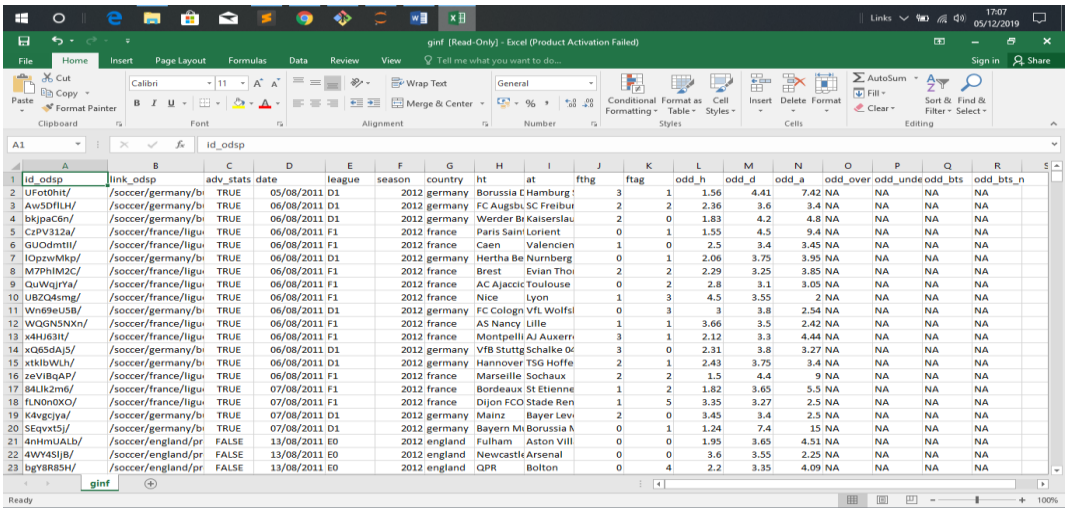
CA270 Project Report

Kian Sweeney DS2

Choosing my dataset

For my project I had to firstly choose my dataset. I have a keen interest in sport, especially Gaelic football and soccer. Initially I wanted to base my project off my favourite soccer team, Leeds United but realised that it would be too small of a dataset and easy to identify specific players when using SQL queries (i.e. if there was only one player who played every game in a season he would be known straight away in a query result). I then decided to look on kaggle to see if there was anything I would have an interest in related to Gaelic football or soccer. There were very little available datasets for GAA so after much deliberation I decided to base it off a 'football events' dataset. It had two datasets sets, one entailing odds of over ten thousand matches that took place between 2012 and 2017 in the top divisions in England, Germany, France, Italy and Spain. It also had a highly detailed events dataset, which took over one hundred events in each game and specified what happened in each respective game in the other dataset. It had over nine hundred thousand rows and contained a lot of long winded values in the columns. I decided it was best to avoid this dataset as it had too many columns and did minor events such as a missed shot on target really correlate to football betting odds or influencing games. I thus decided on using the betting odds dataset on its own.

Cleaning my Data



id_odsp	link_odsp	adv_stats	date	league	season	country	ht	at	fthg	ftag	odd_h	odd_d	odd_a	odd_over	odd_unde	odd_bts	odd_bts_n
UFot0hIt/	/soccer/germany/b	TRUE	05/08/2011	D1	2012	germany	Borussia D	Hamburg S	3	1	1.56	4.41	7.42	NA	NA	NA	NA
Aw5DfILH/	/soccer/germany/b	TRUE	06/08/2011	D1	2012	germany	FC Augsbu	SC Freibur	2	2	2.36	3.6	3.4	NA	NA	NA	NA
bkjpaC6n/	/soccer/germany/b	TRUE	06/08/2011	D1	2012	germany	Werder Br	Kaiserslau	2	0	1.83	4.2	4.8	NA	NA	NA	NA
CzPV312a/	/soccer/france/ligu	TRUE	06/08/2011	F1	2012	france	Paris Sain	Lorient	0	1	1.55	4.5	9.4	NA	NA	NA	NA
GUOdmtIl/	/soccer/france/ligu	TRUE	06/08/2011	F1	2012	france	Caen	Valencien	1	0	2.5	3.4	3.45	NA	NA	NA	NA
IOpzwMkp/	/soccer/germany/b	TRUE	06/08/2011	D1	2012	germany	Hertha Be	Nurnberg	0	1	2.06	3.75	3.95	NA	NA	NA	NA
M7PhlM2C/	/soccer/france/ligu	TRUE	06/08/2011	F1	2012	france	Brest	Evian Tho	2	2	2.29	3.25	3.85	NA	NA	NA	NA
QuWqjrYa/	/soccer/france/ligu	TRUE	06/08/2011	F1	2012	france	AC Ajaccio	Toulouse	0	2	2.8	3.1	3.05	NA	NA	NA	NA
UBZQ4smg/	/soccer/france/ligu	TRUE	06/08/2011	F1	2012	france	Nice	Lyon	1	3	4.5	3.55	2	NA	NA	NA	NA
Wn69eU5B/	/soccer/germany/b	TRUE	06/08/2011	D1	2012	germany	FC Cologn	VfL Wolfsl	0	3	3	3.8	2.54	NA	NA	NA	NA
WQGNSNXn/	/soccer/france/ligu	TRUE	06/08/2011	F1	2012	france	AS Nancy	Lille	1	1	3.66	3.5	2.42	NA	NA	NA	NA
x4HJ63It/	/soccer/france/ligu	TRUE	06/08/2011	F1	2012	france	Montpellli	AJ Auxerr	3	1	2.12	3.3	4.44	NA	NA	NA	NA
xQ65dAJ5/	/soccer/germany/b	TRUE	06/08/2011	D1	2012	germany	VfB Stuttg	Schalke 04	3	0	2.31	3.8	3.27	NA	NA	NA	NA
xtklbWLh/	/soccer/germany/b	TRUE	06/08/2011	D1	2012	germany	Hannover	TSG Hoffe	2	1	2.43	3.75	3.4	NA	NA	NA	NA
zeViBqAP/	/soccer/france/ligu	TRUE	06/08/2011	F1	2012	france	Marseille	Sochaux	2	2	1.5	4.4	9	NA	NA	NA	NA
84Llk2m6/	/soccer/france/ligu	TRUE	07/08/2011	F1	2012	france	Bordeaux	St Etienne	1	2	1.82	3.65	5.5	NA	NA	NA	NA
fLN0n0XO/	/soccer/france/ligu	TRUE	07/08/2011	F1	2012	france	Dijon FCO	Stade Ren	1	5	3.35	3.27	2.5	NA	NA	NA	NA
K4vgcjya/	/soccer/germany/b	TRUE	07/08/2011	D1	2012	germany	Mainz	Bayer Lev	2	0	3.45	3.4	2.5	NA	NA	NA	NA
SEqvxt5j/	/soccer/germany/b	TRUE	07/08/2011	D1	2012	germany	Bayern M	Borussia M	0	1	1.24	7.4	15	NA	NA	NA	NA
4nHmUALb/	/soccer/england/pr	FALSE	13/08/2011	E0	2012	england	Fulham	Aston Vill	0	0	1.95	3.65	4.51	NA	NA	NA	NA
4WY4SljB/	/soccer/england/pr	FALSE	13/08/2011	E0	2012	england	Newcastle	Arsenal	0	0	3.6	3.55	2.25	NA	NA	NA	NA
bgY8R8SH/	/soccer/england/pr	FALSE	13/08/2011	E0	2012	england	QPR	Bolton	0	4	2.2	3.35	4.09	NA	NA	NA	NA

After deciding on using the betting odds dataset I had to see what I could use for my dataset. Initially, it was quite a large dataset with eighteen columns (as per above). It contained a lot of useful and detailed information on the matches and access to further information through the 'link_odsp' column. However, I felt a lot of this was unnecessary and in some cases the data was redundant. There were six columns of betting odds in the table but even though this was what I was basing my project off I felt I needed to cut three of columns. The final three columns contained over 90pc of NA's so I felt this would only disrupt my star schema and queries in the future, this was despite them being quite interesting columns with potential for very interesting OLAP queries. I also felt the need to cut the aforementioned 'link_odsp' column and 'adv_stats' column too. Using Jupyter notebook, I used the 'df.drop' function to remove these columns. These columns contained very little information I actually needed in my dataset and I felt needed to utilise external links on each row for more information would be a nuisance. Finally, I adjusted the unique, 'match_id' column. I felt by adjusting these values to wholly numeric values it would make it easier to use this column as a primary key in one of my dimensions. I used a simple while loop like this to change the values in the column:

```
i = 0

while i < len(df):

        df.at[i, 'match_id'] = 880 + i

        i += 1
```

I also found some irregularities in my data. In the league column all the leagues were referred to by (country name)1, but the English league was E0, something I adjusted to 'E1' in Jupyter notebook. I also added in a 'results' column to make it easier to find out the result instead of checking home goals and away goals columns. After I finished this I felt I was ready to start creating my star schema in MySQL Workbench.

Creating My Star Schema

After cleaning the data, I had to work out how I would operate my star schema. With the betting odds being what I was basing my project on I felt all odds related columns would be in my facts table. This meant I would have to

create appropriate dimensions to accommodate the odds table. I decided I would create a four-dimension star schema with my dimensions being – 'away', 'home', 'location' and 'result'. Away would contain the away team and away goals scored in a specific game. The home table would be the same except for home teams. My location table would contain 'match_id', 'season', 'country' and 'league' with match ID as the primary key. However, as I tried to create the other dimensions I felt I needed integer based values as unique identifiers in the other tables. This led me to using Jupyter notebook again and adding in new columns to these tables to help create primary keys such as below.

```
In [12]: home = df[['fthg', 'ht']]

In [13]: home.head()
```

Out[13]:

	fthg	ht
0	3	Borussia Dortmund
1	2	FC Augsburg
2	1	Bordeaux
3	0	QPR
4	0	Lille

```
In [14]: home['home_id'] = 1
```

```
C:\program files\Anaconda3\lib\site-packages\ipykernel_l
A value is trying to be set on a copy of a slice from a
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pyda
  """Entry point for launching an IPython kernel.
```

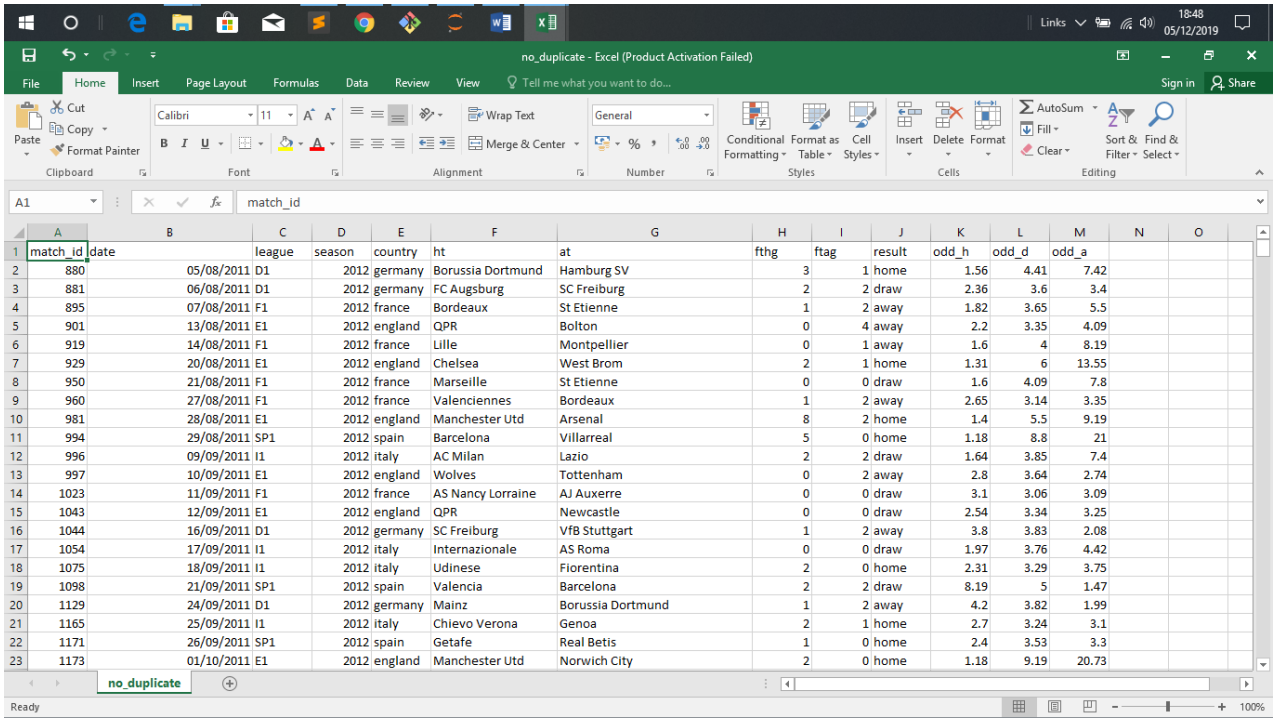
```
In [15]: i = 0
         while i < len(home):
             home.at[i, 'home_id'] = i
             i += 1

In [16]: home.head()
```

Out[16]:

	fthg	ht	home_id
0	3	Borussia Dortmund	0
1	2	FC Augsburg	1
2	1	Bordeaux	2
3	0	QPR	3
4	0	Lille	4

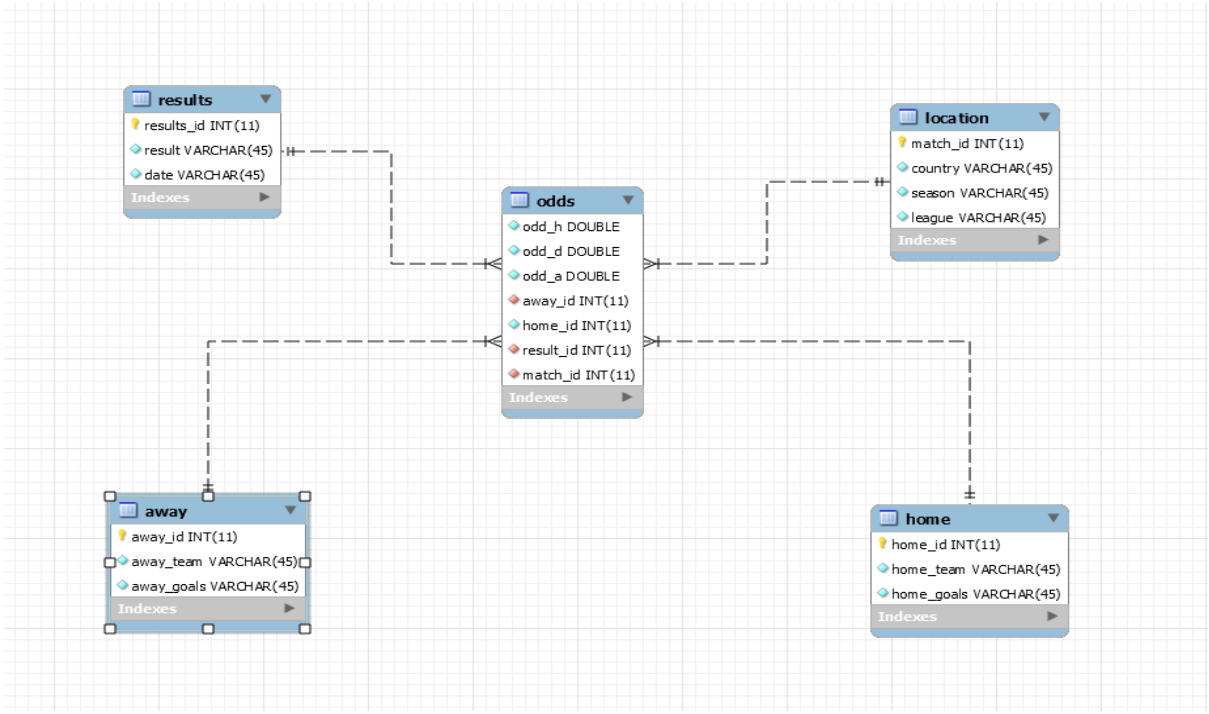
```
In [17]: results = df[['result', 'date']]
```

After doing this for away, home and results I felt I was ready to create my star schema. I saved all these separate dataframe's as CSV files for loading into MySql Workbench. This was after me cutting down the length of my dataset initially. I tried to use dates as a primary key so I cut my dataset from over ten thousand rows to five hundred and twenty-four rows by deleting the duplicate date rows and keeping the first of each date. Although not planned initially, this would reduce the time loading in the dataset to MySql. This is how it looked with all duplicates removed.



match_id	date	league	season	country	ht	at	fthg	ftag	result	odd_h	odd_d	odd_a
880	05/08/2011	D1	2012	germany	Borussia Dortmund	Hamburg SV	3	1	home	1.56	4.41	7.42
881	06/08/2011	D1	2012	germany	FC Augsburg	SC Freiburg	2	2	draw	2.36	3.6	3.4
895	07/08/2011	F1	2012	france	Bordeaux	St Etienne	1	2	away	1.82	3.65	5.5
901	13/08/2011	E1	2012	england	QPR	Bolton	0	4	away	2.2	3.35	4.09
919	14/08/2011	F1	2012	france	Lille	Montpellier	0	1	away	1.6	4	8.19
929	20/08/2011	E1	2012	england	Chelsea	West Brom	2	1	home	1.31	6	13.55
950	21/08/2011	F1	2012	france	Marseille	St Etienne	0	0	draw	1.6	4.09	7.8
960	27/08/2011	F1	2012	france	Valenciennes	Bordeaux	1	2	away	2.65	3.14	3.35
981	28/08/2011	E1	2012	england	Manchester Utd	Arsenal	8	2	home	1.4	5.5	9.19
994	29/08/2011	SP1	2012	spain	Barcelona	Villarreal	5	0	home	1.18	8.8	21
996	09/09/2011	I1	2012	italy	AC Milan	Lazio	2	2	draw	1.64	3.85	7.4
997	10/09/2011	E1	2012	england	Wolves	Tottenham	0	2	away	2.8	3.64	2.74
1023	11/09/2011	F1	2012	france	AS Nancy Lorraine	AJ Auxerre	0	0	draw	3.1	3.06	3.09
1043	12/09/2011	E1	2012	england	QPR	Newcastle	0	0	draw	2.54	3.34	3.25
1044	16/09/2011	D1	2012	germany	SC Freiburg	VfB Stuttgart	1	2	away	3.8	3.83	2.08
1054	17/09/2011	I1	2012	italy	Internazionale	AS Roma	0	0	draw	1.97	3.76	4.42
1075	18/09/2011	I1	2012	italy	Udinese	Fiorentina	2	0	home	2.31	3.29	3.75
1098	21/09/2011	SP1	2012	spain	Valencia	Barcelona	2	2	draw	8.19	5	1.47
1129	24/09/2011	D1	2012	germany	Mainz	Borussia Dortmund	1	2	away	4.2	3.82	1.99
1165	25/09/2011	I1	2012	italy	Chievo Verona	Genoa	2	1	home	2.7	3.24	3.1
1171	26/09/2011	SP1	2012	spain	Getafe	Real Betis	1	0	home	2.4	3.53	3.3
1173	01/10/2011	E1	2012	england	Manchester Utd	Norwich City	2	0	home	1.18	9.19	20.73

I started creating my star schema by creating the tables in the project schema. I decided not to use code to create these tables and instead created them manually. I imported my data in the broken up files using the “import data wizard” and from there I reverse engineered the schema into a model then. After reverse engineering the schema I had to edit the odds table (my fact table) to create the foreign keys. I had to reference the other tables and specific columns in both the fact table and dimensions.



After completing this my model looked like the model above. To finally create my star schema, I would have to "forward engineer" this model. This would provide me the necessary code to create my star schema and allow me to start coding my OLAP queries. There were plenty of other methods I could have used to generate my star schema, but I felt this was the easiest way to go about it.

Specifying my OLAP Queries

After successfully creating my star schema I had to go about coding my OLAP queries. Firstly, I had to come up with two things of interest to me in my dataset. This obviously brought me back to my motives from originally picking the dataset. One of them was to do with how high the odds of draws are within football. Stereotypically, draw odds are a lot higher than home or away wins and I wanted to see if draws were as uncommon as the odds suggest. I also wanted to see how often "unpredictable" wins were i.e. when the opposing team won despite being underdogs. This was certainly of interest to me as I wanted to see does league's like the English Premier League really justify its tag of unpredictability and in contrast the French 'Ligue 1' being labelled predictable and boring. After deciding on investigating the odds of draws and unpredicted home/away wins I had to set about coding my OLAP queries.

Coding the Queries

To start coding the queries I felt I needed to create some basic ones to help me have an understanding of the problem at hand. Firstly, for my unpredictable wins query I coded a basic query to help find the total number of predicted home wins like so:

```
(select match_id from odds where odd_h < odd_a and odd_h < odd_d);
#350 / 524 rows
```

I did a similar query to find expected away wins which returned 165 rows. I then started to code my proper OLAP query then. To find both home and away "unpredicted" wins I would need to use two separate queries. For my home OLAP query I used a series of joins to create a resultant dataset. I tried to use OLAP functions such as "group by" and "rollup" but due to an error 1064 I was not allowed to utilise them properly. My home query looked like this:

```
select home.home_team, away.away_team, results.result, odds.odd_h, odds.odd_a
from home, away, results, odds
where home.home_id = odds.home_id
and away.away_id = odds.away_id
and results.results_id = odds.result_id
and odds.odd_h > odds.odd_a and results.result = 'home';
```

This result only returned 22 rows and when you look at how often home teams are not favourites (165 times) it shows that only 13.3pc of the team the odds are incorrect. For my away unpredicted wins OLAP query I had similar code:

```
select home.home_team, away.away_team, results.result, odds.odd_h, odds.odd_a
from home, away, results, odds
where home.home_id = odds.home_id
and away.away_id = odds.away_id
and results.results_id = odds.result_id
and odds.odd_h < odds.odd_a and results.result = 'away';
```

This returned 47 rows and when I divided it by 350(the total number of predicted home wins) it brought a very similar ratio of unexpected results, 13.4pc. When I looked at my resultant datasets I found a lot of these were indeed found in the English Premier League, while there was also a lot of unexpected results in the Spanish La Liga, a league regarded as predictable. However, when I delved into the results these unpredicted results were not competing at the very top of their respective leagues, something that could put a bit of doubt in my opinion on the “unpredictable” Premier League myth we are always reminded of in the media. From doing these queries I could definitely see how clever and how well set up the odds were by the bookies, with very little of the resultant odds over 10 and as result the bookies could not really lose on pay outs.

I then set about making my OLAP query for my how many draws question. This would be easier and take less external scripts to complete in contrast to the previous two queries. I would simply have to find the number of draws in my schema and see do they stack up to what studies show of how often draws occur. I researched online and roughly between 22pc to 28pc of games end in draws in major leagues across Europe. In my schema only 9 games were expected to finish in draws as this query suggested:

```
SELECT results.result, location.league, odds.odd_d, odds.odd_h, odds.odd_a
FROM odds, results, location
WHERE results.results_id = odds.result_id
and location.match_id = odds.match_id
and odds.odd_d < odds.odd_a and odds.odd_d < odds.odd_h;
```

I then ran my OLAP query to find the total number of draws in my schema:

```
SELECT results.result, location.league, odds.odd_d, odds.odd_h, odds.odd_a
FROM odds, results, location
WHERE results.results_id = odds.result_id
and location.match_id = odds.match_id
and results.result = 'draw';
```

Using a series of joins once again as part of the OLAP query I found that 98 games ended in draws in my dataset. When I did the maths I found 18.7pc of games ended in draws, at least 5pc below what the stats tell you. I felt you could look at these findings in a variety of ways. With only 9 games expected to end in draws going off the odds, how are draws not lower odds in betting prices? There is always going to be a favourite off form or place in league et cetera, but in some cases between teams of similar level I found it strange draws are not lower prices. Again, I tried “group by” and “rollup” functions but found them returning NULL values as they summed through the data while in some other instances I tried the SQL administrator did not have the permissions for me to use them.

Conclusions

I felt I had successfully answered my questions from my OLAP queries by the end of the project. I maybe would have chosen another dataset to base the project off that required less cleaning and tidying but this was something that greatly interested me also.

	A	B	C	D	E	F
1		home_team	away_team	result	odd_h	odd_a
2	0	Bordeaux	St Etienne	away	1.82	5.5
3	1	QPR	Bolton	away	2.2	4.09
4	2	Lille	Montpellier	away	1.6	8.19
5	3	Valenciennes	Bordeaux	away	2.65	3.35
6	4	Novara	Bologna	away	1.94	4.8
7	5	Athletic Bilbao	Granada	away	1.39	10.76
8	6	Bayer Leverkusen	Numberg	away	1.62	6.75
9	7	Manchester Utd	Blackburn	away	1.15	24.29
10	8	Everton	Bolton	away	1.58	7.53
11	9	Swansea	Norwich City	away	1.85	5.09
12	10	Internazionale	Bologna	away	1.57	8.18
13	11	Arsenal	Wigan	away	1.3	12
14	12	Palermo	Parma	away	2.92	3.31
15	13	Chelsea	Newcastle	away	1.49	8
16	14	AJ Auxerre	Bordeaux	away	2.5	3.3
17	15	AC Milan	Sampdoria	away	1.55	8.5
18	16	Liverpool	Aston Villa	away	1.4	10.26
19	17	Schalke 04	SpVgg Greuther Furt	away	1.39	10.84
20	18	Napoli	Chievo Verona	away	1.45	11.59
21	19	Borussia Dortmund	Hannover 96	away	1.33	16
22	20	Barcelona	Celta Vigo	away	1.2	27

Head of unpredicted home and away wins resultant dataset

45	43	Montpellier	GFC Ajaccio	away	1.93	6.32
46	44	Athletic Bilbao	Real Sociedad	away	1.81	5.51
47	45	Bayern Munich	Mainz	away	6	24.43
48	46	Barcelona	Valencia	away	1.22	24.89
49	47	Hannover 96	Bayern Munich	home	9.64	1.43
50	48	Sunderland	Manchester City	home	6.2	1.71
51	49	AC Ajaccio	Marseille	home	4.42	2.06
52	50	QPR	Arsenal	home	6.75	1.63
53	51	Hamburg SV	Borussia Dortmund	home	7.2	1.57
54	52	Borussia Monchengl	VfL Wolfsburg	home	2.88	2.64
55	53	Bolton	Liverpool	home	8.01	1.57
56	54	Genoa	Napoli	home	3.61	2.25
57	55	Granada	Malaga	home	3.02	2.61
58	56	Atalanta	AS Roma	home	3.41	2.5
59	57	Udinese	Fiorentina	home	3.08	2.6
60	58	Real Valladolid	Villarreal	home	4.6	2.03
61	59	Catania	Lazio	home	4.02	2.33
62	60	West Ham	Southampton	home	3.82	2.5
63	61	Real Valladolid	Barcelona	home	17.78	1.25
64	62	Borussia Monchengl	Borussia Dortmund	home	3.18	2.57
65	63	Sunderland	Southampton	home	4.23	2.15
66	64	Sevilla	Barcelona	home	4.82	1.91
67	65	Hamburg SV	Borussia Dortmund	home	10.24	1.43
68	66	Celta Vigo	Barcelona	home	8.52	1.45
69	67	Fulham	Arsenal	home	4.59	1.98
70	68	Granada	Athletic Bilbao	home	3.3	2.55

End of unpredicted home/away win's dataset (above)

	result	league	odd_d	odd_h	odd_a
2	draw	D1	3.6	2.36	3.4
3	draw	F1	4.09	1.6	7.8
4	draw	I1	3.85	1.64	7.4
5	draw	F1	3.06	3.1	3.09
6	draw	E1	3.34	2.54	3.25
7	draw	I1	3.76	1.97	4.42
8	draw	SP1	5	8.19	1.47
9	draw	SP1	9.74	1.14	26
10	draw	F1	3.22	4.09	2.22
11	draw	I1	4.54	8.6	1.51
12	draw	F1	3.87	1.7	7.76
13	draw	D1	4.62	1.58	7.14
14	draw	I1	2.85	2.46	4.11
15	draw	SP1	5.38	1.42	11.3
16	draw	I1	6.43	1.27	15.72
17	draw	E1	3.94	4.93	1.85
18	draw	E1	5.46	1.4	10.48
19	draw	SP1	4.32	1.66	6
20	draw	D1	8.56	1.2	18.34
21	draw	SP1	13.32	1.11	31
22	draw	F1	9.85	1.14	28.28
23	draw	I1	9.07	1.18	26.59
24	draw	I1	2.25	2.85	5.09
25	draw	I1	2.02	3.4	5.9
26	draw	F1	4.35	7.4	1.7

A look at resultant draws dataset