Homework 1

Anisha Mula, Ilana Novakoski, Mohammmed Saad Ahmad, Jiaru Cai, Mainak Manoj Kumar Roy, Dimo Liao

28-09-2019

Contents

1 Overview

1.1 Task

How can we increase AS Roma's chances of securing a top 4 placement in the league? We aim to assist AS Roma to consistently place higher in the league after implementing our recommendations.

1.2 Background

AS Roma can leverage hidden trends in football match data to gain an edge in tight matches. A slight advantage can make the difference between a win or a loss for the team, and can ultimately be the difference between whether or not AS Roma participates in the championship. These advantages become especially pertinent when AS Roma is on the cusp of finishing in the top four and securing a slot in the championships. In order to launch itself over this key barrier, AS Roma must deploy the best mix of players, emphasize training for skills associated with wins, and execute the optimal formation against the opposing team's starting formation.

In a typical season, AS Roman finishes close to 4th place in the league. Their average ranking is 3.81 with a standard deviation of 2.5. This means that the team's average performance falls precisely on the border of championship qualification, which in turn determines whether the team can enjoy the commensurate financial rewards of playing the european championships. Therefore, every slight advantage could determine if AS Roma will continue on to the championships. With this goal in mind, we focused our analysis on how AS Roma can adjust its performance in order to move up in the rankings.

2 Exploratory Analysis

2.1 Data import

Load the required libraries

```
if(require('arules')){library(arules)} else{install.packages('arules')}
if(require('lubridate')){library(lubridate)} else{install.packages('lubridate')}
if(require('RSQLite')){library(RSQLite)} else{install.packages('RSQLite')}
if(require('purrr')){library(arules)} else{install.packages('purrr')}
if(require('tidyverse')){library(arules)} else{install.packages('tidyverse')}
if(require('factoextra')){library(RSQLite)} else{install.packages('factoextra')}
if(require('nFactors')){library(psych)} else{install.packages('nFactors')}
if(require('dplyr')){library(arules)} else{install.packages('dplyr')}
if(require('ggplot2')){library(ggplot2)} else{install.packages('ggplot2')}
if(require('ggthemes')){library(ggplot2)} else{install.packages('ggthemes')}
if(require('plyr')){library(plyr)} else{install.packages('plyr')}
```

Now we have to load the data as dataframe to start processing. It is currently in SQLite.

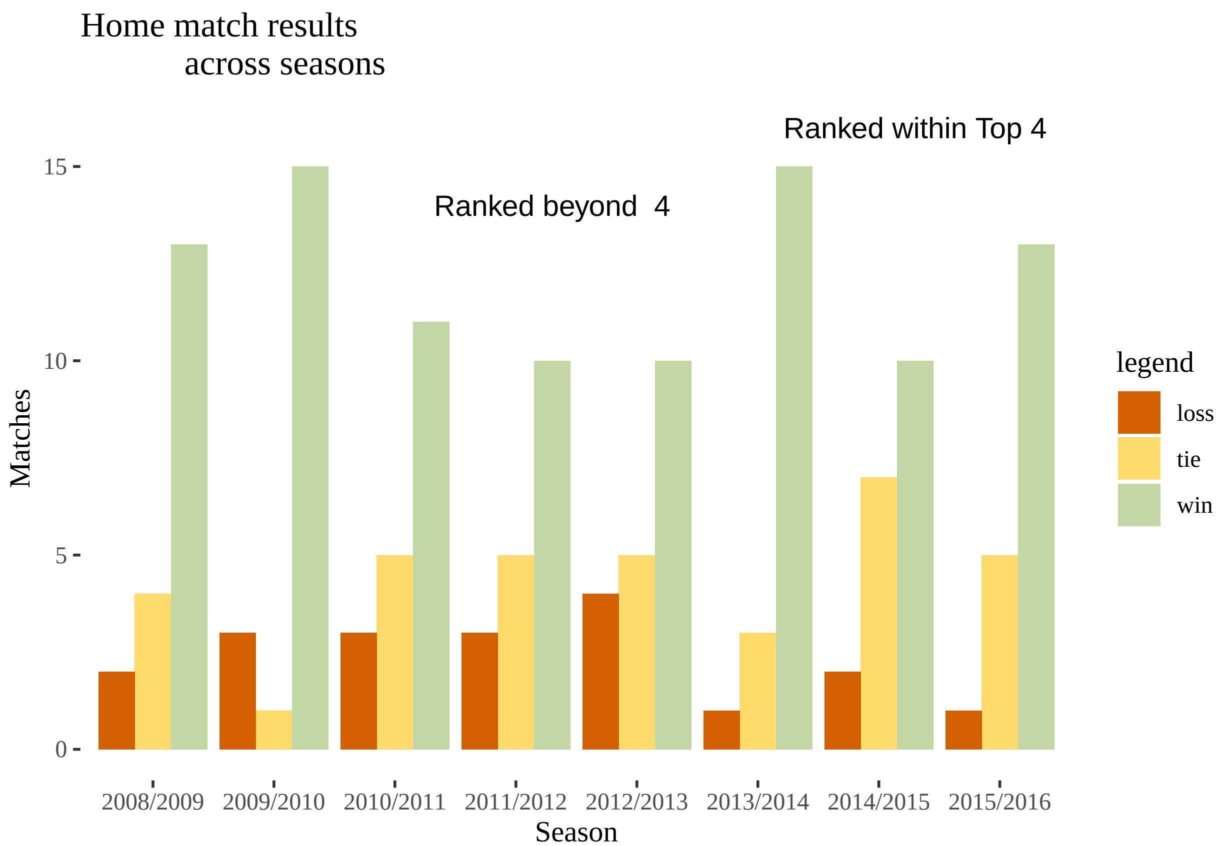
```
sqlite <- dbDriver("SQLite")
euro_soccer_db <- dbConnect(sqlite,"euro_soccer.sqlite")

country <- data.frame(tbl(euro_soccer_db, "Country"))
league <- data.frame(tbl(euro_soccer_db, "League"))
match <- data.frame(tbl(euro_soccer_db, "Match"))
player <- data.frame(tbl(euro_soccer_db, "Player"))
```

```
player_attributes <- data.frame(tbl(euro_soccer_db, "Player_Attributes"))
team <- data.frame(tbl(euro_soccer_db, "Team"))
team_attributes <- data.frame(tbl(euro_soccer_db, "Team_Attributes"))
roma_record <- team %>% filter(team_long_name == 'Roma')
```

2.2 Filtering data

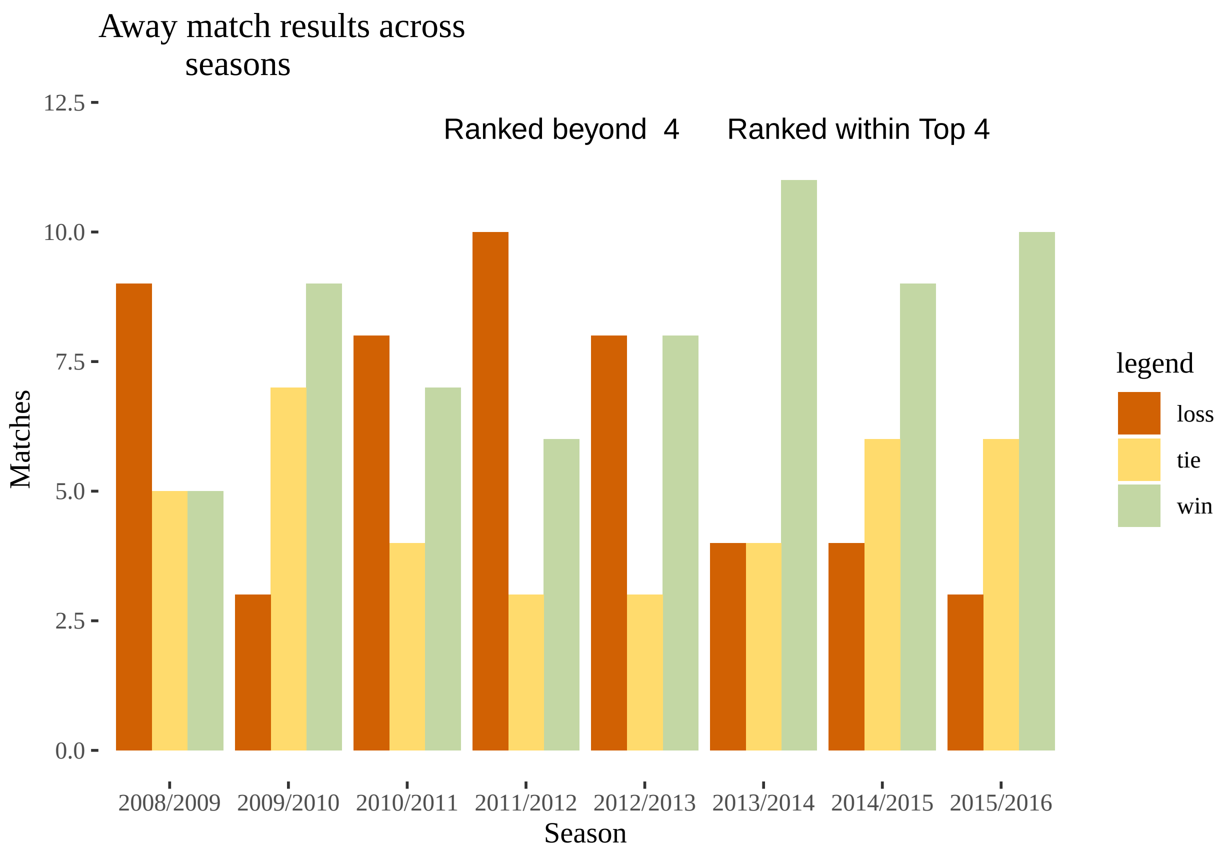
We are primarily interested in exploiting the match and player attributes for the Serie A league. We had filtered the match level data to contain the matches played as a part of Serie A league and performed exploratory analysis.

```
h<- ggplot(roma_matches_home, aes(x=season))

plot_h<- h + geom_bar(position="dodge",aes(fill = result)) + labs(x="Season",
            y="Matches", fill="Match Result", title="Home match results
            across seasons") + scale_fill_manual("legend", values = c("win" = "#C3D7A4",
            "tie" =  "#FFDB6D", "loss" = "#D16103")) + annotate("text", x = 7.3, y = 16,
            label = "Ranked within Top 4") + annotate("text", x = 4.3, y = 14,
                                                    label = "Ranked beyond  4")

plot_h + theme_tufte()
```



AS Roma placed in the top 4 teams for all three of the last three years, but did not place in the top 4 during at all prior to the last three years. The team has enjoyed a higher number of wins and ties in the most recent years, and a corresponding drop in losses. It appears that both AS Roma's wins and losses have converted to ties, but that AS Roma's losses have converted to ties at a higher rate than the team's wins. Knowing that even a slight bump in performance can significantly impact Romas's chances to make it and remain safely in the top 4, we would examine the games that Roma ties or loses at home. However, the team's performance patterns change during away games, as demonstrated by the figure below.

```
a <- ggplot(roma_matches_away, aes(x=season))
plot_a <- a + geom_bar(position="dodge",aes(fill = result)) + labs(x="Season",
          y="Matches", fill="Match Result", title="Away match results across
          seasons") +  scale_fill_manual("legend", values = c("win" = "#C3D7A4",
          "tie" = "#FFDB6D", "loss" = "#D16103")) + annotate("text", x = 6.8, y = 12,
          label = "Ranked within Top 4") + annotate("text", x = 4.3, y = 12,
                                                    label = "Ranked beyond  4")

plot_a+ theme_tufte()
```

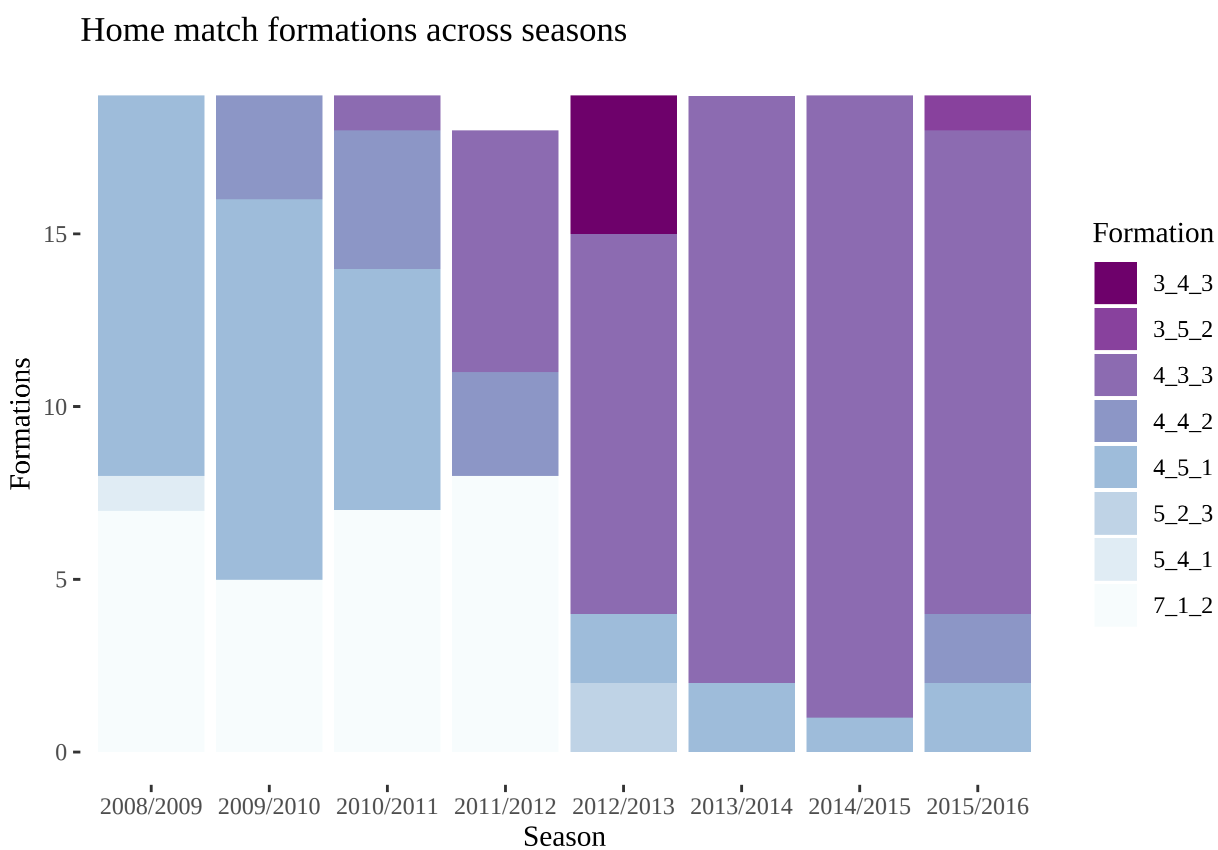


In the following sections we examine two strategies for improving AS Roma's performance: formation selection and player positioning.

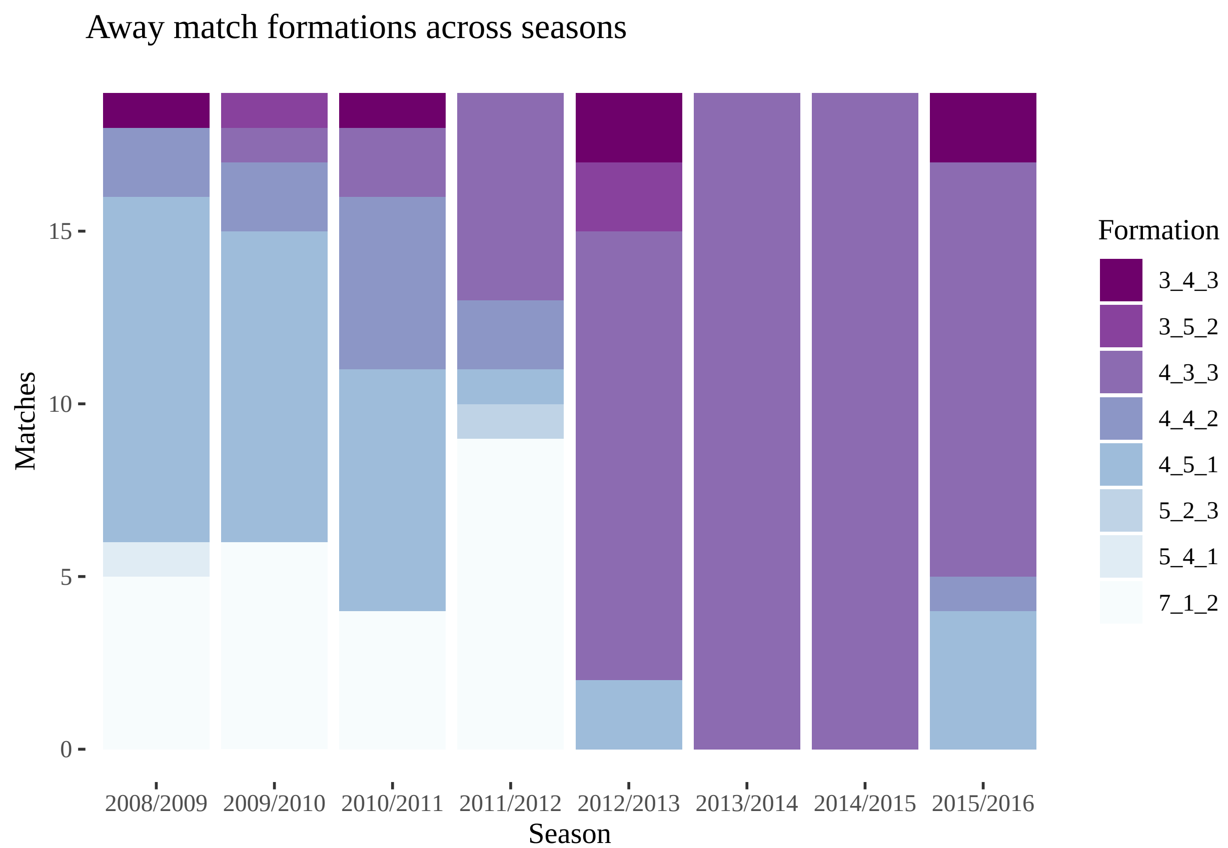
2.3 Starting Formation

In addition to exploring the effects of home versus away games, we analyzed AS Roma's starting formation strategy. We observed that AS Roma had utilized a variety of starting formations during the early years of the available performance history, but settled on primarily using a 4-3-3 starting formation in the past 3 three years. This trend can be observed in the figures below.

```r
h <- ggplot(roma_matches_home, aes(x=season))
plot_h <- h + geom_bar(aes(fill = formation_home)) + labs(x="Season", y="Formations",
          fill = "Formation", title="Home match formations across seasons")

plot_h + scale_fill_brewer(palette="BuPu",direction=-1) + theme_tufte()
```




```r
a<- ggplot(roma_matches_away, aes(x=season))
plot_a <- a + geom_bar(aes(fill = formation_away)) + labs(x="Season", y="Matches",
          fill = "Formation", title="Away match formations across seasons")
plot_a + scale_fill_brewer(palette="BuPu",direction=-1)+theme_tufte()
```



This change in AS Roma's starting formation coincided with AS Roma's improvement in overall rankings; AS Roma earned a top 4 position during the the 2014 - 2016 seasons, the same season in which they used the 4-3-3 formation consistently. This trend held true for both home and away games. While we cannot claim causation, we note that the use of this starting formation is correlated with AS Roma placing in the top 4.

Therefore, our analysis indicates that the 4-3-3 formation plays a key role in AS Roma achieving a top 4 spot. One thing to note is that AS Roma began utilizing the 4-3-3 formation in the 2012/2013 season but still finished 6th. They won 44% of the games in that year, but the win percentage bumped to 67% next season with that formation. In football it normally takes time for a new strategy to be implemented and for the success to follow. Huge strategy changes are difficult to implement as players need to adjust to their new positions and also coordination with other players takes time to establish. Knowing this, and coupled with the fact that 4-3-3 has been working for Roma we aim to suggest recommendations that will boost their performance keeping the 4-3-3 formation intact. We believe, within that formation subtle changes to strategies should be relatively easy to implement, as compared to strategies that will require major changes to team dynamics. The caveat is that there may be formations and strategies that certain opponent teams may be susceptible too, but our focus on 4-3-3 may not allow us to exploit that.

However, we have noticed that in order to reach our goal of being in the top 4 of league, we may have to win only a few more games than we already do. Studying the league rankings from 2008 to 2016, we found that:

- In seasons in which Roma did not rank in the top 4, it landed 6th or 7th places.
- The average number of points between the 4th and 5th position team was 1.72. This difference is equivalent to winning one more game than your rival in 5th position. The difference in points between a 4th and 6th position team is 6 points, which is equivalent to winning 2 more games.

Therefore every game that AS Roma plays has the potential to affect its top 4 status. We treat every home game and away as equal in our analysis.

If we leverage the 4-3-3 combination to win 1-2 more games, we could effectively achieve our goal of consistently being in the top 4. This begs the question, how can we use the 4-3-3 formation more effectively given our players and their attributes to increase our chances of being in the top 4.

We tried to determine what levers a coach has in his/her disposal to make sure Roma is set-up for success. The coach predominantly has two levers to determine the following:

- What training needs to be given to impart certain skills that the coach requires, and more importantly what skills are required to compete against a particular opponent.
- How to distribute these skills on the pitch with a 4-3-3 formation.

The above decisions inform the decision as to which players need to play a particular match that can carry forward the skills required to win to the pitch. This forms the basic model of our recommendations and framework of analysis.

3 Analysis

3.1 Skills and Field Segmentation

3.1.1 Condensing player attributes to reduce analysis complexity

The player attribute table consists of the 38 features a mixture of continuous and categorical variables. We wanted to examine how the collective attributes of players impact the outcome of a match. We wanted to focus our analysis on a smaller subset of columns, which are representative of the player attributes. We also observed significant correlation between the variables and so we employed factor analysis to reduce the dimension of the data. We condensed the 35 numerical attributes to 3 factors. These factors capture 77% of the variation within these variables.

```
col1 <- which(colnames(match)=='home_player_X1')
col2 <- which(colnames(match)=='away_player_11')

match.player_positions <- filter.player(match, col1:col2)
match_select <- match %>% dplyr::select(contains('home'),contains('away'))

player_attributes <- player_attributes[complete.cases(player_attributes),]
player_attributes.factor <- player_attributes %>%
                            dplyr::select(-c('id','player_api_id','player_fifa_api_id','date',
                            'preferred_foot','attacking_work_rate','defensive_work_rate'))


fit <- factanal(player_attributes.factor, 5, rotation="varimax",scores="Bartlett")
print(fit$loadings, digits=2, cutoff=.3, sort=TRUE)
```

```
##
## Loadings:
##                      Factor1 Factor2 Factor3 Factor4 Factor5
## crossing              0.73                    0.36
## finishing             0.77           -0.33
## short_passing         0.76    0.32    0.37
## volleys               0.80
## dribbling             0.79    0.34            0.40
## curve                 0.82
```

```
## free_kick_accuracy  0.80
## long_passing        0.66            0.52
## ball_control        0.79    0.40
## shot_power          0.74    0.34
## long_shots          0.85
## positioning         0.77
## vision              0.82
## penalties           0.74
## heading_accuracy            0.59    0.31            0.49
## gk_diving          -0.39   -0.79
## gk_handling        -0.37   -0.85
## gk_kicking                 -0.72
## gk_positioning     -0.37   -0.85
## gk_reflexes        -0.38   -0.85
## aggression                          0.61            0.38
## interceptions                       0.85
## marking                             0.94
## standing_tackle                     0.95
## sliding_tackle                      0.95
## acceleration        0.36                    0.85
## sprint_speed        0.32                    0.80
## agility             0.52                    0.67
## balance             0.42                    0.56
## overall_rating      0.51                            0.61
## reactions           0.51                            0.51
## strength                                   -0.36    0.63
## potential           0.40                            0.50
## jumping                                             0.43
## stamina             0.37            0.39    0.36
##
##                 Factor1 Factor2 Factor3 Factor4 Factor5
## SS loadings       10.63    5.21    5.20    3.36    2.31
## Proportion Var     0.30    0.15    0.15    0.10    0.07
## Cumulative Var     0.30    0.45    0.60    0.70    0.76
```

The computations and post-processing code for dimensionality reduction can be found in Section 2 and Section 3 of the Appendix

We further analysed the factor loadings to understand the relationships captured within these variables. Below is a list of the three attribute factors and their associated skills.

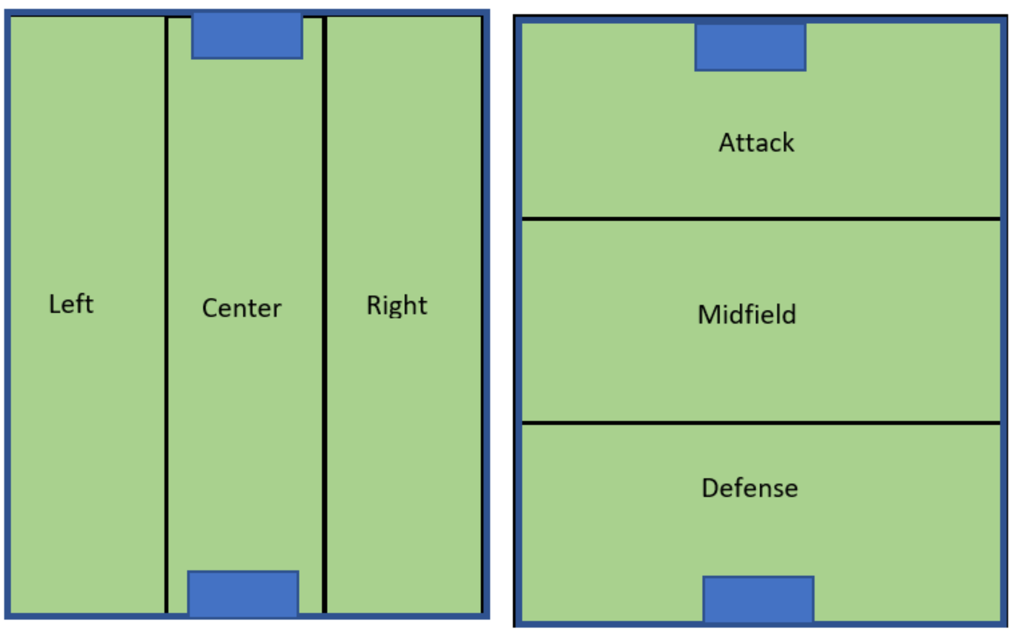
- **Factor 1: Fine ball skills and athleticism**
  - Forward Skills: finishing, dribbling, volleys curve, penalties, shot power
  - Forward and Midfielder Skills: short passing, long passing, crossing, vision, positioning
  - General Skills: acceleration, sprint speed, agility, balance and reactions
  - Summary: players with high Fine ball skills and athleticism should generally be fast, with high ball control and ability to shoot and pass accurately.
- **Factor 2: Gross Ball Skills**
  - Midfielder Skills: short passing, dribbling, ball control ,shot power and heading accuracy
- **Factor 3: Aggressiveness**
  - Interceptions, marking, standing tackle and sliding tackle.

– These characteristics generally determine the capability of the player to dispossess the ball from the opponent and aggressiveness with which they play.

Once we condensed the player attributes, we developed a framework that combined these attributes with specific locations of the football field. The field was then sliced into 6 regions, described below

- Vertical Slices:
  – Left : Includes players with an X position attribute between 1 and 3.
  – Middle : Includes players with an X position attribute between 4 and 6.
  – Right : Includes players with an X position attribute between 7 and 9.
- Horizontal Slices:
  – Defense: Includes players with an Y position attribute between 2 and 5.
  – Midfield: Includes players with an Y position attribute between 6 and 8.
  – Attack: Includes players with an Y position attribute between 9 and 11.

3.1.2 Visualization of the Field Segmentation



We assume that players in each subsubsection can move to a certain extent to other parts of the field, so we did not analyze finer subsections like Left-Defense or Right Attack.

For each match, we summed the player attribute scores described above to create the overall skill level for each slice of the field. (Data transformation from Player -> Team -> Match). We used the average skill level (across all players) to fill in the missing data. We used this framework to analyze how combined skill and position dynamics lead to a particular match outcome.

This numeric data was converted to categorical data and fed into association rules. . We analyzed the distribution of these skill values at a match level for the entire Serie A league, and generated the following rules to categorize them:

1. High : If the values fall above or equal to 70th percentile.
2. Medium : If the values range between the 30th percentile and 70th percentile.

3. Low : If the values range below the 30th percentile.

We assume higher skills in a certain section of the pitch leads to overall dominance in play.

Once we had the skill levels for each slice, we filtered the data to select for matches where a 4-3-3 formation was used, and the match outcome (Win, Loss, or Tie) was labelled as per the team who won using the 4-3-3 formation. This allows us to see what skills and position combinations in a 4-3-3 formation lead to a particular match outcome that apply across all teams. We used this filtered dataset to run our association rules

The code for this computation can be found in Sections 4,5 and 6 of the Appendix. This computation is time consuming (approximately 15-20 minutes) and so we have stored the output in a csv file to avoid repeat calculations.

```
match.subset <- match %>% filter(league_id == 10257)

match.subset <- generate_home_factors(match.subset)
match.subset <- generate_away_factors(match.subset)
```

3.1.2.1 Analysis for home matches

3.1.2.1.1 Vertical partition analysis

Generating Associations Looking at the Left, Right and Middle Slices of the field.

Theme: Italian League teams which secure home wins with a 4-3-3 Formation

The code for precessing the match level data to transactions in attached in Section 7 of Appendix.

```
final_rulesD = apriori(tDFD1,parameter = list(support = 0.035, minlen = 2,target = "rules"),
                       appearance = list(rhs ="DETAILED_OUTCOME=HOME_WIN"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.035      2
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 34
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[59 item(s), 980 transaction(s)] done [0.00s].
## sorting and recoding items ... [59 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10

## Warning in apriori(tDFD1, parameter = list(support = 0.035, minlen =
## 2, : Mining stopped (maxlen reached). Only patterns up to a length of 10
## returned!
```

```
##  done [0.38s].
## writing ... [16 rule(s)] done [0.02s].
## creating S4 object  ... done [0.01s].
```

```r
inspect(final_rulesD)
```

```
##      lhs                                   rhs                            support confidence     lift c
## [1]  {XM_factor2=XM_factor2_Low,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor2A=XM_factor2A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8297872 2.893920
## [2]  {XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor2A=XM_factor2A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.04285714  0.8076923 2.816863
## [3]  {XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor3A=XM_factor3A_Medium,
##       XR_factor2A=XR_factor2A_Low}    => {DETAILED_OUTCOME=HOME_WIN} 0.03571429  0.8333333 2.906287
## [4]  {XM_factor2=XM_factor2_Low,
##       XL_factor1A=XL_factor1A_Low,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor2A=XM_factor2A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03673469  0.8181818 2.853445
## [5]  {XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XL_factor3A=XL_factor3A_Low,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03571429  0.8139535 2.838699
## [6]  {XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor1A=XM_factor1A_Medium,
##       XM_factor2A=XM_factor2A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03673469  0.8000000 2.790036
## [7]  {XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor1A=XM_factor1A_Medium,
##       XM_factor2A=XM_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03877551  0.8260870 2.881015
## [8]  {XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor1A=XM_factor1A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03877551  0.8085106 2.819717
## [9]  {XL_factor1=XL_factor1_High,
##       XM_factor2=XM_factor2_Low,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor2A=XM_factor2A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03775510  0.8409091 2.932708
## [10] {XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor2A=XM_factor2A_Medium,
```

```
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8863636 3.091233
## [11] {XL_factor1=XL_factor1_High,
##       XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor2A=XM_factor2A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.04081633  0.8000000 2.790036
## [12] {XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor1A=XL_factor1A_Low,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor1A=XM_factor1A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03673469  0.8000000 2.790036
## [13] {XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor1A=XL_factor1A_Low,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor2A=XM_factor2A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03673469  0.8780488 3.062234
## [14] {XL_factor1=XL_factor1_High,
##       XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor1A=XM_factor1A_Medium,
##       XM_factor2A=XM_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03673469  0.8181818 2.853445
## [15] {XL_factor1=XL_factor1_High,
##       XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor1A=XM_factor1A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03571429  0.8139535 2.838699
## [16] {XL_factor1=XL_factor1_High,
##       XM_factor2=XM_factor2_Low,
##       XR_factor1=XR_factor1_High,
##       XL_factor2A=XL_factor2A_Low,
##       XM_factor2A=XM_factor2A_Medium,
##       XM_factor3A=XM_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03775510  0.8809524 3.072361
```

Results from Association Rule: A total of 16 rules were generated when the antecedent was restricted to "Home Win". Running the association rules this way tells us what skill-location combination is attributed to a 4-3-3 formation winning at home. The support count of these rules ranges from 35-42, a confidence range of 0.79 to 0.88 and a lift range of 2.79 to 3.09.

Interpretation of the Association Rule results: *Having high factor 1(fine ball skills and athleticism) in the right and left sections of the field is correlated with a win, for teams playing a 4-3-3 formation* The win occurs even when the winning team has low/medium skills levels in the middle slice of the field This suggests that the losing teams are weak on the sides of the pitch and that teams playing against such opponents can utilize this weakness. Factor 1 (fine ball skills and athleticism) can be associated with an attacking ability hence Roma should play with a strong attacking mindset and try and create as many chances as possible.

Opponent teams against whom this strategy should be exploited: Once we uncovered the skill-location combinations, we analyzed which teams were involved in such matches. We compiled the names of teams that either tied with Roma or beat Roma in the last two years and that lost to a 4-3-3 at least three times against other teams. We got the team names by filtering the match data that included the home and away team attribute combination which the association rules correlated with the home team winning using a 4-3-3. The teams are: * Atlanta * Cagliari * Empoli * Palermo

We recommend that Roma employs the strategy of starting with a 4-3-3 formation with high factor 1 (fine ball skills and athleticism) players placed on the sides of the fields against the above teams. This could help Roma win the 1 to 2 more matches a season, which may be necessary to secure a position in top 4.

Translating the strategy to AS Roma player combinations: The player combinations that result in high factor 1(Fine ball skills and athleticism) as compared to other Serie A teams in the Left side of the pitch are (Players present as on 2016 with AS Roma):

The code for generating the important combinations of players is in Section 8

- Alessandro Florenzi,Miralem Pjanic,Gervinho
- Alessandro Florenzi,Miralem Pjanic,Iago Falque
- Alessandro Florenzi,Miralem Pjanic,Mohamed Salah
- Alessandro Florenzi,Radja Nainggolan,Mohamed Salah
- Antonio Ruediger,Miralem Pjanic,Mohamed Salah
- Maicon,Alessandro Florenzi,Gervinho
- Maicon,Alessandro Florenzi,Iago Falque
- Maicon,Miralem Pjanic,Alessandro Florenzi
- Maicon,Miralem Pjanic,Juan Manuel Iturbe

The natural corollary to this recommendation is that the coach can rest tired players from the midsection of the field as they seem to matter less in such games. The coach can try new players and give them valuable game experience.

3.1.2.1.2 Horizontal partition analysis

Generating Associations Looking at the Defense, Midfield and Attack.

Theme: Italian League teams which secure home wins with a 4-3-3 Formation

```
final_rulesD = apriori(tDFD,parameter = list(support = 0.035, minlen = 2,target = "rules"),
                       appearance = list(rhs ="DETAILED_OUTCOME=HOME_WIN"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.035      2
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 34
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[59 item(s), 980 transaction(s)] done [0.00s].
## sorting and recoding items ... [59 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10
```

```
## Warning in apriori(tDFD, parameter = list(support = 0.035, minlen = 2,
## target = "rules"), : Mining stopped (maxlen reached). Only patterns up to a
## length of 10 returned!
```

```
##  done [0.61s].
## writing ... [11 rule(s)] done [0.04s].
## creating S4 object  ... done [0.02s].
```

```r
inspect(final_rulesD)
```

```
##      lhs                                    rhs                              support confidence     lift c
## [1]  {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YA_factor2A=YA_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8125000 2.833630
## [2]  {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YM_factor2=YM_factor2_Medium,
##       YA_factor2A=YA_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8297872 2.893920
## [3]  {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YA_factor1=YA_factor1_High,
##       YA_factor2A=YA_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8297872 2.893920
## [4]  {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YA_factor2=YA_factor2_High,
##       YA_factor2A=YA_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8297872 2.893920
## [5]  {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YM_factor2=YM_factor2_Medium,
##       YA_factor1=YA_factor1_High,
##       YA_factor2A=YA_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8297872 2.893920
## [6]  {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YM_factor2=YM_factor2_Medium,
##       YA_factor2=YA_factor2_High,
##       YA_factor2A=YA_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8297872 2.893920
## [7]  {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YA_factor1=YA_factor1_High,
##       YA_factor2=YA_factor2_High,
##       YA_factor2A=YA_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8297872 2.893920
## [8]  {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YM_factor3=YM_factor3_Medium,
##       YA_factor1=YA_factor1_High,
##       YD_factor3A=YD_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03673469  0.8181818 2.853445
## [9]  {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YM_factor3=YM_factor3_Medium,
##       YA_factor2=YA_factor2_High,
##       YD_factor3A=YD_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03673469  0.8181818 2.853445
## [10] {YD_factor3=YD_factor3_High,
##       YM_factor1=YM_factor1_Medium,
##       YM_factor2=YM_factor2_Medium,
##       YA_factor1=YA_factor1_High,
##       YA_factor2=YA_factor2_High,
##       YA_factor2A=YA_factor2A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03979592  0.8297872 2.893920
## [11] {YD_factor3=YD_factor3_High,
```

```
##       YM_factor1=YM_factor1_Medium,
##       YM_factor3=YM_factor3_Medium,
##       YA_factor1=YA_factor1_High,
##       YA_factor2=YA_factor2_High,
##       YD_factor3A=YD_factor3A_Medium} => {DETAILED_OUTCOME=HOME_WIN} 0.03673469  0.8181818 2.853445
```

Results from Association Rule: Winning strategy for opponents playing with a medium factor 2 (gross ball skills) attack:

A total of 11 rules were generated when the antecedent was restricted to "Home Win". Running the association rules this way tells us what field position-player skill combination is attributed to a 4-3-3 formation winning at home. The support count of these rules range from 36-39, a confidence range of 0.81 to 0.82 and a lift range of 2.83 to 2.89.

Translating the association rule results provides the following field position-player skill: Summary: AS Roma stands the highest possibility to win if we have a strong defense and attack. Nuances are mentioned below:

- The home team should play a high factor 3 (aggressiveness) at Defence, a moderate/medium factor 1,2 in the middle slice, paired with a high factor 1,2 in attack against opponents with a medium attack (factor 2,3)
- The win occurs even when the winning (home) team plays with medium skills levels in the middle slice of the field
- Opponents having a medium attack factor 2,3 translates to a moderately aggressive stance and ball control. It can be countered and overcome by the home team by employing a strong defence which is also aggressive (high factor 3). The association rule's pairing of an aggressive defence with a strong attack ("high factor 1,2 in attack") correlates to a "Counter Attack" style of play.

Opponent teams against whom this strategy should be exploited: Once we got the skill-location combinations we analyzed which teams were involved in such matches. We recommend to use this against teams, that Roma has either drawn or lost at home in the last two years and those teams against which this trend has been observed at least 3 times. The teams against which this strategy can be applied are:

- Torino
- Udinese
- Juventus

Translating the strategy to AS Roma player combinations: The player combinations that result in high factor 3 (aggressiveness) as compared to other Serie A teams in defense are (Players present as on 2016 with AS Roma):

The code for generating the important combinations of players is in Section 8

- Alessandro Florenzi,Konstantinos Manolas,Antonio Ruediger,Lucas Digne.
- Vasilios Torosidis,Konstantinos Manolas,Leandro Castan,Lucas Digne.
- Maicon,Konstantinos Manolas,Antonio Ruediger,Lucas Digne.
- Vasilios Torosidis,Konstantinos Manolas,Antonio Ruediger,Lucas Digne.
- Antonio Ruediger,Konstantinos Manolas,Ervin Zukanovic,Maicon.
- Alessandro Florenzi,Konstantinos Manolas,Lucas Digne,Daniele De Rossi.
- Maicon,Daniele De Rossi,Konstantinos Manolas,Lucas Digne.
- Alessandro Florenzi,Antonio Ruediger,Konstantinos Manolas,Lucas Digne.
- Vasilios Torosidis,Konstantinos Manolas,Daniele De Rossi,Lucas Digne.
- Alessandro Florenzi,Konstantinos Manolas,Ervin Zukanovic,Lucas Digne.
- Maicon,Antonio Ruediger,Ervin Zukanovic,Lucas Digne.

The player combinations that result in high factor 1 (fine ball skills and athleticism) and high factor 2 (gross ball skills) as compared to other Serie A teams

- Mohamed Salah,Diego Perotti,Stephan El Shaarawy.
- Radja Nainggolan,Diego Perotti,Mohamed Salah.
- Mohamed Salah,Edin Dzeko,Gervinho.
- Iago Falque,Gervinho,Mohamed Salah.

This list will help the coach make substitutions and gives more choices in the event that certain defenders are injured. Note that player Mohamed Salah is a very important component of the attack and all combinations of players require him. It's advisable that the club do their best to retain him in the squad.

3.1.2.2 Analysis for away matches

Generating Associations Looking at the Left, Right and Middle Slices of the field.

Theme: Italian League teams which secure away wins with a 4-3-3 Formation

```
final_rules_awayD = apriori(tDFD,parameter = list(support = 0.005, minlen = 2,maxlen=4,target =  "rules
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.005      2
##  maxlen target   ext
##       4  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 4
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[59 item(s), 980 transaction(s)] done [0.00s].
## sorting and recoding items ... [59 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4
```

```
## Warning in apriori(tDFD, parameter = list(support = 0.005, minlen = 2,
## maxlen = 4, : Mining stopped (maxlen reached). Only patterns up to a length
## of 4 returned!
```

```
##  done [0.03s].
## writing ... [24 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(final_rules_awayD)
```

```
##      lhs                                    rhs                              support confidence     lift
## [1]  {YD_factor3=YD_factor3_Low,
```

```
##       YA_factor1=YA_factor1_Low,
##       YA_factor2=YA_factor2_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.010204082  0.8333333 4.537037
## [2]  {YD_factor3=YD_factor3_Low,
##       YA_factor2=YA_factor2_Low,
##       YA_factor3=YA_factor3_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.009183673  0.8181818 4.454545
## [3]  {YD_factor3=YD_factor3_Low,
##       YA_factor2=YA_factor2_Low,
##       YM_factor2A=YM_factor2A_Medium} => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [4]  {YD_factor2=YD_factor2_Low,
##       YA_factor2=YA_factor2_Low,
##       YM_factor2A=YM_factor2A_Medium} => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [5]  {YD_factor1=YD_factor1_Low,
##       YA_factor1=YA_factor1_Low,
##       YA_factor2=YA_factor2_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.011224490  0.8461538 4.606838
## [6]  {YD_factor1=YD_factor1_Low,
##       YA_factor2=YA_factor2_Low,
##       YA_factor3=YA_factor3_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.010204082  0.8333333 4.537037
## [7]  {YD_factor1=YD_factor1_Low,
##       YM_factor1=YM_factor1_Medium,
##       YA_factor2=YA_factor2_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  1.0000000 5.444444
## [8]  {YD_factor1=YD_factor1_Low,
##       YA_factor2=YA_factor2_Low,
##       YM_factor2A=YM_factor2A_Medium} => {DETAILED_OUTCOME=AWAY_WIN} 0.008163265  0.8888889 4.839506
## [9]  {YM_factor1=YM_factor1_Medium,
##       YA_factor2=YA_factor2_Low,
##       YM_factor2A=YM_factor2A_Medium} => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  1.0000000 5.444444
## [10] {YD_factor3=YD_factor3_Low,
##       YM_factor1=YM_factor1_Medium,
##       YA_factor3=YA_factor3_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [11] {YD_factor2=YD_factor2_Low,
##       YM_factor1=YM_factor1_Medium,
##       YA_factor3=YA_factor3_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [12] {YD_factor2=YD_factor2_Low,
##       YA_factor3=YA_factor3_High,
##       YD_factor3A=YD_factor3A_High}   => {DETAILED_OUTCOME=AWAY_WIN} 0.006122449  0.8571429 4.666667
## [13] {YD_factor1=YD_factor1_Low,
##       YM_factor3=YM_factor3_Medium,
##       YA_factor1=YA_factor1_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [14] {YD_factor1=YD_factor1_Low,
##       YM_factor2=YM_factor2_Medium,
##       YA_factor1=YA_factor1_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [15] {YD_factor1=YD_factor1_Low,
##       YD_factor2=YD_factor2_Medium,
##       YA_factor1=YA_factor1_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [16] {YD_factor1=YD_factor1_Low,
##       YD_factor3=YD_factor3_Medium,
##       YA_factor1=YA_factor1_Low}      => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [17] {YA_factor1=YA_factor1_Low,
##       YA_factor3=YA_factor3_Medium,
##       YD_factor3A=YD_factor3A_High}   => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  1.0000000 5.444444
## [18] {YA_factor1=YA_factor1_Low,
##       YA_factor3=YA_factor3_Medium,
##       YD_factor2A=YD_factor2A_High}   => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [19] {YA_factor1=YA_factor1_Low,
```

```
##       YA_factor3=YA_factor3_Medium,
##       YM_factor2A=YM_factor2A_Medium} => {DETAILED_OUTCOME=AWAY_WIN} 0.006122449  1.0000000 5.444444
## [20] {YM_factor1=YM_factor1_Medium,
##       YA_factor1=YA_factor1_Low,
##       YD_factor3A=YD_factor3A_High}   => {DETAILED_OUTCOME=AWAY_WIN} 0.009183673  0.8181818 4.454545
## [21] {YM_factor1=YM_factor1_Medium,
##       YA_factor1=YA_factor1_Low,
##       YM_factor1A=YM_factor1A_Medium} => {DETAILED_OUTCOME=AWAY_WIN} 0.006122449  0.8571429 4.666667
## [22] {YD_factor1=YD_factor1_Low,
##       YM_factor3=YM_factor3_Medium,
##       YD_factor2A=YD_factor2A_High}   => {DETAILED_OUTCOME=AWAY_WIN} 0.006122449  0.8571429 4.666667
## [23] {YD_factor3=YD_factor3_Medium,
##       YM_factor2=YM_factor2_High,
##       YD_factor1A=YD_factor1A_High}   => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
## [24] {YM_factor2=YM_factor2_High,
##       YD_factor3A=YD_factor3A_High,
##       YM_factor2A=YM_factor2A_Low}    => {DETAILED_OUTCOME=AWAY_WIN} 0.005102041  0.8333333 4.537037
```

Results from Association Rule Winning strategy for opponents playing with a medium factor 2 (gross ball skills) attack:

A total of 12 rules were generated when the antecedent was restricted to "Away Win". Running the association rules this way tells us what field position-player skill combination is attributed to a 4-3-3 formation winning at away games. The support count of these rules range from 6-13, a confidence range of 0.80 to 0.85 and a lift range of 4.2 to 4.6.

Translating the association rule results provides the following field position-player skill

Summary: Factor 2 (gross ball skills) seems to play a decisive role in gauging the result, especially for the home team's left and right positions. The details below, highlight additional nuances: * The away team should play a medium (factor 1,2) in the middle section and a high right (factor 2 - gross ball skills), against opponents who have a weak left (factor 2 - gross ball skills), in order to maximize chances of a win * The away team should play a medium (factor 1,2) in the middle section and a high left (factor 2), against opponents who have a weak right (factor 2 - gross ball skills), in order to maximize chances of a win *In general, the right and left sides should be strong in order to maximize chances of a win

Opponent teams against whom this strategy should be exploited * Palermo * Parma * Atalanta * Sampdoria

Refer to the above player lists for the players with the correct skill combinations.

For Away games - Mid,Attack and Defence analysis, the association rules did not generate any significant results.

4 Conclusion

The skill-position framework provides a method to quantitatively analyze matches using different configurations on the football pitch. The analysis highlighted certain features for home and away games that AS Roma can exploit to their advantage:

- HOME Games:
  - When playing at home, against opponents Atlanta, Cagliari, Empoli and Palermo(trends show that we have lost or tied games with them in the last few years), we can utilize the sides of the pitch to increase our chances of winning by playing a fast-paced (high factor 1,2) and attack-based game and use players with high fine ball control and athletic abilities (high factor 2 - gross ball skills)

- Roma should play an aggressive style defense (factor 3 - aggressiveness) against Torino, Udinese and Juventus. This would allow them to soak the aggressive pressure of the opponent's forwards and midfielders. They should also employ fast forwards (high factor 2,3) to counter attack when there is a chance

- AWAY Games:
  - For maximizing chances of a win, AS Roma should play with a strong factor 2 (gross ball skills) skills on left and right sections (based on whether or not the opponent is weak on right or left section respectively).Teams against which this can be applied are Palermo, Parma, Atalanta and Sampdoria.

In both the scenarios, AS Roma should continue to use the 4-3-3 formation. The team can then adjust its player assignments based on the above insights. By keeping the formation intact and changing focus to certain parts of the field we aim to make the recommendations relatively easy to implement and fit into the current team dynamics.

As established in earlier sections, the score difference between the 4th and 6th ranked teams in a given season is quite less. The opponents who have been referenced - especially Atlanta, Udinese, Palermo and Torino are all relatively close rivals that challenge AS ROMA for the top 4 spot (As analyzed by studying the ranking of teams in each individual season). By applying these strategies against these teams, we aim to maximize AS ROMA's chances of winning against these teams and thus giving them an edge on the league table. This strategy excludes putting Juventus and Napoli in focus, which have more resources and skills at their disposal, and hence are difficult to beat with any strategy in general. Instead, we can achieve our objective of securing and safe guarding a top 4 spot by shifting our focus to other, but strategically important, teams that AS Roma can beat with minor strategy adjustments.

Sources Cited
https://www.kaggle.com/hugomathien/soccer/discussion/30200#176414
https://en.wikipedia.org/wiki/List_of_A.S._Roma_seasons
https://en.wikipedia.org/wiki/2013%E2%80%9314_Serie_A
https://en.wikipedia.org/wiki/2014%E2% 80%9315_Serie_A
https://en.wikipedia.org/wiki/2015%E2%80%9316_Serie_A

5 Appendix

This section contains the code which was necessary for transformations and other pre-processing. They were run in the background to generate the visualizations and analysis, but were not deemed relevant for display in the main report.

5.0.1 Section 1: Preprocessing for exploratory data analysis

```
# section 1
romaTeamAttributes <- team_attributes %>%
                      filter(team_api_id==8686) %>%
                      mutate(date = as.Date(date)) %>%
                      mutate(year = year(date))

roma_record <- team %>% filter(team_long_name == 'Roma')
home_matches <- match %>%
                filter(home_team_api_id == roma_record$team_api_id) %>%
                mutate(result = case_when(home_team_goal > away_team_goal ~ "win",
```

```
                                          home_team_goal == away_team_goal ~ "tie",
                                          home_team_goal < away_team_goal ~ "loss"))


roma_matches <- match %>%
                filter(home_team_api_id == roma_record$team_api_id |
                       away_team_api_id == roma_record$team_api_id)

roma_matches$formation_home=1;
roma_matches$formation_away=1;

for(i in 1:nrow(roma_matches))
{
  row_home <- roma_matches[i, names(roma_matches) %in% c("home_player_Y1","home_player_Y2","home_player_
  row_away <- roma_matches[i, names(roma_matches) %in% c("away_player_Y1","away_player_Y2","away_player_

  defender_count <- 0
  midfielder_count <- 0
  attacker_count <- 0

  row_formation_home<-""
  # HOME
  for (val in row_home) {
    if (val >= 2 & val <= 5) {
      defender_count = 1 + defender_count
    } else if (val >= 6 & val <= 8) {
      midfielder_count = 1 + midfielder_count
    } else if (val >= 9 & val <= 11) {
      attacker_count = attacker_count + 1
    }
  }
  row_formation_home<- paste(c(defender_count, midfielder_count, attacker_count), collapse = "_" )
  roma_matches$formation_home[i]=row_formation_home
  defender_count <- 0
  midfielder_count <- 0
  attacker_count <- 0
  row_formation_away<-""

  # AWAY
  for (val in row_away) {
    if (val >= 2 & val <= 5) {
      defender_count = 1 + defender_count
    } else if (val >= 6 & val <= 8) {
      midfielder_count = 1 + midfielder_count
    } else if (val >= 9 & val <= 11) {
      attacker_count = attacker_count + 1
    }
    row_formation_away<- paste(c(defender_count, midfielder_count, attacker_count), collapse = "_" )
    roma_matches$formation_away[i]=row_formation_away
  }

}
```

```
roma_matches_home <- roma_matches %>% filter(home_team_api_id == roma_record$team_api_id)
roma_matches_away <-roma_matches %>% filter(away_team_api_id == roma_record$team_api_id)

roma_matches_home <- roma_matches_home %>%
                     mutate(result = case_when(home_team_goal > away_team_goal ~ "win",
                                               home_team_goal == away_team_goal ~ "tie",
                                               home_team_goal < away_team_goal ~ "loss"))

roma_matches_away <- roma_matches_away %>%
                     mutate(result = case_when( away_team_goal > home_team_goal~ "win",
                                                away_team_goal == home_team_goal  ~ "tie",
                                                away_team_goal < home_team_goal~ "loss"))

roma_matches_home$count=1
roma_matches_away$count=1
```

5.0.2 Section 2: Factor analysis

```
# Function 1
filter.player <- function(data,desiredCols) {

  # remove player's position if it's NA
  # The player postions information will be contained in desiredCols

  filtered.NA <- complete.cases(data[,desiredCols])
  return(data[filtered.NA, ])
}

# Function 5
calculate_mode <- function(x) {
  uniqx <- unique(x)
  uniqx[which.max(tabulate(match(x, uniqx)))]
}
```

5.0.3 Section 3: Post processing the factor analysis results

```
load <- as.data.frame(as.matrix(player_attributes.factor) %*% as.matrix(fit$loadings))
player_attributes.factor <- cbind(player_attributes[c('id','player_api_id','date',
                                                      'preferred_foot','attacking_work_rate',
                                                      'defensive_work_rate')], load)

player_attributes.factor <- player_attributes.factor %>%
                            mutate(date = as.Date(date)) %>%
                            mutate(year = year(date), month = month(date))

player_attributes.factor$season <- ifelse(player_attributes.factor$month >= 6,
                                   paste(player_attributes.factor$year, player_attributes.factor$year+
                                   paste(player_attributes.factor$year-1, player_attributes.factor$yea
```

```r
player_attr <- player_attributes.factor %>%
  dplyr::select(-c('year','month','id','date')) %>%
  group_by(player_api_id,season) %>%
  summarise(preferred_foot = calculate_mode(preferred_foot),
            attacking_work_rate = calculate_mode(attacking_work_rate),
            defensive_work_rate = calculate_mode(defensive_work_rate),
            Factor1 = mean(Factor1),
            Factor2 = mean(Factor2),
            Factor3 = mean(Factor3),
            Factor4 = mean(Factor4),
            Factor5 = mean(Factor5))
```

5.0.4 Section 4: Retrieving formations

```r
# Function 2
get_formation <- function (row, desiredCols) {
  defender_count <- 0
  midfielder_count <- 0
  attacker_count <- 0
  for (val in row[,desiredCols]) {
    if (val >= 2 & val <= 5) {
      defender_count = 1 + defender_count
    } else if (val >= 6 & val <= 8) {
      midfielder_count = 1 + midfielder_count
    } else if (val >= 9 & val <= 11) {
      attacker_count = attacker_count + 1
    }
  }

  formation<- paste(c(defender_count, midfielder_count, attacker_count), collapse = "_" )
  return(formation)
}

# Function 3
get_x_position<- function (val) {
  x_position <- ""
  if (val >= 1 & val <= 3) {
    x_position <- "L"
  } else if (val >= 4 & val <= 6) {
    x_position <- "M"
  } else if (val >= 7 & val <= 9) {
    x_position <- "R"
  }
  return(x_position)
}

# Function 4
get_y_position<- function (val) {
  y_position <- ""
  if (val >= 2 & val <= 5) {
    y_position <- "D"
```

```
  } else if (val >= 6 & val <= 8) {
    y_position <- "M"
  } else if (val >= 9 & val <= 11) {
    y_position <- "A"
  }
  return(y_position)
}
```

5.0.5 Section 5: Combining the factor level attributes of each player to team level for every match

```
# section 5

# Function 2
get_formation <- function (row, desiredCols) {
  defender_count <- 0
  midfielder_count <- 0
  attacker_count <- 0
  for (val in row[,desiredCols]) {
    if (val >= 2 & val <= 5) {
      defender_count = 1 + defender_count
    } else if (val >= 6 & val <= 8) {
      midfielder_count = 1 + midfielder_count
    } else if (val >= 9 & val <= 11) {
      attacker_count = attacker_count + 1
    }
  }

  formation<- paste(c(defender_count, midfielder_count, attacker_count), collapse = "_" )
  return(formation)
}

# Function 3
get_x_position<- function (val) {
  x_position <- ""
  if (val >= 1 & val <= 3) {
    x_position <- "L"
  } else if (val >= 4 & val <= 6) {
    x_position <- "M"
  } else if (val >= 7 & val <= 9) {
    x_position <- "R"
  }
  return(x_position)
}

# Function 4
get_y_position<- function (val) {
  y_position <- ""
  if (val >= 2 & val <= 5) {
    y_position <- "D"
  } else if (val >= 6 & val <= 8) {
    y_position <- "M"
```

```r
  } else if (val >= 9 & val <= 11) {
    y_position <- "A"
  }
  return(y_position)
}



# function 6
generate_home_factors <- function(match){
  for(i in 1:nrow(match)) {
    row <- match[i,]
    position_XL_factor1 <- 0
    position_XM_factor1 <- 0
    position_XR_factor1 <- 0
    position_YA_factor1 <- 0
    position_YM_factor1 <- 0
    position_YD_factor1 <- 0
    position_XL_factor2 <- 0
    position_XM_factor2 <- 0
    position_XR_factor2 <- 0
    position_YA_factor2 <- 0
    position_YM_factor2 <- 0
    position_YD_factor2 <- 0
    position_XL_factor3 <- 0
    position_XM_factor3 <- 0
    position_XR_factor3 <- 0
    position_YA_factor3 <- 0
    position_YM_factor3 <- 0
    position_YD_factor3 <- 0
    position_XL_factor4 <- 0
    position_XM_factor4 <- 0
    position_XR_factor4 <- 0
    position_YA_factor4 <- 0
    position_YM_factor4 <- 0
    position_YD_factor4 <- 0
    position_XL_factor5 <- 0
    position_XM_factor5 <- 0
    position_XR_factor5 <- 0
    position_YA_factor5 <- 0
    position_YM_factor5 <- 0
    position_YD_factor5 <- 0


    # ------------------- iterate player attribute --------
    counter = start.h.id
    for(home_player_id_col in row[start.h.id:end.h.id]) {

      player <- player_attr %>% filter(player_api_id==home_player_id_col , season == row[,4])
      if(nrow(player)>0){
        fac1 = as.integer(player[,6][1])
        fac2 = as.integer(player[,7][1])
        fac3 = as.integer(player[,8][1])
```

```r
    fac4 = as.integer(player[,9][1])
    fac5 = as.integer(player[,10][1])
  }
  if (nrow(player)==0){
    player <- player_attr_mean %>%  filter(player_api_id==home_player_id_col)
    if(nrow(player)>0){

      fac1 = as.integer(player[,5][1])
      fac2 = as.integer(player[,6][1])
      fac3 = as.integer(player[,7][1])
      fac4 = as.integer(player[,8][1])
      fac5 = as.integer(player[,9][1])
    }

    if(nrow(player)==0){
      fac1 = as.integer(attributeMean[,1])
      fac2 = as.integer(attributeMean[,2])
      fac3 = as.integer(attributeMean[,3])
      fac4 = as.integer(attributeMean[,4])
      fac5 = as.integer(attributeMean[,5])
    }
  }

    x_position_col <- counter - end.h.y
    if (get_x_position(row[,x_position_col]) == "L") {
      position_XL_factor1 = position_XL_factor1 + fac1
      position_XL_factor2 = position_XL_factor2 + fac2
      position_XL_factor3 = position_XL_factor3 + fac3
      position_XL_factor4 = position_XL_factor4 + fac4
      position_XL_factor5 = position_XL_factor5 + fac5
    } else if (get_x_position(row[,x_position_col]) == "M") {
      position_XM_factor1 = position_XM_factor1 + fac1
      position_XM_factor2 = position_XM_factor2 + fac2
      position_XM_factor3 = position_XM_factor3 + fac3
      position_XM_factor4 = position_XM_factor4 + fac4
      position_XM_factor5 = position_XM_factor5 + fac5
    } else if (get_x_position(row[,x_position_col]) == "R") {
      position_XR_factor1 = position_XR_factor1 + fac1
      position_XR_factor2 = position_XR_factor2 + fac2
      position_XR_factor3 = position_XR_factor3 + fac3
      position_XR_factor4 = position_XR_factor4 + fac4
      position_XR_factor5 = position_XR_factor5 + fac5
    }

  y_position_col <- counter - end.h.x

  if (get_y_position(row[,y_position_col]) == "D") {
    position_YD_factor1 = position_YD_factor1 + fac1
    position_YD_factor2 = position_YD_factor2 + fac2
    position_YD_factor3 = position_YD_factor3 + fac3
    position_YD_factor4 = position_YD_factor4 + fac4
    position_YD_factor5 = position_YD_factor5 + fac5
  } else if (get_y_position(row[,y_position_col]) == "M") {
```

```
    position_YM_factor1 = position_YM_factor1 + fac1
    position_YM_factor2 = position_YM_factor2 + fac2
    position_YM_factor3 = position_YM_factor3 + fac3
    position_YM_factor4 = position_YM_factor4 + fac4
    position_YM_factor5 = position_YM_factor5 + fac5
  } else if (get_y_position(row[,y_position_col]) == "A") {
    position_YA_factor1 = position_YA_factor1 + fac1
    position_YA_factor2 = position_YA_factor2 + fac2
    position_YA_factor3 = position_YA_factor3 + fac3
    position_YA_factor4 = position_YA_factor4 + fac4
    position_YA_factor5 = position_YA_factor5 + fac5
  }

  counter = counter + 1
}


match[i,]$XL_factor1 <- position_XL_factor1
match[i,]$XL_factor2 <- position_XL_factor2
match[i,]$XL_factor3 = position_XL_factor3
match[i,]$XL_factor4 = position_XL_factor4
match[i,]$XL_factor5 = position_XL_factor5

match[i,]$XM_factor1 = position_XM_factor1
match[i,]$XM_factor2 = position_XM_factor2
match[i,]$XM_factor3 = position_XM_factor3
match[i,]$XM_factor4 = position_XM_factor4
match[i,]$XM_factor5 = position_XM_factor5

match[i,]$XR_factor1 = position_XR_factor1
match[i,]$XR_factor2 = position_XR_factor2
match[i,]$XR_factor3 = position_XR_factor3
match[i,]$XR_factor4 = position_XR_factor4
match[i,]$XR_factor5 = position_XR_factor5

match[i,]$YD_factor1 = position_YD_factor1
match[i,]$YD_factor2 = position_YD_factor2
match[i,]$YD_factor3 = position_YD_factor3
match[i,]$YD_factor4 = position_YD_factor4
match[i,]$YD_factor5 = position_YD_factor5

match[i,]$YM_factor1 = position_YM_factor1
match[i,]$YM_factor2 = position_YM_factor2
match[i,]$YM_factor3 = position_YM_factor3
match[i,]$YM_factor4 = position_YM_factor4
match[i,]$YM_factor5 = position_YM_factor5

match[i,]$YA_factor1 = position_YA_factor1
match[i,]$YA_factor2 = position_YA_factor2
match[i,]$YA_factor3 = position_YA_factor3
match[i,]$YA_factor4 = position_YA_factor4
match[i,]$YA_factor5 = position_YA_factor5
```

```r
  }
  return(match)
}

# function 7
generate_away_factors <- function(match){
  XMODE = TRUE
  for(i in 1:nrow(match)) {
    row <- match[i,]
    position_XL_factor1 <- 0
    position_XM_factor1 <- 0
    position_XR_factor1 <- 0
    position_YA_factor1 <- 0
    position_YM_factor1 <- 0
    position_YD_factor1 <- 0
    position_XL_factor2 <- 0
    position_XM_factor2 <- 0
    position_XR_factor2 <- 0
    position_YA_factor2 <- 0
    position_YM_factor2 <- 0
    position_YD_factor2 <- 0
    position_XL_factor3 <- 0
    position_XM_factor3 <- 0
    position_XR_factor3 <- 0
    position_YA_factor3 <- 0
    position_YM_factor3 <- 0
    position_YD_factor3 <- 0
    position_XL_factor4 <- 0
    position_XM_factor4 <- 0
    position_XR_factor4 <- 0
    position_YA_factor4 <- 0
    position_YM_factor4 <- 0
    position_YD_factor4 <- 0
    position_XL_factor5 <- 0
    position_XM_factor5 <- 0
    position_XR_factor5 <- 0
    position_YA_factor5 <- 0
    position_YM_factor5 <- 0
    position_YD_factor5 <- 0


    # ------------------- iterate player attribute --------
    counter = 68
    for(home_player_id_col in row[68:77]) {

      player <- player_attr %>% filter(player_api_id==home_player_id_col , season == row[,4])
      if(nrow(player)>0){
        fac1 = as.integer(player[,6][1])
        fac2 = as.integer(player[,7][1])
        fac3 = as.integer(player[,8][1])
        fac4 = as.integer(player[,9][1])
        fac5 = as.integer(player[,10][1])
      }
```

```
    if (nrow(player)==0){
      player <- player_attr_mean %>%  filter(player_api_id==home_player_id_col)
      if(nrow(player)>0){

        fac1 = as.integer(player[,5][1])
        fac2 = as.integer(player[,6][1])
        fac3 = as.integer(player[,7][1])
        fac4 = as.integer(player[,8][1])
        fac5 = as.integer(player[,9][1])
      }

      if(nrow(player)==0){
        fac1 = as.integer(attributeMean[,1])
        fac2 = as.integer(attributeMean[,2])
        fac3 = as.integer(attributeMean[,3])
        fac4 = as.integer(attributeMean[,4])
        fac5 = as.integer(attributeMean[,5])
      }
    }

    if(XMODE){
      x_position_col <- counter - 44
      if (get_x_position(row[,x_position_col]) == "L") {

        position_XL_factor1 = position_XL_factor1 + fac1
        position_XL_factor2 = position_XL_factor2 + fac2
        position_XL_factor3 = position_XL_factor3 + fac3
        position_XL_factor4 = position_XL_factor4 + fac4
        position_XL_factor5 = position_XL_factor5 + fac5
      } else if (get_x_position(row[,x_position_col]) == "M") {
        position_XM_factor1 = position_XM_factor1 + fac1
        position_XM_factor2 = position_XM_factor2 + fac2
        position_XM_factor3 = position_XM_factor3 + fac3
        position_XM_factor4 = position_XM_factor4 + fac4
        position_XM_factor5 = position_XM_factor5 + fac5
      } else if (get_x_position(row[,x_position_col]) == "R") {
        position_XR_factor1 = position_XR_factor1 + fac1
        position_XR_factor2 = position_XR_factor2 + fac2
        position_XR_factor3 = position_XR_factor3 + fac3
        position_XR_factor4 = position_XR_factor4 + fac4
        position_XR_factor5 = position_XR_factor5 + fac5
      }
    }
    y_position_col <- counter - 22

    if (get_y_position(row[,y_position_col]) == "D") {
      position_YD_factor1 = position_YD_factor1 + fac1
      position_YD_factor2 = position_YD_factor2 + fac2
      position_YD_factor3 = position_YD_factor3 + fac3
      position_YD_factor4 = position_YD_factor4 + fac4
      position_YD_factor5 = position_YD_factor5 + fac5
    } else if (get_y_position(row[,y_position_col]) == "M") {
      position_YM_factor1 = position_YM_factor1 + fac1
```

```
      position_YM_factor2 = position_YM_factor2 + fac2
      position_YM_factor3 = position_YM_factor3 + fac3
      position_YM_factor4 = position_YM_factor4 + fac4
      position_YM_factor5 = position_YM_factor5 + fac5
    } else if (get_y_position(row[,y_position_col]) == "A") {
      position_YA_factor1 = position_YA_factor1 + fac1
      position_YA_factor2 = position_YA_factor2 + fac2
      position_YA_factor3 = position_YA_factor3 + fac3
      position_YA_factor4 = position_YA_factor4 + fac4
      position_YA_factor5 = position_YA_factor5 + fac5
    }

    counter = counter + 1
  }

  match[i,]$XL_factor1A <- position_XL_factor1
  match[i,]$XL_factor2A <- position_XL_factor2
  match[i,]$XL_factor3A = position_XL_factor3
  match[i,]$XL_factor4A = position_XL_factor4
  match[i,]$XL_factor5A = position_XL_factor5

  match[i,]$XM_factor1A = position_XM_factor1
  match[i,]$XM_factor2A = position_XM_factor2
  match[i,]$XM_factor3A = position_XM_factor3
  match[i,]$XM_factor4A = position_XM_factor4
  match[i,]$XM_factor5A = position_XM_factor5

  match[i,]$XR_factor1A = position_XR_factor1
  match[i,]$XR_factor2A = position_XR_factor2
  match[i,]$XR_factor3A = position_XR_factor3
  match[i,]$XR_factor4A = position_XR_factor4
  match[i,]$XR_factor5A = position_XR_factor5

  match[i,]$YD_factor1A = position_YD_factor1
  match[i,]$YD_factor2A = position_YD_factor2
  match[i,]$YD_factor3A = position_YD_factor3
  match[i,]$YD_factor4A = position_YD_factor4
  match[i,]$YD_factor5A = position_YD_factor5

  match[i,]$YM_factor1A = position_YM_factor1
  match[i,]$YM_factor2A = position_YM_factor2
  match[i,]$YM_factor3A = position_YM_factor3
  match[i,]$YM_factor4A = position_YM_factor4
  match[i,]$YM_factor5A = position_YM_factor5

  match[i,]$YA_factor1A = position_YA_factor1
  match[i,]$YA_factor2A = position_YA_factor2
  match[i,]$YA_factor3A = position_YA_factor3
  match[i,]$YA_factor4A = position_YA_factor4
  match[i,]$YA_factor5A = position_YA_factor5

}
return(match)
```

```
}
```

5.0.6 Section 6: Categorizing the factor values into "High","Medium" and "Low"

```
# function 8
bucket <-function(y){
  q <- quantile(y,c(0.3,0.7))

  val_b = array()
  for(i in 1:length(y)){
    state = "Medium"
    if(y[i] <= q[1]){
      state = "Low"
    }

    if(y[i] >= q[2]){
      state = "High"
    }
    val_b[i] = state
  }
  return(val_b)
}


# function 9
n_append <-function(y,name){
  y1 = array()
  for(i in 1:length(y)){
    y1[i] = paste(name, y[i], sep = '_')
  }
  return(y1)
}

col.start <- which(colnames(match.subset)=='XL_factor1')
col.end <- which(colnames(match.subset)=='YA_factor5A')

temp <- data.frame(apply(match.subset[,col.start:col.end],2,bucket))
colnames(temp) <- colnames(match.subset[,col.start:col.end])

final = data.frame(temp[1])
for(i in 1:ncol(temp)){
  name = colnames(temp)[i]
  y1 = n_append(temp[,i],name)
  final = cbind(final,as.data.frame(y1))
}

final = final[2:61]

colnames(final) <- colnames(temp)
match.subset = cbind(match.subset[1:col.start - 1], final)
```

5.0.7 Section 7: Tranforming the match level attributes to transactions

```
# section 7

full_file <- read.csv("hw1_master_data.csv")

# function 13 - calculate the number of defender, attacker and midfielder

for(i in 1:nrow(full_file)) {
  row <- full_file[i,]
  home_player_Y <- 30:39
  away_player_Y <- 41:50
  home_formation<- paste(c(get_formation(row, home_player_Y)), collapse = "_" )
  full_file$home_formation[i]=home_formation
  away_formation<- paste(c(get_formation(row, away_player_Y)), collapse = "_" )
  full_file$away_formation[i]=away_formation
}

full_file <- full_file %>% filter(home_formation == '4_3_3' | away_formation == '4_3_3')


# detailed results
full_file_WIN_TRY <- ifelse((full_file$home_team_goal> full_file$away_team_goal) & (full_file$home_forma
                            ifelse((full_file$away_team_goal > full_file$home_team_goal) & (full_file$aw
                                   ifelse((full_file$home_team_goal == full_file$away_team_goal),"TIE",
                                          ifelse((full_file$home_team_goal> full_file$away_team_goal) &
                                                 ,"HOME_LOSS"))))



full_file$DETAILED_OUTCOME <- full_file_WIN_TRY

# combine the results above
full_file_filtered_DETAILED <- full_file %>% select('YD_factor1','YD_factor2','YD_factor3','YM_factor1'

full_file_filtered_DETAILED_LEFT_RIGHT <- full_file %>% select('XL_factor1','XL_factor2','XL_factor3','

# association rules for MID-DEFENSER-ATTACKER
tDFD <- as(full_file_filtered_DETAILED, "transactions")
```

5.0.8 Section 8

```
full_file <- read.csv("hw1_master_data.csv")

roma_id <- 'Roma'
roma_match <- full_file %>% filter((home_team_api_id == roma_id) | (away_team_api_id == roma_id))

  for (i in 1:nrow(roma_match)) {
    row <- roma_match[i,]
    home_player_id_count = 51
    away_player_id_count = 62
    XL_h_players <- ""
```

```
XM_h_players <- ""
XR_h_players <- ""
YD_h_players <- ""
YM_h_players <- ""
YA_h_players <- ""
XL_a_players <- ""
XM_a_players <- ""
XR_a_players <- ""
YD_a_players <- ""
YM_a_players <- ""
YA_a_players <- ""

for(home_player_x_pos in row[7:16]) {
  p <- player %>% filter(player_api_id==row[,home_player_id_count])

  if (get_x_position(home_player_x_pos) == "L") {
    XL_h_players = paste(c(XL_h_players,p$player_name), collapse = ',' )
  } else if (get_x_position(home_player_x_pos) == "M") {
    XM_h_players = paste(c(XM_h_players, p$player_name), collapse = ',' )
  } else if (get_x_position(home_player_x_pos) == "R") {
    XR_h_players = paste(c(XR_h_players,p$player_name), collapse = ',' )
  }
  home_player_id_count = home_player_id_count + 1
}
home_player_id_count = 51

for(home_player_y_pos in row[29:38]) {
  p <- player %>% filter(player_api_id==row[,home_player_id_count])
  if (get_y_position(home_player_y_pos) == "D") {
    YD_h_players = paste(c(YD_h_players,p$player_name), collapse = ',' )
  } else if (get_y_position(home_player_y_pos) == "M") {
    YM_h_players = paste(c(YM_h_players,p$player_name), collapse = ',' )
  } else if (get_y_position(home_player_y_pos) == "A") {
    YA_h_players = paste(c(YA_h_players,p$player_name), collapse = ',' )
  }
  home_player_id_count = home_player_id_count + 1
}

for(away_player_x_pos in row[18:27]) {
  p <- player %>% filter(player_api_id==row[,away_player_id_count])

  if (get_x_position(away_player_x_pos) == "L") {
    XL_a_players = paste(c(XL_a_players,p$player_name), collapse = ',' )
  } else if (get_x_position(away_player_x_pos) == "M") {
    XM_a_players = paste(c(XM_a_players, p$player_name), collapse = ',' )
  } else if (get_x_position(away_player_x_pos) == "R") {
    XR_a_players = paste(c(XR_a_players,p$player_name), collapse = ',' )
  }
  away_player_id_count = away_player_id_count + 1
}

away_player_id_count = 62
for(away_player_y_pos in row[40:49]) {
```

```
      p <- player %>% filter(player_api_id==row[,away_player_id_count])

      if (get_y_position(away_player_y_pos) == "D") {
        YD_a_players = paste(c(YD_a_players,p$player_name), collapse = ',' )
      } else if (get_y_position(away_player_y_pos) == "M") {
        YM_a_players = paste(c(YM_a_players,p$player_name), collapse = ',' )
      } else if (get_y_position(away_player_y_pos) == "A") {
        YA_a_players = paste(c(YA_a_players,p$player_name), collapse = ',' )
      }
      away_player_id_count = away_player_id_count + 1
    }

    roma_match$XL_h_players[i] = substring(XL_h_players, 2)
    roma_match$XM_h_players[i] = substring(XM_h_players, 2)
    roma_match$XR_h_players[i] = substring(XR_h_players, 2)
    roma_match$YD_h_players[i] = substring(YD_h_players, 2)
    roma_match$YM_h_players[i] = substring(YM_h_players, 2)
    roma_match$YA_h_players[i] = substring(YA_h_players, 2)
    roma_match$XL_a_players[i] = substring(XL_a_players, 2)
    roma_match$XM_a_players[i] = substring(XM_a_players, 2)
    roma_match$XR_a_players[i] = substring(XR_a_players, 2)
    roma_match$YD_a_players[i] = substring(YD_a_players, 2)
    roma_match$YM_a_players[i] = substring(YM_a_players, 2)
    roma_match$YA_a_players[i] = substring(YA_a_players, 2)
    XL_h_players <- ""
    XM_h_players <- ""
    XR_h_players <- ""
    YD_h_players <- ""
    YM_h_players <- ""
    YA_h_players <- ""
    XL_a_players <- ""
    XM_a_players <- ""
    XR_a_players <- ""
    YD_a_players <- ""
    YM_a_players <- ""
    YA_a_players <- ""
  }

roma_home_match <- roma_match %>% filter(home_team_api_id == 'Roma', home_team_goal>away_team_goal) %>%
factor1High_XL_h <- roma_home_match %>% filter(season=='2015/2016', XL_factor1=='XL_factor1_High') %>% c
factor1High_XR_h <- roma_home_match %>% filter(season=='2015/2016', XR_factor1=='XR_factor1_High') %>% c
factor3High_YD_h <- roma_home_match %>% filter(season=='2015/2016', YD_factor3=='YD_factor3_High') %>% c
factor12Medium_YM_h <- roma_home_match %>% filter(season=='2015/2016', YM_factor1 =='YM_factor1_Medium'
factor12High_YA_h <- roma_home_match %>% filter(season=='2015/2016', YA_factor1=='YA_factor1_High', YA_
factor2High_XR_h <- roma_home_match %>% filter(season=='2015/2016', XR_factor2=='XR_factor2_High') %>% c
factor2High_XL_h <- roma_home_match %>% filter(season=='2015/2016', XL_factor2=='XL_factor2_High') %>% c
facotr12Medium_XM_h <-roma_home_match %>% filter(season=='2015/2016', XM_factor1=='XM_factor1_Medium', 

roma_away_match <- roma_match %>% filter(away_team_api_id == 'Roma', home_team_goal<away_team_goal) %>%
factor1High_XL_a <- roma_away_match %>% filter(season=='2015/2016',XL_factor1A=='XL_factor1A_High') %>%
factor1High_XR_a <- roma_away_match %>% filter(season=='2015/2016',XR_factor1A=='XR_factor1A_High') %>%
factor3High_YD_a <- roma_away_match %>% filter(season=='2015/2016', YD_factor3A=='YD_factor3A_High') %>
factor12Medium_YM_a <- roma_away_match %>% filter(season=='2015/2016', YM_factor1A=='YM_factor1A_Medium
```

```
factor12High_YA_a <- roma_away_match %>% filter(season=='2015/2016', YA_factor1A=='YA_factor1A_High', YA
factor2High_XR_a <- roma_away_match %>% filter(season=='2015/2016', XR_factor2A=='XR_factor2A_High') %>%
factor2High_XL_a <- roma_away_match %>% filter(season=='2015/2016', XL_factor2A=='XL_factor2A_High') %>%
facotr12Medium_XM_a <-roma_away_match %>% filter(season=='2015/2016', XM_factor1A=='XM_factor1A_Medium'


factor1High_XL <- unique(rbind.fill(factor1High_XL_h, factor1High_XL_a))
factor1High_XR <- unique(rbind.fill(factor1High_XR_h, factor1High_XR_a))
factor3High_YD <- unique(rbind.fill(factor3High_YD_h, factor3High_YD_a))
factor1Medium_YM <- unique(rbind.fill(factor1Medium_YM_h, factor1Medium_YM_a))
factor2Medium_YM <- unique(rbind.fill(factor2Medium_YM_h, factor2Medium_YM_a))
factor12High_YA <- unique(rbind.fill(factor12High_YA_h, factor12High_YA_a))
factor2High_XR <- unique(rbind.fill(factor2High_XR_h, factor2High_XR_a))
factor2High_XL <- unique(rbind.fill(factor2High_XL_h, factor2High_XL_a))
facotr12Medium_XM <- unique(rbind.fill(facotr12Medium_XM_h, facotr12Medium_XM_a))
```