Problem Statement

A new software called "TaskTrack" has been rolled out in the company for time tracking, task management, and productivity monitoring. However, there are complaints from customers that the application isn't accurately tracking time on certain tasks. The suspicion is that the system doesn't properly recognize when a task starts and stops, leading to inaccurate time tracking. The provided dataset contains task logs with timestamps, user IDs, task IDs, and event types (Task Started, Task Stopped, or Task Completed). The goal is to analyze the log data, identify discrepancies, and propose solutions to improve accuracy.

Below is the subset of the sample dataset:

```
timestamp, userID, taskID, eventType
1625234567, U01, T01, Task Started
1625234575, U01, T01, Task Stopped
1625234600, U01, T02, Task Started
1625234650, U01, T02, Task Completed
1625234700, U01, T03, Task Started
1625234800, U01, T03, Task Completed
1625234850, U01, T04, Task Started
1625234900, U01, T04, Task Completed
1625234950, U01, T05, Task Started
1625235000, U01, T05, Task Completed
1625235100, U02, T06, Task Started
1625235150, U02, T06, Task Stopped
1625235200, U02, T07, Task Started
1625235250, U02, T07, Task Completed
1625235300, U02, T08, Task Started
1625235350, U02, T08, Task Completed
1625235400, U02, T09, Task Started
1625235500, U02, T09, Task Completed
```

Solution Approach

1. Process of Thinking and Problem Solving

The script is designed to handle a CSV file upload via a Flask web application. The CSV file is expected to contain event data with columns: timestamp, userID, taskID, and eventType. The script processes this data, calculates the total time spent by each user on each task, and identifies any discrepancies in the event data.

2. Issues Identified While Analyzing the Dataset

The script identifies several potential issues in the dataset:

- A task is recording a new eventType “Task Completed”
- A task is started again without being stopped or completed by the same user.
- A task is stopped or completed without being started by the same user.
- An unknown eventType is found for a user and task.
- No data is found in one or more columns (UserID, taskID, timestamp, eventType)
- A task is started only once but never stopped or completed by the same user.

3. Proposed Solutions to These Issues

The script handles these issues as follows:

- “Task Completed” is equivalent to “Task Stopped”, and logic is added to the application to calculate the time spent based on “Task Completed”
- If a task is started again without being stopped or completed by the same user, the script ignores the current started entry and considers the first entry. This ensures the time spent is not impacted by intermittent started events when the originally started event is not completed or stopped. Such cases are logged in the discrepancy report.
- If a task is stopped or completed without being started by the same user, the script ignores the current stopped or completed entry. Such cases are logged in the discrepancy report.
- The script logs a discrepancy if an unknown eventType is found for a user and task. It’s important to review such logs and implement the new eventType in logic to improve the accuracy of the report.
- The script logs a discrepancy if no data is found in one or more columns (UserID, taskID, timestamp, eventType).
- If a task is started but stopped or a completed event record is not found in the log file, the script will log a discrepancy. It’s important to check such logs and ensure they have valid stopped or completed events unless the task is still being worked on by the current user.

4. Testing and Debugging

The script uses Python's built-in debugging capabilities. By running the Flask app with debug=True, any errors will be displayed in the console, along with a traceback. This helps to identify and fix any issues in the script.

The script also prints a report after processing the data, which includes the total time spent by each user on each task and any discrepancies found. This report can be used to verify the correctness of the script.

It's essential to monitor the logs and discrepancies and fix them per the emerging to maintain the accuracy of the time reports.

Besides the above, manual tests are performed using 5 different datasets and the application handles them with no issues.

Dataset 1: Normal Case

timestamp,userID,taskID,eventType
1617184800,user1,task1,Task Started
1617185800,user1,task1,Task Stopped
1617186800,user1,task1,Task Started
1617187800,user1,task1,Task Completed
1617188800,user2,task2,Task Started
1617189800,user2,task2,Task Stopped
1617190800,user2,task2,Task Started
1617191800,user2,task2,Task Completed

Dataset 2: Task Started Again Without Being Stopped or Completed

timestamp,userID,taskID,eventType
1617184800,user1,task1,Task Started
1617185800,user1,task1,Task Started
1617186800,user1,task1,Task Stopped
1617187800,user2,task2,Task Started
1617188800,user2,task2,Task Started
1617189800,user2,task2,Task Stopped
1617190800,user3,task3,Task Started
1617191800,user3,task3,Task Started
1617192800,user3,task3,Task Stopped
1617193800,user4,task4,Task Started

Dataset 3: Task Stopped or Completed Without Being Started

timestamp,userID,taskID,eventType
1617184800,user1,task1,Task Stopped
1617185800,user1,task1,Task Started
1617186800,user1,task1,Task Completed
1617187800,user2,task2,Task Stopped
1617188800,user2,task2,Task Started

1617189800,user2,task2,Task Completed
1617190800,user3,task3,Task Stopped
1617191800,user3,task3,Task Started
1617192800,user3,task3,Task Completed
1617193800,user4,task4,Task Stopped

Dataset 4: Unknown eventType
timestamp,userID,taskID,eventType
1617184800,user1,task1,Task Started
1617185800,user1,task1,Task Paused
1617186800,user1,task1,Task Completed
1617187800,user2,task2,Task Started
1617188800,user2,task2,Task Paused
1617189800,user2,task2,Task Completed
1617190800,user3,task3,Task Started
1617191800,user3,task3,Task Paused
1617192800,user3,task3,Task Completed
1617193800,user4,task4,Task Started

Dataset 5: No Data in One or More Columns
timestamp,userID,taskID,eventType
1617184800,,task1,Task Started
1617185800,user1,,Task Stopped
1617186800,user1,task1,
1617187800,,task2,Task Started
1617188800,user2,,Task Stopped
1617189800,user2,task2,
1617190800,,task3,Task Started
1617191800,user3,,Task Stopped
1617192800,user3,task3,
1617193800,,task4,Task Started

Dataset 6: Missing stopped or completed events for user3
timestamp,userID,taskID,eventType
1649610000,user1,task1,Task Started
1649613600,user1,task1,Task Completed
1649617200,user2,task2,Task Started
1649620800,user2,task2,Task Stopped
1649624400,user2,task2,Task Started
1649628000,user2,task2,Task Completed
1649631600,user3,task3,Task Started
1649631600,user3,task4,Task Started

A high-level overview of the application

This Flask application allows users to upload a CSV file containing task logs, and generates a report detailing the total time spent on each task by each user, as well as any discrepancies in the logs.

Data Preparation:

- The application reads the uploaded CSV file using pandas.
- The CSV file is expected to have four columns: timestamp, userID, taskID, and eventType.
- The timestamp is converted from seconds to a datetime object.
- The eventType is converted to a string.
- The data is sorted by timestamp.

Calculate Time Spent:

- The application initializes two dictionaries: start_time to store the start timestamp for each user-task combination, and total_time to store the total time spent on each task by each user.
- The application iterates through the logs:
    - For “Task Started” events, it records the start timestamp.
    - For “Task Stopped” or “Task Completed” events, it calculates the time spent (difference between the current timestamp and the recorded start timestamp), updates the total time, and removes the start timestamp.

Identify Discrepancies:

- The application checks for discrepancies such as overlapping events (multiple “Task Started” events without corresponding “Task Stopped” events), missing “Task Started” events, and unrecognized event types. It also checks if any data is missing in the columns.
- Any discrepancies found are added to a list.

Generate Report:

- The application generates a report containing the total time spent on each task by each user, and any discrepancies found.
- The report is displayed on a webpage using Flask’s render_template function.

Run Application:

- The application is run using Flask’s built-in server, with debug mode enabled. This allows for automatic reloading of the server when the code is updated, and provides detailed error messages.