WEEK-1 REPORT

Project Title: FIFA 2026 Finalist Prediction
Task: Data Collection, Cleaning and Feature Engineering
Student: Naveen Kumar B N

Reg No: 24UG00321

Course: B.Tech CSE (AI)

University: Chanakya University

1. Objective

The purpose of Week 1 is to build a structured, clean dataset that combines World Cup match data and FIFA ranking data. This dataset will later be used to train predictive models for identifying teams most likely to reach the FIFA World Cup 2026 Final.

2. Datasets Used

Two authentic datasets were used from open sources:

Dataset	Source	Description
WorldCupMatches.csv	GitHub (Open Data 1930–2022)	Contains match-level data such as year, stage, teams, goals, and win conditions.
fifa_ranking-2024-06-20.csv	Kaggle (FIFA Official Ranking)	Contains FIFA rankings (1992–2024) with rank, points, and confederation details.

3. Data Cleaning Steps

- Selected relevant columns: Year, Stage, Teams, Goals, and Win Conditions.

- Removed rows with missing team names.

- Renamed columns for consistency (e.g., Home Team Name → Home_Team).

- Determined match winners using goal and penalty shootout data.

- Identified finalist teams from matches where Stage = Final.

- Processed FIFA ranking file: extracted year from rank_date, selected rank and points, averaged yearly values.

4. Feature Engineering

Feature	Description	Formula / Logic
Goals_For	Total goals scored by a team in that year	Sum(Home + Away goals scored)
Goals_Against	Total goals conceded	Sum(Home + Away goals allowed)
Matches_Played	Number of matches played	Count of matches
Goal_Difference	Net goal performance indicator	Goals_For − Goals_Against
Win_Rate	Attack efficiency ratio	Goals_For / (Goals_Against + 1)
FIFA_Rank	Average rank for that year	From ranking data
FIFA_Points	Average ranking points	From ranking data
Confederation	Team's regional group	UEFA, CONMEBOL, AFC, etc.
Is_Finalist	Target variable (1 = Reached Final)	From Final stage data

5. Data Merging

The cleaned match statistics were merged with the FIFA ranking data using 'Year' and 'Team' as keys. Duplicate rows were removed and missing ranks were dropped to create a consistent combined dataset.

6. Resulting Dataset

Output File: fifa_1930_2022_with_rank.csv

The merged dataset contains around 1000 records and 11 columns, covering data from 1930 to 2022.

Columns: Year, Team, Goals_For, Goals_Against, Matches_Played, Goal_Difference, Win_Rate, FIFA_Rank, FIFA_Points, Confederation, Is_Finalist

7. Observations

- Teams with high Goal Difference and Win Rate usually have low FIFA Ranks (strong performance).

- UEFA and CONMEBOL confederations dominate the finalist positions.

- Dataset is now clean and ready for model training.

8. Conclusion

Week 1 was completed successfully. The datasets were collected, cleaned, and merged to form a structured dataset suitable for machine learning. Important features such as Goal Difference, Win Rate, and FIFA Rank were engineered. The final dataset will be used in Week 2 for model training and prediction of 2026 FIFA World Cup finalists.

9.CODE

```
# ============================================================

# FIFA 2026 Finalist Prediction - Week 1 (Data Preparation)

# ------------------------------------------------------------

# Task: Data Collection, Cleaning, and Feature Engineering

# Output: fifa_1930_2022_with_rank.csv

# ============================================================

# STEP 1: Upload both CSV files separately

from google.colab import files

# STEP 2: Import required libraries

import pandas as pd

import numpy as np

print("Upload WorldCupMatches (1).csv")

uploaded = files.upload()

print("Upload fifa_ranking-2024-06-20.csv")

uploaded = files.upload()


# STEP 3: Load datasets

matches = pd.read_csv("WorldCupMatches (1).csv")

ranking = pd.read_csv("fifa_ranking-2024-06-20.csv")

print("Data loaded successfully.")

print("Matches Shape:", matches.shape)

print("Ranking Shape:", ranking.shape)


# STEP 4: Clean and prepare the matches dataset
```

```
matches = matches[['Year', 'Stage', 'Home Team Name', 'Away Team Name',
                'Home Team Goals', 'Away Team Goals', 'Win conditions']].dropna(
                subset=['Home Team Name', 'Away Team Name'])
matches.columns = ['Year', 'Stage', 'Home_Team', 'Away_Team',
                'Home_Goals', 'Away_Goals', 'Win_Conditions']

def match_result(row):
    if row['Home_Goals'] > row['Away_Goals']:
        return row['Home_Team']
    elif row['Home_Goals'] < row['Away_Goals']:
        return row['Away_Team']
    elif 'pen' in str(row['Win_Conditions']).lower():
        if row['Home_Team'] in row['Win_Conditions']:
            return row['Home_Team']
        else:
            return row['Away_Team']
    else:
        return 'Draw'

matches['Winner'] = matches.apply(match_result, axis=1)

# STEP 5: Identify finalist teams
finals = matches[matches['Stage'].str.contains('Final', case=False, na=False)]
finalist_teams = set(finals['Home_Team']).union(set(finals['Away_Team']))
print("Finalist teams identified:", len(finalist_teams))

# STEP 6: Compute yearly team statistics
home_stats = matches.groupby(['Year', 'Home_Team']).agg({
    'Home_Goals': 'sum', 'Away_Goals': 'sum'}).reset_index()
home_stats['Matches_Played'] = matches.groupby(['Year', 'Home_Team']).size().values
```

```
away_stats = matches.groupby(['Year', 'Away_Team']).agg({
    'Away_Goals': 'sum', 'Home_Goals': 'sum'}).reset_index()
away_stats['Matches_Played'] = matches.groupby(['Year', 'Away_Team']).size().values

home_stats.columns = ['Year', 'Team', 'Goals_For', 'Goals_Against', 'Matches_Played']
away_stats.columns = ['Year', 'Team', 'Goals_For', 'Goals_Against', 'Matches_Played']

team_stats = pd.concat([home_stats, away_stats]).groupby(['Year', 'Team']).sum().reset_index()

# STEP 7: Feature engineering
team_stats['Goal_Difference'] = team_stats['Goals_For'] - team_stats['Goals_Against']
team_stats['Win_Rate'] = np.round(team_stats['Goals_For'] / (team_stats['Goals_Against'] + 1), 2)

# STEP 8: Prepare and clean the FIFA ranking dataset
ranking.columns = [c.strip().lower() for c in ranking.columns]
if 'rank_date' in ranking.columns:
    ranking['year'] = pd.to_datetime(ranking['rank_date']).dt.year

ranking = ranking[['rank', 'country_full', 'total_points', 'confederation', 'year']]
ranking.columns = ['FIFA_Rank', 'Team', 'FIFA_Points', 'Confederation', 'Year']

ranking_yearly = ranking.groupby(['Year', 'Team']).agg({
    'FIFA_Rank': 'mean',
    'FIFA_Points': 'mean',
    'Confederation': 'first'
}).reset_index()

# STEP 9: Merge team stats with FIFA ranking
merged = pd.merge(team_stats, ranking_yearly, how='left', on=['Year', 'Team'])

# STEP 10: Label finalist teams (target variable)
```

```
merged['Is_Finalist'] = merged.apply(
    lambda x: 1 if x['Team'] in finalist_teams and x['Year'] in finals['Year'].values else 0,
    axis=1
)

# STEP 11: Final cleaning
merged = merged.dropna(subset=['FIFA_Rank'])
merged = merged.sort_values(['Year', 'FIFA_Rank']).reset_index(drop=True)

# STEP 12: Save and download
merged.to_csv("fifa_1930_2022_with_rank.csv", index=False)

print("Final dataset saved as fifa_1930_2022_with_rank.csv")
print("Shape:", merged.shape)
print("\nColumns:", merged.columns.tolist())

from google.colab import files
files.download("fifa_1930_2022_with_rank.csv")
```