Data and Information Quality Project

PROJECT ID: 7
ASSIGNED DATASET: 2

Student_1: Jiaxiang Yi 10765316
Student_2: Yang Hao Mao 10705881

1. SETUP CHOICES

This section outlines the libraries and data preparation methodologies we have implemented, which are integral for efficient data handling, analysis, and model building.

1.1 Libraries Used

- **Pandas**: For data manipulation and analysis. Pandas was instrumental in handling data structures and performing operations to cleanse, filter, and transform the dataset.
- **NumPy**: Utilized for numerical operations, especially in data transformation phases where manipulation of numerical data was necessary.
- **Matplotlib**: Utilized for creating static and interactive visualizations in Python, essential for data exploration and results presentation.
- **Scikit-learn**: Though not explicitly mentioned, it could be implied that scikit-learn might be used for handling imputations and potentially for modeling if predictive analytics were to be involved post-preparation.
- **Seaborn**: A visualization library based on matplotlib, providing a high-level interface for drawing attractive and informative statistical graphics.
- **Regular Expressions**: This module is crucial for data cleaning, particularly in parsing and modifying textual data, which is common in data preprocessing.
- **TheFuzz** : This library, based on FuzzyWuzzy, helps in string matching and is particularly useful for data deduplication tasks.
- **Collections**: Utilized to create dictionaries with default values, which help manage data aggregations more seamlessly.
- **Record Linkage**: This toolkit is ideal for linking records within or across datasets, crucial for tasks such as deduplication and merging datasets.
- **YData Profiling**: Provides automated exploratory data analysis, generating detailed profiling reports that help quickly understand the structure, distribution, and relationships within the data.

- **Efficient Apriori**: Used for mining frequent itemsets and association rules from transactional data, valuable for uncovering relationships between variables in large sets.

1.2 Data Preparation Techniques

- **Data Profiling and Quality Assessment**:
  - Extensive use of the unique() function to understand the distribution of data across various columns, identifying unique values and potential irregularities.
  - Initial assessments to spot inconsistencies, such as formats or unexpected data types that could affect further analysis or modeling.
- **Data Transformation and Standardization**:
  - Splitting combined data fields (e.g., Contract dates) into more usable components like ContractStart and ContractEnd.
  - Standardizing date formats and numeric representations (removing currency symbols and units) to ensure uniformity across all data points.
  - Transformation of categorical data into binary columns (e.g., player positions) to facilitate analytical models that require numerical input.
- **Error Detection and Correction**:
  - Handling missing values through imputation, using zero for numeric columns where applicable, and placeholders like 'Unknown' for dates.
  - Outlier detection using statistical techniques like Interquartile Range (IQR), visual methods like Box Plots, and density-based methods such as KDE.
- **Data Deduplication**:
  - Employing fuzzy matching to detect and resolve near-duplicate entries, particularly in columns where textual data might have slight variations.
  - Utilizing both exact and non-exact matching techniques to ensure no duplicate records exist, enhancing the reliability of the dataset for subsequent analyses.

1. 3. Handling Numerical and Categorical Data

- Conversion of strings representing monetary values and physical measurements into pure numeric forms to facilitate quantitative analysis.
- Encoding categorical variables to reflect 'Yes' or 'No' for binary presence in multi-faceted attributes such as player positions, simplifying the complexity of the dataset.

1.4. Comprehensive Data Integrity Measures

- Combining multiple techniques to ensure the dataset is free from errors, outliers, and duplicates, thereby guaranteeing that the data is as accurate and reliable as possible for analysis.

1.5. Efficiency and Scalability Considerations

- Ensuring that the data preparation pipeline is not only thorough but also efficient enough to handle large datasets typically associated with real-world applications, like those derived from FIFA for this exercise.

2. PIPELINE IMPLEMENTATION

This section outlines the comprehensive steps undertaken to ensure data integrity and readiness for analysis in our pipeline. It covers data profiling, transformation, error correction, and deduplication processes, which are critical for preparing the dataset for predictive modeling and insights extraction.

2.1 Data Profiling and Data Quality Assessment

We performed a comprehensive data profiling of each column in the dataset. This included examining column names, the total number of rows, and the data types of each column. Using the `unique()` function, we explored the distinct values present in each column to gain a better understanding of the data distribution. Following this, we conducted an initial data assessment to identify any irregularities or inconsistencies. Once this preliminary analysis was complete, we proceeded to the data transformation phase to prepare the dataset for further analysis.

2.2.1 Data Transformation/Standardization

The first column we transformed was the **Contract** column. We split the original data format into two new columns: **ContractStart** and **ContractEnd**. After creating these new columns, we dropped the original **Contract** column. Additionally, we transformed the **OnLoan** information into a new column called **ContractStatus**. The original loan dates were incorporated into the newly created **ContractStart** and **ContractEnd** columns, while the **ContractStatus** column was used to indicate the player's contract status, distinguishing between loaned and non-loaned players.

We applied a similar transformation to the **Best Position** and **Position** column. During our analysis, we observed that many players had a combination of positions grouped together in this column. To address this, we created a new column for each unique position. For each player, we assigned a value of **Yes** or **No** in these columns to indicate whether they had previously played in that position, and his best position as **Best**.

We converted the **Joined** column, originally in datetime format, into a standardized **YYYY-MM-DD** format for consistency and ease of analysis.

Next, we addressed the columns containing monetary values, such as **Value**, **Wage**, and **Release Clause**, which were originally formatted with the "€" symbol. We transformed these columns to contain only numeric values and created new columns with the same names but appended with "(k€)" to indicate the values are in thousands of euros. After completing this transformation, we dropped the original columns.
We also processed the **Height** and **Weight** columns by removing the "cm" and "kg" units, respectively. Similar to the approach used for monetary columns, we created new columns named **Height (cm)** and **Weight (kg)** containing only numeric values, and then dropped the original columns. Before this, using the `unique()` function, we discovered that the **Height** column contained some values in inches instead of centimeters. We converted these values into centimeters to maintain consistency. Similarly, we found that the **Weight** column included some values in pounds (lbs). These were converted into kilograms to standardize the data.
We also cleaned the **W/F**, **SM**, and **IR** columns by removing the star symbol (★) from their values, ensuring the columns contain only numeric data.
The original **Hits** column contained values in formats like "3k". We converted all such values into pure numeric format.
The same approach was applied to the **Wage**, **Value**, and **Release Clause** columns mentioned earlier. Any values in formats like "3k" or "€3.5M" were converted into pure numeric values.

(*motivation on* ***Splitting Combined Data Fields into Separate Columns***: Often, data fields containing multiple pieces of information (like a start and end date combined) limit the analysis that can be performed. By splitting them, each component can be independently analyzed, which allows for more granular time-series analysis, trend detection over contract durations, or calculating the length of contracts. This separation improves the dataset's utility for tasks such as time-based querying or sorting.)

(*motivation on* ***Standardizing Dates and Numeric Formats***: Standardization of date formats ensures consistency in temporal data, critical for any operations that compare dates, compute durations, or align data across time periods. Numeric standardization of monetary values by removing currency symbols and converting text-based numeric notations (e.g., "3k" to 3000) ensures that mathematical operations can be accurately performed on these values, crucial for financial analysis or budgeting simulations.

(*motivation on* ***Transforming Positional Data into Binary Columns***: Transformation from categorical data to a more structured binary format enables easy filtering, aggregation, and application of machine learning models that require numerical input. It also simplifies the analysis, allowing quick checks on which positions a player has experience in, which can be vital for strategic planning in sports analytics.)

2.2.2 Error Detection and Correction Missing Values

After completing the data transformations, we moved on to the **error detection and correction** phase, addressing any missing values in the dataset to ensure data integrity and completeness.
To handle missing values in numeric columns, such as **Hits**, we used the `fillna()` single imputation technique. For this specific case, we filled the missing values with **0** to ensure consistency and maintain data usability.
We applied the same `fillna()` single imputation technique to the **ContractStart** and **ContractEnd** columns, filling missing values with the specific placeholder **'Unknown'** to indicate unavailable data.

(*motivation on **Handling Missing Values with Appropriate Imputation:*** Choosing zero for Hits assumes that no record means no hits, which is a common approach in scenarios where absence of data can logically imply a zero count. Using 'Unknown' for dates helps maintain data integrity, allowing analysts to distinguish between a lack of data and an actual date, important for maintaining data quality without falsely skewing temporal analyses.)

2.2.3 Error Detection and Correction Outliers

During the **outlier detection phase**, we utilized three main techniques: **Interquartile Range (IQR)**, **Box Plots**, and **Kernel Density Estimation (KDE)**. These methods allowed us to identify and analyze outliers effectively within the dataset.
We chose these techniques because our dataset, derived from FIFA, is primarily a statistics-based dataset containing extensive numerical information. We determined that methods like **IQR**, **Box Plots**, and **KDE** were well-suited for identifying and analyzing outliers in this context.

(*motivation on **Using IQR, Box Plots, and KDE:*** The Interquartile Range (IQR) is robust against outliers, making it suitable for initial outlier detection. Box plots provide a visual method to spot outliers, helping analysts visually confirm anomalies that IQR might suggest. Kernel Density Estimation (KDE) offers a way to understand the data's distribution shape, highlighting outliers in data distributions that aren't well-captured by simpler methods.)

2.2.4 Data Deduplication

In the data deduplication phase, we employed fuzzy matching techniques to meticulously explore potential duplications within the "**Club**" and "**Nationality**" columns. Despite our

thorough examination using a fuzzy clustering function `cluster_by_fuzzy` with similarity threshold=80 no significant duplicates were detected in these columns. Following this, we implemented both exact and non-exact matching methods to further scrutinize the dataset for duplicates. The exact matching approach did not reveal any duplicates, reinforcing the preliminary findings from our fuzzy matching analysis. However, to ensure comprehensive coverage and mitigate any oversights, we also engaged in non-exact matching by utilizing record linkage techniques.
Specifically, we applied the `Sorted Neighbourhood` method with an indexing strategy on the "Club" column, which facilitated a more manageable and focused review of potential duplicates. We then executed exact comparisons on other crucial attributes such as "**Age**," "**Height**," "**Weight**," and "**Nationality**." Additionally, to capture more subtle duplications that might not be evident through exact matching, we implemented a similarity scoring with a threshold of 0.95 on the "**LongName**" column.
Despite these extensive and refined techniques, our analysis concluded without identifying any probable duplicates. This rigorous approach ensured the integrity of our dataset, confirming that the data was free from duplications and ready for subsequent analysis stages.

(*motivation on **Using Fuzzy Matching and Record Linkage:*** Fuzzy matching addresses the common issue of textual discrepancies in data entries, such as slight misspellings or variations in naming conventions, which are common in large datasets. Record linkage complements this by identifying duplicates based not only on direct matches but also on probabilistic matching, crucial for ensuring the dataset's uniqueness without losing valuable data due to stringent matching criteria.)

3. RESULTS

The data quality assessment demonstrates that all columns meet the desired level of completeness, with a completeness score of 1.000000 for each column. This confirms that there are no null or missing values in the dataset, ensuring the dataset's reliability and usability.
The dataset demonstrates a high level of consistency, with all columns achieving 100% completeness and no missing or null values, ensuring its reliability for analysis. The variability across the data aligns with expectations, as attributes like **Preferred Foot** show limited options while columns such as **Joined** exhibit broader diversity. Repeated values in columns like **Height(cm)** and skill metrics reflect expected clustering within defined ranges. However, attributes like **Defending** and goalkeeping metrics display low uniqueness, which is typical for data structured around finite categories. Overall, the dataset is well-prepared for analysis, with potential for refinement in areas where repeated values might benefit from further differentiation depending on specific project goals.

In addition to the standard data quality assessment, we conducted several extra checks to ensure the dataset's integrity and consistency. These included verifying whether each

player belongs to multiple clubs and checking for conflicting club entries for the same player, ensuring there were no discrepancies in player-club relationships. We also examined the distribution of player counts across clubs to identify any anomalies, such as unusually high or low numbers of players per club.

For players without a club, we reviewed their associated attributes, including wage, value, release clause, and contract information, to ensure logical consistency. Furthermore, we validated contract data to confirm accuracy and alignment with player statuses. Finally, we performed checks on player statistics to ensure they were within reasonable and expected ranges, detecting any outliers or errors that could indicate data issues. These additional validations strengthen the dataset's readiness for analysis and its ability to support accurate insights.