Business Intelligence Practical Report

Andrei Karpau
Institute of Technology Blanchardstown
Dublin 16

December 10, 2016

1 Introduction

This practical introduces the analysis that is made on some datasets related to English Football Premier League. This research uses the following tool and methodologies:

1. Use Talend Open Studio for Data Integration to apply some ETL techniques to the dataset.
2. Use Talend Open Studio for Data Quality to profile the data set.
3. Use WampServer and HeidiSQL for storing the data in MySQL database and manipulating it using SQL statements.
4. Use RapidMiner to provide some analysis on the dataset.

The aim of the research is to analyze the transfers and results of the teams in English Premier League during the years 2011-2015.

1.1 Description of the Data Set

The data sets contain the information about the player transfers and the results of the teams in English Premier League. They are downloaded from different opened sources:

1. The information about the transfers is taken from http://www.transfermarkt.co.uk. However, all the data is saved in html tables, so it is manually copied from web browser to excel spreadsheets.
2. The teams dataset is originally downloaded from football-data.co.uk web page. It contains the csv files with information about the games played in different divisions of English football league.

Figure 1: Games data

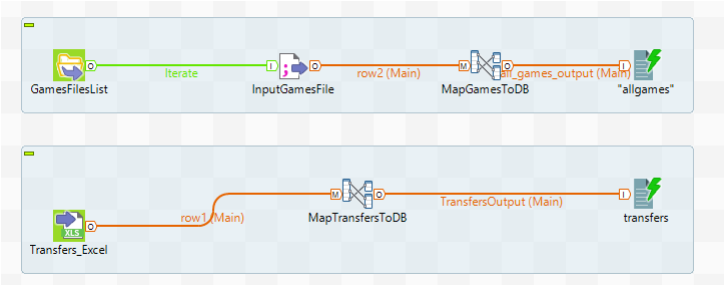

Figure 2: Integration Process

These data sets have the information about the long period of time. But each year is stored in a separate file or table. So, only the data for period between years 2011-2015 is used.
At the beginning the data is spread between the different source and formats. The transfers information is stored in Excel worksheet in xlsx file. Each year is saved in a separate worksheet which makes the analysis more complicated.
Another chunk of data contains the information about the games being played. It is stored in many csv files that are saved in folders structure based on years. The data for each league is stored in a separate csv file. The part of this structure is shown on figure 1. Since, the plenty of different files are used, at the first step it is decided to extract the data from them and copy it into the tables in local MySQL database almost without modifications. The data quality investigation is made on the second step. Then data cleansing and adjusting take place. Afterwards, the final analysis is made.

2 Data Integration

For data integration, Talend Open Studio for Data Integration is used. The whole scheme of the job is shown on figure 2. GamesFilesList component is used for looping through the whole structure of directories, subdirectories, and files that contain the information about the games in different leagues. Then it is connected to InputGamesFile component and calls it on it iteration. InputGamesFile reads each file being sent by the list, makes parsing and sends it further to the mapping phase. MapGamesToDB selects only the columns, which are needed in further research, modifies them a little bit and maps to

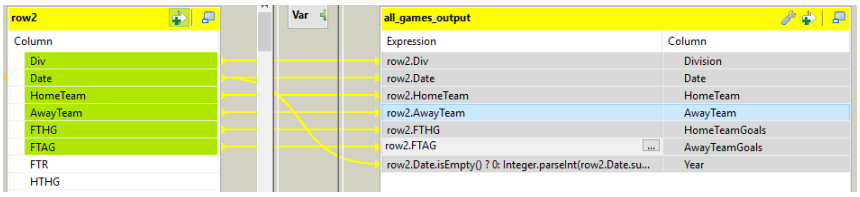

Figure 3: Games mapping scheme

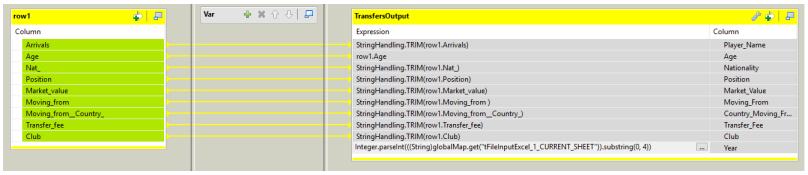

Figure 4: Transfers mapping scheme

the database table. The mapping scheme is shown on figure 3. Finally, the mapped data is stored in AllGames table in Football database on local MySQL server. The second process extracts the data from Excel worksheet. The extraction is done by Transfers_Excel element which extracts the data from all the spreadsheets. Then the mapping is done. Since the year is saved only inside of spreadsheet name, the mapping extracts it from the tFileInputExcel_1_CURRENT_SHEET variable and maps to the Year column of the table. The mapping is shown on figure 4. The mapped data is finally stored inside the Transfers table in Football database.

3 Data Quality

In order to provide some analysis on the data the quality of it should be verified at first. It is done using Talend Open Studio for Data Quality. Since the data was already extracted from all the sources and loaded into the database, only the connection to Football database is done.

For investigating the quality of Transfers table the column analysis is done. The statistics of Player Names shows 847 rows, which is 100%. However, there is one Null value 74 duplicates. The duplicates can theoretically address the players that were transferred in different years, but the further investigation is needed. On figure 5 and 6 are the null value row and duplicated rows respectively.

So, the null value is an empty one and does not address any player at all. The duplicated values contain both valid rows and the empty rows where the country is written inside the player name column. Actually, it occurs because of coping from html to Excel and addresses the case when the player has more than one nationality. Because the nationality is not relevant in current research, this row can be removed.

Age, Nationality, Position, Market Value, Moving From and Country Moving

Player Name	Age	Nationality	Position	Market Value	Moving From	Country Moving From	Transfer Fee	Club	Year
<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>	NEWCASTLE UNITED	2013

Figure 5: Null row

Player Name	Age	Nationality	Position	Market Value	Moving From	Country Moving From	Transfer Fee	Club	Year
George Boyd	28	Scotland	Right Midfield	£1.28m	Hull City	England Hull City	£3.23m	BURNLEY FC	2014
England	<null>	<null>	<null>	<null>	<null>	<null>	<null>	BURNLEY FC	2014
Michael Kightly	28	England	Left Wing	£2.30m	Stoke City	England Stoke City	£1.62m	BURNLEY FC	2014
England	<null>	<null>	<null>	<null>	<null>	<null>	<null>	BURNLEY FC	2014
England	<null>	<null>	<null>	<null>	<null>	<null>	<null>	BURNLEY FC	2014
Brazil	<null>	<null>	<null>	<null>	<null>	<null>	<null>	CHELSEA FC	2014
Italy	<null>	<null>	<null>	<null>	<null>	<null>	<null>	CHELSEA FC	2014
Martinique	<null>	<null>	<null>	<null>	<null>	<null>	<null>	CHELSEA FC	2014
France	<null>	<null>	<null>	<null>	<null>	<null>	<null>	CHELSEA FC	2014
Jamaica	<null>	<null>	<null>	<null>	<null>	<null>	<null>	CRYSTAL PALACE	2014
United States	<null>	<null>	<null>	<null>	<null>	<null>	<null>	CRYSTAL PALACE	2014
Sam Magri	20	Malta	Right-Back	£43k	Queens Park Rangers	England QPR	Free transfer	CRYSTAL PALACE	2014
England	<null>	<null>	<null>	<null>	<null>	<null>	<null>	CRYSTAL PALACE	2014
Andy Johnson	33	England	Centre-Forward	£850k	Unattached	Unattached	-	CRYSTAL PALACE	2014

Figure 6: Duplicated names

Michael Owen	32	England	Centre-Forward	£1.28m	Unattached	Unattached	<null>	STOKE CITY	2012

Figure 7: Null transfer fee

Value	Count	%
2015	20	2.36%
2014	20	2.36%
2013	20	2.36%
2012	20	2.36%
2011	20	2.36%

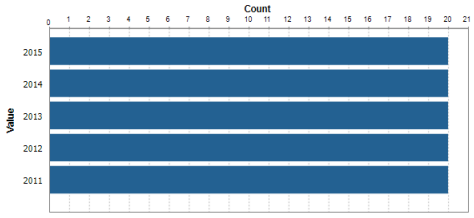

Figure 8: Clubs per season

From all have 185 null values, which address the same coping problem.
Transfer Fee has 186 null values, so in one case it is undefined (see figure 7). To avoid nulls during analysis it can be changed to 0.

The years and clubs has no null values and contain 5 distinct values for years and 30 for football clubs. However, it is worth to verify it there is 20 distinct clubs during each year, which is the rule of the English Premier League. In order to verify it User Defined Frequency Indicator was created based on SQL statement, which groups clubs by year and counts the distinct values among them. The result is shown on figure 8 and shows that the number of clubs is right for each year.

The investigation of table values shows also that Market Value and Transfer Fee contain the values in value m/k format. Some of the values are missed or marked as free transfer. So, it is worth to convert them to integer values and set all missed values to 0.
Country Moving From column has the country as the first world and the club name after it. Because it is worth to find the transfers made from England, it

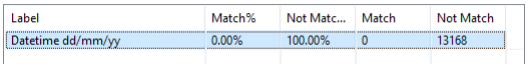

Figure 9: Date verification

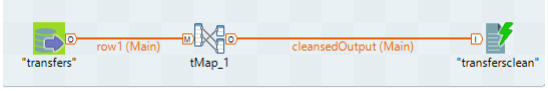

Figure 10: Cleansing transfers table

is enough to split the value by space and take always the first word which is the country name.
The investigation of AllGames table shows that the total rows count is 13168. The table is pretty clean and has no null values at all. The Division column has 5 distinct values E0, E1, E2, E3 and EC, which relate to Premier League, Championship, League 1, League 2 and Conference divisions.
The verification of data format is made by Regex expression and shows no errors (see figure 9). The distinct number of AwayTeams is equal to the distinct number of HomeTeams, which makes sense as well.

The Minimum and Maximum year values are 2011 and 2016 respectively. This is also right because the seasons 2011-2015 are considered and near the half of games in season 2015 are played in year 2016. All in all, no issues are found in AllGames table.
The summary of improvements that should be made:

1. Null rows should be removed from Transfers table.
2. The market value and transfer fee should be parsed to integer values. The null, not defined and Free Transfer values should be set to 0.
3. In Country Moving From column only the country name should be left

4 Data Cleansing

In order to make the improvements Talend Open Studio for Data Integration is used. Since all the data is already loaded into the SQL database, it is needed to extract it from one database table, then preprocess it and load into another table in database. The scheme of the job is shown on figure 10.

All the preprocessing is done during the map phase. It includes parsing of market values and transfer fees, splitting the country names, filtering out the null rows and counting the transfer fee coefficients. The map component is shown on figure 11. For parsing the price values new Routine is created and used during the mapping phase. Actually, this Routine is ParseMarketValueString class with two functions GetNonNullValue and ParseMarketValue that is written on Java.

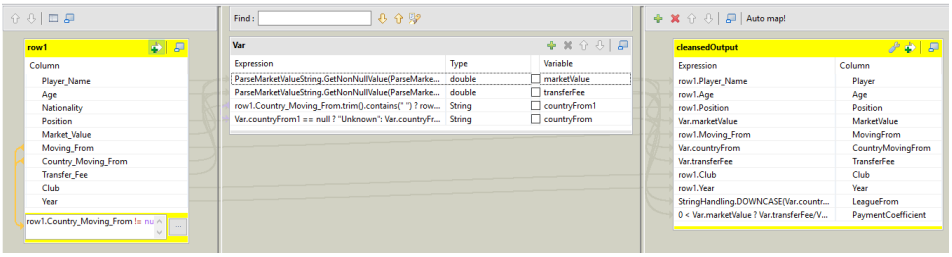

Figure 11: Mapping and cleansing transfers table

```
INSERT INTO gamesresults
SELECT Year, Team, SUM(HomeTeamGoals > AwayTeamGoals), SUM(HomeTeamGoals = AwayTeamGoals), SUM(HomeTeamGoals < AwayTeamGoals)
FROM
   (SELECT Year, HomeTeam as Team, HomeTeamGoals as HomeTeamGoals, AwayTeamGoals as AwayTeamGoals, Division as Division
   FROM allgames
   WHERE Division = "E0"
   UNION ALL
   SELECT Year, AwayTeam as Team, AwayTeamGoals as HomeTeamGoals, HomeTeamGoals as AwayTeamGoals, Division as Division
   FROM allgames
   WHERE Division = "E0") AS teamGames
GROUP BY teamGames.Year, teamGames.Team
```

Figure 12: Extracting and aggregating games results

5 Data preprocessing

In order to prepare data to statistical analysis and data mining algorithm it is needed to extract some important values from it and to load them into the separate tables. An example of such values is PaymentCoefficient, which was calculated on previous phase and saved into TransfersClean table. This coefficient shows the ratio between transfer fee and market value of a player. If it is higher than 1 it means the new club has paid more than the market price is. The values between 0 and 1 shows that the transfer fee is smaller than the actual market price. Another preprocessing is done on allgames table. In order to count all wins, losses and draws of clubs in Premier League, SQL statement is built. It is shown on figure 12. It filters only the premier division and groups all the games by teams and years. Finally, this information is saved in GamesResults table.

After the cleansing on the previous step, the TransfersClean has a column LeagueFrom, which contains information about the league from which the player comes. Currently, the only information here is if the league if foreign or English. It is needed to update the information about the English leagues and set if the player comes from Premier League or from lower league. It is made by the SQL statement, which filters the clubs based on year and joins it with the ClubsFrom information. Finally, all the rows that have LeagueFrom = Eng are analyzed and updated with the information about the league of the club that has sold the player. The SQL statement for doing it is shown on figure 13.

The last extraction being made is the aggregation of all data into the clubsSummary table that contains the data about results of each club of Premier League in each year and summarized information about the transfers. The SQL query is shown on figure 14.

The final aggregated table contains aggregated numerical and categorical

```
UPDATE transfersclean tc INNER JOIN
(
   SELECT engPlayer.Player AS Player, engPlayer.Year AS Year, engPlayer.clubFrom, tc.Club AS Club  FROM
   (
      SELECT Player, UPPER(MovingFrom) AS clubFrom, Year
      FROM transfersclean
      WHERE LeagueFrom = "Eng"
   ) AS engPlayer
   LEFT OUTER JOIN
   (
      SELECT Club, Year
      FROM transfersclean
      GROUP BY Club, Year
   ) AS tc
   ON engPlayer.clubFrom = tc.Club AND engPlayer.Year = tc.Year
) tcClubs
ON tc.Player = tcClubs.Player AND tc.Year = tcClubs.Year
SET LeagueFrom = if(ISNULL(tcClubs.Club), "Low League", "Premier League")
```

Figure 13: Updating the information about the leagues

```
CREATE TABLE clubsSummary
SELECT tr.Club, tr.Year, tr.PaymentCoefficient, tr.FromForeign, tr.FromLowLeague, tr.FromPremierLeague,
(gr.Win / (gr.Win + gr.Draw + gr.Los)) AS winCoeff,
(gr.Draw / (gr.Win + gr.Draw + gr.Los)) AS drawCoeff,
(gr.Los / (gr.Win + gr.Draw + gr.Los)) AS lossCoeff
FROM gamesresults AS gr INNER JOIN
(
   SELECT t.Club AS Club, t.Year AS Year, AVG(t.PaymentCoefficient) AS PaymentCoefficient,
   SUM(t.TransferFee) AS FeeSum, SUM(IF (LeagueFrom = "Foreign", 1, 0)) AS FromForeign,
   SUM(IF (LeagueFrom = "Low League", 1, 0)) AS FromLowLeague,
   SUM(IF (LeagueFrom = "Premier League", 1, 0)) AS FromPremierLeague
   FROM transfersclean AS t
   GROUP BY t.Year, t.Club
) AS tr
ON gr.Year = tr.Year AND gr.Club = tr.Club
```

Figure 14: Aggregating the clubs data

Club	Year	PaymentCoefficient	FromForeign	FromLowLeague	FromPremierLeague	winCoeff	drawCoeff	lossCoeff
AFC BOURNEMOUTH	2015	1,2396254239729565	1	4	3	0,2632	0,2632	0,4737
ARSENAL FC	2011	4,146570035300742	5	2	1	0,5789	0,1579	0,2632
ARSENAL FC	2012	0,9057692307692307	3	1	0	0,5000	0,2632	0,2368
ARSENAL FC	2013	0,54375	3	1	0	0,6579	0,1842	0,1579
ARSENAL FC	2014	3,002196078431372	2	0	3	0,5263	0,2632	0,2105
ARSENAL FC	2015	1,0833333333333335	1	0	1	0,6579	0,1579	0,1842
ASTON VILLA	2011	0,8273124704159187	0	0	3	0,2632	0,4211	0,3158
ASTON VILLA	2012	4,372594492564282	4	4	0	0,1538	0,3846	0,4615
ASTON VILLA	2013	1,2000384243414417	6	0	1	0,2973	0,2703	0,4324
ASTON VILLA	2014	0,4601413974040320 4	3	1	1	0,2632	0,2368	0,5000

Figure 15: Example of aggregated data

data and can be easily analyzed by different data mining tools. Since, the number of games played per season can differ among different clubs (because of games in cups), the coefficients are used for wins, draws, and loses. The part of the summarized data is illustrated on figure 15.

6 Analyzing the data

For data analysis, Rapid Miner is used. The first step is to load the data into the applications. It is made using the default SQL connection functionality. Most of the values are of integer and real types. Only two columns Club and Year are marked as polynomial. The total number of rows is 99 because one of the clubs had no transfers on one year. The simple statistical analysis made by rapid miner shows that most often the players come from foreign leagues (see figure 16). However, the transfers from foreign leagues have the highest deviation, which means that the number of foreign players bought by different clubs in different years differ. Most often from 3 to 6 players are transferred by a club per year.

The number of players that come from lower leagues is the smallest one. But its average does not differ much from inner Premier League transfers. More

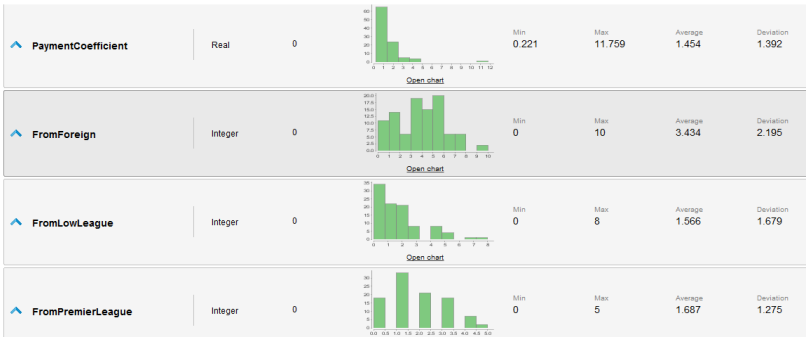

Figure 16: Statistics of transfers

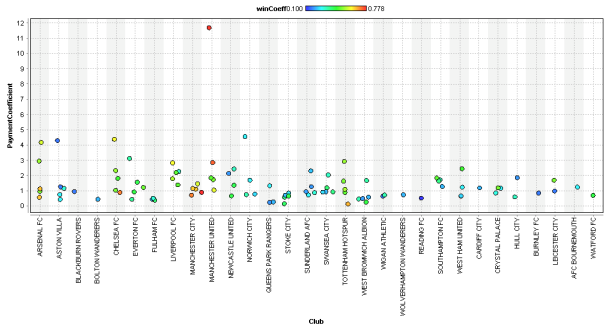

Figure 17: Payment coefficients and wins visualization

than 30% of the clubs do not buy the players from lower leagues. The rest tend to transfer only 1 or 2 players. The maximum number of transfers made by a club per summer window is 10 from foreign countries, 8 from lower leagues and 5 from Premier League. The lowest deviation is observed for inner Premier League transfers, so that means that different clubs tend to buy more or less the same number of players in this league.

Another important attribute is PaymentCoefficient that is the average coefficient per club per year. The high value here shows that the club has paid for transfers much more money that the market price is. The value between 0 and 1 shows that the paid sum is less than the market value. So, almost 70% of transfers are near the market price and often even lower. Near 25% tend to pay 1.5 2.0 market values for transfers. The outlier value is on 11.7 that shows that in one case the payments in transfer window were almost 12 times higher than the estimated prices.
For seeing the dependencies between the payments and scores the scatter plot is built (see figure 17). The Jitter is added to see all the points on the plot. The red and yellow points show the high win coefficient. It can be observed that

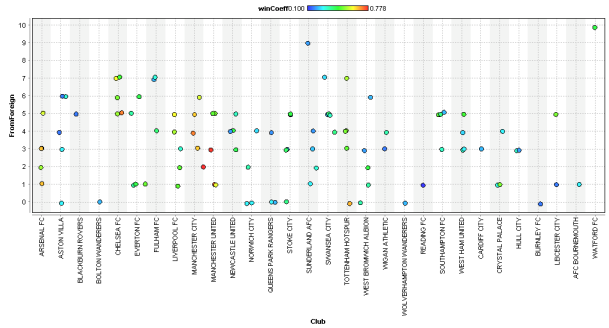

Figure 18: Transfers from foreign leagues and wins visualization

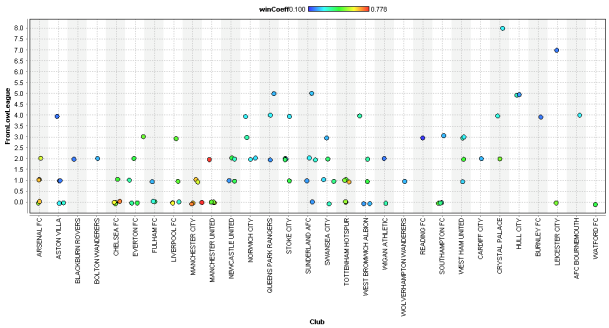

Figure 19: Transfers from lower leagues and wins visualization

except the outlier transfers made Manchester Unites (MU), the high payment coefficients do not mean a lot of wins. The red and orange values are often located around 1. The same, however, can be observed for dark blue points. Mostly, the team that pay much show neither very positive nor negative results. The exceptions are only MU that has two red points above 2 and Aston Villa, which has a blue point on the level of 4.

Also, it is worth to investigate the dependences between the leagues the players come from and the results of the clubs. The plots are shown on figures 18, 19 and 20. It can be observed that the clubs that have high values in any of these plots do not tend to show good results. The only yellow value that is placed high on the foreign leagues transfers plot is shown by Chelsea FC. Almost all the red and orange point are placed in the middle, so successive clubs make mostly the middle number of foreign transfers.

The plot that visualizes the transfers from lower leagues shows that the teams that buy a lot of players from there do not provide high results. Only MU has a high wins coefficient together with 2 transfers from lower leagues. Almost all the other red, yellow and orange points are located on the lowest

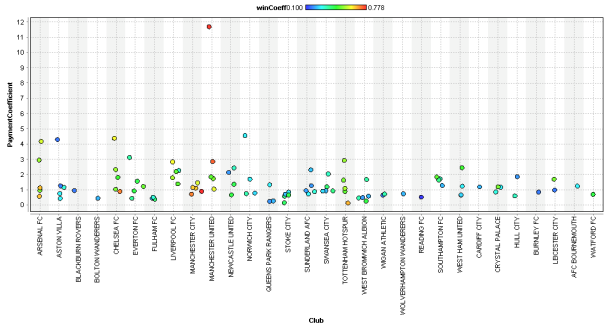

Figure 20: Transfers from Premier League and wins visualization

position.
The Premier League transfers plot shows that the successive clubs mostly make the average number of inner transfers (from 1 to 3). The team that make more inner transfers are mostly colored in green and blue, which illustrates bad win coefficients.

7 Conclusion

During this research the football dataset was extracted from different open sources. Then using the Talend Open Studio for data integration it was adjusted and stored to MySQL database. To investigate the quality of the dataset the Talend Open Studio for data quality was used. During this investigation, some inconsistencies were found in the dataset. For dealing with them, Talend Open Studio for data integration was used. The cleansing of data and storing it to another tables inside the MySQL database was made. Afterwards, to prepare the data to the final analysis, it was aggregated and saved using SQL queries. The final step was the analysis of the data. It was made using RapidMiner. The analysis has shown some dependencies between the transfer politics of the football clubs and their results.