**The Wisdom of Crowds Summary.**

To develop this project, I opted for "clean architecture," which allows me to modularize the main program. This program could have been an API, a service, or a Cron task. In this way, the solution consists of three parts: the program itself, its services and interfaces, and the infrastructure external to the program. Choosing this working method allows for faster and more fluid scalability, understanding, migration, and implementation.

**Analysis and findings.**

1. Odds Aggregation and Implied Probability:

- Collects odds for each outcome (home win, away win, draw if available) from multiple bookmakers for every event.

- Calculates the average odds for each outcome.

- Converts average odds to implied probabilities using the formula:

`Implied Probability = 1 / Odds`.

2. Historical Outcome Analysis:

- Imports historical match results (from Excel or other sources).

- Calculates actual outcome frequencies (e.g., percentage of home wins, away wins, draws) for each team and overall.

3. Team Statistics:

- Computes, for each team:

- Total matches played

- Wins, losses, home wins, away wins

- Win rate (percentage of matches won)

- These statistics are used to compare bookmaker probabilities to real-world performance.

4. Profitability Simulation:

- Simulates betting on the favorite (lowest average odds) for each event.

- Estimates profitability by comparing simulated bets to actual historical outcomes.

- Considers bookmaker margin (overround) by summing implied probabilities.

5. Name Normalization:

- Standardizes team names using a mapping file (`AFL\_teams.txt`) to ensure consistency across data sources.

---

How to Get the Analysis.

1. Configure Data Sources

- Set up API keys and endpoints in `appsettings.json` for odds data.

- Place your historical match data in the configured Excel file.

- Ensure `AFL\_teams.txt` contains correct team name mappings.

2. Run the Application:

- Build and run the solution. The background worker (`Worker.cs`) will:

- Fetch current odds and historical data.

- Normalize team names.

- Calculate average odds, implied probabilities, and team statistics.

- Simulate betting strategies and output profitability.

- Optionally, send notifications via email.

3. View Results:

- Analysis results (such as average odds, implied probabilities, win rates, and profitability) are output to the console or can be extended to be saved or sent via email.

4. Extend or Customize:

- To add new analysis or change existing logic, modify the relevant services:

- `EventAnalysis.cs` for odds and probability calculations.

- `HistoricalTeamStatistics.cs` for team performance metrics.

- `Worker.cs` for orchestration and output.

**Notification System.**

* **Email DTO (WisdomOfCrowndBets.Core.DTO.Email):**  
  Defines the structure for email data, including sender, recipient, subject, body, SMTP server, and port.
* **ISendEmail Interface (WisdomOfCrowndBets.Core.Interfaces.ISendEmail):**  
  Declares the contract for sending emails asynchronously.
* **SendEmail Class (WisdomOfCrowndBets.Infrastructure.Repositories.SendEmail):**  
  Implements the ISendEmail interface.
  + Uses **System.Net.Mail** to construct and send emails.
  + Accepts dynamic message content and email configuration via the Email DTO.
  + Handles SMTP authentication, SSL, and error reporting.

**How It Works.**

* **Configuration:**  
  Email settings (SMTP server, port, sender, recipient, etc.) are provided via the Email DTO, which can be populated from appsettings.json or user input.
* **Sending:**  
  The SentBetEmailNotification method creates a **MailMessage**, sets its properties, and sends it using an **SmtpClient** configured with the provided credentials and server details.
* **Error Handling:**  
  SMTP and general exceptions are caught and logged to the console for troubleshooting.

**Methodology.**

The wisdom of the crowd is the idea that a large group of people's aggregated opinions, estimates, or predictions can be surprisingly accurate, often more so than those of individual experts.

Analyze whether aggregated odds reflect the "wisdom of the crowds" by checking if they predict outcomes better than individual bookmaker odds. For example, do average odds undervalue underdogs or overvalue favorites?

The Underdogs are Overvalued: Bettors tend to over bet on longshots, possibly due to the allure of a large potential payout for a small stake, even if the probability of winning is very low. Bookmakers may also shorten longshot odds due to the higher percentage liability they face if a longshot wins.

The Favorites: Bettors may be less inclined to bet on favorites because the potential return is smaller. Bookmakers might offer slightly better odds-on favorites to attract more volume at the top end of the market.