Zelus Assessment

This assessment involves engineering a data ingest process for ball-by-ball data from professional cricket matches. The data include the overall match results and outcomes for every delivery for both teams. The questions below are an opportunity to showcase your data engineering skills as they involve building a data ingest pipeline and running exploratory analysis on that data set.

There are many implementation options and the assessment provides an opportunity for you to determine what assumptions are appropriate and operate within those. Since the purpose of this assessment is to showcase your technical and problem-solving skills, please include clear, efficient, and well-organized code along with explanations on the justification for your approach. Please also include instructions on how to run your code, and structure it in a way that makes it as reproducible as possible.

This assessment is expected to take approximately 3-6 hours, though you do not need to complete it in one sitting. We would like for you to return your work to us **within 7 days** of receiving the assessment (i.e. if you receive the assessment on Monday, you have until the end of day the following Sunday to return it). If an alternative schedule has been arranged with you personally, please follow the agreed-upon schedule.

Data Set Description

In this assessment, you will be working with One Day International (ODI) **match results** and ball-by-ball innings data. These data were sourced from **cricsheet.org** and include ball-by-ball summaries of ODIs from 2006-present for men and 2009-present, for women.

Both data sets are in JSON format, which have been compiled from the source YAML files on cricsheet. A full description of the source data structure and definition of variables is available **here**.

The basic rules of ODI cricket can be found **here**. We list the key rules that will be the most useful context for the assessment below. For an ODI,

1. Each team plays one innings (yes, ‘innings’ is singular) consisting of 50 overs, with 6 deliveries per over. The team who bats first is determined by a coin toss.

2. A ‘win’ is recorded when one side scores more runs than the opposing side and all the innings of the team that has fewer runs have been completed. The side scoring more runs has ‘won’ the game, and the side scoring fewer has ‘lost’. If the match ends without all the innings being completed, the result may be a tie or no result.

3. There is theoretically no limit to the number of runs that can be earned off a single delivery as the run tally can increase as the striker and non-striker run to opposite ends of the pitch. However, a hit that bounces and reaches the boundary is an automatic 4 runs and a hit that hits or passes the

boundary without a bounce is an automatic 6 runs.

4. A ‘wicket’ is cricket’s equivalent to an out in baseball. A batter continues batting until they are out. The main ways to take a wicket (or ‘dismiss’ an opposing player) are for the bowler to dismiss a batter with the delivery (e.g. bowled out, leg before wicket, etc.), to catch a batted ball on the fly, or to throw out either the batter or the non-striker as they attempt to run between the wickets.

5. Each team starts with 10 wickets. Once all wickets are lost the innings ends, whether the 50 overs have been completed or not.

If you still feel like you need more grounding in the game of cricket, you can take 17min of the assessment time to watch Netflix’s Explained: Cricket which can be watched for free on **Youtube**.

Assessment Questions

Please complete the coding portion of the assessment using Python and present your final results with all required code artifacts to run the solution in a zip file. Questions 0 and 3 include written responses; please include a PDF with your answers. The final submission will include both the zip file with your code and PDF with written answers.

Question 0. We don’t expect you to have any cricket knowledge and that is not a requirement to ace this assessment. But we understand that familiarity with cricket may vary from one candidate to the next so we would like to know how you would rate your knowledge of cricket from 1 to 5, where 1 is basically no knowledge (like you had never seen or read anything about the sport until the days before this assessment) and 5 is highly knowledgeable (you watch matches regularly and have a jersey for the Rajasthan Royals in your closet, for example).

Question 1. Develop a batch data ingest process to load the ODI match results and ball-by-ball innings data to a database of your choosing (as a default, you can use SQLite). The solution should include a step that downloads files directly from cricsheet.org and performs any required preprocessing. The database schema should store match results and ball-by-ball innings data along with the universe of players that appear across all matches. The process should be runnable from the command line, inclusive of creating any dependencies (e.g. local file directories, the database, etc.). Please include a README.md file with instructions on how to build and run the ingest process to reproduce your results.

Question 2. Using the database populated in Question 1, develop queries to answer the questions below. Please include the queries as .sql files with your code submission.

a. The win records (percentage win and total wins) for each team by year and gender, excluding ties, matches with no result, and matches decided by the DLS method in the event that, for whatever reason, the planned innings can’t be completed.
b. Which male and female teams had the highest win percentages in 2019?
c. Which players had the highest strike rate as batsmen in 2019? (Note to receive full credit, you need to account for handling extras properly.)

Question 3. Please provide a brief written answer to the following question. The coding assessment focused on a batch backfilling use case. If the use case was extended to required incrementally loading new match data on a go-forward basis, how would your solution change?