Metals in Focus: Analyzing Iron Concentration Patterns in Mining

By Shekela Mitchell Best

Shaping Our World: The Iron Story You Never Knew

Ever thought about what skyscrapers, bridges, cars, and even your home appliances have in common? It's iron! This metal is like the superhero behind the scenes, making all these things possible. But how does plain old iron become the magic ingredient that builds our world?

That's where I come in. I'm a data analyst at Metals R' Us, a mining company with a mission to extract and refine iron through a process that's as intricate as it is crucial.

In our mining process, we start with clumps of dirt and rock containing iron hidden within. Through advanced techniques in our flotation plant, we unlock the iron's potential, ensuring the high-quality material that forms the backbone of structures and objects shaping our everyday lives.

Preliminary Wrangling

In [1]:
```
### Import Libraries

# Data Manipulation
import pandas as pd

# Data Visualizaiton
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
#Import CSV file
df = pd.read_csv('MiningProcess_Flotation_Plant_Database.csv')
```

About the Dataset

The dataset contains **737,453 rows** and **24 columns**. If you want to access the data, you can find it on Kaggle. This dataset is sourced from real-world manufacturing plants, especially mining plants, making it a valuable resource for understanding real-world processes.

The main feature of primary interest in this dataset is `% Iron Concentrate`. This represents the purity and quality of the final iron product produced by the mineral processing plant. Metallurgists demand specific concentration grades from this product for smelting, making this value crucial in evaluating the effectiveness of the mineral processing operations.

To thoroughly investigate the quality of the concentrate products (`% Iron Concentrate`), several features will provide valuable support:

- `Iron Feed` : The initial iron content in the ore being processed.
- `Silica Feed` : The initial silica content in the ore.
- `Starch Flow` : The use of starch as a depressant for iron.
- `Amina Flow` : The use of amina to collect silica.
- `Ore Pulp Flow` : The flow of ore pulp into the flotation process.
- `Ore Pulp pH` : The pH level affecting chemical reactions.
- `Ore Pulp Density` : The solid density of the flotation feed.
- `Flotation Column Air Flow` : The amount of air affecting bubble formation.
- `Flotation Column Level` : The thickness of the flotation foam.

These features collectively offer valuable context and insights into how the processing plant's operational parameters influence the quality of the iron concentrate product.

Key Takeaways:

1. **Correlation Analysis:** The correlation matrix highlights that there are no strong linear correlations among the variables "% Iron Concentrate," "% Silica Concentrate," "Ore Pulp pH," and "Flotation Column 05 Level." While the correlation coefficients suggest moderate relationships, this analysis doesn't account for potential non-linear connections or other influential factors in the dataset.

2. **Concentration Patterns:** A dip in both iron and silica concentrations during certain hours indicates potential issues in the flotation plant's separation process. Further investigation is needed to pinpoint the root causes, which could range from operational decisions to equipment performance.

3. **Iron-Silica Correlation:** The negative correlation coefficient of -0.8005 between iron and silica concentrations supports the expected inverse relationship. Higher iron content typically corresponds to lower silica content in the ore concentrate, aligning with mineral processing practices.

4. **Monthly Comparison (June and July):** Analyzing iron concentration averages for June and July highlights a consistent pattern of lower iron concentrations on Tuesdays. The average iron concentration on Tuesdays was 64.45 for June and 64.35 for July. This fluctuation indicates potential operational variations on Tuesdays in both months, with June showing greater overall variability.

5. **Monthly Variation:** Assessing iron concentration levels across different months reveals no consistent patterns. Fluctuations occur without a discernible trend, suggesting the influence of various dynamic variables. The absence of distinct trends emphasizes the complexity of iron concentration determinants.

Cleaning the data

Upon reviewing the data, I observe that the data column is classified as an object, which indicates that it's stored as a string. Additionally, it's worth noting that some columns containing numeric data, such as "% Iron Concentrate," "% Silica Concentrate," "Ore Pulp pH," and "Flotation Column 05 Level," use commas as decimal separators instead of periods. To enhance its usability, I'll need to transform the date column into a timeseries format and address the numeric column issue by converting them to the appropriate numerical format, replacing commas with periods for accurate interpretation.

In [3]:
```
#Evalutaing the data
print(df.dtypes)

df.head()
```

```
date                           object
% Iron Feed                    object
% Silica Feed                  object
Starch Flow                    object
Amina Flow                     object
Ore Pulp Flow                  object
Ore Pulp pH                    object
Ore Pulp Density               object
Flotation Column 01 Air Flow   object
Flotation Column 02 Air Flow   object
Flotation Column 03 Air Flow   object
Flotation Column 04 Air Flow   object
Flotation Column 05 Air Flow   object
Flotation Column 06 Air Flow   object
Flotation Column 07 Air Flow   object
Flotation Column 01 Level      object
Flotation Column 02 Level      object
Flotation Column 03 Level      object
Flotation Column 04 Level      object
Flotation Column 05 Level      object
Flotation Column 06 Level      object
Flotation Column 07 Level      object
% Iron Concentrate             object
% Silica Concentrate           object
dtype: object
```

Out[3]:

	date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	...	Flotatio Colum 07 A Flo
0	2017-03-10 01:00:00	55,2	16,98	3019,53	557,434	395,713	10,0664	1,74	249,214	253,235	...	250,88
1	2017-03-10 01:00:00	55,2	16,98	3024,41	563,965	397,383	10,0672	1,74	249,719	250,532	...	248,99
2	2017-03-10 01:00:00	55,2	16,98	3043,46	568,054	399,668	10,068	1,74	249,741	247,874	...	248,07
3	2017-03-10 01:00:00	55,2	16,98	3047,36	568,665	397,939	10,0689	1,74	249,917	254,487	...	251,14
4	2017-03-10 01:00:00	55,2	16,98	3033,69	558,167	400,254	10,0697	1,74	250,203	252,136	...	248,92

5 rows × 24 columns

In [4]:
```
#Change the commas to decimals
df = pd.read_csv('MiningProcess_Flotation_Plant_Database.csv',decimal=",")
df.head()
```

Out[4]:

	date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	...	Flotatio Colum 07 A Flo
0	2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	...	250.88
1	2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	...	248.99
2	2017-03-10 01:00:00	55.2	16.98	3043.46	568.054	399.668	10.0680	1.74	249.741	247.874	...	248.07
3	2017-03-10 01:00:00	55.2	16.98	3047.36	568.665	397.939	10.0689	1.74	249.917	254.487	...	251.14
4	2017-03-10 01:00:00	55.2	16.98	3033.69	558.167	400.254	10.0697	1.74	250.203	252.136	...	248.92

5 rows × 24 columns

In [5]:
```
#Notice the date column is a string
print(type(df['date'][0]))
```

```
<class 'str'>
```

In [6]:
```
#Change date column to timestamp
df['date'] = pd.to_datetime(df['date'])
print(type(df['date'][0]))
```

```
<class 'pandas._libs.tslibs.timestamps.Timestamp'>
```

Descriptive Analytics

My manager has tasked me with providing summary statistics for each column. Specifically, they've asked for the average and median, along with the minimum and maximum values, for every individual column.

In [7]:
```
#Finding  the average & median, as well as the min & max for every column
df.describe()
```

Out[7]:

	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	
count	737453.000000	737453.000000	737453.000000	737453.000000	737453.000000	737453.000000	73
mean	56.294739	14.651716	2869.140569	488.144697	397.578372	9.767639	
std	5.157744	6.807439	1215.203734	91.230534	9.699785	0.387007	
min	42.740000	1.310000	0.002026	241.669000	376.249000	8.753340	
25%	52.670000	8.940000	2076.320000	431.796000	394.264000	9.527360	
50%	56.080000	13.850000	3018.430000	504.393000	399.249000	9.798100	
75%	59.720000	19.600000	3727.730000	553.257000	402.968000	10.038000	
max	65.780000	33.400000	6300.230000	739.538000	418.641000	10.808100	

8 rows × 23 columns

Investigating June 1

The most critical variable is the "% Iron Concentrate." However, our engineering colleague highlights the significance of the "% Silica Concentrate," "Ore Pulp pH," and "Flotation Column 05 Level" as well.

Notably, the date is also of importance. Our supervisor has brought to our attention an anomaly on June 1, 2017, and has requested an investigation into this unusual occurrence.

Let's look into the date to find out!

In [8]:
```
#defining the max and min data
max_date = df['date'].max()
print('The max date is ' + str(max_date))
min_date = df['date'].min()
print('The min date is ' + str(min_date))
```

```
The max date is 2017-09-09 23:00:00
The min date is 2017-03-10 01:00:00
```

In [9]:
```
#creating a dataframe specifically for June
df_june = df[(df['date'] > "2017-05-31 23:59:59") & (df['date'] < "2017-06-02")].reset
```

In [10]:
```
#Defining the important columns

important_cols = [
    'date',
    '% Iron Concentrate',
    '% Silica Concentrate',
    'Ore Pulp pH',
    'Flotation Column 05 Level'
]
```

In [11]:
```
#Creating another dataframe with the specific important columns
df_june_important = df_june[important_cols]
df_june_important
```

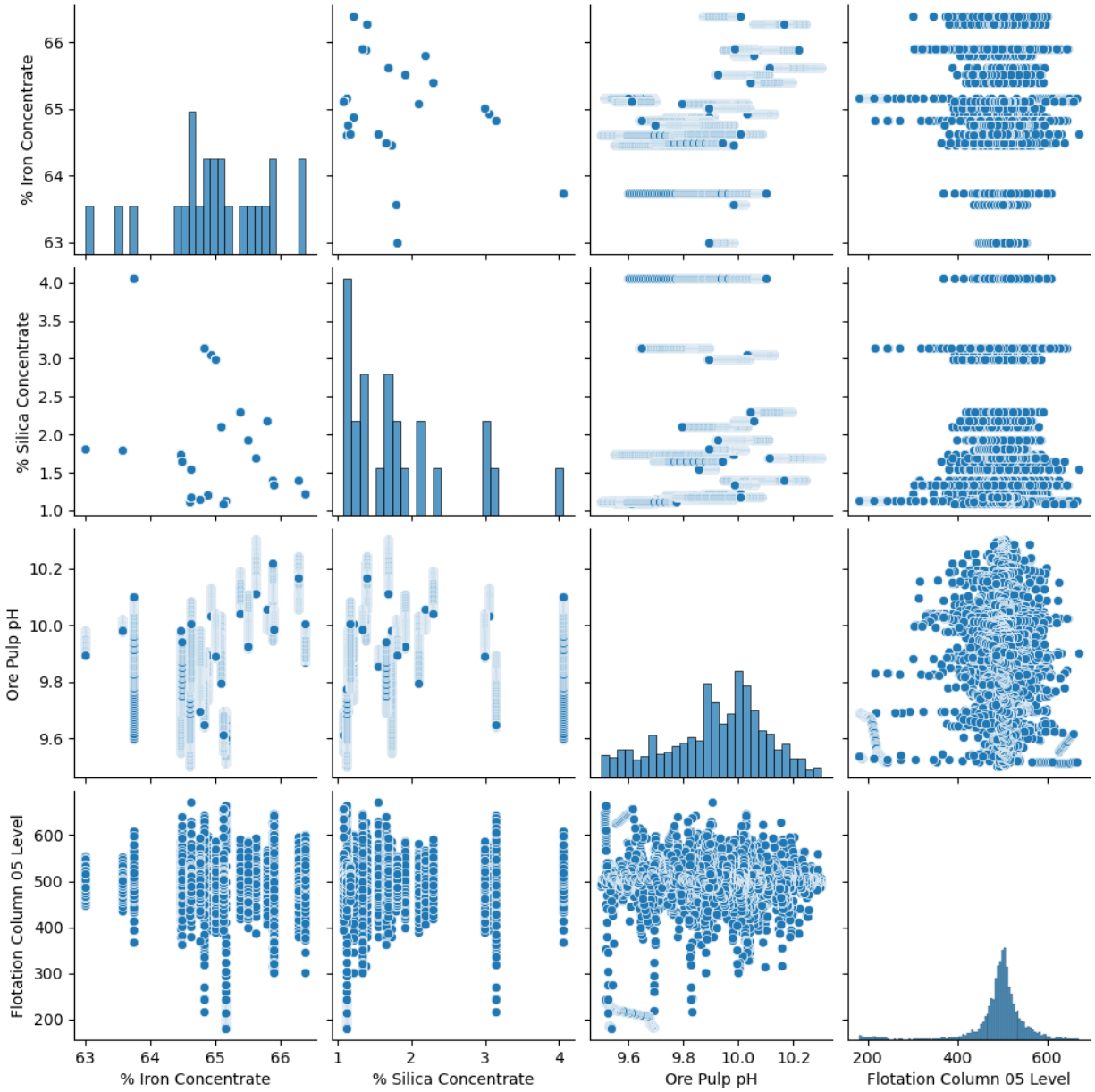
Out[11]:

	date	% Iron Concentrate	% Silica Concentrate	Ore Pulp pH	Flotation Column 05 Level
0	2017-06-01 00:00:00	64.46	1.73	9.55029	522.585
1	2017-06-01 00:00:00	64.46	1.73	9.55201	497.841
2	2017-06-01 00:00:00	64.46	1.73	9.55373	473.097
3	2017-06-01 00:00:00	64.46	1.73	9.55544	455.752
4	2017-06-01 00:00:00	64.46	1.73	9.55716	475.140
...	...	...	...	...	...
4315	2017-06-01 23:00:00	63.00	1.81	9.90185	519.440
4316	2017-06-01 23:00:00	63.00	1.81	9.89964	524.484
4317	2017-06-01 23:00:00	63.00	1.81	9.89743	524.173
4318	2017-06-01 23:00:00	63.00	1.81	9.89521	515.848
4319	2017-06-01 23:00:00	63.00	1.81	9.89300	486.637

4320 rows × 5 columns

In [12]:
```
#Visualize it
sns.pairplot(df_june_important)
```

Out[12]:
```
<seaborn.axisgrid.PairGrid at 0x2b10742caf0>
```



In [13]:
```
df_june_important.corr()
```

Out[13]:

	% Iron Concentrate	% Silica Concentrate	Ore Pulp pH	Flotation Column 05 Level
% Iron Concentrate	1.000000	-0.271731	0.302994	-0.045116
% Silica Concentrate	-0.271731	1.000000	0.191370	0.118911
Ore Pulp pH	0.302994	0.191370	1.000000	0.201715
Flotation Column 05 Level	-0.045116	0.118911	0.201715	1.000000

Results for June 1:

While the correlations are not highly significant, they do offer valuable insights into potential relationships among the variables. The negative correlation between "% Iron Concentrate" and "% Silica Concentrate" aligns with expectations, indicating an inverse relationship. Additionally,

weak correlations between "% Iron Concentrate" and other variables, such as "Ore Pulp pH" and "Flotation Column 05 Level," suggest possible influences but not strong linear associations. Similarly, the correlations involving "% Silica Concentrate," "Ore Pulp pH," and "Flotation Column 05 Level" provide insights into potential connections, although they are not very strong. Overall, these findings imply complex interactions within the iron ore concentration process that merit further exploration.

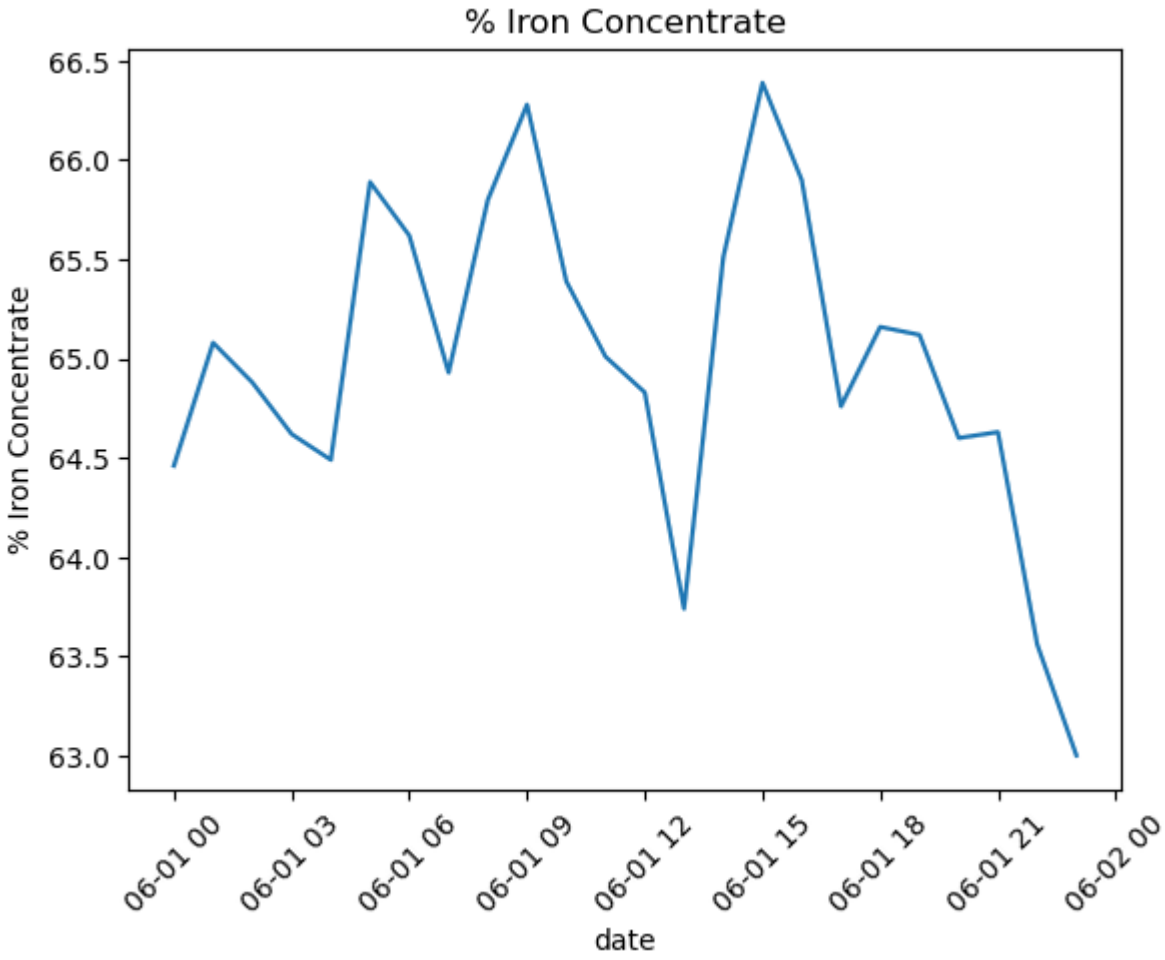
Concentration throughout the day

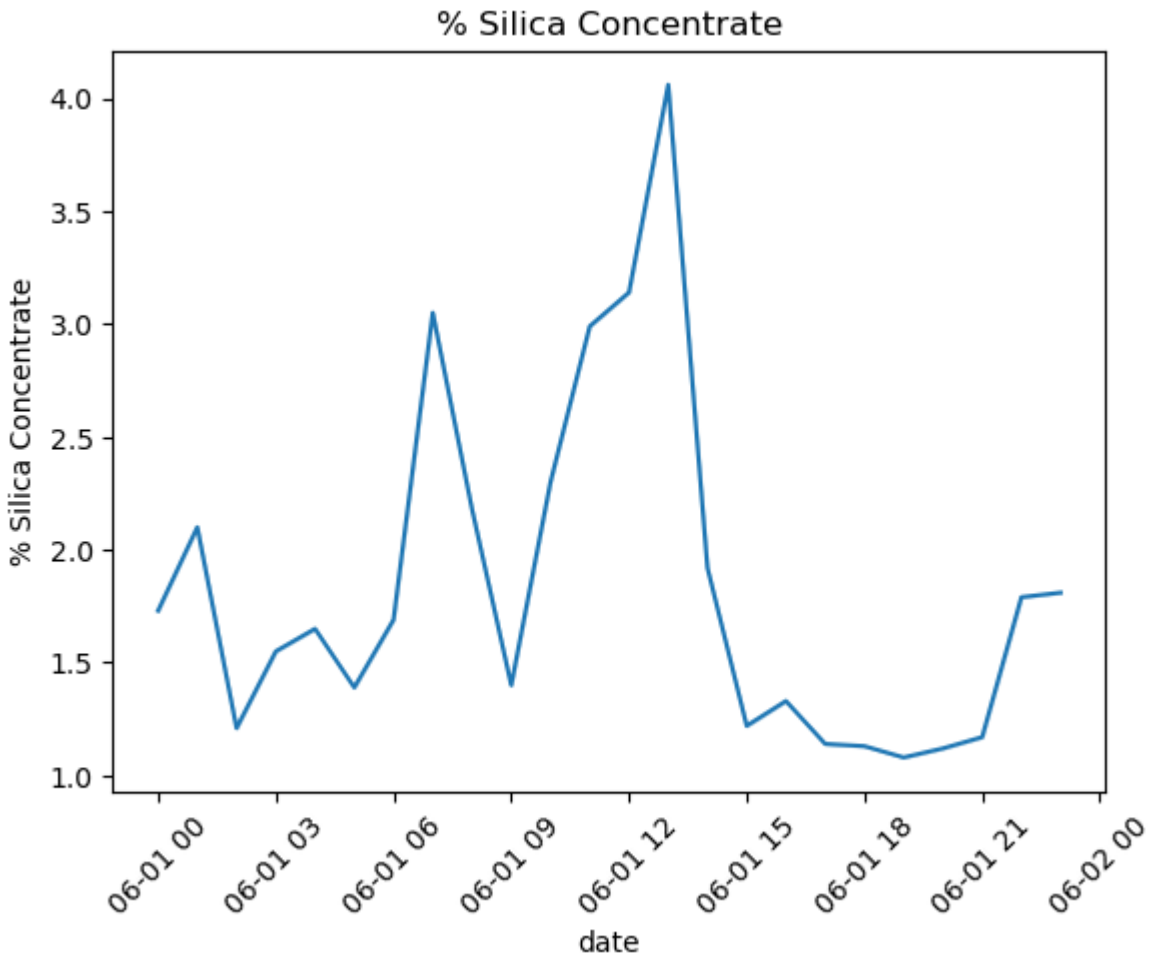
My boss is looking for insights and wants to examine the data to better understand the situation.

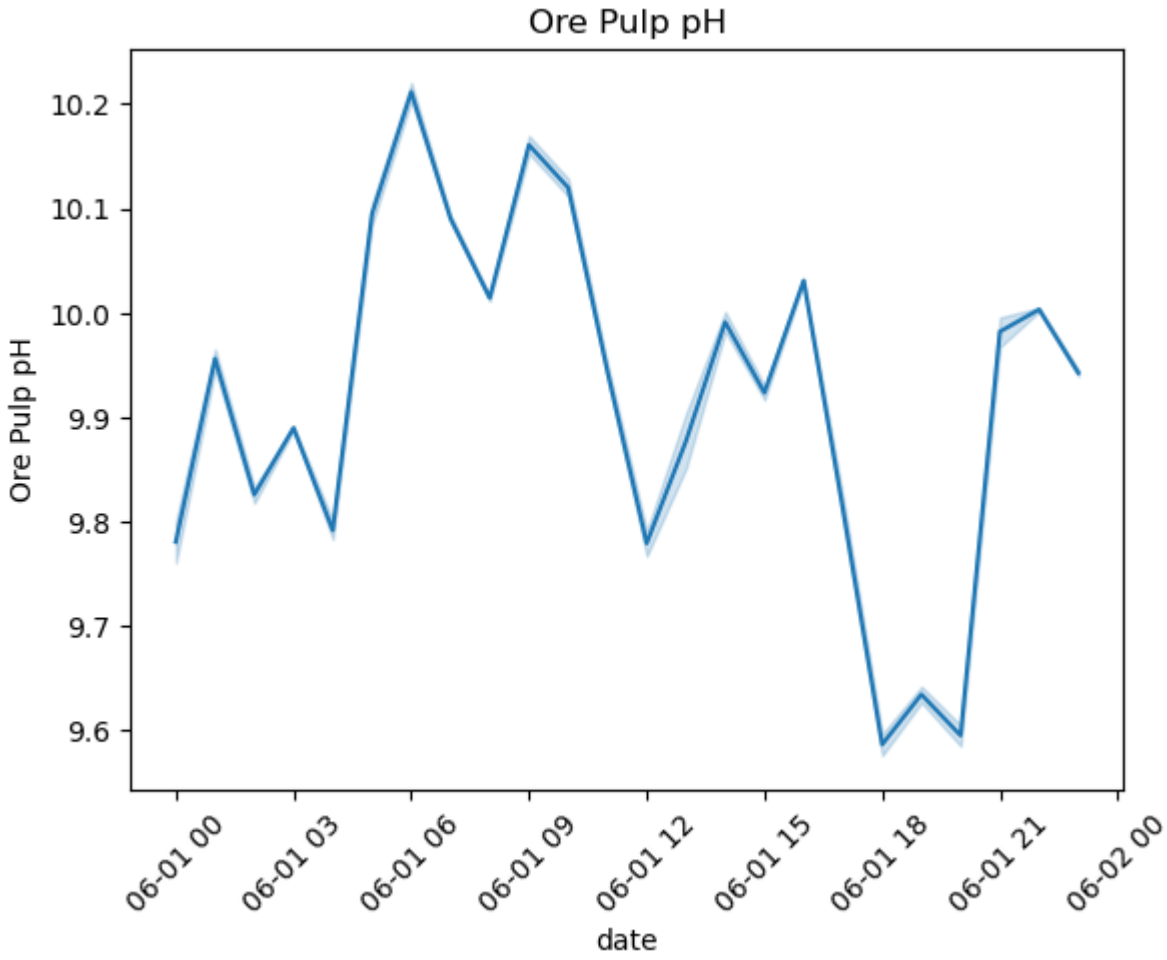
What's the plan? They are particularly interested in visualizing how the "% Iron Concentrate" changes throughout the day. However, they would like to see all the columns as well.

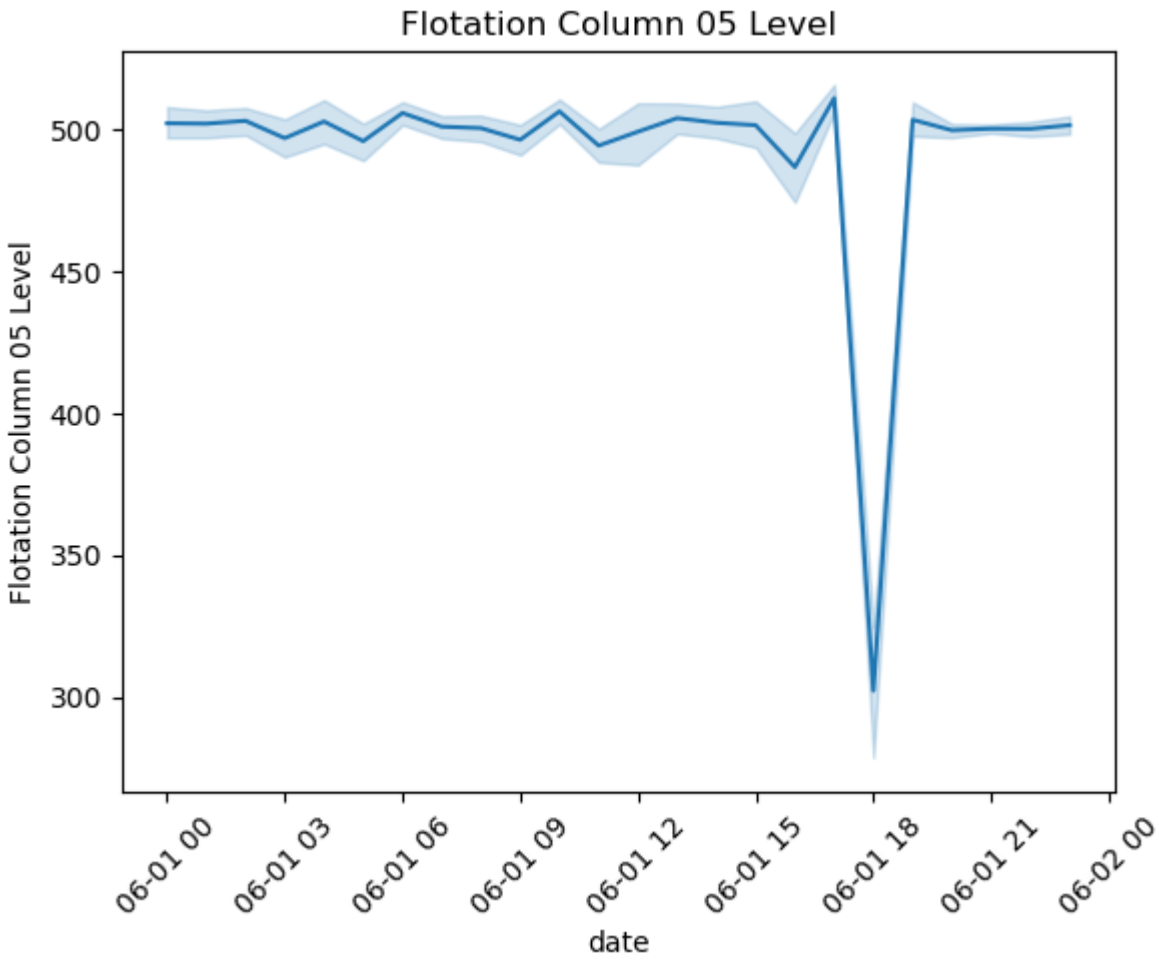
So, creating a graph would be helpful for visualizing the results.

In [14]:
```
import matplotlib.pyplot as plt

for i in important_cols[1:]:
    sns.lineplot(x='date', y=i, data=df_june)
    plt.xticks(rotation=45)  # Adjust the rotation angle as needed
    plt.title(i)
    plt.show()
```


% Iron Concentrate
% Iron Concentrate
66.5
66.0
65.5
65.0
64.5
64.0
63.5
63.0
06-01 00
06-01 03
06-01 06
06-01 09
06-01 12
06-01 15
06-01 18
06-01 21
06-02 00
date

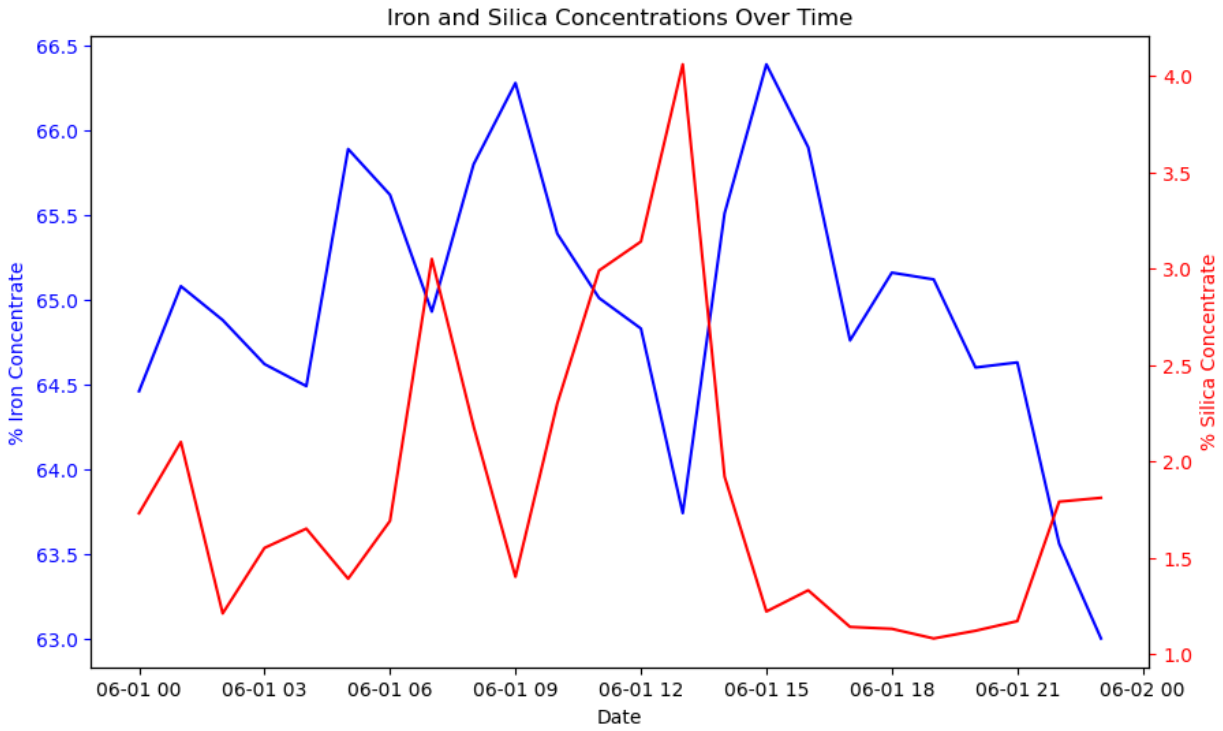

% Silica Concentrate
% Silica Concentrate
4.0
3.5
3.0
2.5
2.0
1.5
1.0
06-01 00
06-01 03
06-01 06
06-01 09
06-01 12
06-01 15
06-01 18
06-01 21
06-02 00
date


Ore Pulp pH
Ore Pulp pH
10.2
10.1
10.0
9.9
9.8
9.7
9.6
06-01 00
06-01 03
06-01 06
06-01 09
06-01 12
06-01 15
06-01 18
06-01 21
06-02 00
date



I noticed that there is a dip in concentrations for iron, while there is a rise for silica. So I wanted to plot it on the same axis to get a better view

In [15]:
```
#Iron and Silica Concentration Over Time

fig, ax1 = plt.subplots(figsize=(10, 6))

ax1.plot(df_june['date'], df_june['% Iron Concentrate'], 'b-', label='% Iron Concentra
ax1.set_xlabel('Date')
ax1.set_ylabel('% Iron Concentrate', color='b')
ax1.tick_params('y', colors='b')

ax2 = ax1.twinx()
ax2.plot(df_june['date'], df_june['% Silica Concentrate'], 'r-', label='% Silica Conce
ax2.set_ylabel('% Silica Concentrate', color='r')
ax2.tick_params('y', colors='r')

plt.title('Iron and Silica Concentrations Over Time')
plt.xticks(rotation=45)
plt.show()
```



Observation

The concurrent occurrence of a drop in iron concentration and an increase in silica concentration suggests potential challenges within the flotation plant's separation process. This phenomenon could stem from issues related to flotation efficiency, chemical balance, equipment performance, or ore feed variability. Identifying the root cause requires a comprehensive analysis of process conditions, operational decisions, and data accuracy. Addressing these issues is crucial to maintaining the desired product quality and optimizing the overall flotation plant's performance.

In [16]:
```
# Calculate the correlation between 'Iron' and 'Silica' columns
correlation = df['% Iron Concentrate'].corr(df['% Silica Concentrate'])

print("Correlation between Iron and Silica:", correlation)
```

```
Correlation between Iron and Silica: -0.800560002864754
```

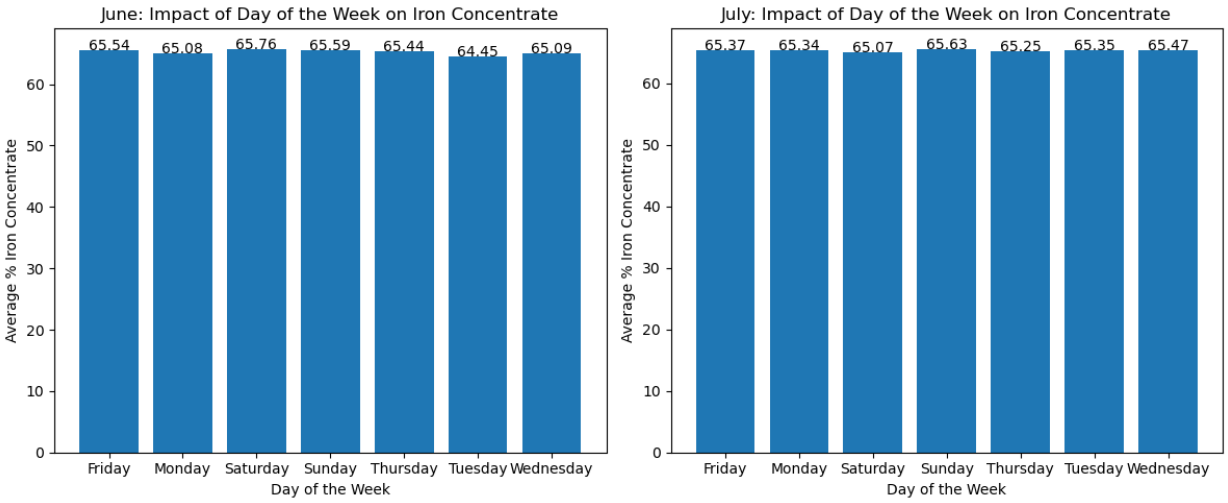
Observation (cont)

The correlation coefficient of -0.8005 between the concentrations of iron and silica reveals a notable pattern: as the iron concentration increases, the silica concentration tends to decrease, and vice versa. This negative correlation aligns with our expectations, as higher iron content typically corresponds to lower silica content in the ore concentrate. This relationship is integral to the mineral processing process, where the extraction and refinement of iron ore often involve the reduction of impurities like silica. Therefore, observing this inverse correlation reinforces the fundamental understanding of how these variables interact and impact the quality of the final iron product.

Comparing the Iron Concentration Per Month of June and July

My boss was interested in comparing the average iron concentration for the months of June and July, focusing on the distribution across different days of the week. To address this, I analyzed the dataset and extracted the iron concentration data for these two months. By calculating the average iron concentration for each day of the week, I was able to provide a clearer understanding of how iron concentration varies throughout the week. This comparison helps highlight any potential trends or patterns in iron concentration between the two months and sheds light on whether any significant differences exist in iron processing during these periods.

In [17]:
```
# Convert 'date' column to datetime
df['date'] = pd.to_datetime(df['date'])

# Extract month and day of the week
df['month'] = df['date'].dt.month
df['day_of_week'] = df['date'].dt.day_name()

# Select two different months for comparison
month1_data = df[df['month'] == 6]  # June
month2_data = df[df['month'] == 7]  # July

# Group by day of the week and calculate mean of '% Iron Concentrate' for each month
iron_by_day_month1 = month1_data.groupby('day_of_week')['% Iron Concentrate'].mean()
iron_by_day_month2 = month2_data.groupby('day_of_week')['% Iron Concentrate'].mean()

# Create subplots for comparison
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

# Bar plot for the first month
ax1.bar(iron_by_day_month1.index, iron_by_day_month1.values)
ax1.set_xlabel('Day of the Week')
ax1.set_ylabel('Average % Iron Concentrate')
ax1.set_title('June: Impact of Day of the Week on Iron Concentrate')

# Annotate each bar with its value for the first month
for i, v in enumerate(iron_by_day_month1):
    ax1.text(i, v + 0.01, f'{v:.2f}', ha='center')

# Bar plot for the second month
ax2.bar(iron_by_day_month2.index, iron_by_day_month2.values)
ax2.set_xlabel('Day of the Week')
ax2.set_ylabel('Average % Iron Concentrate')
ax2.set_title('July: Impact of Day of the Week on Iron Concentrate')

# Annotate each bar with its value for the second month
for i, v in enumerate(iron_by_day_month2):
    ax2.text(i, v + 0.01, f'{v:.2f}', ha='center')

plt.tight_layout()
plt.show()
```



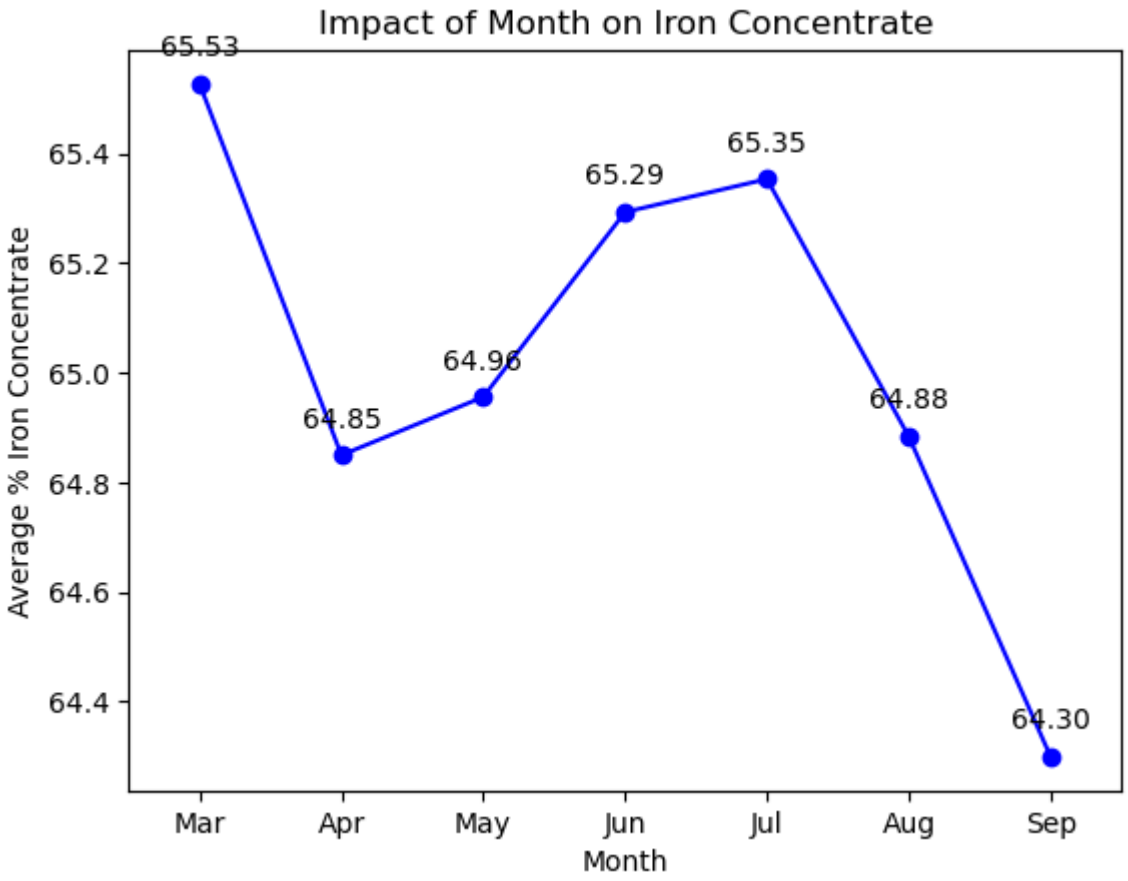
Upon analyzing the data, a notable observation emerges regarding the average iron concentration for the months of June and July, specifically on Tuesdays. In the month of June, the average iron concentration on Tuesdays was 64.45, which was the lowest among all the days of the week for that month. This suggests a potential fluctuation or variation in iron concentration on Tuesdays in June. On the other hand, in July, the average iron concentration on Tuesdays was 64.35, also the lowest among the days of the week for that month. Interestingly, the fluctuation in iron concentration on Tuesdays seems to persist in both months, although the overall range of fluctuation appeared to be more pronounced in June compared to July. This indicates that the iron concentration experienced more variability on Tuesdays in June, whereas the fluctuations were relatively smaller on Tuesdays in July. The rest of the days also show variations, with slight differences in average iron concentration between the two months.

Any Monthly patterns in Iron Concentration?

In response to my boss's request for deeper insights into the iron concentration levels, a comprehensive analysis was conducted to explore potential trends across different months. The objective was to identify any patterns or noticeable changes in the average iron concentration.

In [18]:
```
# Group by month and calculate mean of '% Iron Concentrate'
iron_by_month = df.groupby('month')['% Iron Concentrate'].mean()

# Create a line plot
iron_by_month.plot(kind='line', marker='o', color='blue')

# Annotate data points
for month, iron_value in iron_by_month.iteritems():
    plt.annotate(f'{iron_value:.2f}', (month, iron_value), textcoords="offset points",

plt.xlabel('Month')
plt.ylabel('Average % Iron Concentrate')
plt.title('Impact of Month on Iron Concentrate')
plt.xticks(range(3, 10), ['Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep'])
```

```
plt.xlim(2.5, 9.5)  # Set x-axis limits
plt.show()
```



Upon analyzing the data regarding iron concentration across different months, it becomes evident that there is no discernible consistent pattern in the levels. Instead, the iron concentration levels exhibit fluctuations that do not follow a specific trend. This observation suggests that various factors, including operational conditions and external influences, might contribute to the variability in iron concentration. Consequently, it's challenging to identify a clear trend or pattern in the data, and the levels appear to fluctuate without a consistent direction or correlation with the passage of time. This investigation underlines the complexity of the factors influencing iron concentration, as the absence of an identifiable pattern suggests that various dynamic variables contribute to the observed fluctuations. While the absence of distinct trends challenges the straightforward identification of correlations, this analysis provides valuable insights into the intricate nature of iron concentration levels and the potential factors influencing them.

Key Findings Recap:

1. **Correlation Complexity:** The dataset's correlation matrix revealed moderate relationships between key variables, such as "% Iron Concentrate," "% Silica Concentrate," "Ore Pulp pH," and "Flotation Column 05 Level." While not strongly linear, these correlations provided a foundation for further investigation into potential associations.

2. **Concentration Dynamics:** Fluctuations in iron and silica concentrations throughout the day suggested challenges in the separation process. Identifying the causes of these fluctuations, whether from operational decisions or equipment performance, is pivotal for maintaining product quality and plant efficiency.

3. **Iron-Silica Relationship:** The inverse correlation between iron and silica concentrations, with a coefficient of -0.8005, aligned with expectations. This understanding reinforced the fundamental connection between iron's rise and silica's fall, essential for effective mineral processing.

4. **Monthly Insights (June and July):** A comparative analysis of iron concentrations on Tuesdays in June and July revealed consistent dips. While fluctuations persisted in both months, June showcased more pronounced variations, hinting at operational variations.

5. **Month-to-Month Variation:** Iron concentration across different months exhibited fluctuations without a clear pattern. This complexity underscored the role of multiple dynamic factors influencing iron concentration levels.

Recommendations

1. **Operational Optimization:** Investigate the reasons behind fluctuations in iron and silica concentrations throughout the day. Analyze operational decisions, equipment performance, and process conditions during these periods to improve process efficiency and product quality.

2. **Root Cause Analysis:** Address the variations in iron concentration observed on Tuesdays in June and July. Collaborate with experts to identify whether these variations stem from external factors, equipment issues, or other influences. This could lead to stabilizing iron concentration levels and enhancing process consistency.

3. **Collaboration and Knowledge Sharing:** Engage with domain experts, metallurgists, and engineers to gain deeper insights into the mining and mineral processing processes. Their expertise can provide valuable context and guide the analysis toward more actionable insights, ultimately contributing to process optimization.