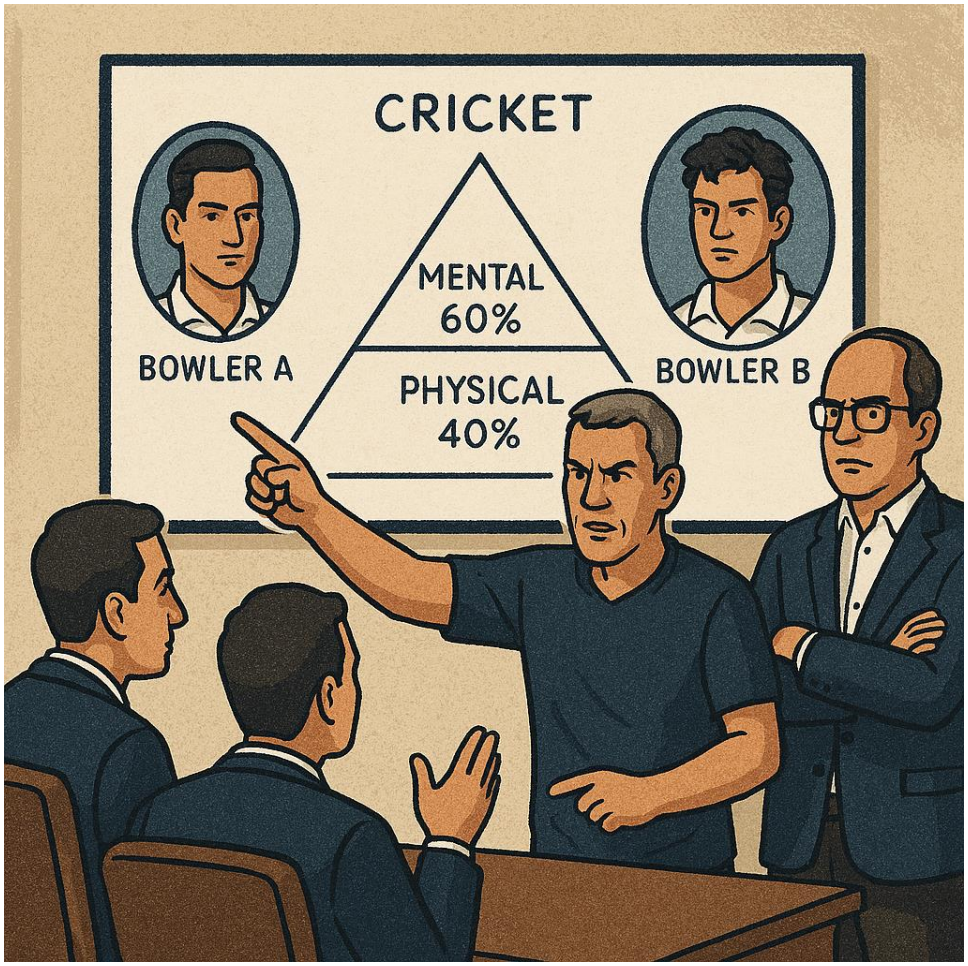
Assignment: The "Ghost in the Machine" (IPL Auction Analytics)

Role: Data Scientist (DCA)

Time Limit: 2 Weeks

Deliverable: Jupyter Notebook + Executive Summary (PDF)



1. The Situation (The War Room)

It is 48 hours before the Mega Auction. You are in the strategy room with the Head Coach and the Owners. You have one slot left for a "Death Over Specialist" (Overs 16-20).

You present two options with identical base prices:

- **Bowler A ("The Machine"):** High accuracy, low economy, perfect yorkers. The "Safe" bet.

- **Bowler B ("The Gambler"):** Aggressive, expensive, sometimes erratic. The "Wildcard."

The Scouts point to the data:

"Bowler A has a better economy rate (7.5 vs 8.9). It's a no-brainer."

The Coach shakes his head. He draws a pyramid on the whiteboard:

"You analysts only see the physical. Cricket is 40% Physical (Pace, Swing, Line) and 60% Mental (Resilience, Aggression, Reading the Batsman).

Bowler A is physically perfect but mentally fragile. If he gets hit, he crumbles.

Bowler B? He has the 'Killer Instinct'. He smells fear. If he bowls a dot ball and puts the batter under pressure, I know the next ball is a wicket.

But you can't measure 'Mental Strength' in a spreadsheet, so I can't prove it to the owners."

2. The Challenge

Your mission is to prove the Coach wrong. **Data *can* measure mental strength.** You must design an experimental framework to quantify "Killer Instinct"—specifically, how a bowler capitalizes on pressure.

3. The Dataset (IPL_Bowler_Detailed_Data.csv)

You are provided with a ball-by-ball dataset for these two bowlers from various T20 leagues over the last 2 years.

Data Dictionary:

Match_ID | Unique identifier for the match.

Match_Date | Date of the match.

Pitch_Type | The condition of the track:

- Batting (Flat, High Scoring)
- Neutral
- Bowling (Green/Dusty, Low Scoring)

Phase | Powerplay (Overs 1-6) or Death (Overs 16-20).

Over | The over number (1-20).

Ball | The ball number (1-6).

Bowler | The Bowler's Name (A or B).

Batter_Avg | The career Batting Average of the batsman facing the ball.

Batter_SR | The career Strike Rate of the batsman facing the ball.

Runs_Conceded | Runs scored off this specific delivery (0, 1, 2, 3, 4, 6).

Is_Wicket | 1 if the ball resulted in a wicket, 0 otherwise.

4. The Experiment (Deliverables)

Phase 1: The "Mental Proxy" (Feature Engineering)

- Translate the Coach's intuition into code.
- **Hypothesis:** "Pressure" is applied when a bowler bowls a **Dot Ball (0 runs)** in the **Death Overs**.
- **Target Metric:** We want to measure the probability of taking a wicket on the *very next ball* after applying pressure.
- *Warning:* Be careful with your logic. A dot ball on the last ball of an over does *not* apply pressure to the first ball of the next over.

Phase 2: The Model (Bayesian Inference)

- Build a probabilistic model (using **PyMC**) to predict Is_Wicket.
- **The Challenge:** The data is noisy. A wicket might happen because it's a "Bowling Pitch" or the batsman is weak (Low Batter_Avg), not because of the bowler's skill.
- **Requirement:** Construct a model that estimates the **"Pressure Effect"** for Bowler A vs. Bowler B, while *controlling* for:
    - Pitch Type
    - Batsman Quality (Batter_Avg)
- *Hint:* Use a Generalized Linear Model (GLM) or Hierarchical Model.

Phase 3: The Verdict (Kaizen)

- Plot the **Posterior Distribution** of the "Pressure Coefficient" for both bowlers.
- **The Decision:** You have the checkbook. Who do you buy?
    - Justify your answer using the **94% High Density Interval (HDI)** from your model.
    - Address the Coach's skepticism: "Here is the graph that proves 'Killer Instinct' exists."

3. A Note on Using AI (The "DCA Way")

At DCA, we embrace tools like ChatGPT, Claude, and Gemini. We expect you to use them to write your PyMC boilerplate code.

However, be warned: LLMs are junior analysts. They make mistakes with:

1. **Sequential Logic:** They often mess up pandas shifts (e.g., using the last ball of Over 1 to calculate pressure for the first ball of Over 2).
2. Context: They often hallucinate "good" priors without understanding the cricket domain.

You are the Lead. The AI is your intern. If the intern bugs the code, the Lead takes the blame.