

Here's a comprehensive **Product Requirements Document (PRD)** for the Python program you've developed. This PRD outlines the purpose, functionality, and technical specifications of the system, making it easy to share with collaborators or use for future enhancements.



Product Requirements Document (PRD)

Title:

Binary Matrix Analysis Tool

Purpose:

To generate a large binary matrix with specified column-wise probabilities, analyze the frequency of row patterns (tuples), and compute key metrics such as support, confidence, and lift for frequent patterns. This tool is useful for data mining, pattern recognition, and association rule analysis.

Features & Functional Requirements

1. Matrix Generation

- **Size:** 5000 rows × 6 columns
- **Values:** Randomly filled with 0s and 1s
- **Column-wise Probability Distribution:**
 - Column 1: 70% chance of 1
 - Column 2: 60% chance of 1
 - Column 3: 50% chance of 1
 - Columns 4–6: 40% chance of 1

2. Initial Output

- Print the **first 10** and **last 10** rows of the matrix for verification.

3. Tuple Frequency Analysis

- Convert each row into a tuple.
- Filter out tuples with fewer than **two 1s**.
- Count the frequency of each unique tuple.
- Identify and print the **top 10 most frequent tuples**.

4. Metric Calculations

For each of the top 10 tuples:

- **Support:**

$$[\text{Support}] = \frac{\text{Tuple Count}}{5000}$$
- **Confidence:**

$$[\text{Confidence}] = \frac{\text{Support}}{\text{Empirical Probability of 1 in Column 1}}$$
- **Lift:**

$$[\text{Lift}] = \frac{\text{Confidence}}{\text{Empirical Probability of Subtuple (Columns 2-6)}}$$

5. Conditional Tuple Analysis

- Filter top 10 tuples where **Column 1 and Column 2 are both 1**.
- For these tuples:
 - **Support:** As above
 - **Confidence:**

$$[\text{Confidence}] = \frac{\text{Support}}{\text{Empirical Probability of 1 in Columns 1 AND 2}}$$
 - **Lift:**

$$[\text{Lift}] = \frac{\text{Confidence}}{\text{Empirical Probability of Subtuple (Columns 3-6)}}$$

Technical Specifications

Language & Libraries

- **Python 3.x**
- **NumPy**: For matrix generation and manipulation
- **Collections.Counter**: For frequency analysis

Output Format

- Console printout of:
 - First and last 10 rows
 - Top 10 tuples with count, support, confidence, lift
 - Filtered tuples with 1s in both first and second columns, with metrics
-

Assumptions

- Random seed is set for reproducibility.
 - Matrix size and probabilities are fixed but can be parameterized for flexibility.
 - Only tuples with at least two 1s are considered for frequency analysis.
-

Future Enhancements

- Export results to CSV or JSON
 - Add visualization (bar charts for frequency, lift distribution)
 - Support for larger matrices or configurable dimensions
 - Integration with association rule mining libraries (e.g., `mlxtend`)
-

Let me know if you'd like this PRD formatted for a specific platform (e.g., Confluence, Notion, Word) or expanded into a technical spec or user manual.