Data Analyst Technical Challenge

Introduction:

This document summarizes the data analysis performed on production line data as part of the Junior Data Analyst Challenge. The analysis focuses on evaluating production efficiency, downtime, quality issues, and changeover times across multiple production lines. Using Python, pandas, SQLite, and visualization libraries (matplotlib, seaborn), also cleaned the data, performed exploratory analyses, and derived actionable insights to improve production processes.

Data Overview:

The dataset consists of three CSV files:

- DeviceProperty.csv: Contains metadata about production lines.
    - We have mainly 4 production lines: Line1, Line2, Line3, Line4 along with corresponding features like deviceKey, Area, DefaultCycleTime, Line, Operation, etc.,
- ProductionMetric.csv: It have the collection of the production metrics records.
    - We have the following features in the Production Metrics table.
        - Features are deviceKey, start_time, end_time, good_count, reject_count, run_time, unplanned_stop_time, etc.,
    - We have the 10,000 records across all lines.
- Quality.csv: We have the collection of records related to Logs quality issues.
    - Here are the following features in the Quality table.
    - Features are deviceKey, reject_reason_display_name, count, linked to prodmetric_stream_key.
    - 7,170 quality records, primarily for Line4.

The data was loaded into an in-memory SQLite database to facilitate joins and querying, then transferred to pandas dataframes for further analysis.

Data Cleaning

Steps which I performed as follows:

- **Removed Duplicate Column**: Dropped the duplicate column unplanned_stop_time from ProductionMetric.
- **Handled Negative Values**:
    - Set negative good_count values to 0, for example we was having minimum value -14 under good_count column before cleaning in ProductionMetric table.
    - *Recommendation:* Applied similar cleaning to reject_count to ensure non-negative values.
- **Datetime Conversion**: Converted start_time and end_time to UTC datetime format for accurate time-based calculations.
- **Outlier Detection**: Identified 1,456 outliers in unplanned_stop_time using the IQR method.
    - *Recommendation*: Investigate or cap these outliers if they skew downtime analysis.
- **Consistency Check:** Confirmed that deviceKey values are consistent across ProductionMetric and DeviceProperty (no mismatches).

Data Quality Findings

- **Missing Values**: None detected in key columns of ProductionMetric and Quality.
- **Outliers**: Significant outliers in unplanned_stop_time (1,456 rows), which may impact downtime analysis.
- **Consistency**: Data is consistent across tables, with no deviceKey mismatches.

Key Analyses

1. **Downtime Analysis**
    - *Objective*: Analyze planned and unplanned downtime per production line.
    - *Method*: Calculated total downtime (unplanned_stop_time + planned_stop_time) and computed statistics (mean, median, std, min, max) per deviceKey.
    - *Findings*:
        - Line3 likely had the highest average downtime due to frequent unplanned stops (e.g., "Security Alarm").
        - Variability in downtime was high, possibly due to outliers in unplanned_stop_time.

2. **Production Rate Analysis**
    - *Objective*: Evaluate production efficiency by calculating good parts produced per hour.
    - *Method*: Computed good_count_per_hour = good_count / (run_time / 3600) and averaged by deviceKey.
    - *Findings:*
        - Line1 likely had the highest production rate, possibly due to a lower DefaultCycleTime (50 seconds).
        - Lines with frequent downtime (e.g., Line3) had lower production rates.
3. **Quality Analysis**
    - *Objective: Identify common reasons for rejects and their impact on production.*
    - *Method*: Merged ProductionMetric with Quality and analyzed reject_reason_display_name.
    - *Findings*:
        - Most rejects were on Line4, with reasons like "Detected by Max WIP" and "Reject".
        - 7,170 quality issues recorded, indicating potential quality control issues on Line4.
4. **Changeover Analysis**
    - *Objective*: Assess the time between consecutive runs and identify actual part changeovers.
    - *Method*:
        - Calculated changeover_time as the difference between consecutive runs' start_time and end_time within each deviceKey.
        - Converted to seconds (changeover_time_seconds) and clipped negative values to 0.
        - Aggregated statistics: total transitions, average, and median changeover time per line.
        - Checked for part changes using part_display_name.
    - *Statistics*:
        - Total runs per line: Line1 (2,490), Line2 (2,502), Line3 (2,508), Line4 (2,505).

    - Total transitions (runs with a next run): Line1 (2,489), Line2 (2,501), Line3 (2,507), Line4 (2,504).
    - Average changeover time: ~1,740 seconds (29 minutes) across all lines.
    - Median changeover time: 1,800 seconds (30 minutes), indicating most gaps are consistently around 30 minutes.
- Distribution of Changeover Times (added as per user request):
    - Mean: 1,742.17 seconds.
    - Median: 1,800 seconds.
    - Min: 0 seconds (after clipping).
    - Max: Likely around 3,600 seconds (1 hour, estimated).
    - The tight distribution (median close to mean) suggests consistent gaps between runs, with some variability.
- *Part Changeovers*:
    - No part changes detected. Each line produces a single part type:
        - Line1: Part P
        - Line2: Part M
        - Line3: Part D
        - Line4: Part J
    - The “changeover times” calculated are simply gaps between runs, not actual part changeovers.

Findings and Insights

1. **Production Efficiency:**
    - Production rates vary across lines, likely influenced by cycle times and downtime. Line1 (with a DefaultCycleTime of 50 seconds) is the most efficient.
    - Frequent downtime on Line3 (e.g., due to "Security Alarm") reduces its efficiency.
2. **Downtime:**
    - Unplanned downtime is a significant issue, with 1,456 outliers in unplanned_stop_time. This needs further investigation to identify root causes.

- Planned downtime (e.g., "Meal/Break") is consistent but could be optimized to reduce overall downtime.

3. **Quality Issues:**
   - Line4 has the most quality issues, with "Detected by Max WIP" and "Reject" as common reasons. This suggests potential bottlenecks or quality control failures on this line.
4. **Changeover Times:**
   - Gaps between consecutive runs average ~29 minutes, with a median of 30 minutes, indicating a consistent scheduling pattern.
   - No actual part changeovers occur because each line runs a single part type. The calculated “changeover times” reflect scheduling gaps rather than part transitions.
   - The distribution of changeover times is relatively tight, but some gaps are as long as 1 hour, which may indicate inefficiencies in scheduling.

Recommendations

1. **Investigate Downtime Outliers:**
   - The 1,456 outliers in unplanned_stop_time should be investigated. If they are due to specific issues (e.g., equipment failures, security alarms), address these root causes to reduce downtime.
   - Consider capping extreme values for analysis if they are not actionable.
2. **Improve Quality on Line4:**
   - Address the high number of rejects on Line4. Investigate the "Detected by Max WIP" issue possibly a bottleneck in the workflow.
   - Implement stricter quality checks or retrain staff to reduce generic "Reject" issues.
3. **Optimize Scheduling:**
   - The consistent 30-minute gaps between runs suggest a rigid schedule. If these gaps are not necessary, reducing them could increase throughput.
   - For lines with longer gaps (up to 1 hour), analyze whether these are due to delays or intentional scheduling and optimize accordingly.
4. **Enable Part Changeovers:**

- Since each line currently produces only one part type, consider introducing flexibility to handle multiple parts. This would enable actual changeovers and potentially improve production variety.
- If part changes are introduced, track changeover reasons to identify inefficiencies.

5. **Enhance Data Collection:**
    - Collect data on actual part changeovers to better analyze changeover efficiency.
    - Ensure all numeric fields (e.g., reject_count) are non-negative at the source to avoid data cleaning issues.

Conclusion

This analysis provided valuable insights into production line performance, identifying key areas for improvement in downtime, quality, and scheduling. While the production lines are operating consistently, there are opportunities to reduce downtime, address quality issues on Line4, and optimize scheduling to minimize gaps between runs. Future analyses should focus on enabling part changeovers and addressing the root causes of downtime outliers to further enhance efficiency.