

Báo cáo BTL Python

Đặng Trung Kiên _ B22DCCN426

Ngày 3 tháng 11 năm 2024

1 3.1

Sử dụng thuật toán K-means để phân loại các cầu thủ thành các nhóm có chỉ số giống nhau

1.1 Dữ liệu

Dữ liệu được đọc từ file CSV có tên `results.csv`. Các chỉ số được sử dụng để phân cụm bao gồm:

- **Min**: Số phút thi đấu
- **Age**: Tuổi của cầu thủ
- **Won%**: Tỷ lệ thắng của cầu thủ

1.2 Quy trình

1.2.1 Bước 1: Đọc Dữ liệu

Sử dụng thư viện `pandas` để đọc dữ liệu từ file CSV.

```
import pandas as pd
data = pd.read_csv('results.csv')
```

1.2.2 Bước 2: Tiền xử lý Dữ liệu

Xóa các hàng có giá trị NaN trong các chỉ số được chọn.

```
features = ['Min', 'Age', 'Won%']
data = data.dropna(subset=features)
```

1.2.3 Bước 3: Phân cụm với K-means

Sử dụng thuật toán K-means để phân cụm dữ liệu thành 3 nhóm.

```
from sklearn.cluster import KMeans

num_clusters = 3
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
X = data[features]
data['cluster'] = kmeans.fit_predict(X)
```

1.2.4 Bước 4: Đặt Tên Nhóm

Gán tên cho các nhóm đã phân cụm.

```
group_names = {
    0: 'Nhóm 1: Cầu thủ xuất sắc',
    1: 'Nhóm 2: Cầu thủ trung bình',
    2: 'Nhóm 3: Cầu thủ kém'
}
data['group_name'] = data['cluster'].map(group_names)
```

1.2.5 Bước 5: Tính Toán Giá Trị Trung Bình

Tính giá trị trung bình cho mỗi nhóm và lưu kết quả.

```
grouped_data = data.groupby('group_name')[features].mean().reset_index()
sorted_grouped_data = grouped_data.sort_values(by='goals', ascending=False)
```

1.2.6 Bước 6: Lưu Kết Quả

Lưu dữ liệu đã phân cụm và dữ liệu nhóm trung bình vào các file CSV.

```
data.to_csv('results3.csv', index=False)
sorted_grouped_data.to_csv('sorted_groups.csv', index=False)
```

2 3.3

Viết chương trình python vẽ biểu đồ rada (radar chart) so sánh cầu thủ với đầu vào như sau: + python radarChartPlot.py -p1 <player Name 1> -p2 <player Name 2> -Attribute

2.1 Dữ liệu

Dữ liệu được đọc từ file CSV có tên **results.csv**, với định dạng sử dụng dấu chấm phẩy (;) làm dấu phân cách. Các chỉ số cầu thủ cần so sánh sẽ được chỉ định qua tham số dòng lệnh.

2.2 Quy trình

2.2.1 Bước 1: Nhập Thư viện

Các thư viện cần thiết được nhập vào:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import argparse
```

2.2.2 Bước 2: Định nghĩa Hàm vẽ Biểu đồ Radar

Hàm **radar_chart** nhận vào các thống kê của hai cầu thủ và danh sách các thuộc tính để vẽ biểu đồ.

```
def radar_chart(player1_stats, player2_stats, attributes):
    num_vars = len(attributes)
    angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tolist()

    player1_stats = np.concatenate((player1_stats, [player1_stats[0]]))
    player2_stats = np.concatenate((player2_stats, [player2_stats[0]]))
    angles += angles[:1]
```

```

fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(polar=True))
ax.fill(angles, player1_stats, color='blue', alpha=0.25, label='Player 1')
ax.fill(angles, player2_stats, color='red', alpha=0.25, label='Player 2')
ax.set_yticklabels([])
ax.set_xticks(angles[:-1])
ax.set_xticklabels(attributes)
plt.legend(loc='upper right')
plt.title('So sánh cầu thủ')
plt.show()

```

2.2.3 Bước 3: Nhận Tham số Dòng lệnh

Hàm main sử dụng argparse để nhận vào tên của hai cầu thủ và danh sách các chỉ số cần so sánh.

```

def main():
    parser = argparse.ArgumentParser(description='Vẽ biểu đồ radar so sánh cầu thủ.')
    parser.add_argument('--p1', required=True, help='Tên cầu thủ thứ nhất')
    parser.add_argument('--p2', required=True, help='Tên cầu thủ thứ hai')
    parser.add_argument('--Attribute', required=True, help='Danh sách các chỉ số cần so sánh, cách nhau')
    args = parser.parse_args()

```

2.2.4 Bước 4: Kiểm tra và Lấy Thông tin Cầu thủ

Kiểm tra xem các cầu thủ có tồn tại trong dữ liệu không, và lấy thông tin chỉ số của họ.

```

data = pd.read_csv('results.csv', sep=';')

player1 = args.p1
player2 = args.p2
attributes = args.Attribute.split(',')

if player1 not in data['Player'].values or player2 not in data['Player'].values:
    print(f"Một hoặc cả hai cầu thủ '{player1}' và '{player2}' không tồn tại trong dữ liệu.")
    return

player1_stats = data.loc[data['Player'] == player1, attributes].values.flatten()
player2_stats = data.loc[data['Player'] == player2, attributes].values.flatten()

```

2.2.5 Bước 5: Vẽ Biểu đồ Radar

Cuối cùng, gọi hàm radar_chart để vẽ biểu đồ radar so sánh hai cầu thủ.

```

radar_chart(player1_stats, player2_stats, attributes)

```

2.3 Bước 6: Chạy Chương trình

Chương trình được chạy thông qua lệnh sau:

```

if __name__ == '__main__':
    main()

```