Classification of Chess Games and Players by Styles Using Game Data

M. G. P. B. Jayasekara

2018

Classification of Chess Games and Players by Styles Using Game Data

A dissertation submitted for the Degree of Master of Computer Science

M. G. P. B. Jayasekara
University of Colombo School of Computing
2018

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: M. G. P. B. Jayasekara

Registration Number: 2014/MCS/033

Index Number: 14440336

_____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr. M. G. P. B. Jayasekara

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. D. A. S. Atukorale

_____

Signature: Date:

Abstract

In a strategic game like chess, players have perfect information and there is no involvement of chances or physical skills. Even though the game is played on a board, the entire game runs on the two players' minds. Players have to think smart and make plans to win the game. But, the way two players think about the same situation could be completely different. Therefore their approach to playing could be different from each other. Due to this very reason, there are recognized well-known playing styles which are bound to different persons.

Current well-known styles such as Aggressive, Positional, Solid, Defensive etc. are introduced by chess experts with the help of their experience and expertise. However, these are not of an outcome of any scientific research. Therefore, in this research, I'm exploring if these well-known styles can be scientifically and logically separable and if those can be predicted looking at game data.

Since the raw data (i.e. moves) of a chess game does not provide any direct information about the game's nature, but only contains notations of a sequence of moves, to extract features from a game, those notations should be understood and certain preprocessing steps are required. Chess game engines can be used for this. Raw game data can be fed to chess game engines and information regarding states of the game can be obtained at each move. Such data can be collected for all or selected steps of a game and then that information can be used to construct features of the game. This process was followed in this project and a set of features were extracted from thousands of games.

Those extracted data were then clustered and identified the natural clusters in those. In this research, I prove that some of those well known styles are separable from others and some are overlapping. I also prove that prediction models can be created to predict such styles looking at the game data.

Acknowledgements

First and foremost, I take the immense pleasure in thanking my supervisor, Dr. D.A.S. Atukorale, The Deputy Director, University of Colombo School of Computing, and Dr. Srinath Perera, Vice President, WSO2, for the tremendous support throughout my research project with his knowledge, motivation, and patience.

I would also like to thank the chess experts who were involved in this research project in numerous ways. Without their passionate participation and input, the this research project could not have been successfully conducted.

Then, I wish to extend my gratitude to Prof K. P. Hewagamage, The Director, University of Colombo School of Computing, for giving me the opportunity to carry on with my research work.

Moreover, I wish to extend my deep sense of gratitude to the Non-Academic staff members of Department of Computer Science, Faculty of Science, University of Colombo School of Computing,

Finally, I must express my very profound gratitude to my parents and to my wife for providing me with unfailing support and continuous encouragement throughout the years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Content

Chapter 1: Introduction **1**

1.1 The Importance of identifying styles 1

1.2 The Problem and Motivation 3

1.3 Objectives and Scope 4

1.4 Deliverables 4

1.5 Organization of the Thesis 4

Chapter 2: Background and Literature Review **5**

2.1 A multidimensional approach to positional chess 5

2.2 Popularity Distribution of Chess Openings 6

2.3 Cheating detection in chess 7

2.4 Move similarity analysis in chess programs 7

Chapter 3: Methodology and Solution **9**

3.1 Raw Data Analysis 9

3.1.1 Standard Algebraic notation (SAN) 9

Square Naming Notation 9

Piece Naming Notation 10

Move Naming Notation 10

3.1.2 Portable Game Notation (PGN) 12

3.2 Dataset 13

3.3 Feature Engineering 13

3.3.1 Identifying Features 13

Straightforward Features 16

Complex Features 17

3.3.2 Feature Extraction 17

3.3.3 Manual Categorization 18

3.3.4 Data Preprocessing 20

3.4 Analysis 20
3.4.1 Initial Clustering 20
3.4.2 New Dataset 23
3.4.3 K-Means Clustering 23
3.4.4 Advanced clustering 24
3.4.5 Visualization 24
3.4.6 Principal Component Analysis (PCA) 25
3.4.7 Gaussian Mixture Models (GMM) with Expectation Maximization (EM) 26
3.4.8 Self Organizing Maps (SOM) 29
3.4.9 Cluster Evaluation 30
3.4.10 Classification 45
Chapter 4: Evaluation and Results **47**
4.1 Evaluation of Clustering Output 47
4.2 Evaluation of Classification Model 47
4.2.1 Using games of players who're well-known for certain styles 48
4.2.2 Using opinions of chess experts 49
Chapter 5: Conclusion **50**
Chapter 6: Future Work **52**
References **54**
Table of Content **57**

List of Figures

Figure 3.1: Square Naming in Chess Board 09

Figure 3.2: Nf3 10

Figure 3.3: Ncd3 or Ned3 11

Figure 3.4: Nxd4 11

Figure 3.5: Interactive chess analysis forms 18

Figure 3.6: Games categorized by chess experts 19

Figure 3.7: Online form for Player-to-Style mapping 22

Figure 3.8: Player-to-Style mapping results 22

Figure 3.9: Features of new dataset 23

Figure 3.10: Visualized dataset (dimension reduced) 25

Figure 3.11: K-Means applied on dimension reduced data 25

Figure 3.12: Confusion matrix of K-Means applied on dimension reduced data 26

Figure 3.13: GMM applied on dimension reduced data 28

Figure 3.15: Silhouette Analysis for 2 clusters 36

Figure 3.16: Silhouette Analysis for 3 clusters 36

Figure 3.17: Silhouette Analysis for 4 clusters 37

Figure 3.18: Silhouette Analysis for 5 clusters 37

Figure 3.19: Silhouette Analysis for 6 clusters 38

Figure 3.20: Calinski Harabaz Analysis for 2 clusters 39

Figure 3.21: Calinski Harabaz Analysis for 3 clusters 39

Figure 3.22: Calinski Harabaz Analysis for 4 clusters 40

Figure 3.23: Calinski Harabaz Analysis for 5 clusters 40

Figure 3.24: Calinski Harabaz Analysis for 6 clusters 41

Figure 3.25: Calinski Harabaz Analysis for 7 clusters 41

List of Tables

Table 3.1: Notations of chess pieces 10
Table 3.2: K-Means clusters with different numbers of clusters 21
Table 3.3: K-Means clustering results 24
Table 3.4: Confusion matrices of GMM applied on dimension reduced data 28
Table 3.5: Internal cluster validity indices of GMM 31
Table 3.6: External cluster validity indices of GMM 34
Table 3.7: Silhouette Coefficients of different cluster counts with GMM 35
Table 3.8: Calinski Harabaz Index of different cluster counts with GMM 38
Table 3.9: Confusion matrices of GMM applied for 4 clusters 42
Table 3.10: Internal cluster validity indices of GMM for 4 clusters 43
Table 3.11: External cluster validity indices of GMM for 4 clusters 43
Table 3.12: Confusion matrices of GMM with Tied covariance applied for 4 clusters 44

List of Abbreviations

Abbreviation	***Explanation***
ARI	Adjusted Rand Index
EM	Expectation Maximization
GMM	Gaussian Mixture Models
PCA	Principal Component Analysis
PGN	Portable Game Notation
RI	Rand Index
SAN	Standard Algebraic Notation
SOM	Self Organizing Maps
UCI	Universal Chess Interface

Chapter 1: Introduction

Chess is a two-player strategy board game played on a chessboard, which is a checkered game-board with 64 squares arranged in an 8×8 grid. In the game of chess, players move pieces on the board as per strategic plans in order to win against the opponent. A game of chess consists of a sequence of such chess piece moving steps by each player where they respond to each other's moves.

When it comes to responses, different players respond in different ways. There can be many reasons for this difference. For example, it could be due to different personalities of the players, or due to different levels of thinking capabilities of players etc.

People, who are experienced in chess, have identified certain patterns in chess players' ways of responding. Sets of such patterns are called 'styles'. Some such known styles are aggressive, positional, intuitive, creative etc. A few important facts of these styles are, a player could have more than one style, and certain games of a particular player could be played in a style which was not known for that player before, etc.

1.1 The Importance of identifying styles

In strategic games like chess, players have perfect information and there is no involvement of chances or physical skills. Even though the game is played on a board, the entire game runs on the two players' minds. The board is just a tool to represent the current state of the game. Players have to think smart and make plans to win the game. But the way two players think about the same situation could be completely different. Therefore their approach of playing could be different from each other. Naturally, some approaches are better than others. For example, the approach of Kasparov, who is a chess Grandmaster (GM), is obviously quite a bit better than my mental approach to chess.

Just like in other games, in chess, players try to develop their skills and become better in the game. For that, they usually learn new things like openings and endgame puzzles to refine winning approaches and to develop their thinking patterns in the game. One of the most beautiful things about humans is that most of the things we learn and do regularly go to our subconscious and then those become habits or becomes natural within us. After some time, it lets us do those things without any explicit thoughts or effort. This is valid for games such as

chess as well. So when we're refining our approaches, we have to refine those which are gone to our subconscious as well.

The great thing about the subconscious is it really works[1]. If you teach your subconscious to look at some chess position in a particular way, it really starts to work that way. It works not only in the domain of chess, but in your daily life as well. That's the beauty of subconscious because it does not store that information in your subconscious with a label like "this is for chess only". Instead, it can reuse the same information in other cases as well[1]. The importance of this in our case is that the other way of this is also true. That means, what you learn in your daily life and what you put into your subconscious through your daily work can be used in a way you think of a chess position as well. The way you play chess might accurately reflect the way you live your life. If there were only one way to live our life, which means if everyone lives the same life, there would be one single playing style in chess too. If that were the case, chess wouldn't be an interesting game. But fortunately, that's not the case. There are countless ways to live our life, which introduces different natures of humans, and that leads to different playing styles in chess too.

But, how is it so important that figuring out those styles? It can be explained like this. If you are a young player, you might be trying different things in the game just to understand what works best for you. That means you might be trying different playing styles. But if you can get your real life style into the game too, the way you play becomes more natural and that could be the best way you can play.

And then, if you realize your own playing style, and if you know great players of the same style, you could learn a lot of things from them and improve yourself in a natural way that best suits for yourself. For example, if you are a tactical player, an opening such as Giuoco Piano[2], which makes the game open early by letting the pieces meet quickly, would be more advantages than playing an opening such as Queen's Gambit[3], where it requires a lot of maneuvering before the game becomes open and more tactical. On the other hand, if you know the style of your opponent, it might be easy to predict them and not to let them make a move which leads to a position of their favour. Let's take the other side of the same previous example. If you know your opponent is a tactical player and if he starts with e4, you could guess that he's trying to go for Giuoco Piano as he's a tactical player, and you could go for something other than e5 to not to let them take the path for Giuoco Piano. Another example is in the case of you having to start the game and if you know your opponent is a

strong positional player, you have the chance to start the game which does not lead to a state which is advantageous to a positional player.

1.2 The Problem and Motivation

Aforementioned styles such as aggressive, positional, intuitive, creative etc. are identified and named by chess experts using their expertise and long term experience in the game. However, these opinions and categorizations are subjective. One expert might say a game is of a particular style, but someone else's opinion might be different on the same game. The main idea of this research arises from there. What if we had a proper and systematic way to figure out a style of a given game or a player? How about using a large set of game data and use machine learning and big data techniques to come up with a good classifier for chess playing styles?

The game of Chess adapted databases and some engines early, compared to other games. For example, there are websites such as http://analysis.cpuchess.com/ , https://www.chess.com/analysis-board-editor, https://en.lichess.org/analysis etc, where people can play chess online and then the website analyses the game realtime and gives feedback about its current position (eg. the winning probability of each colour at a given state etc.).

However, it still has not evolved much in the age of big data. For example, as mentioned before, right now databases/engines still give simple predictions and statistics such as win ratio in each color, openings used etc. than going to more in-depth analysis to recognize patterns that might capture the "human component" of the game. Playing style is one of those main features of the human component.

The main objective of this research is two folded. That is developing a classifier for both chess games and players using the playing styles. There's a difference between these two. If we consider a single game, the white player can have one style and the black player can have a different style for that particular game. That can be identified as the style of that game for each player. Even for a single player, style can be different from game to game. But if we take a large number of games of a single player, we might be able to identify a natural and mostly used style for that player. However, the latter is dependent on the former, because classification of games by style is required for the classification of players by style.

1.3 Objectives and Scope

The project consists of 2 main objectives.

1. Identify similarities and differences of chess games, based on the way the players play the game. Then, based on that, identify different playing styles.

    ➢ Use unsupervised/semi-supervised learning (e.g. Clustering) and try to find the relationship between different clusters and playing styles by inspecting clusters and structures.

    ➢ There are known patterns and playing styles in chess. The idea of this objective is to check the possibility of finding new styles using clustering techniques.

2. Develop a classifier that will recognize the playing style of players involved in a given game or a set of games by a single player.

    ➢ The output of this objective is the ability to classify unknown games and players into known categories.

The scope of this research project includes,

- building a tool to collect and parse game data
- analyzing data to select and extract features based on playing styles
- trying to understand natural categories based on styles
- developing a model to classify players and games by style

1.4 Deliverables

- Tools developed to collect and parse chess game data
- Model to predict the style of play of players

1.5 Organization of the Thesis

This dissertation presents a model which was developed to predict chess playing style of a player or a game. The organization of this thesis is as follows. Chapter 1 gives an introduction to the domain and problem space. Chapter 2 discusses about related researches done so far. Chapter 3 presents the research methodology, details of analysis, information about dataset used and the solution. Chapter 4 presents the evaluation methods and results. Chapter 5 and 5 discusses about conclusion and future work respectively.

Chapter 2: Background and Literature Review

Due to the complex and interesting nature of the game, a lot of researchers have carried out researches about chess in many aspects. Some of those researches, for example, building improved chess engines, cheat detection in chess games, and move-similarity analysis about chess programs etc., are directly about the game itself. And other research such as how chess helps in improving memory capabilities and cognitive abilities, how chess performance is affected by the gender and how playing chess affects personality of players etc, are not directly about the game itself, but about how it affects lives of players.

2.1 A multidimensional approach to positional chess

R.H.Atkin and I.H.Witten have analyzed chess games by considering the relation between pieces and the squares. They have come up with a mathematical representation for this relation. They say that "this relation is mathematically equivalent to a simplicial complex which, in its turn, possesses a geometrical representation in the euclidean space E^{53}."[4] It is, therefore, possible to interpret the course of a Game of chess as the expansion and contraction of two geometrical structures (one for White and the other for Black) in this multi-dimensional space[4]. They have analyzed the strength of states/positions of a chess game and come up with a relative ranking value for each position which tells how good a move was.

They had used 2 approaches to analyze games. One is an interactive analysis of live keyboard sessions with chess players, and the other one is an analysis of existing game data of master players. They had mentioned that the latter was better and effective than the former as interactive keyboard sessions were very time-consuming and when existing data was analyzed they could analyze different phases of games separately.

In their method of evaluation of each position before assigning a ranking value, they assigned values for each square and then they prepared some rules to assign values for each piece. For example, if a piece $\mathbf{W_i}$ is in a square $\mathbf{S_i}$ which is a center square, that piece has a relatively higher value because having the center control has some tactical advantage.

At one point, they state that "Material sacrifices for positional gain are not uncommon in master chess"[4], which means sometimes strong positional players sacrifice pieces for better positions. It tells us that the one who has the most (or most valuable) pieces may not be

the one who has the most advantage at that moment. Therefore in their method, Atkin and Witten have been very careful about that while assigning values for squares and pieces.

Although the results showed that the way they quantified the states of the chessboard needed improvements, their method is very important to consider in any case of quantifications of chessboard positions.

2.2 Popularity Distribution of Chess Openings

Bernd Blasius and Ralf Tönjes, in their research, perform a quantitative analysis of extensive chess databases to find out the distribution of frequencies of opening moves. They explain how complex human decision-making process is and there can be a number of factors that influence each choice. They state that "such (decision-making) processes are ubiquitous, ranging from one's personal life to business, management, and politics, and have a large part in shaping our life and society"[5]. They further say that investigating such human behavior becomes harder as there are no much data available to be analyzed because quantification of such human nature is not a very easy task. However, board games like chess provide a very good data sets, which are being directly generated as a result of human decision-making processes.

They further state that "The total number of different games that can be played, i.e., the game-tree complexity of chess, has roughly been estimated as the average number of legal moves in a chess position to the power of the length of a typical game, yielding the Shannon number $30^{80} \approx 10^{120}$. Obviously, only a small fraction of all possible games can be realized in actual play. But even during the first moves of a game, when the game complexity is still manageable, not all possibilities are explored equally often"[4]. However, after their research and tests, they have found that majority of chess games are distributed among a very small number of popular opening patterns. For example, for d=12, 80% of all games in the database are concentrated in about 23% of the most popular openings, where d is the number of initial moves.In their methodology, they have used a statistical approach to come up with those findings.

However, scope of their research was limited to opening moves. Going beyond the analysis of opening moves is tougher and requires a lot of research.

2.3 Cheating detection in chess

David J.Barnes and Julio Hernandez-Castro discussed the difficulties of detecting cheating in online chess games where players are not really present physically over the board. They show how cheat detection approaches which completely based on the moves of the chess pieces have a high risk of giving false positive results[6]. That means even for cheat detection, some higher level aspect of the game such as playing style is required to be considered.

In their research methodology, they used available chess game data, and fed them to a chess engine to get the best possible set of moves step by step. That is one important and efficient way of analyzing chess game data for any chess analysis work.

2.4 Move similarity analysis in chess programs

The environment around the game of chess has evolved tremendously with time and technology. Now there are a number of chess engines and programs which can play better than the best human player ever. With that, there are competitions for such programs as well. World Microcomputer Chess Championship (WMCC)[7], World Computer Chess Championship[8] and Top Chess Engine Championship[9] are some of most popular such competitions for chess programs. With these competitions, the need arose to find out if any program uses algorithms taken from other known programs.

Even if the source code of those programs are available to be analyzed, code level comparison does not give an important output as the same program can be written in many ways. Therefore, a higher level of comparison is required for this. Checking for move similarity is one of them. Here, the way one program moves pieces is compared with others. This move similarity is not about each move individually. It's more about sequences of moves, or in a more higher level, we could say it's the style of play which should be compared among each program.

D. Dailey, A. Hair, and M. Watkins have tried in their research[10] to find out whether move similarity is a good way of finding stolen algorithms. Their approach was to pick 25 chess engines and follow these steps for each engine.

1. Feed the same input for each engine

   eg. ***position startpos moves e2e4 d7d6; go depth 50;*** in UCI[11] code)
2. Let them analyze it for the same amount of time
3. Stop analyzing and get results (eg. ***stop*** in UCI[11] code)
4. Collect results. These results contain next best move and optionally the expected next move from the opponent.
5. Compare the sequences of results from each engine.

At the end of their research and tests, they, however, were able to identify that strong engines have higher move-matching than weaker ones, and similarly when time allotments are increased[10].

2.5 Identified Research Gap

Among all above chess related researches, only a very few number of researches have been conducted on chess playing styles related topics. Even among them, none of them are directly about chess playing styles. They rather can relate to playing styles. Therefore, this research is basically focusing on that particular gap and trying to analyze chess playing styles directly.

Chapter 3: Methodology and Solution

3.1 Raw Data Analysis

A chess game consists of a sequence of moves done by each player, one after the other. So a game is recorded as a list of moves. To record a move there should be a standard notation. In chess this notation is called Standard Algebraic notation (SAN).

3.1.1 Standard Algebraic notation (SAN)

SAN notation mainly consists of 3 parts;

- ➢ Square naming notation
- ➢ Piece naming notation
- ➢ Move naming notation

Square Naming Notation

Rows (ranks) and columns (files) of chess board are named as shown in Figure 3.1. There, files are named from **'a'** to **'h'**, while ranked are numbered from 1 to 8. Using that notation, each square in the chessboard can be identified with a letter and a number. For example, **g5** in Figure 3.1.

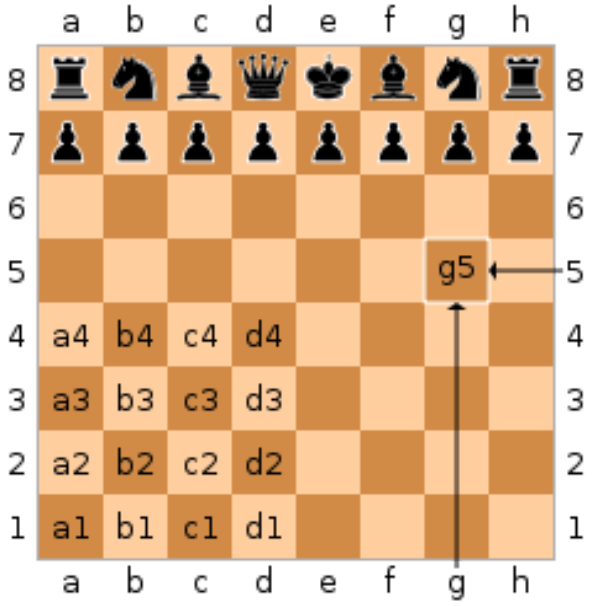

Figure 3.1: Square Naming in Chess Board

Piece Naming Notation

Pieces (except for pawns) are represented by the first letter (upper case) of their name. For example, K for King, Q for Queen etc. But N is used for Knight, as K is already given for King. Pawns are represented by the absence of a letter. (Table 3.1)

Piece	Notation
King	K
Queen	Q
Bishop	B
Knight	N (because K is already taken)
Rook	R
Pawn	[No notation]

Table 3.1: Notations of chess pieces

Move Naming Notation

A typical move is represented by the piece abbreviation followed by the square of arrival. For example, **Nf3** stands for a Knight moving to **f3** square. (Figure 3.2)

Figure 3.2: Nf3

In case of there are 2 pieces of the same type that can come to the same square, source file or rank is also added to the notation to avoid the ambiguity. For example, when 2 Knights are in c5 and e5, both can move to d3. So **Kd3** is ambiguous. Therefore the move is represented as either **Ncd3** or **Ned3**, depending on which Knight is moved. (Figure 3.3)

Figure 3.3: Ncd3 or Ned3

In SAN, a capture is represented by **'x'**. For example, a Knight capturing the piece which was on d4, is represented by **Nxd4** (Figure 3.4) .

Figure 3.4: Nxd4

3.1.2 Portable Game Notation (PGN)

The notation used to record chess games is called Portable Game Notation (PGN) It was introduced by Steven J. Edwards in 1993. PGN is a sequence of moves represented in SAN notation.

In addition to moves, PGN also consists of information about the game, such as player names, their ELO rating, event, site, date, result etc. PGN of a game starts with those information followed by the list of moves. The list of moves are numbers starting from 1. For each number, there is a move by each player.

This is a sample PGN of a chess game between Viswanathan Anand and Garry Kasparov in November, 1995.

```
[Event "PCA Intel-GP"]
[Site "Moskou rapid"]
[Date "1995.10.08"]
[Round "1"]
[White "Anand, V. (wh)"]
[Black "Kasparov, G. (bl)"]
[Result "1-0"]
[WhiteElo "2715"]
[BlackElo "2805"]
[ECO "B53"]
    1. e4 c5 2. Nf3 d6 3. d4 cxd4 4. Qxd4 Bd7 5. c4 Nc6 6. Qd2 g6 7. Be2 Bg7 8. O-O
Nf6 9. Nc3 O-O 10. Rb1 a6 11. b3 Qa5 12. Bb2 Rfc8 13. Rfd1 Bg4 14. Qe3 Nd7 15. Nd5
Bxb2 16. Rxb2 Bxf3 17. Bxf3 e6 18. Nc3 Rd8 19. Rbd2 Nde5 20. Be2 Nb4 21. h4 b5 22.
cxb5 axb5 23. Nxb5 Nbc6 24. a3 d5 25. exd5 Rxd5 26. Rxd5 exd5 27. b4 Qa4 28. Rxd5 1-0
```

3.2 Dataset

The dataset was taken from http://www.top-5000.nl/pgn.htm. The raw dataset had 2.2 million chess games and the size of the dataset was more than 1.5 Gigabytes. It consisted of games from number of international tournaments including World Chess Championships. Games in the dataset spanned from 1801 to 2013.

3.3 Feature Engineering

As mentioned in the previous section, recorded data of a chess game usually contains metadata about the game and its move sequence. Since these raw data can't be analyzed directly, they should be quantified. For that, features need to be identified. However, unlike in other games, feature extraction in chess is not so straightforward, as it's just a sequence of moves.

3.3.1 Identifying Features

Domain expertise is essential to identify features in a chess game. So inputs from a number of chess experts were required for this. One of the main challenges was to identify features which have relations to playing styles. When a game of chess is considered, it can be split it into 3 main phases.

1. Opening
2. Middlegame
3. Endgame

Opening is how a game is started. Since all games are started from the same positions, there are well- known patterns to start a game. Those openings even have names. Some of them are Ruy Lopez, Giuoco Piano, King's Gambit, Sicilian Defense and so on. Usually which opening to be used depends on the criticality of the game or the nature of the opponent. It hardly depends on the style of the player. The same is applied to the endgame as well. For example, when black has the king and a knight, and white has the king and a rook, for the white to win, it should try to get black king and knight to be on the same line and then white root should go between them. It's a well-known endgame strategy, which is independent of the player's style.

Middlegame is where a player can play the way they prefer, and due to the same reason, middlegame is what shows a player's playing style. Therefore, when extracting features, opening and endgame were ignored and only middle game was considered.

Feature extraction from middlegame can be done from different perspectives. Mainly those aspects can be categorized into 3 main sections[12]. They are,

1. **Material balance**

Material balance is the how values of pieces in the board. One is said to have the material advantage if the total value of their pieces is greater than the total value of the opponent's pieces. Having more material is important in the long run and helps to win the endgame[13]. However, sometimes it may not be much important in the short run as there can be tactical advantages one can take by sacrificing a valuable piece for a higher gain in the future. One example is a sacrificing queen to get a chance for a checkmate.

Depending on a player's playing style, the material value can go high and low in different phases of a game. For example, an aggressive player may have a high rate of losing material due to their tactics, but that usually become stable at the middle game. For a defensive player, that may not be the case.

Therefore, the material changing rate of a player, material exchanges between players etc. can be good measures one can take to measure different aspects of different playing styles.

Sometimes, the material value may not provide the exact advantages a player possesses. That's because different combinations of pieces may be of higher value than the other combinations even in case of the values of each combinations being similar.

Therefore, in addition to material values, piece combinations and the phase of the game can also provide important information about a player's style in a game.

2. **Space**

Space is the number of squares which is under the control of a player. Controlling a square is 2 folded. It's either the player has a piece on that square, or they can safely move a piece to that square[14].

Space gives a player more chances to move their pieces freely. That gives the player opportunity to do better moves to gain the advantage. That helps in both preparing for advanced attacks and defending opponent's attacks.

One important example of space is "center control". That is having the control of 4 center squares in the chess board. Having center control is very advantages for a player. We will be discussing about this later in this chapter.

The way a player uses space can be different based on the style of the player. For example, an aggressive player may try to acquire space faster than a positional player. So measures such as center control, space acquired by a player at a given stage of the game etc. can be important features to identify different playing styles.

3. **Initiatives**

Initiatives mean one having control of the game. In others words, if a player is only responding to their opponent's moves, but unable to initiate any attacks, that means the player is not controlling the game, but the opponent is[15]. To control the game, one should have time. Time in a chess game can be obtained if one is not getting attacked frequently by the opponent. One way of having that is by making moves which force the opponent to make several moves. That can buy time which helps to be initiative.

So it's a cycle of these 2 steps which depend on each other. The one who breaks the cycle gets the chance to be initiative. How a player achieves this can depend on the playing style of the player. For example, aggressive players may be more initiative than other types of players. On the other hand, defensive players may not be much initiative in the initial phase of the game.

However, initiatives is a complex aspect to be measured. Mostly it depends on both players' behaviors. Any approach to measure this will need to follow up on moves by each player and figure out a measurement.

In this project, we will be considering Material Balance and Space perspectives only. The reason is that due to the complexity of initiatives, it needs a separate and deeper study to identify features in that perspective. So it can be done as a future work of this project.

With the help of some chess experts, the following list of straightforward features and complex features was prepared.

Straightforward Features

- Elo rating of the player
- ECO (i.e. Chess Opening Code)
- Total number of moves
- Total number of checks
- Is Queen available after 10 steps?
- Is King castled after 10/20 steps?
- Number of pawn sacrifices after 10/20 steps
- Number of piece sacrifices after 10/20 steps
- Number of Rook sacrifices after 10/20 steps
- Number of Bishop sacrifices after 10/20 steps
- Number of Knight sacrifices after 10/20 steps
- Number of semi open files after 10/20 steps
- Number of King moves up to 20th move
- Number of Queen moves up to 20th move
- Number of Bishop moves up to 20th move
- Number of Knight moves up to 20th move
- Number of Rook moves up to 20th move
- Number of Piece moves up to 20th move
- Number of Pawn moves up to 20th move

Complex Features

- Center control
    - How many pawns have been attacking 4 center squares throughout the game?
    - How many pieces have been attacking 4 center squares throughout the game?
- Pawn structure
    - How many pawn islands after 10 steps?
    - Length of the longest pawn chain
    - Number of passed pawns

3.3.2 Feature Extraction

Extracting features from a PGN is a complex task. Doing that manually needs a huge amount of work. However, existing chess game engines could be helpful in this task to obtain information which can be useful to extract features. So the following chess engines, which support UCI (Universal Chess Interface) protocol, were evaluated.

- Komodo - Commercial
- Stockfish - Free and open source, supports up to 512 cores
- SugaR - Free and open source, supports up to 128 cores
- Houdini - Commercial
- Fire - Free, but not open source (anymore)
- Gull - Free and open source

After the evaluation, Stockfish was selected to proceed with, as it performs better than others, and was free, open source.

The UCI protocol, which was used to communicate with chess engines, was written to play games using the chess engine. Therefore the things such as predicting the opponent's moves, calculating and finding the best move etc. was very easier to do with UCI supported engines. But feature extraction of a given game was not that simple. First of all, PGN was parsed and then the games were fed to the chess engine via UCI protocol. Then, multiple commands were required to be sent to the chess engine to manipulate returned data to extract the features that were required. To avoid that complexity, a 3rd party python library (python-chess) was used to communicate with Stockfish engine.

With python-chess, most of the required features were directly extracted, while some required further processing with some custom python code.

3.3.3 Manual Categorization

The main purpose of this research is to identify chess playing styles and try to categorize games into those styles. For that, a model needed to be created, and that model needed to be trained. For that, a ground truth dataset was required. In this particular case, a set of games with known styles were needed. That needed help from experts of the domain. So basically what was required was to show a game to a chess expert and record the style they think the game has. But it was challenging because the games were in PGN format and it took time to understand those.

To make this process faster, PGN games were converted to interactive chess boards using a 3rd party service and a set of web forms were generated to get input (Figure 3.5).

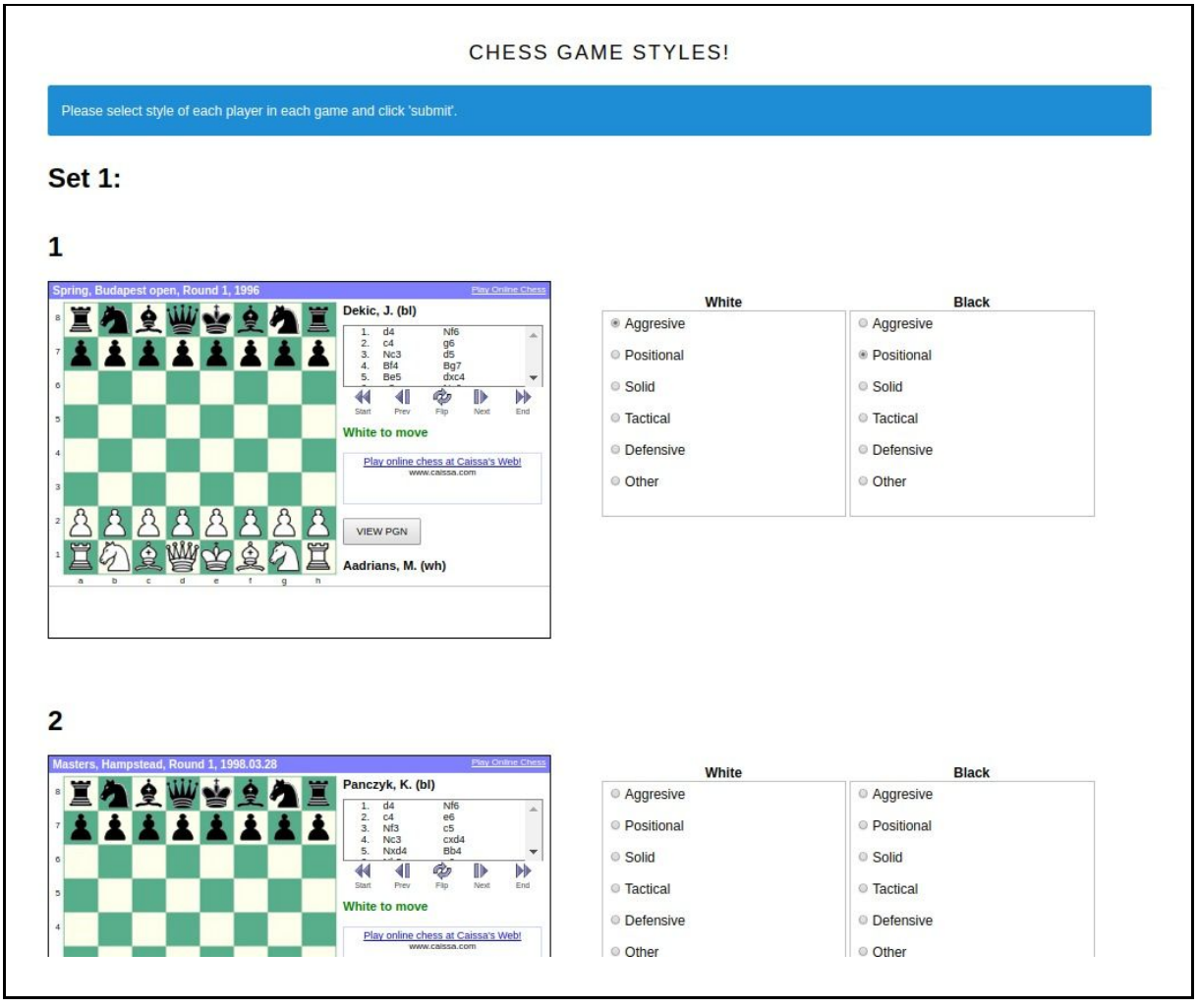

Figure 3.5: Interactive chess analysis forms

With this, chess experts could replay games in an interactive manner and give their opinion on game styles. Help was taken from 9 chess experts, and here is one of the games, categorized by them (Figure 3.6).

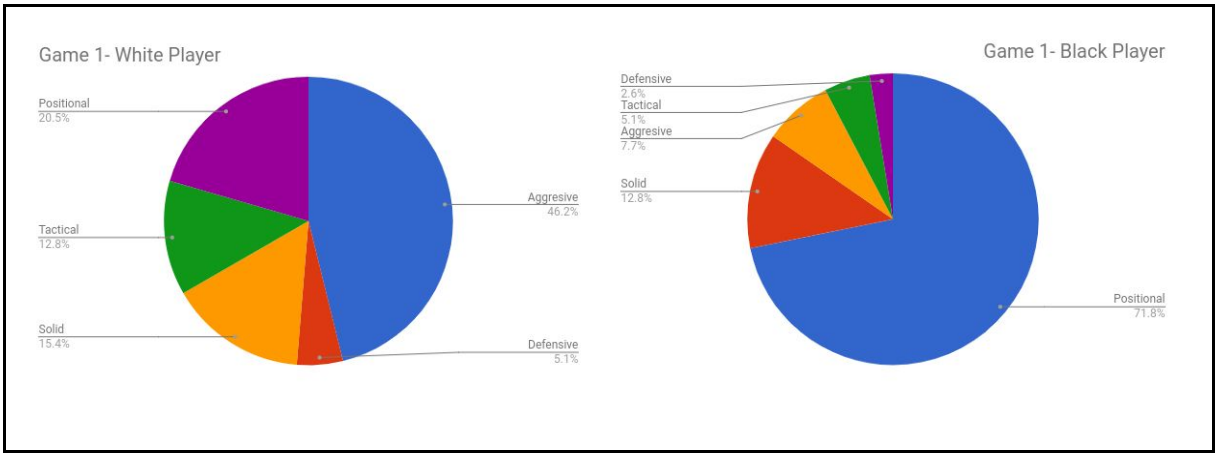

Figure 3.6: Games categorized by chess experts

However, this was very challenging due to following problems, and after categorizing about 50 games, it was realized that it's not practical.

1. The same game was categorized into completely different styles by different people (i.e. the decision was subjective to the person).
2. Most of the games were categorized into either 'Aggressive' or 'Positional'.
3. A lot of games were classified as no-style.
4. Categorizing a game took a lot of time and effort. (Categorizing 50 games took more than 2 weeks, and at least 1000 games were required to be categorized for a proper training set.)

Due to these challenges in manual categorization, classification of chess games based on styles was not practical at this stage. Therefore, it was decided to spend more time on clustering and try to get a natural categorization among games.

3.3.4 Data Preprocessing

Extracted features of games had certain anomalies. Therefore they had to be preprocessed and corrected before applying any algorithm on them. For example, some data entries had special characters, and those had to be removed, sanitized or replaced with some other characters

Certain metadata of games such as Event, Site, Location, Round, Result etc., which were not much related to this research, were also removed from the data at the beginning itself.

Since different games had different lengths, not all games had all features. For example, some games had less than 20 steps. In such cases, the features such as “Is King castled after 20 steps?” didn’t have any meaning. Therefore, such data was removed from the dataset.

Outliers were removed from the dataset and all features were scaled to reduce the variance differences between features because if it’s not done, some algorithms tend to bias to the features with high variances.

3.4 Analysis

3.4.1 Initial Clustering

After the data was preprocessed, they were ready for the initial analysis. The analysis was started with clustering using weka tool[16]. As the very first step, K-Means was tried with different numbers of clusters. (Table 3.2)

No of clusters	K-Means Clusters
2	0 855 (40%) 1 1305 (60%)
3	0 331 (15%) 1 1305 (60%) 2 524 (24%)

4	0 234 (11%) 1 1304 (60%) 2 342 (16%) 3 280 (13%)
5	0 155 (7%) 1 706 (33%) 2 599 (28%) 3 176 (8%) 4 524 (24%)

Table 3.2: K-Means clusters with different numbers of clusters

Since the primary target was to cluster data based on chess game styles, the intention was to compare these clustered with game styles. Since manual categorization didn't work, an assumption had to be made to proceed. There are international players well known for different playing styles. For example,

- Garry Kasparov - **Aggressive**
- Anatoly Karpov - **Positional**
- Tigran Petrosian - **Defensive**
- Peter Leko - **Solid**
- Mikhail Tal - **Tactical**

So, an assumption was made that above players' games can be categorized under the style each player is known for. There can be exceptions, but in most of the cases, this assumption will be valid.

However, to confirm the player-to-style mapping is correct, again the help of chess experts was used. A set of online forms (Figure 3.7) were created for this purpose and their responses were recorded.

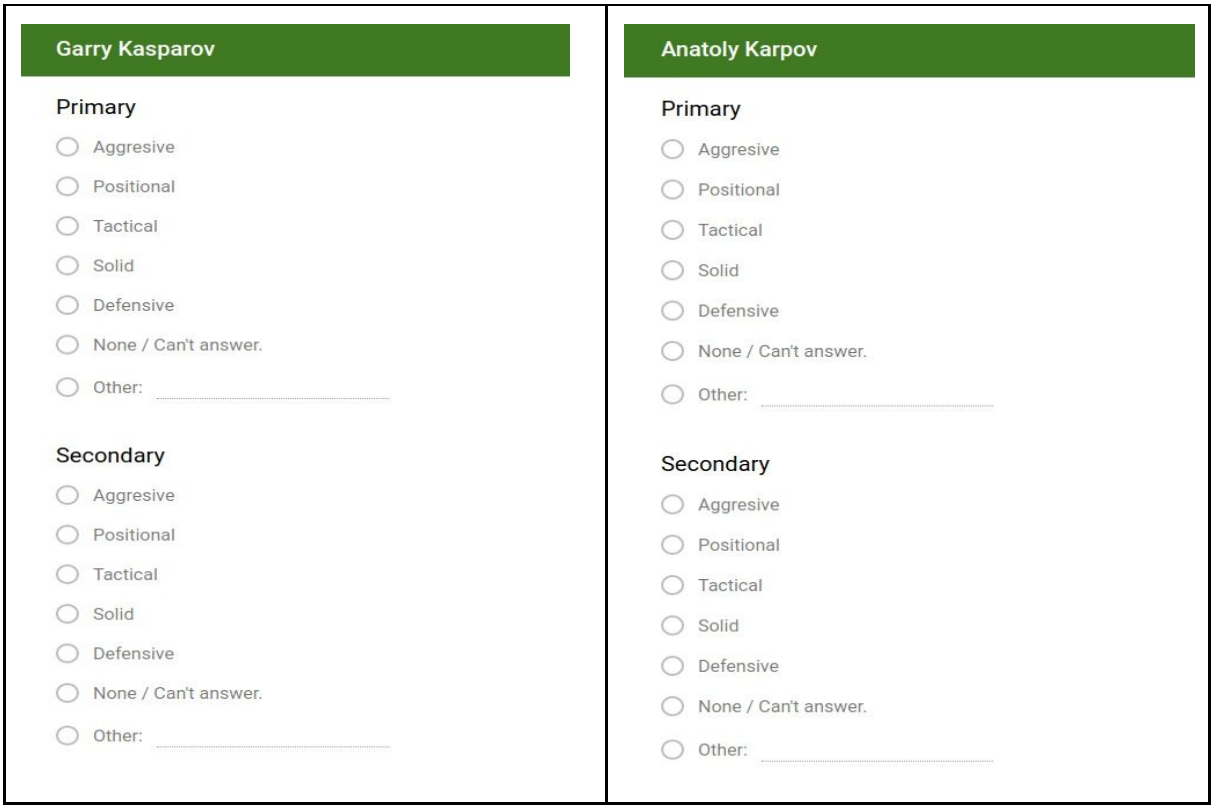

Figure 3.7: Online form for Player-to-Style mapping

The responses from experts (Figure 3.8) clearly confirmed my player-to-style mapping was correct.

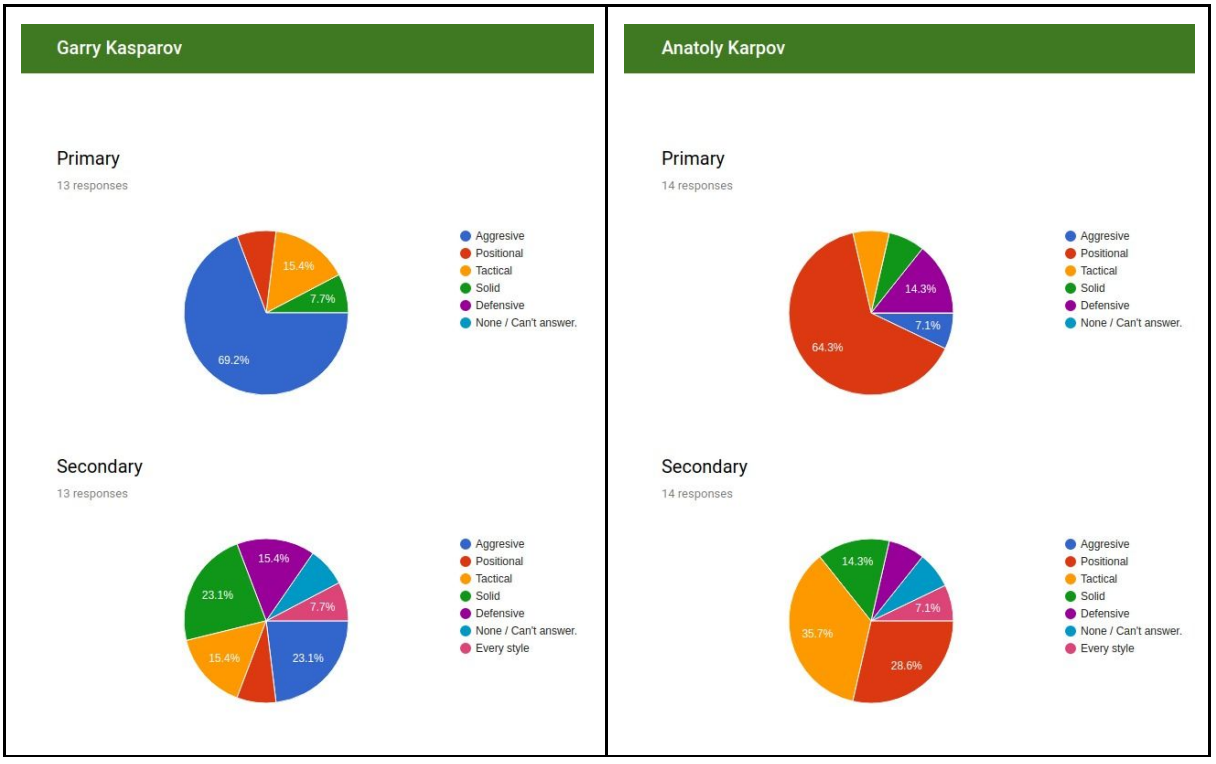

Figure 3.8: Player-to-Style mapping results

3.4.2 New Dataset

To go in the new path, the dataset had to be changed. From the raw game data, only the games of aforementioned 5 players were filtered out. The new dataset roughly had 450 games for each player (or style). Figure 3.9 shows the distributions of data in each feature.

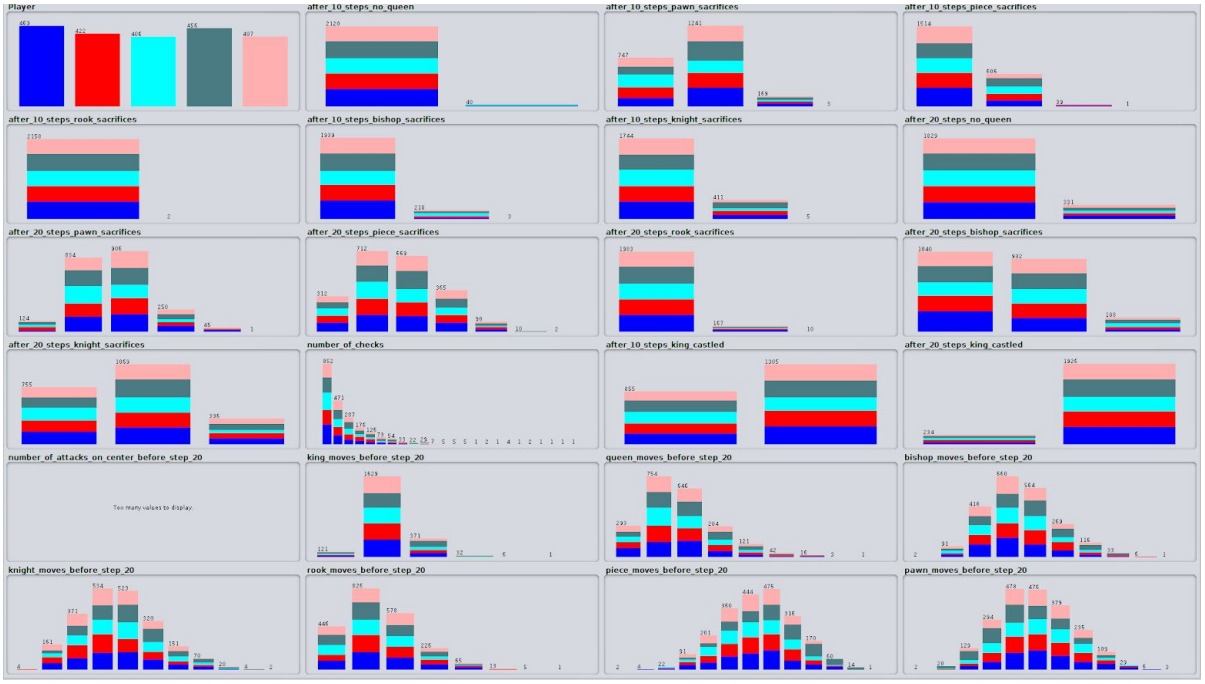

Figure 3.9: Features of new dataset

I again used weka to cluster with K Means and validated results with known styles.

3.4.3 K-Means Clustering

K-Means was run again with different cluster sizes, but the error rate was very high even in the new approach. Output for 5-clusters is shown in Table 3.3.

Clustered Instances:

0 155 (7%)

1 706 (33%)

2 599 (28%)

3 176 (8%)

Classes to Clusters:

0	1	2	3	4	<-- assigned to cluster
30	161	135	43	100	Kasparov. G. (Aggressive)
38	141	113	30	100	Karpov. A. (Positional)

<table>
<tr><td>4 524 (24%)</td><td>
<table>
<tr><td>39</td><td>126</td><td>91</td><td>31</td><td>119</td><td>Petrosian. T. (Defensive)</td></tr>
<tr><td>26</td><td>137</td><td>138</td><td>43</td><td>112</td><td>Leko. P. (Solid)</td></tr>
<tr><td>22</td><td>141</td><td>122</td><td>29</td><td>93</td><td>Tal. M. (Tactical)</td></tr>
</table>
</td></tr>
<tr><td colspan="2">Cluster 0 <-- Karpov. A.(Positional)
Cluster 1 <-- Kasparov. G. (Aggressive)
Cluster 2 <-- Leko. P. (Solid)
Cluster 3 <-- Tal. M. (Tactical)
Cluster 4 <-- Petrosian. T. (Defensive)</td></tr>
<tr><td colspan="2">Incorrectly clustered instances : 1675.0 77.5463 %</td></tr>
</table>

Table 3.3: K-Means clustering results

3.4.4 Advanced clustering

Clustering approach that should be used on a dataset is heavily affected by the nature of the dataset. Not every dataset can be clustered using any random clustering algorithm. The initial effort of clustering above failed due to this very reason. Therefore, some high-level idea on how data is spanned in an N-dimensional space was required first before applying any clustering algorithm. Visualization of the dataset would become very helpful in such a case.

3.4.5 Visualization

The dataset had 23 features, which meant it needed 23-dimensional space to represent them. However, for a human to understand a visualization, it should be either in 2D or 3D space. Therefore, some dimension reduction approach was needed for this.

After analyzing available dimension reduction methodologies, Principal Component Analysis (PCA) was selected as it was easy and straightforward.

3.4.6 Principal Component Analysis (PCA)

PCA was applied on the dataset and reduced the dimension to 2 so that it can be viewed on a 2D space. (Figure 3.10)

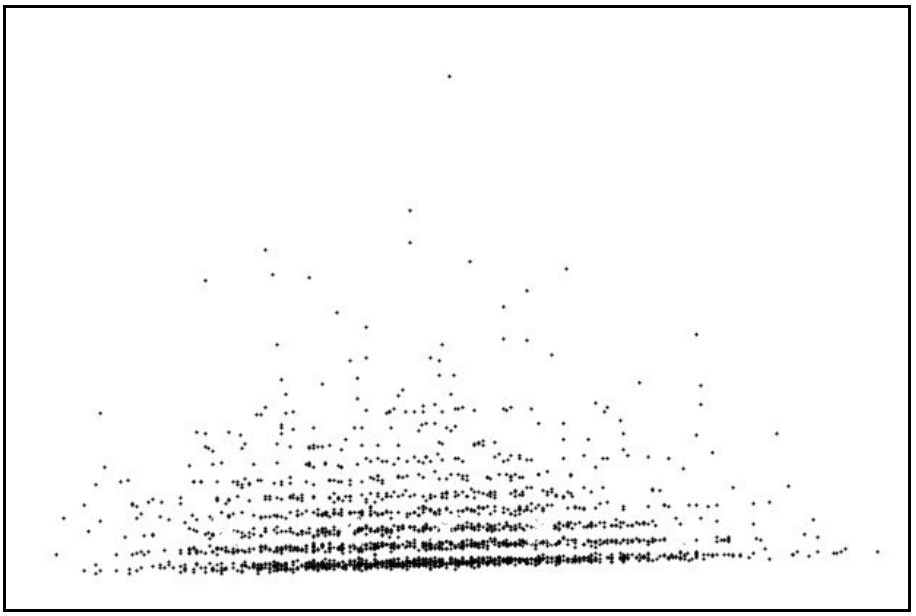

Figure 3.10: Visualized dataset (dimension reduced)

This gave a clear picture how the dataset is naturally clustered. Then, to see how K-Means performs on this dimension reduced dataset, K-Mean was applied on top of it. (Figure 3.11 and 3.12)

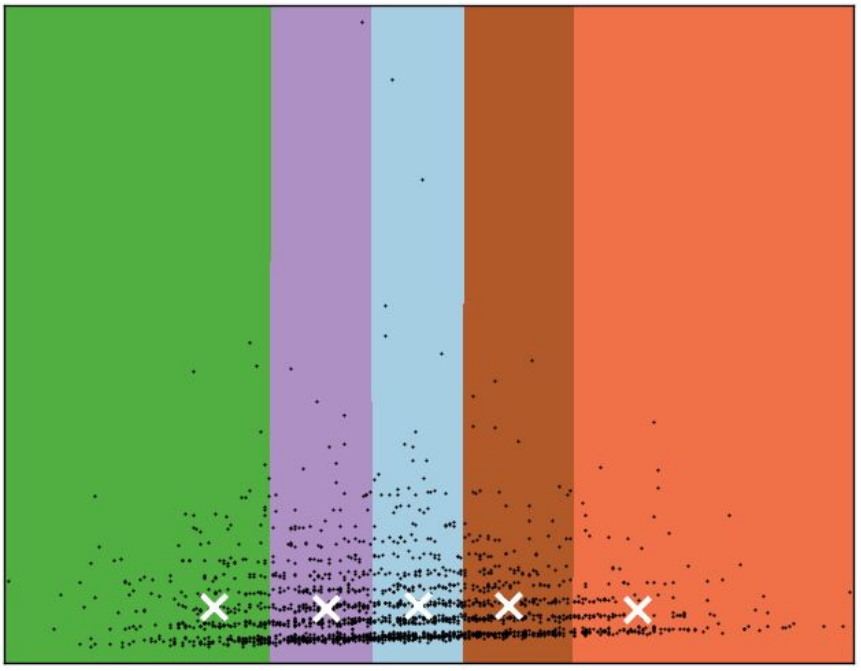

Figure 3.11: K-Means applied on dimension reduced data

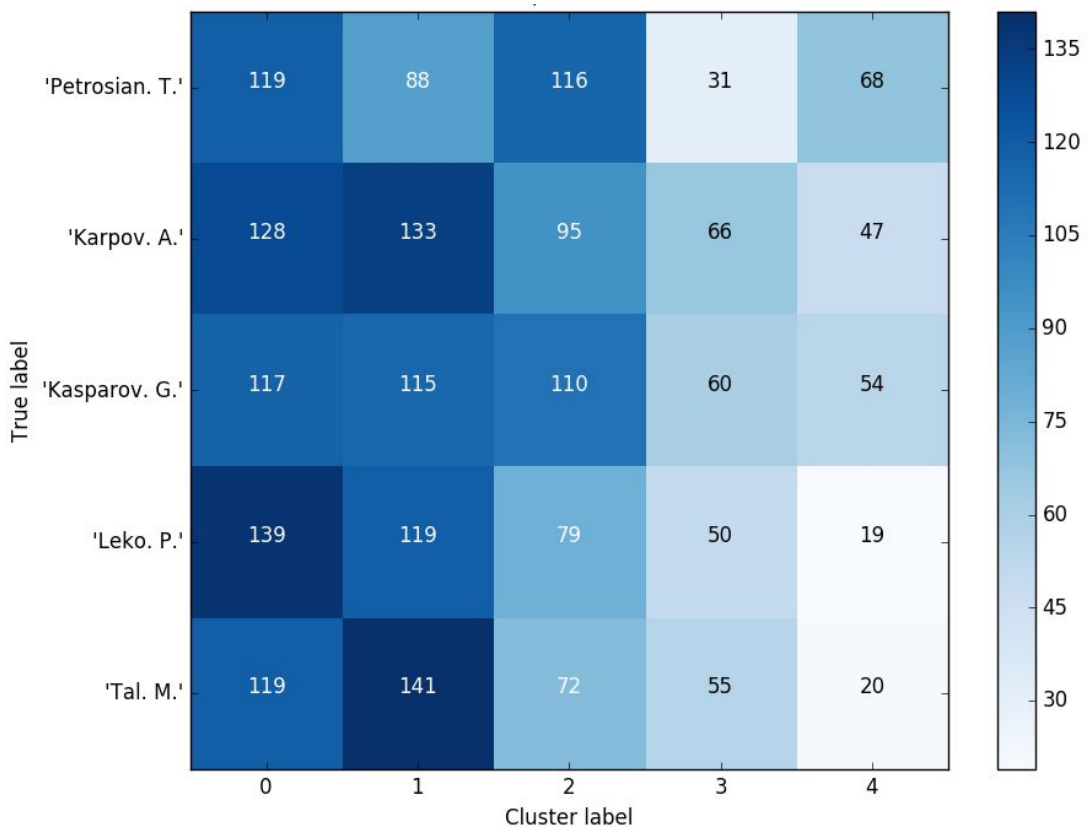

Figure 3.12: Confusion matrix of K-Means applied on dimension reduced data

This showed why K-Means gave bad results in the initial clustering effort too. The natural clusters were elongated, but K-Means wasn't able to capture that because K-Means is naturally good for spherical clusters, but not much good for other shapes. When the real clusters are in shapes of non-circular, K-Means tries to fit then into circles forcefully, which gives artificial and inaccurate results.

Then it was decided to apply Gaussian Mixture Models (GMM) as it usually performs better when the clusters are in shape of ellipses.

3.4.7 Gaussian Mixture Models (GMM) with Expectation Maximization (EM)

A Gaussian mixture model is a probabilistic model which assumes that data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. That means it assumes each data cluster individually forms a Gaussian distribution. Therefore, it is able to cluster under that assumption GMM was applied to the dataset.

Gaussian mixture models are an extension of K-Means where clusters are modeled with Gaussian distributions, so we have not only their mean but also a covariance that describes their elongated shape. Then we can fit the model by maximizing the likelihood of observed data with Expectation Maximization (EM) algorithm. Expectation Maximization algorithm assigns data points to clusters with probability values.

In Gaussian mixture models, the probability distribution is defined by the weighted average of each individual components (i.e. clusters) which are Gaussian distributions (Eq. 3.1).

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x \mid \mu_i, \sigma_i) \tag{3.1}$$

Each component has a mean(μ_i), a variance/covariance(σ_i) and a size(Φ_i). Here, covariance can have different types. There are 4 main types.

1. **Full**: In this, each component may independently adopt any position and shape.
2. **Tied**: In this type, all components share the same general covariance matrix, which means they have the same shape, but the shape may be anything.
3. **Diagonal**: This means the contour axes are oriented along the coordinate axes and each component has its own covariance matrix.
4. **Spherical** is a "diagonal" situation with circular contours (spherical in higher dimensions, hence the name).

Which type to be used depends on the dataset itself. For our dataset, all 4 types of covariances were used and them compared visually (Figure 3.13) and mathematically (Table 3.4).

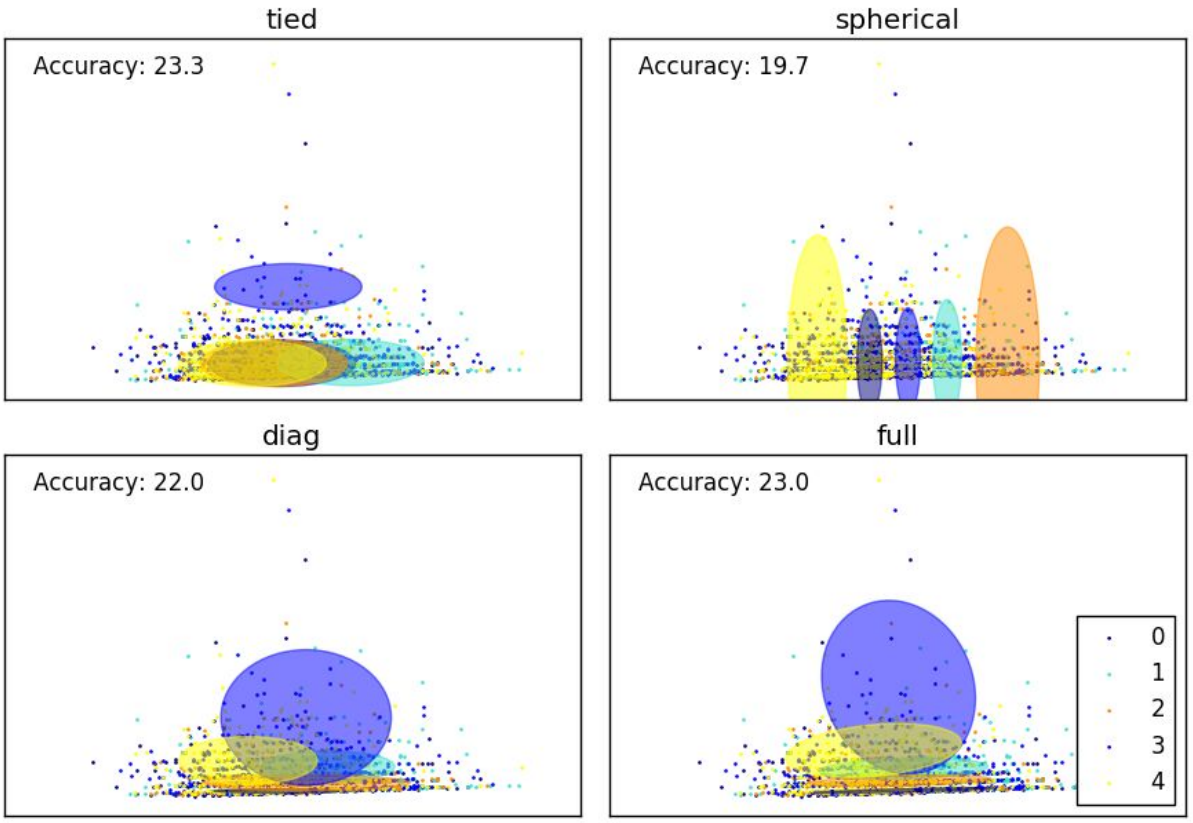

Figure 3.13: GMM applied on dimension reduced data

Tied

Clusters	0	1	2	3	4
Labels					
0	266	181	22	0	0
1	252	121	49	0	0
2	316	0	79	0	11
3	0	0	40	230	186
4	0	0	31	0	376

Spherical

Clusters	0	1	2	3	4
Labels					
0	113	81	61	110	104
1	83	102	82	106	49
2	116	63	35	114	78
3	109	91	73	95	88
4	134	63	29	90	91

Diagonal

Clusters	0	1	3	4
Labels				
0	131	257	64	17
1	117	225	64	16
2	96	206	84	20
3	160	217	61	18
4	98	243	52	14

Full

Clusters	0	1	3	4
Labels				
0	55	270	130	14
1	63	254	97	8
2	59	265	73	9
3	58	243	139	16
4	39	258	100	10

Table 3.4: Confusion matrices of GMM applied on dimension reduced data

Usually, in most of the cases, the type “Full” gives better results. But confirming that which type should be used must depend on the dataset, it was clear that “Tied” covariance was giving clear better results than the other 3 types, for this particular dataset.

Another important observation here is that even if we tried to cluster data into 5 clusters, both Diagonal and Full methods has only 4 clusters. It gives us a hint that 5 may not be the best number of clusters, that we should be using for clustering. We will further discuss this in a later chapter.

3.4.8 Self Organizing Maps (SOM)

The Self-Organizing Map is one of the most popular neural network unsupervised clustering models. It belongs to the class of competitive learning networks. The Self-Organizing Map was developed by professor Kohonen [17]. It provides a topology preserving mapping from the high dimensional space to a 2-dimensional neuron map [18]. Neurons in the 2-dimensional plain are connected to each other and they are affected by the input values.

The algorithm starts with a random input. Then a winner neuron is selected which is the closest to the selected input value. Then the value of the winner neuron is updated so that it is more similar to the input value. When the value of winner neuron is updated, its neighboring neurons are also updated as they are connected with each other. This process is followed for all inputs and at the end, all neurons in the two-dimensional lattice come to an equilibrium state. At this point, all the input data points which are closer to each neuron are considered to belong to a single class.

Self-Organizing Maps were applied to the selected chess dataset and clustering was attempted to see if we can observe clear data clusters. However, the results were not satisfactory as even though there were multiple clusters, more than 95% of the input points were closer to a single neuron. The same was attempted with different sizes of neuron maps, but the result was more or less the same.

3.4.9 Cluster Evaluation

Evaluation of Covariance Types

Cluster evaluation based on a standard criteria is important before deciding which clustering method works best for a particulate dataset. Cluster evaluation can be done in 2 ways. Those are internal cluster validity indices and external cluster validity indices. The latter can be used only if the true labels of the dataset is known.

In our case, since the true labels are based on an assumption, first we have to give priority to internal cluster validity indices which provide information on how good the clusters are (i.e. how similar the points in a single cluster and how dissimilar the points across different clusters). Once we pick a suitable clustering method using internal cluster validity indices, then external cluster validity indices can be used to find how close the identified clusters are to the true categories made under the initial assumption.

For this analysis, following 2 popular internal cluster validity indices were used.

1. **Silhouette Coefficient**: Silhouette analysis provides information about the separation distance between each cluster. The silhouette plot presents a measure of how close each point in one cluster is to points in the neighboring clusters. This measure has a range of [-1, 1].

   If the sample being tested is far away from the closeby clusters, the Silhouette coefficient goes near +1. A value of 0 means that the sample is on or very close to the decision boundary between two closeby clusters and negative values mean that it's highly likely that those samples are assigned to a wrong cluster.

2. **Calinski-Harabaz Index:** This index represents how dense the clusters are and how well they are separated from each other, which actually represent the standard concept of a cluster[19]. The higher the value of the index, the better the clustering result.

   For k clusters, the Calinski-Harabaz score s is given as the ratio of the between-cluster dispersion mean and the within-cluster dispersion. (Eq. 3.2)

$$s(k) = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \times \frac{N-k}{k-1} \tag{3.2}$$

where B_K is the between group dispersion matrix and W_K is the within-cluster dispersion matrix defined by Eq. 3.3 and Eq. 3.4 respectively.

$$B_k = \sum_q n_q (c_q - c)(c_q - c)^T \tag{3.3}$$

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T \tag{3.4}$$

with N be the number of points in our data, C_q be the set of points in cluster q, c_q be the center of cluster q, c be the center of E, n_q be the number of points in cluster q.

The clusters given by Gaussian Mixture Models (GMM) with Expectation Maximization (EM) were evaluated with aforementioned methods.

	Tied	**Spherical**	**Diagonal**	**Full**
Silhouette Coefficient	**0.102**	-0.106	-0.087	-0.042
Calinski Harabaz Index	**43.082**	6.227	28.287	40.806

Table 3.5: Internal cluster validity indices of GMM

The results (Table 3.5) were aligned with the confusion matrices and visual graphs we saw earlier. The "Tied" covariance type gives clearly better results in both methods.

Now we know the clusters we observed are in good shape. That means they are clearly separated from the others and each cluster has very similar datasets internally. Then the next step is to find out if these clusters are aligning with the true labels that were assigned based on the initial assumption. For this, we used following external cluster validity indices.

1. **Homogeneity**: This represents how similar the members of a particular cluster are. In order to satisfy our homogeneity criteria, a clustering must assign **only** those data points that are members of a single class to a single cluster. That is, the class distribution within each cluster should be skewed to a single class, that is, zero entropy.[20]
2. **Completeness**: This represents if all members of a given class are assigned to the same cluster. Completeness is symmetrical to homogeneity. In order to satisfy the completeness criteria, a clustering must assign **all** of those data points that are members of a single class to a single cluster.[20]
3. **V-measure**: This is defined as the harmonic mean of homogeneity and completeness of the clustering[20]. The harmonic mean H of the positive real numbers $x_1, x_2, \ldots, x_n$ is defined to be Eq. 3.5.

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}} = \left(\frac{\sum_{i=1}^{n} x_i^{-1}}{n} \right)^{-1} \tag{3.5}$$

4. **Adjusted Rand Index (ARI)**: The Rand Index (RI) is a measure of the similarity between two clusters[21]. The Adjusted Rand Index is a form of Rand Index and is chance-corrected. That means the RI score is adjusted in a way that a random result (i.e. result by chance) gets a score of 0 (i.e. invalidated).

Let's consider the following contingency table where X_i and Y_i are elements of 2 different clusters, and n_{ij} is the number of elements common in both clusters.

$X \backslash Y$	Y_1	Y_2	$\ldots$	Y_s	Sums
X_1	n_{11}	n_{12}	$\ldots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\ldots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\ldots$	n_{rs}	a_r
Sums	b_1	b_2	$\ldots$	b_s	

Now, the Adjusted Rand Index (ARI) can be expressed like this. (Eq. 3.6)

$$\overbrace{ARI}^{\text{Adjusted Index}} = \frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Index}} - \overbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}^{\text{Expected Index}}}{\underbrace{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]}_{\text{Max Index}} - \underbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}_{\text{Expected Index}}} \tag{3.6}$$

5. **Adjusted Mutual Information based score**: The Mutual Information is a measure of how much one variable depends on another. If this value is higher, that means the 2 variables are highly dependent. When the 2 variables are completely independent, mutual information is 0. In this method, mutual information is used to measure the similarity between 2 clusters. The Mutual Information between clusterings U and V is given in Eq. 3.7.

$$MI(U,V) = \sum_{i=1}^{|} U| \sum_{j=1}^{|} V| \frac{|U_i \cap V_j|}{N} \log \frac{N|U_i \cap V_j|}{|U_i||V_j|} \tag{3.7}$$

Here, $|U_i|$ is the number of the elements in cluster U_i and $|V_j|$ is the number of the elements in cluster V_j. Once this is corrected for chance just like in the case of ARI above, we get Eq. 3.8. There we use expected value for the calculation.

$$AMI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{\max\{H(U), H(V)\} - E\{MI(U,V)\}} \tag{3.8}$$

6. **Fowlkes-Mallows score**[22]: This expresses the similarity between 2 clusters. It is calculated using the number of true positives, false positives and false negatives. Fowlkes and Mallows introduced their index as a measure for comparing hierarchical clusterings. However, it can also be used for flat clusterings since it consists in calculating an index B_i for each level i=2,...,n−1 of the hierarchies in consideration and plotting B_i against i [23]. The measure B_i is easily generalized to a measure for clusterings with different numbers of clusters. The generalized Fowlkes–Mallows Index is defined by Eq. 3.9.

$$FM = \sqrt{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}} \tag{3.9}$$

Here, TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. As per above formula, since Fowlkes-Mallows score is proportional to the count of true positive elements, a higher the FM value represents a better clustering.

The clusters given by Gaussian Mixture Models (GMM) with Expectation Maximization (EM) were evaluated with aforementioned methods.

	Tied	**Spherical**	**Diagonal**	**Full**
Homogeneity	**0.464**	0.013	0.005	0.005
Completeness	**0.509**	0.013	0.007	0.007
V-measure	**0.485**	0.013	0.005	0.006
Adjusted Rand Index (ARI)	**0.400**	0.009	0.002	-0.000
Adjusted Mutual Information based score	**0.432**	0.012	0.002	0.001
Fowlkes - Mallows score	**0.550**	0.235	0.291	0.341

Table 3.6: External cluster validity indices of GMM

Confirming what we already have seen in the confusion matrices and visual graphs, “Tied” covariance type gives clearly better results in all 6 methods. (Table 3.6)

Evaluation of the number of clusters

So far we were under the assumption that since we used games of 5 players of 5 knows styles, the natural clustering would give better results if we stick to 5 as the cluster number. But that may not be true for always. The natures of certain styles could be overlapping. For example, the styles “Solid” and “Defensive” may have overlapping natures while styles “Tactical” and “Positional” may have similar natures in different ways. To find if there can be better ways of clustering in terms of the number of clusters, GMM with EM was applied for the dataset with different numbers of clusters and evaluated the clusters with Silhouette Coefficient.

Number of clusters	Silhouette Coefficient
2	0.0675023370311
3	0.0214370462517
4	**0.2265011588981**
5	0.102359226614
6	0.0367574663978
7	0.0864429992963

Table 3.7: Silhouette Coefficients of different cluster counts with GMM

The results of the evaluation (Table 3.7) shows that the assumption we made is slightly incorrect. We expected it to give better results when the number of clusters was 5. But it has given better results when the number of clusters id 4. That means 2 of the styles we initially took into consideration shows similar and overlapping natures.

The visual representation of this is shown below. (Figure 3.15 - 3.19)

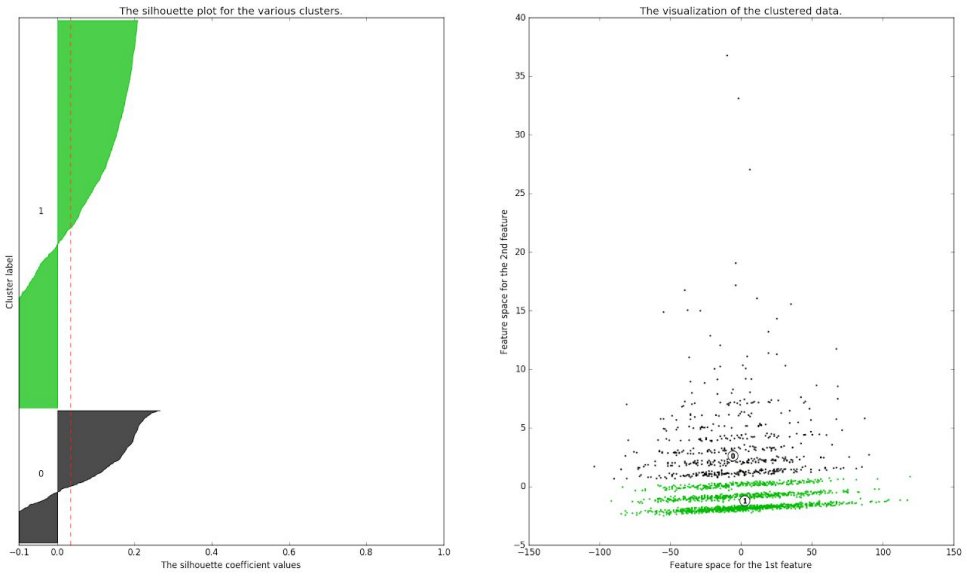

Figure 3.15: Silhouette Analysis for 2 clusters

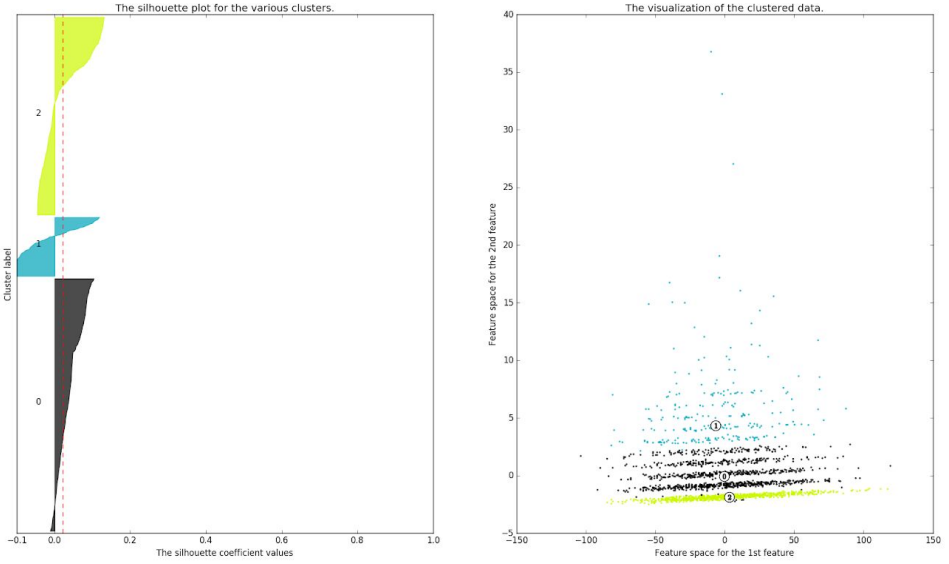

Figure 3.16: Silhouette Analysis for 3 clusters

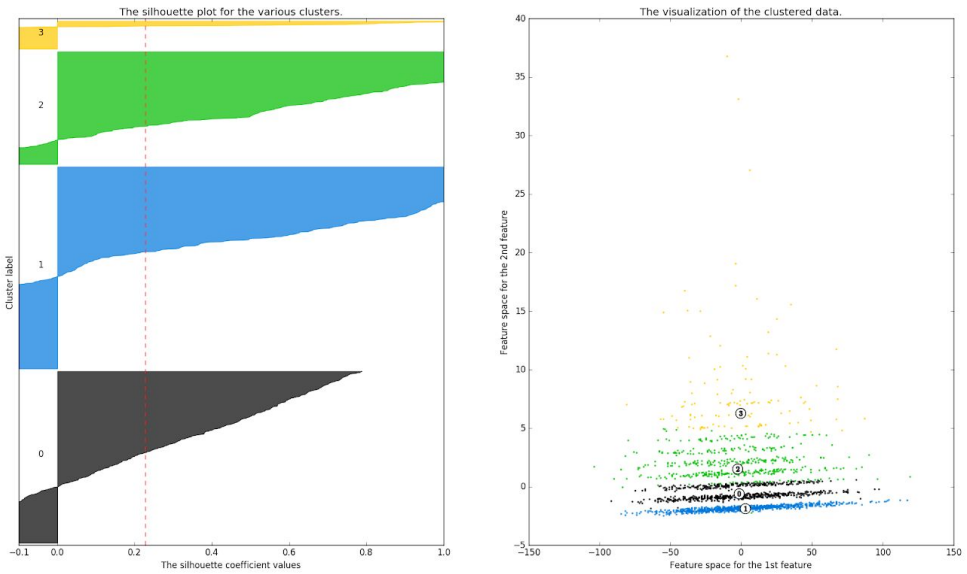

Figure 3.17: Silhouette Analysis for 4 clusters

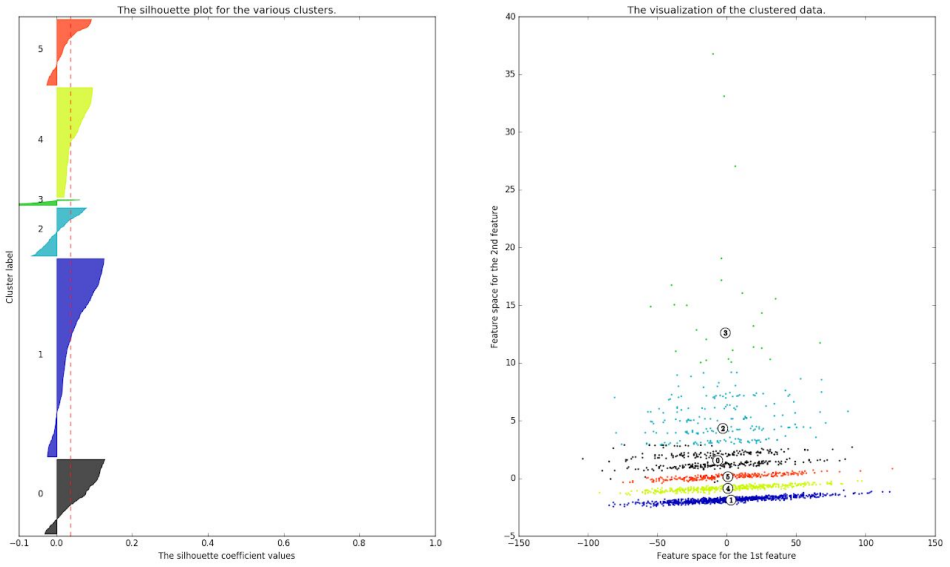

Figure 3.18: Silhouette Analysis for 5 clusters

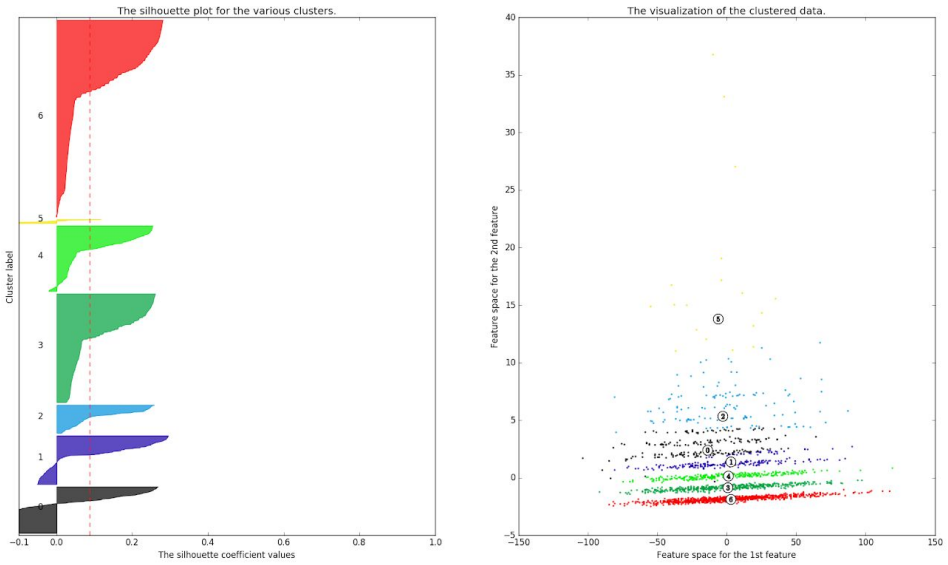

Figure 3.19: Silhouette Analysis for 6 clusters

Since it's unreliable to depend on just 1 evaluation method, **Calinski Harabaz Index** was also used to find out the optimum value for the number of clusters.

GMM with EM was applied for the dataset with different numbers of clusters and evaluated the clusters with Calinski Harabaz Index. (Table 3.8)

Number of clusters	Calinski Harabaz Index
2	23.6562639051
3	12.7363698381
4	41.469459849
5	**43.0823801635**
6	6.79770375493
7	11.60164274

Table 3.8: Calinski Harabaz Index of different cluster counts with GMM

A visual representation of above is shown below. (Figure 3.20 - 3.25)

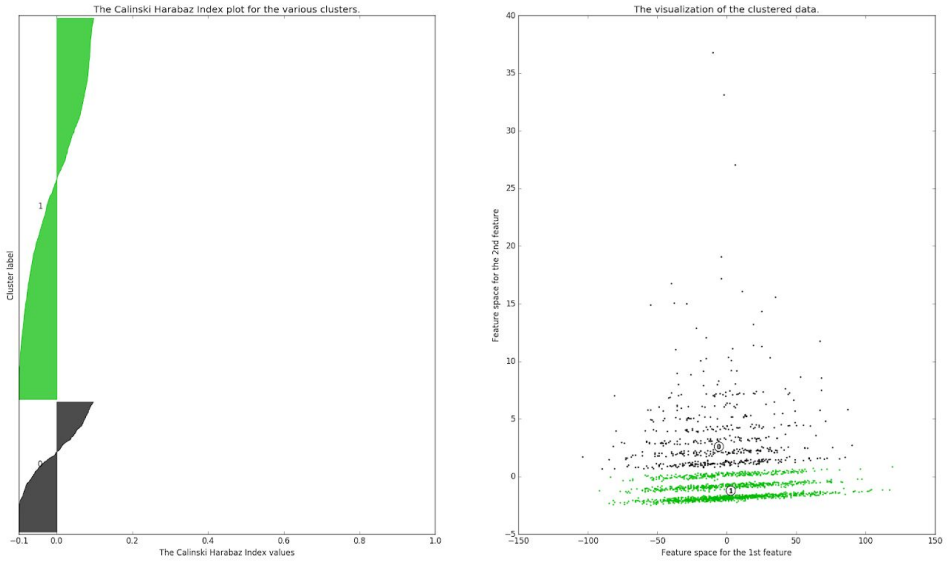

Figure 3.20: Calinski Harabaz Analysis for 2 clusters

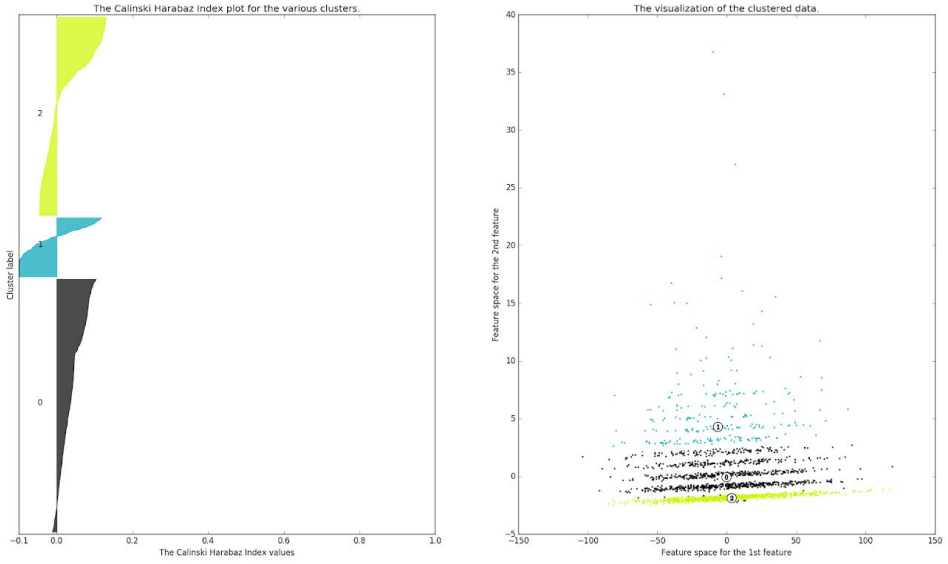

Figure 3.21: Calinski Harabaz Analysis for 3 clusters

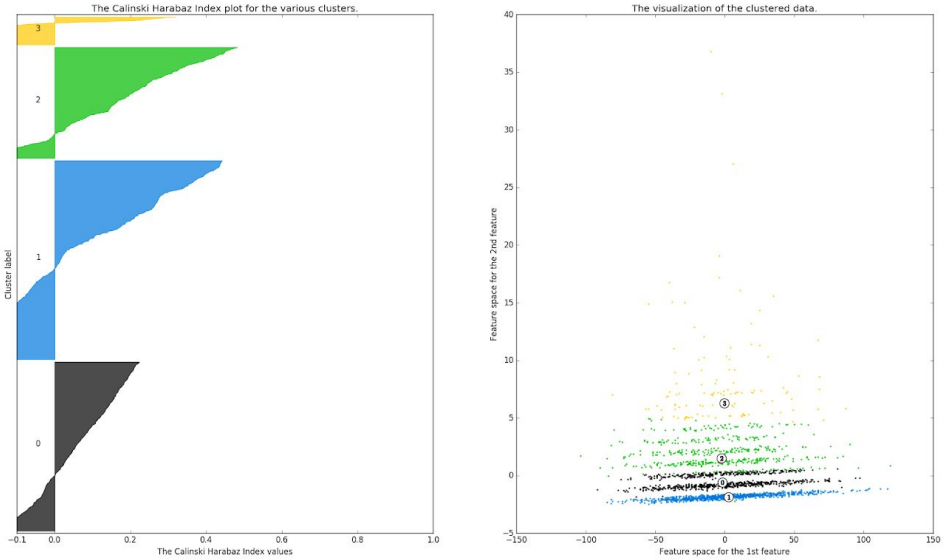

Figure 3.22: Calinski Harabaz Analysis for 4 clusters

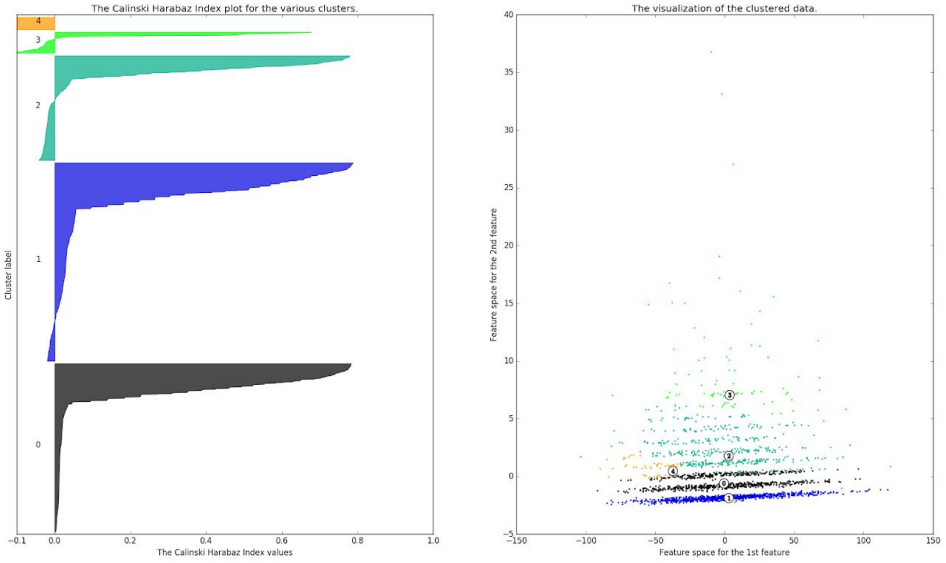

Figure 3.23: Calinski Harabaz Analysis for 5 clusters

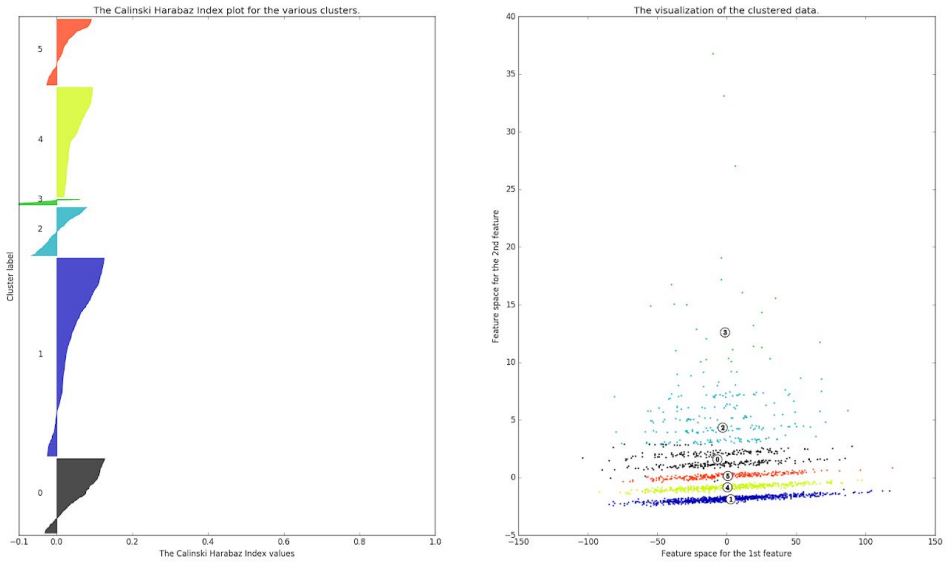

Figure 3.24: Calinski Harabaz Analysis for 6 clusters

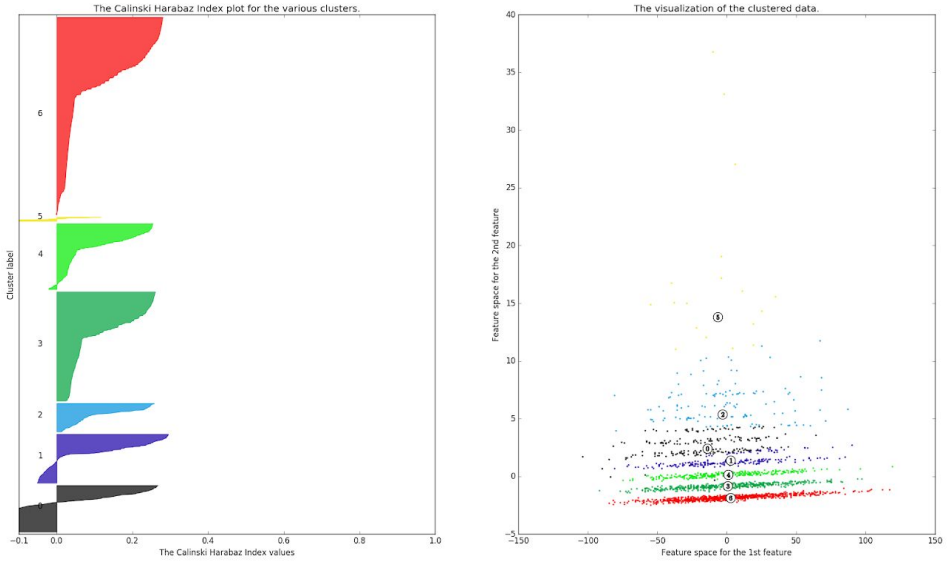

Figure 3.25: Calinski Harabaz Analysis for 7 clusters

Here, the result is a bit different from the previous one. In the Silhouette analysis, we observed that the optimum value for the number of clusters was 4. But here, what we observe is 5. However, in case of Silhouette analysis, 4-clusters was a clear winner with a 2x value than 5-clusters.

i.e. **0.227** (4-clusters) **> 2 x 0.102** (5-clusters)]

But in case of Calinski Harabaz Index, 5-clusters is winning only by a very small margin.

i.e. **43.082** (5-clusters) ≈ **41.469** (4-clusters)

Since we have to pick one value out of 4 and 5 here, picking 4 as the optimum value technically makes sense due to above fact.

In the previous subchapter (3.4.4.1), we evaluated the clusters which were clustered into 5. But since now we know the optimum value is 4, we have to do the evaluation again for 4 clusters. Hence it was carried out and the results were like this (Table 3.9).

Tied

Clusters / Labels	0	1	2	3
0	358	89	22	0
1	133	240	49	0
2	36	179	181	10
3	0	0	40	416
4	0	0	31	376

Full

Clusters / Labels	0	1	3
0	11	317	141
1	4	301	117
2	8	304	94
3	14	335	107
4	8	288	111

Diagonal

Clusters / Labels	0	1	3
0	57	258	154
1	53	225	144
2	57	206	143
3	51	217	188
4	37	243	127

Spherical

Clusters / Labels	0	1	2	3
0	142	106	83	138
1	72	136	110	104
2	109	97	50	150
3	125	114	96	121
4	129	90	38	150

Table 3.9: Confusion matrices of GMM applied for 4 clusters

To the naked eye, it seemed Tied covariance type still gives better results over other the others. However, the how good it was should be measured using internal and external cluster validity indices.

The internal cluster validity index evaluation results for the clusters given by Gaussian Mixture Models (GMM) with Expectation Maximization (EM) were like this. (Table 3.10)

	Tied	Spherical	Diagonal	Full
Silhouette Coefficient	**0.227**	0.024	-0.036	-0.013
Calinski Harabaz Index	**41.469**	5.816	28.287	40.806

Table 3.10: Internal cluster validity indices of GMM for 4 clusters

This proves that "Tied" covariance gives similarly better results even when the number of clusters is 4. That means, with "Tied" covariance the clusters are separated from each other in a clear way and each cluster has similar elements.

The next step is calculating external cluster validity indices and find out how good the alignment of clusters with the real data labels. For that, previously used external cluster validity indices were calculated again for the new clusters.

	Tied	Spherical	Diagonal	Full
Homogeneity	**0.433**	0.014	0.003	0.002
Completeness	**0.535**	0.016	0.005	0.005
V-measure	**0.479**	0.015	0.003	0.003
Adjusted Rand Index (ARI)	**0.400**	0.009	0.002	- 0.000
Adjusted Mutual Information based score	**0.432**	0.012	0.002	0.001
Fowlkes - Mallows score	**0.550**	0.235	0.291	0.341

Table 3.11: External cluster validity indices of GMM for 4 clusters

As per the Table 3.11, it's again clear that "Tied" covariance is the most suitable covariance type to be used with Gaussian Mixture Models for this dataset. Now let's have a look at the confusion matrix of "Tied" covariance type in case of 4 clusters. (Table 3.12)

Clusters Labels	0	1	2	3
0	358	89	22	0
1	133	240	49	0
2	36	179	181	10
3	0	0	40	416
4	0	0	31	376

Table 3.12: Confusion matrices of GMM with Tied covariance applied for 4 clusters

Here, labels from 0 to 4 represent Aggressive, Positional, Defensive, Solid and Tactical styles respectively. Looking at the confusion matrix, we can clearly see that Cluster 0 represents Aggressive style (with 358 games), Cluster 1 represents Positional style (with 240 games) and cluster 2 represents Defensive style (with 181 games). An important observation we can make here is that both Solid and Tactical styles are divided into Cluster 2 and Cluster 3 in similar ratios. And this ratio highly tends towards the Cluster 3. Therefore, we can conclude that Cluster 3 represents both Solid and Tactical styles.

3.4.10 Classification

As mentioned in a previous section, clustering was the main focus of this research and classification based on that was not much focused. However, for the sake of completeness, a few popular classification algorithms were applied for the clustered dataset with percentage split of 66% (i.e. Training set - 66% and Test set - 34%).

Logistic Regression Classifier

For the Logistic Regression Classifier, the class variable required to be of the nominal type. Therefore, before applying it to the dataset, the cluster label was converted to nominal type. The result after applying the classifier was like this.

Correctly Classified Instances:	112	51.8519%
Incorrectly Classified Instances:	104	48.1481%

Naive Bayes Classifier

Under the assumption that all features in the dataset are independent, Naive Bayes classifier was applied to the dataset and result was like this.

Correctly Classified Instances:	117	54.1667 %
Incorrectly Classified Instances:	99	45.8333 %

J48 Classifier

J48 is a type of Decision Tree and it is the implementation of algorithm ID3 (Iterative Dichotomiser 3) developed by the WEKA project team[24]. When it was applied to the dataset, following results were observed.

Correctly Classified Instances	368	50.1362 %
Incorrectly Classified Instances	366	49.8638 %

Random Forest Classifier

Random forest classifier fits a set of decision trees from randomly selected subsamples of training data set[25]. It then averages the votes from different decision trees to improve the predictive accuracy and control over-fitting. and decide the final class of the test object.

Correctly Classified Instances	419	57.0845 %
Incorrectly Classified Instances	315	42.9155 %

Multilayer Perceptron Classifier

Multilayer Perceptron (MLP) classifier is based on the feedforward artificial neural network[26]. It consists of multiple layers (i.e. an input and an output layer with one or more hidden layers) of nodes and each layer is interconnected. Nodes in the input layer take the input to the system. Then those data is processed in the hidden layers and output layer outputs the results. When this classifier was applied to the dataset, it gave the following result.

Correctly Classified Instances	384	52.3161 %
Incorrectly Classified Instances	350	47.6839 %

When comparing above results of each classifier, we can observe that Random Forest Classifier which is a type of decision tree gives more accurate results compared to the other classifiers.

Chapter 4: Evaluation and Results

The ultimate goal of this research is to identify the natural style of a given chess game. So, when a set of steps of a chess game is given, it should be able to predict which styles the 2 players have. However, it's not just a single task, but a goal with multiple objectives. In this process, evaluation of findings is crucial. And this evaluation process is also not just a 1-time task, but it's coupled with multiple phases of the project. In this project, basically, we can divide the evaluation process into 2 phases.

1. Clustering - Evaluation of clustering output
2. Classification - Evaluation of classification model

4.1 Evaluation of Clustering Output

We already discussed this in the previous chapter. We analyzed and evaluated different clustering mechanisms with different attributes of those. Cluster evaluation was done using both internal cluster validity indices and external cluster validity indices. The final result of the evaluation revealed following important facts.

- Due to the nature of chess dataset, Gaussian Mixture Models gave better clusters than other clustering methods.
- In Gaussian Mixture Models, the best set of clusters were given by "Tied" covariance.
- Well-known 5 chess playing styles (Aggressive, Positional, Defensive, Solid and Tactical) were reduced to 4 as clustering effort showed that 2 of aforementioned styles (i.e. Solid and Tactical) showed similar characteristics. Hense, those 2 styles were treated as the same afterward.

4.2 Evaluation of Classification Model

Once the 4 cluster model is finalized, a few classification algorithms were applied to the dataset. It showed that "Random Forest Classifier" gives more accuracy (i.e. 57.0845%) than the others. The next step was to evaluate this classification (or prediction) model.

The evaluation approach of the prediction model of this project had 2 ways to proceed.

1. Using games of players who are well known for certain styles
2. Using opinions of chess experts

These 2 evaluation methods have their own strengths and limitations. They are discussed below.

4.2.1 Using games of players who're well-known for certain styles

Chess grandmaster, who usually have an ELO rating of more than 2500, mostly have a known style of play. As mentioned in a previous chapter as well, each of following world class player is known for a particular playing style.

- Garry Kasparov - **Aggressive**
- Anatoly Karpov - **Positional**
- Tigran Petrosian - **Defensive**
- Peter Leko - **Solid**
- Mikhail Tal - **Tactical**

So, a set of games of each of above players were selected, just like how data was selected for analysis and predicted the style of those games. And then checked if they match with the players' well-known style. However, this evaluation method has some limitations. One main problem is that even though above players are well known for a particular style, there can be games which may showcase a different nature than their known style. Therefore, this evaluation method will not always give an accurate evaluation result. However, since each player's games are already available, evaluation can be done with less effort and in large scale, which is an advantage of this method.

For the evaluation, 50 chess games of different players were used and the outcome was like this.

Correctly Classified Instances:	29	58 %
Incorrectly Classified Instances:	21	52 %

4.2.2 Using opinions of chess experts

If the set of moves of a game is presented to a chess expert, they can usually analyze the game and tell which style(s) can be observed in the game. However, it's possible that some games are categorized as no style or has more than one style.

So, a set of games were selected randomly from different players and provided those to a set of chess experts to analyze. Here, to make sure the outputs don't depend on the personal opinions of each chess expert, a criterion was defined to select games for the evaluation. That is, one game was presented to at least 4 chess experts, and then the game was accepted to be taken into the evaluation, only if at least 3 of them put the game into a single style. This made sure the credibility of the data which is used for the evaluation of the model. High accuracy was the main advantage of this evaluation method.

However, the advantage of accuracy comes with a price, which is time, as mentioned in a previous chapter as well. The chess game data is recorded in PGN (Portable Game Notation) format. PGN contains the game steps in chess move notation (eg. e4 c5). Looking at these notations, anyone who's familiar with chess could imagine the moves in an imaginary chess board in their mind. But that had 2 problems.

1. It took time to imagine the moves as it's hard to remember all previous steps.
2. Remembering steps and analyzing styles was too much work for the mind, and it could reduce the accuracy of the outcome.

To overcome these challenges, the same Java program which was written earlier to generate online forms consisting virtual chess boards was used so that chess experts can interactively replay the given games on the screen. After they decided each game's styles, they submitted the online form. When they submitted, their analysis outcome was recorded. This reduced the time and increased the accuracy of the outcome in great margins.

For the evaluation, 10 chess games of different players were used and the outcome was like this.

Correctly Classified Instances:	4	40 %
Incorrectly Classified Instances:	6	60 %

Chapter 5: Conclusion

A raw dataset of a set of chess games does not provide any direct information about their nature. It only contains notations of sequences of moves. To extract features from a game, that notation should be understood and certain preprocessing steps are required. Chess game engines are very useful in this case. Raw game data can be fed to chess game engines and information regarding states of the game can be obtained at each move. Such data can be collected for all or selected steps of a game and then that information can be used to construct features of the game. This process was followed in this project and a set of features were extracted from thousands of games.

Then, an effort was put to categorize games into different styles with the help of chess experts. That effort wasn't much successful due to some critical challenges, but it revealed a few important facts.

- The style of a chess game can be subjective to the person who is analyzing the game. It can depend on expertise, experience and also the thinking style.
- However, when this is done in large scale, patterns can be identified which are independent of personal opinions.

Then the immediate objective was changed to clustering chess games in an unsupervised manner to identify natural clusters of games based on their styles. At this point, an assumption had to be made to obtain real style-labels for the selected dataset. That was "Games of world class players who are well known for particular playing styles always exhibit the corresponding style". This may not be true for all cases, but it was good enough for the objective of the project.

Under that assumption, different types of clustering methods were evaluated and finally, Gaussian Mixture Models (GMM) with Expectation Maximization (EM) was identified as the best way to cluster this particular set of game data. Then the output of that effort was compared with known knowledge about their styles, which revealed followings.

- Natural clusters align with the known categories more or less the same way.
- However, there can be exceptions. One example is that when chess games were clustered, both "Solid and Tactical" styles were clustered into the same cluster.

This shows that there can be cases where the difference between 2 known styles not being very considerable when it's quantified logically.

Then the new clusters were used to build a model to classify games based on their styles. To identify a better classification algorithm, results of a set of classification algorithms were compared and "Random Forest Classifier" was selected as the best classifier for the dataset.

To the final evaluation of the clusters and the prediction model, both well-known style information and input from chess experts were used. The prediction model at the end gave an accuracy of 58%. Given the novelty of the research area, which was leading to do every small bit of work from scratch, this accuracy number is satisfactory. However, there's a lot of ways to improve this number and those will be discussed in the next section as future work.

Chapter 6: Future Work

In the “feature engineering” subchapter, it was mentioned that feature extraction from the middle game can be done from different perspectives and mainly those aspects can be categorized into 3 main areas. Among those areas, only “Material balance” and “Space” was considered in this project, due to the complexity of the other area which was “Initiatives”. Extracting features related to “Initiatives” could be a separate study and will be helpful to have more important features which can be used to improve the outcome of this project in future.

Chess playing style analysis could be a massive research area. And this particular research can be considered as the initial step of that. Therefore, when selecting games for the analysis, only games related to 5 popular playing styles were selected to keep the analysis from becoming too complex in this initial stage. However, in future, other not-so-much popular playing styles such as Technical, Calculating, Practical, Intuitive, Creative, Logical etc. could be taken into consideration as well.

At the initial stage of this research project, an effort was put to manually categorize chess games with the help of chess experts. But that attempt was failed as it took a considerably huge amount of time to analyze and categorize games into different chess playing styles. Therefore, due to the time limitations, the assumption which was explained in the middle chapters had to be made. However, that assumption does not provide a 100% accuracy. Therefore, as a future task of this, we can spend time (it could be months or years) to get chess experts involved in the manual categorization of games. This can be done via some organized event and in fun and interesting methods.

Due to the difficulty of preparing a ground truth dataset for chess game categories, as mentioned in previous chapters, the main focus of this project was data clustering but not classification. So, once a set of ground truth data is collected for categories by above method, more focus can be put on classification.

Another important thing which was in the initial plan of this project was to study how players and games have evolved over time. Once the above classification objective is achieved with a good accuracy, the classifier can be applied to one player’s different games which span over a long time period, like a few years or decades. The expectation is to

recognize how the playing style of a single player evolves with age. Then this can also be generalized to all games to identify whether game styles have been evolved over time.

The plan was to cover this if the time permitted. However, as feature engineering took more time than expected, there wasn't enough time to cover this. Therefore this can be done as a future work of the project.

References

[1] Bargh, J. and Morsella, E. (2008). The Unconscious Mind. Perspectives on Psychological Science, [online] 3(1), pp.73-79. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2440575/ [Accessed 2 Jul. 2017].

[2] Chessgames.com. (2017). Chess openings: Giuoco Piano (C50). [online] Available at: http://www.chessgames.com/perl/chessopening?eco=C50 [Accessed 20 Jul. 2017].

[3] Chessgames.com. (2017). Chess openings: Queen's Gambit Declined (D37). [online] Available at: http://www.chessgames.com/perl/chessopening?eco=d37 [Accessed 20 Jul. 2017].

[4] Atkin, R. and Witten, I. (1975). A multi-dimensional approach to positional chess. International Journal of Man-Machine Studies, [online] 7(6), pp.727-750. Available at: http://www.sciencedirect.com/science/article/pii/S0020737375800356 [Accessed 16 Jun. 2017].

[6] Barnes, D. and Hernandez-Castro, J. (2015). On the limits of engine analysis for cheating detection in chess. Computers & Security, [online] 48, pp.58-73. Available at: http://www.sciencedirect.com/science/article/pii/S0167404814001485 [Accessed 16 May 2017].

[7] Game-ai-forum.org. (2017). World Microcomputer Chess Championship (ICGA Tournaments). [online] Available at: https://www.game-ai-forum.org/icga-tournaments/competition.php?id=2 [Accessed 24 Jul. 2017].

[8] Fifteenth International Conference on Advances in Computer Games 2017. (2017). World Computer Chess Championship. [online] Available at: https://acg2017.wordpress.com/2017/06/19/world-computer-chess-championship/ [Accessed 13 Jun. 2017].

[9] Thoresen, M. (2017). TCEC - Top Chess Engine Championship. [online] Tcec.chessdom.com. Available at: http://tcec.chessdom.com/ [Accessed 17 Jul. 2017].

[10] Dailey, D., Hair, A. and Watkins, M. (2014). Move similarity analysis in chess programs. Entertainment Computing, [online] 5(3), pp.159-171. Available at: http://www.sciencedirect.com/science/article/pii/S1875952113000177 [Accessed 5 Jul. 2017].

[11] Wbec-ridderkerk.nl. (2017). UCI protocol. [online] Available at: http://wbec-ridderkerk.nl/html/UCIProtocol.html [Accessed 15 Jul. 2017].

[12] Hartston, W. (1985). *Chess (Teach Yourself)*. 4th ed. Pp.102-103.

[13] "ChessBase 9's material balance display", Chess News, 2018. [Online]. Available: https://en.chessbase.com/post/chebase-9-s-material-balance-display. [Accessed: 25- Mar- 2018].

[14] "Positional Chess Understanding: The Space Advantage", Thechessworld.com, 2018. [Online]. Available: https://thechessworld.com/articles/middle-game/positional-chess-understanding-the-space-advantage/. [Accessed: 25- Feb- 2018].

[15] J. Silman, "What Is The Initiative? - Chess.com", Chess.com, 2018. [Online]. Available: https://www.chess.com/article/view/what-is-the-initiative. [Accessed: 25- Feb- 2018].

[16] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016. Available at: https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf [Accessed 2 Aug. 2017]

[17] Teuvo Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, 1995.

[18] "Self-Organizing Map (SOM)", Users.ics.aalto.fi, 2018. [Online]. Available: http://users.ics.aalto.fi/jhollmen/dippa/node9.html. [Accessed: 19- Mar- 2018].

[19] T. Caliński & J Harabasz (2007) A dendrite method for cluster analysis, Communications in Statistics, 3:1, 1-27, DOI: 10.1080/03610927408827101

[20] Rosenberg, Andrew & Hirschberg, Julia. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure.. 410-420.

[21] "The Adjusted Rand index - Dave Tang's blog", Dave Tang's blog, 2018. [Online]. Available: https://davetang.org/muse/2017/09/21/adjusted-rand-index/. [Accessed: 13- Feb- 2018].

[22] Fowlkes, E. B., Mallows, C. L.: A Method for Comparing Two Hierarchical Clusterings. Journal of the American Statistical Association, 78(383):553–569, 1983.

[23] Silke Wagner, Dorothea Wagner: Comparing Clusterings - An Overview, January 12, 2007.

[24] "J48 decision tree - Mining at UOC", Data-mining.business-intelligence.uoc.edu, 2018. [Online]. Available: http://data-mining.business-intelligence.uoc.edu/home/j48-decision-tree. [Accessed: 20- Mar- 2018].

[25] "3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.19.1 documentation", Scikit-learn.org, 2018. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed: 19- Mar- 2018].

[26] "Classification and regression - Spark 2.3.0 Documentation", Spark.apache.org, 2018. [Online]. Available: https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier. [Accessed: 20- Mar- 2018].