DATA DESIGN & REPRESENTATION
FINAL PROJECT:
MARKET VALUE OF PREMIER LEAGUE PLAYERS

Group 12

Deeptish Mukherjee

Jordan Deng

Miaoqi Yang

Yvette Peng

Contents

Executive Summary

Soccer is the sport played most consistently around the world. Premier League is the most-watched sports league in the world. The transfer of players between clubs in Premier League are permitted during the winter and summer transfer windows. Such purchase and sale of players forces clubs to determine how much they need to pay for a player's transfer-in based on this player's market value. To investigate what factors affect a player's market value, this project focuses on extracting and organizing players-related data of Premier League from *premierleague* and *transfermarkt* as the 2 websites comprehensively include all the information we need. Our data includes players' demographic information, performances in historical games and their market values. Initially, we will introduce the industrial background and the reasons for choosing this project. Then, we will introduce our data sources, the routines of our web-scraping process and our database design choices. Then through comparing different database design choices such as NoSQL and SQL, we will choose the relational database in this project and highlight the advantages of the chosen database implementation. After data collection and database design, 56 variables have been categorized into 8 tables and are organized to put into SQL schemas for future analysis. Based on the database, the final section of the report proposes several implications we can conducted for further analysis.

1. Introduction

Soccer is one of the most popular sports in the world. More than 4 out of 10 people consider themselves soccer fans (Boudway, 2018). According to Statista, an online portal of statistical data with reliable sources, the total revenue of the European professional soccer market in the 2017/18 season was estimated to be at 28.4 billion euro (Gough, 2019). In terms of overall level of competition and revenue generated by this industry, Europe is the most prominent soccer market in the world (Gough, 2019). Premier League is one of the top and most-watched European leagues. According to ESPN, for the 2018/19 season the average Premier League's aggregated attendance across all matches is the highest compared to any other leagues(ESPN, n.d.). Clubs in Premier League are permitted to purchase and sale players during the winter and summer transfer windows (Premier League, n.d.). The transfer of players means transferring a player's registration from one soccer club to another, and how much a club needs to pay for a player's transfer-in is based on this player's market value. A player's market value is an estimate of the amount of money a club would like to pay to have the player sign a contract. But how do we judge what a player is actually worth? It is very interesting to investigate what factors affect a player's market value. Therefore, in this project, we'd like to web-scrape and collect players-related data in Premier League from Premier League official website and transfer market website for future analysis.

2. Data Characteristics

2.1 Data Source

We web-scraped premierleague.com and transfermarkt.us and collected 56 attributes about players in

premier league. For detailed descriptions of the 56 variables, please see data dictionary in Appendix I[1].

2.1.1 premierleague[2]

premierleague is the official website of Premier League. It includes comprehensive information of clubs and players in Premier League.

In this project, we focus on players' demographic information and performances in historical games. And compare those attributes with their market values. *premierleague* contains the information we need. In each player's page, we can collect the player's age, height, nationality, and their performances in previous games in terms of attack, defense, team play and discipline.

When doing research, we also looked at other relevant websites such as other statistical soccer websites such as ESPN, but those websites mainly focus on news about soccer games and players and do not include the information we want as comprehensive as *premierleague* does. Therefore, we use the official website of Premier League in this project.

2.1.2 transfermarkt[3]

transfermarkt, a German-based website founded in 2000, contains detailed information about soccer, including scores, results, transfer news, fixtures, and player

[1] Appendix I: Data Dictionary

[2] https://renegadeexpressions.com/2016/08/03/renegades-2016-2017-english-premier-league-predictions/

[3] https://www.transfermarkt.com/

values. According to the IVW, *transfermarkt* is in the top 25 most visited German websites, and one of the largest sport websites (Schröder, 2010). According to researchers from the Centre for Economic Performance, the players transfer values listed on *transfermarkt* are largely accurate (Bryson et al., 2009), making this website a relatively reliable source to extract players' market value.

2.2 Web-Scraping Routines

2.2.1 premierleague

On Premier League official website, we need to extract data from 2 web pages:

Player Overview: demographic information about players such as age, nationality and height

Player Stats: players' performances which are divided into 4 categories: attack, defence, team play and discipline. Steps are listed as follows:

1. Navigate to the url "www.premierleague.com/clubs" where 20 clubs are listed
2. Save this web page to local file as htm and name it "club.htm"
3. Open club.htm file and then parse it with Beautiful Soup
4. Fetch the "href" values in the webpage and store these links of team pages in an empty list; each value in the list represents the link of a club
5. Store the values in the list to in a text file called "clubs.txt" in local
6. Make adjustments to the links so that they can direct to the right place. The urls generated by the last step direct to an "overview" page that doesn't contain the information we want. Therefore, we enter the "squad" page where each player is listed by replacing "overview" with "squad" at the end of the url. We also need to add "http://www.premierleague.com" at the beginning of each link
7. Created a folder "Clubs" to store the downloaded htm files of the 20 clubs

8. Open the htm files of 20 clubs in “Clubs” folder and download players’ web pages from the clubs’ web pages and store the players’ links in a text file “players.txt”
9. Adjust players' overview links to the correct links and save to local text file “players_overview.txt” which contains links for all players in premier league
10. Open each link in “players_overview.txt”, fetch the "href" values in the webpage and store these links of player pages in an empty list; each value in the list represents the link of a player
11. Create a local folder “Players_overview” to store the downloaded htm files of all players’ overview pages
12. Read each htm file of players in the “Players_overview” folder, extract demographic information of each player from the “Player Overview” page and store the data into a list called “rows”.
13. Adjust the urls to redirect to “stats” page by replacing “overview” at the end of urls to “stats”
14. Create another folder “Players_stats” to store the downloaded htm files of all players’ stats pages
15. Read each htm file of players in the “Players_stats” folder, extract information about each player’s detailed performances in attack, defence, team play and discipline and store the data in another list called “rows_stats”.
16. Transform the 2 lists to dataframe and merge them together.

2.2.2 transfermarkt

In order to obtain each player's market value, we also scraped data on transfermarkt website (https://www.transfermarkt.com/). Steps are listed as follows:

1. Save the webpage (https://www.transfermarkt.com/premier-league/startseite/wettbewerb/GB1) to local and name it as "transfermarket.htm".

2. Open the htm file and use BeautifulSoup to parse the htm file.

3. Fetch the "href" values in the webpage and store these links of team pages in an empty list "teamlinks".

4. Create "marketvalue" folder, traverse down through “teamlinks” list and store the webpage of each team(.htm).

5. Read each htm file in the marketvalue folder, parse the htm file, extract player id, player name, market value, team name.

6. Because the current_market_value is not in the digital form we define a function to convert the market value into “thousand” format and apply the function. Then drop the original current_market_value column.

7. To create a unique 4-digit id as the primary key revealing which team the player belongs to and the corresponding player_id for each player, we first define a function to create a new player id which can convert a one-digit player id to the format with leading zero and create a unique id for each player 'team_player_id' by combining team id and player id.

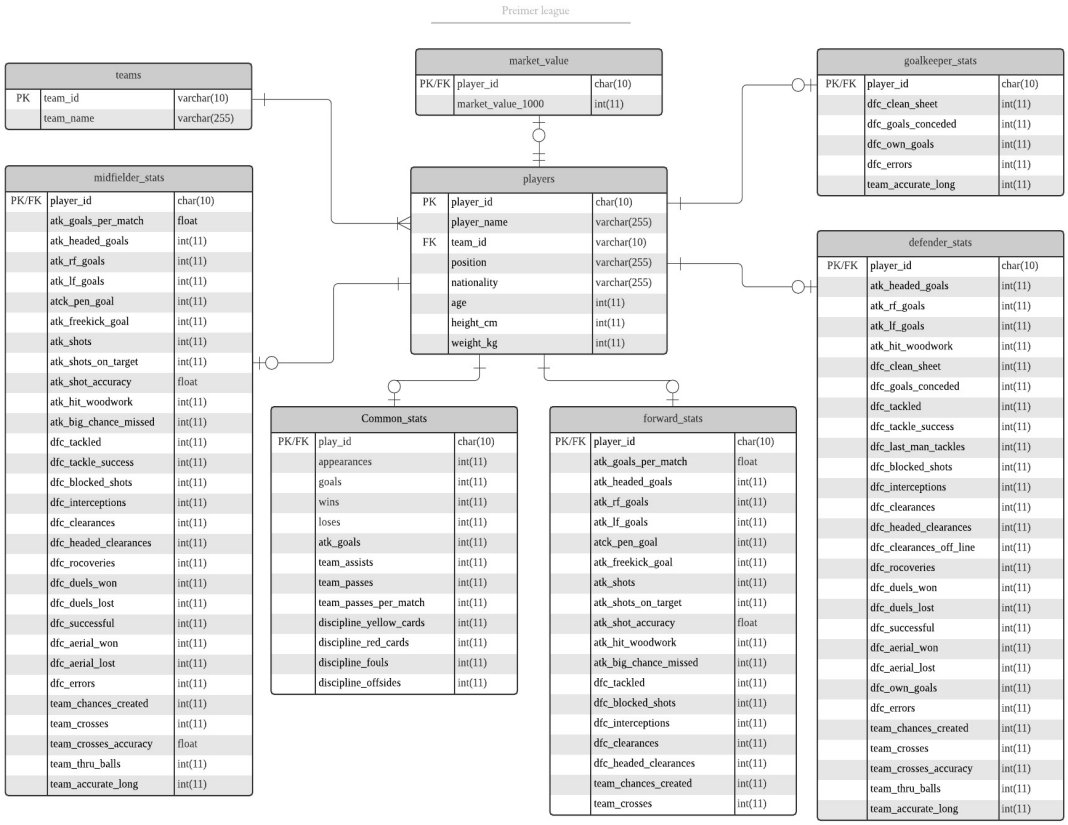
8. Connect the data to MySQL database. Specific steps are as follows: We first establish connection to mysql and create the database 'PremierLeague'.

9. Create a table 'market_value' in the database, and the data frame was inserted row by row into the table 'market_value’.

2.3 Database Design

To reduce data redundancy, we designed a relational database with 8 tables: *teams*, *players*,

market_values, common_stats, defender_stats, forward_stats, goal_keeper_stats and midfielder_stats. The linkage between these tables are shown in the ERD. In the *teams* table, we store names of 20 clubs in the Premier League. Players' personal information such as name, nationality and age are stored in *players* table. The *market_value* table includes the market value of players in units of one thousand dollars. To avoid including any redundant null values, we did not dump all the performance metrics for all the players in one SQL entity table, because some player positions may have distinct metrics that other positions do not possess. We categorized the metrics that are commonly shared between players or are specific to one kind of position. For example, the attributes in the *common_stats* table, such as appearance and goals, are shared by all players. However, the *defender_stats* table displays the metrics that the defender players have while players in other positions may or may not exhibit. Table Schemas are illustrated in Appendix II[4]. The Entity Relationship Diagram is shown below:



[4] Appendix II: Table Schemas

3. Dataset Implication

In practice, there are other database design choices other than relational databases, such as NoSQL databases. It is acknowledged that NoSQL databases are more flexible that the data structure is schema-free, and at the same time, NoSQL are user-friendly in preparing data for text analysis. Nevertheless, in this project, it would be more appropriate to choose the relational database, because it requires a pre-designed schema that gives the designers more control in terms of the relationship between tables. Since the majority of data in this project are continuous variables, a stringent rule of data type of each field attribute can be defined in SQL database. Another merit of SQL database is that it pushes the future users who desire to vertically scale the database to follow the data type rule designers wish them to use. The database implemented in this project will massively facilitate the process of querying data in our ongoing machine learning analysis. The research aims to quantify the impact of performance metrics on the market value of players, and naturally the market value varies in different patterns based on different player positions.

Therefore, we will conduct analysis separately for players of different positions, for example, conducting one linear regression on defenders and another regression on goalkeepers. The current relational database stores the data of each position respectively, thereby eliminating the need for researchers to categorize the data on their own. The data extracted from the database will have the data types in compliance with the researchers' needs, reducing researchers' time to tidy up the data format as well.

To wrap up, in this project, the relational database will reduce researchers' efforts wasted on data cleaning and will allow them to focus on the core analysis.

4. Summary and Conclusion

To conclude, information of players of Premier League are web-scraped from 2 reliable websites, *premierleague* official website and *transfermarkt* website. The collected data has 56 attributes in total, which comprehensively includes players' demographic information, performances in historical games and their market values. Based on the nature of variables we collected, a relational database was used to store our data because it gives researchers more controls and saves them time cleaning the data.

5. Reference

(1) Boudway, I. (2018, June 12). *Soccer Is the World's Most Popular Sport and Still Growing*. Bloomberg. https://www.bloomberg.com/news/articles/2018-06-12/soccer-is-the-world-s-most-popular-sport-and-still-growing

(2) Gough, C. (2019, Oct 23). *Market size of the European professional football market from 2006/07 to 2017/18.* Statista. https://www.statista.com/statistics/261223/european-soccer-market-total-revenue/

(3) ESPN. (n.d.). *English Premier League Performance Stats - 2018-19.* https://global.espn.com/soccer/stats/_/league/ENG.1/season/2018/view/performance

(4) Premier League (n.d.). *Find about the transfer system in the Premier League*. https://www.premierleague.com/transfers

(5) Schröder, J. (2010, Sept 09). *Online-IVW: Sport gewinnt, News verliert.* Meedia. https://meedia.de/2010/09/09/online-ivw-sport-gewinnt-news-verliert/

(6) Bryson, A., Frick, B. & Simmons, R. (2009). The Returns to Scarce Talent: Footedness and Player Remuneration in European Soccer. *Center for Economic Performance*, *CEP Discussion Paper, No. 948.*

6. Appendix

Appendix I Data Dictionary

Field_name	Description
player_id	unique key identifying the players in this table
player_name	name of each player
team_id	unique key identifying the teams in this table
team_name	name of each team in Premier League
position	position of the player in the game
nationality	where the player comes from
age	age of players based on current date
height_cm	the heights of players in cm
weight_kg	the weights of players in kg
appearances	total appearances in the games
goals	total goals
wins	total wins
loses	total loses
dfc_clean_sheet	when a team does not allow the opponent team to score in the match
dfc_goals_conceded	to fail to stop the opponent team from winning a point
dfc_own_goals	a goal scored inadvertently when the ball is struck into the goal by a player on the defensive team
dfc_errors	when making an error which leads to a goal or shot conceded
team_accurate_long	accurate long pass
atk_goals	score a goal
atk_headed_goals	using head to score a goal
atk_rf_goals	score a goal using right foot
atk_lf_goals	score a goal using left foot
atk_hit_woodwork	hitting the post, to almost score
dfc_tackled	attempting to take the ball away from an opponent's possession
dfc_tackle_success	successful taking the ball away from an opponent's possession
dfc_last_man_tackles	
dfc_blocked_shots	to block the shot by throwing their body between the ball and the goal
dfc_interceptions	intentionally intercepting a pass by moving into the line of the intended ball
dfc_clearances	kicking the ball away from the goal they are defending
dfc_headed_clearances	moving the ball away from the goal they are defending with head
dfc_clearances_off_line	the whole of the ball passes over the goal line
dfc_rocoveries	gaining possession after control of the ball has been lost by the opposition
dfc_duels_won	a direct competition or faceoff between opposing players at any point on the pitch competing to get possession of the ball and win
dfc_duels_lost	a direct competition or faceoff between opposing players at any point on the pitch competing to get possession of the ball and lost
dfc_successful	
dfc_aerial_won	
dfc_aerial_lost	

Field_name	Description
team_chances_created	a pass/cross that is instrumental in creating a goal-scoring opportunity
team_crosses	a medium- to-long-range pass from a wide area of the field towards the centre of the field near the opponent's goal
team_crosses_accuracy accuracy of crosses	
team_thru_balls	The ball is sent from the back line or midfield between opposing defenders and into open space for an attacker to run onto the ball and threaten goal
atk_goals_per_match	goals per match
atck_pen_goal	goal when restarting the game by taking a single shot on the goal while it is defended only by the opposing team's goalkeeper
atk_freekick_goal	goal when restarting the game following an offence by the opposing side
atk_shots	an attempt that is taken with the intent of scoring and is directed toward the goal
atk_shots_on_target	any goal attempt that ... Is a clear attempt to score that would have gone into the net but for being saved by the goalkeeper or is stopped by a player who is the last-man with the goalkeeper having no chance of preventing the goal (last line block)
atk_shot_accuracy	the accuracy of shots
atk_big_chance_missed	when a player should reasonably be expected to score, usually in a one on one scenario or from very close range when the ball has a clear path to goal and there is low to moderate pressure on the shooter
team_assists	passing or crossing the ball to the scorer which scores a goal
team_passes	kicking the ball to a teammate
team_passes_per_match	passes per match
discipline_yellow_cards	total number of yellow cards received
discipline_red_cards	total number of red cards received
discipline_fouls	an unfair act by a player, deemed by the referee to contravene the game's laws, that interferes with the active play of the game
discipline_offsides	at the position where is nearer to his opponents' goal line than both the ball and the second last opponent
market_value_1000	market valuations of players assessed at the beginning of a season as a proxy for undisclosed salary (in “thousand” format).

Appendix II Table Schemas

List of Tables:

Table_name
Players
Teams
Goalkeeper_stats
Defender_stats
Midfielder_stats
Forward_stats
Common_stats
market_value

players

Field_name	Datatype	Key	Default_values
player_id	char(10)	Primary Key	Null
Player_name	varchar(255)		Null
Team_id	varchar(10)	Foreign Key	Null
Position	varchar(255)		Null
Nationality	varchar(255)		Null
Age	int(11)		Null
Height_cm	int(11)		Null
Weight_kg	int(11)		Null

teams

Field_name	Datatype	Key	Default_values
Team_id	varchar(10)	Primary Key/ Foreign Key	Null
team_name	varchar(255)		Null

Goalkeeper_stats

Field_name	Datatype	Key	Default_values
player_id	char(10)	Primary Key/ Foreign Key	Null
dfc_clean_sheet	int(11)		Null
dfc_goals_conceded	int(11)		Null
dfc_own_goals	int(11)		Null
dfc_errors	int(11)		Null
team_accurate_long	int(11)		Null

Defender_stats

Field_name	Datatype	Key	Default_values
player_id	char(10)	Primary Key/ Foreign Key	Null
atk_headed_goals	int(11)		Null
atk_rf_goals	int(11)		Null
atk_lf_goals	int(11)		Null
atk_hit_woodwork	int(11)		Null
dfc_clean_sheet	int(11)		Null
dfc_goals_conceded	int(11)		Null
dfc_tackled	int(11)		Null
dfc_tackle_success	int(11)		Null
dfc_last_man_tackles	int(11)		Null
dfc_blocked_shots	int(11)		Null
dfc_interceptions	int(11)		Null
dfc_clearances	int(11)		Null
dfc_headed_clearances	int(11)		Null
dfc_clearances_off_line	int(11)		Null
dfc_rocoveries	int(11)		Null
dfc_duels_won	int(11)		Null
dfc_duels_lost	int(11)		Null
dfc_successful	int(11)		Null
dfc_aerial_won	int(11)		Null
dfc_aerial_lost	int(11)		Null
dfc_own_goals	int(11)		Null
dfc_errors	int(11)		Null
team_chances_created	int(11)		Null
team_crosses	int(11)		Null
team_crosses_accuracy	int(11)		Null
team_thru_balls	int(11)		Null
team_accurate_long	int(11)		Null

Midfielder_stats

Field_name	Datatype	Key	Default_values
player_id	char(10)	Primary Key/ Foreign Key	Null
atk_goals_per_match	float		Null
atk_headed_goals	int(11)		Null
atk_rf_goals	int(11)		Null
atk_lf_goals	int(11)		Null
atck_pen_goal	int(11)		Null
atk_freekick_goal	int(11)		Null
atk_shots	int(11)		Null
atk_shots_on_target	int(11)		Null
atk_shot_accuracy	float		Null
atk_hit_woodwork	int(11)		Null
atk_big_chance_missed	int(11)		Null
dfc_tackled	int(11)		Null
dfc_tackle_success	int(11)		Null
dfc_blocked_shots	int(11)		Null
dfc_interceptions	int(11)		Null
dfc_clearances	int(11)		Null
dfc_headed_clearances	int(11)		Null
dfc_rocoveries	int(11)		Null
dfc_duels_won	int(11)		Null
dfc_duels_lost	int(11)		Null
dfc_successful	int(11)		Null
dfc_aerial_won	int(11)		Null
dfc_aerial_lost	int(11)		Null
dfc_errors	int(11)		Null
team_chances_created	int(11)		Null
team_crosses	int(11)		Null
team_crosses_accuracy	float		Null
team_thru_balls	int(11)		Null
team_accurate_long	int(11)		Null

Forward_stats

Field_name	Datatype	Key	Default_values
player_id	char(10)	Primary Key/ Foreign Key	Null
atk_goals_per_match	float		Null
atk_headed_goals	int(11)		Null
atk_rf_goals	int(11)		Null
atk_lf_goals	int(11)		Null
atck_pen_goal	int(11)		Null
atk_freekick_goal	int(11)		Null
atk_shots	int(11)		Null
atk_shots_on_target	int(11)		Null
atk_shot_accuracy	float		Null
atk_hit_woodwork	int(11)		Null
atk_big_chance_missed	int(11)		Null
dfc_tackled	int(11)		Null
dfc_blocked_shots	int(11)		Null
dfc_interceptions	int(11)		Null
dfc_clearances	int(11)		Null
dfc_headed_clearances	int(11)		Null
team_chances_created	int(11)		Null
team_crosses	int(11)		Null

Common_stats

Field_name	Datatype	Key	Default_values
player_id	char(10)	Primary Key/ Foreign Key	Null
appearances	int(11)		Null
goals	int(11)		Null
wins	int(11)		Null
loses	int(11)		Null
atk_goals	int(11)		Null
team_assists	int(11)		Null
team_passes	int(11)		Null
team_passes_per_match	int(11)		Null
discipline_yellow_cards	int(11)		Null
discipline_red_cards	int(11)		Null
discipline_fouls	int(11)		Null
discipline_offsides	int(11)		Null

Market_value

Field_name	Datatype	Key	Default_values
player_id	char(10)	Primary Key/ Foreign Key	Null
market_value_1000	int(11)		Null