CASE 02
ICC Men's Cricket World Cup - A Journey Through History

This case study is about analysing all the ICC Men's Cricket World Cup matches held from 1975-2023.

For the analysis and visualization, you will be provided a folder "WorldCup_Stats" which contains 13 csv files each corresponding to a World Cup series. Each file contains detailed match statistics in a uniform format. The table below outlines the key attributes included in the dataset:

Attribute	Description
date	The date the match was played (in YYYY-MM-DD format)
venue	The city where the match was played
match_category	The stage of the tournament (League-Match, Semi-Final, Final)
team_1	The name of the first competing team
team_2	The name of the second competing team
team_1_runs	The total runs scored by Team 1 in their innings
team_1_wickets	The number of wickets lost by Team 1 during their innings
team_2_runs	The total runs scored by Team 2 in their innings
team_2_wickets	The number of wickets lost by Team 2 during their innings
result	The outcome of the match ex: Team 1 won by X runs, Team 2 won by Y wickets
pom	The name of the player awarded "Player of the Match"
best_batters	A list of notable batters from the match, along with the runs scored ex: ['CJ Anderson - 75 runs ', 'BB McCullum - 65 runs ']
best_bowlers	A list of notable bowlers from the match, along with wickets taken Ex: ['SL Malinga - 3', 'Hamid Hassan - 3']
commentary_line	A commentary excerpt from the match
world_cup_year	The year of the World Cup edition
host_country	Country/s hosting the World Cup edition

Apart from the above, you will also receive an additional file "commentary_2023.csv". This contains all the commentary excerpts from the 2023 World Cup final match between India and Australia. This file should be used to complete Task 03.

TASK 01: Maintain a GitHub Repository

- From the beginning, create and maintain a GitHub repository for the project.
- Follow proper version control practices and GitHub etiquettes (ex: meaningful commits).
- We will limit our evaluation to the Python scripts and Jupyter notebooks present in the repository. Please ensure all your code is pushed promptly!
- Refer to the marking grid to ensure all necessary components are addressed for evaluation.

TASK 02: Data Preparation

To achieve the passing mark, the following tasks are mandatory. Implementing advanced techniques will earn extra credit. Carry out the below tasks in a Jupyter Notebook.

1. **Reading and combining data**
   - Load all 13 CSV files into a list.
   - Concatenate the files into a single DataFrame, named **crick_df**.

2. **Initial data exploration and cleaning**
   - Examine the DataFrame structure, including its features and data types.
   - Remove any duplicate records.
   - Remove null records if they exist.

3. **Handle outliers and missing values**
   - Perform outlier removal and missing value imputation only if necessary.
   - State the reason for any such actions (you can state the reasons within the notebook).

4. **Adding new columns to the DataFrame:**
   1. match_status
      - A string column indicating the final status of the match as either "abandoned" or "played."

      Hint: *Extract this information from the result column using appropriate processing steps.*
   2. winning_team
      - A string column indicating the winning team of the match.
      - If the match was abandoned, leave this column empty. Otherwise, derive the winning team from the result column.
   3. The two columns best_batters and best_bowlers contain values in a list format. You are required to create new variables to store each list element.
      - Ex: For best_bowlers if the value is ['SL Malinga - 3', 'Hamid Hassan - 3'], you will create 4 new columns with the values as follows:

best_bowler_1	best_bowler_1_wick	best_bowler_2	best_bowler_2_wick
"SL Malinga"	3	"Hamid Hassan"	3

- Similarly, split the best_batters column into new columns for batter names and runs scored.

***Hint**: At most, two best batters and two best bowlers are recorded for each match, so this step will introduce a total of 8 new columns to the dataframe. Please note that for some World Cup series, this stat is missing, ignore such.*

5. **Column Removal**
   - Drop the commentary_line column and any other irrelevant columns.

TASK 03: Deploying a HuggingFace Model

Complete this task in a separate Jupyter notebook. Treat it as an independent task, and there's no need to consider it in relation to the rest of the tasks.

- Read the data from commentary_2023.csv file. It has 375 commentary excerpts.
- Fit a hugging face model to detect the sentiment of each excerpt.
- Provide the rationale behind selecting the hugging face model.
- Add the sentiment to the dataset as a new column (name the column as "sentiment").
- Visualize the sentiment spread using a suitable chart.
- Ensure to push both the updated dataset and the notebook to the GitHub repo.

TASK 04: Dashboard Creation

- Design a dashboard using Plotly Dash that tells an insightful story with the data.
- Be SMART!!! There are many different charts you can use to visualize data. Refer to Plotly documentation to decide the best and most interactive charts to showcase your story.
- Refer to the marking grid to cover all required aspects.

SUBMISSION GUIDELINES

- **Python scripts and notebooks:** Push to a public GitHub repository
- **Dashboard:** Screen record and submit as a video
- **Presentation:** A maximum of 5 slides explaining what you did in the analysis
- Upload the below items to the Google Form (will be shared on the 5th of Dec):
  1. GitHub repository link (public)
  2. Video clip of the dashboard
  3. PowerPoint Presentation

Deadline: 6th of December 2024 11:59 PM

MARKING GRID

	Task	Weight	Evaluation Criteria - minimum requirements	
Git	**Maintain a git repo for the project**	10%	1.1	All the team members should be added to the project
			1.2	Maintain branches for each component/member
			1.3	At least two commits per member
			1.4	At least one completed pull request
			1.5	Make commits on-the-spot (not at the end)
			1.6	Maintain proper branch naming conventions
			1.7	Maintain meaningful commits
			1.8	Main branch should be free of conflicts
Pandas	**Data preparation**	30%	2.1	Read data files
			2.2	Merge files
			2.3	Remove duplicate/null records
			2.4	Impute missing values (only if required)
			2.5	Outlier removal (only if required)
			2.6	Pivoting / Grouping
NLP	**Deploying a Hugging Face model**	20%	3.1	Pick a suitable model
			3.2	Reliability of the model
Visualization	**Dash Dashboard**	40%	4.1	Use correct charts to represent data
			4.2	Include at least 5 different types of charts
			4.3	Call the charts to a dashboard
			4.4	Use interactive features on the dashboard (ex: filters)
			4.5	Clarity of the dashboard
			4.6	Story-telling

To pass, you must score at least 65% of the allocated marks in each section.

If you have any queries reach out to us via:

uvini.ranaweera@acuitykp.com
samujitha.senaratne@acuitykp.com