Low Level Design

FIFA WORLD CUP ANALYSIS

	Author 1
Document Version	0.3
Last Revised Date	

DOCUMENT CONTROL

Change Record:

VERSION	DATE	AUTHOR	COMMENTS

Reviews:

VERSION	DATE	REVIEWER	COMMENTS

Approval Status:

VERSION	REVIEW DATE	REVIEWED BY		APPROVED BY	COMMENTS

Contents

1. Introduction

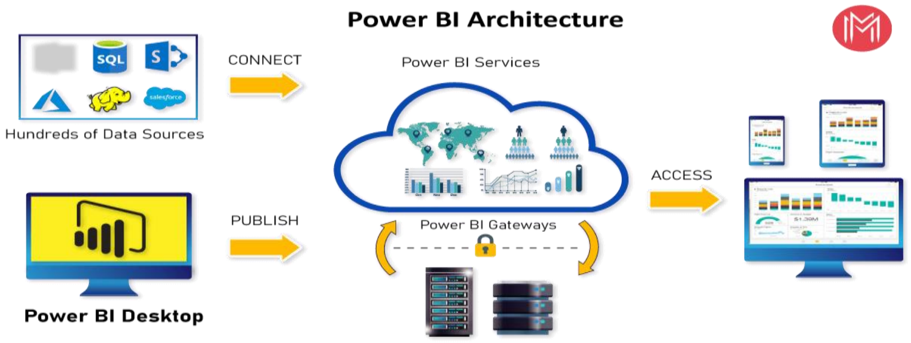
1.1 What is Low-Level design document?

- The purpose of this low-level design document is to outline the steps and components involved inanalyzing the FIFA World Cup dataset using Power BI and Python.
- The analysis aims to gain insights into various aspects of the World Cup, such as team performance,player statistics, match outcomes, and trends over the years.

1.2 Scope

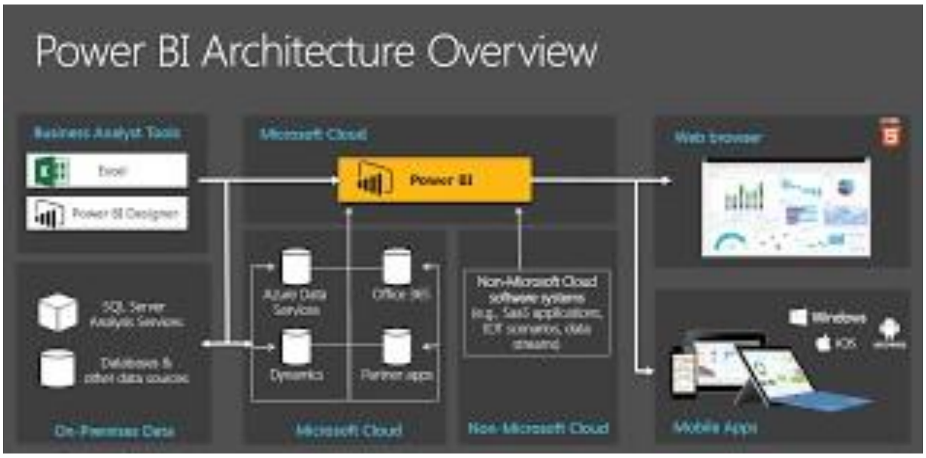
- Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organizationmay be defined during requirement analysis and then refined during data

2. Architecture



Power BI Architecture

- The architecture of Power BI involves several components working together to enable data connectivity, data transformation, data modeling, visualization, and collaboration. Here is an overview of the architectureof Power BI:
- The following diagram shows Powerbi architecture:



Power BI Desktop is a Windows application that provides a comprehensive set of tools for creating, modeling, analyzing, and visualizing data. Here is an overview of the architecture of Power BI Desktop:

User Interface

- Power BI Desktop has a user-friendly interface that allows users to perform various tasks related to data analysis and visualization.
- The interface includes different panes, such as the Report, Fields, Visualizations, and Properties panes, which provide access to different functionalities and options.

Data Sources

- Power BI Desktop supports a wide range of data sources, including databases, cloud services, Excel files, SharePoint lists, web APIs, and more.
- Users can connect to multiple data sources simultaneously to create a unified dataset for analysis.

Data Connectivity

- Power BI Desktop provides various connectivity options to access and retrieve data from different sources.
- DirectQuery: Enables real-time querying of data from on-premises or cloud-based databases.
- Import: Allows data to be imported into Power BI Desktop's in-memory engine for faster performance during data analysis.
- Live Connection: Establishes a connection to an external data source without importing the data, providing real-time access.

Data Modeling

- Power BI Desktop allows users to create a data model by defining relationships between different tables in the dataset.
- Users can import multiple tables from different data sources and define relationships based on common fields.
- Power BI Desktop automatically detects and suggests relationships based on the data model.

Data Transformation

- Power Query Editor, an integrated component within Power BI Desktop, enables users to connect to data sources, apply transformations, and shape the data before loading it into the data model.
- Power Query supports a wide range of data transformations, including filtering, merging, appending, pivoting, grouping, data type conversions, and more.
- Users can visually apply transformations using a user-friendly interface or write advanced transformation scripts using the M language.

Data Analysis

- Power BI Desktop provides a range of data analysis capabilities to explore and derive insights from the data.
- Users can create calculated columns, measures, and calculations using the DAX (Data Analysis Expressions) language.
- Advanced analytics features, such as forecasting, clustering, and regression, are available for data analysis.

- Power BI Desktop also supports the integration of Python and R scripts for advanced analytics and custom calculations.

Data Visualization

- Power BI Desktop offers a wide range of visualizations, including charts, graphs, maps, tables, and custom visuals.
- Users can drag and drop fields from the data model onto the canvas to create visualizations.
- Formatting options, interactions, and conditional formatting can be applied to enhance the visual appeal and convey insights effectively.
- Power BI Desktop supports interactive features like drill-down, filtering, highlighting, and cross-filtering to enable exploration of data.

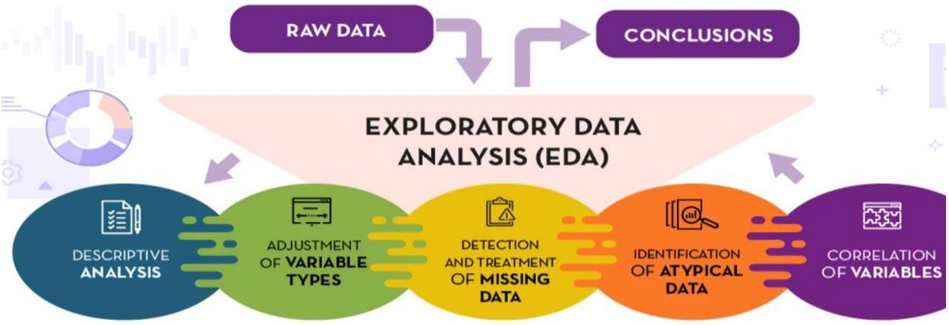
Collaboration and Sharing

- Power BI Desktop allows users to save their reports and dashboards as PBIX files, which can be shared with others who have Power BI Desktop installed.
- PBIX files can be published to the Power BI Service, a cloud-based platform, to enable collaboration and sharing with other users.
- Power BI Desktop also supports exporting reports in various formats, such as PDF, PowerPoint, Excel, and CSV.

Security and Governance

- Power BI Desktop provides options for securing data and reports through features such as row-level security and data encryption.
- Data privacy settings can be configured to control access to sensitive data.
- Administrators can define roles and permissions for users to ensure proper governance of reports and data.

Python EDA (Exploratory Data Analysis) typically follows a modular and iterative architecture. Here is a high-level overview of the architecture for Python-based EDA


RAW DATA
CONCLUSIONS
EXPLORATORY DATA ANALYSIS (EDA)
DESCRIPTIVE ANALYSIS
ADJUSTMENT OF VARIABLE TYPES
DETECTION AND TREATMENT OF MISSING DATA
IDENTIFICATION OF ATYPICAL DATA
CORRELATION OF VARIABLES

Data Loading

- Load the dataset into Python using appropriate libraries such as pandas, NumPy, or CSV readers.
- Handle various file formats like CSV, Excel, JSON, or databases using relevant libraries or modules.

Data Cleaning and Preprocessing

- Perform data cleaning tasks to handle missing values, outliers, and inconsistencies.
- Handle data quality issues, such as correcting errors, standardizing formats, or removing duplicates.
- Apply data transformations like feature scaling, normalization, or encoding categorical variables.
- Remove irrelevant or redundant features that do not contribute to the analysis.

Data Exploration

- Perform initial exploratory data analysis to gain insights into the dataset.
- Calculate summary statistics, such as mean, median, mode, or standard deviation.
- Generate basic visualizations like histograms, scatter plots, box plots, or correlation matrices.
- Identify patterns, trends, or outliers that may require further investigation.

Feature Engineering

- Create new features from existing data that could potentially enhance the analysis.
- Perform feature extraction, such as deriving datetime features, extracting text information, or engineering new variables.
- Use domain knowledge and statistical techniques to create meaningful features.

Statistical Analysis

- Apply statistical techniques and tests to gain deeper insights into the data.
- Conduct hypothesis testing, such as t-tests or chi-square tests, to validate assumptions or evaluate significance.
- Perform correlation analysis, ANOVA, or regression analysis to examine relationships between variables.
- Utilize statistical models or algorithms for predictive analysis or classification tasks.

Advanced Visualization

- Generate more sophisticated visualizations using libraries such as Matplotlib, Seaborn, or Plotly.
- Create advanced plots like heatmaps, time series plots, violin plots, or interactive visualizations.
- Visualize complex relationships or patterns in the data using appropriate techniques.

Iteration and Refinement

- Continuously iterate through the previous steps to refine the analysis.
- Explore different subsets of the data or adjust preprocessing techniques based on insights gained.
- Incorporate feedback from stakeholders or domain experts to improve the analysis.

Documentation and Reporting

- Document the findings, insights, and key observations during the EDA process.
- Prepare reports or presentations summarizing the analysis.
- Clearly communicate the results and share visualizations with stakeholders or team members.

3. Architecture Description

3.1. Data Description

The Dataset contains three Datasets containing World Cups Stats , World Cup Matches information and World Cup Players which can provide the insights like , Stadium Attendance, Successful Countries , Most goals Scored by the countries, Outcomes , Total Matches played and evolution of the tournament over the years.The following insights can be drawn from the datasets

- Most Number of World Cup Winning Title
- Number of Goal Per Country
- Attendance, Number of Teams, Goals, and Matches per Cup
- Goals Per Team Per World Cup
- Matches With Highest Number Of Attendance
- Stadium with Highest Average Attendance
- Which countries had won the cup ?
- Number of goal per country

Components

Power BI:

- A powerful data visualization and business intelligence tool that will be used to create interactive dashboards and reports.

Python:

- A programming language that will be used for data preprocessing, statistical analysis, and advanced calculations.

Data Preprocessing

- Load the FIFA World Cup dataset into Python for preprocessing.
- Handle missing values, inconsistencies, and data quality issues.
- Perform necessary data transformations, such as data type conversions, normalization, and standardization.
- Clean and prepare the dataset to ensure it is suitable for analysis.

Data Analysis

- Use Python to perform exploratory data analysis (EDA) on the preprocessed dataset.
- Extract key insights from the dataset, such as team performance metrics, player statistics, goals scored, etc.
- Apply statistical techniques to identify trends, patterns, and relationships within the data.
- Generate meaningful visualizations, such as histograms, scatter plots, bar charts, and heatmaps, to represent the analyzed data.

Power BI Integration

- Import the preprocessed and analyzed dataset into Power BI.
- Create a data model in Power BI by establishing relationships between relevant tables.
- Design interactive dashboards and reports using Power BI's intuitive interface.
- Utilize various visualizations, charts, and tables to present the analyzed data effectively.
- Incorporate filters, slicers, and drill-down options to enable dynamic exploration of the dataset.

Dashboard Components

- Team Performance: Visualizations depicting team rankings, historical performance, goals scored, etc.
- Player Analysis: Statistics on top players, goals per player, player positions, and performance over the years.
- Match Insights: Interactive displays of match outcomes, goals per match, average scores, etc.
- Tournament Trends: Trends analysis of goals, attendance, venues, and host countries over different World Cup editions.

User Interactions

- Enable user interactions with the Power BI dashboard.
- Allow users to filter and drill down into specific aspects of the dataset.
- Provide tooltips and contextual information to enhance the user experience.
- Allow users to export and share the generated reports and visualizations.

Deployment

- Publish the Power BI dashboard to a suitable platform for wider access.
- Share the insights and reports with stakeholders, such as sports analysts, FIFA enthusiasts, and World Cup fans.

7 LOW LEVEL DESIGN

4 .Conclusion

- This low-level design document describes the procedures for utilizing Power BI and Python to analyze the FIFA World Cup dataset.
- Meaningful insights can be drawn from the dataset by combining the capabilities of Power BI for data visualization and Python for data preprocessing and analysis.
- The resulting interactive dashboards and reports will offer useful information about team performance, player statistics, match outcomes, and tournament trends, facilitating a deeper understanding of the FIFA World Cup.