Task 1: Data Collection and Preparation (20 Marks)

Name: Nagul Krishnan

Course : CSE &AI

1. Introduction

This report details the data collection, cleaning, and feature engineering processes for two datasets related to the FIFA 2022 World Cup: FIFA_2022_Full_Matches_Cleaned.csv (match-level data) and FIFA_2022_Team_Averages.xlsx (team-level summary statistics). The match-level dataset includes details such as dates, stages, teams, goals, winners, losers, and results, while the team averages dataset provides aggregated metrics like average goals scored, conceded, win/draw/loss rates, and goal differences. This report summarizes the data collection, cleaning, and feature engineering for both datasets, along with hypothetical scraper documentation for the match-level data.

2. Data Collection

The datasets likely originated from official FIFA records, sports websites, or APIs. The match-level dataset (FIFA_2022_Full_Matches_Cleaned.csv) includes 48 matches from the 2022 World Cup in Qatar, covering group stages to the final. The team averages dataset (FIFA_2022_Team_Averages.xlsx) summarizes performance metrics for 32 teams. Potential sources include:

- **FIFA Official Website**: Match reports and team statistics.
- **Sports APIs**: Platforms like SportsRadar or Opta for structured data.
- **Web Scraping**: Websites like ESPN, BBC Sport, or Wikipedia for match results and team summaries.

Hypothetical Scraper for Match-Level Data:

- **Source**: Wikipedia's 2022 FIFA World Cup page (e.g., match results table).
- **Tools**: Python with requests and BeautifulSoup for web scraping.
- **Process**:
    1. Fetch the webpage using requests.
    2. Parse HTML tables with BeautifulSoup to extract match details (Date, Stage, Home Team, Away Team, Goals, Result).

3. Derive Winner and Loser based on goals or penalty shootout notes.
4. Save to CSV.

- **Challenges**: Standardizing team names, handling penalty shootouts, and ensuring complete data.

Team Averages Data: The team averages dataset was likely derived from the match-level data by aggregating metrics per team. For example, Avg_Goals_Scored was calculated by summing a team's goals across all matches and dividing by the number of matches. This could have been done using a script or manual aggregation from the same sources as the match data.

3. Data Cleaning

Both datasets are clean and well-structured, requiring minimal additional cleaning.

Match-Level Dataset (FIFA_2022_Full_Matches_Cleaned.csv):

- **Characteristics**:
  - No missing values across 48 rows.
  - Consistent formatting: Dates in YYYY-MM-DD, team names standardized (e.g., "Usa"), goals as integers, and results including "Win (Penalties)" for shootouts.
  - Columns: Date, Stage, Home Team, Away Team, Home Goals, Away Goals, Winner, Loser, Result.
- **Cleaning Steps (Hypothetical)**:
  - Removed duplicates (none found).
  - Standardized team names (already done, e.g., "Usa" vs. "United States").
  - Ensured integer goals and datetime dates (already correct).
  - Normalized Result for penalty shootouts (already standardized).

Team Averages Dataset (FIFA_2022_Team_Averages.xlsx):

- **Characteristics**:
  - 32 rows, one per team, with columns: Team, Matches, Avg_Goals_Scored, Avg_Goals_Conceded, Win_Rate, Draw_Rate, Loss_Rate, Avg_Goal_Difference.
  - No missing values.

    - Numeric columns (e.g., Avg_Goals_Scored) are floats, with consistent precision.
    - Note: Win_Rate, Draw_Rate, and Loss_Rate are all 0 or contain decimal values, suggesting they may represent partial calculations or a specific subset of matches (e.g., group stage only). This may require clarification or correction.

- **Cleaning Steps Performed**:
    - Verified team names match between datasets (e.g., "Usa" in both).
    - Checked for outliers: Avg_Goal_Difference aligns with match data (e.g., Argentina's +0.83 is plausible given their 13 goals scored and 8 conceded over 6 matches).
    - Noted potential issue: Win_Rate, Draw_Rate, and Loss_Rate are 0 for many teams or inconsistent (e.g., Argentina's rates are 0 despite winning the tournament). This suggests the rates may not reflect the full tournament or were miscalculated. For this report, I assume these columns are placeholders or incomplete and recommend recalculating them from the match-level data.

Recalculating Rates (Example for Argentina): Using the match-level data, Argentina played 7 matches (6 wins, 1 draw in the final, resolved by penalties):

- Win Rate: 6/7 ≈ 0.857
- Draw Rate: 1/7 ≈ 0.143
- Loss Rate: 0/7 = 0 These values differ from the dataset's 0s, indicating a need for correction.

4. Feature Engineering

To enhance both datasets for analysis, the following features were conceptualized (not added to the provided files but could be implemented):

Match-Level Dataset:

1. **Goal Difference**: Home Goals - Away Goals (e.g., Qatar vs. Ecuador: -2 for Qatar).
2. **Match Type**: Group Stage (e.g., "Group A") vs. Knockout Stage (e.g., "Round of 16").
3. **Total Goals**: Home Goals + Away Goals (e.g., England vs. Iran: 8 goals).
4. **Is Draw**: Binary (1 for Draw, 0 otherwise).

5. **Penalty Shootout Indicator**: Binary (1 for "Win (Penalties)", 0 otherwise).
6. **Team Continent**: Map teams to continents (e.g., Argentina → South America) for regional analysis.

Team Averages Dataset:

1. **Corrected Rates**: Recalculate Win_Rate, Draw_Rate, and Loss_Rate using match-level data:
    - Win Rate = Wins / Matches
    - Draw Rate = Draws / Matches
    - Loss Rate = Losses / Matches
2. **Goal Efficiency**: Avg_Goals_Scored / Avg_Goals_Conceded to measure offensive vs. defensive performance.
3. **Knockout Stage Indicator**: Binary feature (1 if the team reached the knockout stage, 0 otherwise), derived from match-level data.
4. **Tournament Progression**: Categorical feature indicating the furthest stage reached (e.g., "Final", "Semifinal", "Group Stage").

Example Feature Engineering Code (Python):

```
# Step 1: Ensure openpyxl is installed for Excel export
try:
    import openpyxl
except ImportError:
    subprocess.check_call([sys.executable, "-m", "pip", "install", "openpyxl"])

# Step 2: Load your FIFA CSV
df = pd.read_csv("FIFA_2022_Full_Matches_Cleaned.csv")

# Step 3: Feature engineering for home team
home = df[["Home Team", "Home Goals", "Away Goals", "Result"]].copy()
home.columns = ["Team", "Goals Scored", "Goals Conceded", "Result"]
home["Win"] = home["Result"].apply(lambda x: 1 if x == "Home Win" else 0)
home["Draw"] = home["Result"].apply(lambda x: 1 if x == "Draw" else 0)
home["Loss"] = home["Result"].apply(lambda x: 1 if x == "Away Win" else 0)

# Step 4: Feature engineering for away team
away = df[["Away Team", "Away Goals", "Home Goals", "Result"]].copy()
away.columns = ["Team", "Goals Scored", "Goals Conceded", "Result"]
away["Win"] = away["Result"].apply(lambda x: 1 if x == "Away Win" else 0)
away["Draw"] = away["Result"].apply(lambda x: 1 if x == "Draw" else 0)
away["Loss"] = away["Result"].apply(lambda x: 1 if x == "Home Win" else 0)

# Step 5: Combine home and away stats
all_matches = pd.concat([home, away], ignore_index=True)

# Step 6: Aggregate averages per team
team_avg = all_matches.groupby("Team").agg(
    Matches=("Team", "count"),
    Avg_Goals_Scored=("Goals Scored", "mean"),
    Avg_Goals_Conceded=("Goals Conceded", "mean"),
    Win_Rate=("Win", "mean"),
    Draw_Rate=("Draw", "mean"),
    Loss_Rate=("Loss", "mean")
).reset_index()

# Step 7: Add goal difference
team_avg["Avg_Goal_Difference"] = team_avg["Avg_Goals_Scored"] - team_avg["Avg_Goals_Conceded"]

# Step 8: Save to Excel
team_avg.to_excel("FIFA_2022_Team_Averages.xlsx", index=False)
print("✅ Excel file created: FIFA_2022_Team_Averages.xlsx")
```

5. Scraper Documentation (Hypothetical)

For the match-level dataset, a scraper could be designed as follows:

- **Source**: Wikipedia's 2022 FIFA World Cup page or a sports API.
- **Tools**: Python (requests, BeautifulSoup for web scraping; pandas for data handling).
- **Steps**:
    1. Fetch the webpage or API endpoint.
    2. Parse HTML tables or JSON responses to extract match details.

3. Standardize team names and handle penalty shootouts.
4. Save to CSV.
5.

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

url = "https://en.wikipedia.org/wiki/2022_FIFA_World_Cup"
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")

table = soup.select_one("table.match-results")
rows = table.find_all("tr")[1:]

data = []
for row in rows:
    cols = row.find_all("td")
    match_data = {
        "Date": cols[0].text.strip(),
        "Stage": cols[1].text.strip(),
        "Home Team": cols[2].text.strip(),
        "Away Team": cols[3].text.strip(),
        "Home Goals": int(cols[4].text.strip()),
        "Away Goals": int(cols[5].text.strip()),
        "Result": cols[6].text.strip()
    }
    data.append(match_data)

df = pd.DataFrame(data)
df.to_csv("fifa_2022_raw.csv", index=False)
```

6. Summary and Insights

- **Match-Level Dataset**: 48 matches, clean and consistent, with opportunities for feature engineering (e.g., goal difference, match type). Key insights include high-scoring matches (e.g., Spain 7-0 Costa Rica) and penalty shootout outcomes (e.g., Argentina's final win).

- **Team Averages Dataset**: 32 teams, with useful metrics like Avg_Goals_Scored (e.g., Portugal's 2.75) and Avg_Goal_Difference (e.g., Spain's +2.33). The Win_Rate, Draw_Rate, and Loss_Rate columns appear incomplete and should be recalculated.
- **Combined Insights**:
    - Top performers: Argentina (2.17 goals/match, +0.83 goal difference), France (2.67 goals/match, +1.5 goal difference).
    - Defensive strength: Morocco (0.67 goals conceded/match) and Spain (0.33 goals conceded/match).
    - Upsets: Saudi Arabia's 2-1 win over Argentina (match-level) aligns with their -0.5 goal difference, indicating a single strong performance.

7. Conclusion

Both datasets are clean and ready for analysis, with the match-level dataset providing granular insights and the team averages dataset offering summarized performance metrics. Feature engineering enhances analytical potential, and recalculating rates in the team averages dataset is recommended. A hypothetical scraper ensures reproducibility for match-level data collection. Future work could involve merging with player statistics or visualizing team performance (e.g., via charts of goal differences).