# ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
# TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

# BÁO CÁO ĐỒ ÁN
# KHAI THÁC DỮ LIỆU

# ĐỀ TÀI:
# PHÂN TÍCH DỮ LIỆU VỀ FIFA WORLD CUP

**GVHD: Phạm Nguyễn Thanh Bình**

**Lớp: IS252.O21**


**Sinh viên thực hiện:**

**20521158 - Nguyễn Hải Đăng**

**21521943 - Nguyễn Tiến Đạt**

**21521994 - Lê Anh Duy**


**TP. HCM, ngày 24 tháng 5 năm 2024**

## Mục lục

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# CHƯƠNG 1. NHẬN DIỆN BÀI TOÁN KHAI THÁC DỮ LIỆU

## 1. Tổng quan về dữ liệu

- Bộ dữ liệu về FIFA World Cup là dữ liệu đầy đủ về những trận bóng quốc tế tổ chức ở các quốc gia khác nhau. Bộ dữ liệu bao gồm các trận bóng được thống kê từ năm 1993 đến năm 2022. Bộ dữ liệu có các thuộc tính như thời gian, đội chủ nhà, đội khách, đội bóng, quốc gia,…
- Nguồn dữ liệu: https://www.kaggle.com/datasets/brenda89/fifa-world-cup-2022
- Lý do chọn đề tài: chủ đề hấp dẫn.

## 2. Mô tả thuộc tính

- Dữ liệu có 23921 dòng dữ liệu và 25 cột thuộc tính.

| STT | Tên thuộc tính | Mô tả |
|---|---|---|
| 1 | date | Ngày diễn ra trận đấu |
| 2 | home_team | Tên đội chủ nhà |
| 3 | away_team | Tên đội khách |
| 4 | home_team_contient | Châu lục đội nhà |
| 5 | away_team_contient | Châu lục đội khách |
| 6 | home_team_fifa_rank | Thứ hạng FIFA của đội nhà tại thời điểm diễn ra trận đấu |
| 7 | away_team_fifa_rank | Thứ hạng FIFA của đội khách tại thời điểm diễn ra trận đấu |
| 8 | home_team_total_fifa_points | Tổng số điểm FIFA của đội nhà tại thời điểm diễn ra trận đấu |
| 9 | away_team_total_fifa_points | Tổng số điểm FIFA của đội khách tại thời điểm diễn ra trận đấu |
| 10 | home_team_score | Tỷ số toàn trận của đội nhà bao gồm hiệp phụ và không bao gồm loạt sút luân lưu |

| | | |
|---|---|---|
| 11 | away_team_score | Tỷ số toàn trận của đội khách bao gồm hiệp phụ và không bao gồm loạt sút luân lưu |
| 12 | tournament | Tên của giải đấu |
| 13 | city | Tên thành phố diễn ra trận đấu |
| 14 | country | Tên đất nước diễn ra trận đấu |
| 15 | neutral_location | Cho biết trận đấu có diễn ra ở vị trí trung lập hay không (True/False) |
| 16 | shoot_out | Cho biết trận đấu có bao gồm loạt sút luân lưu không (True/False) |
| 17 | home_team_result | Kết quả trận đấu của đội nhà, bao gồm loạt sút luân lưu |
| 18 | home_team_goalkeeper_score | Điểm trận đấu của thủ môn có hạng cáo nhất của đội chủ nhà |
| 19 | away_team_goalkeeper_score | Điểm trận đấu của thủ môn có hạng cáo nhất của đội khách |
| 20 | home_team_mean_defense_score | Trung bình điểm trận đấu của 4 cầu thủ hậu vệ của đội chủ nhà |
| 21 | home_team_mean_offense_score | Trung bình điểm trận đấu của 4 cầu thủ tiền vệ của đội chủ nhà |
| 22 | home_team_mean_midfield_score | Trung bình điểm trận đấu của 3 cầu thủ tiền đạo của đội chủ nhà, bao gồm tiền đạo cánh |
| 23 | away_team_mean_defense_score | Trung bình điểm trận đấu của 4 cầu thủ hậu vệ của đội khách |
| 24 | away_team_mean_offense_score | Trung bình điểm trận đấu của 4 cầu thủ tiền vệ của đội khách |
| 25 | away_team_mean_midfield_score | Trung bình điểm trận đấu của 3 cầu thủ tiền đạo của đội khách, bao gồm tiền đạo cánh |

### 3. Phát biểu bài toán

- Dựa trên thuộc tính đã có sẵn trong bộ dữ liệu, dự đoán đội bóng nào sẽ vô địch WC.

### 4. Công cụ khai thác dữ liệu

- Công cụ khai thác dữ liệu: Jupyter notebook
- Phiên bản Python: 3.12.2

### 5. Thư viện kèm theo

```
import time
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn import tree
from sklearn import metrics
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

# CHƯƠNG 2. TIỀN XỬ LÝ DỮ LIỆU

## 1. Xóa những thuộc tính không có ý nghĩa trong quá trình khai thác

- Đọc dữ liệu

```
[4]: # Đọc dữ liệu
     df = pd.read_csv("international_matches.csv")
```

- Dữ liệu gốc ban đầu gồm 23921 dòng dữ liệu và 25 thuộc tính.

[5]:

| | date | home_team | away_team | home_team_continent | away_team_continent | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1993-08-08 | Bolivia | Uruguay | South America | South America | 59 | 22 | 0 | |
| **1** | 1993-08-08 | Brazil | Mexico | South America | North America | 8 | 14 | 0 | |
| **2** | 1993-08-08 | Ecuador | Venezuela | South America | South America | 35 | 94 | 0 | |
| **3** | 1993-08-08 | Guinea | Sierra Leone | Africa | Africa | 65 | 86 | 0 | |
| **4** | 1993-08-08 | Paraguay | Argentina | South America | South America | 67 | 5 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **23916** | 2022-06-14 | Moldova | Andorra | Europe | Europe | 180 | 153 | 932 | |
| **23917** | 2022-06-14 | Liechtenstein | Latvia | Europe | Europe | 192 | 135 | 895 | |
| **23918** | 2022-06-14 | Chile | Ghana | South America | Africa | 28 | 60 | 1526 | |
| **23919** | 2022-06-14 | Japan | Tunisia | Asia | Africa | 23 | 35 | 1553 | |
| **23920** | 2022-06-14 | Korea Republic | Egypt | Asia | Africa | 29 | 32 | 1519 | |

23921 rows × 25 columns

- Xem các thông tin của các thuộc tính

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23921 entries, 0 to 23920
Data columns (total 25 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   date                           23921 non-null  object
 1   home_team                      23921 non-null  object
 2   away_team                      23921 non-null  object
 3   home_team_continent            23921 non-null  object
 4   away_team_continent            23921 non-null  object
 5   home_team_fifa_rank            23921 non-null  int64
 6   away_team_fifa_rank            23921 non-null  int64
 7   home_team_total_fifa_points    23921 non-null  int64
 8   away_team_total_fifa_points    23921 non-null  int64
 9   home_team_score                23921 non-null  int64
 10  away_team_score                23921 non-null  int64
 11  tournament                     23921 non-null  object
 12  city                           23921 non-null  object
 13  country                        23921 non-null  object
 14  neutral_location               23921 non-null  bool
 15  shoot_out                      23921 non-null  object
 16  home_team_result               23921 non-null  object
 17  home_team_goalkeeper_score     8379 non-null   float64
 18  away_team_goalkeeper_score     8095 non-null   float64
 19  home_team_mean_defense_score   7787 non-null   float64
 20  home_team_mean_offense_score   8510 non-null   float64
 21  home_team_mean_midfield_score  8162 non-null   float64
 22  away_team_mean_defense_score   7564 non-null   float64
 23  away_team_mean_offense_score   8312 non-null   float64
 24  away_team_mean_midfield_score  7979 non-null   float64
dtypes: bool(1), float64(8), int64(6), object(10)
memory usage: 4.4+ MB
```

- Thay đổi kiểu dữ liệu của một số thuộc tính để phù hợp hơn

[49]:
```
# Thay đổi kiểu dữ liệu để chính xác hơn
df["date"] = pd.to_datetime(df['date'])
df = df.replace({'shoot_out': {'Yes': True, 'No': False}})
```

- Lấy ra các đội đủ điều kiện tham gia World Cup 2022

[68]:
```
world_cup_teams = ['Qatar', 'Netherlands', 'Senegal', 'Ecuador', 'England', 'USA', 'IR Iran', 'Wales',
                   'Argentina', 'Mexico', 'Poland', 'Saudi Arabia', 'France', 'Denmark', 'Tunisia', 'Australia',
                   'Spain', 'Germany', 'Japan', 'Costa Rica', 'Belgium', 'Croatia', 'Morocco', 'Canada',
                   'Brazil', 'Switzerland', 'Serbia', 'Cameroon', 'Portugal', 'Uruguay', 'Korea Republic', 'Ghana']
```

- Giữ nguyên các trận đấu giữa các đội trong World Cup, các trận đấu của họ với các đội có thứ hạng dưới 100

```
[59]: # Xóa các kết quả trùng khớp không liên quan
      # Giữ nguyên các trận đấu giữa các đội trong World Cup, các trận đấu của họ với các đội có thứ hạng dưới 100
      # Giữ trận đấu giữa các đội có thứ hạng dưới 50

      df = df[((df['home_team'].isin(world_cup_teams) & df['away_team'].isin(world_cup_teams)) |
                       (df['home_team'].isin(world_cup_teams) & (df['away_team_fifa_rank'] <= 100)) |
                       ((df['home_team_fifa_rank'] <= 100) & df['away_team'].isin(world_cup_teams))|
                       ((df['home_team_fifa_rank'] <= 50) & (df['away_team_fifa_rank'] <= 50)))]
      df.reset_index(drop=True, inplace=True)
      df.shape

[59]: (8653, 25)
```

- Thêm một số thuộc tính cho việc phân tích ở sau

```
[21]: # Tạo các thuộc tính mới để thực hiện phân tích
      df['goal_difference'] = df['home_team_score'] - df['away_team_score']
      df['rank_difference'] = df['home_team_fifa_rank'] - df['away_team_fifa_rank']
      df['Friendly'] = df['tournament'] == 'Friendly'
      df['year'] = df['date'].dt.year
```

Loại bỏ các thuộc tính không có ý nghĩa trong quá trình phân tích:

- Xóa các thuộc tính **home_team_continent** và **away_team_continent** vì không có ý nghĩa.

```
[7]: # Xóa home_team_continent và away_team_continent vì không có ý nghĩa
     df = df.drop(['home_team_continent', 'away_team_continent'], axis = 1)
```

- Xóa các thuộc tính **city, country, neutral_location** vì các thuộc tính này không cần thiết trong quá trình phân tích.

```
[8]: # Xóa các thuộc tính city, country, neutral_location vì không cần trong quá trình phân tích
     df = df.drop(['city', 'country', 'neutral_location'], axis = 1)
```

- Xóa các thuộc tính **home_team_goalkeeper_score, away_team_goalkeeper_score, home_team_mean_defense_score, home_team_mean_offense_score, home_team_mean_midfield_score, away_team_mean_defense_score, away_team_mean_offense_score** và **away_team_mean_midfield_score** vì không cần thiết và có nhiều giá trị null.

[31]:
```
df = df.drop(['home_team_goalkeeper_score',
              'away_team_goalkeeper_score'], axis = 1)
```

[32]:
```
df = df.drop(['home_team_mean_defense_score',
              'home_team_mean_offense_score',
              'home_team_mean_midfield_score'], axis = 1)
```

[33]:
```
df = df.drop(['away_team_mean_defense_score',
              'away_team_mean_offense_score',
              'away_team_mean_midfield_score'], axis = 1)
```

- Kiểm tra dữ liệu sau khi xóa các thuộc tính không cần thiết

[75]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8653 entries, 0 to 8652
Data columns (total 12 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   date                        8653 non-null   datetime64[ns]
 1   home_team                   8653 non-null   object
 2   away_team                   8653 non-null   object
 3   home_team_fifa_rank         8653 non-null   int64
 4   away_team_fifa_rank         8653 non-null   int64
 5   home_team_total_fifa_points 8653 non-null   int64
 6   away_team_total_fifa_points 8653 non-null   int64
 7   home_team_score             8653 non-null   int64
 8   away_team_score             8653 non-null   int64
 9   tournament                  8653 non-null   object
 10  shoot_out                   8653 non-null   bool
 11  home_team_result            8653 non-null   object
dtypes: bool(1), datetime64[ns](1), int64(6), object(4)
memory usage: 752.2+ KB
```

## 2. Xóa bỏ những thuộc tính null hoặc Unknown

Kiểm tra các dữ liệu bị thiếu

- Tìm trong dữ liệu có giá trị null hay không

```
[35]: # Kiểm tra số lượng giá trị null trong mỗi cột
      missing_values = df.isna().sum()

      # Hiển thị số lượng giá trị null trong mỗi cột
      print(missing_values)
```

```
date                          0
home_team                     0
away_team                     0
home_team_fifa_rank           0
away_team_fifa_rank           0
home_team_total_fifa_points   0
away_team_total_fifa_points   0
home_team_score               0
away_team_score               0
tournament                    0
shoot_out                     0
home_team_result              0
dtype: int64
```

- Kiểm tra có dữ liệu bị trùng lặp không

```
[48]: # Kiểm tra dữ liệu trùng lặp
      df.duplicated().sum()
```

```
[48]: 0
```

### 3. Xuất kết quả tiền xử lý ra file csv

- Sau khi thực hiện tiền xử lý dữ liệu xong, bộ dữ liệu còn lại 8653 dòng dữ liệu và 16 cột thuộc tính.

[28]:
```
df
```

[28]:

| | date | home_team | away_team | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team_total_fifa_points | home_team_score | away_tean |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1993-08-08 | Bolivia | Uruguay | 59 | 22 | 0 | 0 | 3 | |
| **1** | 1993-08-08 | Brazil | Mexico | 8 | 14 | 0 | 0 | 1 | |
| **2** | 1993-08-08 | Ecuador | Venezuela | 35 | 94 | 0 | 0 | 5 | |
| **3** | 1993-08-08 | Paraguay | Argentina | 67 | 5 | 0 | 0 | 1 | |
| **4** | 1993-08-11 | Sweden | Switzerland | 4 | 3 | 0 | 0 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **8648** | 2022-06-14 | Poland | Belgium | 26 | 2 | 1544 | 1827 | 0 | |
| **8649** | 2022-06-14 | Ukraine | Republic of Ireland | 27 | 47 | 1535 | 1449 | 1 | |
| **8650** | 2022-06-14 | Chile | Ghana | 28 | 60 | 1526 | 1387 | 0 | |
| **8651** | 2022-06-14 | Japan | Tunisia | 23 | 35 | 1553 | 1499 | 0 | |
| **8652** | 2022-06-14 | Korea Republic | Egypt | 29 | 32 | 1519 | 1500 | 4 | |

8653 rows × 16 columns

[23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8653 entries, 0 to 8652
Data columns (total 16 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   date                        8653 non-null   datetime64[ns]
 1   home_team                   8653 non-null   object
 2   away_team                   8653 non-null   object
 3   home_team_fifa_rank         8653 non-null   int64
 4   away_team_fifa_rank         8653 non-null   int64
 5   home_team_total_fifa_points 8653 non-null   int64
 6   away_team_total_fifa_points 8653 non-null   int64
 7   home_team_score             8653 non-null   int64
 8   away_team_score             8653 non-null   int64
 9   tournament                  8653 non-null   object
 10  shoot_out                   8653 non-null   bool
 11  home_team_result            8653 non-null   object
 12  goal_difference             8653 non-null   int64
 13  rank_difference             8653 non-null   int64
 14  Friendly                    8653 non-null   bool
 15  year                        8653 non-null   int32
dtypes: bool(2), datetime64[ns](1), int32(1), int64(8), object(4)
memory usage: 929.7+ KB
```

- Xuất dữ liệu sang file csv

[26]:
```
# Xuất dữ liệu
df.to_csv('data_daxuly.csv')
```

# CHƯƠNG 3. ỨNG DỤNG GIẢI THUẬT PHÂN LỚP VÀO TẬP DỮ LIỆU

## 1. Giải thuật cây ID3

- Import dữ liệu đã xử lý vào và xem dữ liệu đó

```
[30]: # Import dữ liệu đã được xử lý
matches = pd.read_csv('data_daxuly.csv')
matches
```

[30]:

| | Unnamed: 0 | date | home_team | away_team | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team_total_fifa_points | home_team_score |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 1993-08-08 | Bolivia | Uruguay | 59 | 22 | 0 | 0 | 3 |
| **1** | 1 | 1993-08-08 | Brazil | Mexico | 8 | 14 | 0 | 0 | 1 |
| **2** | 2 | 1993-08-08 | Ecuador | Venezuela | 35 | 94 | 0 | 0 | 5 |
| **3** | 3 | 1993-08-08 | Paraguay | Argentina | 67 | 5 | 0 | 0 | 1 |
| **4** | 4 | 1993-08-11 | Sweden | Switzerland | 4 | 3 | 0 | 0 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **8648** | 8648 | 2022-06-14 | Poland | Belgium | 26 | 2 | 1544 | 1827 | 0 |
| **8649** | 8649 | 2022-06-14 | Ukraine | Republic of Ireland | 27 | 47 | 1535 | 1449 | 1 |
| **8650** | 8650 | 2022-06-14 | Chile | Ghana | 28 | 60 | 1526 | 1387 | 0 |
| **8651** | 8651 | 2022-06-14 | Japan | Tunisia | 23 | 35 | 1553 | 1499 | 0 |
| **8652** | 8652 | 2022-06-14 | Korea Republic | Egypt | 29 | 32 | 1519 | 1500 | 4 |

8653 rows × 17 columns

- Xóa thuộc tính **Unnamed: 0** và xem lại dữ liệu

```
[31]: # Xóa thuộc tính Unnamed: 0
matches = matches.drop(['Unnamed: 0'], axis = 1)
```

```
[32]: matches
```

[32]:

| | date | home_team | away_team | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team_total_fifa_points | home_team_score | away_tean |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1993-08-08 | Bolivia | Uruguay | 59 | 22 | 0 | 0 | 3 | |
| **1** | 1993-08-08 | Brazil | Mexico | 8 | 14 | 0 | 0 | 1 | |
| **2** | 1993-08-08 | Ecuador | Venezuela | 35 | 94 | 0 | 0 | 5 | |
| **3** | 1993-08-08 | Paraguay | Argentina | 67 | 5 | 0 | 0 | 1 | |
| **4** | 1993-08-11 | Sweden | Switzerland | 4 | 3 | 0 | 0 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **8648** | 2022-06-14 | Poland | Belgium | 26 | 2 | 1544 | 1827 | 0 | |
| **8649** | 2022-06-14 | Ukraine | Republic of Ireland | 27 | 47 | 1535 | 1449 | 1 | |
| **8650** | 2022-06-14 | Chile | Ghana | 28 | 60 | 1526 | 1387 | 0 | |
| **8651** | 2022-06-14 | Japan | Tunisia | 23 | 35 | 1553 | 1499 | 0 | |
| **8652** | 2022-06-14 | Korea Republic | Egypt | 29 | 32 | 1519 | 1500 | 4 | |

8653 rows × 16 columns

```
[26] matches['date'] = pd.to_datetime(matches['date'])
```

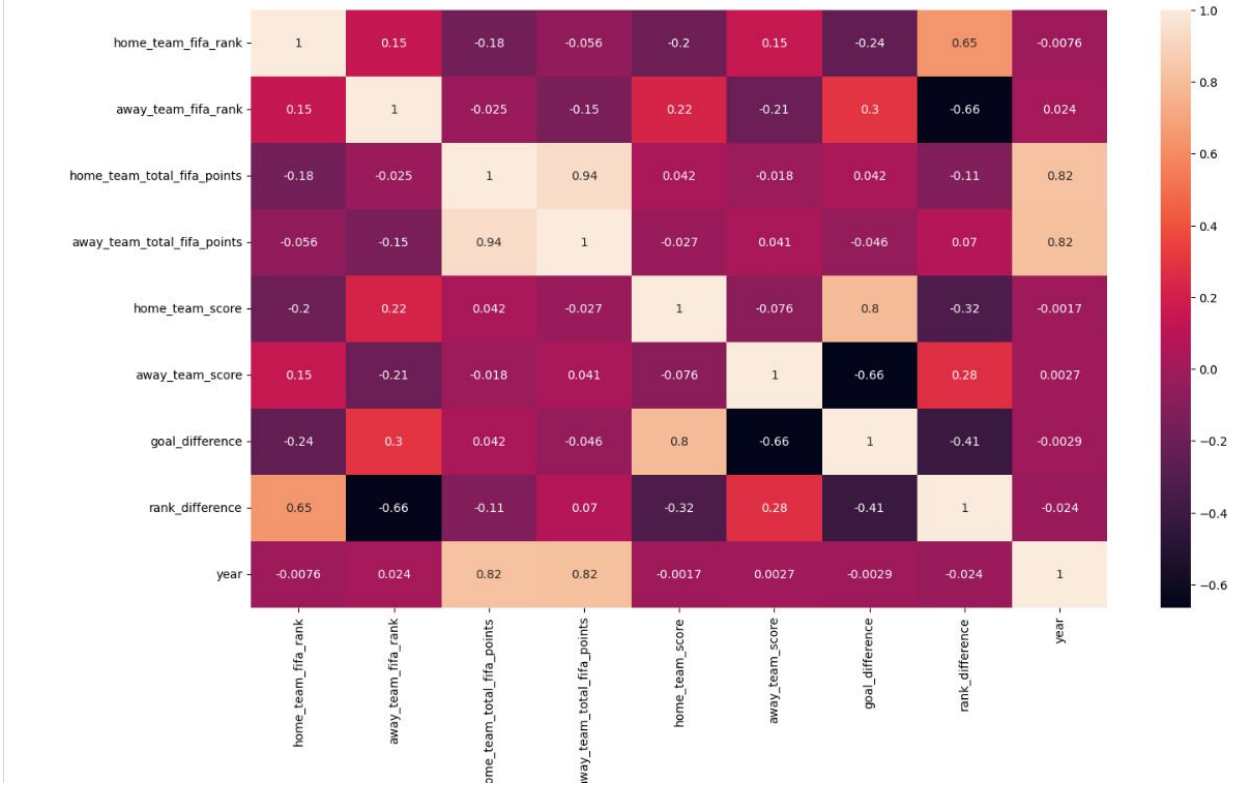
```
[27] matches = matches.select_dtypes(include=np.number)
```

```
[28] for col in matches.columns:
         matches[col] = pd.to_numeric(matches[col], errors='coerce')
```

- Xét thuộc tính tương đồng của các thuộc tính bằng dòng lệnh sau và cho kết quả như hình

```
plt.figure(figsize=(16,9))
sns.heatmap(matches.corr(method='pearson'),annot=True)
```

```
<Axes: >
```




Các cặp thuộc tính có độ tương quan cao:

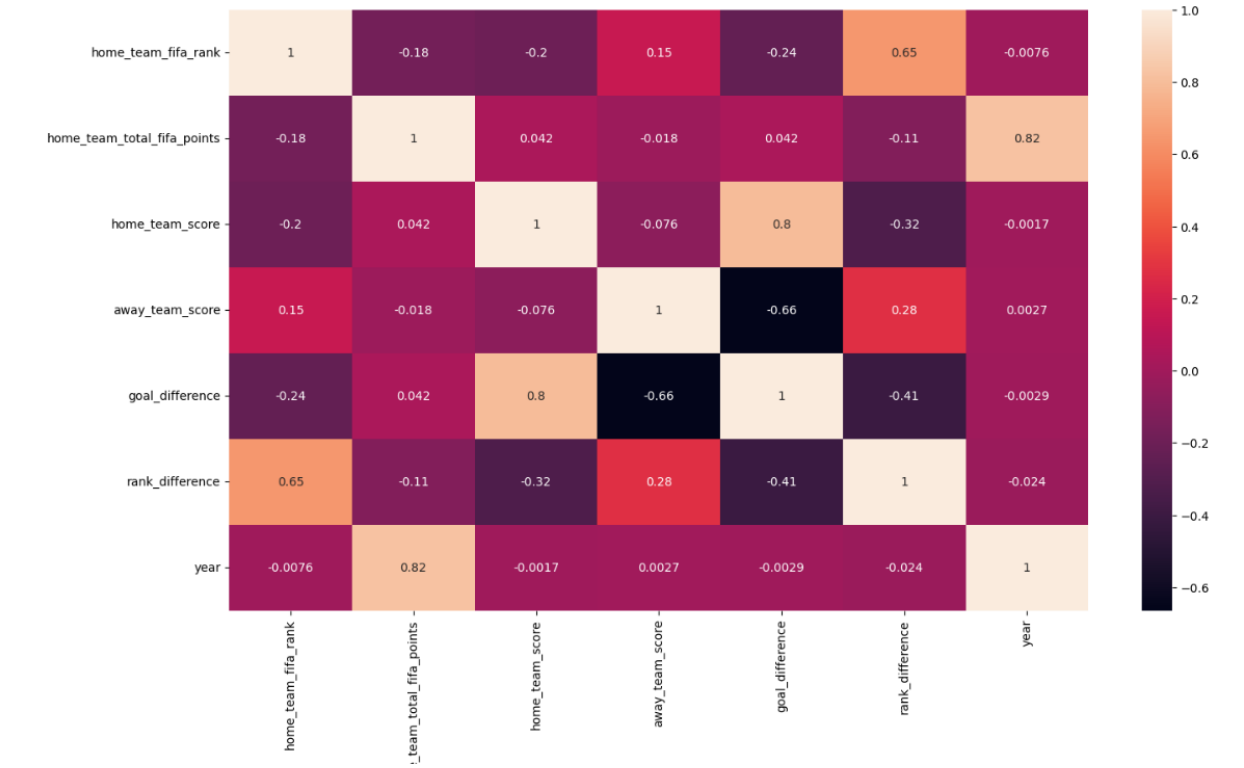
home_team_fifa_rank và away_team_fifa_rank -> bỏ away_team_fifa_rank

home_team_total_fifa_points và away_team_total_fifa_points -> bỏ away_team_total_fifa_points

```
[30] columns = ['away_team_fifa_rank', 'away_team_total_fifa_points']

     matches.drop(columns, inplace=True, axis=1)
```

```
plt.figure(figsize=(16,9))
sns.heatmap(matches.corr(method='pearson'),annot=True)
```

```
<Axes: >
```




- Tách dữ liệu thành 2 phần: Dữ liệu bình thường (feature) và dữ liệu chứa thuộc tính quyết định.

```
[32] data = matches.copy()
```

```
[33] # Tách dữ liệu feature
     X = data.drop('goal_difference', axis=1)
```

```
     # Tách dữ liệu chứa thuộc tính quyết định
     y = data['goal_difference']
```

```
[35] print("Dữ liệu feature:")
     print(X.head())
```

```
Dữ liệu feature:
   home_team_fifa_rank  home_team_total_fifa_points  home_team_score  \
0                   59                            0                3
1                    8                            0                1
2                   35                            0                5
3                   67                            0                1
4                    4                            0                1

   away_team_score  rank_difference  year
0                1               37  1993
1                1               -6  1993
2                0              -59  1993
3                3               62  1993
4                2                1  1993
```

```
[36] print("\nDữ liệu chứa thuộc tính quyết định:")
     print(y.head())
```

```
Dữ liệu chứa thuộc tính quyết định:
0    2
1    0
2    5
3   -2
4   -1
Name: goal_difference, dtype: int64
```

- Chia dữ liệu test và train theo tỉ lệ Train: Test=70:30 với test_size=30%

```
[38] # Chia dữ liệu thành tập train và test với tỉ lệ 70:30
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

- Kiểm tra thời gian chạy thuật toán

```
[ ]  #Xây dựng cây ID3
     #Đặt time chạy thuật toán
     start_ID3 =time.time()
     clf1 = tree.DecisionTreeClassifier(criterion="entropy", random_state=0)
     clf1.fit(X_train, y_train)
     end_ID3=time.time()
     thoigian1=timedelta(seconds=round(end_ID3-start_ID3,4))
     print(thoigian1)
```

```
0:00:00.020500
```

- Chạy thuật toán bằng dòng lệnh sau

```
[ ] cl1= tree.DecisionTreeClassifier(criterion="entropy", random_state=0)
    clf1.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

```
tree_pred1 = clf1.predict(X_test)
tree_score1 = metrics.accuracy_score(y_test, tree_pred1)
print("Độ chính xác: ",tree_score1)
print("Report:",metrics.classification_report(y_test, tree_pred1))
```

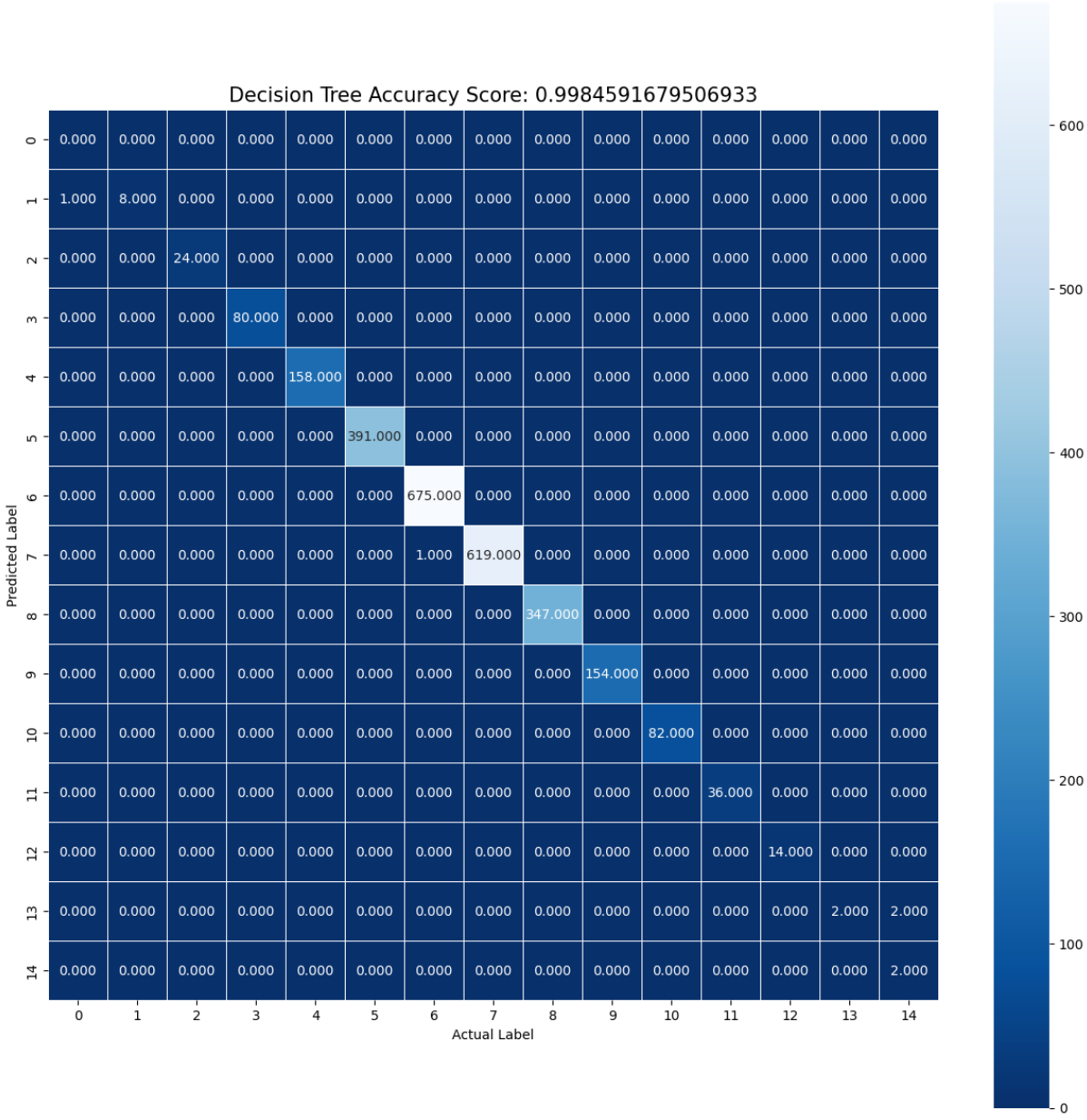
```
Độ chính xác:  0.9984591679506933
Report:               precision    recall  f1-score   support

          -6       0.00      0.00      0.00         0
          -5       1.00      0.89      0.94         9
          -4       1.00      1.00      1.00        24
          -3       1.00      1.00      1.00        80
          -2       1.00      1.00      1.00       158
          -1       1.00      1.00      1.00       391
           0       1.00      1.00      1.00       675
           1       1.00      1.00      1.00       620
           2       1.00      1.00      1.00       347
           3       1.00      1.00      1.00       154
           4       1.00      1.00      1.00        82
           5       1.00      1.00      1.00        36
           6       1.00      1.00      1.00        14
           7       1.00      0.50      0.67         4
           8       0.50      1.00      0.67         2

    accuracy                           1.00      2596
   macro avg       0.90      0.89      0.88      2596
weighted avg       1.00      1.00      1.00      2596
```

- Tính toán ma trận nhầm lẫn

```
[ ] #tính toán ma trận nhầm lẫn
    tree_cm1 = metrics.confusion_matrix(y_test, tree_pred1)
```

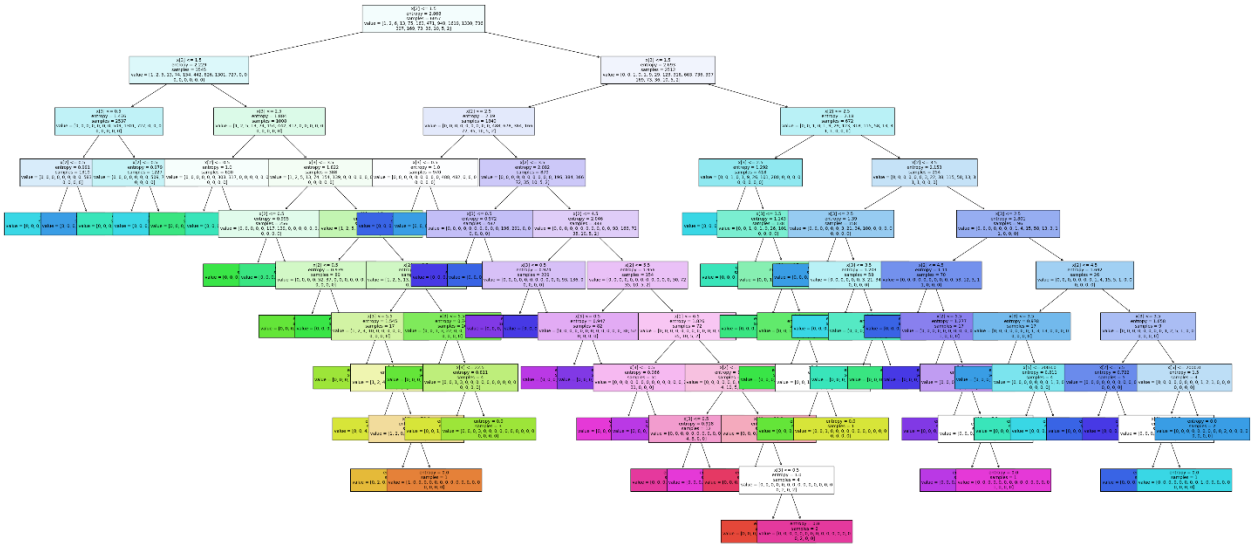
```
[ ] plt.figure(figsize=(15,15))
    sns.heatmap(tree_cm1, annot=True, fmt=".3f", linewidth=0.5, square=True, cmap="Blues_r");
    plt.xlabel("Actual Label");
    plt.ylabel("Predicted Label");
    title = 'Decision Tree Accuracy Score: {0}'.format(tree_score1)
    plt.title(title, size=15);
```

- Đồ thị thể hiện cho ma trận nhầm lẫn




- Hình ảnh mô tả cây ID3 của thuật toán

```
# Import the necessary module
import sklearn.tree as tree

# Plot the decision tree
fig, ax = plt.subplots(figsize=(50, 24))
tree.plot_tree(clf1, filled=True, fontsize=10)
plt.savefig('decision_tree', dpi=100)
plt.show()
```



## 2. Giải thuật cây CART

- Import dữ liệu đã xử lý vào và xem dữ liệu đó

```
[30]: # Import dữ liệu đã được xử lý
matches = pd.read_csv('data_daxuly.csv')
matches
```

[30]:

| | Unnamed: 0 | date | home_team | away_team | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team_total_fifa_points | home_team_score |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 1993-08-08 | Bolivia | Uruguay | 59 | 22 | 0 | 0 | 3 |
| **1** | 1 | 1993-08-08 | Brazil | Mexico | 8 | 14 | 0 | 0 | 1 |
| **2** | 2 | 1993-08-08 | Ecuador | Venezuela | 35 | 94 | 0 | 0 | 5 |
| **3** | 3 | 1993-08-08 | Paraguay | Argentina | 67 | 5 | 0 | 0 | 1 |
| **4** | 4 | 1993-08-11 | Sweden | Switzerland | 4 | 3 | 0 | 0 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **8648** | 8648 | 2022-06-14 | Poland | Belgium | 26 | 2 | 1544 | 1827 | 0 |
| **8649** | 8649 | 2022-06-14 | Ukraine | Republic of Ireland | 27 | 47 | 1535 | 1449 | 1 |
| **8650** | 8650 | 2022-06-14 | Chile | Ghana | 28 | 60 | 1526 | 1387 | 0 |
| **8651** | 8651 | 2022-06-14 | Japan | Tunisia | 23 | 35 | 1553 | 1499 | 0 |
| **8652** | 8652 | 2022-06-14 | Korea Republic | Egypt | 29 | 32 | 1519 | 1500 | 4 |

8653 rows × 17 columns

- Xóa thuộc tính **Unnamed: 0** và xem lại dữ liệu

```
[31]: # Xóa thuộc tính Unnamed: 0
      matches = matches.drop(['Unnamed: 0'], axis = 1)

[32]: matches
```

[32]:

| | date | home_team | away_team | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team_total_fifa_points | home_team_score | away_tean |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1993-08-08 | Bolivia | Uruguay | 59 | 22 | 0 | 0 | 3 | |
| **1** | 1993-08-08 | Brazil | Mexico | 8 | 14 | 0 | 0 | 1 | |
| **2** | 1993-08-08 | Ecuador | Venezuela | 35 | 94 | 0 | 0 | 5 | |
| **3** | 1993-08-08 | Paraguay | Argentina | 67 | 5 | 0 | 0 | 1 | |
| **4** | 1993-08-11 | Sweden | Switzerland | 4 | 3 | 0 | 0 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **8648** | 2022-06-14 | Poland | Belgium | 26 | 2 | 1544 | 1827 | 0 | |
| **8649** | 2022-06-14 | Ukraine | Republic of Ireland | 27 | 47 | 1535 | 1449 | 1 | |
| **8650** | 2022-06-14 | Chile | Ghana | 28 | 60 | 1526 | 1387 | 0 | |
| **8651** | 2022-06-14 | Japan | Tunisia | 23 | 35 | 1553 | 1499 | 0 | |
| **8652** | 2022-06-14 | Korea Republic | Egypt | 29 | 32 | 1519 | 1500 | 4 | |

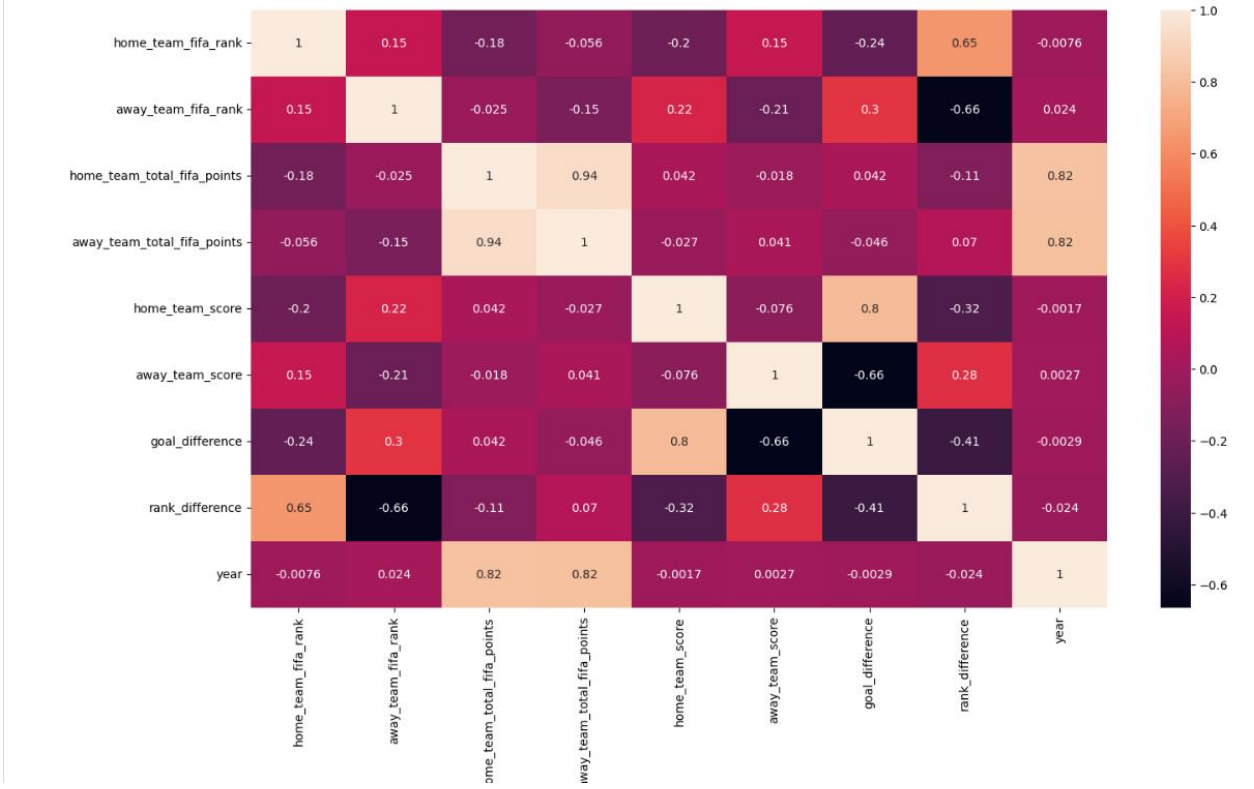
8653 rows × 16 columns

```
[26] matches['date'] = pd.to_datetime(matches['date'])

[27] matches = matches.select_dtypes(include=np.number)

[28] for col in matches.columns:
         matches[col] = pd.to_numeric(matches[col], errors='coerce')
```

Xét thuộc tính tương đồng của các thuộc tính bằng dòng lệnh sau và cho kết quả như hình

```
plt.figure(figsize=(16,9))
sns.heatmap(matches.corr(method='pearson'),annot=True)
```

```
<Axes: >
```




Các cặp thuộc tính có độ tương quan cao:

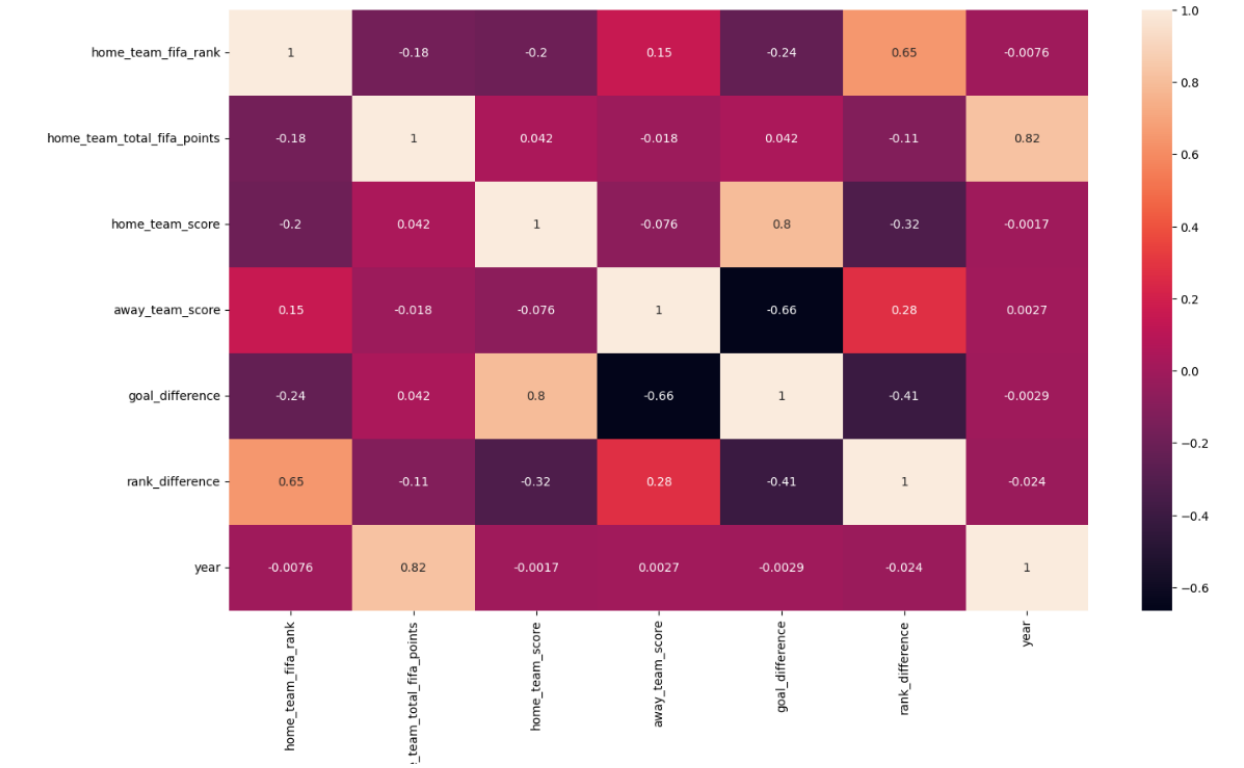
home_team_fifa_rank và away_team_fifa_rank -> bỏ away_team_fifa_rank

home_team_total_fifa_points và away_team_total_fifa_points -> bỏ away_team_total_fifa_points

```
[30] columns = ['away_team_fifa_rank', 'away_team_total_fifa_points']

     matches.drop(columns, inplace=True, axis=1)
```

```
plt.figure(figsize=(16,9))
sns.heatmap(matches.corr(method='pearson'),annot=True)
```

```
<Axes: >
```




- Tách dữ liệu thành 2 phần: Dữ liệu bình thường (feature) và dữ liệu chứa thuộc tính quyết định.

```
[32] data = matches.copy()
```

```
[33] # Tách dữ liệu feature
     X = data.drop('goal_difference', axis=1)
```

```
     # Tách dữ liệu chứa thuộc tính quyết định
     y = data['goal_difference']
```

```
[35] print("Dữ liệu feature:")
     print(X.head())
```

```
Dữ liệu feature:
   home_team_fifa_rank  home_team_total_fifa_points  home_team_score  \
0                   59                            0                3
1                    8                            0                1
2                   35                            0                5
3                   67                            0                1
4                    4                            0                1

   away_team_score  rank_difference  year
0                1               37  1993
1                1               -6  1993
2                0              -59  1993
3                3               62  1993
4                2                1  1993
```

```
[36] print("\nDữ liệu chứa thuộc tính quyết định:")
     print(y.head())
```

```

Dữ liệu chứa thuộc tính quyết định:
0    2
1    0
2    5
3   -2
4   -1
Name: goal_difference, dtype: int64
```

- Chia dữ liệu test và train theo tỉ lệ Train: Test=70:30 với test_size=30%

```
[38] # Chia dữ liệu thành tập train và test với tỉ lệ 70:30
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

- Kiểm tra thời gian chạy thuật toán

```
[40] # Kiểm tra thời gian chạy thuật toán
     start_CART = time.time()
     clf2 = tree.DecisionTreeClassifier(criterion='gini', random_state = 0)
     clf2.fit(X_train, y_train)
     end_CART = time.time()
     thoigian2 = timedelta(seconds = round(end_CART - start_CART, 4))
     print(thoigian2)
```

```
0:00:00.032000
```

- Chạy thuật toán bằng dòng lệnh

```
[43] from sklearn import metrics
     # Chạy thuật toán
     tree_pred2 = clf2.predict(X_test)
     tree_score2 = metrics.accuracy_score(y_test, tree_pred2)
     print("Độ chính xác:", tree_score2)
     print("Report:", metrics.classification_report(y_test, tree_pred2))
```

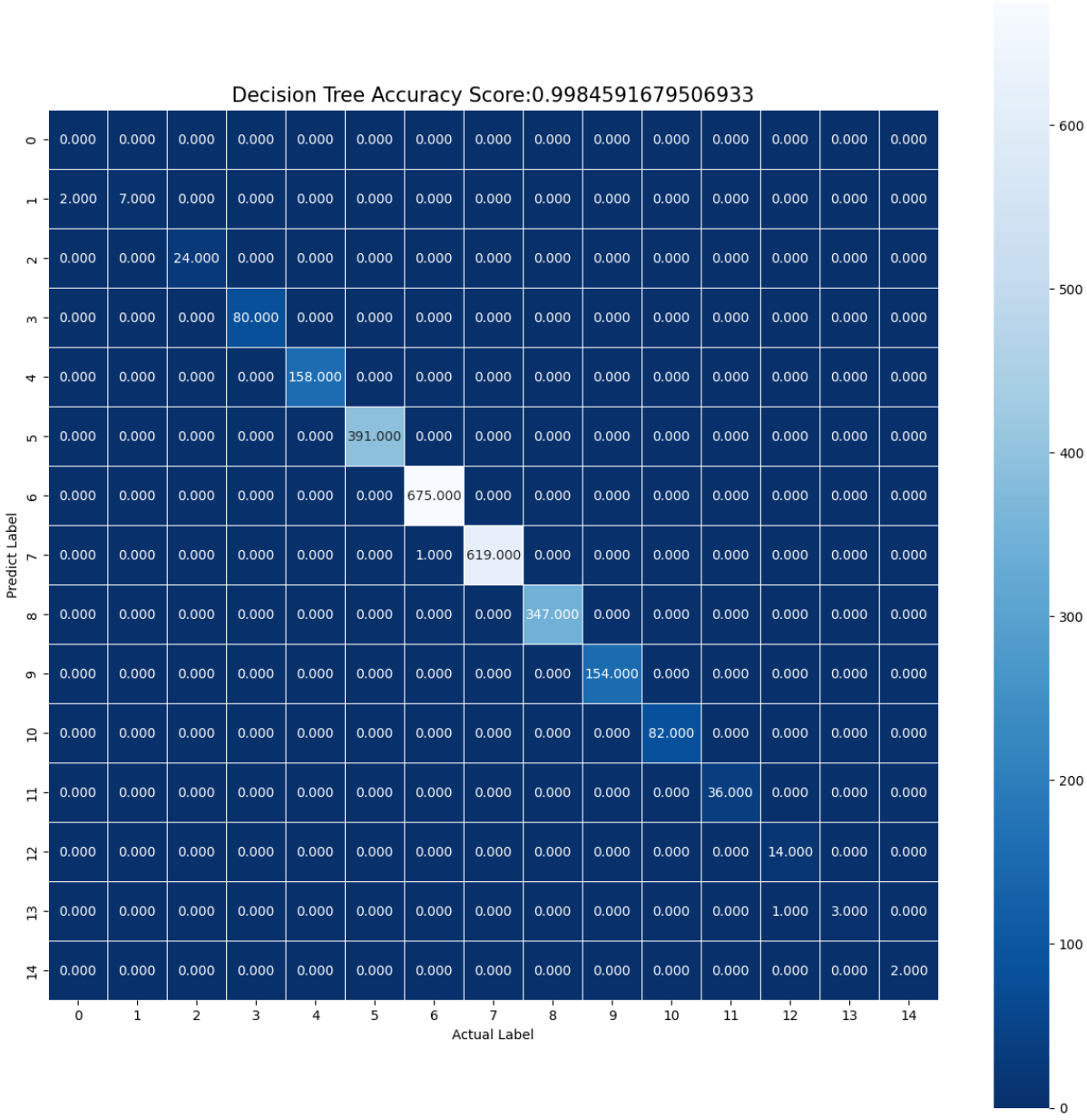
```
Độ chính xác: 0.9984591679506933
Report:               precision    recall  f1-score   support

          -6       0.00      0.00      0.00         0
          -5       1.00      0.78      0.88         9
          -4       1.00      1.00      1.00        24
          -3       1.00      1.00      1.00        80
          -2       1.00      1.00      1.00       158
          -1       1.00      1.00      1.00       391
           0       1.00      1.00      1.00       675
           1       1.00      1.00      1.00       620
           2       1.00      1.00      1.00       347
           3       1.00      1.00      1.00       154
           4       1.00      1.00      1.00        82
           5       1.00      1.00      1.00        36
           6       0.93      1.00      0.97        14
           7       1.00      0.75      0.86         4
           8       1.00      1.00      1.00         2

    accuracy                           1.00      2596
   macro avg       0.93      0.90      0.91      2596
weighted avg       1.00      1.00      1.00      2596
```

- Tính toán ma trận nhầm lẫn

```
[44] # Tính ma trận nhầm lẫn
     tree_cm2 = metrics.confusion_matrix(y_test, tree_pred2)
```

```
[45] # Vẽ ma trận nhầm lẫn
     plt.figure(figsize=(15, 15))
     sns.heatmap(tree_cm2, annot = True, fmt = '.3f', linewidth = .5, square = True, cmap = 'Blues_r')
     plt.xlabel('Actual Label')
     plt.ylabel('Predict Label')
     title = 'Decision Tree Accuracy Score:{0}'.format(tree_score2)
     plt.title(title, size=15)
```

- Đồ thị thể hiện cho ma trận nhầm lẫn




## 3. Giải thuật Naïve Bayes

- Import dữ liệu đã xử lý vào và xem dữ liệu đó

[30]:
```
# Import dữ liệu đã được xử lý
matches = pd.read_csv('data_daxuly.csv')
matches
```

[30]:

| | Unnamed: 0 | date | home_team | away_team | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team_total_fifa_points | home_team_score |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 1993-08-08 | Bolivia | Uruguay | 59 | 22 | 0 | 0 | 3 |
| **1** | 1 | 1993-08-08 | Brazil | Mexico | 8 | 14 | 0 | 0 | 1 |
| **2** | 2 | 1993-08-08 | Ecuador | Venezuela | 35 | 94 | 0 | 0 | 5 |
| **3** | 3 | 1993-08-08 | Paraguay | Argentina | 67 | 5 | 0 | 0 | 1 |
| **4** | 4 | 1993-08-11 | Sweden | Switzerland | 4 | 3 | 0 | 0 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **8648** | 8648 | 2022-06-14 | Poland | Belgium | 26 | 2 | 1544 | 1827 | 0 |
| **8649** | 8649 | 2022-06-14 | Ukraine | Republic of Ireland | 27 | 47 | 1535 | 1449 | 1 |
| **8650** | 8650 | 2022-06-14 | Chile | Ghana | 28 | 60 | 1526 | 1387 | 0 |
| **8651** | 8651 | 2022-06-14 | Japan | Tunisia | 23 | 35 | 1553 | 1499 | 0 |
| **8652** | 8652 | 2022-06-14 | Korea Republic | Egypt | 29 | 32 | 1519 | 1500 | 4 |

8653 rows × 17 columns

- Xóa thuộc tính **Unnamed: 0** và xem lại dữ liệu

[31]:
```
# Xóa thuộc tính Unnamed: 0
matches = matches.drop(['Unnamed: 0'], axis = 1)
```

[32]:
```
matches
```

[32]:

| | date | home_team | away_team | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team_total_fifa_points | home_team_score | away_tean |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1993-08-08 | Bolivia | Uruguay | 59 | 22 | 0 | 0 | 3 | |
| **1** | 1993-08-08 | Brazil | Mexico | 8 | 14 | 0 | 0 | 1 | |
| **2** | 1993-08-08 | Ecuador | Venezuela | 35 | 94 | 0 | 0 | 5 | |
| **3** | 1993-08-08 | Paraguay | Argentina | 67 | 5 | 0 | 0 | 1 | |
| **4** | 1993-08-11 | Sweden | Switzerland | 4 | 3 | 0 | 0 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **8648** | 2022-06-14 | Poland | Belgium | 26 | 2 | 1544 | 1827 | 0 | |
| **8649** | 2022-06-14 | Ukraine | Republic of Ireland | 27 | 47 | 1535 | 1449 | 1 | |
| **8650** | 2022-06-14 | Chile | Ghana | 28 | 60 | 1526 | 1387 | 0 | |
| **8651** | 2022-06-14 | Japan | Tunisia | 23 | 35 | 1553 | 1499 | 0 | |
| **8652** | 2022-06-14 | Korea Republic | Egypt | 29 | 32 | 1519 | 1500 | 4 | |

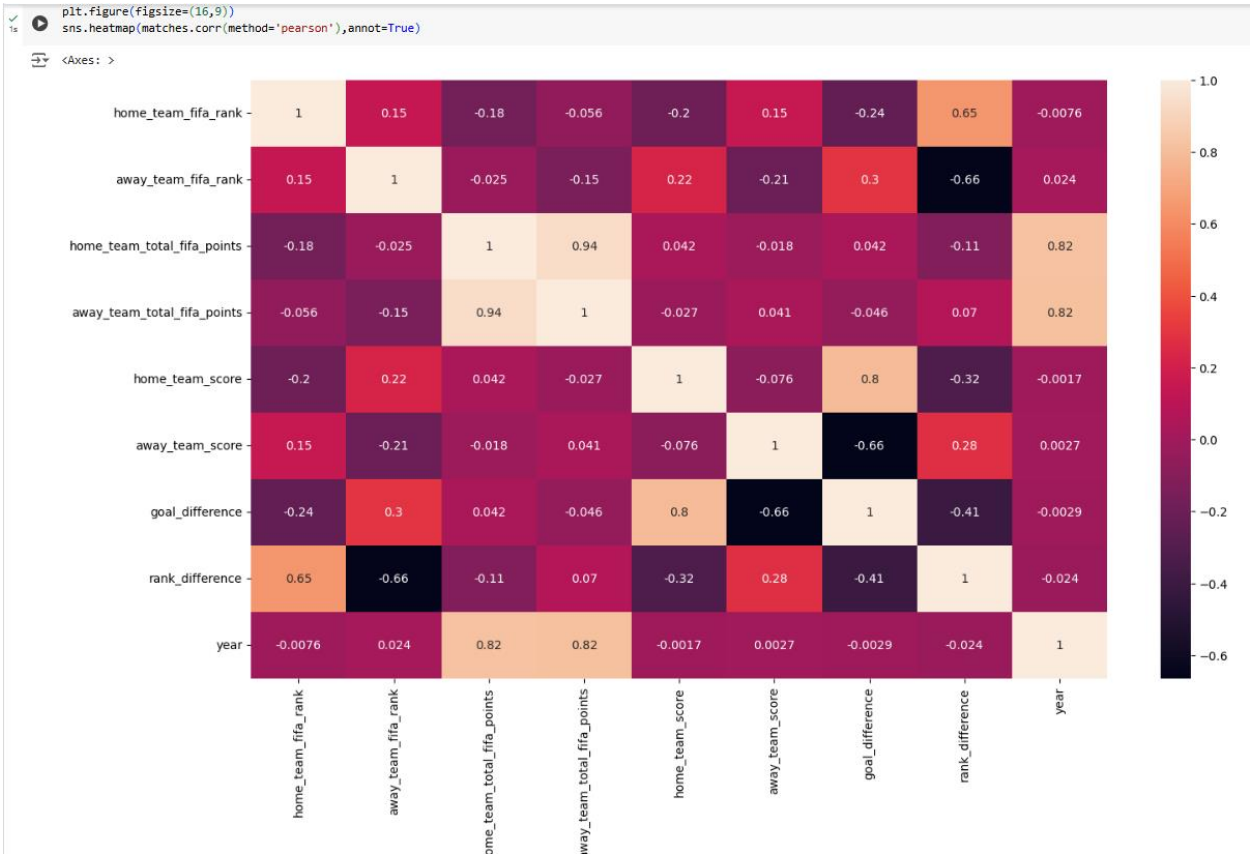
8653 rows × 16 columns

```
[26] matches['date'] = pd.to_datetime(matches['date'])

[27] matches = matches.select_dtypes(include=np.number)

[28] for col in matches.columns:
         matches[col] = pd.to_numeric(matches[col], errors='coerce')
```

Xét thuộc tính tương đồng của các thuộc tính bằng dòng lệnh sau và cho kết quả như hình

```
plt.figure(figsize=(16,9))
sns.heatmap(matches.corr(method='pearson'),annot=True)
```

```
<Axes: >
```




Các cặp thuộc tính có độ tương quan cao:

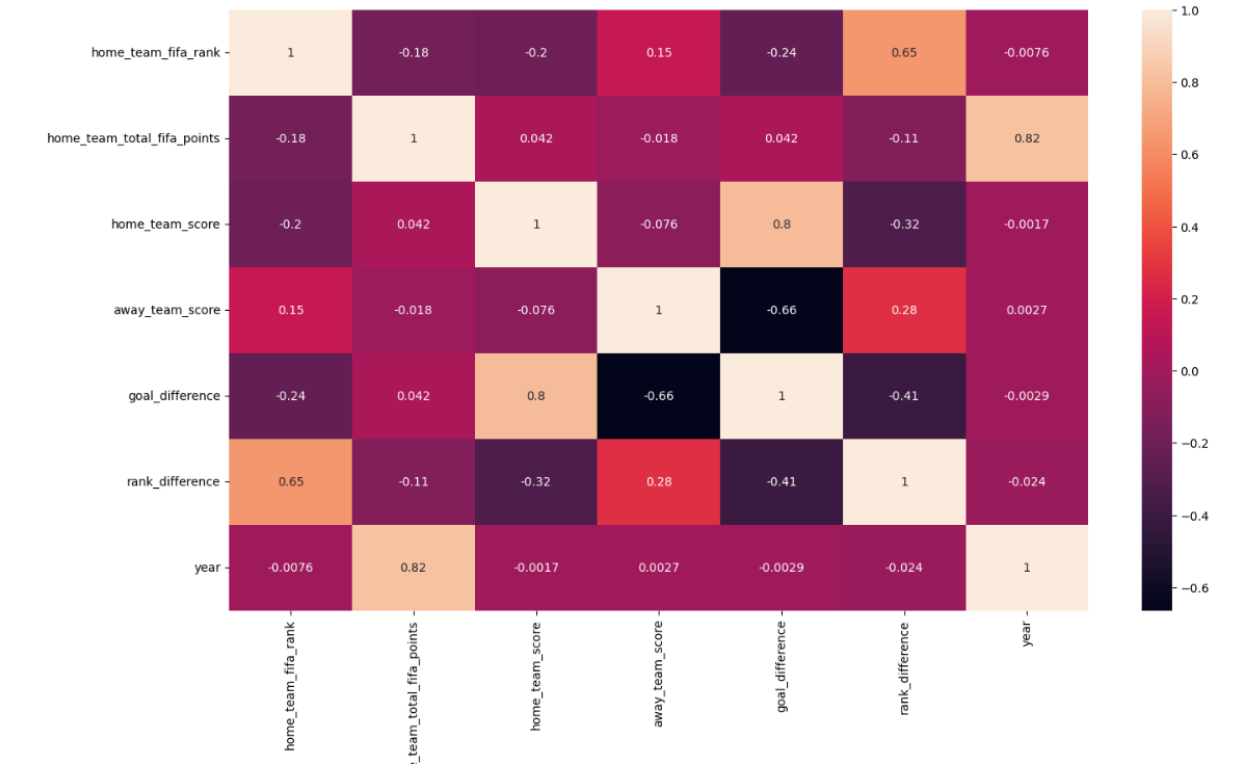
home_team_fifa_rank và away_team_fifa_rank -> bỏ away_team_fifa_rank

home_team_total_fifa_points và away_team_total_fifa_points -> bỏ away_team_total_fifa_points

```
[30] columns = ['away_team_fifa_rank', 'away_team_total_fifa_points']

     matches.drop(columns, inplace=True, axis=1)
```

```
plt.figure(figsize=(16,9))
sns.heatmap(matches.corr(method='pearson'),annot=True)
```

```
<Axes: >
```




- Tách dữ liệu thành 2 phần: Dữ liệu bình thường (feature) và dữ liệu chứa thuộc tính quyết định.

```
[32] data = matches.copy()
```

```
[33] # Tách dữ liệu feature
     X = data.drop('goal_difference', axis=1)
```

```
     # Tách dữ liệu chứa thuộc tính quyết định
     y = data['goal_difference']
```

```
[35] print("Dữ liệu feature:")
     print(X.head())
```

```
Dữ liệu feature:
   home_team_fifa_rank  home_team_total_fifa_points  home_team_score  \
0                   59                            0                3
1                    8                            0                1
2                   35                            0                5
3                   67                            0                1
4                    4                            0                1

   away_team_score  rank_difference  year
0                1               37  1993
1                1               -6  1993
2                0              -59  1993
3                3               62  1993
4                2                1  1993
```

```
[36] print("\nDữ liệu chứa thuộc tính quyết định:")
     print(y.head())
```

```
Dữ liệu chứa thuộc tính quyết định:
0    2
1    0
2    5
3   -2
4   -1
Name: goal_difference, dtype: int64
```

- Chia dữ liệu test và train theo tỉ lệ Train: Test=70:30 với test_size=30%

```
[38] # Chia dữ liệu thành tập train và test với tỉ lệ 70:30
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

- Kiểm tra thời gian chạy của thuật toán

```
[77]: # Kiểm tra thời gian chạy thuật toán
      start_Bayes = time.time()
      gnb = GaussianNB()
      bayes_pred = gnb.fit(X_train, y_train).predict(X_test)
      end_Bayes = time.time()
      thoigian3 = timedelta(seconds=round(end_Bayes - start_Bayes, 4))
      print(thoigian3)

      0:00:00.005000
```

- Chạy thuật toán bằng các dòng lệnh sau

[78]:
```
# Chạy thuật toán
bayes_score = metrics.accuracy_score(y_test, bayes_pred)
print("Độ chính xác:", bayes_score)
print("Report:", metrics.classification_report(y_test, bayes_pred))
```

```
Độ chính xác: 0.6224961479198767
Report:               precision    recall  f1-score   support

          -6       0.00      0.00      0.00         0
          -5       0.50      0.33      0.40         9
          -4       0.61      0.46      0.52        24
          -3       0.52      0.19      0.28        80
          -2       0.50      0.42      0.46       158
          -1       0.57      0.46      0.51       391
           0       0.64      0.84      0.73       675
           1       0.63      0.74      0.68       620
           2       0.55      0.33      0.42       347
           3       0.85      0.72      0.78       154
           4       0.81      0.67      0.73        82
           5       0.43      0.64      0.52        36
           6       0.70      0.50      0.58        14
           7       0.60      0.75      0.67         4
           8       1.00      1.00      1.00         2

    accuracy                           0.62      2596
   macro avg       0.59      0.54      0.55      2596
weighted avg       0.62      0.62      0.61      2596
```

## 4. Giải thuật Random Forest

- Import dữ liệu đã xử lý vào và xem dữ liệu đó

[30]:
```
# Import dữ liệu đã được xử lý
matches = pd.read_csv('data_daxuly.csv')
matches
```

[30]:

| | Unnamed: 0 | date | home_team | away_team | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team_total_fifa_points | home_team_score |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 1993-08-08 | Bolivia | Uruguay | 59 | 22 | 0 | 0 | 3 |
| **1** | 1 | 1993-08-08 | Brazil | Mexico | 8 | 14 | 0 | 0 | 1 |
| **2** | 2 | 1993-08-08 | Ecuador | Venezuela | 35 | 94 | 0 | 0 | 5 |
| **3** | 3 | 1993-08-08 | Paraguay | Argentina | 67 | 5 | 0 | 0 | 1 |
| **4** | 4 | 1993-08-11 | Sweden | Switzerland | 4 | 3 | 0 | 0 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **8648** | 8648 | 2022-06-14 | Poland | Belgium | 26 | 2 | 1544 | 1827 | 0 |
| **8649** | 8649 | 2022-06-14 | Ukraine | Republic of Ireland | 27 | 47 | 1535 | 1449 | 1 |
| **8650** | 8650 | 2022-06-14 | Chile | Ghana | 28 | 60 | 1526 | 1387 | 0 |
| **8651** | 8651 | 2022-06-14 | Japan | Tunisia | 23 | 35 | 1553 | 1499 | 0 |
| **8652** | 8652 | 2022-06-14 | Korea Republic | Egypt | 29 | 32 | 1519 | 1500 | 4 |

8653 rows × 17 columns

- Xóa thuộc tính **Unnamed: 0** và xem lại dữ liệu

```
[31]: # Xóa thuộc tính Unnamed: 0
      matches = matches.drop(['Unnamed: 0'], axis = 1)
```

```
[32]: matches
```

[32]:

| | date | home_team | away_team | home_team_fifa_rank | away_team_fifa_rank | home_team_total_fifa_points | away_team_total_fifa_points | home_team_score | away_tean |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1993-08-08 | Bolivia | Uruguay | 59 | 22 | 0 | 0 | 3 | |
| **1** | 1993-08-08 | Brazil | Mexico | 8 | 14 | 0 | 0 | 1 | |
| **2** | 1993-08-08 | Ecuador | Venezuela | 35 | 94 | 0 | 0 | 5 | |
| **3** | 1993-08-08 | Paraguay | Argentina | 67 | 5 | 0 | 0 | 1 | |
| **4** | 1993-08-11 | Sweden | Switzerland | 4 | 3 | 0 | 0 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **8648** | 2022-06-14 | Poland | Belgium | 26 | 2 | 1544 | 1827 | 0 | |
| **8649** | 2022-06-14 | Ukraine | Republic of Ireland | 27 | 47 | 1535 | 1449 | 1 | |
| **8650** | 2022-06-14 | Chile | Ghana | 28 | 60 | 1526 | 1387 | 0 | |
| **8651** | 2022-06-14 | Japan | Tunisia | 23 | 35 | 1553 | 1499 | 0 | |
| **8652** | 2022-06-14 | Korea Republic | Egypt | 29 | 32 | 1519 | 1500 | 4 | |

8653 rows × 16 columns

- Import thư viện và thực hiện chạy thuật toán

```
[84]: from sklearn.ensemble import RandomForestClassifier
```

```
[85]: # Kiểm tra thời gian chạy thuật toán
      start_RF = time.time()
      clf = RandomForestClassifier(n_estimators = 100)
      clf.fit(X_train, y_train)
      y_pred = clf.predict(X_test)
      end_RF = time.time()
      thoigian4 = timedelta(seconds = round(end_RF - start_RF, 4))
      print(thoigian4)
```

```
0:00:00.496100
```

```
[86]: print("Accuracy: ", metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy:  0.9849768875192604
```

## 5. Giải thích các độ đo

- Độ đo thời gian được tính từ lúc bắt đầu chạy thuật toán đến lúc hoàn thành thuật toán.
- Về độ công thức tính độ chính xác dựa vào ma trận nhầm lẫn (Confusion Matrix).

| | | Lớp dự đoán được từ mô hình | |
|---|---|---|---|
| | | *Lớp dương* | *Lớp âm* |
| **Lớp trên thực tế** | *Lớp dương* | a | b |
| | *Lớp âm* | c | d |

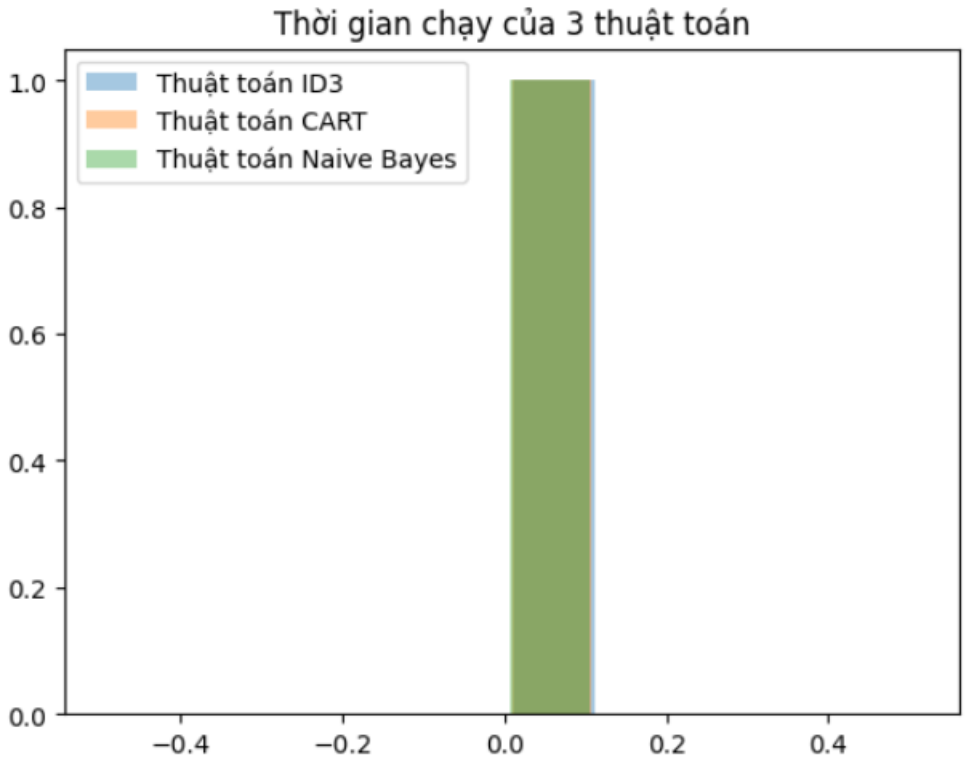
- Độ chính xác thuật toán precision(M) = $\frac{a}{a+b}$
- Độ phủ recall(M) = $\frac{a}{a+b}$
- Độ chính xác thuật toán Accuracy = $\frac{a+d}{a+b+c+d}$

# CHƯƠNG 4. PHÂN TÍCH ĐÁNH GIÁ CÁC THUẬT TOÁN VÀ DỰ BÁO

## 1. Đánh giá về thời gian chạy thuật toán

- Đo thời gian của 3 thuật toán bằng các dòng lệnh sau

```
[153]: # Đo thời gian của 3 thuật toán
ax = sns.distplot(end_ID3-start_ID3, bins=10, label = 'Thuật toán ID3', kde = False)
ax = sns.distplot(end_CART-start_CART, bins=10, label = 'Thuật toán CART', kde = False)
ax = sns.distplot(end_Bayes-start_Bayes, bins=10, label = 'Thuật toán Naive Bayes', kde = False)
ax.legend()
ax.set_title('Thời gian chạy của 3 thuật toán')
```



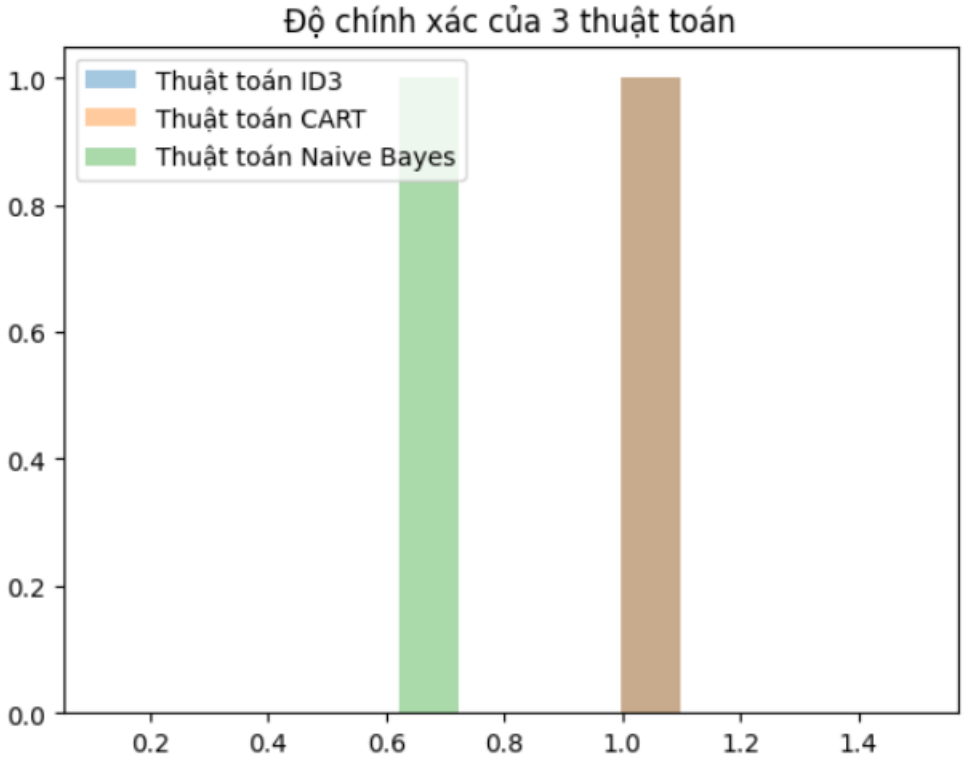

Nhận xét:

- Thời gian chạy của thuật toán Naïve Bayes là nhanh nhất đối với tập dữ liệu, tùy thuộc tốc độ chạy của máy tính mà thời gian chạy của thuật toán sẽ thay đổi.

## 2. Đánh giá về độ chính xác của thuật toán

- Đo độ chính xác của 3 thuật toán bằng các dòng lệnh sau

[154]:
```
# Đo độ chính xác của 3 thuật toán
ax = sns.distplot(tree_score1, bins=10, label = 'Thuật toán ID3', kde = False)
ax = sns.distplot(tree_score2, bins=10, label = 'Thuật toán CART', kde = False)
ax = sns.distplot(bayes_score, bins=10, label = 'Thuật toán Naive Bayes', kde = False)
ax.legend()
ax.set_title('Độ chính xác của 3 thuật toán')
```




Nhận xét:

- Cột màu nâu là cột biểu diễn độ chính xác của 2 thuật toán ID3 và CART dường như ghi đè lên nhau vì độ chính xác của 2 thuật toán này gần bằng nhau.

### 3. Dự báo

- Import dữ liệu để tiến hành dự báo

[52]:
```
# Load the training dataset from the 'data' DataFrame
X_train = data.drop('goal_difference', axis=1)
y_train = data['goal_difference']
```

[53]:
```
# Load the testing dataset from the 'test_data.csv' file
test1 = pd.read_csv('test_data.csv')
test1 = test1.drop(['Unnamed: 0'], axis = 1)
```

[54]:
```
# Ensure consistent feature names between training and testing data
X_test = test1.drop('goal_difference', axis=1)
y_test = test1['goal_difference']
```

- Chạy thuật toán như sau

[56]:
```
clf1= tree.DecisionTreeClassifier(criterion="entropy", random_state=0)
clf1.fit(X_train, y_train)
```

[56]:
```
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

- Tiến hành dữ báo

[58]:
```
tree_pred1 = clf1.predict(X_test)
tree_score1 = metrics.accuracy_score(y_test, tree_pred1)
print("Report:",metrics.classification_report(y_test, tree_pred1))
```

```
Report:               precision    recall  f1-score   support

          -8       1.00      1.00      1.00         1
          -7       1.00      1.00      1.00         2
          -6       1.00      1.00      1.00         6
          -5       1.00      1.00      1.00        22
          -4       1.00      1.00      1.00        99
          -3       1.00      1.00      1.00       243
          -2       1.00      1.00      1.00       629
          -1       1.00      1.00      1.00      1340
           0       1.00      1.00      1.00      2294
           1       1.00      1.00      1.00      1950
           2       1.00      1.00      1.00      1083
           3       1.00      1.00      1.00       551
           4       1.00      1.00      1.00       251
           5       1.00      1.00      1.00       109
           6       1.00      1.00      1.00        50
           7       1.00      1.00      1.00        14
           8       1.00      1.00      1.00         7
           9       1.00      1.00      1.00         2

    accuracy                           1.00      8653
   macro avg       1.00      1.00      1.00      8653
weighted avg       1.00      1.00      1.00      8653
```

## 4. Kết luận

- **Ưu điểm**: có khả năng tìm hiểu và khai thác dữ liệu, thực hiện tiền xử lý và phân lớp bằng các thuật toán đã học, có kinh nghiệm làm việc trong nhóm và xây dựng mô hình dự đoán nhà vô địch World Cup.
- **Nhược điểm**: Do hạn chế về thời gian trong một học kì, nhóm chưa thể tối ưu hóa độ chính xác của thuật toán, cũng cần nắm vững hơn về ngôn ngữ Python.
- **Hướng phát triển**: nghiên cứu các thuật toán để giảm số lượng cột trong việc tiền xử lý dữ liệu, từ đó chọn ra cột phù hợp để thuật toán đạt được độ chính xác cao nhất.

## BẢNG PHÂN CÔNG CÔNG VIỆC

| | Nguyễn Hải Đăng | Nguyễn Tiến Đạt | Lê Anh Duy |
|---|---|---|---|
| Chọn và đánh giá dữ liệu | x | x | x |
| Mô tả dữ liệu, phát biểu bài toán | | x | |
| Tiền xử lý dữ liệu, loại bỏ các dữ liệu null, unknown | | x | |
| ID3, CART | x | | |
| Naïve Bayes, Random Forest | | | x |
| Đánh giá thời gian, độ chính xác, kết luận thuật toán | | | x |
| Dự báo | x | | |
| Báo cáo | x | x | x |

# TÀI LIỆU THAM KHẢO

1. Slide bài giảng môn Khai thác dữ liệu

2. Các bài tập thực hành môn Khai thác dữ liệu

3.