TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO HỌC THUẬT

PHÂN TÍCH DỮ LIỆU SỰ KIỆN BẰNG PYTHON

PHẠM AN CƯƠNG

HÀ NỘI, 6-2024

MỤC LỤC

MỞ ĐẦU

Khoa học dữ liệu là một cách để nắm bắt những hiểu biết sâu sắc từ dữ liệu. Khoa học dữ liệu đang tác động đến rất nhiều lĩnh vực, bao gồm cả bóng đá.

Bóng đá, là một trong những môn thể thao phổ biến nhất và được theo dõi rộng rãi trên toàn cầu. Trong những năm gần đây, sự sẵn có của lượng lớn dữ liệu bóng đá đã mở ra những khả năng thú vị cho việc phân tích và hiểu biết dựa trên dữ liệu. Python, là một ngôn ngữ lập trình linh hoạt và mạnh mẽ, là một lựa chọn tuyệt vời cho người mới bắt đầu cũng như các chuyên gia để thực hiện phân tích dữ liệu bóng đá.

Bóng đá chứa rất nhiều dữ liệu cần bao quát, từ khía cạnh cá nhân đến khía cạnh đồng đội. Với dữ liệu, chúng ta có thể hiểu trò chơi theo cách có ý nghĩa hơn. Phạm vi phân tích dữ liệu trong bóng đá bao gồm nhiều chủ đề bao gồm phân tích sự kiện trong trận đấu, tìm kiếm cầu thủ cho các đội, ra quyết định chiến thuật và trực quan hóa chiến thuật.

Ngoài ra, đối với các nhóm, dữ liệu có thể tạo ra những hiểu biết sâu sắc giúp đưa ra quyết định. Vì vậy, đội có thể tìm ra chiến lược để giành chiến thắng trong trận đấu.

1. PHÂN TÍCH DỮ LIỆU

1.1 Lấy dữ liệu

Đối với dữ liệu, chúng tôi sẽ sử dụng dữ liệu từ StatsBomb. StatsBomb là một công ty phân tích hoạt động cụ thể trong lĩnh vực bóng đá, cung cấp rất nhiều dữ liệu bóng đá, đặc biệt là dữ liệu sự kiện.
Đối với những người muốn tìm hiểu phân tích bóng đá, StatsBomb đã xuất bản dữ liệu mở. Dữ liệu bao gồm các trận đấu giải bóng đá đã kết thúc.

1.2 Khám phá dữ liệu

Sau khi bạn tải xuống dữ liệu, bước tiếp theo là khám phá nó. Cấu trúc thư mục của dữ liệu sẽ như sau:

```
|   LICENSE.pdf
|   README.md
|
+---data
|   |   competitions.json
|   |
|   +---events
|   |       15946.json
|   |       15956.json
|   |       15973.json
|   |       15978.json
|   |       15986.json
|   |
|   +---lineups
|   |       15946.json
|   |       15956.json
|   |       15973.json
|   |       15978.json
|   |       15986.json
|   |
|   \---matches
|       +---11
|       |       1.json
|       |       2.json
|       |
|       +---16
|       |       1.json
|       |       2.json
|       |
|
+---doc
|       Open Data Competitions v2.0.0.pdf
|       Open Data Events v4.0.0.pdf
|       Open Data Lineups v2.0.0.pdf
|       Open Data Matches v3.0.0.pdf
|       StatsBomb Open Data Specification v1.1.pdf
|
\---img
        statsbomb-logo.jpg
```

Ngoài ra còn có các thư mục như sự kiện, đội hình và trận đấu.
• Thư mục sự kiện chứa các tệp tóm tắt trận đấu ở định dạng JSON.
• Thư mục đội hình chứa đội hình của mỗi đội trong mỗi trận đấu.

• Thư mục trận đấu chứa các trận đấu mà mỗi cuộc thi có. Nó cũng được chia thành các mùa khác nhau cho một số trận đấu.

1.3 Truy xuất dữ liệu sự kiện

Dữ liệu sự kiện có thể được truy xuất bằng các bước này. Đầu tiên, chúng ta mở tệp competitions.json. Tệp này là cổng đầu tiên để truy cập dữ liệu StatsBomb. Lý do làm điều đó là vì chúng tôi cần giải đấu và ID mùa giải để truy cập danh sách các trận đấu từ giải đấu đó.
Để xử lý tệp JSON, thư viện pandas đã cung cấp chức năng đọc tệp JSON dưới dạng khung dữ liệu bằng cách sử dụng hàm read_json. Đây là mã để làm điều đó:

```
import pandas as pd
competition = pd.read_json('open-data/data/competitions.json')
competition.head()
```

Bây giờ bạn có thể thấy các hàng chứa thông tin về tất cả các trận đấu mà StatsBomb đã cung cấp.
Tóm lại, các giải đấu được đưa vào dữ liệu này là La Liga (Giải VĐQG Tây Ban Nha), UEFA Euro, FIFA World Cup (Nam và Nữ) và UEFA Champions League. Hiện tại, chúng tôi muốn lấy hàng có thông tin FIFA World Cup. Hãy lọc tập dữ liệu bằng cách sử dụng dòng mã này:

```
# Get the FIFA World Cup

competition[competition.competition_name == 'FIFA World Cup']
```

FIFA World Cup lần lượt là 43 và 3. Bây giờ hãy truy cập vào thư mục chứa ID. Đối với mỗi trận đấu, thư mục được đặt tên bằng ID trận đấu. Và mỗi thư mục chứa tệp JSON. Mỗi tệp được đính kèm với ID phần làm tên.
Bây giờ hãy truy cập tệp bằng cách sử dụng các dòng mã sau:

```
import json


with open('open-data/data/matches/43/3.json') as f:
    data = json.load(f)
data
with open('open-data/data/matches/43/3.json') as f:
    data = json.load(f)
    for i in data:

        print('ID:', i['match_id'], i['home_team']['home_team_name'], i['home_score'], '-',

i['away_score'], i['away_team']['away_team_name'])
with open('open-data/data/events/7567.json') as f:
    korger = json.load(f)

korger
```

Để dễ dàng phân tích, thư viện pandas đã cung cấp hàm json_normalize. Điều làm cho hàm này trở nên mạnh mẽ là vì nó có thể xử lý JSON lồng nhau. Bây giờ hãy viết những dòng mã này:

```
# from pandas.io.json import json_normalize

df = pd.json_normalize(korger, sep='_').assign(match_id="7567")
df.head()
```

2. TRỰC QUAN HÓA DỮ LIỆU

2.1 Thu thập và tiền xử lý dữ liệu

Bước đầu tiên trong bất kỳ dự án phân tích dữ liệu nào là thu thập dữ liệu. Có nhiều nguồn khác nhau để lấy dữ liệu bóng đá, chẳng hạn như API, cơ sở dữ liệu trực tuyến hoặc thậm chí các bộ dữ liệu được quản lý thủ công. Khi bạn đã có được dữ liệu, việc xử lý trước là rất quan trọng để làm sạch và sắp xếp dữ liệu để phân tích.

```
import pandas as pd

# Load data from CSV file

df = pd.read_csv('football_data.csv')

# Data cleaning and preprocessing

# ... (e.g., handling missing values, data type conversion, etc.)
```

2.2 Khám phá và trực quan hóa dữ liệu

Sau khi xử lý trước dữ liệu, bước tiếp theo là khám phá dữ liệu đó để hiểu rõ hơn và xác định các mẫu. Trực quan hóa đóng một vai trò quan trọng trong việc hiểu dữ liệu và làm cho nó dễ hiểu hơn.

```
import matplotlib.pyplot as plt

# Basic data exploration

print(df.head())

# Plotting goals scored by each team

team_goals = df.groupby('Team')['Goals'].sum()

team_goals.plot(kind='bar', figsize=(10, 6))

plt.title('Total Goals Scored by Each Team')

plt.xlabel('Team')

plt.ylabel('Goals')

plt.show()
```

2.3 Số liệu hiệu suất

Để đánh giá các đội bóng và cầu thủ, bạn sẽ cần các số liệu hiệu suất. Các số liệu như tỷ lệ chuyển đổi mục tiêu, tỷ lệ hoàn thành đường chuyền và độ chính xác của cú sút là rất quan trọng để phân tích hiệu suất dựa trên dữ liệu.

```
# Calculating goal conversion rate

df['Goal Conversion Rate'] = (df['Goals'] / df['Shots']) * 100

# Visualizing goal conversion rate

plt.scatter(df['Goal Conversion Rate'], df['Player'], alpha=0.5)

plt.title('Goal Conversion Rate vs. Player')

plt.xlabel('Goal Conversion Rate (%)')

plt.ylabel('Player')

plt.show()
```

2.4 Phân tích xu hướng

Phân tích xu hướng theo thời gian có thể cung cấp những hiểu biết có giá trị về hiệu suất của một đội qua các mùa giải hoặc trận đấu khác nhau.

```
# Converting 'Date' column to datetime format

df['Date'] = pd.to_datetime(df['Date'])

# Analyzing goals scored over time

goals_over_time = df.groupby('Date')['Goals'].sum()

goals_over_time.plot(figsize=(12, 6))

plt.title('Goals Scored Over Time')

plt.xlabel('Date')

plt.ylabel('Goals')

plt.show()
```

2.5 Tạo bản đồ cú sút

Một trong những hình ảnh trực quan mà chúng ta có thể tạo là bản đồ sút bóng. Trên bản đồ này, chúng tôi muốn xem mỗi đội đã thực hiện được bao nhiêu cú sút. Ngoài ra, chúng tôi muốn biết cơ hội ghi bàn là bao nhiêu. Chúng tôi gọi cơ hội này là những bàn thắng dự kiến.

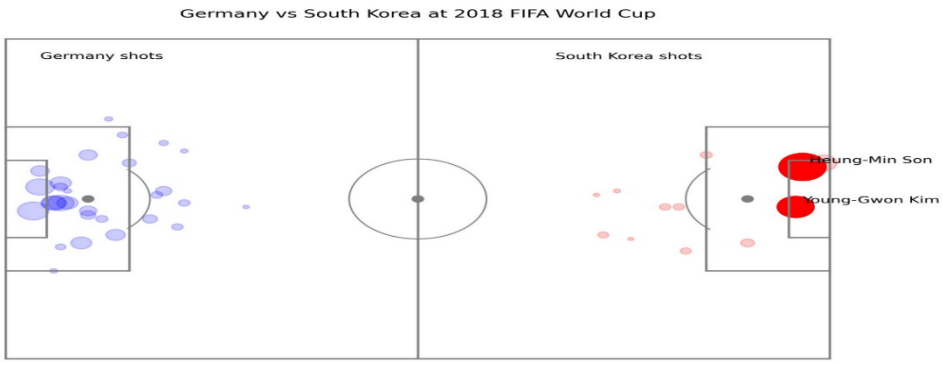
Để tạo trực quan hóa, trước tiên chúng ta cần lấy dữ liệu sự kiện. Sau đó, lọc dữ liệu dựa trên tên sự kiện.

```
shots = df[df.type_name == 'Shot'].set_index('id')

shots.head()
```

Sau khi nhận được dữ liệu, bây giờ hãy viết những dòng mã này để tạo bản đồ:

```
import numpy as np
import matplotlib.pyplot as plt
from FCPython import createPitch
pitch_width = 120
```

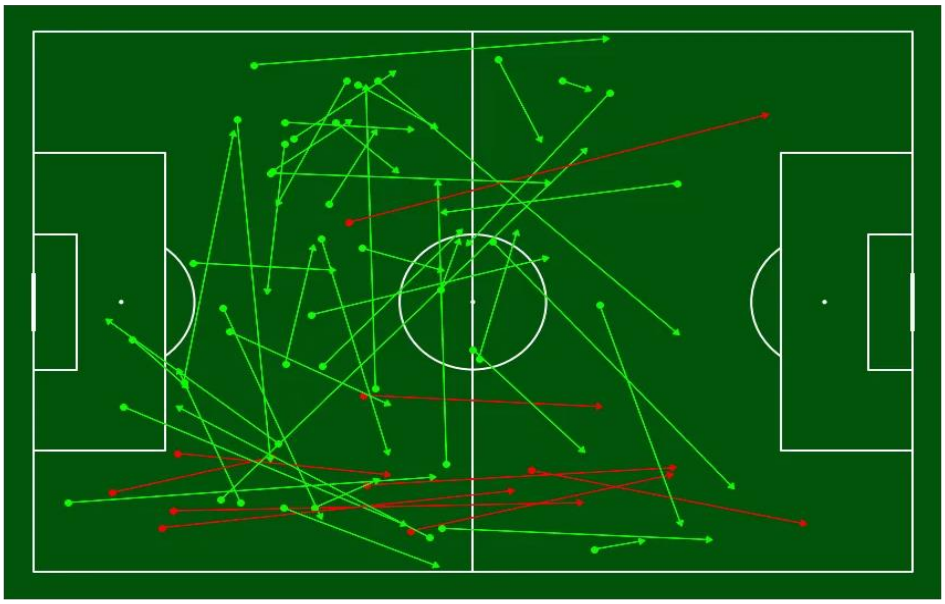
```
pitch_height = 80
fig, ax = createPitch(pitch_width, pitch_height, 'yards', 'gray')
home_team = 'South Korea'
away_team = 'Germany'
for i, shot in shots.iterrows():
    x = shot['location'][0]
    y = shot['location'][1]
    goal = shot['shot_outcome_name']=='Goal'
    team_name = shot['team_name']
    circle_size = 2
    circle_size = np.sqrt(shot['shot_statsbomb_xg'] * 15)
    if team_name == home_team:
        if goal:
            shot_circle = plt.Circle((x, pitch_height-y), circle_size, color='red')
            plt.text((x+1), pitch_height-y+1, shot['player_name'])
        else:
            shot_circle = plt.Circle((x, pitch_height-y), circle_size, color='red')
            shot_circle.set_alpha(.2)
    elif team_name == away_team:
        if goal:
            shot_circle = plt.Circle((pitch_width-x, y), circle_size, color='blue')
            plt.text((pitch_width-x+1), y+1, shot['player_name'])
        else:
            shot_circle = plt.Circle((pitch_width-x, y), circle_size, color='blue')
            shot_circle.set_alpha(.2)
    ax.add_patch(shot_circle)
plt.text(5, 75, away_team + ' shots')
plt.text(80, 75, home_team + ' shots')
plt.title('Germany vs South Korea at 2018 FIFA World Cup')
fig.set_size_inches(10, 7)
fig.savefig('korger_shots.png', dpi=300)

plt.show()
```



2.6 Tạo bản đồ đường chuyền

```
# to take a better look at player pass map
```

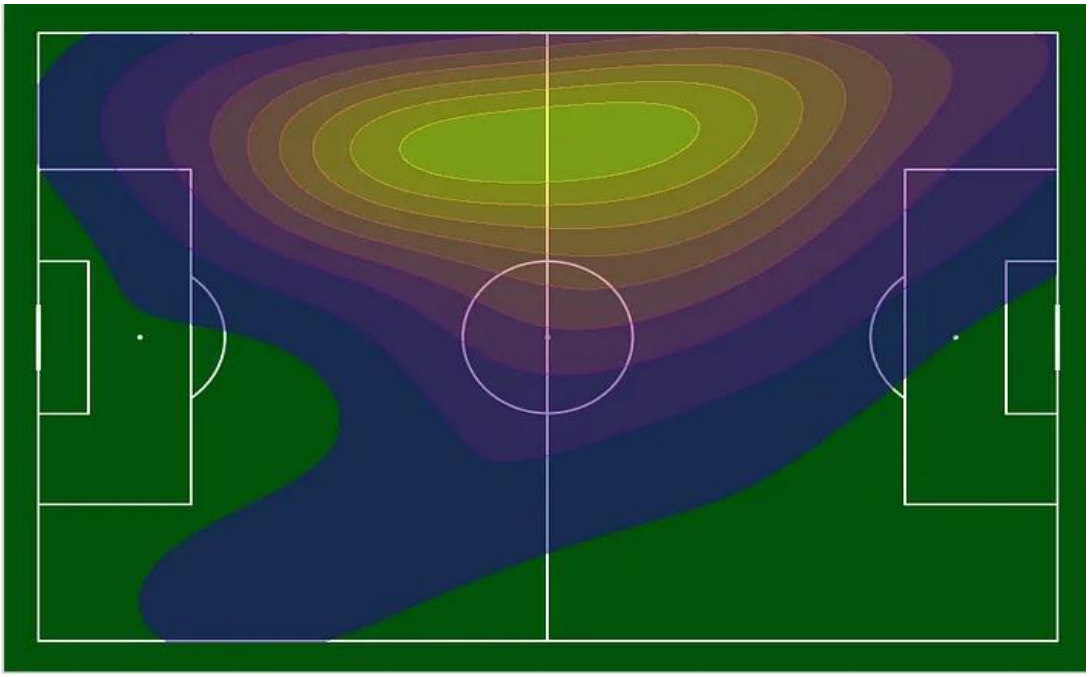
```
def generatePlayerPassMap(player_name):
    player_filter = (df.type_name == 'Pass') & (df.player_name ==
player_name)
    player_df = df.loc[player_filter, ['x', 'y', 'end_x', 'end_y']]

    pitch = Pitch(line_color='white',pitch_color='#02540b')
    fig, ax = pitch.grid(grid_height=0.9, title_height=0.06,
axis=False,endnote_height=0.04, title_space=0, endnote_space=0)
    for i in player_df.index:
        x = player_df['x'][i]
        y = player_df['y'][i]
        dx = player_df['end_x'][i] - player_df['x'][i]
        dy = player_df['end_y'][i] - player_df['y'][i]
        if df['outcome_name'][i] != 'Incomplete':

ax['pitch'].arrow(x,y,dx,dy,color='#0dff00',length_includes_head=True,head_
width=1,head_length=0.8)

pitch.scatter(player_df['x'][i],player_df['y'][i],color='#0dff00',ax=ax['pi
tch'])
        else:

ax['pitch'].arrow(x,y,dx,dy,color='red',length_includes_head=True,head_widt
h=1,head_length=0.8)

pitch.scatter(player_df['x'][i],player_df['y'][i],color='red',ax=ax['pitch'
])
    fig.suptitle(player_name+" passes", fontsize = 20)

# we can do this for all players individually
generatePlayerPassMap("Konsham Chinglensana Singh")
```



2.7 Tạo bản đồ nhiệt

```
def generatePlayerHeatMap(player_name):
    player_filter = (df.type_name == 'Pass') & (df.player_name ==
player_name)
```

```
    player_df = df.loc[player_filter, ['x', 'y', 'end_x', 'end_y']]

    pitch = Pitch(line_color='white',pitch_color='#02540b')
    fig, ax = pitch.grid(grid_height=0.9, title_height=0.06,
axis=False,endnote_height=0.04, title_space=0, endnote_space=0)
    #Create the heatmap
    pitch.kdeplot(
        x=player_df['x'],
        y=player_df['y'],
        shade = True,
        shade_lowest=False,
        alpha=.5,
        n_levels=10,
        cmap = 'plasma',
        ax=ax['pitch']
        )
    fig.suptitle(player_name+" Heatmap", fontsize = 20)
```



KẾT LUẬN

Python là một lựa chọn tuyệt vời cho người mới bắt đầu và các chuyên gia muốn thực hiện phân tích dữ liệu Bóng đá. Với thư viện của Python, bạn có thể thu thập, xử lý trước, khám phá và trực quan hóa dữ liệu bóng đá một cách hiệu quả. Bằng cách tạo các tập lệnh Python cơ bản, bạn có thể có được thông tin chi tiết có giá trị về hiệu suất của người chơi và đội, xác định xu hướng và đưa ra kết luận dựa trên dữ liệu. Tuy nhiên, điều cần thiết là phải thừa nhận những thách thức và hạn chế cũng như nhận thức được các kỹ thuật nâng cao hơn khi bạn tiến bộ trong hành trình phân tích dữ liệu bóng đá của mình.

TÀI LIỆU THAM KHẢO

[1] Germany out of tournament after losing to South Korea. BBC. https://www.bbc.com/sport/football/44439270

[2] StatsBomb: Open Data. https://github.com/statsbomb/open-data