Client Delivery Document: Betfair BSP Scraper

Project: Python Script for Adding Betfair Starting Price (BSP) to Historical Bet Data

Date: June 20, 2024

Version: 1.0

---------------------------------------------------------------------------------------------------------------------------------------

**1. Project Goal**

This Python script is designed to read your historical betting data from a CSV file, automatically visit the Betfair Australia Results Hub website to find the official Betfair Starting Price (BSP) for each bet, and then add this BSP information to your original data, saving it in a new CSV file. This provides you with enriched data for more comprehensive analysis.

**2. What You Need (Prerequisites)**

Before running the script, please ensure you have the following:

1. Python 3: The script is written in Python 3 (preferably Python 3.7 or newer). If you don't have Python, you can download it from python.org.

2. Google Chrome Browser: The script uses Selenium to control a Chrome browser to access the Betfair website. Please ensure Chrome is installed.

3. Internet Connection: An active internet connection is required to access the Betfair website.

**3. Setup Instructions (Step-by-Step)**

1. Create a Project Folder:

\* Create a new folder on your computer where you will keep the script and related files (e.g., Betfair\_BSP\_Project).

2. Save the Script:

\* Save the Python script file provided (e.g., bsp\_finder.py) into this project folder.

3. Create `requirements.txt` file:

\* In the same project folder, create a new text file named `requirements.txt`.

\* Open this file and add the following lines (these are the Python libraries the script needs):

pandas

selenium

webdriver-manager

\* Save and close `requirements.txt`.

4. Install Required Python Libraries:

\* Open a command prompt or terminal.

\* Navigate to your project folder using the `cd` command (e.g., `cd path\to\your\Betfair\_BSP\_Project`).

\* Run the following command to install the libraries:

`pip install -r requirements.txt`

\* This will download and install `pandas` (for data handling), `selenium` (for web browser automation), and `webdriver-manager` (to automatically manage the Chrome browser driver).

5. Prepare Your Input CSV File:

\* Place your historical betting data CSV file into the same project folder.

\* Important: The script expects this file to be named `bet\_sample.csv`. If your file has a different name, please rename it or update the filename in the script (in the `get\_input\_csv` function).

---

**4. How to Run the Script**

1. Ensure all setup steps above are complete.

2. Open a command prompt or terminal.

3. Navigate to your project folder (e.g., `cd path\to\your\Betfair\_BSP\_Project`).

4. Run the script using the command:

`python bsp\_finder.py`

(Replace `bsp\_finder.py` with the actual name of your script file if different).

5. The script will start, and you will see log messages in the console. A new Chrome browser window will open and navigate automatically. Do not close this browser window or interact with it manually unless instructed. The script will close it when finished.

6. The first time you run the script, `webdriver-manager` might take a moment to download the correct driver for your Chrome browser. This is normal.

**5. Input File (`bet\_sample.csv`)**

\* Name: The script is hardcoded to look for `bet\_sample.csv`.

\* Format: Standard Comma Separated Values (CSV) file.

\* Required Columns: The CSV file must contain the following columns (case-insensitive headers, but these specific names are good practice):

\* `Time`: The date and time of the bet (e.g., `06/13/2025 14:30` or `13/06/2025 14:30`). The script can handle common date formats.

\* `Venue`: The name of the racetrack or venue (e.g., `MILDURA`, `ALBION PARK`). This must match the venue names on the Betfair results website.

\* `Code`: The type of racing (e.g., `HARNESS`, `GREYHOUNDS`, `THOROUGHBRED`, or abbreviations like `H`, `G`, `R`). The script maps these to website IDs.

\* `RaceNo`: The race number (e.g., `1`, `5`).

\* `RunnerNo`: The runner's number/bib in the race (e.g., `1`, `10`).

\* `RunnerName`: The name of the runner (e.g., `My Champion`).

\* Other Columns: Any other columns in your CSV will be preserved and included in the output file.

\* Data Accuracy: The more accurate your input data (especially Venue, RaceNo, RunnerNo, RunnerName, and Date), the higher the chance of finding a correct BSP match.

**6. How the Script Works (Simplified Technical Flow)**

1. Reading Data: The script first reads all your bet records from the `bet\_sample.csv` file into memory using the `pandas` library.

2. Grouping Bets: To make the web scraping process efficient, it groups your bets by Date, then by Race Type (Code), and then by Venue. This minimizes how often it needs to change settings on the Betfair website.

3. Web Interaction (Selenium): The script opens a Google Chrome browser window and controls it automatically.

4. Navigating Betfair Results Hub:

\* It goes to the Betfair Australia Horse Racing Results page.

\* For each group of bets:

\* Selects Date: It interacts with the on-page calendar to select the correct date.

\* Date Data Check 1: After selecting a date, it waits for the main "filter panel" (where race types and venues are listed) to appear.

\* Date Data Check 2: It then waits up to 2 minutes for the list of \*venue filters\* to actually populate for that date. If this fails, the script assumes no valid data is available for that date, marks all bets for that date with an error ("Date Data Not Loaded"), and moves to the next date.

\* Selects Race Type: It clicks the correct race type button (e.g., Harness, Greyhound, Thoroughbred) based on the `Code` from your CSV.

\* Selects Venue: It finds and clicks the specific venue filter (e.g., 'MILDURA') from the list.

\* Venue Failure Check: If, after successfully loading a date's initial data, the script fails to select/load two different venues for that \*same date\*, it will mark that date as problematic and skip any further attempts for that date. This prevents getting stuck on dates with consistently failing venue data.

5. Extracting Race Details:

\* Once on the correct date, race type, and venue page, the script iterates through the races you have bets for.

\* For each race:

\* It clicks the correct race tab (e.g., "R1", "R5").

\* It then looks for the specific runner (by `RunnerNo` and `RunnerName`) from your CSV within that race's displayed results.

6. Getting BSP: For each matched runner, it extracts the "Win BSP" and "Place BSP" values shown on the website.

7. Collecting Data: All your original bet data, along with the newly found BSP Win and BSP Place prices, are collected.

8. Handling Missing Data/Errors:

\* If a BSP cannot be found for a runner (e.g., runner not listed, data mismatch, or no BSP available), it will be marked as 'Not Found' or with a specific error message (e.g., 'Race Error', 'Venue Load Error').

\* The script logs all its actions and any errors encountered.

9. Removing Duplicates: Before saving, the script checks for and removes any rows that are exact duplicates across all columns.

10. Saving Output: Finally, the script saves all the processed data (original + BSP columns) into a new CSV file.

**7. Output Files**

1. `final\_results.csv`:

\* This is the main output file.

\* It contains all the columns from your original `bet\_sample.csv` file.

\* It will have two new columns added:

\* `BSP` (Header Level 1) -> `Price Win` (Header Level 2): The Betfair Starting Price for Winning.

\* `BSP` (Header Level 1) -> `Price Place` (Header Level 2): The Betfair Starting Price for Placing (if available).

\* Values like "N/A", "Not Found", or other error messages will appear in the BSP columns if the price could not be retrieved.

2. `bsp\_scraping\_detailed.log`:

\* This is a detailed log file.

\* It records every major step the script takes, any data it processes, XPaths it uses, elements it interacts with, and any warnings or errors encountered.

\* This file is very useful for troubleshooting if the script doesn't behave as expected or if you want to see exactly what happened for a specific bet.

**8. Understanding the Logs**

\* Console Output (on your screen):

\* You will see `INFO` level messages. These give you a high-level overview of the script's progress, such as:

\* Starting and finishing the script.

\* Loading the input CSV.

\* Navigating to the website.

\* When it's changing Date, Race Type, or Venue.

\* Major errors encountered (e.g., if a date's data cannot be loaded).

\* Successfully saving the output file.

\* `bsp\_scraping\_detailed.log` File:

\* Contains `DEBUG` level messages, which are much more granular.

\* Includes all `INFO` messages plus:

\* Specific XPaths or CSS selectors used to find elements.

\* Confirmation when elements are found or clicked.

\* Values being compared or extracted.

\* More detailed error tracebacks.

\* If you encounter issues, this log file is the first place to look for clues.

**9. Important Notes & Basic Troubleshooting**

\* Internet Connection: A stable internet connection is vital. Interruptions can cause the script to fail.

\* Website Changes: Betfair may update their website structure. If this happens, the script might stop working correctly as it relies on specific HTML element layouts. This is a common challenge with web scraping.

\* Input Data Quality: The script's success heavily depends on the accuracy of your `bet\_sample.csv` data (especially date, venue name, race number, and runner details).

\* Run Time: For large CSV files, the script can take a significant amount of time to run as it's visiting many web pages.

\* First Run: The very first time you run the script, `webdriver-manager` will download the ChromeDriver. This might add a slight delay to the startup. Subsequent runs will be faster.

\* "Date Data Not Loaded" Errors: If you see this for a date, it means the Betfair website did not provide the list of venues for that date within the 2-minute timeout, or the initial filter panel didn't appear. The script will skip all bets for such dates.

\* "Venue Load Error" / "Max Venue Failures": If a date initially seems okay, but then two different venues on that date fail to load their race details, the script will mark that entire date as problematic and skip further processing for it.

\* Firewall/Antivirus: In rare cases, security software might interfere with Selenium's operation or network access.

**10. Deliverables Summary**

You have received:

1. The Python script file (e.g., `bsp\_finder.py`).

2. This documentation file.

3. A `requirements.txt` file (to be created by you as per setup).

Please let me know if you have any questions or require further assistance.