MANUFACTURING DOWNTIME REPORT

Round Code
DEPI_ONL2_DAT1_G5

Presented To
Digital Egypt Pioneers Initiative

Githup link

Table Of Contents

1 -Introduction

We’re a team that worked on a project called "**Manufacturing Downtime Analysis & Optimization Using Data Analytics** "to help a beverage production facility run smoother.

Downtime -when machines stop unexpectedly- can really slow things down and cost money. Our goal was to figure out why these stoppages happen and how to reduce them.

Using tools like Python, SQL, and Power BI, we analyzed data from 38 production batches. We found that downtime added up to 1,398 minutes, with 776 minutes caused by **human mistakes** and 612 minutes due to other issues like **machine breakdowns**.

The biggest problem was **inventory shortages**, causing 439 minutes of delays. On average, operators worked at 80% efficiency, machines at 83.4%, but only 21% of batches hit our efficiency target. We’ve come up with practical ideas to fix these issues, like better inventory tracking and more training for the team.

1.1 - What We Set Out to Do

As we planned in our proposal, downtime from machine failures, human errors, or process hiccups can hurt production. We wanted to dig into the data to find the root causes and suggest ways to keep things running more smoothly. Here’s what we aimed for:

1. Clean and organize the data so it’s ready to analyze.
2. Find out what’s causing downtime and where we’re losing efficiency.
3. Set up a database to make data easy to work with.
4. Create dashboards in Power BI to show our findings in real time.

We also wanted to measure things like how often machines break down, how long they take to fix, and how much downtime is due to human errors.

2-Methodology page

2.1-What We Worked With

We used four sets of data:

- **Downtime Factors**: A list of 12 reasons why machines stop, like "machine failure" or "human error."
- **Line Downtime**: Details on downtime for each of the 38 batches.
- **Line Productivity**: Information on each batch, like the date, product, operator, and how long it took.
- **Products**: A list of 6 products, their flavors, sizes, and expected batch times.

2.1.1- Step 1: Cleaning the Data (Python)

In our Python work we:

- Loaded the data and checked for errors or missing pieces.
- Filled in gaps in the downtime data with zeros.
- Fixed the start and end times for each batch, especially for overnight shifts, and calculated how long each batch took.
- Figured out which shift (Night, Morning, Afternoon, Evening) each batch happened in.
- Calculated the total downtime for each batch and how efficient each batch was by comparing actual time to downtime.
- Added details about what caused the downtime and whether it was a human error or not.
- Saved the cleaned data for the next steps.

```python
# Group by operator and calculate total downtime
total_downtime_per_operator = data.groupby('Operator')['Total Downtime'].sum().reset_index()

# Find the operator with the highest downtime
operator_with_highest_downtime = total_downtime_per_operator.loc[total_downtime_per_operator['Total Downtime'].idxmax()]
print(f"Operator with the highest total downtime: {operator_with_highest_downtime['Operator']} ({operator_with_highest_downtime['Total Downtime']} minutes)"
```

Python

```
Operator with the highest total downtime: Charlie (384 minutes)
```

2.1.2- Step 2: Digging Deeper (SQL)

Using SQL we:

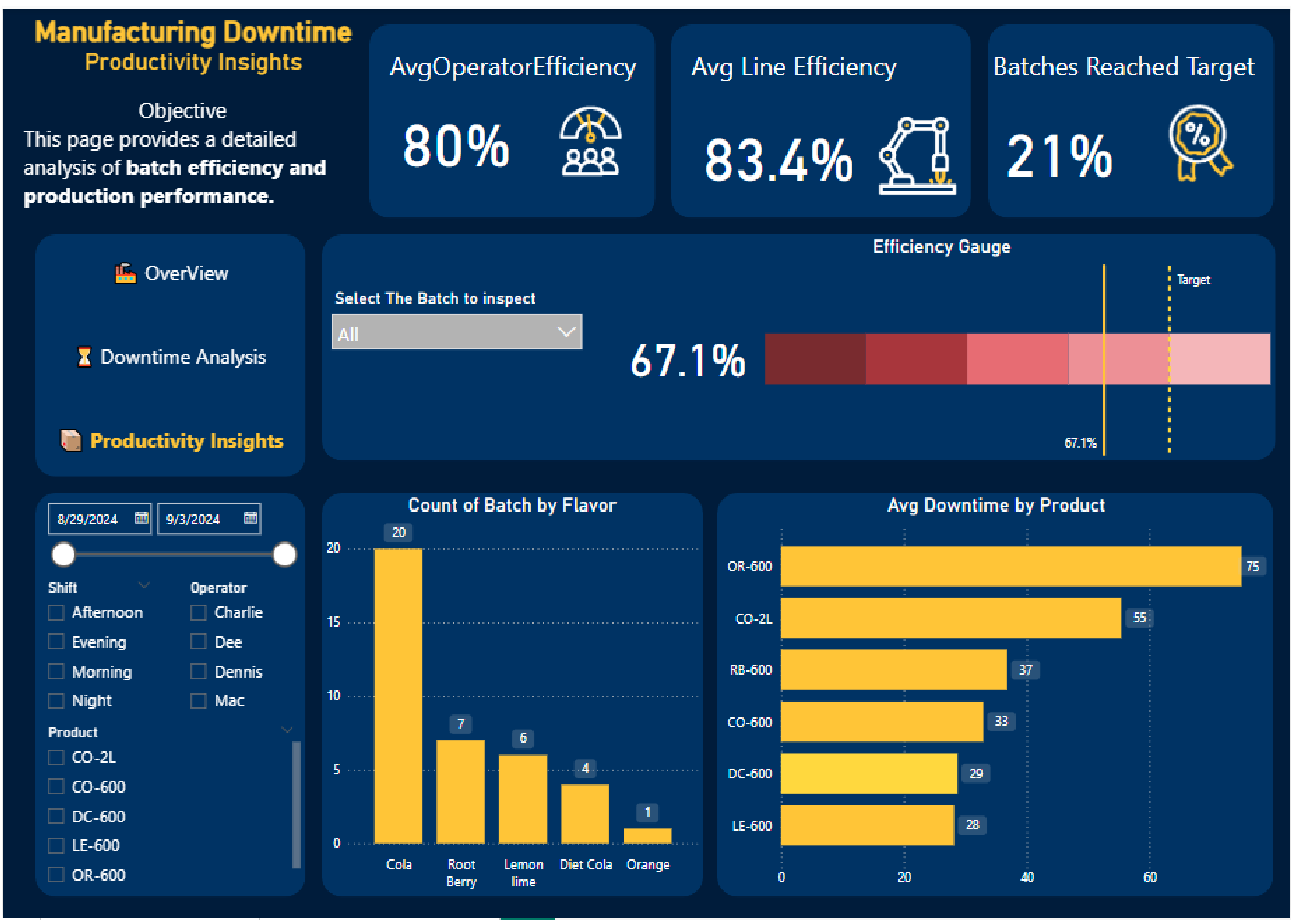
- Renamed some columns to make them clearer, like changing "Efficiency" to "BatchOverallEfficiency."
- Looked at how much downtime each operator and shift had, and how efficient they were.
- Ranked batches by downtime for each product to see which ones had the most issues.
- Broke down the reasons for downtime to see how much was due to human errors versus other causes.

```sql
WITH DowntimeDetails AS (
    SELECT
        Batch,
        HumanErrors,
        DowntimeFactorsDescriptions,
        TotalDowntime,
        TotalDowntimewithHumanErrors,
        TotalDowntimewithoutHumanErrors
    FROM
        line_downtime
),
ParsedFactors AS (
    SELECT
        Batch,
        HumanErrors,
        TotalDowntime,
        TotalDowntimewithHumanErrors,
        TotalDowntimewithoutHumanErrors,
        ROW_NUMBER() OVER (PARTITION BY Batch ORDER BY (SELECT NULL)) AS RowNum,
        TRIM(value) AS DowntimeFactor
    FROM DowntimeDetails
    CROSS APPLY STRING_SPLIT(DowntimeFactorsDescriptions, ',')
),
ParsedErrors AS (
    SELECT
        Batch,
        TRIM(value) AS ErrorType,
        ROW_NUMBER() OVER (PARTITION BY Batch ORDER BY (SELECT NULL)) AS RowNumError
    FROM DowntimeDetails
    CROSS APPLY STRING_SPLIT(HumanErrors, ',')
)
SELECT
    p.Batch,
    p.DowntimeFactor,
    CASE
        WHEN pe.ErrorType = 'Yes' THEN 'Yes'
        ELSE 'No'
    END AS HumanError,
    CASE
        WHEN pe.ErrorType = 'Yes' THEN p.TotalDowntimewithHumanErrors
        WHEN pe.ErrorType = 'No' THEN p.TotalDowntimewithoutHumanErrors
    END AS [Total Downtime with Human or without Human]
FROM ParsedFactors p
JOIN ParsedErrors pe
    ON p.Batch = pe.Batch
    AND p.RowNum = pe.RowNumError
ORDER BY p.Batch, p.RowNum;
```

2.1.3- Step 3: Showing Our Findings (Power BI)

We created three dashboards in Power BI to make the data easy to understand:

- **Overview**: Shows the big picture, like total batches (38), average batch time (101.5 minutes), average downtime (36.5 minutes), and overall efficiency (67.08%). It also compares operators and shows efficiency over time.
- **Downtime Analysis**: Highlights total downtime (1,398 minutes), with 776 minutes from human errors and 612 minutes from other issues. It shows the top reasons, like inventory shortages (439 minutes).
- **Productivity Insights**: Shows operator efficiency (80%), machine efficiency (83.4%), and that only 21% of batches hit our target. It also breaks down efficiency by product and flavor.



3-Key Analytical Questions & Purpose

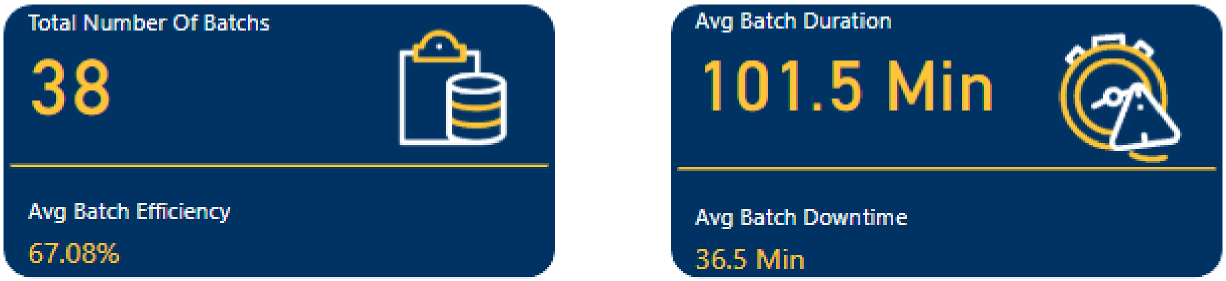
-To guide our analysis, we came up with nine key questions to help us understand downtime and efficiency better. These questions shaped our work and made sure we focused on the most important areas:

- **Q1: What are the most common downtime factors?**
  We wanted to know the biggest reasons machines stop so we can focus on fixing them.
- **Q2: Which shift experiences the most downtime?**
  This helps us see if certain shifts need more support to keep things running smoothly.
- **Q3: How does downtime vary by operator? (Operator Analysis)**
  We looked at how each operator's performance affects downtime to see who might need extra training.
- **Q4: What is the average efficiency across all batches?**
  This gives us a big-picture view of how well the production process is working.
- **Q5: What is the average downtime per product? (Product Analysis)**
  We wanted to see which products have the most downtime so we can improve their processes.
- **Q6: Human and non-human errors?**
  Splitting downtime into human errors and other issues helps us know where to focus our improvements.
- **Q7: Batches efficiency breakdown (top and bottom 3)?**
  This shows us which batches are doing well and which ones need the most help.
- **Q8: Defining significant downtime factors per product (for deeper analysis)?**
  We dug deeper to see if certain products have specific issues causing downtime.
- **Q9: Percentage of targeted batches efficiency?**
  This tells us how many batches are meeting our efficiency goals and where we need to improve.

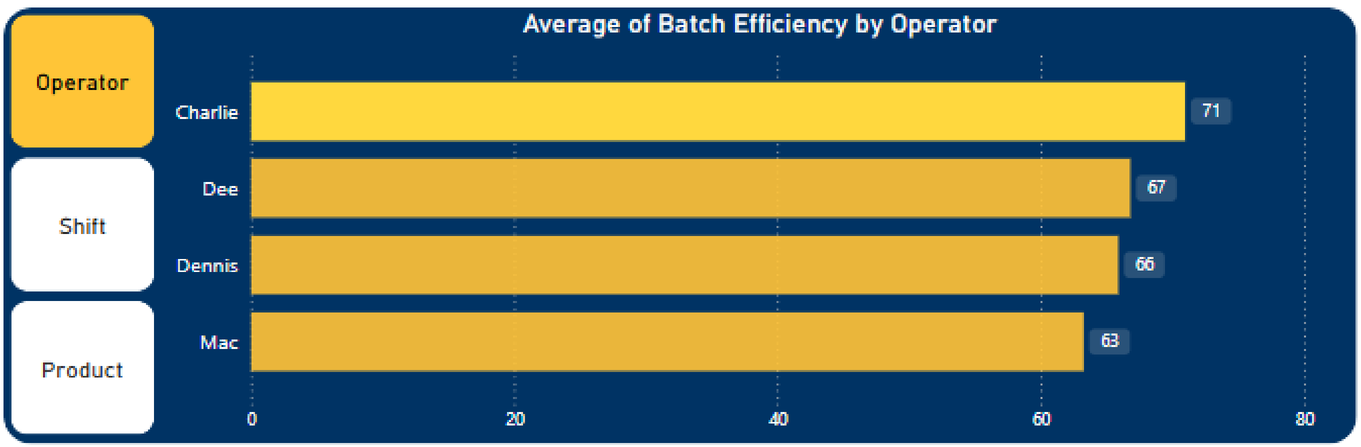
4-What We Found

From the Dashboards

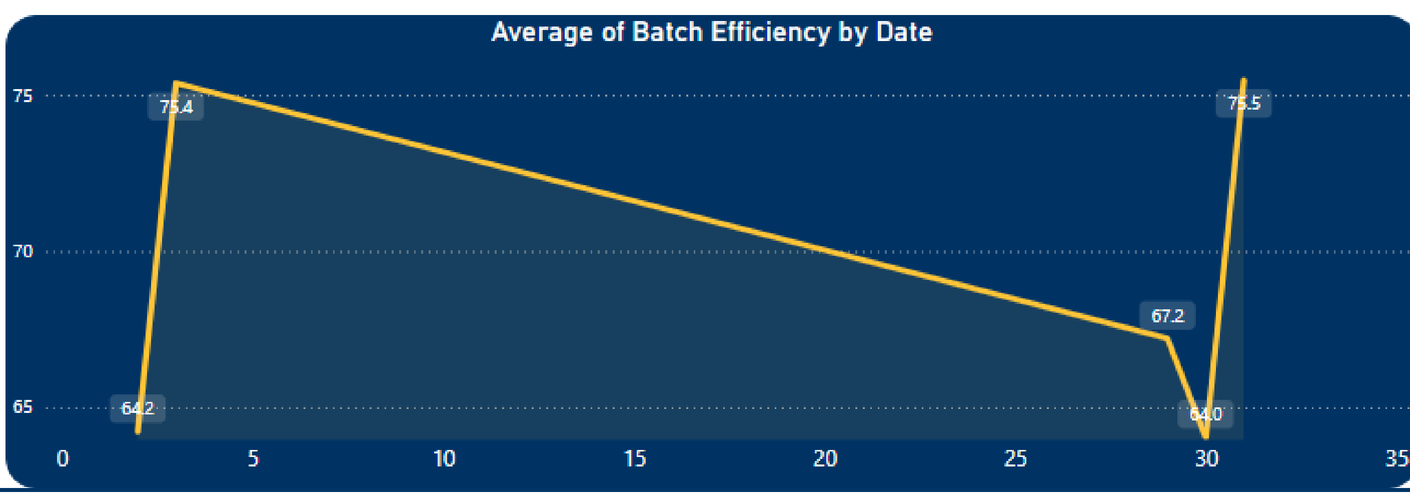
- **The Big Picture (Overview):**



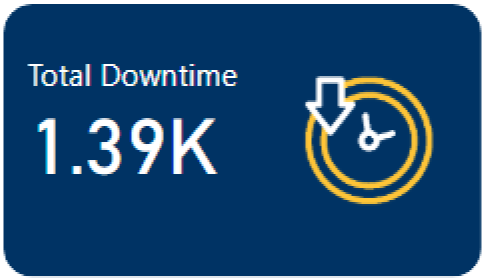
We looked at 38 batches, with each taking about 101.5 minutes on average, including 36.5 minutes of downtime. Overall efficiency was 67.08%.

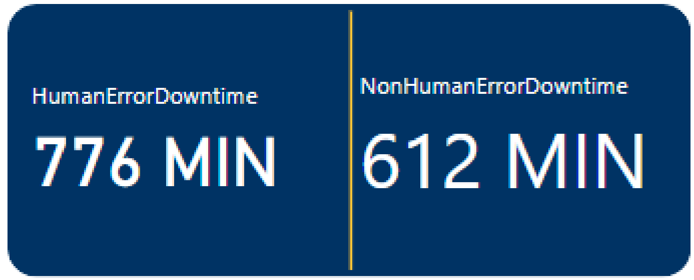


Charlie was the most efficient operator at 71%, followed by Dee (67%), Dennis (66%), and Mac (63%). Efficiency dropped from 75.4% on August 29 to 64% on September 2, but went up a bit to 67.2% on September 3.

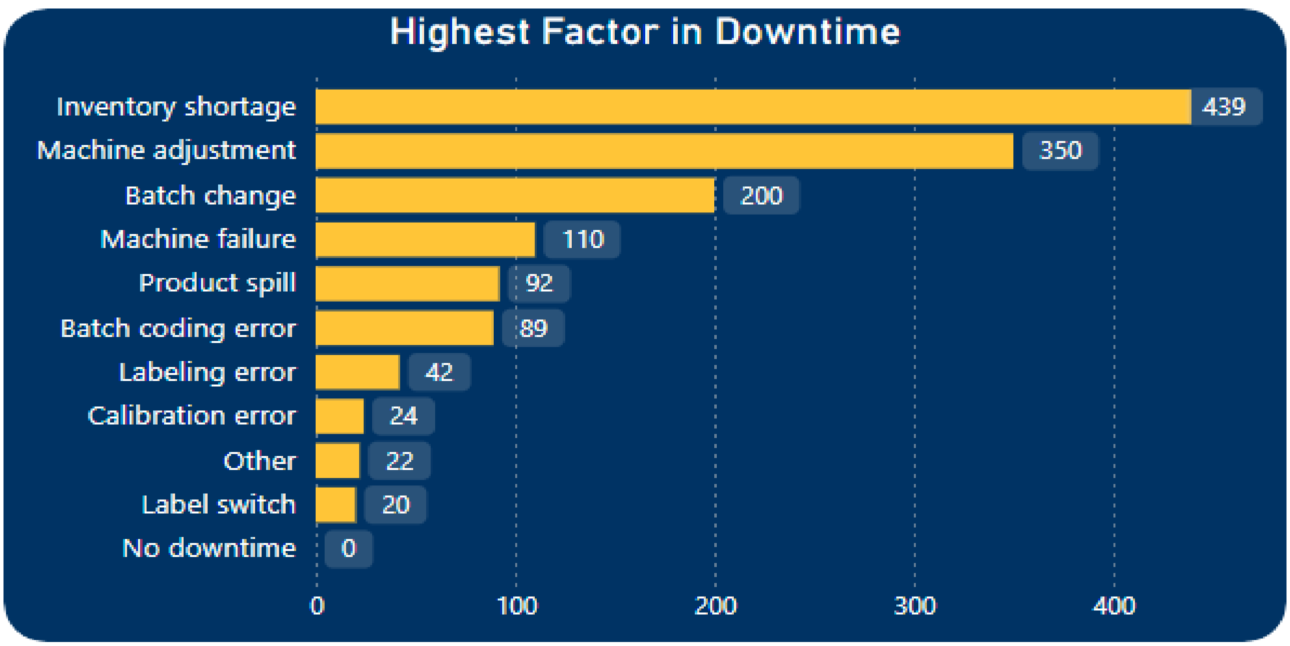


- **Downtime Details (Downtime Analysis):**



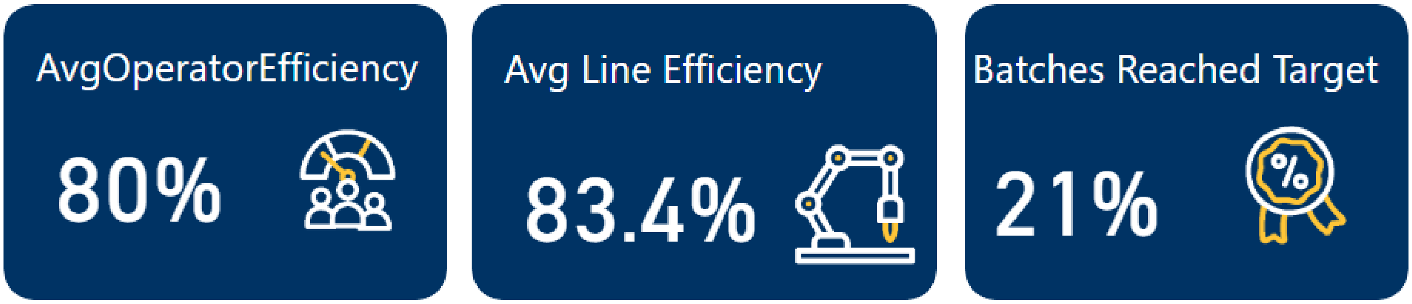


Total downtime was 1,398 minutes. Human errors caused 776 minutes (44.09%), and other issues caused 612 minutes (55.91%).

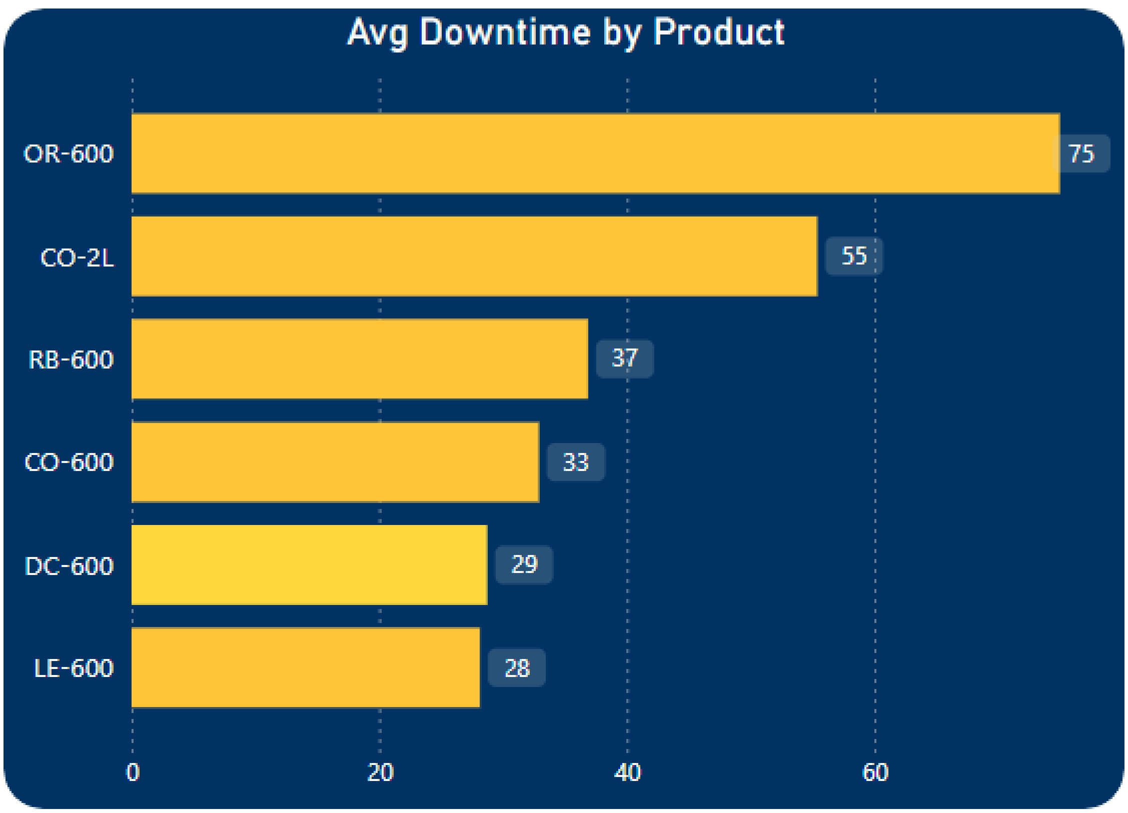


The biggest issue was inventory shortages (439 minutes), followed by machine adjustment (350 minutes) and batch change (200 minutes). Downtime was highest on August 29 (503 minutes) and lowest on September 3 (155 minutes).

- **Efficiency Insights (Productivity Insights):**



Operators were 80% efficient on average, and machines were 83.4% efficient. Only 21% of batches met our efficiency goal.



Cola products had the most batches (20), followed by Root Berry (7) and Lemon Lime (6). OR-600 had the highest downtime per batch (75 minutes), while LE-600 had the least (26 minutes).

5- General Conclusion and recommendation

5.1 -From Our Analysis

- **Operator Performance**:
- Charlie handled the most batches and was the most efficient (71%), but had some big human errors, like a 44-minute delay in batch 422126 due to a coding mistake.
- Mac had the lowest efficiency (63%) and struggled with OR-600 batches.
- **Shift Performance**:
- Afternoon and Evening shifts had the most downtime, probably because they had more batches.
- Night shifts had less downtime but were less efficient, maybe due to tiredness.
- **Product Performance**:
- OR-600 and CO-2L batches had more downtime, often due to inventory issues or machine adjustments.
- LE-600 batches were the most efficient, with only 26 minutes of downtime on average.

5.2-Why This Matters

-Our proposal highlighted the benefits of this project, and we've seen them come to life:

- **Better Efficiency**: Knowing that inventory shortages and machine failures are big issues helps us plan better and keep production running.
- **Smarter Decisions**: Our dashboards and metrics, like downtime percentage, help managers make informed choices quickly.
- **Using Resources Wisely**: We can now focus on fixing the biggest problems, like inventory, and adjust shifts to work better.
- **Room to Grow**: Our database and dashboards can be used for other production lines in the future, and we can keep improving with the insights we've gained.
- **Saving Money and Time**: Less downtime means more production, which saves money and boosts profits.

5.3-What We Suggest

Here are some practical steps to improve things:

1. **Fix Inventory Issues**:

- Set up a system to track inventory in real time to avoid shortages (439 minutes of downtime).
- Check inventory regularly to make sure there's enough for production.

2. **Train the Team**:

- Give extra training to operators like Mac, who had the lowest efficiency (63%), to help them avoid mistakes like coding errors or machine adjustments.
- Even Charlie, who did well, could use training to cut down on errors like the 44-minute delay in batch 422126.

3. **Maintain Machines Better**:

- Schedule regular maintenance to prevent machine failures (110 minutes of downtime).
- Look into why machines need so many adjustments (350 minutes) and see if they need upgrades.

4. **Improve Shifts**:

- Add more support during Afternoon and Evening shifts to reduce downtime.
- Help Night shifts work better by adjusting their workload or adding support.

5. **Set Better Goals**:

- Aim for a realistic efficiency target since we're at 67.08% now, and only 21% of batches hit the goal.
- Focus on improving OR-600 and CO-2L batches, which had the most downtime.

6 - Who We Are

Our team worked together to make this project a success. Here’s who did what:

No	Team Member	Role
1	Kareem Ashraf Mohamed	Team Leader: Kept us on track and coordinated everything.
2	Mahmoud Waleed Mahmoud	Data Cleaner: Made sure our data was accurate and ready to use.
3	Jhad Ibrahem Ahmed	KPI Expert: Set up the goals and metrics we’d measure.
4	Mohamed Shaaban Mohamed	Data Checker: Double-checked our work for accuracy.
5	Mohamed Abdel Rahman Mohamed	Dashboard Creator: Built the visuals to show our results.

We each had a role, and together, we made sure everything from data cleaning to final visuals went smoothly.