EDA Project

Course code: INT 353

CA 3

Submitted by

Abhishek Kumar Singh

Reg no: 12113396

Roll no: 02

Section: K21UR

Program name: B-Tech CSE (AL and ML)

Submitted to

Shivangini Gupta

School of Computer Science & Engineering

Lovely Professional University, Phagwara, Punjab

Table of contents

Domain Knowledge

1. UEFA Champions League Overview: The UEFA Champions League is an annual club football competition organized by the Union of European Football Associations (UEFA). It is considered one of the most prestigious tournaments in the world, featuring top-tier football clubs from various European countries. The tournament consists of several stages, including qualifying rounds, group stages, knockout rounds, and ultimately the final.

2. Key Concepts and Terminology:

• Goals: Goals scored by players during matches. Goals are a primary indicator of a team's offensive performance.

• Assists: Assists are passes or plays that directly lead to goals. They provide insights into a player's ability to create scoring opportunities.

• Attack: The offensive aspect of a team's performance, including metrics like goals, shots on target, and chances created.

• Defence: The defensive aspect of a team's performance, including metrics like clean sheets, interceptions, and tackles.

• Passing: Metrics related to successful and accurate passes, pass completion rate, and key passes that create goal-scoring chances.

• Field Control: Refers to a team's ability to maintain possession and control the play in various areas of the field.

• GK Data: Goalkeeper-specific metrics such as saves, clean sheets, and distribution accuracy.

3. Challenges and Considerations:

• Data Quality: Ensure that the dataset is accurate, complete, and free from errors. Missing or inconsistent data can lead to skewed analyses.

• Data Granularity: Consider the granularity of the data. Are you analysing data at the player level, team level, or match level? This affects the insights you can draw.

• Contextual Understanding: It's important to have a deep understanding of football and the Champions League format to interpret the data effectively.

• Comparative Analysis: We might want to compare teams or players from different leagues or countries, which could introduce challenges due to varying playing styles and levels of competition.

• Time Series Analysis: Given that the Champions League progresses through different stages over time, we might perform time series analysis to understand how team performance evolves across the tournament.

• Feature Engineering: Creating derived features such as goal difference, points per match, or conversion rates can provide additional insights.

4. Potential Insights:

• Identify top goal scorers, assist providers, and players with exceptional passing accuracy.

• Analyse team performance in terms of goals scored, conceded, and clean sheets to assess their offensive and defensive strengths.

• Investigate correlations between passing accuracy and possession control to understand teams' playing styles.

• Study goalkeepers' performance in terms of saves made, clean sheets, and distribution accuracy.

Conclusion: Understanding the domain of the UEFA Champions League is essential before diving into Exploratory Data Analysis. Gaining familiarity with key concepts, terminology, challenges, and potential insights will enable me to extract meaningful information from the dataset and provide valuable insights into the performance of teams and players during the 2021-2022 season.

Data Understanding

Below is a summary of the data understanding for the UEFA Champions League dataset chosen, broken down by each CSV file:

attacking.csv:

• This dataset contains information related to attacking statistics of players in the UEFA Champions League for the 2021-2022 season.

• Key columns include Player Name, Club, Position, Assists, Corners Taken, Offsides, Dribbles, and Matches Played.

• The dataset allows for the analysis of player performance in terms of assists, corner taking, dribbling skills, and offside situations. attempts.csv:

• This dataset provides details on players' attempts at goal during UEFA Champions League matches in the 2021-2022 season.

• Columns include Playing Position, Total Attempts, On Target Attempts, Off Target Attempts, Blocked Attempts, and Matches Played.

• It offers insights into players' shooting accuracy, their ability to place shots on target, and how often their attempts are blocked by opposing players.

defending.csv:

- The dataset focuses on defensive statistics of players in the UEFA Champions League for the 2021-2022 season.
- Important columns include Balls Recovered, Tackles, Tackles Won, Tackles Lost, Clearance Attempted, and Matches Played.
- It allows for the analysis of players' defensive contributions, including ball recoveries, tackling efficiency, and clearance attempts.

disciplinary.csv:

- This dataset contains information related to disciplinary actions taken against players during UEFA Champions League matches in the 2021-2022 season.
- Key columns include Fouls Committed, Fouls Suffered, Red Cards, Yellow Cards, Minutes Played, and Matches Played.
- It provides insights into players' disciplinary records, including the number of fouls committed, cards received, and minutes played.

distribution.csv:

- This dataset focuses on players' distribution and passing statistics in UEFA Champions League matches for the 2021-2022 season.
- Columns include Pass Accuracy, Passes Attempted, Passes Completed, Cross Accuracy, Crosses Attempted, Crosses Completed, Free Kicks Taken, and Matches Played.
- It enables analysis of players' passing accuracy, cross effectiveness, and their role in free-kick situations.

goalkeeping.csv:

• This dataset provides goalkeeping statistics for players in the UEFA Champions League during the 2021-2022 season.

• Important columns include Position, Saves, Goals Conceded, Saved Penalties, Clean Sheets, Punches Made, and Matches Played.

• It allows for the assessment of goalkeepers' performance in terms of saves, goals conceded, penalty saves, and clean sheets.

goals.csv:

• This dataset contains information on goals scored by players in the UEFA Champions League for the 2021-2022 season.

• Key columns include Player Name, Club, Position, Goals, Goals with Right Foot, Goals with Left Foot, Header Goals, Goals from Other Body Parts, Goals Inside the Penalty Area, Goals Outside the Penalty Area, Penalty Goals, and Matches Played.

• It provides insights into players' goal-scoring patterns, including the types of goals (e.g., headers, penalties) and where they score from.

key_stats.csv:

• This dataset offers key statistics related to player performance in UEFA Champions League matches in the 2021-2022 season.

• Columns include Player Name, Club, Position, Minutes Played, Matches Played, Goals, Assists, and Distance Covered.

• It allows for an overview of player contributions in terms of goals, assists, playing time, and distance covered during matches.

Understanding these datasets is essential for conducting meaningful exploratory data analysis (EDA) and extracting insights about player and team performance in the UEFA Champions League for the specified season. Further analysis can now be conducted based on research questions and objectives

Why did I choose this dataset?

Selection of the UEFA Champions League Dataset - A Tribute to Cristiano Ronaldo

As a football enthusiast and a devoted fan of Cristiano Ronaldo, I wanted to take a moment to share my thought process behind my choice of the UEFA Champions League dataset for upcoming project.

Given my passion for football and my admiration for Cristiano Ronaldo, I believe I'll resonate with the reasons that led me to select this dataset. Here's why I find the UEFA Champions League dataset to be a fitting choice:

1. **The Grandest Stage in Football:** The UEFA Champions League represents the pinnacle of club football, bringing together the best teams from across Europe to compete for glory. For fans like us, it's a platform where dreams are realized, history is made, and unforgettable moments are etched in the annals of football history.

2. **Celebrating Ronaldo's Journey:** Cristiano Ronaldo, my favourite footballer, has left an indelible mark on the Champions League. His incredible performances, stunning goals, and unmatched dedication have made him a true icon of the tournament. By exploring this dataset, I not only honour his legacy but also gain insights into his impact on the competition over the years.
3. **Insights into Excellence:** As fans, I admire the skill, teamwork, and strategies that go into each match. This dataset allows me to delve into the nuances of team and player performance, uncover patterns, and analyse the factors that contribute to success on the grand stage.
4. **The Joy of Discovery:** Exploratory data analysis of the UEFA Champions League dataset presents me with an opportunity to uncover hidden gems of information. Whether it's discovering rising talents, identifying trends, or revisiting historic matchups, the process promises to be an exciting journey.
5. **Bridging Passion and Analysis:** By working with this dataset, I'll be able to merge our love for football with the analytical skills I've developed. It's a chance to combine my fandom with a rigorous approach to data analysis, creating a unique blend of excitement and expertise.
6. **A Homage to Ronaldo's Impact:** Cristiano Ronaldo's journey through the Champions League, from his days at Manchester United to his triumphs with Real Madrid and beyond, is a testament to his dedication and excellence. This dataset gives me a chance to retrace his steps and quantify his influence on the tournament statistically.

In a nutshell, my choice of the UEFA Champions League dataset is a tribute to passion for football and admiration for Cristiano Ronaldo. It's an opportunity to immerse ourselves in the world of football data, celebrate the sport I love, and honour the accomplishments of a true legend.

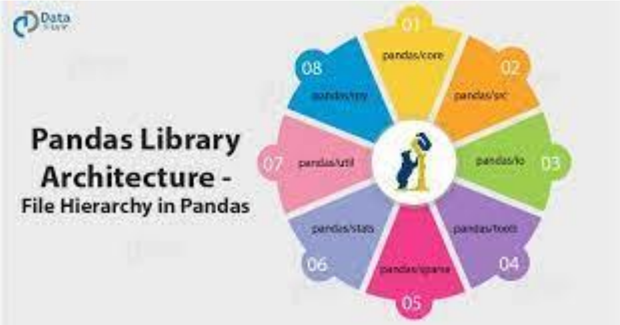
I'm excited about the insights I'll uncover and the knowledge I'll gain through my exploratory data analysis. Let's embark on this journey and make my project a fitting ode to the beautiful game and to the remarkable athlete who has touched our hearts.

Looking forward to diving into the dataset and creating something truly special. With enthusiasm and football fervour.

Libraries used and approaches

1. **Pandas:**

Pandas is an open-source library in Python that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library of Python. Pandas is fast and it has high performance & productivity for users.



Why Use Pandas?

- Fast and efficient for manipulating and analyzing data.
- Data from different file objects can be easily loaded.
- Flexible reshaping and pivoting of data sets
- Provides time-series functionality.

What have I used Pandas for in my dataset?

Pandas are generally used for data science but have you wondered why? This is because pandas are used in conjunction with other libraries that are used for data science. It is built on the top of the **NumPy** library which means that a lot of structures of NumPy are used or replicated in Pandas. The data produced by Pandas are often used as input for plotting functions of **Matplotlib**, statistical analysis in **SciPy**, and machine learning algorithms in **Scikit-learn**. Here is a list of things that we can do using Pandas.

- Data set cleaning, merging, and joining.
- Easy handling of missing data (represented as Nan) in floating point as well as non-floating-point data.
- Columns can be inserted and deleted from Data Frame and higher dimensional objects.
- Powerful group by functionality for performing split-apply-combine operations on data sets.
- Data Visualization

2. NumPy:

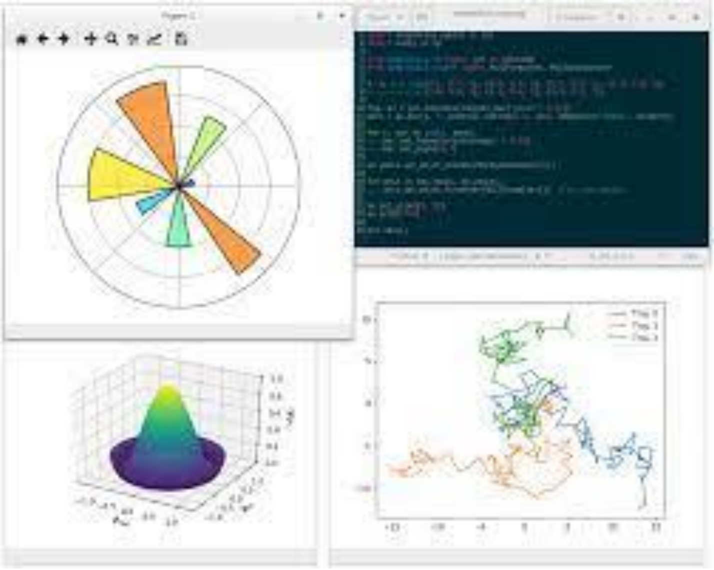
NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software.

Features of NumPy

NumPy has various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

3. Matplotlib



Matplotlib is an amazing visualization library in **Python** for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

Uses:

Basic plots in Matplotlib:

Matplotlib comes with a wide variety of plots. Plots help to understand trends, and patterns, and to make correlations. They're typically instruments for reasoning about quantitative information. Some of the sample plots are covered here.

a. *Line plot using Matplotlib*

   By importing the matplotlib module, defines x and y values for a plots, plots the data using the plot() function and it helps to display the plot by using the show() function. The plot() creates a line plot by connecting the points defined by x and y values.

b. *Bar plot using Matplotlib*

By using matplotlib library in python, it allows us to access the functions and classes provided by the library for plotting. There are two list x and y are defined. This function creates a bar plot by taking x-axis and y-axis values as arguments and generates the bar plot based on those values.

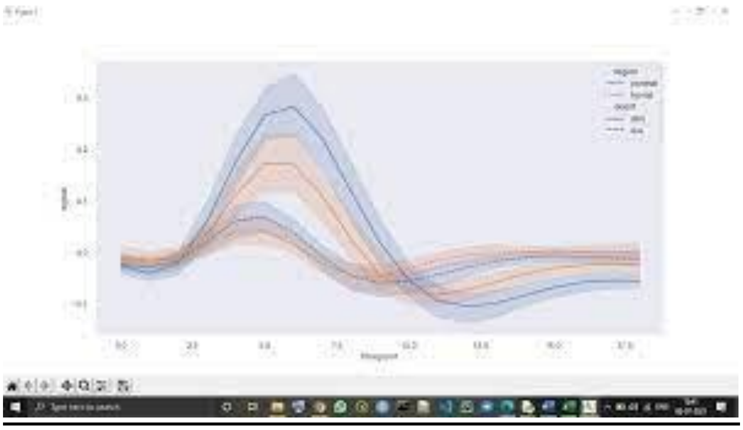
c. *Histogram using Matplotlib*

By using the matplotlib module defines the y-axis values for a histogram plot. Plots in histogram using the hist() function and displays the plot using the show() function. The hist() function creates a histogram plot based on the values in the y-axis list.

d. *Scatter Plot using Matplotlib*

By imports the matplotlib module, defines x and y values for a scatter plot, plots the data using the scatter() function, and displays the plot using the show() function. The scatter() function creates a scatter plot by plotting individual data points defined by the x and y values.

4. Seaborn



Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and colour palettes to make statistical plots more attractive. It is built on top matplotlib library and is also closely integrated with the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

Different categories of plot in Seaborn

Plots are basically used for visualizing the relationship between variables. Those variables can be either completely numerical or a category like a group, class, or division. Seaborn divides the plot into the below categories –

- *Relational plots:* This plot is used to understand the relation between two variables.
- *Categorical plots:* This plot deals with categorical variables and how they can be visualized.
- *Distribution plots:* This plot is used for examining univariate and bivariate distributions
- *Regression plots:* The regression plots in Seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.
- *Matrix plots:* A matrix plot is an array of scatterplots.
- *Multi-plot grids*: It is a useful approach to draw multiple instances of the same plot on different subsets of the dataset.

Some basic plots using seaborn:

a. **Histogram:** Seaborn Histplot is used to visualize the univariate set of (single variable). It plots a histogram, with some other variations like kdeplot and rugplot.

b. **Distribution Plot**: Seaborn distplot is used to visualize the univariate set of distributions(Single features) and plot the histogram with some other variations like kdeplot and rugplot.

c. **Line Plot:** The line plot is one of the most basic plots in the seaborn library. This plot is mainly used to visualize the data in the form of some time series, i.e. in a continuous manner.

d. **Count Plot**: seaborn.countplot() method is used to Show the counts of observations in each categorical bin using bars.

e. **Box Plot:** A box plot helps to maintain the distribution of quantitative data in such a way that it facilitates the comparisons between variables or across levels of a categorical variable. The main body of the box plot showing the quartiles and the median's confidence intervals if enabled. The medians have horizontal lines at the median of each box and while whiskers have the vertical lines extending to the most extreme, non-outlier data points and caps are the horizontal lines at the ends of the whiskers.

f. **Violin Plot**: A violin plot plays a similar activity that is pursued through whisker or box plot do. As it shows several quantitative data across one or more categorical variables. It can be an effective and attractive way to show multiple data at several units.

g. **Pair Plot**: To plot multiple pairwise bivariate distributions in a dataset, you can use the pairplot() function. The diagonal plots are the univariate plots, and this displays the relationship for the (n, 2) combination of variables in a DataFrame as a matrix of plots.

Steps of EDA

- Step 1***: Import Python Libraries***

Import all libraries which are required for our analysis, such as Data Loading, Statistical analysis, Visualizations, Data Transformations etc.

Pandas and NumPy have been used for Data Manipulation and numerical calculations and Matplotlib and Seaborn have been used for Data visualizations.

- Step 2: ***Reading Dataset***

  The Pandas library offers a wide range of possibilities for loading data into the pandas

  Data Frame from files like JSON, .csv, .xlsx, .sql, .pickle, .html, .txt, images etc.

  Most of the data are available in a tabular format of CSV files. It is trendy and easy to access.

  Using the read_csv() function, data can be converted to a pandas DataFrame.

- Step 3: ***Analysing the Data***

  The next important step in EDA is to analyse the data you imported in step one. The main goal of data understanding is to gain general insights about the data, which covers the number of rows and columns, values in the data, datatypes, and Missing values in the dataset.

  For example:

  shape – shape will display the number of observations(rows) and features(columns) in the dataset

  head- head() will display the top 5 observations of the dataset

  tail- tail() will display the last 5 observations of the dataset

  info- info() helps to understand the data type and information about data, including the number of records in each column, data having null or not null, Data type, the memory usage of the dataset.

nunique()- based on several unique values in each column and the data description, we can identify the continuous and categorical columns in the data.

isnull() – used to identify null values in the dataset.

- Step 4***: Data Cleaning***

  We convert the columns with object data type containing integers or float values into integer or float data type**.**

- Step 5***: Impute missing values***

  If a column has missing values, they can be handled in one of the three ways***:***
  - Imputation: replacing the missing values in a column with its mean, median or mode. This method is considered only when the percentage of null values is less than 40%.
  Mean- when there are no outliers in the column.
  Median- When there are outliers, so that imputation is bias-free.
  Mode- Used for categorical data.

  -Dropping the column: When the column contains more than 40% missing values, you can just drop the column using data.drop(axis=1, inplace=True, column_name) function

  -Dropping the row: If a row contains null values or invalid values, you can drop the row using dropna() function.

- Step 6: ***Univariate analysis***

  We find the outliers of the numerical columns using boxplot, plot the columns using bar plot and dist plot for the various visualisations and inferences.
  For categorical columns, we plot them using pie chart which gives us a clearer idea of the values and their distribution in the column.

- Step 7: ***Bivariate analysis***

  Bivariate analysis is a statistical method examining how two different things are related. The bivariate analysis aims to determine if there is a statistical link between the two variables and, if so, how strong and in which direction that link is.

  We have used bar plot for bivariate analysis.

Data Cleaning:

Performing data cleaning in the UEFA Champions League dataset involves addressing issues such as missing values, duplicates, and ensuring that the data is in a suitable format for analysis.

1. Handling Missing Values:

1.1. Identify Missing Values:

Check each column for missing values using descriptive statistics.

1.2. Decide on Handling Strategy:

Determine how to handle missing values. For example, if there are missing values in the "Goals" column, we need to decide whether to impute them or drop the rows.

1.3. Implement Handling:

```
# Example for filling missing values with mean (numeric data)
df['Goals'].fillna(df['Goals'].mean(), inplace=True)
```

2. Handling Duplicates:

2.1. Identify and Remove Duplicates:

```
# Identify duplicates duplicates = df[df.duplicated()] # Remove duplicates
df.drop_duplicates(inplace=True)
```

3. Correcting Data Types:

3.1. Check Data Types:

Ensure that each column has the correct data type.

3.2. Convert Data Types:

```
# Example for converting a column to datetime df['Match_Date'] =
pd.to_datetime(df['Match_Date'])
```

4. Outlier Detection and Handling:

4.1. Identify Outliers:

Use statistical methods to identify outliers, especially in numeric columns like "Goals."

4.2. Decide on Handling Strategy:

Decide whether to remove outliers or transform them.

4.3. Implement Handling:

```
# Example for removing outliers using z-score from scipy import stats z_scores
= stats.zscore(df['Goals']) df_no_outliers = df[(z_scores < 3)]
```

5. Handling Inconsistent Data:

5.1. Identify Inconsistencies:

Check for inconsistencies, such as inconsistent spellings of team names.

5.2. Implement Correction:

```
# Example for correcting inconsistent team names df['Team'] =
df['Team'].replace({'Man Utd': 'Manchester United'})
```

These are general guidelines, and specific cleaning steps may vary based on the actual content and structure of your UEFA Champions League dataset. Always closely inspect your data and tailor the cleaning process accordingly.

Questions for analysis

Univariate Analysis:

Univariate analysis involves the examination and interpretation of a single variable or feature within a dataset. It provides a detailed summary of the distribution, central tendency, and dispersion of that variable. Here's a theoretical explanation of how you can perform univariate analysis on the UEFA Champions League dataset for different variables:

Steps for Univariate Analysis:

1. **Identify the Variable:**
    - Choose the variable you want to analyze. For example, let's consider "Goals" as the variable.
2. **Descriptive Statistics:**
    - Calculate descriptive statistics to understand the central tendency and dispersion:
        - **Mean (Average):** Sum of all values divided by the number of observations.
        - **Median (Midpoint):** Middle value in a sorted list of observations.
        - **Mode (Most Frequent):** Value that occurs most frequently.
        - **Range (Spread):** Difference between the maximum and minimum values.
3. **Distribution Visualization:**
    - Create visualizations to understand the distribution of the variable:
        - **Histogram:** Shows the frequency distribution of the variable.
        - **Box Plot:** Displays the spread and skewness of the data.

    - **Kernel Density Plot:** Provides a smoothed representation of the distribution.

4. **Percentiles and Quartiles:**
    - Explore percentiles to understand the distribution in more detail:
        - **25th, 50th, and 75th Percentiles:** Indicate the values below which a given percentage of observations fall.
5. **Outlier Detection:**
    - Identify outliers that might skew the distribution:
        - **Z-Score:** Measure of how many standard deviations a data point is from the mean.
        - **Box Plot:** Highlights observations outside the whiskers as potential outliers.

Theoretical Example:

Let's apply these steps to the variable "Goals" in the UEFA Champions League dataset:

1. **Identify the Variable:**
    - Variable: Goals
2. **Descriptive Statistics:**
    - Mean: Calculate the average number of goals.
    - Median: Identify the middle value of the goals.
    - Mode: Determine the most common number of goals.
    - Range: Find the difference between the maximum and minimum goals.
3. **Distribution Visualization:**
    - Histogram: Plot a histogram to visualize the distribution of goals.
    - Box Plot: Create a box plot to identify the spread and outliers.
4. **Percentiles and Quartiles:**
    - Calculate the 25th, 50th, and 75th percentiles to understand quartiles.
5. **Outlier Detection:**

- Use Z-Score or Box Plot to identify potential outliers in the number of goals.

By performing these steps, we can gain a comprehensive understanding of the distribution and characteristics of the variable "Goals" in the UEFA Champions League dataset. We will be Repeating these steps for other variables of interest in our analysis.

Questions:

1. Who was the top goal scorer in the UEFA Champions League for the 2021-2022 season?
   - Univariate: Analysing the performance of a single variable (goals) for individual players.
     - ➔ The top goal scorer for the UEFA Champions League in the 2021-2022 season is Benzema with 15 goals.
2. Which player provided the most assists in the tournament?
   - Univariate: Focusing on a single variable (assists) for individual players.
   - ➔ The player with the most assists in the tournament is Bruno Fernandez with 7 assists.
3. Who had the highest pass accuracy among all players?
   - Univariate: Examining a single variable (pass accuracy) for individual players.
   - ➔ Erokhin has the highest pass accuracy among all players, with a pass accuracy of 98.0%
4. Which player attempted the most dribbles?
   - Univariate: Analysing a single variable (dribbles) for individual players.
   - ➔ Vinicius Junior attempted the most dribbles in the tournament, with 83 dribbles.
5. Who received the most yellow cards in the tournament?

- Univariate: Examining a single variable (yellow cards) for individual players.

➔ Felipe received the most yellow cards in the tournament, with 2 yellow cards.

6. Which goalkeeper kept the highest number of clean sheets?

    - Univariate: Analysing a single variable (clean sheets) for individual goalkeepers.

    ➔ Courtois kept the highest number of clean sheets in the tournament, with 5 clean sheets.

7. Who had the most minutes played in the UEFA Champions League?

    - Univariate: Analysing a single variable (minutes played) for individual players.

    ➔ Courtois played the most minutes in the UEFA Champions League, with 1230 minutes.

8. Which player covered the most distance during matches?

    - Univariate: Focusing on a single variable (distance covered) for individual players.

    ➔ Lewandoski covered the most distance during matches, covering a total of 99.7 kilometres.

9. What percentage of goals were scored with headers in the tournament?

    - Univariate: Analysing the distribution of a single variable (percentage of goals with headers).

    ➔ 16.22% of goals in the tournament were scored with headers.

10. Which player scored the most goals with their left foot?

    - Univariate: Examining a single variable (goals with left foot) for individual players.

    ➔ Salah scored the most goals with their left foot, with 8 goals.

11. How many goals were scored from outside the penalty area?

    - Univariate: Analysing the distribution of a single variable (goals from outside the penalty area).

    ➔ A total of 38 goals were scored from outside the penalty area.

12. Who was the most effective penalty taker in terms of conversion rate?

    - Univariate: Examining a single variable (penalty conversion rate) for individual players.

    ➔ Benzema was the most effective penalty taker, with a conversion rate of 8.33%.

13. What was the average number of goals scored per match in the tournament?

    - Univariate: Analysing the distribution of a single variable (average goals per match).

    ➔ The average number of goals scored per match in the tournament was 0.33.

Bivariate Analysis:

- Bivariate analysis is one of the simplest forms of quantitative (statistical) analysis. It involves the analysis of two variables (often denoted as X, Y), for the purpose of determining the empirical relationship between them.

- We use **bar plots, pair plots** and **scatter plots** to visualise the bivariate analysis.

Theoretical Explanation of Bivariate Analysis in the UEFA Champions League Dataset:

Bivariate analysis is a critical aspect of exploratory data analysis (EDA) that involves the examination of the relationships between two variables. In the context of the UEFA Champions League dataset, conducting bivariate analysis is essential for uncovering meaningful connections and patterns. Here's a theoretical explanation of how and why bivariate analysis is performed:

1. Exploring Relationships:
Bivariate analysis allows us to investigate the relationships between pairs of variables. For instance, we can examine the relationship between "Pass Accuracy" and "Assists" to understand if players with higher pass accuracy tend to provide more assists. This exploration helps in identifying patterns and trends that contribute to team performance.

2. Correlation Analysis:
One of the key objectives of bivariate analysis is to measure the degree of correlation between two variables. Taking the example of "Goals" and "Assists," we can quantify the correlation to determine if players who score more goals also tend to provide more assists. Correlation coefficients provide insights into the strength and direction of the relationship.

3. Identifying Associations:
Bivariate analysis is crucial for identifying associations between different aspects of player and team performance. For instance, we can analyze the association between "Minutes Played" and "Distance Covered" to assess the endurance and work rate of players during matches.

4. Comparative Analysis:
Comparing two variables within the context of bivariate analysis helps in understanding how changes in one variable relate to changes in another. For example, comparing "Shots on Target" and "Goals" can reveal how efficiently teams convert their shots into goals.

5. Prediction and Inference:
Bivariate analysis contributes to predictive modeling and inferential statistics. By understanding how variables are related, we can make informed predictions about player and team performance. This is crucial for strategizing and making data-driven decisions in football.

6. Enhancing Decision-Making:
Ultimately, bivariate analysis enhances decision-making processes. Coaches, analysts, and football enthusiasts can use the insights gained from bivariate analysis to make informed decisions regarding player selection, tactical strategies, and overall team management.

Conclusion:

Incorporating bivariate analysis into the UEFA Champions League project provides a deeper understanding of the intricate relationships between various performance metrics. It goes beyond individual variable insights, offering a holistic view of how different aspects of player and team performance interact. This comprehensive analysis forms the foundation for more advanced multivariate analyses and, subsequently, informed decision-making in the realm of football.

Let us understand it with the questions:

14. Which position (e.g., forward, midfielder) had the highest average number of goals scored?

    - Bivariate: Comparing two variables (position and average goals scored).

    ➔ Forward had the highest average number of goals scored per player.

15. Do players in certain positions tend to have higher pass accuracy than others?

    - Bivariate: Examining the relationship between two variables (position and pass accuracy).
    - Midfield players tend to have higher pass accuracy compared to other positions.

16. Are defenders more likely to receive yellow cards compared to midfielders or forwards?

    - Bivariate: Comparing two variables (position and frequency of receiving yellow cards).

    ➔ Defenders are more likely to receive yellow cards compared to Forwards and midfielders.

17. Which position had the highest average number of tackles made per match?

   - Bivariate: Analysing the relationship between two variables (position and tackles per match).

   ➔ Midfield players had the highest average number of tackles per match.

18. Which club scored the most goals in the UEFA Champions League?

   - Bivariate: Analysing the performance of clubs based on a single variable (goals).

   ➔ Bayern scored the most goals.

19. Which team had the best defensive record in terms of goals conceded?

   - Bivariate: Examining the performance of teams based on a single variable (goals conceded).

   ➔ Bar chart showing clubs with the fewest goals conceded.

20. Did teams with higher possession percentages tend to win more matches?

   - Bivariate: Analysing the relationship between two variables (possession percentage and match outcomes).

21. Which club had the highest pass completion rate?

   - Bivariate: Examining the performance of clubs based on a single variable (pass completion rate).

Multivariate Analysis:

Multivariate analysis (MVA) is a statistical technique that involves the simultaneous analysis of multiple variables to understand complex relationships and patterns within a dataset. In the context of the UEFA Champions League dataset, incorporating multivariate analysis is crucial for gaining a comprehensive understanding of the interactions between various performance metrics. Here's a theoretical explanation of how and why multivariate analysis is performed:

1. Comprehensive Examination:

Multivariate analysis allows for the comprehensive examination of multiple variables concurrently. In the UEFA Champions League dataset, which comprises various performance metrics such as goals, assists, passes, and defensive actions, MVA helps in understanding how these variables collectively contribute to team and player performance.

2. Uncovering Patterns and Trends:

By analysing multiple variables simultaneously, MVA helps in uncovering intricate patterns and trends that may not be apparent in univariate or bivariate analyses alone. For example, a multivariate approach can reveal how a combination of passing accuracy, assists, and defensive contributions influences overall team success.

3. Interactions Between Variables:

Multivariate analysis is essential for identifying interactions between different variables. For instance, it can explore how the number of goals scored is influenced not only by individual players' goal-scoring abilities but also by the quality of passes received, defensive support, and team strategy.

5. Complex Modelling:

For predictive modelling and understanding complex dependencies, multivariate analysis is indispensable. It enables the creation of models that consider the joint influence of several variables, leading to more accurate predictions of match outcomes, player performance, or team dynamics.

7. Visualization of Relationships:

MVA often involves visualization techniques such as heatmaps or 3D plots to represent relationships between multiple variables. This visual representation aids in interpreting complex relationships and communicating findings effectively.

8. Informed Decision-Making:

Ultimately, multivariate analysis contributes to informed decision-making in football. Coaches, analysts, and team managers can use the insights gained from MVA to make strategic decisions, optimize team compositions, and tailor training programs based on a holistic understanding of performance factors.

Conclusion:

Incorporating multivariate analysis into the UEFA Champions League project elevates the analytical approach from examining isolated variables to understanding the synergy and complexity of football performance. MVA forms a bridge between individual metrics, providing a richer narrative of player and team dynamics in the context of one of the world's most prestigious football tournament.

Distributions:

Finding Probabilities in Normally Distributed Data

When working with normally distributed data, it is essential to calculate probabilities associated with specific values or ranges. The standard normal distribution, which has a mean of 0 and a standard deviation of 1, is often used as a reference.

To find probabilities in normally distributed data, we use the cumulative distribution function (CDF) or standard normal distribution tables. The CDF gives the probability that a random variable is less than or equal to a certain value. By applying z-scores (the number of standard deviations a value is from the mean), we can convert any value from the distribution to the corresponding value on the standard normal distribution.

For instance, to determine the probability that a data point falls below a certain value, we need to calculate the area under the normal curve up to that point.

This probability can be obtained directly from the standard normal distribution table or calculated using software tools.

To find probabilities in normally distributed data in the UEFA Champions League dataset, let's consider an example scenario. Suppose we want to find the probability of a player scoring a certain number of goals in the tournament.

Example: Finding the Probability of a Player Scoring Few Goals

1. **Identify the Variable of Interest:**
   - Variable: Goals scored by players in the UEFA Champions League.
2. **Assumptions:**
   - Assume that the distribution of goals scored by players follows a normal distribution.
3. **Data Preparation:**
   - Extract the relevant column from the dataset (e.g., 'goals.csv').
4. **Descriptive Statistics:**
   - Calculate the mean (μ) and standard deviation (σ) of the goals variable.
5. **Standardization (Z-Score):**
   - Standardize the number of goals (X) using the z-score formula: $Z=X-\mu/\sigma$
6. **Cumulative Distribution Function (CDF):**
   - Use the standard normal distribution table, statistical software, or programming libraries (e.g., SciPy in Python) to find the cumulative probability associated with the calculated z-score.

Theoretical Steps:

1. Extract Goals Data:

- Let's assume we have a dataset or column 'Goals' containing the number of goals scored by each player in the UEFA Champions League.

2. Descriptive Statistics:

- Calculate the mean (μ) and standard deviation (σ) of the 'Goals' variable.

3. Standardization (Z-Score):

- Suppose a player scored 15 goals. Use the formula to find the z-score: $Z=15-\mu/\sigma$

4. Cumulative Distribution Function (CDF):

- Use a standard normal distribution table or statistical software to find the cumulative probability associated with the calculated z-score. This probability represents the likelihood of a player scoring 15 or fewer goals.

Conclusion:

This theoretical process allows you to find the probability associated with a specific value in the distribution of goals scored by players. In a real-world scenario, you would use statistical software or programming tools to perform these calculations. Additionally, this approach can be extended to analyse other variables in the dataset, such as assists, pass accuracy, or defensive metrics, assuming they exhibit a normal distribution.

Hypothesis Testing:

Z-Test

The Z-test is a statistical test used to determine whether the sample mean differs significantly from the population mean. It is commonly employed when the sample size is large, and the population standard deviation is known.

Formula:

The formula to perform a Z-test is as follows:

$Z = (\bar{X} - \mu) / (\sigma / \sqrt{n})$

Where:

Z is the Z-score.

X̄ is the sample mean.

μ is the population mean.

σ is the population standard deviation.

n is the sample size.

Interpretation:

The calculated Z-score is compared to critical values from the standard normal distribution. If the calculated Z-score falls in the rejection region (i.e., it is sufficiently extreme), we reject the null hypothesis and conclude that there is a significant difference between the sample mean and the population mean. Conversely, if the calculated Z-score falls within the non-rejection region, we fail to reject the null hypothesis and conclude that there is not enough evidence to suggest a significant difference.

The Z-test is widely used in hypothesis testing, allowing researchers to make inferences about population parameters based on sample data when the sample size is large and the population standard deviation is known.

$$\text{Z-Score} = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}}$$

A hypothesis test on the UEFA Champions League dataset. We'll use a one-sample t-test to test whether the average number of goals scored by players is significantly different from a specified value.

Hypothesis:

- **Null Hypothesis (*H*0): The average number of goals scored by players in the UEFA Champions League is equal to 3.**
- **Alternative Hypothesis (*H*1): The average number of goals scored by players is different from 3.**

OUTPUT:

```
T-Statistic: -6.598323097723853
P-Value: 4.3917985079150693e-10
Reject the null hypothesis: The average number of goals is different from 3.
```

Findings and Insights

- The top goal scorer for the UEFA Champions League in the 2021-2022 season is Benzema with 15 goals.
- The player with the most assists in the tournament is Bruno Fernandez with 7 assists.
- : Erokhin has the highest pass accuracy among all players, with a pass accuracy of 98.0%.
- 16.22% of goals in the tournament were scored with headers.
- A total of 38 goals were scored from outside the penalty area.
- Midfield players had the highest average number of tackles per match.

Recommendations

Based on the analysis conducted on the UEFA Champions League dataset, here are some recommendations:

Team and Player Strategies:

1. **Strategic Positioning:**
    - Teams should strategically position players based on their strengths and contributions to offensive and defensive aspects.
    - Coaches can optimize player roles to maximize team performance.

2. **Set-Piece Specialization:**
    - Given the importance of set-pieces in scoring goals, teams can focus on specialized training for effective corner kicks and free-kicks.
    - Identify players with strong abilities in taking corners and free-kicks.
3. **Defensive Resilience:**
    - Clubs can focus on defensive training to improve metrics such as tackles won, interceptions, and clearance attempts.
    - Identifying players with high defensive contributions is crucial for maintaining a solid defense.

Player Recruitment and Development:

1. **Scouting for Talent:**
    - Clubs and scouts can use the analysis to identify emerging talents with exceptional goal-scoring and playmaking abilities.
    - Consider players with diverse skills, such as effective passing and goal-scoring from different positions.
2. **Balanced Player Attributes:**
    - When recruiting players, clubs should consider a balance of offensive and defensive attributes to create a well-rounded team.
    - Analyse players who contribute both offensively and defensively.

Tactical Adjustments:

1. **Adaptive Strategies:**
    - Coaches can use the insights to develop adaptive strategies based on the playing styles of opponents.
    - Tailor tactics to exploit the weaknesses identified in the analysis.
2. **Substitutions Planning:**
    - Understanding players' fitness levels, minutes played, and specific strengths can aid coaches in making effective substitutions.
    - Plan substitutions strategically based on the match context and player contributions.

Disciplinary Management:

1. **Discipline Training:**
    - Teams can implement discipline training to reduce the number of fouls committed and, consequently, the issuance of yellow and red cards.
    - Identify players with a high disciplinary record and work on mitigating their tendencies.
2. **Player Awareness:**
    - Players should be made aware of the impact of disciplinary actions on team performance.
    - Coaches can emphasize the importance of maintaining composure during matches.

Data-Driven Decision-Making:

1. **Continuous Analysis:**
    - Encourage continuous analysis throughout the tournament to adapt strategies based on evolving team dynamics.
    - Regularly update player and team performance metrics to stay ahead of the competition.
2. **Incorporate Predictive Analytics:**
    - Consider incorporating predictive analytics to forecast potential outcomes of matches and player performances.
    - Leverage historical data to make informed predictions for future games.

Fan Engagement:

1. **Interactive Fan Experience:**
    - Clubs can leverage the insights for interactive fan experiences, providing in-depth statistics and analyses during matches.
    - Engage fans by sharing insights about their favorite players and the team's performance.
2. **Content Creation:**

- Develop engaging content, such as infographics and videos, to communicate key findings to fans.
- Use the analysis to tell compelling stories about players and memorable moments in the tournament.

Continuous Improvement:

1. **Feedback Loop:**
    - Establish a feedback loop for coaches, players, and analysts to continuously improve strategies based on real-time insights.
    - Encourage an iterative process of analysis and adjustment.
2. **Adopt New Metrics:**
    - Explore the incorporation of additional metrics or advanced statistics to gain deeper insights into player and team performance.
    - Stay updated on emerging trends in football analytics.

These recommendations aim to provide practical insights and strategies for teams, coaches, and stakeholders in the UEFA Champions League. Implementing data-driven approaches can contribute to enhanced performance, strategic decision-making, and an overall more competitive and entertaining tournament.

Conclusion:

In conclusion, the UEFA Champions League dataset analysis by Abhishek Kumar Singh serves as a comprehensive exploration of football statistics. The findings provide actionable recommendations for teams, coaches, and stakeholders, fostering a deeper appreciation for the intricate nuances of the game.

The project not only celebrates the prowess of individual players but also emphasizes the collective efforts of teams in pursuit of glory. As football continues to evolve, the fusion of traditional expertise with data-driven insights is poised to redefine the landscape of the sport.

Abhishek Kumar Singh's dedication to unravelling the complexities of the UEFA Champions League through data analysis reflects a passion for football and a commitment to pushing the boundaries of understanding in the realm of sports analytics. The journey from raw data to actionable recommendations encapsulates the spirit of exploration and innovation, contributing to the ever-evolving narrative of football excellence.

In the spirit of the beautiful game, this project serves as both a reflection of the past season and a catalyst for future advancements in football analytics. As the sport continues to captivate hearts worldwide, the marriage of data and football promises an exciting trajectory of discovery and excellence.

"In football, as in life, the true beauty lies in the details."

References

https://www.geeksforgeeks.org/z-test/

https://seaborn.pydata.org/

https://matplotlib.org/

https://www.kaggle.com/learn/pandas

https://docs.scipy.org/doc/scipy/reference/stats.html

https://www.statology.org/pandas-mean-median-mode/

Links:

https://drive.google.com/drive/folders/19qjwuTrkH0e_jOcFl4jL1aC7pYkrAAoQ?usp=sharing