Group Information

Student Name	Student ID
蔡文婧 Wenjing, CAI	116010004
兰一帆 Yifan, LAN	116010103
高尔笛 Erdi, GAO	117010066
徐志豪 Zhihao, XU	117010330
涂毅磊 Yilei, TU	117010251

1. **Background**:

   The Fédération Internationale de Football Association (FIFA) is a non-profit organization which describes itself as an international governing body of association football, fútsal, beach soccer, and e-football. It is the highest governing body of football.

   There is also a video game called FIFA, which is a series of association football video games or football simulator, released annually by Electronic Arts under the EA Sports label. When the series began in late 1993, it was notable for being the first to have an official license from FIFA, the world governing body of football.

   Motivation:

   FIFA is a world-renowned football video game with a large video game player base. In reality, the number of football fans is also extremely large. A well-informed football player database and beautiful display interface can help the fans to understand the basic situation of each player's clubs and better participate in online and offline football matches. With enough information to support, fans can also predict the status of future players or events.

   Goal:

   We hope to make a database with GUI to show the basic information of each football star to the football fans or football event audiences, including their photos, nationalities, clubs, height, weight, position, etc. We will use dynamic radar chart to display player's ability value. Based on it, we plan to introduce some machine learning methods to recommend the appropriate position according to each player's ability value. Users can also change the player's ability value so that the system gives different recommendations, in order to guide the actual ball game or electronic game.

2. **Data Sources**:

Our data is scraped from https://sofifa.com/, which is a professional FIFA information website.

Attached is our Python WebCrawler code.

```
import pandas as pd
import re
import requests
from bs4 import BeautifulSoup
from tqdm import tqdm

# Get basic players information for all players
base_url = "https://sofifa.com/players?offset="
columns = ['ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag',
           'Overall', 'Potential', 'Club', 'Club Logo', 'Value', 'Wage', '
Special']
data = pd.DataFrame(columns=columns)

# basic info
for offset in tqdm(range(300)):
    url = base_url + str(offset * 61)
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text, 'html.parser')
    table_body = soup.find('tbody')
    for row in table_body.findAll('tr'):
        td = row.findAll('td')
        picture = td[0].find('img').get('data-src')
        pid = td[0].find('img').get('id')
        nationality = td[1].find('a').get('title')
        flag_img = td[1].find('img').get('data-src')
        name = td[1].findAll('a')[1].text
        age = td[2].text.strip()
        overall = td[3].text.strip()
        potential = td[4].text.strip()
        club = td[5].find('a').text
        club_logo = td[5].find('img').get('data-src')
        value = td[6].text.strip()
        wage = td[7].text.strip()
        special = td[8].text.strip()
        player_data = pd.DataFrame([[pid, name, age, picture, nationality,
                                     flag_img, overall, potential, club, c
lub_logo, value, wage, special]])
        player_data.columns = columns
        data = data.append(player_data, ignore_index=True)
data = data.drop_duplicates()
data.to_csv('basicdata.csv', encoding='utf-8-sig')
```

```
# store the basic data
data = pd.read_csv('basicdata.csv')

# Get detailed player information from player page
detailed_columns = ['Preferred Foot', 'International Reputation', 'Weak Fo
ot', 'Skill Moves', 'Work Rate', 'Body Type', 'Real Face', 'Position', 'Je
rsey Number', 'Joined', 'Loaned From', 'Contract Valid Until', 'Height', '
Weight', 'LS', 'ST', 'RS', 'LW', 'LF', 'CF', 'RF', 'RW', 'LAM', 'CAM', 'RA
M', 'LM', 'LCM', 'CM', 'RCM', 'RM', 'LWB', 'LDM', 'CDM', 'RDM', 'RWB', 'LB
', 'LCB', 'CB', 'RCB', 'RB', 'Crossing', 'Finishing', 'HeadingAccuracy',
                    'ShortPassing', 'Volleys', 'Dribbling', 'Curve', 'FKAc
curacy', 'LongPassing', 'BallControl', 'Acceleration', 'SprintSpeed', 'Agi
lity', 'Reactions', 'Balance', 'ShotPower', 'Jumping', 'Stamina', 'Strengt
h', 'LongShots', 'Aggression', 'Interceptions', 'Positioning', 'Vision', '
Penalties', 'Composure', 'Marking', 'StandingTackle', 'SlidingTackle', 'GK
Diving', 'GKHandling', 'GKKicking', 'GKPositioning', 'GKReflexes', 'ID']
detailed_data = pd.DataFrame(index=range(
    0, data.count()[0]), columns=detailed_columns)
detailed_data.ID = data.ID.values

player_data_url = 'https://sofifa.com/player/'
for i in tqdm(range(data.ID.shape[0])):
    id = data.ID[i]

    url = player_data_url + str(id)
    source_code = requests.get(url)
    plain_text = source_code.text
    soup = BeautifulSoup(plain_text, 'html.parser')
    skill_map = {}
    columns = soup.find('div', {'class': 'teams'}).find(
        'div', {'class': 'columns'}).findAll('div', {'class': 'column col-
6'})
    columns.append(soup.find('div', {'class': 'teams'}).find(
        'div', {'class': 'columns'}).findAll('div', {'class': 'bp3-
callout'})[0])
    for column in columns:
        skills = column.findAll('li')
        for skill in skills:
            if(skill.find('label') != None):
                label = skill.find('label').text
                value = skill.text.replace(label, '').strip()
                skill_map[label] = value
    meta_data = soup.find('div', {'class': 'meta'}).text.split(' ')
    length = len(meta_data)
    weight = meta_data[length - 1]
    height = meta_data[length - 2].split('\'')[0] + '\'' + \
        meta_data[length - 2].split('\'')[1].split('\"')[0]
    skill_map["Height"] = height
```

```
    skill_map['Weight'] = weight

    sections = soup.find('article').findAll(
        'div', {'class': 'column col-4'})[:-1]
    for section in sections:
        items = section.find('ul').findAll('li')
        for item in items:
            value = int(re.findall(r'\d+', item.text)[0])
            name = ''.join(re.findall('[a-zA-Z]*', item.text))
            skill_map[str(name)] = value
    for key, value in skill_map.items():
        detailed_data.loc[detailed_data.ID == id, key] = value

full_data = pd.merge(data, detailed_data, how='inner', on='ID')
full_data.to_csv('fifa19data.csv', encoding='utf-8-sig')
```

Here are some basic descriptions of each entity and attribute.

ID - unique id for every player
Name - name
Age - age
Photo - url to the player's photo
Nation ID – unique id for every country
Nation Name – nationality
Flag - url to players' country flag
Overall - overall rating
Potential - potential rating
Club ID - unique id for every club
Club Name - current club name
Logo - url to club logo
Value - current market value
Wage - current wage
Preferred Foot - left/right
International Reputation - rating on scale of 5
Weak Foot- rating on scale of 5
Skill Moves - rating on scale of 5
Position - position on the pitch
Height - height of the player
Weight - weight of the player
Crossing - rating on scale of 100
Finishing - rating on scale of 100
Heading Accuracy - rating on scale of 100
Short Passing - rating on scale of 100
Volleys - rating on scale of 100
Dribbling - rating on scale of 100

Curve - rating on scale of 100
FK Accuracy - rating on scale of 100
Long Passing - rating on scale of 100
Ball Control - rating on scale of 100
Acceleration - rating on scale of 100
Sprint Speed - rating on scale of 100
Agility - rating on scale of 100
Reactions - rating on scale of 100
Balance - rating on scale of 100
Shot Power - rating on scale of 100
Jumping - rating on scale of 100
Stamina - rating on scale of 100
Strength - rating on scale of 100
Long Shots - rating on scale of 100
Aggression - rating on scale of 100
Interceptions - rating on scale of 100
Positioning - rating on scale of 100
Vision - rating on scale of 100
Penalties - rating on scale of 100
Composure - rating on scale of 100
Marking - rating on scale of 100
Standing Tackle - rating on scale of 100
Sliding Tackle - rating on scale of 100
GK Diving - rating on scale of 100
GK Handling - rating on scale of 100
GK Kicking - rating on scale of 100
GK Positioning - rating on scale of 100
GK Reflexes - rating on scale of 100

3. **Database Schema**: