ad court" and "serves to deuce court" on the next level.



By pre-grouping them first, they can cluster the serve shots against other serve shots that are based on a more similar scenario. This is similar to our approach of pre-grouping shots into different game components before clustering. Next, they used a Latent Dirichlet Allocation method to create a “serve dictionary” that contains 14 different types of serve shots. As a result, they can now represent every player’s serve style as a 14-dimensional feature vector, called “style prior”, which shows the likelihood of a player using each one of the 14 serves. In order to predict the serve type of a player, they interpreted the problem as a classification problem, in which they try to predict which of the 7 sides of each court the player was most likely to hit, given the current game context feature vector. They then used a Random Decision Forest classifier to predict the player’s next serve location, based on a variety of context feature vectors, that had increasingly more information about the game state. Their best result achieved 27.8% accuracy, which is roughly twice as good as a random guess of 1 in 7 possible locations (~14.2%). Ultimately, their study made use of a vast amount of spatiotemporal data to reveal stark differences in preferred serve locations for different players. They also showed interesting scenarios where a player such as Rafael Nadal may have a different serve profile while normally vs. under a break-point scenario.

We should note several differences between their study and ours. First, they have access to Hawk-Eye’s spatial-temporal data, including x, y, z coordinates of the ball and player at distinct times, which is not in the scope of our study, in which we are given only publicly accessible and scraped data from Tennis-Abstract.com. Thus, the accuracy of location predictions cannot reach the same level of granularity. Next, their study focused on a smaller part of the game, which is the serve shot, whereas our study will focus on a wider range of a tennis match, covering serve shots, return shots, and rally shots.

4. Objective

In tennis, much work has been put into the goal of predicting match outcomes. Our focus will be on cluster players into different playstyles, which will be helpful for a number of applications:

- Clustering players will help to provide a better understanding of player types, which can be used as an additional feature when predicting match outcomes
- Clustering players can help to generalize a player's playstyle, which will allow the player to have access to a much larger dataset pertaining to that cluster.

To build on top of Dong et al.'s research work, we want to further enable our PCSP models to be applicable to the lower-tier players, where we still need large enough data sets to produce usable insights. This means we need a way to generate sufficient data for each player to feed into the existing PCSP model. In a match between two players, for example, Player A and Player B, we ideally need data on games between these two specific players. Otherwise, the data will not be fine-grained enough to produce useful insights. For instance, take a match between Rafael Nadal and Novak Djokovic. In this matchup, Nadal, known as a defensive baseliner, may adopt a certain strategy, which may translate to a particular shot selection distribution against a player like Djokovic, who is also known to be a defensive baseliner. Whereas if Nadal plays against Roger Federer, known more as a "serve-and-volley" type of player, Nadal may choose to play with an entirely different strategy (source). Therefore, high-level players may adopt a different shot profile selection depending on the opponent's strengths and weaknesses. We will take a look at this claim in this report.

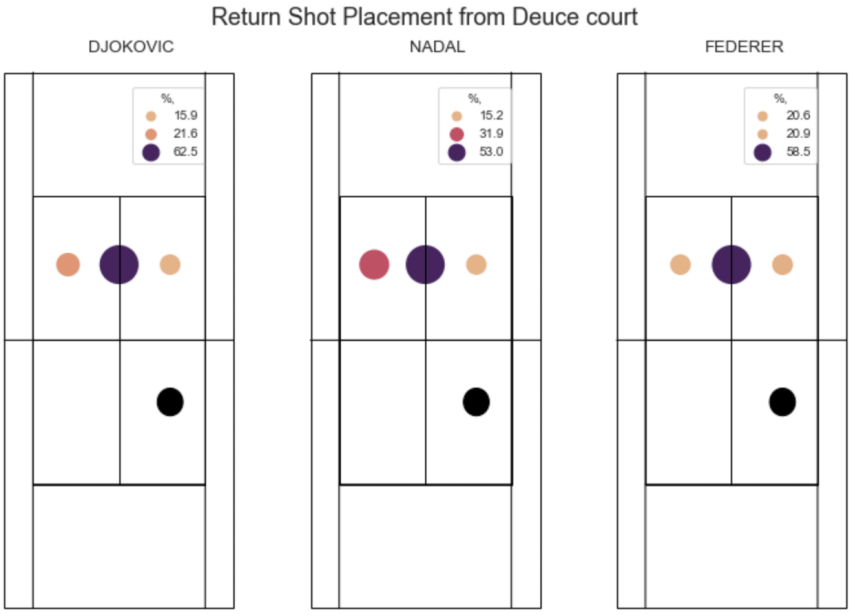
With this need for specific matchup data, the problem lies in the fact that this type of fine-grained historical data is practically non-existent for lower-level players. To counter this issue, we introduce the concept of playstyles to generalize and abstract out the need for immense amounts of data on specific players. By grouping similar players into distinct profiles based on style, we no longer need to collect data of matches between *player A* vs. *player B*, but more so between *playstyle A* vs. *playstyle B*. By clustering similar tennis players into distinct playstyles based on on-court tendencies, we do not need to spend inordinate amounts of time to learn about each individual player. By contrast, we only need enough data to correctly place a player into a playstyle, before we can extract useful information about them. Applications include predicting a match's outcome between two players simply based on their playstyle. The assumption here is

that our grouping must be accurate enough to represent what the player will likely do in certain in-game scenarios. We will introduce in the next section the specific game components we will focus on to wholly represent a player's playstyle.

In the context of Dong et al.'s research work, the outcome of this study will provide an alternative and larger dataset to plug into the existing PCSP game prediction model for each player. Ultimately, this will ideally improve the accuracy of prediction results where the amount of data is lacking. The assumption here is that the prediction between a playstyle against another playstyle is sufficiently as accurate as predicting the outcome between an individual player against another individual player. Ultimately, the implied goals of this project can:

- Make probabilistic model checking techniques accessible to the average low-tier player with a limited amount of data recorded for them and for certain match-ups. This can extend to being a scalable solution for any casual player interested in analyzing and improving their tennis skills.
- Use data to reveal useful indicators into how a player plays in a variety of game scenarios, which they can use to identify similarities and differences between peers, or strengths and weaknesses to improve the focus of practice sessions.

As a more concrete example, charting the frequencies in which the top players return a serve shot reveals tendencies in their "return game", which is one of the game components we will look at in our study. The illustration below shows that Nadal prefers to return the ball significantly more frequently to the deuce-court side, or "cross-court", when being served from deuce court, as compared to his peers, Nadal and Djokovic.



(Visual generated from this study)

The illustration extracted from our exploratory data analysis phase shows that when returning a serve shot on the deuce-court side, Nadal returns the ball to the opponent's deuce-court side at 31.9% of time, whereas Djokovic and Federer do the same at a rate of 21.6% and 20.9%. This tendency shows a difference of over 30% in how they prefer to return the ball in a controlled game scenario. The goal of our study is to be able to cluster players based on these differing in-game tendencies.

The logic follows that since Federer has a certain winning chance against Nadal, then the combination of Federer's clusters should fare similarly against the combination of Nadal's clusters. Of course, this interpretation may be overly simplistic, and ignores a host of other factors, such as the difference in players' skill level, on top of the accuracy and granularity of clustering methods. With that said, the purpose of this study is to reach a clustering methodology that sits at an abstraction level accurate enough to produce results that are acceptable to use for the majority of players in the PCSP system. Hence, even a slight gain in prediction results due to the clustering proves to be a helpful step in the direction of our objectives.

5. Methodology

In order to represent a player's playstyle in an encapsulated way, we want to capture their tendencies in certain game scenarios. However, we cannot split this into too many subcategories as we need sufficient data in every game component, and we want to abstract out components into mutually exclusive or orthogonal subcomponents of the game.

In this study, we will mainly focus on three game components, which will each produce a cluster label for each player:

- **Service Game:**
    - This component is related to the serve shots (first and second). Every point is started with a serve shot from the player who acts as the "server" for that tennis game. Players then alternate to serve between each game. A player that missed their first serve gets a second serve right afterwards. Both of these shots are captured in this game component.
- **Return Game:**
    - This component is related to the first return shot after the opponent has served.
- **Rally Game:**
    - This component is related to any shot after the return shot, or the third shot and after.

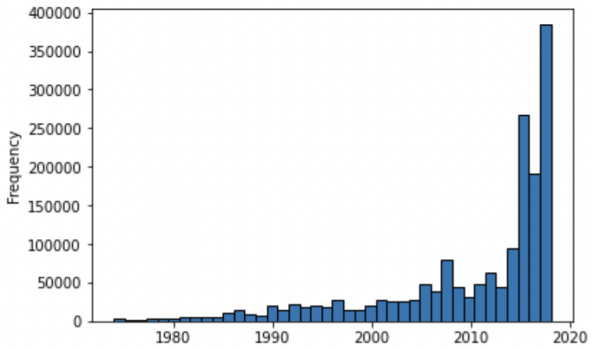
Consequently, each player will have a separate cluster label for each of the 3 game components. As mentioned, we chose to have 3 game components, rather than 1 general one, to avoid over-generalizations of a player's overall playstyle since we recognize each player may have vastly different styles depending on different components of their game. For example, Player A and Player B may have very similar serve tendencies whilst having vastly different rally and return game styles. This serves a more realistic representation of tennis players' play styles.

5.1. Data Source

There are two sets of data primarily used for this study - both of which were extracted from the Tennis-Abstract.com website:

- **Low-level data:**

This dataset contains individual shot-level data scraped from Tennis-Abstract's open sourced Github page using a web scraping script that was provided by Dong et al.'s research group [20]. The final processed csv file contains over 3.3 million rows of data, with each row representing details of a single shot, and spans from 885 players from 3881 matches over a span of years from 1974 to 2018, with over 50% of the data from matches between 2014 to 2018.



The low-level data set was used for multiple purposes, including for initial exploratory data analysis and visualization; for constructing multiple dataframes during the feature selection phase before clustering; and finally, for the final phase of replacing player numbers with player cluster numbers during the final PCSP data-replacement phase.

Field Name	Feature Definition
Player 1 name	Player's name
Player 2 name	Opponent's name
Player 1 handness	Dominant hand of player (RH for Right-hand, LH for Left-hand)
Player 2 handness	Dominant hand of opponent
Date	Date of match
Tournament Name	Name of tournament
Shot Type	Type of shot (1=1st serve, 2=2nd serve, 3=return, 4=rally)
From which court	Court location of when ball is hit by Player 1 (1=deuce court, 2=middle court, 3=ad court, 99=unknown)
Shot	Shot description (1~20 are forehand shots, 21~40 are backhand shots, 41 is trick shot, 99=unknown), detailed codes for all shots are:
Direction (serve)	Serve direction (1=deuce court, 2=middle court, 3=ad court, 4=serve to wide, 5=serve to body, 6=server to T, 99=unknown)
To which court	Shot direction (1=deuce court, 2=middle court, 3=ad court, 99=unknown)
Depth	(1=shallow, 2=deep, 3=very deep, 99=unknown)
Touched Net?	(1=yes touched net, 2=not)
Hit at what depth?	(1=at net, 2=at baseline, 99=unknown)
Approach shot?	(1=yes, 2=no)
Shot outcome	Outcome of shot (1=ace, 2=fault, 3=forced error, 4=unforced error, 5=-winner, 6=service winner, 7=no outcome)
Fault type	Fault type, if Shot Type = serve (1=(net), 2=(wide), 3=(long), 4=(wide and long), 5=(foot fault), 6=(shank), 7=no fault, 99=other)

- **High-level data:**

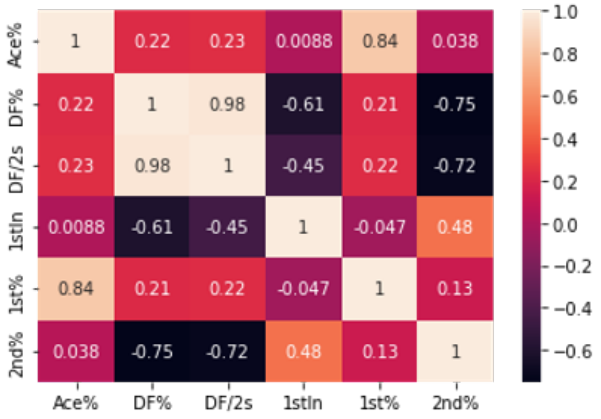
  The high-level data was taken directly from the summary data on Tennis-Abstract's website, under **Men's Report → Career** section, where there are four categories including Serve, Return, Rally, and Tactics [8]. We make use of data from the first three categories. Unlike the low-level data where the bulk of the processing and transformation work still had to be done, the high-level data is already in the summary table format, containing career statistics for every player for each of the four categories. The tables contain 133 of the top players recorded on the website throughout the years. The data for "Serve stats" are shown below:

Service Game	Feature Definition
Matches	Matches logged by the Match Charting Project
Unret%	Percent of serves that were unreturned
<=3 W%	Percent of service points won on either the serve or second shot
RiP W%	Percent of points won when return was put in play
SvImpact	Serve Impact: Advanced stat estimating how many service points were won due to the serve
1st: Unret%	Percent of first serves that were unreturned
1st: <=3 W%	Percent of first serve points won on either the serve or second shot
1st: RiP W%	Percentage of first serve points won when return was put in play
1st: SvImpact	Serve Impact: Advanced stat estimating how many first serve points were won due to the serve
D Wide%	Percent of deuce-court first serves that were hit wide
A Wide%	Percent of ad-court first serves that were hit wide
BP Wide%	Percent of (ad-court) break point first serves that were hit wide
2nd: Unret%	Percent of second serves that were unreturned
2nd: <=3 W%	Percent of second serve points won on either the serve or second shot
2nd: RiP W%	Percentage of second serve points won when return was put in play
1st: D Wide%	Percent of deuce-court second serves that were hit wide
1st: A Wide%	Percent of ad-court second serves that were hit wide
1st: BP Wide%	Percent of (ad-court) break point second serves that were hit wide
2ndAgg	2nd Serve Aggression Score (0 is average, higher = more aggressive)

The data for the other two categories are shown in **Appendix 2**.

5.2. Exploratory Data Analysis (EDA)

For the EDA phase, there were a few things that we wanted to learn about our data. For the low-level data, we filtered down the DataFrame so it only contained data for the same 133 players that were in the high-level dataset. The below showcases a few samples of EDA plots and the purposes of them.

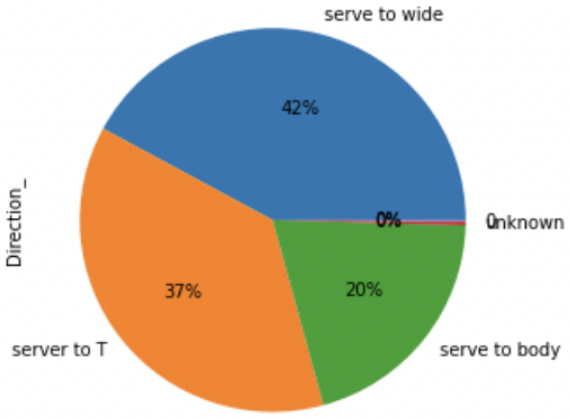


We ran a correlation plot for features in the high-level dataset, which was used for clustering players. Naturally, there was a lot of correlation between some features, such as Ace % on 1st serve and Ace % on 2nd serve. Hence we also ran clustering on a slimmed down version of the original feature set that did not have near-duplicates of features. Having too many features that resembled too closely to another feature may introduce the issue of multicollinearity, which would produce less than ideal clustering groups.

				Percent of returns put in play	Percent of points won when return was put in play	Return winners (and induced forced errors) as a percentage of return points	Forehand winners (and induced forced errors) as a percentage of all return winners	Return Depth Index (higher = deeper)	Slice/chip returns as a percentage of all in-play returns
	Player	Return Cluster_Agglom	Return Cluster_KMeans	RiP%	RiP W%	RetWnr%	Wnr FH%	RDI	Slice%
1	Mats Wilander	0	2	0.753	0.495	0.046	0.64	1.85	0.308
6	Bjorn Borg	0	2	0.742	0.528	0.04	0.437	2.09	0.362
27	Novak Djokovic	0	1	0.708	0.521	0.026	0.587	2.16	0.202
35	Mischa Zverev	0	0	0.7	0.486	0.095	0.327	2.01	0.109
49	Roger Federer	0	1	0.689	0.505	0.042	0.513	2.05	0.159
50	Michael Chang	0	1	0.687	0.504	0.032	0.602	2.21	0.218
58	Roberto Bautista Agut	0	3	0.682	0.47	0.029	0.47	2.32	0.149
62	David Nalbandian	0	1	0.677	0.522	0.046	0.497	2.17	0.175
64	Philipp Kohlschreiber	0	1	0.676	0.485	0.039	0.467	2.19	0.144
65	Lleyton Hewitt	0	1	0.676	0.507	0.04	0.618	2.22	0.256
70	Benoit Paire	0	1	0.673	0.486	0.044	0.465	2.19	0.152
71	Martin Klizan	0	1	0.672	0.526	0.033	0.404	2.29	0.147
73	Grigor Dimitrov	0	1	0.668	0.475	0.039	0.488	2.2	0.177
74	Pablo Carreno Busta	0	3	0.668	0.474	0.038	0.488	2.34	0.128

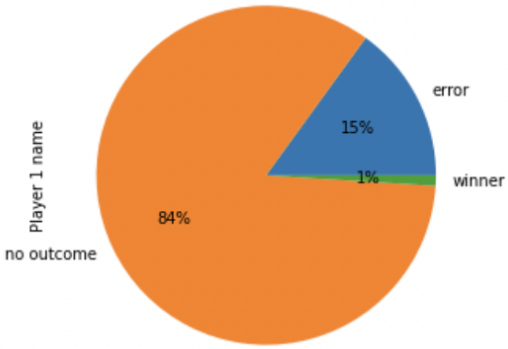
Since we are clustering players solely on the high-level data feature sets, it is useful to see the level of variance for each feature between every player. We used conditional formatting by colour scale in Excel to visually see the variance for each feature. Also, after we have the cluster results, we can get an idea of the key differences between each cluster to get a better sense of how the cluster algorithm decided on the cluster outcome, and whether it lines up with expectation.

Lastly, we want to get an understanding of shot location preferences for each player. The below is an interactive visual chart built in Python that can be filtered by player (or player's cluster), shot types, and "from court" location.

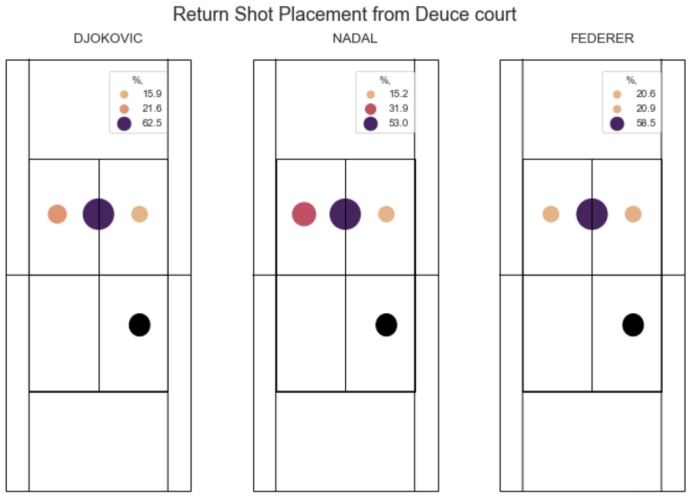



```
Returns from Deuce Court:

To deuce court:
Federer :
Shot_Outcome_group
error         15.024631
no outcome    83.990148
winner         0.985222
Name: Player 1 name, dtype: float64
```




```
Djokovic
   court  count  percent
0    mid   3594     62.5
1  deuce   1242     21.6
2     ad    912     15.9
Nadal
   court  count  percent
0    mid   3363     53.0
1  deuce   2023     31.9
2     ad    963     15.2
Federer
   court  count  percent
0    mid   4678     58.5
1     ad   1668     20.9
2  deuce   1647     20.6
```



Filterable chart parameters in Python:

```
#Set 3 variables, and then run the cell to get the shot charts

players = ["Djokovic","Nadal","Federer"]
shot_type = "return"
from_court = "ad court"
cluster=True
```

From the above example, we learn that Nadal returns the ball to the opponent's deuce-court side at 31.9% of time, whereas Djokovic and Federer do the same at a rate of 21.6% and 20.9%. Having the ability to compare filtered down shot location statistics between players can both uncover interesting insights and confirm or refute our intuition about how a player likes to play. This can also benefit our feature selection, since if we can find major differentiating points between players in our dataset, we can use the exact same features to be our clustering feature set.

5.3. 3-Step Iterative Experimental Process

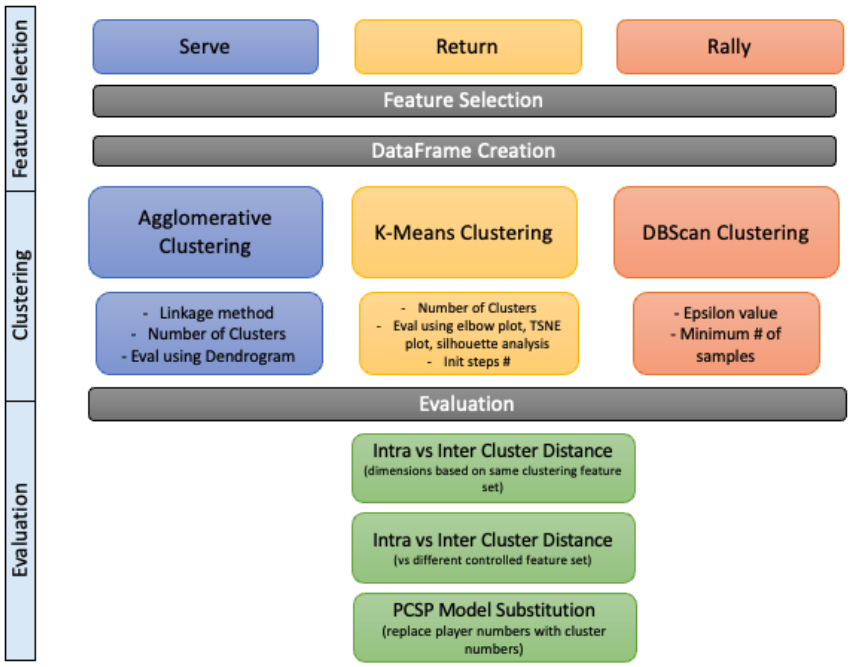
After our ETL step, we will run our 3-step iterative experimental process. Each cycle of our iteration experimental process includes the following for each game component. So for example, for the serve component, we conducted the following steps:

1. **Feature Selection & DataFrame Creation:** Select a set of features out of the raw datasets to use for clustering. After which, we will transform the original dataset and create the DataFrame which holds the probability of each needed feature (Eg. "backhand to ad court" probability).
2. **Clustering:** Cluster data using multiple clustering algorithms and experimentation with various hyper-parameters.
    a. K-Means
    b. Agglomerative
    c. DBScan
3. **Evaluation:** Evaluate the clustering result based on:
    a. Intra-Inter Cluster Distance metric, using the same set of features as clustering to evaluate. This should give an indicator of how well-separated each cluster is in the clustering result.
    b. Intra-Inter Cluster Distance metric, using a "controlled" feature set that is distinct from the clustering feature set. This acts as a consistent "truth" in which we test

how similar the results of our clustering feature set is against a base set of features, which we will decide at the beginning of experimentation.

c. Input the clustering results into the PCSP, by replacing each player's probability numbers with each of their cluster's probability numbers.

Lastly, for each game component, we iteratively repeat steps 1-3 until we have sufficient results, so that we can choose the feature set that generates the best results based on our evaluation criteria from step 3.



The diagram above shows that for every game component, we go through the full process of **Feature Selection & Dataframe Creation → Clustering → Evaluation**. We will go through each stage of the process in more detail below.

5.3.1. Feature Selection

Selecting a feature set for clustering data points is a challenging task. In our context, the features we choose should ideally be a complete and self-contained abstraction of the player's playstyle for each game component. For example, the feature set we choose for "Service game" should by itself, fully represent the player's tendencies whenever he is serving the ball. As is with any machine learning task, finding the right level of abstraction can often be both an empirical

process and an intuitive one. In our study, we employ a trial-and-error approach to evaluate and choose an appropriate feature set for our clustering task.

To select the best set of features, we employ the following methodology to find the best possible set of features to carry out clustering, and then enter them into the PCSP to do a final evaluation comparison. We recognize there are challenges in identifying an overarching or unifying set of features that can wholly represent a player's play style without fault. For this reason, we've experimented with a variety of datasets, at varying levels of abstraction, in order to accurately depict the players' playstyle for a particular game component.

1. **High-Level Summary Statistics:**

   In our first experiment, the clusters were formed using the following summary statistic fields obtained from Tennis-Abstract's ATP Stat Leaders board, related to the top 133 players.

Player	Matches	Unret%	<=3 W%	RiP W%	SvImpact	1st: Unret%	<=3 W%	RiP W%	SvImpact	D Wide%	A Wide%	BP Wide%	2nd: Unret%	<=3 W%	RiP W%	D Wide%	A Wide%	BP Wide%	2ndAgg
Ivo Karlovic	35	48.0%	63.2%	50.9%	55.3%	59.0%	74.8%	57.0%	70.1%	42.9%	56.9%	57.5%	31.6%	48.0%	42.7%	26.4%	30.2%	14.3%	126
John Isner	83	45.4%	59.1%	48.4%	52.1%	54.0%	67.2%	51.0%	63.4%	50.0%	45.7%	57.9%	26.9%	43.0%	43.9%	22.0%	52.4%	68.5%	34
Maxime Cressy	25	43.9%	59.0%	55.8%	52.0%	51.0%	68.4%	58.0%	64.3%	55.2%	60.1%	73.2%	44.0%	59.6%	51.6%	46.1%	54.9%	56.5%	248
Reilly Opelka	52	46.1%	58.5%	48.4%	51.5%	57.3%	68.9%	50.4%	65.4%	50.0%	44.2%	45.2%	26.6%	41.9%	46.0%	33.2%	54.2%	82.4%	31
Goran Ivanisevic	44	42.1%	53.9%	49.8%	47.4%	58.9%	72.1%	55.4%	68.6%	38.1%	55.0%	56.8%	26.3%	38.3%	45.4%	16.9%	51.7%	65.6%	119
Milos Raonic	78	39.7%	54.0%	51.6%	47.1%	52.0%	68.1%	57.3%	63.8%	53.6%	50.2%	45.5%	21.4%	34.1%	45.0%	28.1%	54.4%	71.8%	76
Richard Krajicek	20	40.2%	53.6%	47.8%	46.8%	54.9%	69.8%	56.7%	66.5%	45.7%	39.9%	27.8%	23.9%	37.0%	39.9%	12.4%	49.8%	70.0%	60
Nick Kyrgios	111	40.5%	51.8%	51.0%	46.4%	51.0%	63.2%	54.2%	59.8%	43.3%	40.9%	57.0%	23.0%	33.7%	46.4%	31.1%	50.5%	43.3%	103
Gilles Muller	20	37.3%	54.3%	53.1%	45.9%	49.3%	69.5%	60.4%	63.2%	45.5%	46.6%	40.0%	19.8%	33.2%	44.5%	24.2%	30.4%	23.5%	75
Sam Querrey	33	39.7%	51.3%	48.3%	45.6%	54.0%	67.3%	52.5%	63.7%	47.4%	47.3%	52.9%	20.1%	30.1%	44.0%	35.9%	47.4%	41.5%	78

The data dictionary can be seen in the Data Source section under "High-level data".

For the serve game component, there are 18 fields, that contain features such has the following:

- Percent of serves that were unreturned
- Percent of ad-court first serves that were hit wide
- 2nd serve aggression score

Next, we then run a set of clustering algorithms on this dataset to come up with a set of serve clusters, which we will go through in more detail in the next section.

2. **Low-Level Shot-by-shot Statistics:**

   To compare our clustering results with the first set, we also run a second set of features, which contains a lot more low-level statistics on each player. For the serve game component, one of the feature sets used contained 36 features, including:

- 1st serve-deuce court-serve to body-fault: Conditional on 1st serve and on deuce court side, probability of serving towards opponent's body and faulting
- 1st serve-deuce court-serve to body-no outcome: Conditional on 1st serve and on deuce court side, probability of serving towards opponent's body and there is no outcome, meaning opponent returns the ball
- 1st serve-deuce court-serve to body-service winner: Conditional on 1st serve and on deuce court side, probability of serving towards opponent's body and winning the point immediately

	feature	Player 1 name
0	1st serve-deuce court-serve to body-fault	0.027980
1	1st serve-deuce court-serve to body-no outcome	0.076998
2	1st serve-deuce court-serve to body-service wi...	0.006317
3	1st serve-deuce court-serve to wide-fault	0.145073
4	1st serve-deuce court-serve to wide-no outcome	0.246390
...	...	...
31	2nd serve-ad court-serve to wide-no outcome	0.470045
32	2nd serve-ad court-serve to wide-service winne...	0.001484
33	2nd serve-ad court-server to T-fault	0.025402
34	2nd serve-ad court-server to T-no outcome	0.161896

Evidently, the level of data is much more granular as compared to the high-level data. Each one of the 36 probabilities covers a likelihood a player will serve to a specific part of the court in a scenario. More specifically, the dataframe was created based on all the permutations of serve type, court, direction, and outcome.

Serve	Court	Direction	Outcome
1st serve	Deuce court	Serve to body	Fault
2nd serve	Ad court	Serve to wide	No outcome
		Serve to T	Serve winner

Hence the final dataframe has 2*2*3*3 = 36 features.

There is a second, and perhaps more important reason we chose to use this specific set of features to do the clustering for the serve game component. This level of detail emulates the serve functions from the original PCSP model:

```
De_Ply1Serve = pcase {                              // all probability is base
             231: ServeT_in{ball= 6} -> Ply2_BackHandR  // T will have opponer
             119: ServeT_err{ball=9} -> De_Ply1Serve_2nd   // Federer tries to
             337: ServeWide_in{ball =6} -> Ply2_ForeHandR
             173: ServeWide_err{ball=9} -> De_Ply1Serve_2nd
             92: ServeBody_in{ball=6} -> (Ply2_BackHandR [] Ply2_ForeHandR)
             48: ServeBody_err{ball=9} -> De_Ply1Serve_2nd};
```

For example, the above PCSP function is showing the likelihood numbers of every direction and outcome, while Player 1 is on his 1st serve from the deuce court side. Since the end goal of this study is to plug the numbers back into the PCSP module, intuitively, it makes sense to use the same features to cluster the players, and then output the new numbers from each cluster into the PCSP.

We then follow the same logic for the Return and Rally game components, which we will not go into detail in this report. The features used for the other two game components can be found in the **Appendix 3**. A sample of a selected feature set for Return game component is shown:

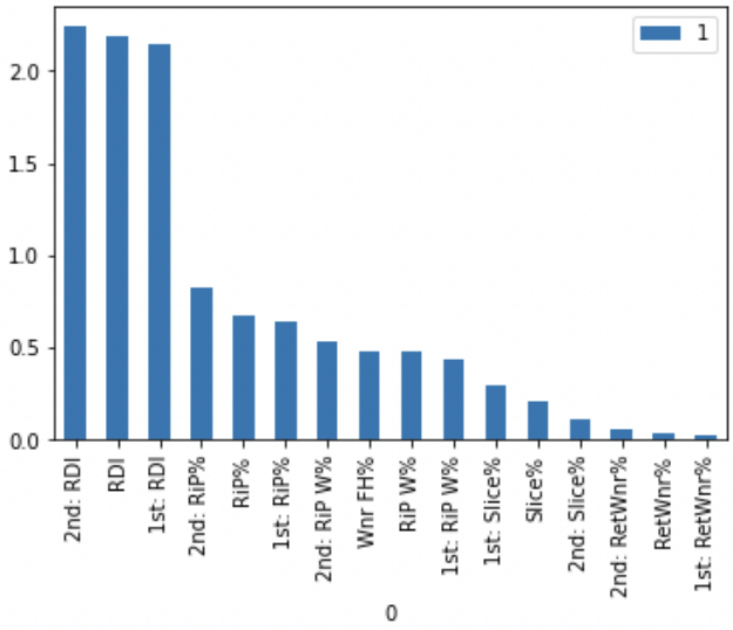
	Feature	Value
0	backhand-ad court-forced error	0.040823
1	backhand-ad court-no outcome	0.266929
2	backhand-ad court-unforced error	0.011016
3	backhand-ad court-winner	0.007635
4	backhand-deuce court-forced error	0.029213
5	backhand-deuce court-no outcome	0.094844
6	backhand-deuce court-unforced error	0.007243
7	backhand-deuce court-winner	0.009683
8	backhand-middle court-forced error	0.058122
9	backhand-middle court-no outcome	0.461389
10	backhand-middle court-unforced error	0.012224
11	backhand-middle court-winner	0.000879
12	forehand-ad court-forced error	0.047287
13	forehand-ad court-no outcome	0.167080
14	forehand-ad court-unforced error	0.009430
15	forehand-ad court-winner	0.010445
16	forehand-deuce court-forced error	0.041661
17	forehand-deuce court-no outcome	0.142993
18	forehand-deuce court-unforced error	0.010318
19	forehand-deuce court-winner	0.014865
20	forehand-middle court-forced error	0.072570
21	forehand-middle court-no outcome	0.469735
22	forehand-middle court-unforced error	0.012365
23	forehand-middle court-winner	0.001250

The value column shows the average percentage of each feature for all players. Below is another DataFrame which shows the average percentages for every player over all 24 features of the above feature set. Essentially, this DataFrame is needed as the input into our clustering algorithms.

Some additional factors of consideration during DataFrame creation before the clustering phase include: Choosing the number of effective clusters per game component (eg. default = 4),

whether to scale or normalize the dataset, and whether to use PCA (principal component analysis) to abstract out noise. The combination of factors will be tested out depending on the performance of algorithms. Also, we can look at the feature importance or total variance of each feature in our feature sets and then choose to remove the features that have less variance. Theoretically, this will leave us with a set of features that represent more of a differentiating factor between players, since low variance means all players have a similar value for that feature. This acts as a quick way to determine whether a feature is important in our clustering step.

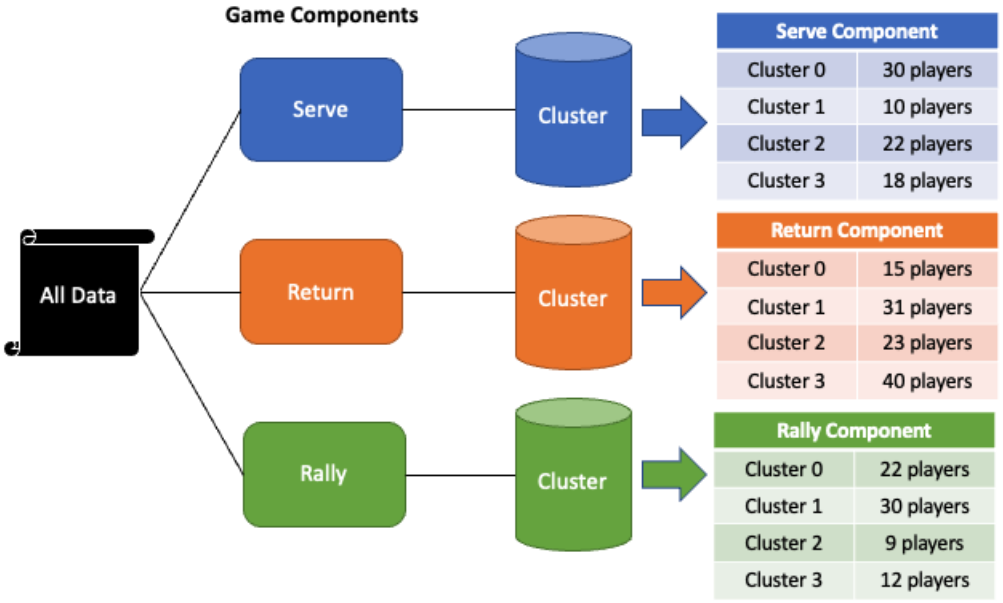
```
feature: 0 Mean difference: 1.69 Total Variance: 459.464
feature: 1 Mean difference: 0.879 Total Variance: 449.829
feature: 2 Mean difference: 66.213 Total Variance: 154932.184
feature: 3 Mean difference: 2.076 Total Variance: 2731.953
```



In the above feature importance chart for our Return game component, for example, we can see that the features with the highest feature importance came from “2nd: RDI”, “RDI”, and “1st RDI”. However, this in itself may not tell the full story, since these three features also happen to be highly correlated. Therefore, this shows that it is often best to use both quantitative methods and intuition to make decisions on what features to use to represent a game component.

5.3.2. Clustering

After we have this dataframe with every player's averages for the feature set we want, we will explore three machine learning techniques to properly cluster different players into groups, namely K-Means, Agglomerative, and DBScan, to test out the groupings of different players.



For this phase, we make use of 3 helper functions: cluster_agglomerative(), cluster_kmeans(), and cluster_dbscan() in our scripts. As the names suggest, the functions simply automate the process of clustering the players with each algorithm. The functions wrap around preparation tasks such as:

- Scaling the data using the scale function from sklearn.preprocessing library
- Running the KMeans algorithm from SkLearn library, while setting the number of clusters, and choosing the linkage type for the Agglomerative algorithm.

```
kmeans = KMeans(n_clusters=no_of_clusters,
random_state=0,n_init=n_init).fit(df_return_scaled)
```

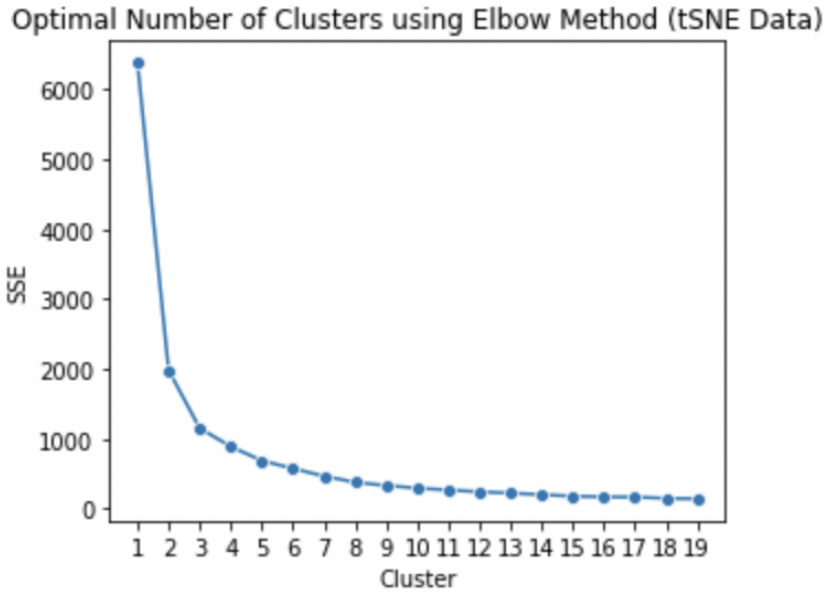
- Outputting result of final distribution of players into clusters, displayed as follows:

```
[3 3 3 3 3 3 3 3 3 3 0 0 3 3 3 3 3 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 2 0 0 0 0 2 1 0 2 0
 0 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 2 0 2 2 2 0 2 2 2 2 2 2 2 2 0 0 2 0 2 2
 2 2 2 2 1 2 2 1 1 2 1 2 1 2 2 1 1 1 2 2 2 1]
0    70
2    36
3    17
1    10
Name: Return Cluster, dtype: int64
```

- The results in this example show that 70 players were clustered into the Cluster #0, 36 in Cluster #2, and so on.

K-Means Clustering:

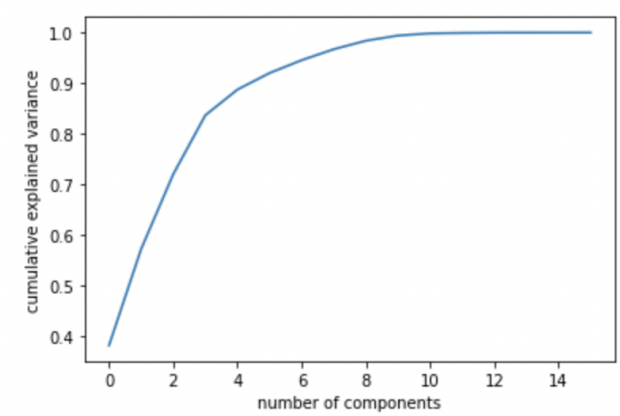
For K-Means, the main decision factor was to decide on the optimal number of clusters ("k"). To decide on this, we employed the t-SNE algorithm and elbow method technique. The elbow method is a heuristic in which we plot "# of clusters" on the x-axis and "sum of squared errors" on the y-axis and pick a sharp point that resembles an "elbow" to be an optimal # of clusters. In essence, we want to pick a # of clusters where the sum of squared errors is seen to decrease at a slower, linear rate. Before we were able to plot this, we had to make use of t-SNE, which is a dimensionality reduction algorithm, to reduce our dataset into two components.



From the elbow method plot shown above, we can see that k = 3 or 4 seems to be a reasonable choice.

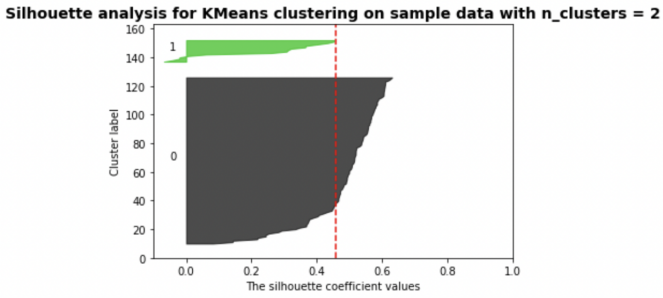
Principal Component Analysis:

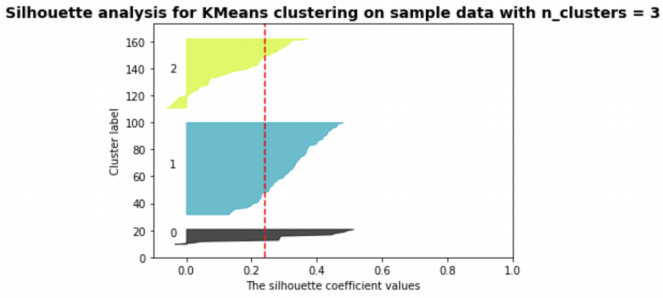
For our first feature set, we also conducted an experiment to see if using PCA to reduce the data's dimensionality would have a positive effect on clustering results.

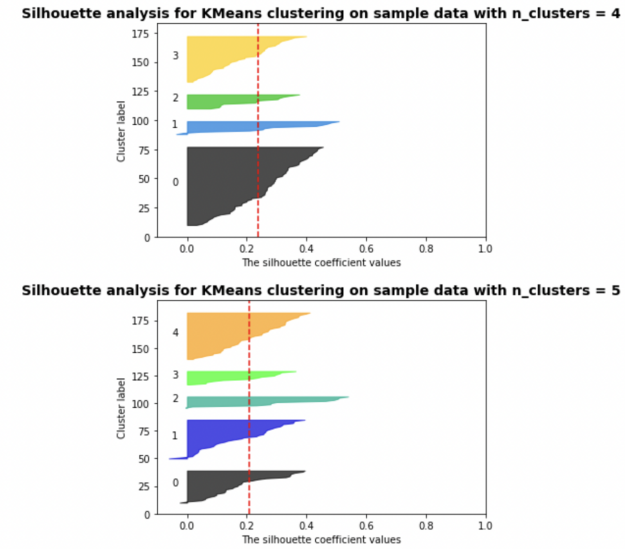


For example, for this feature set, we see the 8 components can explain approximately 90% of the variance. We also conducted a silhouette analysis to determine how many clusters would be optimal:

```
For n_clusters = 2 The average silhouette_score is : 0.45822852658085733
For n_clusters = 3 The average silhouette_score is : 0.2426829050388124
For n_clusters = 4 The average silhouette_score is : 0.23845310029601707
For n_clusters = 5 The average silhouette_score is : 0.20715750453137297
For n_clusters = 6 The average silhouette_score is : 0.19811178894807893
For n_clusters = 7 The average silhouette_score is : 0.19632872307010626
For n_clusters = 8 The average silhouette_score is : 0.1903637817775905
For n_clusters = 9 The average silhouette_score is : 0.19407308422636788
For n_clusters = 10 The average silhouette_score is : 0.1986173450144459
```







Basically, we want to choose the “k” number that has a relatively higher score, while having relatively equal proportional sizes for each cluster, which can be seen in the plots above.

Agglomerative Clustering:

For the Agglomerative clustering method, our decision factors have to do with number of clusters, and linkage method. Linkage determines the distance metric when choosing which clusters to merge successively. The available options are ‘ward’, ‘average’, ‘complete’, and ‘single’. From testing between each method, we determined “ward” is the most suitable choice, as it often gave the most evenly balanced cluster sizes.

DBScan Clustering:

For density-based data. One shortcoming is we cannot decide on the number of clusters as a part of the algorithm initialization. But we can adjust the epsilon value until we get a desired number of clusters.

After seeing the results, we concluded DBScan was not a suitable algorithm for our purpose, since the proportion of cluster sizes seem to be heavily skewed, perhaps indicating the fact that many of each players’ stats are perhaps more similar than they are different. Hence, there is not enough variance in our data to have separately dense clusters. This makes sense since in our case, every tennis player is essentially playing the same game, such that there will not be any major differences that can be gleaned from the numbers. DBScan is a suitable choice for scenarios such as clustering data relating to different species of animals or plants, which would result in dense clusters of data that can be easily picked out by this algorithm.

5.3.3. Evaluation Metrics

For evaluation, we make use of a 3-part evaluation process, in which we make use of the evaluate() helper function in our script to run a set of evaluations.

In order to decide whether a selected feature set is suitable for clustering, we need to select an evaluation metric. For the first two evaluation steps, we start by generating two scores: Intra-cluster distance, and Inter-cluster distance for each cluster:

Intra-cluster Distance of a Cluster:

The sum of squared differences between the cluster's centroid values (or cluster's averages) vs. all of the features for every player inside this cluster. For each cluster, each player will have one score, which is the sum of all squared differences of all its features. Then we take the average of scores over all players inside each cluster. For example, the formula for Intra-cluster Distance for Cluster 1 is:

$$\frac{\sum_{p \in P_{C1}} \sum_{f \in F} (f_{C1} - f_p)^2}{|P_{C1}|}$$

(p: player, P_{C1}: all players in Cluster 1, f: feature, F: all features, C_1: cluster 1)

Inter-cluster Distance:

The average of sum of squared differences between the cluster's centroid values (or cluster's averages) vs. all of the features for every player that is **not in its cluster**. It is essentially the same as Intra-cluster distance, except instead of comparing the point inside the cluster with the cluster's average, we compare each point **outside** the cluster with that cluster's average. The higher this score is, the less "similar" each player is with the players in this cluster, which is the desired outcome for Inter-Cluster distance. The formula for Inter-cluster distance for Cluster 1 is:

$$\frac{\sum_{p \in P_{\neg C1}} \sum_{f \in F} (f_{C1} - f_p)^2}{|P_{\neg C1}|}$$

(As mentioned, the only difference here is that we take all players that are **not in Cluster 1**, denoted by $P_{\neg C_1}$.)

Intuitively, for each feature used, we are checking how "similar" each player is towards other players in their own cluster, and checking how "dissimilar" each player is towards players in clusters that they are not part of. Thus, we want to have a **low** intra-cluster distance, while having a **high** inter-cluster distance. By looking at the difference in these 2 scores, we can quantify how effective a clustering result is. We will refer to this evaluation as the "Intra-Inter Cluster Distance" score.

Evaluation #1: Intra-Inter Cluster Distance Difference using Same Feature Set:

As the name suggests, we will simply evaluate the cluster results using the same feature set in which it was clustered. This should return a positive result (high difference) for every cluster, since the algorithm should be optimized to separate each cluster in the same way. By right, this would already be a sufficient way to evaluate a cluster's quality. However, we wanted to take it a step further, in which the process is described in Evaluation step #2.

Evaluation #2: Intra-Inter Cluster Distance Difference against a Control Feature Set:

Since a cluster algorithm like K-Means will by definition inherently maximize the difference between these two metrics, one could argue there is not too much purpose in collecting this score. In other words, in a sense, we are simply "validating" that the algorithm is doing what it's supposed to be doing, and that there were no errors. We will call this evaluation method "evaluating against its own feature set", since we are using the same feature set to run the "Intra-Inter Cluster Distance" evaluation against cluster results that came from the same feature set. Although this in itself is still a very good metric, since it determines how well separated clusters are, we can test the results further by cross-verifying with a completely separate feature set.

In order to rigorously test and evaluate our cluster results, we decided to take a further step and run the "Intra-Inter Cluster Distance" evaluation using a separate standardized feature set, which acts as a "control" feature set in experimentation. Put more simply, for instance, we will cluster players based on feature sets A, B, C, D; but we will always run the "Intra-Inter Cluster Distance" evaluation using the same feature set Z, called a controlled feature set. By doing so, getting positive "Intra-Inter Cluster Distance" scores will be much more difficult, but will prove to

be more meaningful, since it is basically saying that using feature set A players gives a largely similar cluster result as using a separate standardized feature set Z. Having two different feature sets agree with each other thereby acts as an additional layer of performance validation. The challenge here, of course, becomes determining a standardized feature set that acts as a good gauge to compare variable feature sets against. For this study, we chose to use the low-level data which emulates the same features used in the PCSP study to be the controlled feature set for each game component.

A sample of results from evaluation steps #1 and #2:

```
---Evaluating Cluster 0---
Size of Cluster: 70
Intra-cluster Distance Avg: 10.06
Inter-cluster Distance Avg: 27.78
% Difference (Inter - Intra) for Cluster 0: 176.10%
-------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 10
Intra-cluster Distance Avg: 10.14
Inter-cluster Distance Avg: 41.94
% Difference (Inter - Intra) for Cluster 1: 313.73%
-------------------------------------------------
---Evaluating Cluster 2---
Size of Cluster: 36
Intra-cluster Distance Avg: 9.80
Inter-cluster Distance Avg: 32.95
% Difference (Inter - Intra) for Cluster 2: 236.37%
-------------------------------------------------
---Evaluating Cluster 3---
Size of Cluster: 17
Intra-cluster Distance Avg: 10.68
Inter-cluster Distance Avg: 52.43
% Difference (Inter - Intra) for Cluster 3: 391.07%
-------------------------------------------------
Overall Average of % Difference (Inter-Intra): 279.32%
```

Evaluation #3: PCSP Player-Cluster Replacement Difference:

The final step to evaluate the quality of a cluster result is to simply input clustered data into the PCSP model in the PAT model checker. Since this is the objective of our paper, it also naturally makes sense as the main evaluation metric of our clustering results.

For example, we will first run the PCSP model for two chosen players, such as Roger Federer and Rafael Nadal. This is the existing process flow for Dong et al.'s study. This will produce winning probability ranges for both players that will act as a "baseline" result, such as [48-51%] winning chance for Federer, and [51-53%] chance for Nadal.

Next, we produce PCSP probability numbers for each of Federer's clusters. For example, if our results show that Federer belongs to cluster 1 for his Service game component, we will then replace all of Federer's probability distribution numbers for his "service game" function in PCSP with Serve cluster 1's probability distribution numbers.

PCSP model's serve function with Federer's numbers:

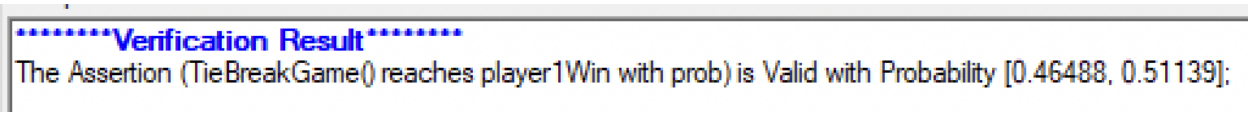
```
De_Ply1Serve = pcase {                                  // all probability is base
              231: ServeT_in{ball= 6} -> Ply2_BackHandR  // T will have opponer
              119: ServeT_err{ball=9} -> De_Ply1Serve_2nd   // Federer tries to
              337: ServeWide_in{ball =6} -> Ply2_ForeHandR
              173: ServeWide_err{ball=9} -> De_Ply1Serve_2nd
              92: ServeBody_in{ball=6} -> (Ply2_BackHandR [] Ply2_ForeHandR)
              48: ServeBody_err{ball=9} -> De_Ply1Serve_2nd};
```

PCSP model's serve function with Cluster 1's numbers:

```
De_Ply1Serve = pcase {                                  // all probability is base
              231: ServeT_in{ball= 6} -> Ply2_BackHandR  // T will have opponer
              119: ServeT_err{ball=9} -> De_Ply1Serve_2nd   // Federer tries to
              337: ServeWide_in{ball =6} -> Ply2_ForeHandR
              173: ServeWide_err{ball=9} -> De_Ply1Serve_2nd
              92: ServeBody_in{ball=6} -> (Ply2_BackHandR [] Ply2_ForeHandR)
              48: ServeBody_err{ball=9} -> De_Ply1Serve_2nd};
```

Next, we repeat the above steps for the other game components - Return game and Rally game. Lastly, we will repeat all above steps for the opposing player ("Ply2" in the PCSP functions), by replacing all of Rafael Nadal's numbers from the "Player 2" functions in the PCSP, with numbers from Nadal's cluster for each game component.

Finally, we can run the PCSP verification function to check the winning probabilities of Federer's cluster vs Nadal's cluster by running the verification functions "`TieBreakGame() reaches player1Win with prob`" and "`TieBreakGame() reaches player2Win with prob`" in the PAT model checker. A sample of the verification results obtained from the PAT model checker:



This indicates that player 1, Federer, has a win probability between **46.5% - 51.1%** against player 2, Nadal. After recording results for both player's clusters, we will then record the averages of both player's winning chances, and check the difference of results between this and the baseline winning probabilities between the individual players themselves.

(Winning % Federer vs Nadal) vs (Winning % of Federer's Clusters vs Nadal's Clusters)

On top of running the above experiment, as a controlled experiment, we should also compare the results from players not in the cluster of either of the players. To do so, we will also run the following scenarios:

- Federer's Cluster vs. Not-Nadal's Cluster
- Not-Federer's Cluster vs. Nadal's Cluster
- Not-Federer's Cluster vs. Not-Nadal's Cluster

Since there can be many total combinations of the above, we will simply choose a random cluster that fulfills each condition for each scenario to complete our experiment. With these additional scenarios done, we can then compare whether our initial simulation of the Cluster vs Cluster scenario generates relatively more similar results to the baseline scenario as compared to the other scenarios. If the winning probability predictions are nearer to the baseline model's, we can conclude that the clusters form a good enough representation of the players themselves, to be able to be used in the PCSP model to obtain insightful results.

6. Experiment Results

For illustration's sake, we will only walk through the experiment process and results in detail for the **Service game component**, which can be extended to the other game components since they follow the same experimental procedures. They can also be referenced in **Appendix 5**.

For the Service component, we will focus on evaluating the cluster results that resulted from using the following feature set for clustering:

- High-level data: the original 18 Serve features from Tennis-Abstract.com's summary data. The definitions of each feature can be found in the data dictionary in **Appendix 2**.

Player	Unret%	<=3 W%	RiP W%	SvImpact	1st: Unret%	<=3 W%.1	RiP W%.1	SvImpact.1	D Wide%	A Wide%	BP Wide%	2nd: Unret%	<=3 W%.2	RiP W%.2	D Wide%.1	A Wide%.1	BP Wide%.1	2ndAgg
Ivo Karlovic	0.48	0.632	0.509	0.553	0.59	0.748	0.57	0.701	0.429	0.569	0.575	0.316	0.48	0.427	0.264	0.302	0.143	126
John Isner	0.454	0.59	0.482	0.52	0.54	0.671	0.508	0.633	0.5	0.451	0.585	0.268	0.43	0.44	0.211	0.519	0.69	33
Reilly Opelka	0.457	0.582	0.483	0.511	0.571	0.685	0.502	0.651	0.5	0.432	0.424	0.26	0.418	0.459	0.333	0.542	0.804	33
Goran Ivanisevic	0.421	0.539	0.498	0.474	0.589	0.721	0.554	0.686	0.381	0.55	0.568	0.263	0.383	0.454	0.169	0.517	0.656	119
Milos Raonic	0.397	0.54	0.516	0.471	0.52	0.681	0.573	0.638	0.536	0.502	0.455	0.214	0.341	0.45	0.281	0.544	0.718	76
Richard Krajicek	0.402	0.536	0.478	0.468	0.549	0.698	0.567	0.665	0.457	0.399	0.278	0.239	0.37	0.399	0.124	0.498	0.7	60
Sam Querrey	0.403	0.52	0.487	0.461	0.542	0.678	0.527	0.638	0.478	0.474	0.538	0.206	0.306	0.446	0.37	0.482	0.42	69
Nick Kyrgios	0.403	0.511	0.505	0.46	0.511	0.63	0.54	0.597	0.427	0.411	0.583	0.225	0.322	0.458	0.301	0.498	0.437	98

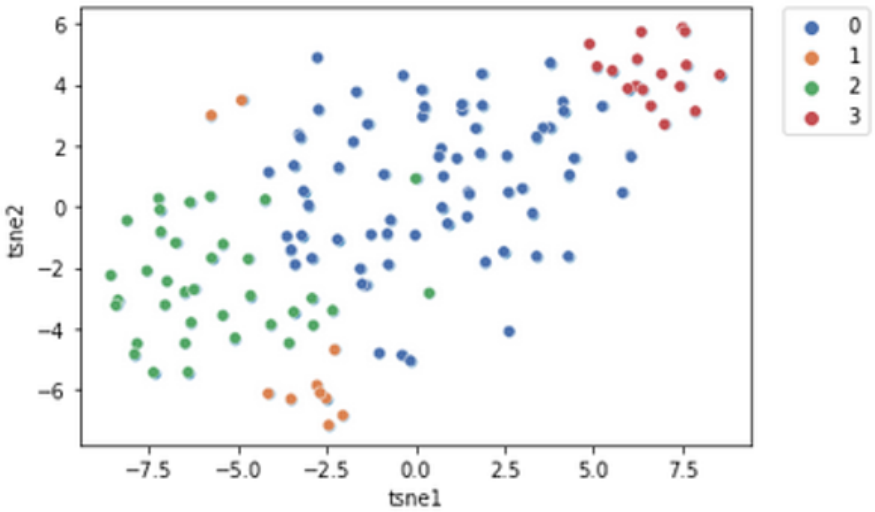
6.1. Evaluation #1: Intra-Inter Cluster Distance Difference using Same Feature Set

Game Component: **Service game**
Experiment Type: **Agglomerative, Ward linkage, 4 clusters**
Clustering Feature Set: **18 Tennis-Abstract features**
Evaluation Feature Set: **18 Tennis-Abstract features**

```
[3 3 3 3 3 3 3 3 3 3 3 0 0 3 3 3 3 3 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 2 0 0 0 0 2 1 0 2 0
 0 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 2 0 2 2 2 0 2 2 2 2 2 2 2 2 0 0 2 0 2 2
 2 2 2 2 1 2 2 1 1 2 1 2 1 2 2 1 1 1 2 2 2 1]
0       70
2       36
3       17
1       10
Cluster Results:
---Evaluating Cluster 0---
Size of Cluster: 70
Intra-cluster Distance Avg: 10.06
Inter-cluster Distance Avg: 27.78
% Difference (Inter - Intra) for Cluster 0: 176.10%
-------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 10
```

```
Intra-cluster Distance Avg: 10.14
Inter-cluster Distance Avg: 41.94
% Difference (Inter - Intra) for Cluster 1: 313.73%
-------------------------------------------------
---Evaluating Cluster 2---
Size of Cluster: 36
Intra-cluster Distance Avg: 9.80
Inter-cluster Distance Avg: 32.95
% Difference (Inter - Intra) for Cluster 2: 236.37%
-------------------------------------------------
---Evaluating Cluster 3---
Size of Cluster: 17
Intra-cluster Distance Avg: 10.68
Inter-cluster Distance Avg: 52.43
% Difference (Inter - Intra) for Cluster 3: 391.07%
-------------------------------------------------
Overall Average of % Difference (Inter-Intra): 279.32%
```

To interpret the results, the first 4 lines above show us which cluster label was assigned to each player. The next 4 lines give us a summary of the number of players in each cluster. Afterwards, we see the evaluation metrics for each cluster. For example, for Cluster 0, there are 70 players, and the difference between Inter and Intra cluster distances are **176%**. To be clear, any positive % number indicates that the average inter-cluster distance is larger than the average intra-cluster distance. Hence a number larger than 100% means the inter-cluster distance is double of the intra-cluster distance, which is an excellent indicator of well-separated clusters. Next we generate a t-SNE plot, which first reduces the dimensionality to two components, and plots each player as a point in a colour-coded fashion. This gives us a good visual cue of how well the clusters were formed.



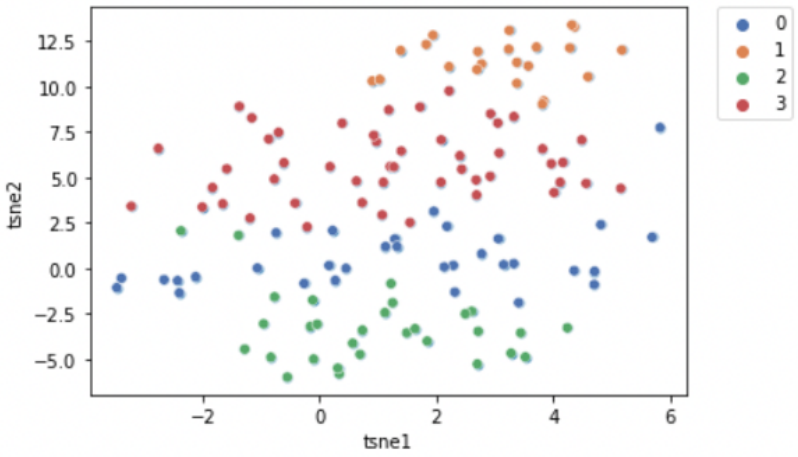
	Cluster	Size of Cluster	Intra-cluster Distance Avg	Inter-cluster Distance Avg	% Difference (Inter - Intra)
0	0.0	70.00	10.060416	27.776937	176.101275
1	1.0	10.00	10.136118	41.935996	313.728382
2	2.0	36.00	9.795738	32.949838	236.369122
3	3.0	17.00	10.676474	52.429086	391.071148
Overall Average		33.25	10.167187	38.772964	279.317482

Comment: The % difference between Inter- and Intra- cluster distance is quite large for every cluster (eg. over 100%). This is great because this means the intra-cluster distance is very small compared to inter-cluster distance, meaning every cluster is well separated. Visually, the t-SNE plot also gives us a reasonable sense-check that the clusters were formed correctly. However, this is to be expected, since the algorithm is using Euclidean distance to determine the successive grouping of clusters, for each feature in our feature set. Hence, the positive results simply confirm that the algorithm is working as expected. Next, we repeat the same experiment using the K-Means algorithm:

Game Component: **Service Game**
Experiment Type: **K-Means, init=100 iterations, 4 clusters**
Clustering Feature Set: **18 Tennis-Abstract features**
Evaluation Feature Set: **18 Tennis-Abstract features**

```
[1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 3 1 1 3 3 3 1 1 3 3 3 3 1 3 3 3 3 3 3
 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 0 2 3 3 3 3 2 0 0 2 0
 0 2 0 3 3 0 0 3 3 0 3 0 0 2 0 0 2 2 0 0 2 2 0 2 2 2 2 0 0 2 2 0 0 2 0 2 2
 2 2 0 2 0 2 2 0 0 2 0 0 0 2 2 0 0 0 2 2 2 0]
3      49
0      32
2      30
1      22
Cluster Results:
---Evaluating Cluster 0---
Size of Cluster: 32
```

```
Intra-cluster Distance Avg: 10.21
Inter-cluster Distance Avg: 29.73
% Difference (Inter - Intra) for Cluster 0: 191.12%
-------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 22
Intra-cluster Distance Avg: 10.88
Inter-cluster Distance Avg: 48.06
% Difference (Inter - Intra) for Cluster 1: 341.83%
-------------------------------------------------
---Evaluating Cluster 2---
Size of Cluster: 30
Intra-cluster Distance Avg: 9.68
Inter-cluster Distance Avg: 33.70
% Difference (Inter - Intra) for Cluster 2: 248.04%
-------------------------------------------------
---Evaluating Cluster 3---
Size of Cluster: 49
Intra-cluster Distance Avg: 8.67
Inter-cluster Distance Avg: 25.16
% Difference (Inter - Intra) for Cluster 3: 190.15%
-------------------------------------------------
Overall Average of % Difference (Inter-Intra): 242.79%
```



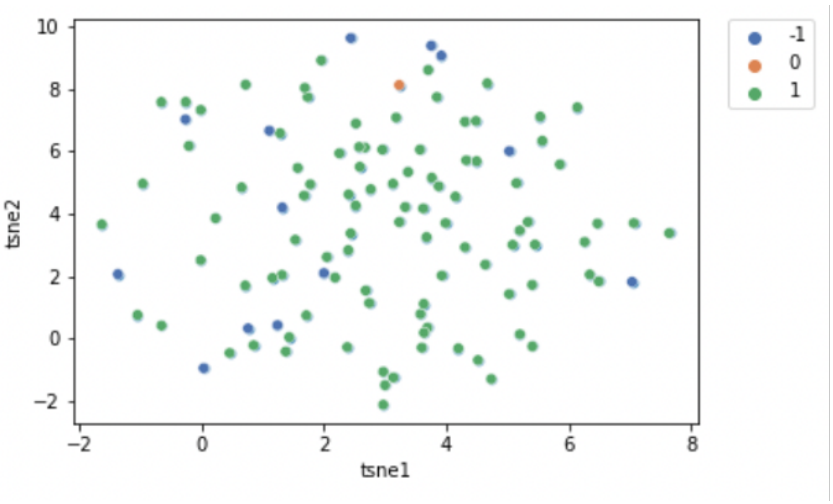
	Cluster	Size of Cluster	Intra-cluster Distance Avg	Inter-cluster Distance Avg	% Difference (Inter - Intra)
0	0.0	32.00	10.211176	29.726521	191.117518
1	1.0	22.00	10.878137	48.063281	341.833770

2	2.0	30.00	9.682951	33.700189	248.036357
3	3.0	49.00	8.670093	25.156656	190.154397
Overall Average		33.25	9.860589	34.161662	242.785511

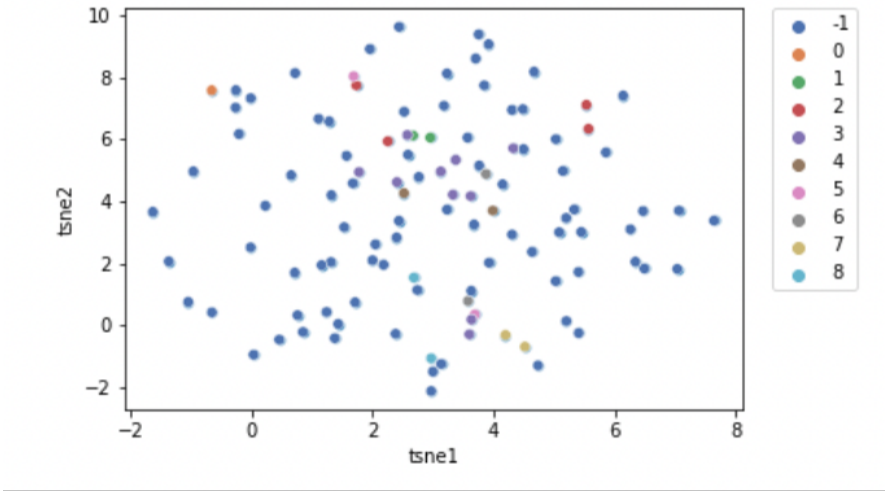
Comment: Once again, all the Inter-Intra cluster distance numbers are all >100%, which again is an indicator that each cluster is well separated, and there isn't too much ambiguity. Once again, since we are using the same feature set from our clustering to do this evaluation, the positive numbers simply confirm the algorithm works as expected. Next, we try using the DBScan algorithm:

Game Component: **Service Game**
Experiment Type: **DBScan, epsilon = 3**
Clustering Feature Set: **18 Tennis-Abstract features**
Evaluation Feature Set: **18 Tennis-Abstract features**

```
 1      113
-1      18
 0      2
---Evaluating Cluster 0---
Size of Cluster: 2
Intra-cluster Distance Avg: 1.4326
Inter-cluster Distance Avg: 77.2583
% Difference (Inter - Intra) for Return Cluster 0: 5292.8604%
-------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 113
Intra-cluster Distance Avg: 13.5387
Inter-cluster Distance Avg: 44.0564
% Difference (Inter - Intra) for Return Cluster 1: 225.4101%
-------------------------------------------------
---Evaluating Cluster 2---
Invalid results
Size of Cluster: 0
Intra-cluster Distance Avg: nan
Inter-cluster Distance Avg: nan
% Difference (Inter - Intra) for Return Cluster 2: nan%
-------------------------------------------------
```



Comment: Most players fall into Cluster 1, which is not insightful for our purposes. This could be due to the fact that our Epsilon value is set to too large of a value, which can cause most data points to fall into the same cluster. To try again, we repeat the experiment using a smaller Epsilon value = 2.



Comment: Now, most players fall into the "-1" cluster, which means they do not belong to any cluster. Other than that, there are 9 other small clusters formed, each with about 2 players inside. From these experiments, we can conclude DBScan may not be the appropriate algorithm for our purposes. As mentioned earlier, this may be due to the fact that all tennis players' play styles are more similar than they are different, such that a density-based algorithm would simply classify most of them into the same cluster, or if the Epsilon value is too small, all into separate small clusters.

6.2. Evaluation #2: Evaluating against a Control Feature Set

As mentioned earlier, the large positive numbers in our first evaluation method simply confirm our algorithm is working properly, and that the clusters are well-spaced apart. In order to further test the quality of cluster results, we try using a different feature set as the controlled feature set. Here, we use the 36 low-level features that are identical to the 36 features used in the PCSP function. A sample of 10 of the 36 features are shown below:

	feature	Player 1 name
0	1st serve-deuce court-serve to body-fault	0.028941
1	1st serve-deuce court-serve to body-no outcome	0.077920
2	1st serve-deuce court-serve to body-service wi...	0.006855
3	1st serve-deuce court-serve to wide-fault	0.149017
4	1st serve-deuce court-serve to wide-no outcome	0.244477
5	1st serve-deuce court-serve to wide-service wi...	0.041445
6	1st serve-deuce court-server to T-fault	0.192752
7	1st serve-deuce court-server to T-no outcome	0.202980
8	1st serve-deuce court-server to T-service winn...	0.055612
9	1st serve-ad court-serve to body-fault	0.028903
10	1st serve-ad court-serve to body-no outcome	0.071141

For comparison, a sample of features from the original PCSP function are shown below:

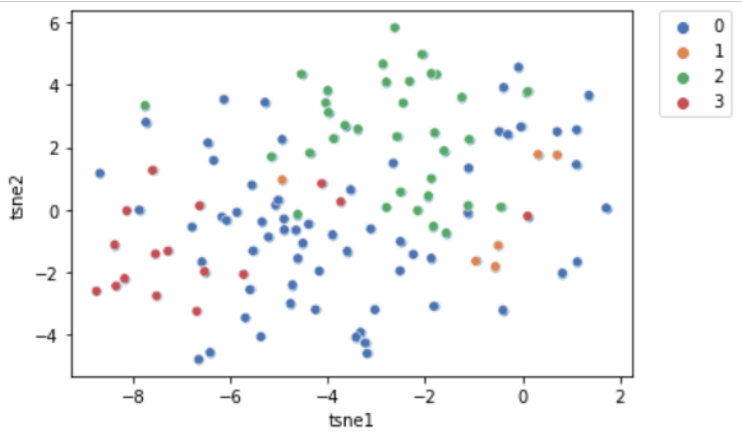
```
De_Ply1Serve = pcase {                                  // all probability is bas
               231: ServeT_in{ball= 6} -> Ply2_BackHandR  // T will have oppone
               119: ServeT_err{ball=9} -> De_Ply1Serve_2nd   // Federer tries t
               337: ServeWide_in{ball =6} -> Ply2_ForeHandR
               173: ServeWide_err{ball=9} -> De_Ply1Serve_2nd
               92: ServeBody_in{ball=6} -> (Ply2_BackHandR [] Ply2_ForeHandR)
               48: ServeBody_err{ball=9} -> De_Ply1Serve_2nd};
```

For example, in the PCSP function named "`De_Ply1Serve`", the 1st feature named `ServeT_in{ball=6}` corresponds directly to the feature named "`1st serve-deuce court-serve to T-no outcome`" in our evaluation feature set. This was done by design since we want the evaluation set to resemble the features used in the PCSP as closely as possible.

Game Component: **Service Game**
Experiment Type: **Agglomerative, Ward linkage, 4 clusters**
Clustering Feature Set: **18 Tennis-Abstract features**
Evaluation Feature Set: **36 low-level features resembling PCSP**

```
[3 3 3 3 3 3 3 3 3 3 0 0 3 3 3 3 3 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 2 0 0 0 2 0 0 0 0 2 1 0 2 0
 0 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 2 0 2 2 2 0 2 2 2 2 2 2 2 2 0 0 2 0 2 2
 2 2 2 2 1 2 2 1 1 2 1 2 1 2 2 1 1 1 2 2 2 1]
0    70
2    36
3    17
1    10
Cluster Results:
---Evaluating Cluster 0---
Size of Cluster: 57
Intra-cluster Distance Avg: 28.14
Inter-cluster Distance Avg: 46.32
% Difference (Inter - Intra) for Cluster 0: 64.59%
-------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 7
Intra-cluster Distance Avg: 24.95
Inter-cluster Distance Avg: 55.82
% Difference (Inter - Intra) for Cluster 1: 123.70%
-------------------------------------------------
---Evaluating Cluster 2---
Size of Cluster: 33
Intra-cluster Distance Avg: 27.58
Inter-cluster Distance Avg: 49.83
% Difference (Inter - Intra) for Cluster 2: 80.68%
-------------------------------------------------
---Evaluating Cluster 3---
Size of Cluster: 14
Intra-cluster Distance Avg: 45.72
Inter-cluster Distance Avg: 54.84
% Difference (Inter - Intra) for Cluster 3: 19.93%
-------------------------------------------------
Overall Average of % Difference (Inter-Intra): 72.22%
```

Comment: Evidently our Overall Average of % Difference (Inter-Intra) metric is much smaller than the ones seen from Evaluation #1, but considering we are essentially evaluating with a completely different feature set, any positive value is an encouraging result, as it indicates that the clusters from our clustering feature set largely agree with clusters that would have been formed using the evaluation feature set. We will also run the K-Means algorithm to see which feature set gives a higher output value for this metric.



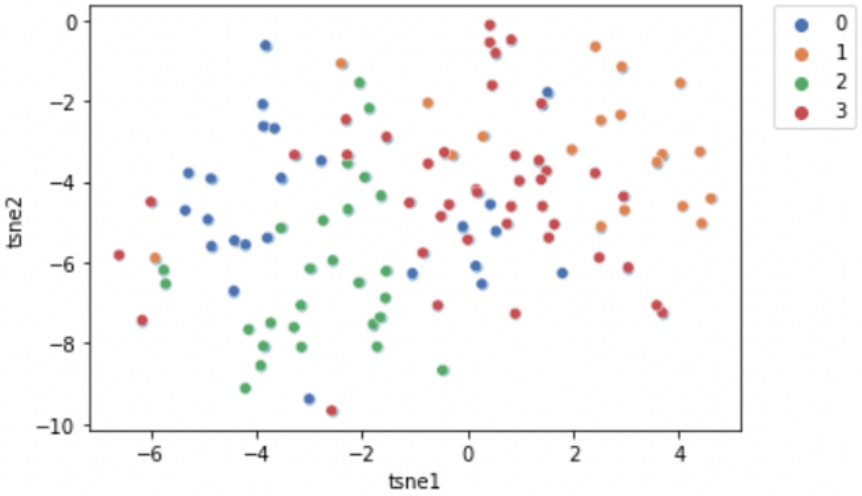
Comment: Visually, we can see that the new colour-coded clusters are still largely clustered correctly, even though the colours are now according to a different feature set. This confirms the positive **72.22%** metric output we saw in the output above.

	Cluster	Size of Cluster	Intra-cluster Distance Avg	Inter-cluster Distance Avg	% Difference (Inter - Intra)
0	0.0	57.00	28.141701	46.318048	64.588661
1	1.0	7.00	24.954416	55.822499	123.697875
2	2.0	33.00	27.579535	49.831647	80.683417
3	3.0	14.00	45.723782	54.836351	19.929605
Overall Average	1.5	27.75	31.599859	51.702136	72.224890

Game Component: **Service Game**
Experiment Type: **K-Means, init=100 iterations, 4 clusters**
Clustering Feature Set: **18 Tennis-Abstract features**
Evaluation Feature Set: **36 low-level features resembling PCSP**

```
[1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 3 1 1 3 3 3 1 1 3 3 3 3 1 3 3 3 3 3 3
 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 0 2 3 3 3 3 2 0 0 2 0
 0 2 0 3 3 0 0 3 3 0 3 0 0 2 0 0 2 2 0 0 2 2 0 2 2 2 2 0 0 2 2 0 0 2 0 2 2
 2 2 0 2 0 2 2 0 0 2 0 0 0 2 2 0 0 0 2 2 2 0]
3    49
0    32
2    30
1    22
Cluster Results:
---Evaluating Cluster 0---
Size of Cluster: 24
Intra-cluster Distance Avg: 28.49
Inter-cluster Distance Avg: 44.78
% Difference (Inter - Intra) for Cluster 0: 57.15%
-------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 19
Intra-cluster Distance Avg: 44.95
Inter-cluster Distance Avg: 47.78
```

```
% Difference (Inter - Intra) for Cluster 1: 6.28%
-------------------------------------------------
---Evaluating Cluster 2---
Size of Cluster: 27
Intra-cluster Distance Avg: 28.49
Inter-cluster Distance Avg: 50.40
% Difference (Inter - Intra) for Cluster 2: 76.94%
-------------------------------------------------
---Evaluating Cluster 3---
Size of Cluster: 41
Intra-cluster Distance Avg: 25.21
Inter-cluster Distance Avg: 45.01
% Difference (Inter - Intra) for Cluster 3: 78.53%
-------------------------------------------------
Overall Average of % Difference (Inter-Intra): 54.72%
```



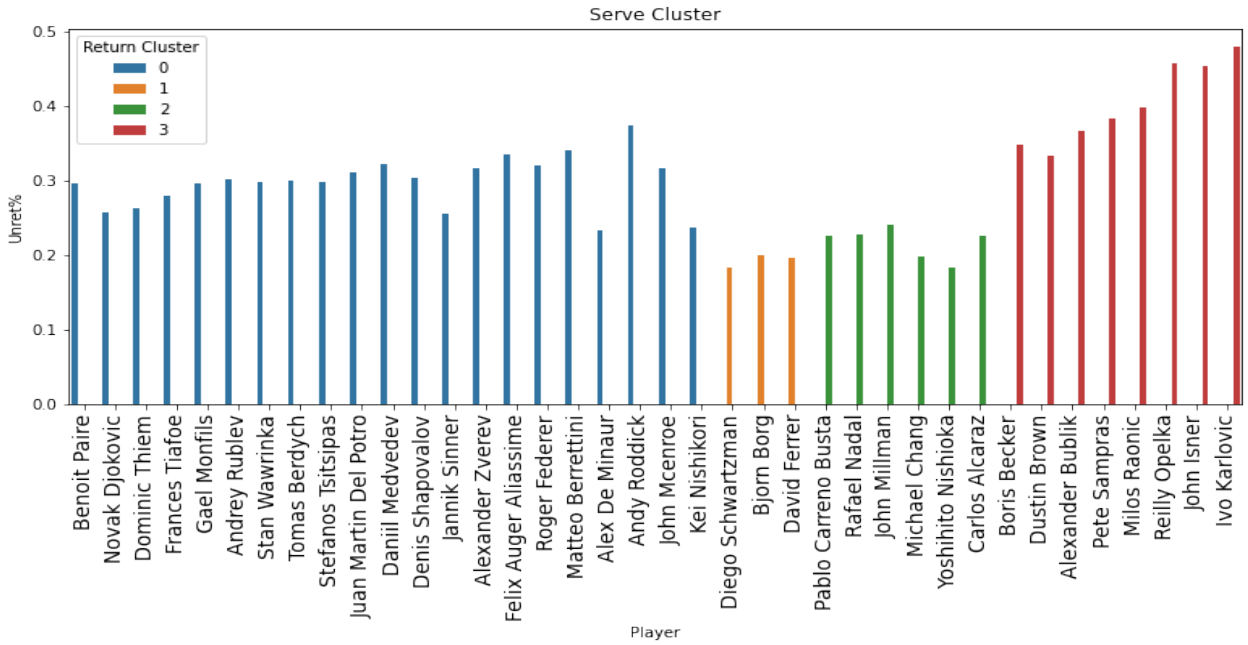
	Cluster	Size of Cluster	Intra-cluster Distance Avg	Inter-cluster Distance Avg	% Difference (Inter - Intra)
0	0.0	24.00	28.493507	44.776580	57.146606
1	1.0	19.00	44.952389	47.776558	6.282578
2	2.0	27.00	28.485983	50.403024	76.939738
3	3.0	41.00	25.211976	45.010399	78.527850
Overall Average	1.5	27.75	31.785964	46.991640	54.724193

Comment: From our K-Means clustering, it is still encouraging to see a very positive value (**54.72%**) for our metric Overall Average of % Difference (Inter-Intra). Also, visually, the clusters still follow the correct colours for the most part. Comparatively, the output metric is lower than the result we got from using Agglomerative clustering (**72.22%**). Hence, for the Service game component, we will opt to use the clustering results obtained from our Agglomerative clustering algorithm. For evaluation results for Return and Rally game, please reference **Appendix 5**.

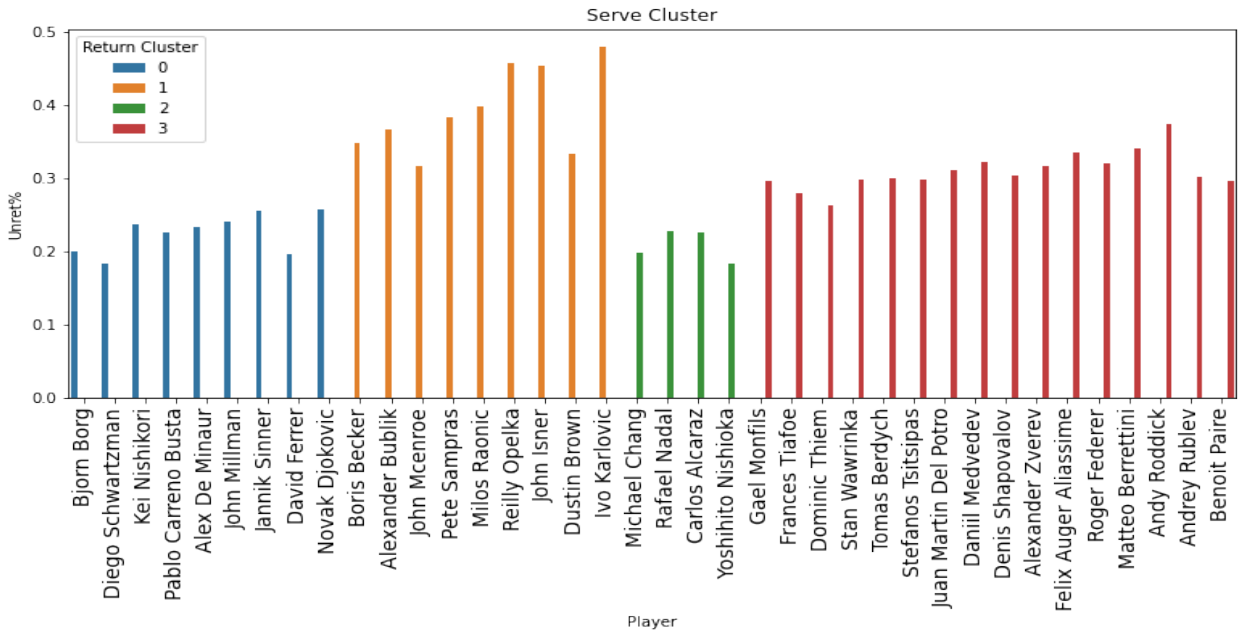
6.3. Cluster Grouping Results

As an example, below is the player clustering result for the Service game component, output from both K-Means, and Agglomerative methods, using the same feature set. We filtered to display only 38 players for visualization and ease of inference purposes:

Service Cluster Results from Agglom:



Service Cluster Results from K-Means:



Comment: For the Agglomerative method, it is interesting to see that Roger Federer and Novak Djokovic both fall into Cluster 0, while Rafael Nadal is in Cluster 2. This is generally in line with intuition since Nadal is generally known to have a less impactful serve than the other two, also confirmed by a lower "Unreturned serves %" feature metric, denoted by the height of the bars. For the K-Means method, it is interesting to note that all 3 players fall into separate clusters. Lastly, it is also an encouraging sign to see all the players who are considered "big servers" fall into the same cluster (Cluster 3 in Agglomerative, and Cluster 1 in K-Means). Players such as John Isner, Reilly Opelka, Ivo Karlovic, Milos Raonic, and Pete Sampras are all well known for their tremendous serves, which can be confirmed by each of their high "Unret %" bars. The rest of the clustering group results for the other two game components can be referenced in **Appendix 4**.

Summary for Evaluations #1 and #2:

Serve Cluster				
Evaluation #1	**Cluster Method**	**Cluster Feature Set**	**Evaluation Feature Set**	**Overall Average of % Difference (Inter-Intra)**
	Agglomerative, Ward linkage, 4 clusters	19 summary-level features	19 summary-level features	279.32%
	Kmeans, init=100, 4 clusters	19 summary-level features	19 summary-level features	242.79%
Evaluation #2	**Cluster Method**	**Cluster Feature Set**	**Evaluation Feature Set**	**Overall Average of % Difference (Inter-Intra)**
	Agglomerative, Ward linkage, 4 clusters	19 summary-level features	36 PCSP serve function features	72.22%
	Kmeans, init=100, 4 clusters	19 summary-level features	36 PCSP serve function features	54.72%

Return Cluster				
Evaluation #1	Cluster Method	Cluster Feature Set	Evaluation Feature Set	Overall Average of % Difference (Inter-Intra)
	Agglomerative, Ward linkage, 4 clusters	16 summary-level features	16 summary-level features	222.54%
	Kmeans, init=100, 4 clusters	16 summary-level features	16 summary-level features	254.57%
Evaluation #2	Cluster Method	Cluster Feature Set	Evaluation Feature Set	Overall Average of % Difference (Inter-Intra)
	Agglomerative, Ward linkage, 4 clusters	16 summary-level features	72 PCSP return function features	14.85%
	Kmeans, init=100, 4 clusters	16 summary-level features	72 PCSP return function features	24.39%

Rally Cluster				
Evaluation #1	Cluster Method	Cluster Feature Set	Evaluation Feature Set	Overall Average of % Difference (Inter-Intra)
	Agglomerative, Ward linkage, 4 clusters	13 summary-level features	13 summary-level features	173.18%
	Kmeans, init=100, 4 clusters	13 summary-level features	13 summary-level features	186.19%
Evaluation #2	Cluster Method	Cluster Feature Set	Evaluation Feature Set	Overall Average of % Difference (Inter-Intra)
	Agglomerative, Ward linkage, 4 clusters	13 summary-level features	24 PCSP rally function features	34.66%
	Kmeans, init=100, 4 clusters	13 summary-level features	24 PCSP rally function features	31.90%

Comments:

- For the Service game component, we will use the results obtained from the Agglomerative method, since the results were higher than K-Means in both evaluation methods, at 279.32% and 72.22%. This indicates the clusters are well separated, and they agree with the controlled feature set.
- For the Return game component, K-Means performed slightly better than Agglomerative in both evaluation methods, at 254.57% and 24.39%. Hence, we will use the results from the K-Means method.
- For the Rally game component, we have a tie between both methods since K-Means performed slightly better in evaluation #1 with 186.19%, while Agglomerative performed better in evaluation #2 with 34.66%. To break the tie, we can conclude evaluation #2 holds more weight, since it has a much stricter evaluation criteria. Therefore, we will use the results from the Agglomerative method.

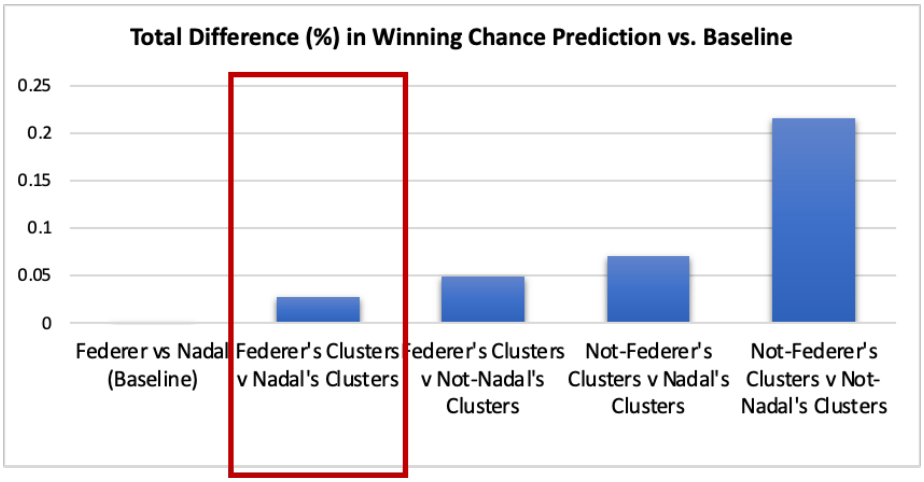
6.4. Evaluation #3: PCSP Player-Cluster Replacement Difference

The final stage of our evaluation involves taking the numbers from each of our cluster results per game component, and inputting them back into the original PCSP module functions to replace the individual player's numbers. For this study, we will focus on Roger Federer and Rafael Nadal as our primary targets in our baseline scenario.

Our baseline scenario, which is highlighted in green in the table below, produced an average winning probability of **49.51**% for Roger Federer and **50.49%** for Rafael Nadal, which is the average taken from the Min and Max ranges returned by the PAT verification function. Afterwards, we replaced the individual numbers with the cluster numbers and produced the results highlighted in yellow below, with Federer's clusters winning at **48.8%** average, and Nada's clusters winning at **51.2%** average. This gives us a sum of **2.67%** difference from the baseline model. But to preserve a controlled experiment, we also ran 3 more scenarios with combinations such as Federer's clusters vs. Not-Nadal's clusters; Not-Federer's clusters vs. Nadal's clusters; and lastly, Not-Federer's clusters vs. Not-Nadal's clusters.

				Game components			Results				Difference from Baseline		
#	Test Case	Ply1	Ply2	Serve	Return	Rally	Ply1 prob	Ply1 2 prob	Ply1 Avg	Ply2 Avg	Ply1 Difference	Ply2 Difference	Total Difference
0	**Baseline**	Federer	Nadal	Fed vs Nadal	Fed vs Nadal	Fed vs Nadal	[48.084, 50.94]	[49.059,51.915]	0.49512	0.50487	-	-	-
1	**Cluster v Cluster**	Federer Cluster	Nadal Cluster	Fed's cluster (3) vs Nadal's cluster (2)	Fed's cluster (1) vs Nadal's cluster (1)	Fed's cluster (0) vs Nadal's cluster (1)	[0.46736, 0.5087];	[0.4913, 0.53263];	0.4880	0.5120	1.43%	-1.41%	**2.67%**
2	Cluster v Non-Cluster	Federer Cluster	Non-Nadal Cluster	Fed cluster (3) vs Isner cluster (1)	Fed cluster (1) vs Isner cluster (0)	Fed cluster (0) vs Medvedev cluster (2)	[0.48341, 0.53201];	[0.46798, 0.51658];	0.5077	0.4923	-2.54%	2.49%	**4.91%**
3	Non-Cluster v Cluster	Non-Federer Cluster	Nadal Cluster	John Isner cluster (1) vs Nadal cluster (2)	John Isner cluster (0) vs Nadal cluster (1)	Medvedev cluster (2) vs Nadal cluster (1)	[0.45157, 0.50242];	[0.49757, 0.54842];	0.4770	0.5230	3.66%	-3.59%	**7.07%**
4	Non-Cluster v Non-Cluster	Non-Federer Cluster	Non-Nadal Cluster	John Isner cluster (1) vs Mats Wilander cluster (0)	John Isner cluster (0) vs Karlovic cluster (2)	Medvedev cluster (2) vs Sampras cluster (3)	[0.51802, 0.58273];	[0.41726, 0.48197];	0.55038	0.44962	-11.16%	10.94%	**21.55%**

After running all the scenarios, we obtained our final results summarized in the table above. The “Total Difference” metric summarizes the sum of differences in winning chance % for both players 1 and 2, between the baseline model against every scenario. As observed, Test Case #1 (Cluster vs. Cluster) has the narrowest range of differences and the closest to the baseline scenario. It achieves the lowest difference for winning chance % compared to the baseline model, with a total difference of **2.67%**, while the other 3 scenarios have a significantly higher difference of **4.91%** or more. These results confirm that the clustered representation of both players essentially provides an acceptable representation of the match outcome between the two individual players.



Based on the findings in this paper, we can conclude that the clustering methods using the concept of game components and our thorough 3-step evaluation approach can give a very strong representation of the individual players when predicting the match outcome through probabilistic model checking in the PAT model checker. For the next steps, we can expand our scope and repeat this experiment for a number of other players to see if the clustering is successful to produce similar PCSP winning probability results.

7. Discussion and Future Work

While conducting this study, it became evident that there is a lot more potential in the area of research for sports analytics. The iterative approach that we chose to take to come up with the most optimal feature sets and cluster results was rewarding in the end as we were able to obtain encouraging results. In the end, we also built a tested and thorough experiment pipeline that can be repeatedly run for a variety of different factors and variables. In this section, we will discuss what other elements can be improved, tested, and implemented in future work.

Given more time, we can try more combinations of feature sets from our data. Even though we tested a good amount of feature sets in this study, we can still create a much larger combination of features in a more systematic way to be tested with our 3-step evaluation process. For example, in our Return game component, we tested a set of feature sets with 72 features, 36 features, and 24 features. We can repeat this process in a systematic way for all game components, and then iteratively add or subtract certain features that prove to improve the metrics in our evaluation phase.

There are a number of other variables we can try experimenting with as well, including:

- **Number of Players:**

  The number of players clustered in our study included 133 players, which was directly taken from the high-level data from Tennis-Abstract. But due to the large fluctuations in players from different eras, and also the number of matches recorded, it could have an impact on our cluster results. For example, we can try clustering with a varying number of players, such as the top 50 or 30 players, to see if this has an effect in our study. Alternatively, we can set a minimum game threshold for players or a specific year range to be selected. By doing this, there will be less unexpected variation or noise for players who only have a few matches' of data. A tighter year range can also ensure that a varying era in tennis doesn't impact results.

- **Isolating Left-handed players:**

  Our study did not take into account the handed-ness of a player, which can evidently impact our cluster results. From a data-perspective, left-handed players may look like a drastically different play style, even though the results may only be flipped. However, this element needs more research to be done to decide which approach is the best to deal with handed-ness in a cluster problem. For example, from our cluster results,

we observed that all left-handed players ended up in the same cluster for the Rally component, in which it had no right-handed players. This comes as no surprise, and is an indicator that our clustering was working properly, but we should decide whether to simply treat left-handed players as a separate cluster, or to study them separately since it is technically not a “play style”.

- **Data Variety:**

  In this study, we only had access to a limited variety of data, which were extracted from the Tennis-Abstract website. If we had more variety of data, there will no doubt be significant improvements in clustering performance. For example, data such as ball speed, ball rotation speed, or location data, such as specific player location and specific ball location, can make clustering much easier. For instance, ball rotation and ball speed can distinguish a player who prefers to use a lot of top-spin in their shots, or a player who prefers slicing the ball more using back-spin. This data can potentially be sourced from the Hawk-Eye system if it becomes open-sourced, computer vision scripts, or other online sources.

- **More Game Components:**

  In this study, we chose to have 3 game components to represent each player’s play style, but in future studies, we can expand to having more game components that reflect different parts of the game. For example, we can have a “tactics” component, where features focus more on certain tactics that players like to take. A few examples of tactics may be “serve-and-volley”, “baseline player”, or “one-handed backhand player”. Another possible component may be “high-stress performance”, where we cluster players based on how they perform in “clutch” scenarios, such as during breakpoints. Having these extra components may bring more insight into clustering and match prediction accuracy as well as connecting the raw data to how we see the game intuitively.

- **Correlation of Game Components:**

  It would be interesting to see the correlation of game components for each player. For example, if Player A and Player B both fall into Serve Cluster 1, what is the likelihood that their Return Cluster and Rally Cluster are also the same one. Having this insight can tell us how correlated each game component is, and whether they are intuitively supported by the domain knowledge of the game. For example, we can ask questions such as “if a player is a great server, does that indicate that he is also a great returner?”

- **Number of Clusters:**

We used 4 clusters per game component, which was taken from our K-Means elbow method results, but the optimal number of clusters may be different depending on the game component. This is also another variable we can adjust in future studies.

- **Skill level:**

  One factor that can benefit the match prediction accuracy would be to scale the player's features with a skill-level factor. This would make sense, since play styles should ideally be separate from skill-level. For example, if a top-50 player was to play against an amateur player, their distinctive play styles should have a lesser impact than the raw skill-level in dictating who would win the match. A plausible data point to input as a feature can be the Elo rating, which is a popular rating system used in games such as chess.

- **Recency of Data:**

  Similar to skill-level, how a player performs yesterday vs. how he performs 10 years ago may be quite different. If we weigh a player's entire dataset the same way over an extensive period of time, we may get results that are outdated. To counter this, we can introduce a "date" scale, which gives more weight to more recent data.

8. Appendices

8.1. Appendix 1 - Data Dictionary for Low-level Dataset

Extracted and processed from Tennis-Abstract's Github page (showing main fields used only) [20]:

Field Name	Feature Definition
Player 1 name	Player's name
Player 2 name	Opponent's name
Player 1 handness	Dominant hand of player (RH for Right-hand, LH for Left-hand)
Player 2 handness	Dominant hand of opponent
Date	Date of match
Tournament Name	Name of tournament
Shot Type	Type of shot (1=1st serve, 2=2nd serve, 3=return, 4=rally)
From which court	Court location of when ball is hit by Player 1 (1=deuce court, 2=middle court, 3=ad court, 99=unknown)
Shot	Shot description (1~20 are forehand shots, 21~40 are backhand shots, 41 is trick shot, 99=unknown), detailed codes for all shots are:
Direction (serve)	Serve direction (1=deuce court, 2=middle court, 3=ad court, 4=serve to wide, 5=serve to body, 6=server to T, 99=unknown)
To which court	Shot direction (1=deuce court, 2=middle court, 3=ad court, 99=unknown)
Depth	(1=shallow, 2=deep, 3=very deep, 99=unknown)
Touched Net?	(1=yes touched net, 2=not)
Hit at what depth?	(1=at net, 2=at baseline, 99=unknown)
Approach shot?	(1=yes, 2=no)
Shot outcome	Outcome of shot (1=ace, 2=fault, 3=forced error, 4=unforced error, 5=-winner, 6=service winner, 7=no outcome)
Fault type	Fault type, if Shot Type = serve (1=(net), 2=(wide), 3=(long), 4=(wide and long), 5=(foot fault), 6=(shank), 7=no fault, 99=other)

8.2. Appendix 2 - Data Dictionary for High-level Dataset

Extracted from Tennis-Abstract Men's Report → Career Summary page [8]:

Appendix 2a - Service Game High-level Feature Set:

Service Game	Feature Definition
Matches	Matches logged by the Match Charting Project
Unret%	Percent of serves that were unreturned
<=3 W%	Percent of service points won on either the serve or second shot
RiP W%	Percent of points won when return was put in play
SvImpact	Serve Impact: Advanced stat estimating how many service points were won due to the serve
1st: Unret%	Percent of first serves that were unreturned
1st: <=3 W%	Percent of first serve points won on either the serve or second shot
1st: RiP W%	Percentage of first serve points won when return was put in play
1st: SvImpact	Serve Impact: Advanced stat estimating how many first serve points were won due to the serve
D Wide%	Percent of deuce-court first serves that were hit wide
A Wide%	Percent of ad-court first serves that were hit wide
BP Wide%	Percent of (ad-court) break point first serves that were hit wide
2nd: Unret%	Percent of second serves that were unreturned
2nd: <=3 W%	Percent of second serve points won on either the serve or second shot
2nd: RiP W%	Percentage of second serve points won when return was put in play
1st: D Wide%	Percent of deuce-court second serves that were hit wide
1st: A Wide%	Percent of ad-court second serves that were hit wide
1st: BP Wide%	Percent of (ad-court) break point second serves that were hit wide
2ndAgg	2nd Serve Aggression Score (0 is average, higher = more aggressive)

Appendix 2b - Return Game High-level Feature Set:

Return Game	Return Definition
Matches	Matches logged by the Match Charting Project
RiP%	Percent of returns put in play
RiP W%	Percent of points won when return was put in play
RetWnr%	Return winners (and induced forced errors) as a percentage of return points
Wnr FH%	Forehand winners (and induced forced errors) as a percentage of all return winners
RDI	Return Depth Index (higher = deeper)
Slice%	Slice/chip returns as a percentage of all in-play returns
1st: RiP%	Percent of first-serve returns put in play
1st: RiP W%	Percent of first-serve points won when return was put in play
1st: RetWnr%	Return winners (and induced forced errors) as a percentage of first-serve return points
1st: RDI	Return Depth Index (higher = deeper) on first serves
1st: Slice%	Slice/chip returns as a percentage of all in-play first-serve returns
2nd: RiP%	Percent of second-serve returns put in play
2nd: RiP W%	Percent of second-serve points won when return was put in play
2nd: RetWnr%	Return winners (and induced forced errors) as a percentage of second-serve return points
2nd: RDI	Return Depth Index (higher = deeper) on second serves
2nd: Slice%	Slice/chip returns as a percentage of all in-play second-serve returns

Appendix 2c - Rally Game High-level Feature Set:

Rally Game	Rally Definition
Matches	Matches logged by the Match Charting Project
RallyLen	Average rally length
RLen-Serve	Average rally length on serve points
RLen-Return	Average rally length on return points
1-3 W%	Win percentage on points between 1 and 3 shots
4-6 W%	Win percentage on points between 4 and 6 shots
7-9 W%	Win percentage on points between 7 and 9 shots
10+W%	Win percentage on points of at least 10 shots
FH/GS	Forehands per groundstroke
BH Slice%	Slices per backhand groundstroke
FHP/Match	Forehand Potency per match (higher is better)
FHP/100	Forehand Potency per 100 forehands
BHP/Match	Backhand Potency per match (higher is better)
BHP/100	Backhand Potency per 100 backhands

8.3. Appendix 3 - Low-level Feature Set Option for Clustering & Evaluation

These features resemble the ones in the features from the original PCSP functions.

Appendix 3a - Evaluation Feature Set for Service Game (36 total features):

	feature	percentage
0	1st serve-deuce court-serve to body-fault	0.028941
1	1st serve-deuce court-serve to body-no outcome	0.077920
2	1st serve-deuce court-serve to body-service wi...	0.006855
3	1st serve-deuce court-serve to wide-fault	0.149017
4	1st serve-deuce court-serve to wide-no outcome	0.244477
...	...	...
31	2nd serve-ad court-serve to wide-no outcome	0.473369
32	2nd serve-ad court-serve to wide-service winne...	0.001430
33	2nd serve-ad court-server to T-fault	0.025572
34	2nd serve-ad court-server to T-no outcome	0.154602
35	2nd serve-ad court-server to T-service winner ...	0.005051

Appendix 3b - Sample Evaluation Feature Set for Return Game (72 total features):

	feature	percentage
0	backhand-middle court-ad court-forced error	0.034057
1	backhand-middle court-ad court-no outcome	0.179579
2	backhand-middle court-ad court-unforced error	0.006807
3	backhand-middle court-ad court-winner	0.005520
4	backhand-middle court-deuce court-forced error	0.030261
...	...	...
67	forehand-deuce court-deuce court-winner	0.015918
68	forehand-deuce court-middle court-forced error	0.072939
69	forehand-deuce court-middle court-no outcome	0.460369
70	forehand-deuce court-middle court-unforced error	0.012417
71	forehand-deuce court-middle court-winner	0.001297

Appendix 3c - Sample Evaluation Feature Set for Rally Game (24 total features):

	feature	percentage
0	middle court-BH_Error	0.043231
1	middle court-BH_cross_court	0.120947
2	middle court-BH_down_mid	0.108360
3	middle court-BH_inside_out	0.063293
4	middle court-FH_Error	0.079516
...	...	...
19	ad court-BH_down_mid	0.187522
20	ad court-FH_Error	0.031965
21	ad court-FH_down_mid	0.030494
22	ad court-FH_inside_in	0.054781
23	ad court-FH_inside_out	0.103043

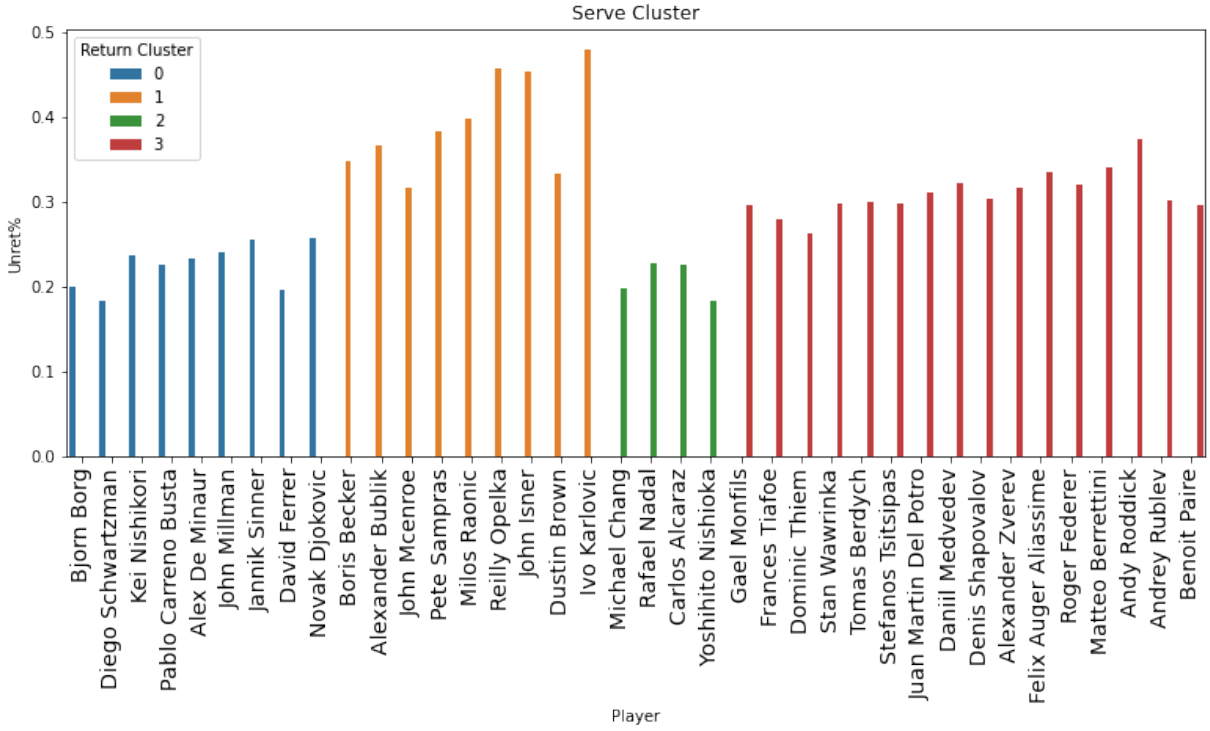
Appendix 3d - DataFrame with every Player's Average Values for Rally Feature Set:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Guillermo Coria	0.0338	0.3055	0.0064	0.0048	0.0048	0.0820	0.0016	0.0097	0.0579	0.4968	0.0064	0.0009	0.0192	0.0769	0.0110	0.0055	0.0412	0.2390	0.0027	0.0082	0.0495	0.5357	0.0110	0.0013
Marton Fucsovics	0.0445	0.3037	0.0105	0.0026	0.0314	0.0471	0.0026	0.0097	0.0602	0.4921	0.0052	0.0009	0.0630	0.1693	0.0094	0.0079	0.0394	0.0827	0.0103	0.0039	0.1063	0.5197	0.0079	0.0013
Sergi Bruguera	0.0341	0.3765	0.0119	0.0136	0.0239	0.0647	0.0068	0.0034	0.0494	0.4003	0.0153	0.0009	0.0648	0.1852	0.0123	0.0062	0.0154	0.1852	0.0185	0.0031	0.0679	0.4383	0.0031	0.0013
Andy Murray	0.0474	0.3428	0.0136	0.0140	0.0148	0.0651	0.0041	0.0048	0.0462	0.4361	0.0102	0.0010	0.0345	0.1766	0.0047	0.0045	0.0371	0.1506	0.0081	0.0115	0.0648	0.5015	0.0055	0.0006
Bjorn Borg	0.0557	0.2191	0.0099	0.0309	0.0507	0.0668	0.0037	0.0124	0.0606	0.4827	0.0062	0.0012	0.0440	0.0924	0.0117	0.0191	0.0396	0.1598	0.0015	0.0176	0.0528	0.5543	0.0073	0.0013
Nikolay Davydenko	0.0374	0.2326	0.0034	0.0034	0.0407	0.1205	0.0119	0.0136	0.0560	0.4737	0.0068	0.0009	0.0571	0.1051	0.0060	0.0030	0.0480	0.1922	0.0180	0.0090	0.0871	0.4655	0.0090	0.0013
Alex Corretja	0.0298	0.3830	0.0069	0.0023	0.0206	0.0573	0.0161	0.0046	0.0459	0.4289	0.0046	0.0009	0.0601	0.2275	0.0043	0.0043	0.0343	0.1116	0.0043	0.0086	0.0644	0.4678	0.0086	0.0043
Marco Cecchinato	0.0658	0.3243	0.0113	0.0076	0.0454	0.1066	0.0045	0.0045	0.0317	0.4014	0.0023	0.0023	0.0679	0.2038	0.0075	0.0038	0.0415	0.1849	0.0075	0.0038	0.0604	0.4113	0.0075	0.0013
Gael Monfils	0.0370	0.2819	0.0087	0.0047	0.0236	0.0740	0.0055	0.0047	0.0606	0.4866	0.0126	0.0009	0.0325	0.1174	0.0062	0.0012	0.0387	0.2297	0.0175	0.0137	0.0574	0.4719	0.0125	0.0012
Daniil Medvedev	0.0437	0.2245	0.0062	0.0042	0.0353	0.0977	0.0104	0.0166	0.0561	0.4990	0.0062	0.0009	0.0384	0.1458	0.0094	0.0026	0.0307	0.1381	0.0051	0.0026	0.0742	0.5575	0.0051	0.0013
Damir Dzumhur	0.0289	0.2459	0.0036	0.0018	0.0380	0.0850	0.0018	0.0108	0.0542	0.5208	0.0090	0.0009	0.0548	0.1952	0.0071	0.0024	0.0357	0.1333	0.0103	0.0149	0.0500	0.5167	0.0048	0.0013
Gustavo Kuerten	0.0558	0.3001	0.0080	0.0106	0.0292	0.0704	0.0120	0.0159	0.0412	0.4515	0.0040	0.0013	0.0552	0.1444	0.0106	0.0170	0.0594	0.2059	0.0085	0.0149	0.0594	0.4225	0.0021	0.0013
Richard Gasquet	0.0398	0.3769	0.0122	0.0023	0.0191	0.0520	0.0038	0.0031	0.0466	0.4327	0.0115	0.0009	0.0203	0.1311	0.0032	0.0011	0.0618	0.1652	0.0085	0.0213	0.0746	0.5075	0.0053	0.0013
Dusan Lajovic	0.0541	0.2770	0.0068	0.0068	0.0439	0.0743	0.0072	0.0097	0.0439	0.4764	0.0169	0.0009	0.0301	0.0836	0.0067	0.0104	0.0803	0.2508	0.0067	0.0100	0.0903	0.4381	0.0033	0.0013
David Ferrer	0.0447	0.3179	0.0075	0.0075	0.0186	0.0800	0.0023	0.0035	0.0580	0.4507	0.0093	0.0009	0.0515	0.1694	0.0090	0.0090	0.0336	0.1261	0.0045	0.0097	0.0851	0.4955	0.0060	0.0007
Tommy Robredo	0.0479	0.2217	0.0050	0.0025	0.0277	0.0831	0.0050	0.0050	0.0504	0.5416	0.0101	0.0009	0.0279	0.2043	0.0062	0.0124	0.0217	0.2043	0.0093	0.0217	0.0588	0.4211	0.0124	0.0013

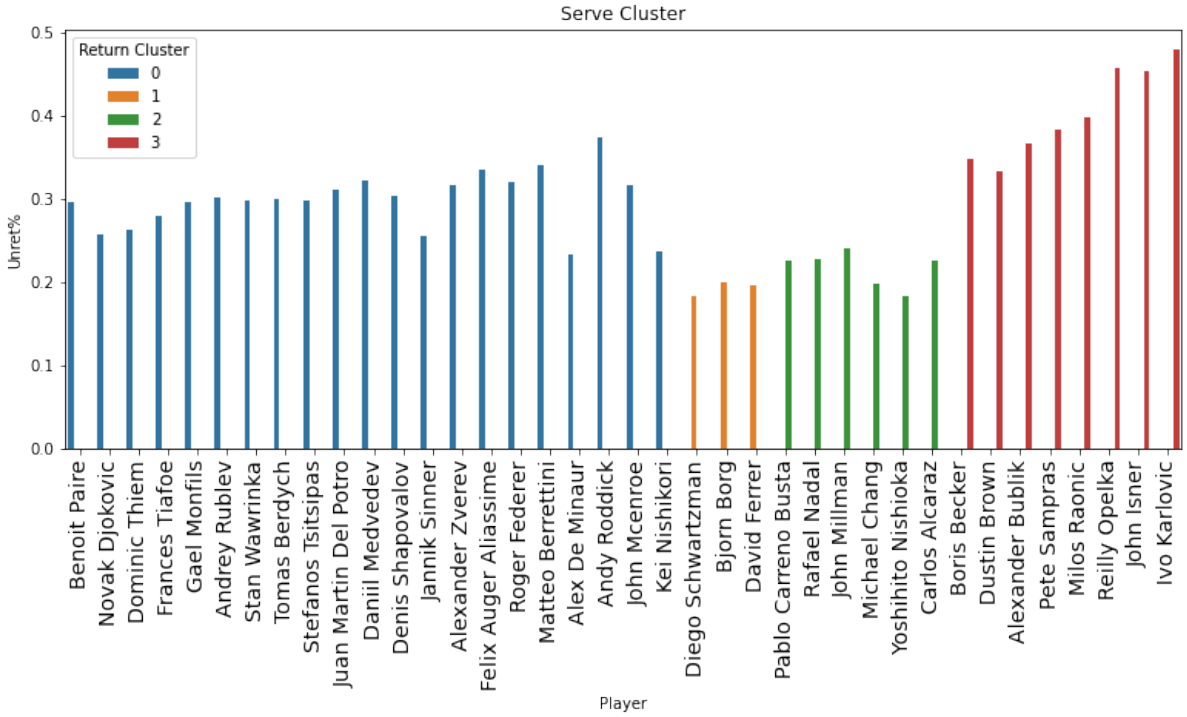
8.4. Appendix 4 - Cluster Group Results

Results were filtered down to 38 players for visualization and inference purposes.

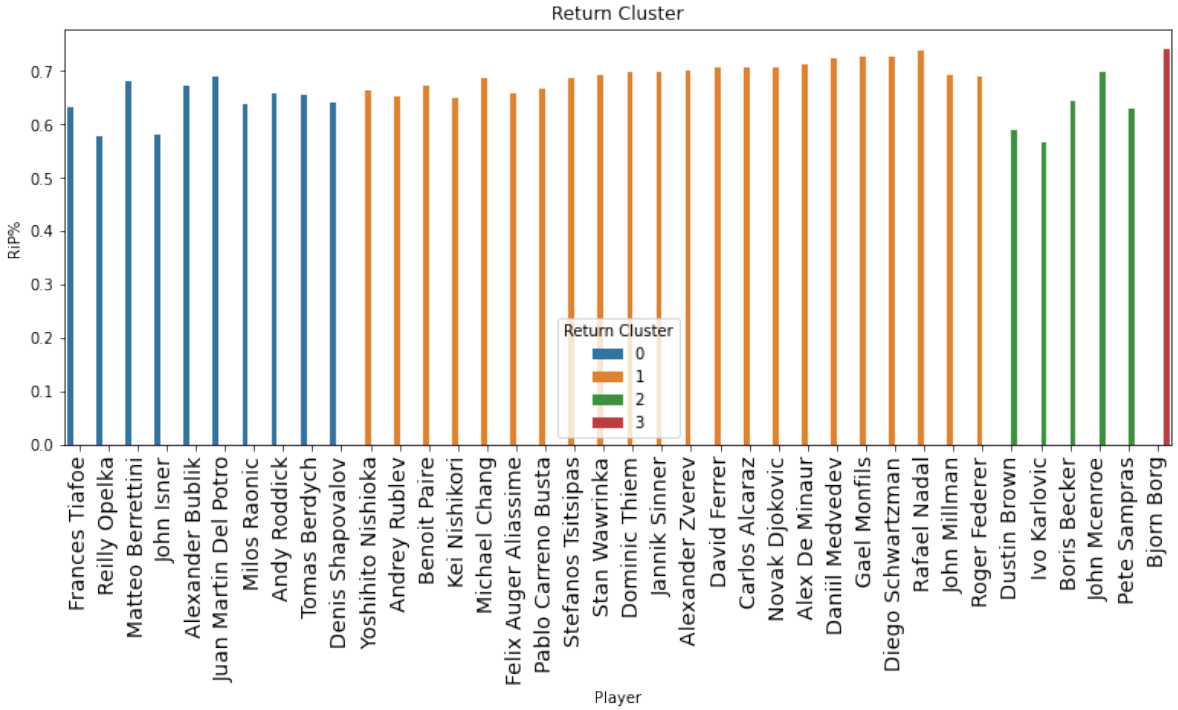
Appendix 4a - Service Cluster Results from K-Means:



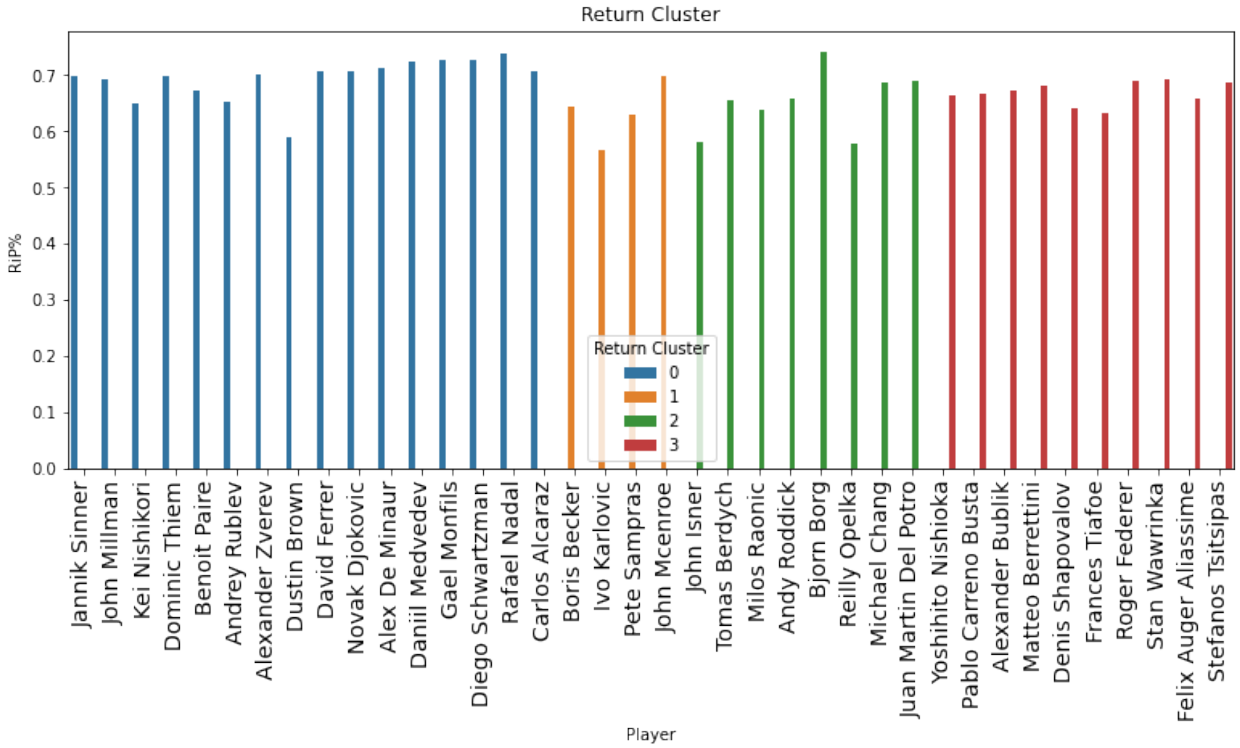
Appendix 4b - Service Cluster Results from Agglomerative:



Appendix 4c - Return Cluster Results from K-Means:

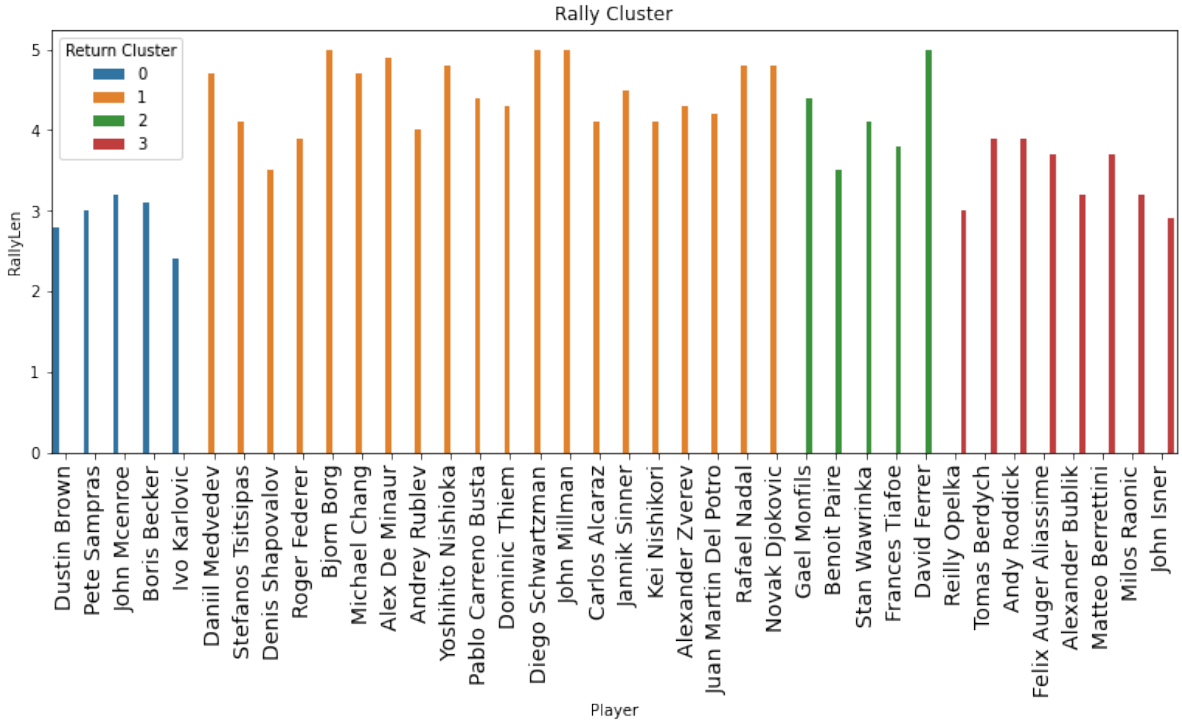


Appendix 4d - Return Cluster Results from Agglomerative:

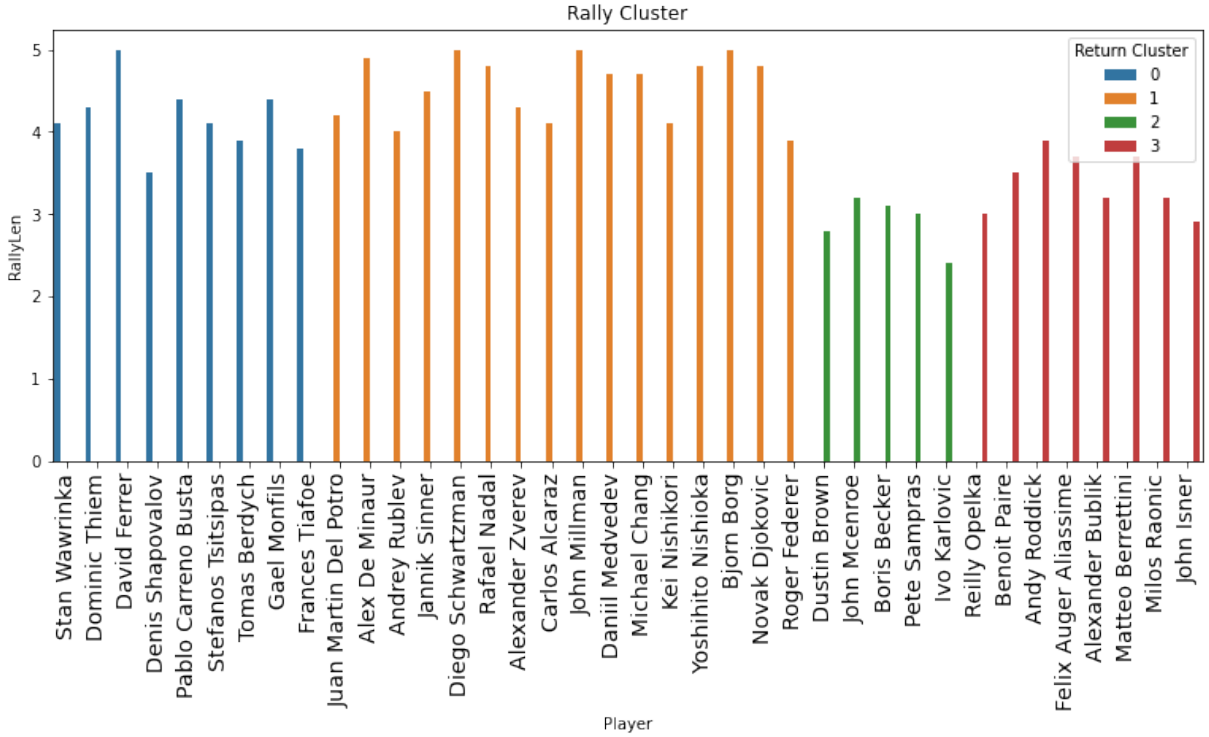


	Cluster	Size of Cluster	Intra-cluster Distance Avg	Inter-cluster Distance Avg	% Difference (Inter - Intra)
0	0.0	31.00	71.841930	75.023987	4.429248
1	1.0	56.00	55.773840	92.860921	66.495477
2	2.0	11.00	106.838081	104.796103	-1.911282
3	3.0	13.00	63.282018	81.353779	28.557499
Overall Average	1.5	27.75	74.433967	88.508698	24.392735

Appendix 4e - Rally Cluster Results from Agglomerative:



Appendix 4f - Rally Cluster Results from K-Means:

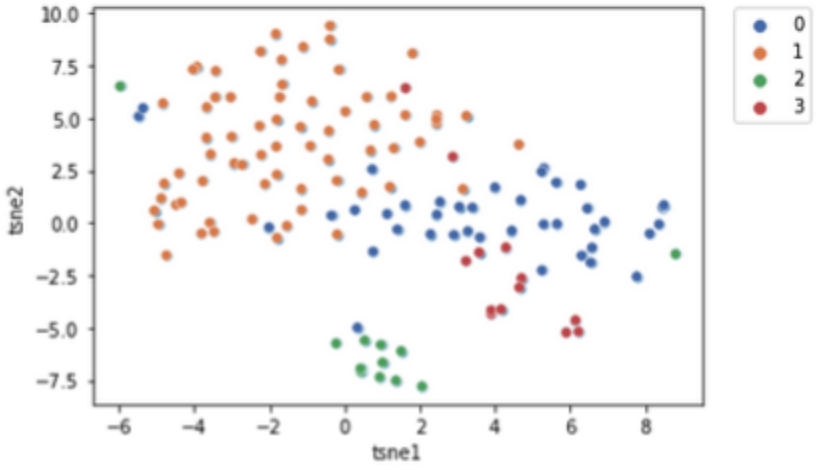


	Cluster	Size of Cluster	Intra-cluster Distance Avg	Inter-cluster Distance Avg	% Difference (Inter - Intra)
0	0.0	9.00	21.736816	37.396770	72.043455
1	1.0	52.00	22.335334	26.650491	19.319865
2	2.0	35.00	24.171774	24.286985	0.476636
3	3.0	27.00	19.334891	28.385143	46.807873
Overall Average	1.5	30.75	21.894704	29.179847	34.661957

8.5. Appendix 5 - Evaluation #1 and #2 Results for Return and Rally Games:

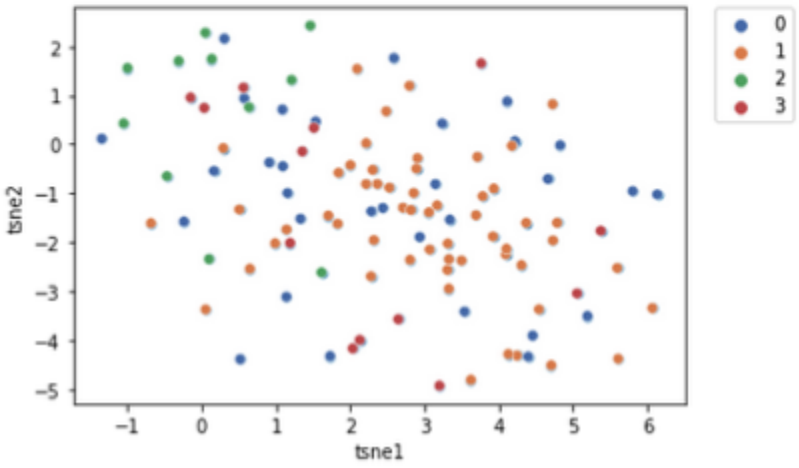
Appendix 5a - Evaluation #1 Results for Return Game using K-Means:

```
[1 3 1 1 3 1 3 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1
 2 1 1 1 0 1 3 3 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0 0 0 1 0 1 3 0 1 0 1 1 1 0 1
 1 1 0 2 1 0 3 3 2 1 3 0 1 0 0 0 0 0 0 3 1 1 1 1 1 1 0 0 0 2 0 0 0 0 0 2 1
 1 0 1 2 0 2 3 0 0 0 0 1 0 0 0 2 2 0 0 2 2 0]
1    68
0    40
3    13
2    12
Name: Return Cluster, dtype: int64
Cluster Results:
---Evaluating Cluster 0---
Size of Cluster: 40
Intra-cluster Distance Avg: 8.01
Inter-cluster Distance Avg: 24.30
% Difference (Inter - Intra) for Cluster 0: 203.35%
-------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 68
Intra-cluster Distance Avg: 7.25
Inter-cluster Distance Avg: 31.53
% Difference (Inter - Intra) for Cluster 1: 334.98%
-------------------------------------------------
---Evaluating Cluster 2---
Size of Cluster: 12
Intra-cluster Distance Avg: 17.76
Inter-cluster Distance Avg: 62.13
% Difference (Inter - Intra) for Cluster 2: 249.81%
-------------------------------------------------
---Evaluating Cluster 3---
Size of Cluster: 13
Intra-cluster Distance Avg: 8.60
Inter-cluster Distance Avg: 28.41
% Difference (Inter - Intra) for Cluster 3: 230.14%
-------------------------------------------------
Overall Average of % Difference (Inter-Intra): 254.57%
<AxesSubplot:xlabel='tsne1', ylabel='tsne2'>
```



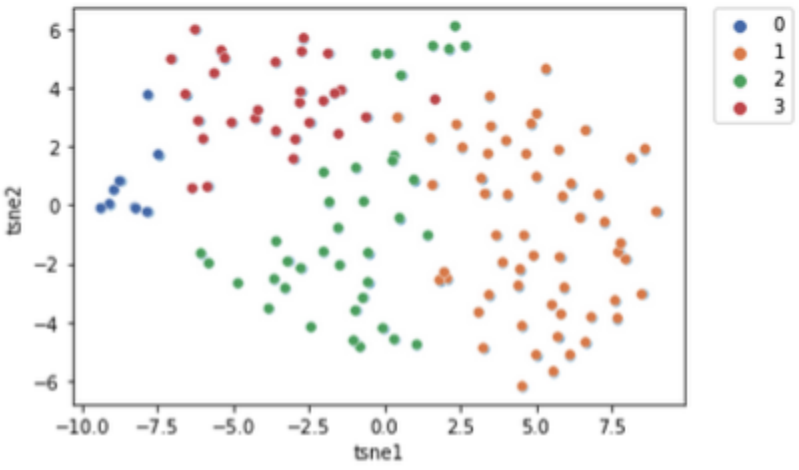
Appendix 5b - Evaluation #2 Results for Return Game using K-Means:

```
[1 3 1 1 3 1 3 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1
 2 1 1 1 0 1 3 3 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0 0 0 1 0 1 3 0 1 0 1 1 1 0 1
 1 1 0 2 1 0 3 3 2 1 3 0 1 0 0 0 0 0 0 3 1 1 1 1 1 1 0 0 0 2 0 0 0 0 0 2 1
 1 0 1 2 0 2 3 0 0 0 0 1 0 0 0 2 2 0 0 2 2 0]
1    68
0    40
3    13
2    12
Name: Return Cluster, dtype: int64
Cluster Results:
---Evaluating Cluster 0---
Size of Cluster: 31
Intra-cluster Distance Avg: 71.84
Inter-cluster Distance Avg: 75.02
% Difference (Inter - Intra) for Cluster 0: 4.43%
-------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 56
Intra-cluster Distance Avg: 55.77
Inter-cluster Distance Avg: 92.86
% Difference (Inter - Intra) for Cluster 1: 66.50%
-------------------------------------------------
---Evaluating Cluster 2---
Size of Cluster: 11
Intra-cluster Distance Avg: 106.84
Inter-cluster Distance Avg: 104.80
% Difference (Inter - Intra) for Cluster 2: -1.91%
-------------------------------------------------
---Evaluating Cluster 3---
Size of Cluster: 13
Intra-cluster Distance Avg: 63.28
Inter-cluster Distance Avg: 81.35
% Difference (Inter - Intra) for Cluster 3: 28.56%
-------------------------------------------------
Overall Average of % Difference (Inter-Intra): 24.39%
<AxesSubplot:xlabel='tsne1', ylabel='tsne2'>
```



Appendix 5c - Evaluation #1 Results for Rally Game using Agglomerative:

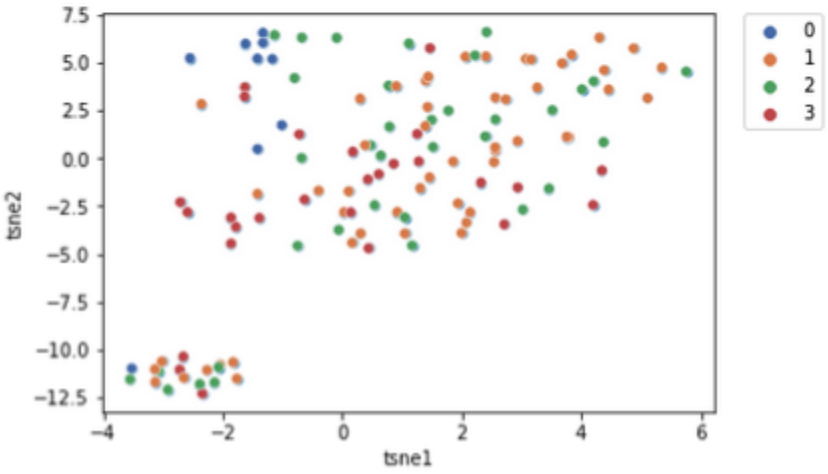
```
[0 1 1 1 1 1 1 1 1 1 1 1 2 1 1 0 0 1 0 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 0
 2 2 3 1 1 2 1 1 2 1 1 2 1 1 0 1 2 1 1 1 3 2 2 1 2 3 1 1 2 2 1 0 1 2 2 2 1
 2 2 2 2 1 2 1 1 2 1 3 1 2 1 2 3 3 2 3 3 3 1 2 3 2 0 3 2 3 1 2 1 1 2 2 3 3
 2 3 2 2 3 3 2 3 3 2 3 3 3 3 2 3 3 3 3 3 3 0]
1    57
2    38
3    29
0     9
Name: Return Cluster, dtype: int64
Cluster Results:
---Evaluating Cluster 0---
Size of Cluster: 9
Intra-cluster Distance Avg: 14.48
Inter-cluster Distance Avg: 34.28
% Difference (Inter - Intra) for Cluster 0: 136.79%
-------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 57
Intra-cluster Distance Avg: 7.75
Inter-cluster Distance Avg: 23.40
% Difference (Inter - Intra) for Cluster 1: 201.86%
-------------------------------------------------
---Evaluating Cluster 2---
Size of Cluster: 38
Intra-cluster Distance Avg: 6.89
Inter-cluster Distance Avg: 17.39
% Difference (Inter - Intra) for Cluster 2: 152.44%
-------------------------------------------------
---Evaluating Cluster 3---
Size of Cluster: 29
Intra-cluster Distance Avg: 7.99
Inter-cluster Distance Avg: 24.11
% Difference (Inter - Intra) for Cluster 3: 201.63%
-------------------------------------------------
Overall Average of % Difference (Inter-Intra): 173.18%
<AxesSubplot:xlabel='tsne1', ylabel='tsne2'>
```



Appendix 5d - Evaluation #2 Results for Rally Game using Agglomerative:

```
[0 1 1 1 1 1 1 1 1 1 1 2 1 1 0 0 1 0 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 0
 2 2 3 1 1 2 1 1 2 1 1 2 1 1 0 1 2 1 1 1 3 2 2 1 2 3 1 1 2 2 1 0 1 2 2 2 1
 2 2 2 2 1 2 1 1 2 1 3 1 2 1 2 3 3 2 3 3 3 1 2 3 2 0 3 2 3 1 2 1 1 2 2 3 3
 2 3 2 2 3 3 2 3 3 2 3 3 3 3 2 3 3 3 3 3 3 0]
1    57
2    38
3    29
0     9
Name: Return Cluster, dtype: int64
Cluster Results:
---Evaluating Cluster 0---
Size of Cluster: 9
Intra-cluster Distance Avg: 21.74
Inter-cluster Distance Avg: 37.40
% Difference (Inter - Intra) for Cluster 0: 72.04%
-----------------------------------------------------
---Evaluating Cluster 1---
Size of Cluster: 52
Intra-cluster Distance Avg: 22.34
Inter-cluster Distance Avg: 26.65
% Difference (Inter - Intra) for Cluster 1: 19.32%
-----------------------------------------------------
---Evaluating Cluster 2---
Size of Cluster: 35
Intra-cluster Distance Avg: 24.17
Inter-cluster Distance Avg: 24.29
% Difference (Inter - Intra) for Cluster 2: 0.48%
-----------------------------------------------------
---Evaluating Cluster 3---
Size of Cluster: 27
Intra-cluster Distance Avg: 19.33
Inter-cluster Distance Avg: 28.39
% Difference (Inter - Intra) for Cluster 3: 46.81%
-----------------------------------------------------
Overall Average of % Difference (Inter-Intra): 34.66%
```

```
<AxesSubplot:xlabel='tsne1', ylabel='tsne2'>
```



9. References

[1] "Summary: Moneyball: The Art of Winning an Unfair Game by Michael Lewis – WordsRated." https://wordsrated.com/summary-moneyball-the-art-of-winning-an-unfair-game-by-michael-lewis/ (accessed Nov. 15, 2022).

[2] C. Andrews, "Germany are turning to technology to get even better at penalties," *The Set Pieces*, Jun. 16, 2016. https://thesetpieces.com/latest-posts/germany-turning-technology-get-even-better-penalties/ (accessed Nov. 15, 2022).

[3] "Tennis Popularity Statistics 2021," *Chase Your Sport*. https://www.chaseyoursport.com/Tennis/Tennis-Popularity-Statistics-2021/3472 (accessed Nov. 22, 2022).

[4] A. Blicher, "The history of analytics and statistics in tennis," *The Sports Scientist*, May 18, 2020. https://medium.com/the-sports-scientist/the-history-of-analytics-and-statistics-in-tennis-e19aa206fdf0 (accessed Nov. 15, 2022).

[5] J. Marfatia, "Wimbledon Predictions using Neural Network.," *Medium*, Jun. 13, 2021. https://towardsdatascience.com/predicting-wimbledon-matches-using-neural-network-e2ee4d3dead2 (accessed Nov. 14, 2022).

[6] S. Carver, "Markov Chain Models in Sports," *Medium*, Jul. 04, 2019. https://towardsdatascience.com/markov-chain-models-in-sports-7cb907a6c52f (accessed Nov. 22, 2022).

[7] "Hawk-Eye," *Wikipedia*. Nov. 10, 2022. Accessed: Nov. 18, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Hawk-Eye&oldid=1121020192

[8] "Tennis Abstract: ATP Match Results, Splits, and Analysis." http://www.tennisabstract.com/cgi-bin/leaders.cgi?f=F0s00o1 (accessed Nov. 14, 2022).

[9] "Communicating sequential processes," *Wikipedia*. Nov. 16, 2022. Accessed: Nov. 22, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Communicating_sequential_processes&oldid=1122259130

[10] "3.3 Probability CSP (PCSP) Module." https://www.comp.nus.edu.sg/~pat/OnlineHelp/scr/3.3%20PCSP%20Module/3.3%20Probability%20CSP%20Module.htm (accessed Nov. 22, 2022).

[11] J. S. Dong, L. Shi, L. V. N. Chuong, K. Jiang, and J. Sun, "Sports Strategy Analytics Using Probabilistic Reasoning," in *2015 20th International Conference on Engineering of Complex Computer Systems (ICECCS)*, Dec. 2015, pp. 182–185. doi: 10.1109/ICECCS.2015.28.

[12] "What is Clustering? | Machine Learning," *Google Developers*. https://developers.google.com/machine-learning/clustering/overview (accessed Nov. 17, 2022).

[13] E. Ecosystem (LEDU), "Understanding K-means Clustering in Machine Learning," *Medium*, Sep. 12, 2018. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1 (accessed Nov. 17, 2022).

[14] "Agglomerative Hierarchical Clustering," *Datanovia*. https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/ (accessed Nov. 17, 2022).

[15] K. S. do Prado, "How DBSCAN works and why should we use it?," *Medium*, Jun. 03, 2019. https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80 (accessed Nov. 17, 2022).

[16] "ML | Intercluster and Intracluster Distance," *GeeksforGeeks*, May 10, 2019. https://www.geeksforgeeks.org/ml-intercluster-and-intracluster-distance/ (accessed Nov. 22,

2022).
[17] A. C, “NUS MComp Capstone Project 2022.” Nov. 22, 2022. Accessed: Nov. 23, 2022. [Online]. Available: https://github.com/aar809/sports_analytics
[18] HARVARDSPORTS, “Sorting Strokes: Classifying Tennis Players Based on Style | The Harvard Sports Analysis Collective,” Jul. 28, 2021. https://harvardsportsanalysis.org/2021/07/sorting-strokes-classifying-tennis-players-based-on-stats-and-style/ (accessed Nov. 18, 2022).
[19] “Elo rating system,” *Wikipedia*. Nov. 16, 2022. Accessed: Nov. 18, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Elo_rating_system&oldid=1122138971
[20] J. Sackmann, “Doubles.” Nov. 10, 2022. Accessed: Nov. 14, 2022. [Online]. Available: https://github.com/JeffSackmann/tennis_atp