Zelus Assessment

This assessment involves analysis of ball-by-ball data from professional cricket matches. The data include the overall match results and outcomes for every delivery for both teams.

The problems below showcase your ability to gather inferences from real-world data using methods in statistics and machine learning, as well as your ability to productionalize and deploy models for consumption. After the first question, there are no right or wrong answers; in fact, we have left questions quite open-ended as an opportunity for you to determine what assumptions are appropriate and operate within those. In addition to a sensible problem-solving strategy, we'll be looking for correct implementation and validation of the appropriate techniques you've chosen, as well as an understanding of their limitations.

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 problem-solving approach, its limitations, and the conclusions you're able to draw at each step. 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 on the **seventh day** after receiving the assessment (i.e if you receive the assessment on Monday, you have until the end of day Sunday to return it). If an alternative schedule has been arranged with you personally, please follow the agreed-upon schedule.

Part 1: Data Analysis

Please complete **Part 1** of the assessment using either R or Python and present your work in a **Jupyter notebook** or **R notebook**, including both the .ipynb or .Rmd notebook file as well as a PDF or HTML version of the notebook showing all cells evaluated. You may use any online resources or additional software (with citations) in your work. You may complete **Part 2** of the assessment in either R or Python as well, plus the shell of your choosing, and Docker if you prefer.

To get started, download the **One Day International** match results and ball-by-ball innings data. These data were sourced from **cricsheet.org** and includes 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.

Question 0. We don't expect you to have any cricket knowledge and that isn't 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. Determine 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. Consider only data from 2019. Which male and female teams had the highest win percentages? Which had the highest total wins? Were these teams the same as those with the highest win percentages? Comment on why the leaders of these two stats might differ.

Question 2. Setting aside individual batter production, cricket teams have two main 'resources' for producing runs: remaining overs and wickets. The role resources have on run production is central to the statistical method known as 'DLS', which is used to award a winner in the case of incomplete/disrupted matches. Use the ball-by-ball summaries under the innings descriptions of each men's match to make a dataset with the run and wicket outcomes for each delivery in a match, excluding matches with no result.

Develop a model to predict an average team's expected runs per over. Please state or include the assumptions/validation used to justify your model choice. A visualization prior to modelling could be helpful to justify your modelling decisions. Save your intermediate data with team, inning order, remaining overs, and remaining wickets to a JSON/CSV file for Q4. Summarize your conclusions.

Question 3. More generally and unrelated to cricket or the previous questions, model deployment in a production environment is an important aspect of an engineer's toolkit. Describe a scalable architecture (a diagram may be helpful) that would be appropriate for deploying a model that predicts frame-level play values into a cloud environment with the following assumptions:

- Spatial temporal high frame-rate data (~1 GB per game)
- Play-values are predicted at each frame of a game
- Delivery of game predictions are expected to be delivered overnight
- 500 games per season with 50 games a day
- 5 seasons of existing data
- Model training resources:
  - 8 hour runtime with multiple cores (8) and large memory usage
- Model prediction resources:
  - 60 min runtime per game with a single CPU and 4 GB of memory usage

List out the services, tooling, and reasoning for the choices of architecture. For example, a LAMP stack could be appropriate for an internal home network webpage on a Raspberry Pi.

Part 2: Deployment

This part of the assessment showcases your ability to deploy a model in a local environment. Please complete **Part 2** of the assessment by packaging your working directory in a zip file with the intermediate data file from Q2, model files, and any necessary scripts.

Question 4. Save your model from Q2 into a file and create a packaged solution for being able to build, deploy and run your model locally. We are expecting a solution where local runs can be initiated from the command line, not an API-style deployment. As a way to test your package, create a shell script that takes data saved from Q2, filters for the first 5 Ireland overs, sends them to your model, and displays the model results to `stdout`.