Table of Contents

[1.INTRODUCTION: 1](#_Toc1653764242)

[1.1 PREREQUISITES AND LIBRARIES USED: 2](#_Toc1111030811)

[2.OBJECTIVES: 3](#_Toc1514697615)

[2.1 Score potential: 4](#_Toc1076156011)

[2.2 Win/Loss probability: 4](#_Toc985866796)

[2.3 Momentum Shift: 4](#_Toc201556377)

[2.4 Check and notify Match milestone: 4](#_Toc43331567)

[2.5 Game control function: 4](#_Toc905093478)

[2.6 True Score Model: 4](#_Toc169876903)

[3.METHODOLOGY: 4](#_Toc242211605)

[3.1 Functions used to perform analysis: 5](#_Toc2028242049)

[3.1.1 Dot ball functions: 5](#_Toc1484351443)

[3.1.2 Momentum shift functions: 5](#_Toc1100198495)

[3.1.3 Bowler’s milestone: 5](#_Toc167195609)

[3.1.4 Batsman’s milestone: 5](#_Toc219805995)

[3.1.5 Score potential notification: 5](#_Toc300006305)

[3.1.6 Win/Loss probability notification: 5](#_Toc1592986195)

[3.1.8 Game control function: 5](#_Toc1915843283)

[3.1.9 The “check\_and\_notify” function: 6](#_Toc1013711111)

[3.1.10 True Score function: 6](#_Toc2031086876)

[3.2 Data Retrieval: 6](#_Toc622280940)

[3.3 Data Processing: 6](#_Toc1900815775)

[3.4 Data Processing: 7](#_Toc393915822)

[3.5 Notification Message Construction: 7](#_Toc2012560996)

[3.6 Notification Sending 7](#_Toc1130449257)

[3.7 Check and notify season stats: 8](#_Toc2100494434)

[3.7.1 Bowler’s milestone: 8](#_Toc741069686)

[3.7.2 Batsman’s milestone: 8](#_Toc681853266)

[3.7.3 Momentum shift function: 8](#_Toc1156696465)

[3.8 Batsman’s milestone notification: 8](#_Toc1345123230)

[3.8.1 New Record Notification: 9](#_Toc650420108)

[3.8.2 Leader board Comparison: 9](#_Toc499247661)

[3.8.3 Season Milestone Notification: 9](#_Toc982731722)

[3.9 Bowler’s milestone: 9](#_Toc1362368167)

[3.9.1 Leaderboard Record Notification: 9](#_Toc491680250)

[3.9.2 Season Statistics Verification: 9](#_Toc1790456470)

[3.9.3 Seasonal Milestone Notification: Fans 10](#_Toc2014725501)

[3.10 Database Structure: 10](#_Toc2002641757)

[3.10.1 Database Tables: 10](#_Toc1523766404)

[3.11 Data Source: 11](#_Toc259933129)

[3.12 Data Collection Process: 12](#_Toc1122935027)

[3.13 Data Components: 12](#_Toc1483689577)

[3.13.1 Historic Data: 12](#_Toc1305982533)

[3.13.2 Player wise Stats: 12](#_Toc2128221185)

[4. LIVE IMPLEMENTATION AND INTEGRATION: 14](#_Toc1614046713)

[5. CHALLENGES ENCOUNTERED: 15](#_Toc2120234854)

[6. CONCLUSION: 15](#_Toc1753890539)

# 1.INTRODUCTION:

Cricket is a widely celebrated sport that has the charm to captivate fans on a global scale with its blend of strategy, athleticism, and suspense. We explore the implementation of automated notifications to enrich the spectator experience during cricket matches. These notifications serve to keep fans informed about key events such as wickets, boundaries, milestones, and match results, ensuring they stay engaged and up to date throughout the game. This guide elaborates how automated systems enhance viewer engagement and provide timely updates, thereby enhancing the overall spectator experience of cricket matches.

* Automated Milestone Notifications: We used the One signal API key to deliver alerts straight to an app developed so that viewers would be aware of player achievements during the cricket match. By providing real-time updates on significant events like 50s, 100s and noteworthy accomplishments, this improves viewer engagement and enriches the viewing experience.
* Estimating Win/Loss probabilities: The prediction of the results of a cricket match required multidimensional strategy. To confirm the likelihood of a team winning or losing, we used a Random Forest Classifier model that used a variety of match factors, team compositions, and past performance data.
* Prediction of Score potential: We used pitch conditions, player form, and past data, along with the capabilities of Artificial Neural Networks (ANNs), to forecast a batting team's probable score. Through extensive parameter tuning and the use of a Stacking Regressor.
* 4) Usage of data: We used the One signal API to quickly alert spectators about upcoming milestones that we predicted based on player statistics. Throughout the cricket match, our data-driven strategy guarantees prompt and accurate information on noteworthy player accomplishments.

.

## 1.1 PREREQUISITES AND LIBRARIES USED:

* port = mysql.connector
* import ‘pandas’ as ‘pd’
* from **‘onesignal\_sdk.client’** import **‘**Client’
* from **‘onesignal\_sdk.error’** import **‘OnesignalHTTPError’**
* import ‘os’
* import ‘sys’
* import ‘time’
* import ‘warnings’
* import ‘pickle’
* import ‘math’
* import ‘json’
* python   = 3.11.0
* pandas   = 2.2.2
* mysql-connector   = 2.2.9
* onesignal-sdk   = 2.0.0

# 2.OBJECTIVES:

2.1 Score potential: The model that is developed on ANN (Artificial Neural Network) which predicts the first innings score by leveraging data from various factors. We estimate the score by calculating the average of scores typically observed after the 7th and 8th overs of current matches. This method utilizes current data to project the likely progression of scoring based on early innings performance, providing a predictive framework that adjusts to new data from the ongoing match.

2.2 Win/Loss probability:The Win/Lose model functions as a predictive tool to determine the likely winner between teams, utilizing multiple factors and historical data. This model gives a probability graph depicting the likelihood of either team winning. If the graph shows a cross or a sudden change indicating a shift in the game's context, we quickly alert the app user, ensuring they are informed about significant developments during the match. This real-time notification system enhances fan engagement by providing timely updates on the evolving dynamics of the game.

2.3 Momentum Shift:If there is a difference greater than 15 in the momentum between the two teams, a notification containing the details of these occurrences will be sent to the spectators of the game. This ensures that spectators are promptly informed when significant developments arise simultaneously. Such notifications aim to keep spectators updated in real-time about concurrent events that may impact the game's dynamics or outcomes, thereby enhancing their engagement and understanding of key moments during the match. This proactive approach helps to enrich the overall viewing experience by providing timely and relevant information.

2.4 Check and notify Match milestone:Using real-time data from all players in the live match, our system continuously checks for significant events such as a player taking three wickets ("3 fer"), a batsman reaching fifty or hundred runs, instances of dot balls, and more. This automated process ensures prompt updates on these milestones as they occur during the match. Each function is made separately for both the innings.

2.5 Game control function:A notification will be sent after the 8th and the 15th over confirming who has control over the game based on momentum data.

The Game Control function analyses momentum data from every over between the 1st and 8th, and again from the 1st over to 15th onward, to update fans on which team is gaining the upper hand during the game. It records momentum shifts, identifying when one team surpasses the other by these intervals in both the first and second innings.

2.6 True Score Model: To assess the impact of each delivery by batsmen or bowlers based on the match context, an Impact Score is generated dynamically from the 1st over to the 20th over, updated after every ball. This process requires low latency, aiming to predict scores within 3 to 5 seconds after each delivery.

# 3.METHODOLOGY:

## 3.1 Functions used to perform analysis:

3.1.1 Dot ball functions:Analysing each delivery data to track dot balls and if there are three or more consecutive dot balls and the sequent delivery results in a wicket or a boundary (four or six), fans are promptly notified about these developments. This notification system keeps fans informed about key moments in the match, providing a narrative of the unfolding events in real time.

3.1.2 Momentum shift functions:If there is a difference greater than 15 in the momentum between the two teams, a notification containing the details of these occurrences will be sent to the spectators of the game. This ensures that spectators are promptly informed when significant developments arise simultaneously. Momentum shift is notified along with the events.

3.1.3 Bowler’s milestone:Keeping fans engaged with rapid updates on match events: Notifications are triggered when a bowler bowls more than 15 dot balls or takes three or more wickets during their spell. This ensures fans are promptly informed of significant occurrences during the match, enhancing their experience by providing timely and impactful updates on key performances and milestones.

3.1.4 Batsman’s milestone:Notifying the app user when a batsman approaches a half century or a century and if he does hit 50 or 100 with his Strike Rate, hits more than 10 fours, or hits more than 7 sixes during the match. This feature ensures that users receive timely updates on significant batting milestones, enhancing their engagement by keeping them informed of noteworthy performances as they occur.

3.1.5 Score potential notification:The model that is developed on ANN (Artificial Neural Network) which predicts the first innings score by leveraging data from various factors. We estimate the score by calculating the average of scores typically observed after the 7th and 8th overs of current matches. This method utilizes current data to project the likely progression of scoring based on early innings performance, providing a predictive framework that adjusts to new data from the ongoing match.

3.1.6 Win/Loss probability notification:The Win/Lose model functions as a predictive tool to determine the likely winner between teams, utilizing multiple factors and historical data. This model gives a probability graph depicting the likelihood of either team winning. If the graph shows a cross or a sudden change indicating a shift in the game's context, we quickly alert the app user, ensuring they are informed about significant developments during the match. This real-time notification system enhances fan engagement by providing timely updates on the evolving dynamics of the game.

3.1.8 Game control function:A notification will be sent after the 8th and the 15th over confirming who has control over the game based on momentum data.

The Game Control function analyses momentum data from every over between the 1st and 8th, and again from the 1st over to 15th onward, to update fans on which team is gaining the upper hand during the game. It records momentum shifts, identifying when one team surpasses the other by these intervals in both the first and second innings.

3.1.9 The “check\_and\_notify” function:Designed to facilitate the retrieval, merging, and preparation of live cricket match statistics for further analysis or notification. Here’s a breakdown of its functionality:

3.1.10 True Score function: To assess the impact of each delivery by batsmen or bowlers based on the match context, an Impact Score is generated dynamically from the 1st over to the 20th over, updated after every ball. This process requires low latency, aiming to predict scores within 3 to 5 seconds after each delivery.

We integrated a true score model that combines scoring potential and win/loss probabilities. This model identifies key performers who significantly influence the outcome of the match. Notifications are then sent to fans, highlighting these impactful players with updates on their batting and bowling contributions according to the true score model.

# 3.2 Data Retrieval:

* + Batting and Bowling Statistics: Fetches live batting and bowling statistics for a specified match ID from the database. These statistics include player names, unique IDs, and performance metrics such as runs scored, balls faced, wickets taken, and more.
  + Score Potential Data: Retrieves score potential information from a specific table based on the match ID. This data includes team names, innings numbers, over numbers, current scores, projected scores, win/loss probabilities for batting and bowling, and remaining wickets.
  + Historical and Current Season Data: Loads historical and current season statistics from Excel files (overall\_t20wc\_bat\_stats.xlsx, Current\_season\_batting.xlsx, overall\_t20wc\_bowl\_stats.xlsx, Current\_season\_bowling.xlsx). These files contain aggregated statistics of players' performances across various seasons.

# 3.3 Data Processing:

* + Merge Operations: Integrates live statistics with historical and current season data using pandas merge operations. This integration is performed separately for batting and bowling statistics to create comprehensive datasets for analysis.
* The “send\_notification” function is designed to send updates about a cricket match to a specified client. It fetches live match details from a database and incorporates them into a notification message. Key details included in the message are the current score, number of wickets fallen, and the name of the batting team.

Arguments:

* client (str): Identifier or destination for the notification.
* message (str): Base message content to which match details are appended.
* player\_name (str): Name of the player associated with the notification (not currently used).
* milestone (str): Milestone information related to the player or match (not currently used).
* title (str): Title or subject of the notification (not currently used).

Steps:

1. Database Connection**:** Connects to a database to retrieve live cricket match details.
2. SQL Query Execution: Executes an SQL query to fetch:
   * Batting team name
   * Inning number
   * Over number
   * Ball number
   * Total runs and extras for each ball
   * Type of wicket (if applicable)

# 3.4 Data Processing:

* + Computes cumulative scores (Current\_score) and wickets (wickets\_count) for both innings.
  + Filters and manipulates DataFrame to prepare match details for the notification.

# 3.5 Notification Message Construction:

* + Appends the latest batting team's score and wickets to the original notification message.
  + Constructs the final message to be sent to the client.

3.6 Notification Sending**:** Sends the constructed message to the specified client for update.

* This function assumes the existence of database connections and SQL query execution functions (connect\_database and execute\_query).
* The actual mechanism for sending notifications (client.send\_notification(message)) is not implemented in the function and should be added according to the specific notification platform or method used.

# 3.7 Check and notify season stats:

3.7.1 Bowler’s milestone:Keeping the fan engagement on high with quick updates when the bowler bowls more than 50 dot balls in the season and sets a record for the overall season with his wickets.

3.7.2 Batsman’s milestone:Fans are notified when a batsman scores over 500 runs in a tournament or hits more than 25 fours and 15 sixes throughout the tournament. The system checks the batsman's overall statistics and personal bests to alert fans of significant milestones such as reaching their 5th fifty, 2nd hundred, or achieving 600 runs in that tournament. This ensures fans stay updated with key player performances and milestones.

3.7.3 Momentum shift function:It is built separately for both batsman and bowlers. If the momentum shift is created because of batsman, then a notification is simultaneously along with the batsman’s name. The notification includes the boundaries scored. Similarly for bowlers, a notification is sent along with the number of wickets taken and dot balls bowled.

**THE STATISTICS ARE UPDATED IN THE OVERALL TOURNAMENT RECORDS AFTER EVERY MATCH. THE UPDATED DATA IS CONCOMITANTLY ARCHIVED IN A SEPARATE EXCEL SHEET.**

## 3.8 Batsman’s milestone notification:

Notifications are sent to end users when a batsman achieves a new record during the current season. These records are compared against the current season's benchmarks to determine if the batsman surpasses the overall leaderboard record. Additionally, fans receive notifications when a batsman reaches a season milestone, such as achieving the highest number of sixes, fours, runs, and other notable statistics.

3.8.1 New Record Notification: End users receive notifications whenever a batsman sets a record during the ongoing season. This ensures that fans are promptly informed of any significant achievements by their favourite players.

3.8.2 Leader board Comparison: The system checks if the batsman's new record exceeds the overall leaderboard record for the current season. If the batsman surpasses this benchmark, additional notifications may be sent to highlight this accomplishment.

3.8.3 Season Milestone Notification: Fans are notified when batsman achieves a season milestone. This could include milestones such as hitting the most sixes, fours, accumulating the highest run total, and other noteworthy accomplishments throughout the season.

Overall, the process aims to keep fans engaged and informed about the batsman's performance milestones and record-breaking achievements during the cricket season.

# 3.9 Bowler’s milestone:

Notifications are activated when the current bowler surpasses the tournament's overall leader board record. The system also verifies if the bowler matches the current season's statistics. Fans receive notifications when the bowler achieves a seasonal milestone, such as taking the most wickets, bowling the most dot balls, and other notable achievements.

3.9.1 Leaderboard Record Notification: Notifications are sent out when the current bowler surpasses the existing record holder in the tournament's overall leaderboard. This ensures fans are informed when a bowler achieves a significant milestone in terms of overall performance.

3.9.2 Season Statistics Verification: The system checks whether the current bowler matches the statistics set for the ongoing season. If the bowler reaches or surpasses these benchmarks, notifications are triggered to highlight their performance relative to the current season's standards.

3.9.3 Seasonal Milestone Notification: Fans receive notifications when the bowler reaches specific seasonal milestones. These milestones could include achieving the highest number of wickets, bowling the most dot balls (deliveries with no runs scored), and other noteworthy accomplishments during the current season.

In summary, the notification system aims to keep fans updated on the bowler's progress in both overall tournament records and season-specific achievements, enhancing engagement and recognition of outstanding performances on the cricket field.

# 3.10 Database Structure:

The MySQL database structure employed for this project was meticulously designed to capture a comprehensive range of match-level and player-level attributes. The database structure for the automated notification project revolves around several key tables, each capturing distinct facets of player performance across different seasons and disciplines. The primary tables include `curr\_season\_bat`, `curr\_season\_bowl`, `overall\_season\_bat`, and `overall\_season\_bowl`, all interconnected through a common identifier, `PLAYER\_UNIQUE\_ID`. These tables collectively house detailed statistics such as runs scored, wickets taken, balls faced, and overs bowled, segmented by both current and cumulative seasons.

## 3.10.1 Database Tables:

* curr\_season\_bat**:**
  + PlayerName: Name of the player
  + PLAYER\_UNIQUE\_ID: Unique identifier for the player (primary key)
  + Innings: Number of innings played in the current season
  + RunsScored: Total runs scored in the current season
  + BallsFaced: Total balls faced in the current season
  + Fours: Total fours hit in the current season
  + Sixes: Total sixes hit in the current season
  + Most\_50s: Most number of half-centuries scored in the current season
  + Most\_100s: Most number of centuries scored in the current season
* curr\_season\_bowl:
  + PlayerName: Name of the player
  + PLAYER\_UNIQUE\_ID: Unique identifier for the player (primary key)
  + Innings: Number of innings bowled in the current season
  + Overs: Total overs bowled in the current season
  + Wickets: Total wickets taken in the current season
  + DotBalls: Total dot balls bowled in the current season
  + Balls: Total balls bowled in the current season
  + Most\_4\_Wk: Most number of 4-wicket hauls in an innings in the current season
  + Most\_5\_Wk: Most number of 5-wicket hauls in an innings in the current season
* overall\_season\_bat:
  + PlayerName: Name of the player
  + PLAYER\_UNIQUE\_ID: Unique identifier for the player (primary key)
  + Innings: Total number of innings played across all seasons
  + TotalRuns: Total runs scored across all seasons
  + Balls: Total balls faced across all seasons
  + Fours: Total fours hit across all seasons
  + Sixes: Total sixes hit across all seasons
  + HS: Highest score across all seasons
  + Most\_50s: Most number of half-centuries scored across all seasons
  + Most\_100s: Most number of centuries scored across all seasons
* overall\_season\_bowl:
  + PlayerName: Name of the player
  + PLAYER\_UNIQUE\_ID: Unique identifier for the player (primary key)
  + Innings: Total number of innings bowled across all seasons
  + Balls: Total balls bowled across all seasons
  + Wickets: Total wickets taken across all seasons
  + Most\_4\_Wk: Most number of 4-wicket hauls in an innings across all seasons
  + Most\_5\_Wk: Most number of 5-wicket hauls in an innings across all seasons

3.11 Data Source:The dataset used in this project comprises live ball-by-ball data sourced directly from the database.

# 3.12 Data Collection Process:

* The collection of this data was facilitated using custom-designed stored procedures. These procedures were tailored to extract granular information at the ball level for every match.

# 3.13 Data Components:

3.13.1 Historic Data:Historical data from previous T20 World Cups is included in the dataset, providing a comprehensive view of past tournaments.

3.13.2 Player wise Stats:The dataset encompasses detailed player statistics, offering insights into individual player performances across various metrics.

This structured approach ensures that the project leverages a robust database architecture to handle live match updates and historical data seamlessly. By focusing on detailed player performance metrics and leveraging custom extraction procedures, the system supports real-time analysis and reporting, catering to the needs of stakeholders and enthusiasts alike.

|  |  |
| --- | --- |
| **CURRENT MATCH:** | **EVENTS:** |
| Batsman | * On the brink of 50s and 100s. * More than 7 sixes and 10 fours. * 50 and 100 milestone. * Boundary or a six after consecutive dot balls. |
| Bowler | * “3 fer” and “5 fer” milestone. * More than 15 dot balls in a match. * Wicket after consecutive dot balls. |

|  |  |
| --- | --- |
| **SEASON STATS:** | **EVENTS:** |
| Batman | * On the brink of 200, 300, 400, etc runs in a season. * Notification with player’s nth 50 or 100 in that season. * A notification for highest number of sixes and fours on the leader board of that season |
| Bowler | * A notification is sent when a bowler more wickets than the current season leader board. * A notification is sent when a bowler bowls the highest number of dot balls. hence placing at the top of the leader board of that season. |

|  |  |
| --- | --- |
| **EVENTS:** |  |
| 1st innings | * A notification of the score potential is sent after the 8th over. * Momentum shift alert. * A notification containing a summary of the match after the 8th and 15th over is sent. |
| 2nd innings | * A notification regarding the win/loss probability is sent. * Momentum shift alert. * A notification containing a summary of the match after the 8th and 15th over is sent. |

# 4. LIVE IMPLEMENTATION AND INTEGRATION:

* In our quest for real-time insights and the practical application of predictive models, our project smoothly transitioned into live implementation. This crucial phase focused on integrating our predictive models with live match data sourced from a read replica of the live database.
* We introduced live notifications during the latter stages of the IPL tournament. Additionally, we implemented automated notifications for all World Cup matches, including those played simultaneously.
* Score potential and win/lose probability notification has been implemented and sent to the TNCA App for every match played in this world cup.
* Sample output of all the notification is attached in the below PowerPoint for further reference.



# 5. CHALLENGES ENCOUNTERED:

* Gathering data to compile tables for comprehensive statistical analysis.
* Standardizing the data posed difficulties.
* Ensuring data integrity after each game.
* Updating and verifying the Excel sheet after every game to ensure accurate data updates.
* Dealing with multiple records and notifications for a single delivery.

# 6. CONCLUSION:

* Phrases and functions were adjusted as per requirements.
* Changes underwent rigorous testing during live matches.
* Results of testing verified the effectiveness of modifications.
* Overall, the updated approach proved reliable and aligned with intended goals.