Product Viewing Analysis Tool

Git commit with doc: fcd4d89
HTML version

The `frequent` script (see technical details) analyzes customer product viewing patterns to identify the most engaging products within a 5-minute timeframe.

1. Purpose

This tool helps identify which products receive the most attention from customers by analyzing their viewing behavior. It's particularly useful for:

- Understanding which products attract the most customer interest
- Identifying potential candidates for promotional campaigns
- Analyzing customer engagement patterns

2. How it Works

2.1. Input

The script processes a JSON Lines file containing product viewing events, where each line represents a customer interaction with a product page.

2.2. Output

The script generates a JSON file that ranks products based on two key metrics:

- `occurrences`: How many times customers viewed the product
- `total_seconds`: Total time spent viewing the product (in seconds)

Products are sorted by:

1. Number of views (highest to lowest)
2. Total viewing time (highest to lowest) when number of views is equal

2.3. Time Window

The analysis focuses on the first 5 minutes of activity, starting from the timestamp of the first event. This helps identify products that generate immediate interest from customers.

3. Latest Analysis Results


```
{
  "SP Dunk Low Retro": {
    "occurrences": 12,
    "total_seconds": 1850
  },
  "SP Air Force 1 Shadow": {
    "occurrences": 10,
    "total_seconds": 1516
  },
  "Total Orange": {
    "occurrences": 7,
    "total_seconds": 958
  },
  "SP Air Max Plus 3": {
    "occurrences": 4,
    "total_seconds": 501
  },
  "SP Flex Runner 2": {
    "occurrences": 3,
    "total_seconds": 547
  }
}
```


This output shows the most engaging products in the analyzed timeframe, with "SP Dunk Low Retro" being the most viewed product with 12 views and 1,850 seconds of total viewing time.

4. Technical Details

The frequent (← Bash wrapper script) is implemented in three different programming languages to demonstrate language-agnostic processing capabilities. All implementations produce identical results:

4.1. Available Implementations

- **JavaScript**: frequent.js - Node.js implementation
- **Python**: frequent.py - Python 3 implementation
- **jq**: frequent.jq - jq implementation

4.2. Usage

The script can be executed using any of these implementations:

```
./frequent js  # Run JavaScript version
./frequent py  # Run Python version
./frequent jq  # Run jq version
```

If no implementation is specified, the script defaults to JavaScript:

```
./frequent
```

All implementations:

- Read JSON Lines from standard input
- Process the first 5 minutes of events
- Sort products by views and viewing time
- Output formatted JSON to standard output

The consistent output across all implementations ensures reliability and provides flexibility in choosing the most suitable technology for your environment.