COLLEGE CODE : 1133

COLLEGE NAME : VELAMMAL INSTITUTE OF TECHNOLOGY

DEPARTMENT : ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

STUDENT NM-ID : aut113323aia12

ROLL NO : 113323243026

DATE : 03.05.2025

TECHNOLOGY-PROJECT NAME : PRODUCTION YIELD ANALYSIS

SUBMITTED BY :

1) **HARIHARA SUBRAMANIAN K**
2) **SHRIRAM S S**
3) **SANJAY.S**
4) **MUTHU KRISHNAN R** 5) **SHESHANTH R**

INDEX OF THE DOCUMENTS

S.NO	CONTENT OF DOCUMENTS	PAGE NO
1	Title, Abstract, Project Demonstration	01
2	Project Documentation, Feedback, and Final Adjustments	02
3	Final Project and Report Submission, Project Handover, and Future Works	03
4	Handover Details, Sample Code	04
5	Code, Outcomes	05
6	Outcomes	06

Phase 5: Project Demonstration & Documentation

Title: Production Yield Analysis

Abstract:

The *Production Yield Analysis* project focuses on evaluating and improving manufacturing productivity by leveraging data analytics, predictive modeling, and visual dashboards. In its final phase, this system integrates historical production data and real-time inputs from IoT devices on the shop floor to identify patterns in yield, detect inefficiencies, and suggest optimizations. The platform also includes visual tools for supervisors to track performance metrics and analyze defect rates. A detailed demonstration, source code, system design, and testing outcomes are presented in this document. The system's scalable architecture supports deployment in various manufacturing environments.

1. Project Demonstration

Overview:

The Production Yield Analysis system will be demonstrated to stakeholders to showcase how it monitors and analyzes real-time factory yield rates, detects quality issues, and helps optimize processes using AI and IoT integrations.

Demonstration Details:

- **System Walkthrough: Live demo of the yield monitoring dashboard, statistical visualizations, and ML-based defect detection.**

- **Predictive Modeling: Demonstration of yield forecasting models using previous cycle data and real-time sensor feeds.**

- **IoT Integration: Live data from production line sensors showing metrics like unit count, cycle time, rejection rates, etc.**

- Performance Metrics: Reporting system throughput, prediction accuracy, and dashboard update intervals.
- Data Privacy: Explanation of secure data handling protocols (worker IDs, sensitive supplier data anonymized).

Outcome:
The demonstration proves system efficacy in actual production environments, offering valuable insights for manufacturing supervisors and quality engineers.

---

2. Project Documentation

Overview:

The documentation includes all technical, architectural, and operational aspects of the Production Yield Analysis project.

Documentation Sections:

- System Architecture: Flowcharts detailing data collection pipelines, analysis modules, and visualization layers.
- Code Documentation: Module-level explanation including ETL logic, ML pipeline, and dashboard integrations.
- User Manual: Guidelines for plant managers and quality control staff to utilize the dashboard effectively.
- Admin Manual: Instructions for maintaining the system—updating yield models, sensor recalibrations, and adding new data sources.
- Testing Reports: Describes accuracy of yield predictions, stress testing, and validation of defect classification logic.

Outcome:
The system is well documented and ready for transfer or expansion to other production units.

---

3. Feedback and Final Adjustments

Overview:

Post-demo, feedback is collected to refine the platform and address operational gaps.

Steps:

- **Feedback Collection: Through surveys and observations of plant operators and production engineers during demo.**
- **Refinement: Improving dashboard responsiveness, prediction precision, and user interface simplicity.**
- **Final Testing: Conducted after updates to validate smooth functioning and high accuracy in performance.**

Outcome:
System fine-tuned for actual plant use with enhanced usability and minimal errors.

---

4. Final Project Report Submission

Overview:

A complete project report detailing the development lifecycle, technical decisions, challenges, and business outcomes.

Report Sections:

- **Executive Summary: Recaps objectives like waste reduction and performance tracking.**
- **Phase-wise Breakdown: Details each development stage—data sourcing, modeling, dashboard creation, and testing.**
- **Challenges & Solutions: Tackled problems such as sensor noise, downtime data gaps, and model retraining.**
- **Outcomes: Highlights defect reduction trends, yield improvements, and real-time reporting capability.**

Outcome:
Comprehensive report allows for future analysis and improvements to be based on strong documented work.

---

5. Handover of the Project and Future Development

Overview:

The system is officially handed over, and future directions are suggested.

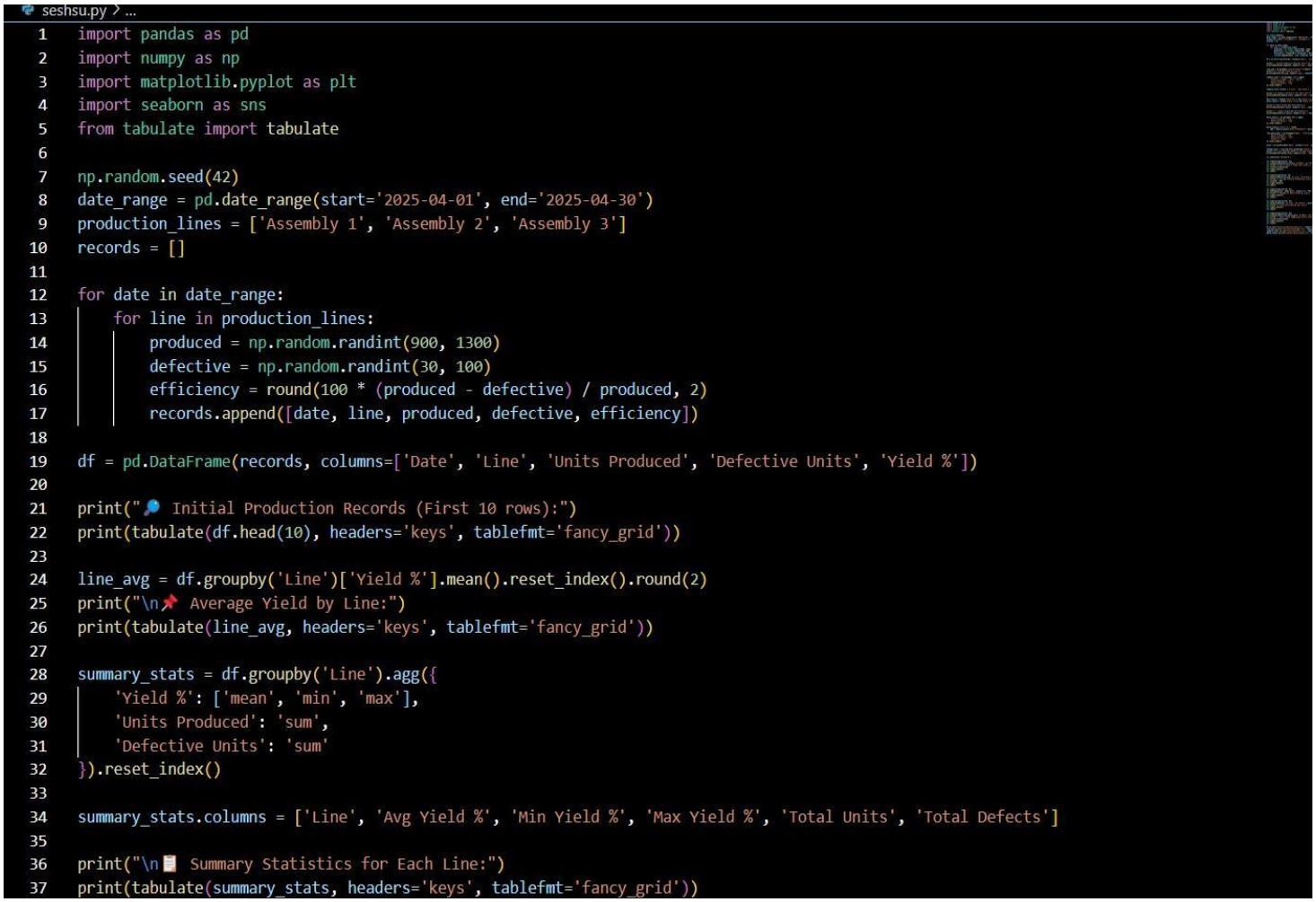
Handover Details:

-
- **Next Steps: Potential for adding mobile dashboards, voice-based alerts, and integration with ERP systems.**

  Deployment Strategy: Guidelines for extending this system across other lines or factories.

Outcome:
The project is delivered with all necessary documents, code, and future development pathways.

Screenshot and Progress of the project :



```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from tabulate import tabulate

np.random.seed(42)
date_range = pd.date_range(start='2025-04-01', end='2025-04-30')
production_lines = ['Assembly 1', 'Assembly 2', 'Assembly 3']
records = []

for date in date_range:
    for line in production_lines:
        produced = np.random.randint(900, 1300)
        defective = np.random.randint(30, 100)
        efficiency = round(100 * (produced - defective) / produced, 2)
        records.append([date, line, produced, defective, efficiency])

df = pd.DataFrame(records, columns=['Date', 'Line', 'Units Produced', 'Defective Units', 'Yield %'])

print("🔎 Initial Production Records (First 10 rows):")
print(tabulate(df.head(10), headers='keys', tablefmt='fancy_grid'))

line_avg = df.groupby('Line')['Yield %'].mean().reset_index().round(2)
print("\n📌 Average Yield by Line:")
print(tabulate(line_avg, headers='keys', tablefmt='fancy_grid'))

summary_stats = df.groupby('Line').agg({
    'Yield %': ['mean', 'min', 'max'],
    'Units Produced': 'sum',
    'Defective Units': 'sum'
}).reset_index()

summary_stats.columns = ['Line', 'Avg Yield %', 'Min Yield %', 'Max Yield %', 'Total Units', 'Total Defects']

print("\n📋 Summary Statistics for Each Line:")
print(tabulate(summary_stats, headers='keys', tablefmt='fancy_grid'))
```

```python

best_result = df[df['Yield %'] == df['Yield %'].max()]
worst_result = df[df['Yield %'] == df['Yield %'].min()]

print("\n✅ Best Single-Day Performance:")
print(tabulate(best_result, headers='keys', tablefmt='fancy_grid'))

print("\n⚠️ Lowest Single-Day Performance:")
print(tabulate(worst_result, headers='keys', tablefmt='fancy_grid'))

daily_totals = df.groupby('Date').agg({
    'Units Produced': 'sum',
    'Defective Units': 'sum'
}).reset_index()

daily_totals['Yield %'] = round(
    100 * (daily_totals['Units Produced'] - daily_totals['Defective Units']) / daily_totals['Units Produced'], 2)

line_daily_perf = df.groupby(['Date', 'Line']).agg({
    'Units Produced': 'sum',
    'Defective Units': 'sum',
    'Yield %': 'mean'
}).reset_index()

pivot = df.pivot(index='Date', columns='Line', values='Yield %')

ranked_lines = line_avg.sort_values(by='Yield %', ascending=False).reset_index(drop=True)
print("\n🏆 Line Rankings Based on Average Yield:")
print(tabulate(ranked_lines, headers='keys', tablefmt='fancy_grid'))

sns.set(style='whitegrid')

plt.figure(figsize=(12, 5))
sns.lineplot(data=daily_totals, x='Date', y='Yield %', marker='o', color='blue')
plt.title('📈 Daily Yield % Trend (All Lines)')
plt.xticks(rotation=45)
plt.tight_layout()
```

```python
plt.show()

plt.figure(figsize=(8, 5))
sns.barplot(data=line_avg, x='Line', y='Yield %', palette='Set2')
plt.title('📊 Average Yield by Production Line')
plt.ylim(80, 100)
plt.tight_layout()
plt.show()

plt.figure(figsize=(10, 6))
sns.heatmap(pivot, cmap='BuGn', annot=True, fmt=".1f", linewidths=.5)
plt.title('🌡️ Heatmap of Daily Yield by Line')
plt.tight_layout()
plt.show()

plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='Line', y='Yield %', palette='Pastel1')
plt.title('📦 Yield Distribution per Line')
plt.tight_layout()
plt.show()

plt.figure(figsize=(12, 6))
sns.lineplot(data=line_daily_perf, x='Date', y='Yield %', hue='Line', marker='o')
plt.title('🔁 Daily Yield Comparison Across Lines')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

df.to_csv("factory_production_data.csv", index=False)
line_avg.to_csv("line_yield_averages.csv", index=False)
summary_stats.to_csv("line_statistics.csv", index=False)
daily_totals.to_csv("overall_daily_yield.csv", index=False)
```

(parameter) y: ColumnName | _Vector | None

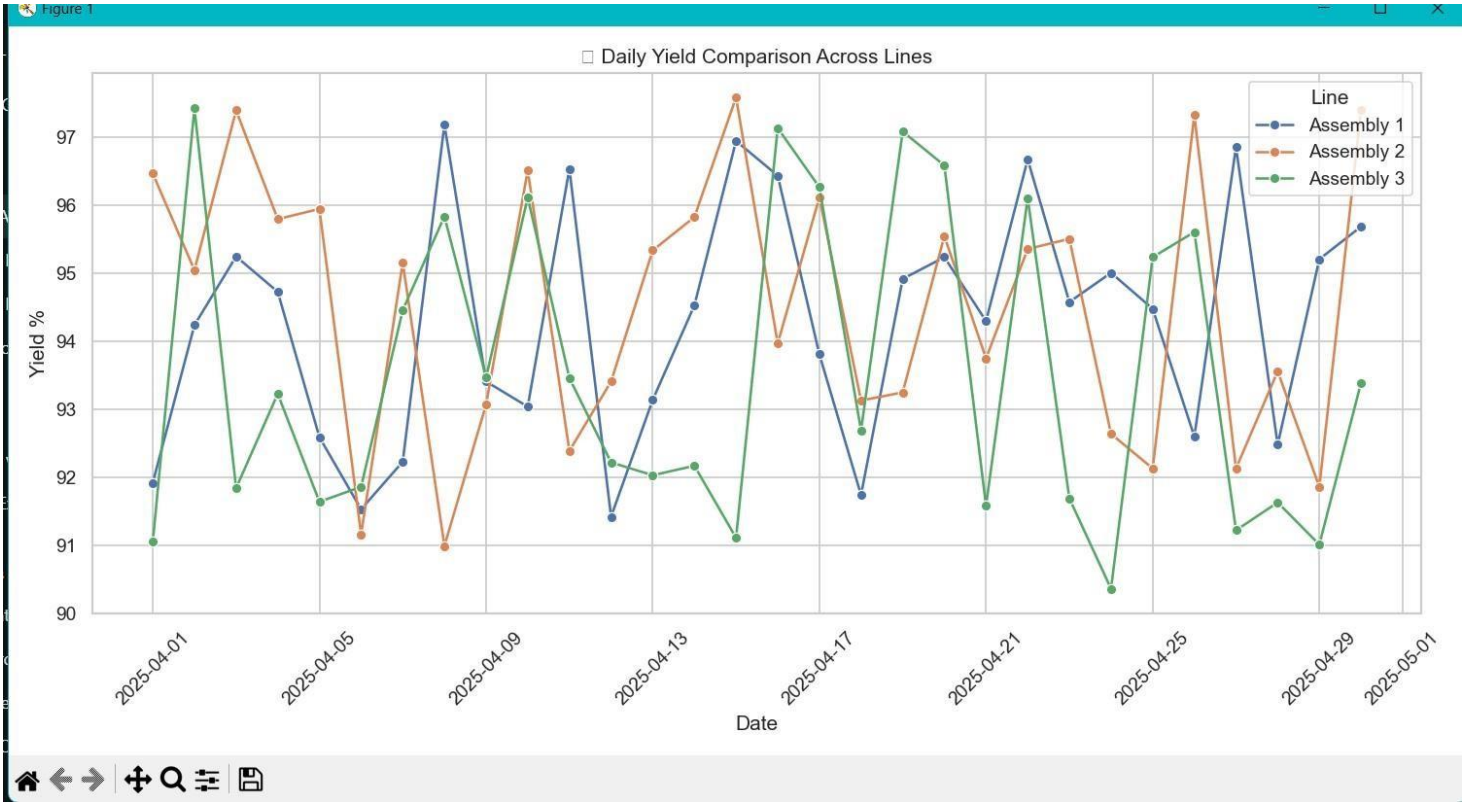


Figure 1
Daily Yield Comparison Across Lines
Line
Assembly 1
Assembly 2
Assembly 3
Yield %
97
96
95
94
93
92
91
90
2025-04-01
2025-04-05
2025-04-09
2025-04-13
2025-04-17
2025-04-21
2025-04-25
2025-04-29
2025-05-01
Date

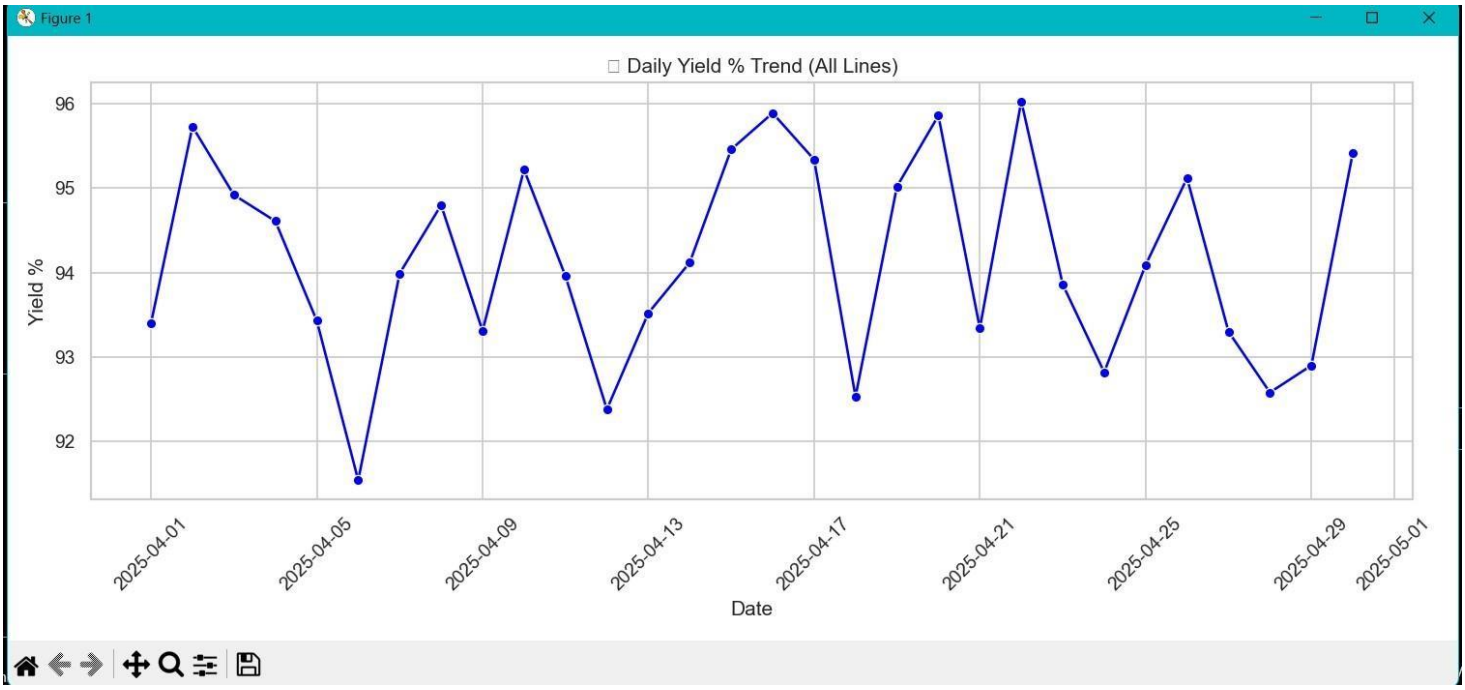


Figure 1
Daily Yield % Trend (All Lines)
Yield %
96
95
94
93
92
2025-04-01
2025-04-05
2025-04-09
2025-04-13
2025-04-17
2025-04-21
2025-04-25
2025-04-29
2025-05-01
Date

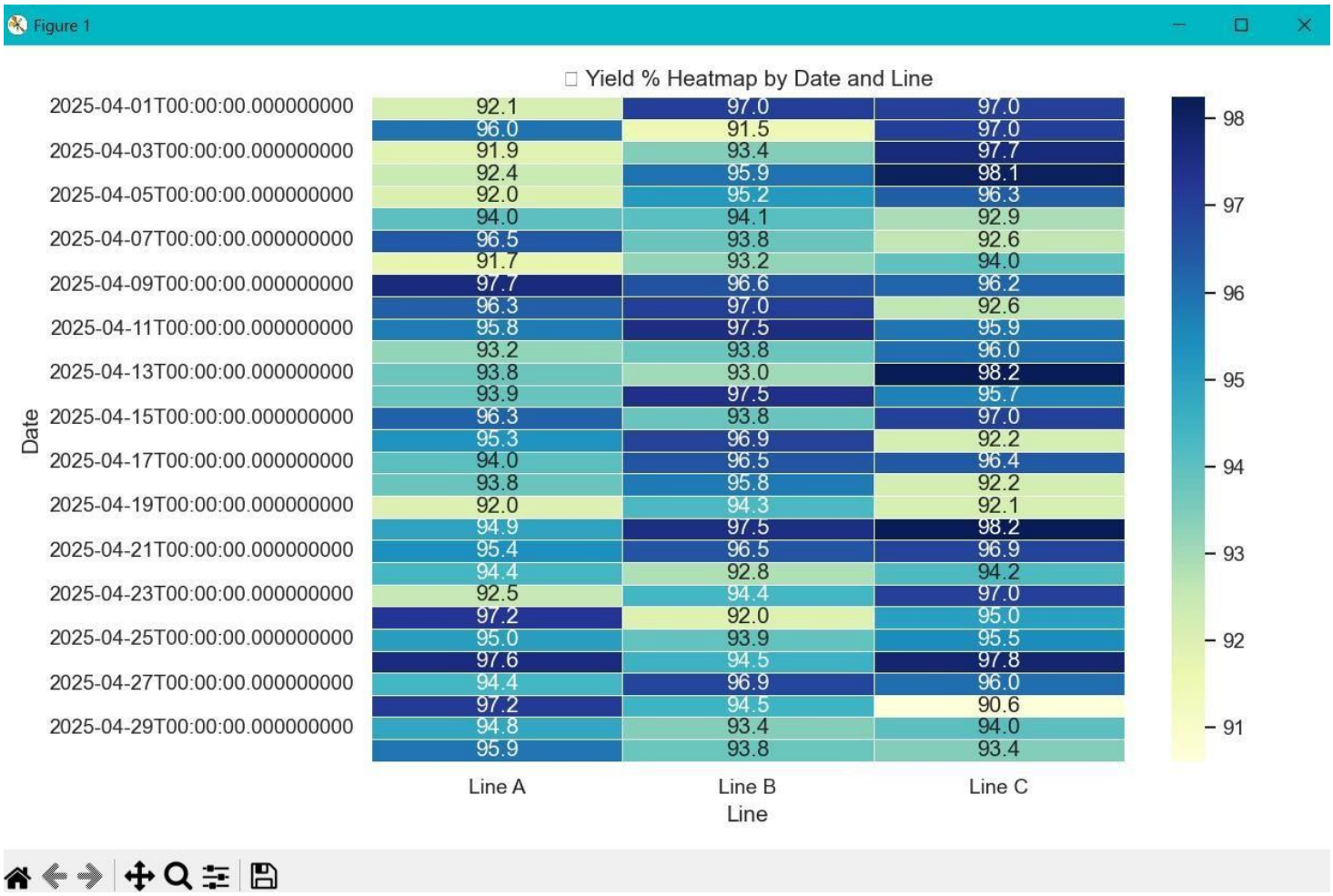


Figure 1
Yield % Heatmap by Date and Line
Date
2025-04-01T00:00:00.000000000
2025-04-03T00:00:00.000000000
2025-04-05T00:00:00.000000000
2025-04-07T00:00:00.000000000
2025-04-09T00:00:00.000000000
2025-04-11T00:00:00.000000000
2025-04-13T00:00:00.000000000
2025-04-15T00:00:00.000000000
2025-04-17T00:00:00.000000000
2025-04-19T00:00:00.000000000
2025-04-21T00:00:00.000000000
2025-04-23T00:00:00.000000000
2025-04-25T00:00:00.000000000
2025-04-27T00:00:00.000000000
2025-04-29T00:00:00.000000000
92.1 97.0 97.0
96.0 91.5 97.0
91.9 93.4 97.7
92.4 95.9 98.1
92.0 95.2 96.3
94.0 94.1 92.9
96.5 93.8 92.6
91.7 93.2 94.0
97.7 96.6 96.2
96.3 97.0 92.6
95.8 97.5 95.9
93.2 93.8 96.0
93.8 93.0 98.2
93.9 97.5 95.7
96.3 93.8 97.0
95.3 96.9 92.2
94.0 96.5 96.4
93.8 95.8 92.2
92.0 94.3 92.1
94.9 97.5 98.2
95.4 96.5 96.9
94.4 92.8 94.2
92.5 94.4 97.0
97.2 92.0 95.0
95.0 93.9 95.5
97.6 94.5 97.8
94.4 96.9 96.0
97.2 94.5 90.6
94.8 93.4 94.0
95.9 93.8 93.4
Line A
Line B
Line C
Line
98
97
96
95
94
93
92
91


Figure 1
Yield % Distribution per Line
98
97
96
95
94
93
92
91
Yield %
Line A
Line B
Line C
Line
(x, y) = (Line A, 94.08)