IS5126 – Hands on with Business Analytics

Guided Project - 1

Group No : 06

Submitted on : 09-Feb-2015

Submitted By,

Jayakumar Alagappan Meenakshi	**(A0134431U)**
Krishna Kannan	**(A0134475A)**
Raghavendhra Balaraman	**(A0123443R)**
Vishnu Gowthem Thangaraj	**(A0134525L)**

Part 1 - Web Scrapping

Q1) Identify one or more starting URL to crawl:

Players :

The players details are crawled starting from the below URL http://www.basketball-reference.com/players/. From this page individual player details are recursively crawled based on their alphabets. http://www.basketball-reference.com/players/<alphabet>/<playername.html>

For ex., a player name 'Alaa Abdelnaby' is identified by first crawling the main page http://www.basketball-reference.com/players/ . From this page, http://www.basketball-reference.com/players/a is identified and from there the player's page i.e., http://www.basketball-reference.com/players/a/abdelal01.html is scrapped to obtain the player information of 'Alaa Abdelnaby'. Similarly the other player informations are obtained. The scrapped data includes 4288 players.

Team:

The team details are crawled starting from the below URL. http://www.basketball-reference.com/teams/. This page redirects us to all the franchises available. This URL is recursively crawled to obtain the individual team page.

For ex., for a team name 'Atlanta Hawks' details is first identified by crawling the main page http://www.basketball-reference.com/teams/ . From the franchise page, 'Altanta Hawks' team details are identified by crawling their corresponding team page here., http://www.basketball-reference.com/teams/ATL/ .

Q2) Identify the links to follow using both Beautiful Soup and Regular Expressions. You may want to first grab the HTML, store it on your local computer which you can then use to tweak your parser.

a) Provide the regular expressions to do so

<u>To Crawl URLS:</u>

```
txt='<a href="/players/a/">A</a>'
regex='.*?((?:\\\/[\\w\\.\\-]+)+)'
rg = re.compile(regex,re.IGNORECASE|re.DOTALL)
m = rg.search(txt)
if m:
    path=m.group(1)
    print (path)
```

<u>To identify the details.</u>

```
txt='<td align="right" >934</td>'
regex='.*?(\\d+)'
rg = re.compile(regex,re.IGNORECASE|re.DOTALL)
m = rg.search(txt)
if m:
    id=m.group(1)
    print (id)
```

Player:

b) Provide the Beautiful Soup version code.

Team :

- Identify the main team url ie., http://www.basketball-reference.com/teams
- Identify all the individual franchise url using

```
scrapeContent = soup.find_all('tr', {'class' : 'full_table'})
team[content.td.a.text]['name'] = content.td.a.text
team[content.td.a.text]['url'] = url +
  content.find('a').get('href').replace("/teams", "")
```

- Get team info using

```
scrapeContent = soup.find_all('div', {'class' :
'mobile_text'})
statsData = content.find_all('span')
```

- Get team stats using

```
scrapeContent = soup.find_all('tr', {'class' : ''})
statsData = content.find_all('td')
```

Player:

- Identify the alphabetical url i.e.,http://www.basketball-reference.com/players/a
- Identify the table containing the list of players using

```
players = soup.find("div_players", id=divid)
```

- Obtain the player link from the table using

```
table_td_th.find("a")['href']
```

- Obtain the player statistical information

```
players_total = soup.find("all_totals", id=divid)
table_rows = players_total.find_all("tr")
table_tds_ths = table_row.find_all("td");
```

- Obtain the player salary informations

```
player_salaries = soup.find("all_salaries", id=divid)
table_rows = players_total.find_all("tr")
table_tds_ths = table_row.find_all("td");
```

c) You can choose either code to use to grab your data, which did you choose and Why ?

We chose to use BeautifulSoup. BeautifulSoup is a fully auto-tolerant module which scrapes the web pages smartly. On the other hand regexp is very bounded in what it can correctly extract from the HTML. Also Beautiful soup does not have complicated syntax which makes coding easy.

Q3) On each player's page, you should at least parse the following information using BeautifulSoup.

a) Basic Player Profile Information.

The scratched data is available in File/PlayerInfo.csv. The following are the details that are available.

Name,From – Starting year of a player's basket ball career ,Ending year of a player's basket ball career, Position, Height, Weight, Date of Birth, College, URL – Players page, Shoots, Deceased Date

b) Player statistics

The crawled data is available in Files/PlayerStatistics.csv. The file contains all the details from the 'Totals' div of the player's page which includes

Name, URL – Players page, Season, Age, Team, League, Position in that season, Games , Games started, Minutes Played, Field Goals, Field Goals Attempt, Field Goals Percentage etc.,

c) Player Salaries

The players salaries are available in File/PlayerSalaries.csv which includes the following information.

Player Name, Player URL, Season, Team, Team URL, League, Salary

d) Can you repeat part (a) using Regex ? Which method is better, Regex or BS4 ? What do you mean by better ? Which do you better ?

HTML is not a typical language, but it is a context-free language. On the contrary, HTML has structures that run peremptorily deep.

"Regexp is so powerful, yet easy to use, and saying that they are just wicked, would be an insult." (http://www.digitalamit.com/article/regular_expression.phtml)
But regex queries are not armed to break down HTML into its succinct parts. Using regex is convenient for highly structured language.
BeautifulSoup is a fully auto-tolerant module which scrapes the web pages smartly. On the other hand regexp is very bounded in what it can correctly extract from the HTML.
Both the methods have their pros and cons; and have their own use cases.

Scenario 1 : Extraction of Franchise table information

Data points captured : Name, Championship
Here BeautifulSoup takes longer than regex. This is because the BeautifulSoup library slurps the entire document into its internal format, when only Name and Championship data points are required.

Scenario 2: **Extraction of Player Information**

Here beautiful soup is preferred because, the player statistical informations are available in HTMLS divisions. Thus by providing appropriate division ids to BeautifulSoup helps retrieving the necessary data easily. Whereas regex takes very careful scripting of identifying the opening and closing of HTML divisions properly.
As a conclusion, if more than 50% of the data from a web page is scrapped BeautifulSoup is very ideal.

Q4) On each team (Franchise) page, you should at least parse the following.

a) Basic Team Information

The team information are available in TeamInfo.csv. This file contains the main required team information. This file also serves as the mapping between the team names and their franchises.

TeamName, URL – Individual Team URL, Championship,

b) Team Statistics by Season

The team statistics during every season is available in TeamStats.csv. This file includes the following information.

FranchiseCode, FranchiseURL, Team, TeamCode, TeamURL, Season, League, Wins, Loss, Win-Loss %, RegularSeasonFinish , Playoffs, Coaches, Top WinShare et.,

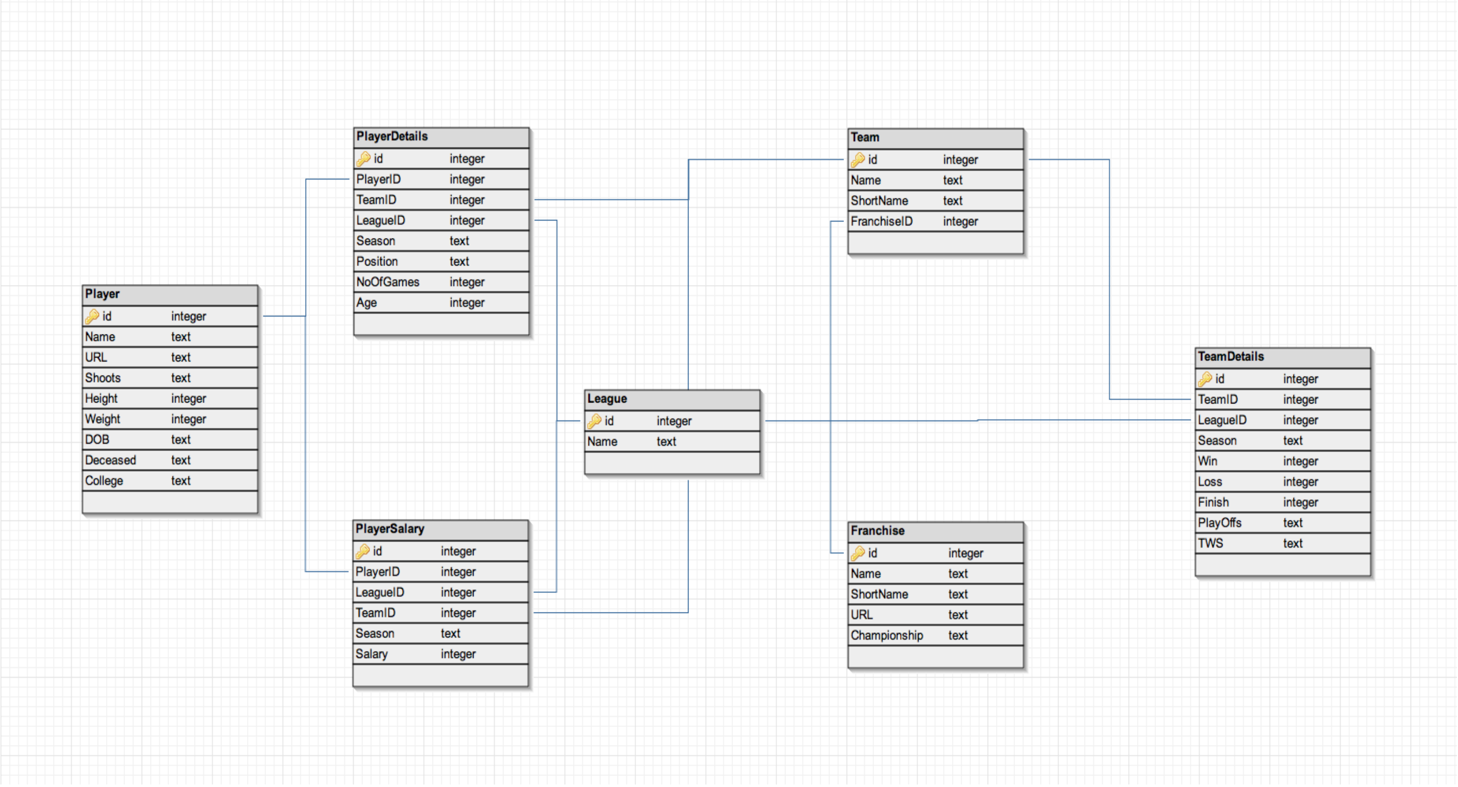
Q5) Other Information. What other raw data from the site might be useful to build analytics for team owners?

In order to analyze the performance of their own team, team owners could use the Team Details table from our db. (Refer in Schema.png). Team owners could also use other metrics to measure player performances of the past season **using metrics such as Field Goal %, 3 Pointer %, Rebound %, offensive win shares, defensive win shares, total win shares** and the awards per season, and also find players who would strengthen the weak areas of the team. **Awards such as Rookie player of the year, MVP of the year indicate the top performers in the league.** Individual player popularity should also be taken account. A team owner could also look to add such a player to their roster, for gains on and off the pitch, from a financial standpoint.

Also, team owners could decide to sack and replace a general manager of the team. The metrics for general managers such as previous championships won, previous teams managed, win-loss % are available at http://www.basketball-reference.com/executives/

ETL Part 2 : For all the question below, their corresponding query is in Query.pdf

Q1) Design the database schema using the normalization guideline for your data, and load the data. Please try to reduce it as much as possible



PlayerDetails	
id	integer
PlayerID	integer
TeamID	integer
LeagueID	integer
Season	text
Position	text
NoOfGames	integer
Age	integer

Team	
id	integer
Name	text
ShortName	text
FranchiseID	integer

Player	
id	integer
Name	text
URL	text
Shoots	text
Height	integer
Weight	integer
DOB	text
Deceased	text
College	text

League	
id	integer
Name	text

TeamDetails	
id	integer
TeamID	integer
LeagueID	integer
Season	text
Win	integer
Loss	integer
Finish	integer
PlayOffs	text
TWS	text

PlayerSalary	
id	integer
PlayerID	integer
LeagueID	integer
TeamID	integer
Season	text
Salary	integer

Franchise	
id	integer
Name	text
ShortName	text
URL	text
Championship	text

Q2). For 2011-2012 player details,

1) Find how many active players are there.

There are 478 distinct active players available during the season 2011-2012. The above information was obtained by querying the 'Playerdetails' table for total count of players played during 2011-2012.

2) How many play in each position ?

The following are the position information obtained from the 'PlayerDetails' table.

C	101
PF	98
PG	94
SF	95
SG	93

(C – Centre, PF – Power Forward, PG – PointGuard, SF – Small Forward,SG-Shooting Guard)

The above information also includes data in which a single player played two different positions for two different teams during a season

3) What is the average age, average weight, average experience, average salary in the season and average career salary ?

Average Age - 26.6 yrs
Average Weight – 209
Average Experience - 5.8 yrs
Average Salary in the season - $4,289,844
Average Career Salary - $36,358,655.5

Q3). More descriptive statistics on salaries :

a) Who are the top 10% best paid players in 2011-2012 season ? Which teams did they play for.

The above information is obtained by using the 'PlayerSalary' , 'Player' and 'Team' tables. The records are first sorted in DESCENDING order, with highest paid player in the start of the list. The records are then limited using the 'LIMIT' keyword provided by SQLITE3. The snippet of restricting 10% records used in the query is given below. The total records obtained was 46.

LIMIT (select cast(count(distinct playerid) /10 as INTEGER) from playersalary where season='2011-12').

Note: The data obtained is available in Results/Q3.xls – Sheet1(Q3-a)

b) Who are the top 10% worst paid players in 2011-2012 season ? Which teams did they play for.

The above information is obtained by using the 'PlayerSalary' , 'Player' and 'Team' tables. The records are first sorted in ASCENDING order, with highest paid player in the start of the list. The records are then limited using the 'LIMIT' keyword provided by SQLITE3. The snippet of restricting 10% records used in the query is given below. The total records obtained was 46

LIMIT (select cast(count(distinct playerid) /10 as INTEGER) from playersalary where season='2011-12').

Note: The data obtained is available in Results/Q3.xls – Sheet2(Q3-b)

c) Who are the middle 50% by pay ? Which teams did they play for ?

The above information is obtained by using the 'PlayerSalary' , 'Player' and 'Team' tables. The records are first sorted in ASCENDING or DESCENDIN order, The records are then limited using the 'LIMIT' keyword provided by SQLITE3. A range of records can be obtained from a particular record using the 'Offset' option in LIMIT keywords provided by SQLITE3. The general syntax is as follows

LIMIT VALUE, OFFSET

Here VALUE contains 25% of total records(here its 46). The OFFSET contains 50% of total records (here its 228). Now the above syntax changes as follows,

LIMIT 46, 228

which fetched 228 records from the 46th record, which obviusly is 50% of the middle records. The total records obtained was 228.

The SQL snippet is as follows.

LIMIT (select cast(count(distinct playerid) /4 as INTEGER) from playersalary where season='2011-12'), (select cast(count(distinct playerid) /2 as INTEGER) from playersalary where season='2011-12')

Note: The data obtained is available in Results/Q3.xls – Sheet3(Q3-c)

d) Over all seasons of the active players in 2011-2012 season, how much money was paid to all users by season, how many players were active in each season, what is the average per player by season

The above information was obtained by first querying the 'PlayerDetails' table to obtain the active players of '2011-2012' . This information is then used to obtain the season wise sum of salary and count of active players from the 'PlayerSalary' and 'PlayerDetails' tables respectively.

Note: The data obtained is available in Results/Q3.xls – Sheet4(Q4-d)

Q4) Team Player Statistics :

a) What is the average salary of each team by season of each team starting in 2002 and finishing 2012 season ? What is the variance of the salaries

The average salaries are obtained by querying the 'Player', 'PlayerDetails' , 'Team' and 'Player Salaries'. The variance is then obtained by substituting the mathematical equation

$$Var(X) = E(X^2) – [E(X)]^2$$

where E(X) is the average or mean of salaries

b) What is the average age of players by season ? Average and Variance of experience by season of each team ?

This information is obtained by querying the 'playerdetails' table and finding the average age of the players during that season

c) Can you provide the above in a "cross tabulation format" ? That is teams are on each row, each column is a year and the values are the metrics above.

The above data is tabulated and following are its details

Cross Tabulation for (a) : Results/Q4.xls – Sheet1(Q4-a)

Cross Tabulation for (b) : Results/Q4.xls – Sheet2(Q4-b)

Q5) What other data from the basketball-reference.com can you use to explain salary? You may wish to scrape more data from the website. What is your recommendation to team owners? How can you justify high prices for players?

Other data that can be used to explain the salary of a player are :

Contract information :

Players contract information can be used to draw various conclusions regarding salary. Consider a scenario where a team may decide to offer a new player or tie down an existing player with a huge salary contract for several years, so as to ward off competition from other teams to sign the player. **Although the player has not played well in recent seasons or not even played at all, due to such an existing contract, his salary could still be high.** Player contract information can be scraped from http://www.basketball-reference.com/contracts/players.html.

Age, Potential and College :

In today's sports, a player's Age is a vital factor. Although there are cases where older players are valued more, **the opinion by large is an equally performing younger player has more potential than an older player, because of the years ahead of him. This subsequently can lead to cases with a younger player getting offered a higher salary** than an older player, although the latter's current performance might even be slightly more for the season. Also college basketball statistics can be scraped from http://www.sports-reference.com/cbb/ which could be used to determine the salary of a new player drafted into the league.

Team Owners, Philosophy and High Prices :

Though different teams could have different philosophy and styles of play (Some might prefer to stay offensively stronger, rather than to strengthen a mediocre defense. After all, Attack is the best form of defense.) And teams could decide to draft players accordingly. Team Owners/Executives should conduct an analysis of their team's performances over the year, find areas and skills they are weak in and strengthen correspondingly, according to their philosophy of play to achieve their targets for the season. **Teams playing over a long period are likely to have a high probability of having good team chemistry. So a team**

could decide to retain a good player, offering him a higher salary, rather than buy another player. Also a team, weak in a particular area, Say they require a center position player and desperate to strengthen, may be required to overpay for the required player leading to a high salary contract because of the lesser availability of center position players in the market.