

REPORT

Sports Analytics Assignment

(OR Applications in Infrastructural and Service Sectors)

BY:

Shubham Joshi (23M1531)

Introduction:

Bundesliga 2 is a German football league with 18 teams. In the league, each team plays each other twice, home and away, leading to 34 games for each team, played on 34 "match days".

We have come up with a method to show which three teams are going to be at the top of the list at the end of this regular session (i.e. 19 May 2024). Using our devised method we will try to predict the top three teams in Bundesliga 2 as of 5 May that is upto round 32.

Methodology:

1. **Data Scraping:** We had a lot of data available in the site of Bundesliga 2 but the main problem was that it was not available in file or other format. So we did data scraping from the site <https://fbref.com> . We created a list of data ranging from 2019 to 2024. Then for this purpose, we use the Python language and the libraries used are 'requests', 'BeautifulSoup', and 'pandas'. The link for the collab file is provided below.

2. **Data Cleaning:** The data that we scraped from the site was not at all usable because it had many errors and null values between different cells. So, before predicting the value of the data we cleaned the data. For cleaning purposes, we used the Python language in which libraries like numpy and pandas are used.

3. **Model Used:** The main problem that arose was how to predict the result we wanted. So, we decided to go for a machine-learning model. We used a random forest classification technique to predict the result. The main steps of doing this are

1. Initializing the random forest classifier with the help of the Scikit learn library in Python. Random Forest is a versatile machine-learning algorithm that can be applied to various tasks, including multiclass classification. In multiclass classification (e.g. Win, Draw, Loss), the goal is to predict one of several classes for each instance.
2. Fitting the training dataset which we scraped using the scraping technique in Python. The features used for prediction are ["venue_code", "opp_code", "hour", "day_code"], where

Venue code: The venue at which the game is played. It is Home or Away (used as 0 and 1 for model preparation). This is surely a good attribute because the team playing at home is having an edge over the opponent team.

opp_code: This is the code for opponents. Each team has a particular code so that it becomes easy for model preparation. This is a good attribute to consider because it matters a lot which team is playing against which team.

hour: This is the time at which the game is played. This also contributes to match results because of the factors of time at which the game is played like dew, environment, etc.

day_code: This is the attribute for the day of the week. Like Monday is given 1, Tuesday 2, and so on.

There is one more reason to select these features, it is the availability of both training data and the test data. Test data means the data for which the prediction is to be done. So when we tested the effectiveness of our model we got the following result.

Here -1,0,1 denote Loss, Draw,Win respectively.

predicted	-1	0	1
actual			
-1	8	15	9
0	9	22	16
1	6	18	23


Accuracy: 0.42063492063492064
Precision: 0.41627990568207957

3. After getting the model trained we then move to test it in the data for future goal prediction. We used a fixture table for this purpose since we had the above attributes in that table till the last game. Then according to our test data, we got the predicted outcome.

4. The next and last step was to make a table showing the points of each team after all the matches were over (i.e. 32 games are over). So we again used Python for that. We define a user-defined function for updating the score table in which based on our prediction we give different points to the home team winning or losing and the away team winning or losing (The code file is attached below).

Data Files and Collab File :

Current Scorecard:  [score_table](#)

Collab File used for web scraping:  [scraping.ipynb](#)

Previous 5 Year Record of Bundesliga 2 Participants using Web scraping:  [matches](#)

Collab File used for for training and prediction:  [Actual_Working_Model.ipynb](#)

Fixture file containing upcoming match details:  [fixtures](#)

Results:

So this is the Snapshot of the teams we got at the end of round 32:

Rank	Club	Pld	W	D	L	Pts
1	St. Pauli	32	16	13	3	61
2	Holstein Kiel	32	13	12	7	51
3	Hamburger SV	32	12	10	10	46
4	Hannover 96	32	11	14	7	47
5	Greuther Fürth	32	11	13	8	46
6	Paderborn 07	32	12	10	10	46
7	Düsseldorf	32	11	14	7	47
8	Karlsruher	32	14	11	7	53
9	Hertha BSC	32	12	11	9	47
10	Nürnberg	32	10	12	10	42
11	Elversberg	32	10	10	12	40
12	Magdeburg	32	8	12	12	36
13	Wehen	32	9	12	11	39
14	Schalke 04	32	9	6	17	33
15	Kaiserslautern	32	9	10	13	37
16	Braunschweig	32	8	9	15	33
17	Hansa Rostock	32	7	10	15	31
18	Osnabrück	32	4	16	12	28

So according to our model the top three teams are :

St. Pauli (1st position),

Karlsruher (2nd Position),

Holstein Kiel (3rd Position).

Scope for Improvement:

Our model currently incorporates only a limited set of factors to predict outcomes, which may not be comprehensive enough for accurate predictions. The primary challenge lies in acquiring and preprocessing data, which demands a strong grasp of machine learning techniques. Despite these limitations, our model exhibits potential for yielding significantly improved predictions.

To enhance the predictive accuracy, we should consider incorporating a broader range of factors. These could include metrics such as goals scored, goals conceded, total shots, shots on target, shooting accuracy, goals per shot ratio, average shot distance, free kick attempts, penalty kicks conversion rate, expected goals, and non-penalty expected goals. By integrating these additional parameters into our model, we can anticipate a notable enhancement in prediction accuracy.