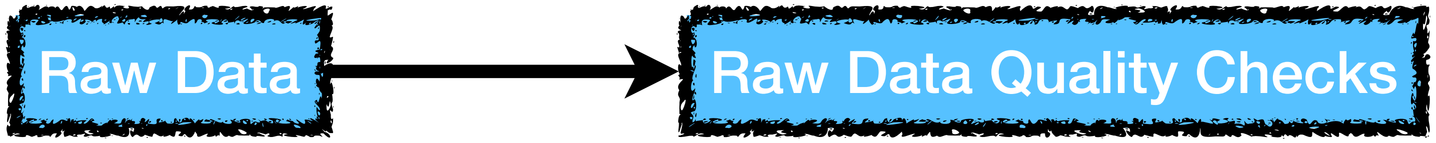
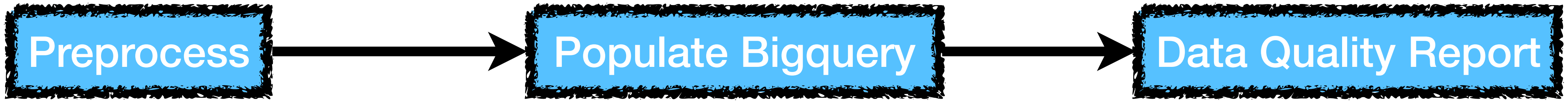
Overview of Workflow & Setup

Stage 1: Raw Data Quality Checks



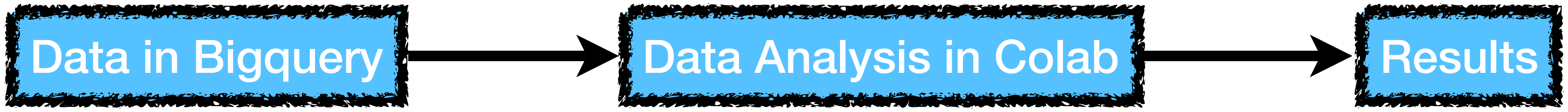
Raw Data → Raw Data Quality Checks

Stage 2: Prepare & Organise Data



Preprocess → Populate Bigquery → Data Quality Report

Stage 3: Analysis



Data in Bigquery → Data Analysis in Colab → Results

Big-query scripts

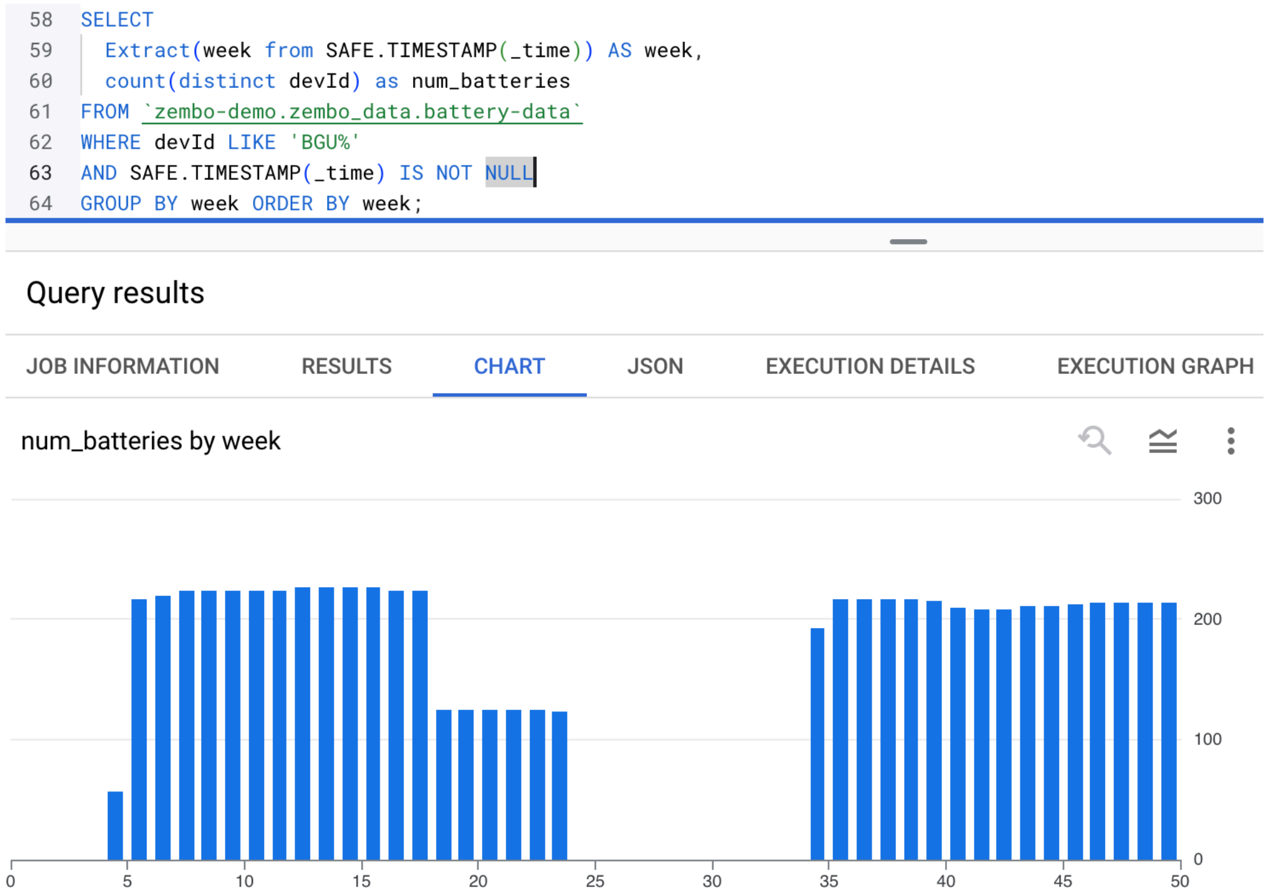
- Data Quality Checks
- Data Validity
- Merge Swap Data

Colab Notebooks

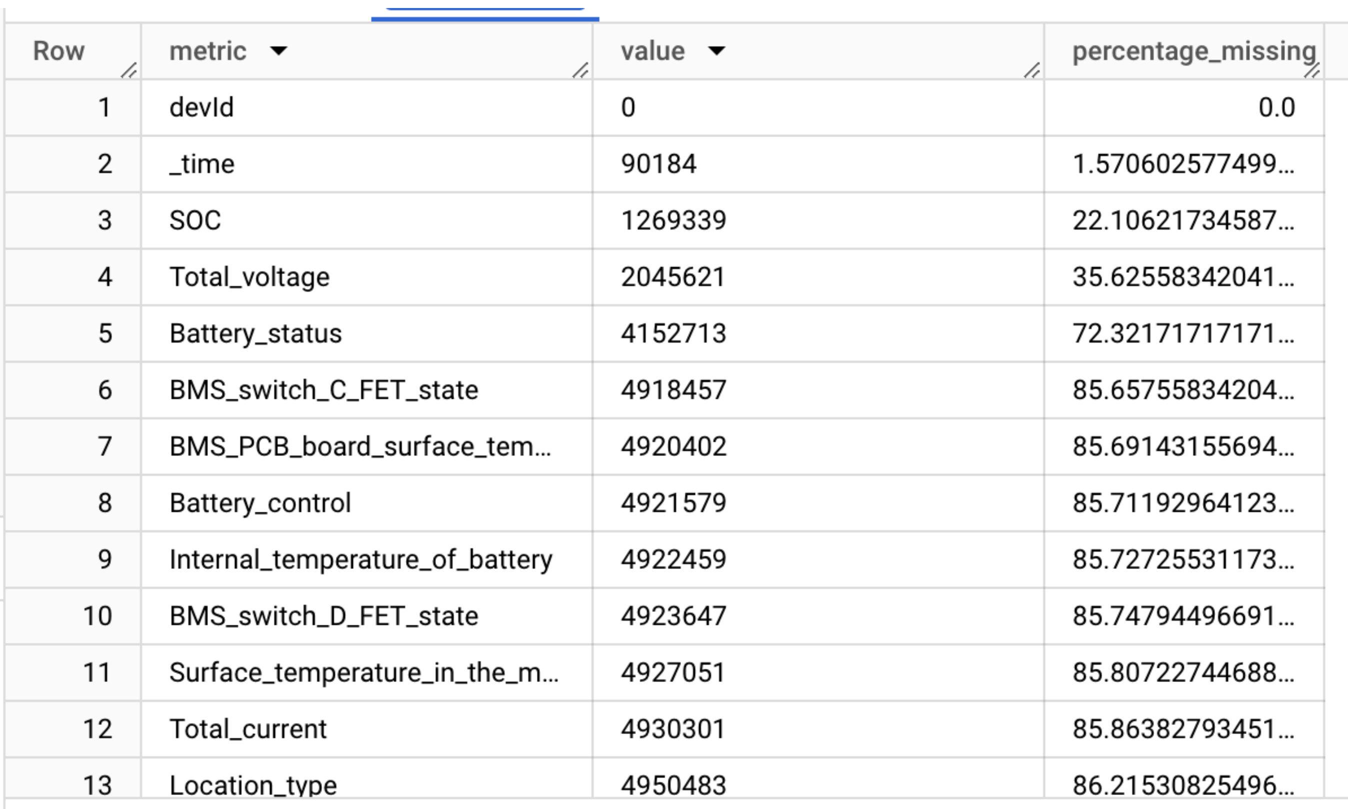
- Populate Bigquery
- Analysis
- Predictive modelling

Data Quality

Completness



```
SELECT
  Extract(week from SAFE.TIMESTAMP(_time)) AS week,
  count(distinct devId) as num_batteries
FROM `zembo-demo.zembo_data.battery-data`
WHERE devId LIKE 'BGU%'
AND SAFE.TIMESTAMP(_time) IS NOT NULL
GROUP BY week ORDER BY week;
```



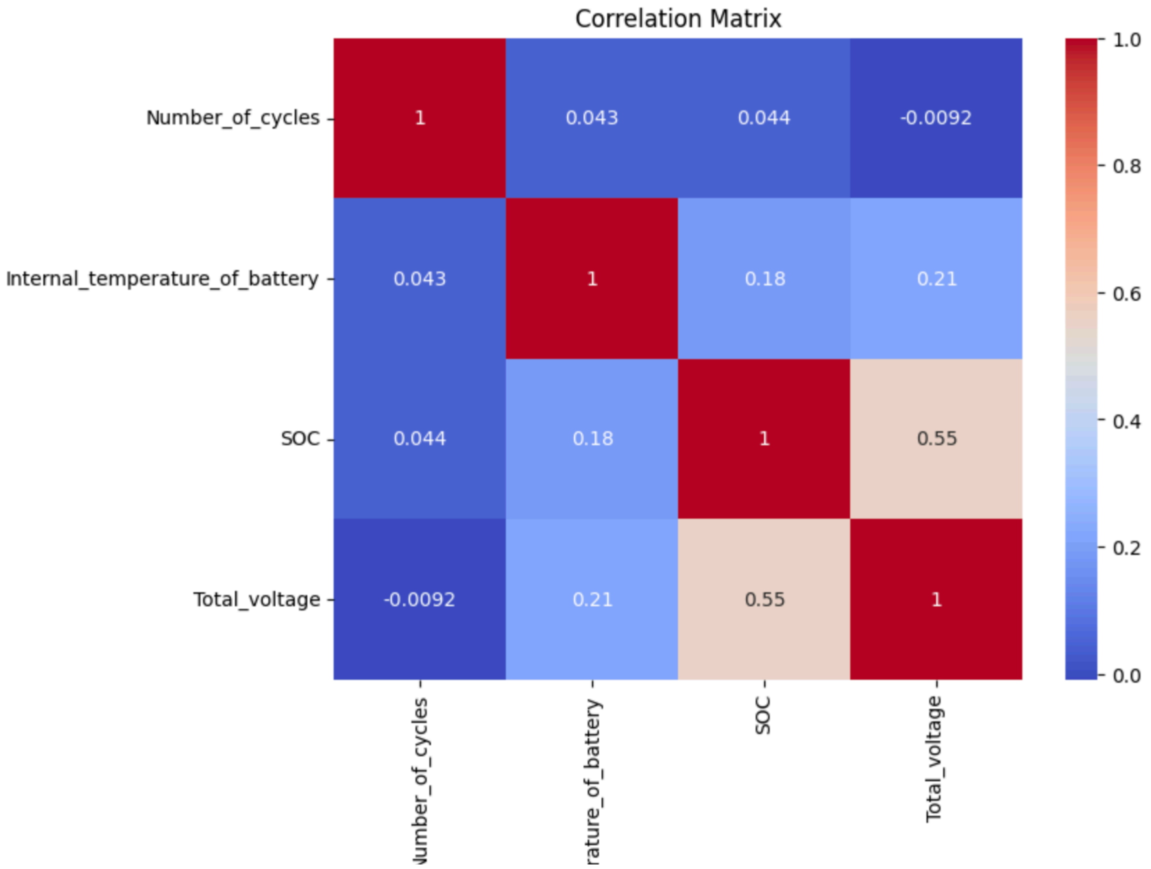
Row	metric	value	percentage_missing
1	devId	0	0.0
2	_time	90184	1.570602577499...
3	SOC	1269339	22.10621734587...
4	Total_voltage	2045621	35.62558342041...
5	Battery_status	4152713	72.32171717171...
6	BMS_switch_C_FET_state	4918457	85.65755834204...
7	BMS_PCB_board_surface_tem...	4920402	85.69143155694...
8	Battery_control	4921579	85.71192964123...
9	Internal_temperature_of_battery	4922459	85.72725531173...
10	BMS_switch_D_FET_state	4923647	85.74794496691...
11	Surface_temperature_in_the_m...	4927051	85.80722744688...
12	Total_current	4930301	85.86382793451...
13	Location_type	4950483	86.21530825496...

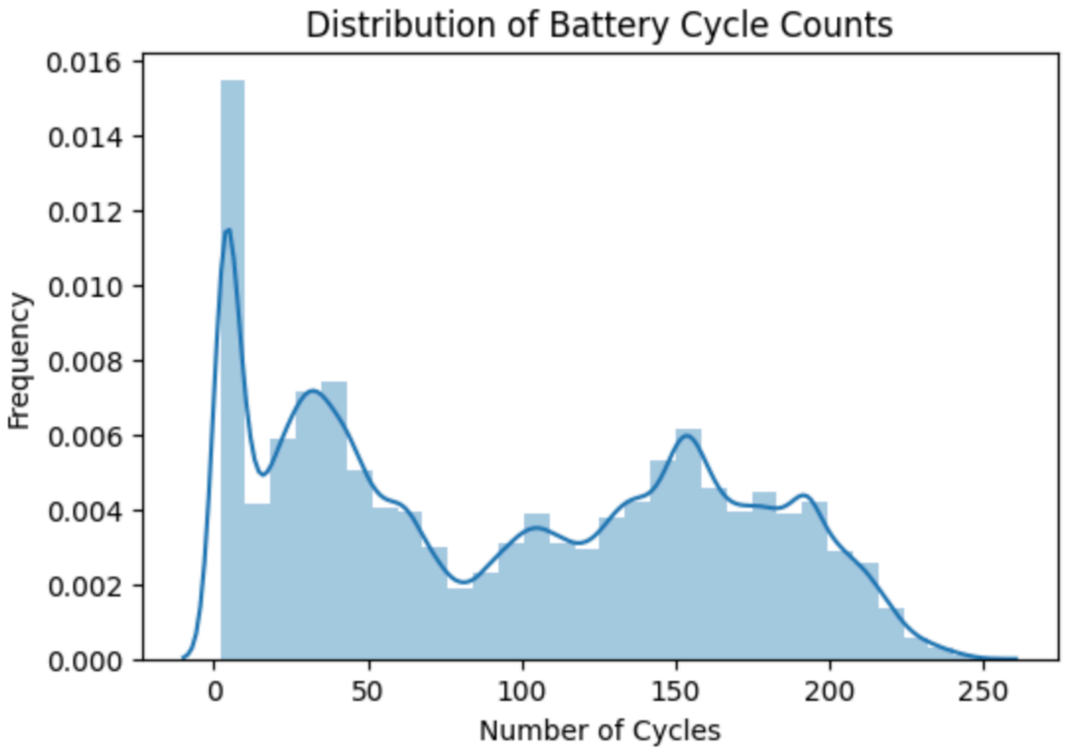
- Number of battery’s drops to zero around week 23
- Majority of data points missing
  - 53 out of 57 columns have more than 70% missing data

Consistency

- A lot of inconsistencies across different columns or tables (e.g., conflicting data, format variations)
  - Different number of columns (expected 57 columns)
  - Numeric fields contain strings

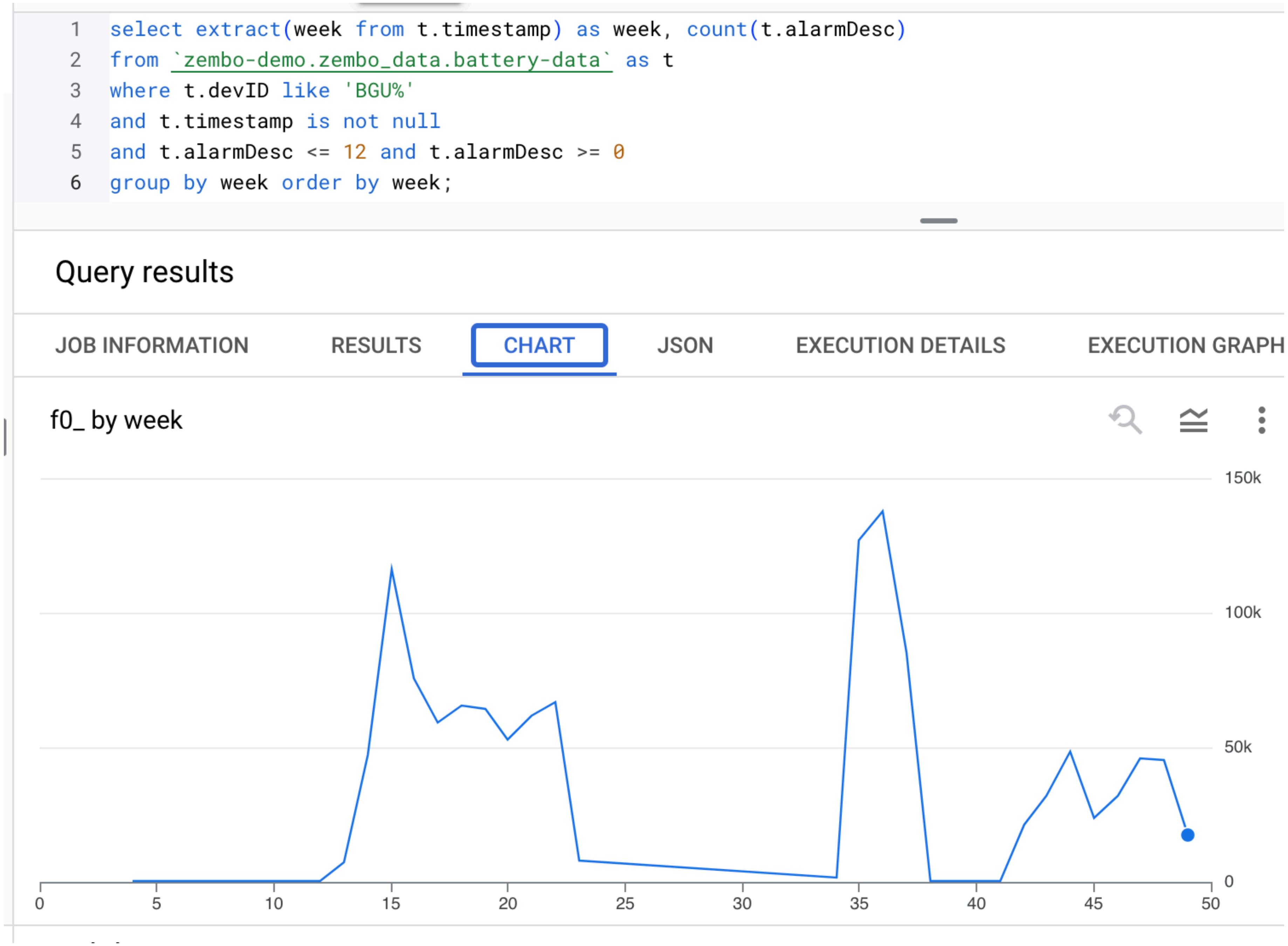
EDA





- No major correlation structure except for SOC-NOC
- We observe multiple modes at approximately [50, 100, 150, 200].
  - This implies that there are clusters of batteries that tend to have cycle counts concentrated around these values

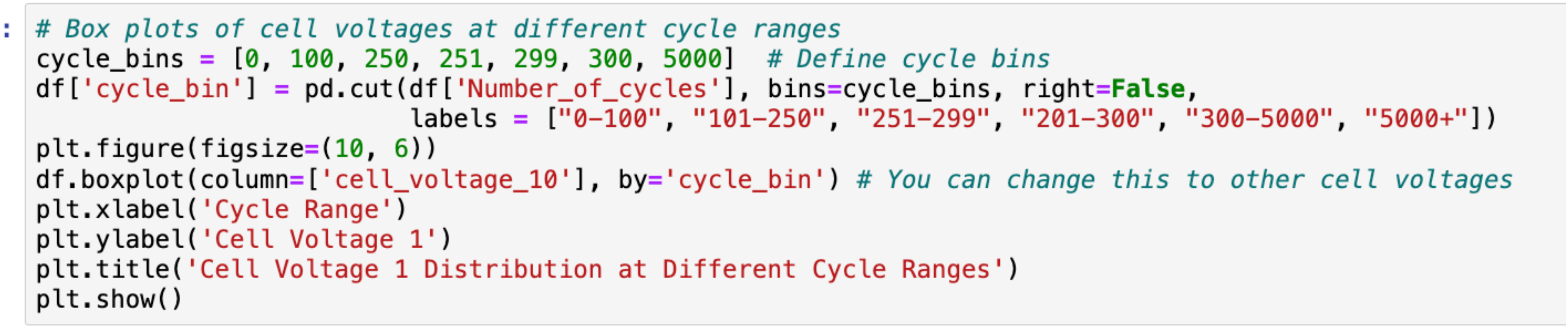
EDA



```
select extract(week from t.timestamp) as week, count(t.alarmDesc)
from `zembo-demo.zembo_data.battery-data` as t
where t.devID like 'BGU%'
and t.timestamp is not null
and t.alarmDesc <= 12 and t.alarmDesc >= 0
group by week order by week;
```

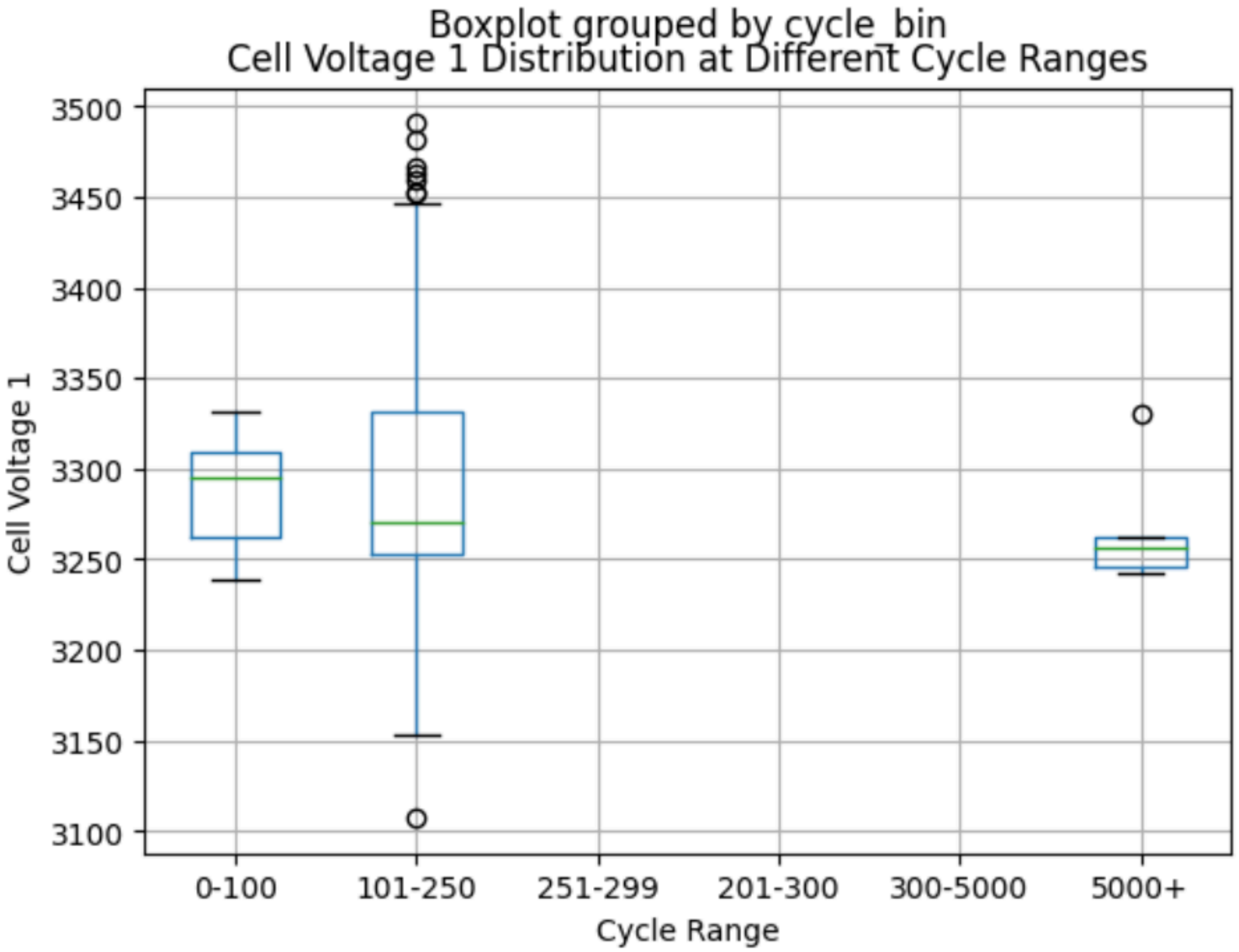
- Large number of alerts during certain weeks (226 batteries)
    - Further investigations into the source of a high number of alerts

EDA



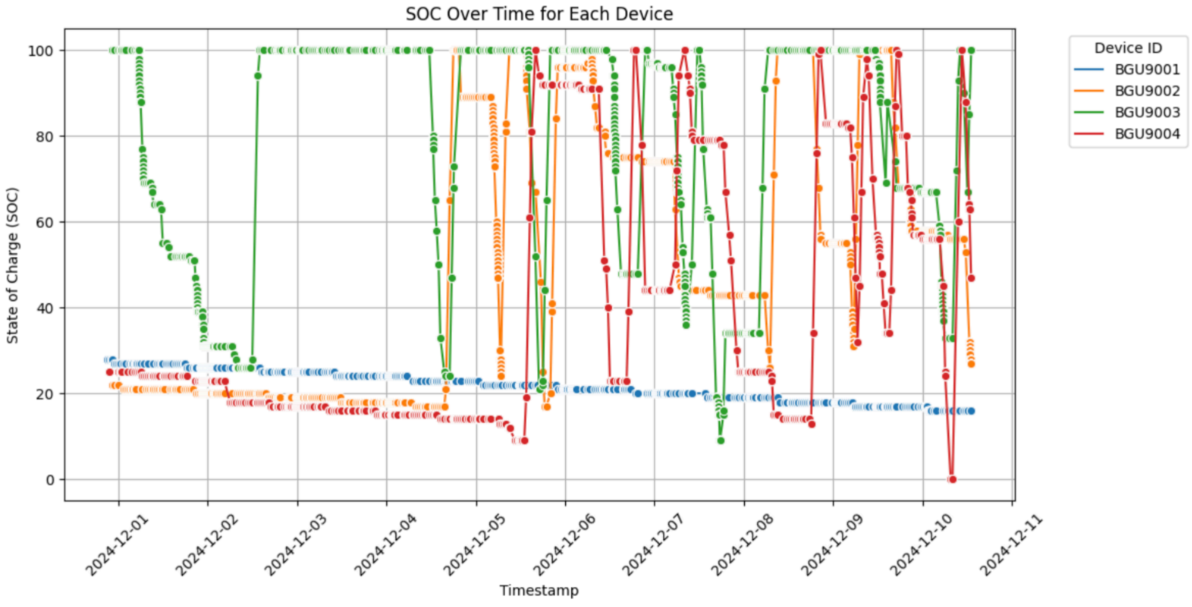
```
# Box plots of cell voltages at different cycle ranges
cycle_bins = [0, 100, 250, 251, 299, 300, 5000]  # Define cycle bins
df['cycle_bin'] = pd.cut(df['Number_of_cycles'], bins=cycle_bins, right=False,
                         labels = ["0-100", "101-250", "251-299", "201-300", "300-5000", "5000+"])
plt.figure(figsize=(10, 6))
df.boxplot(column=['cell_voltage_10'], by='cycle_bin') # You can change this to other cell voltages
plt.xlabel('Cycle Range')
plt.ylabel('Cell Voltage 1')
plt.title('Cell Voltage 1 Distribution at Different Cycle Ranges')
plt.show()
```

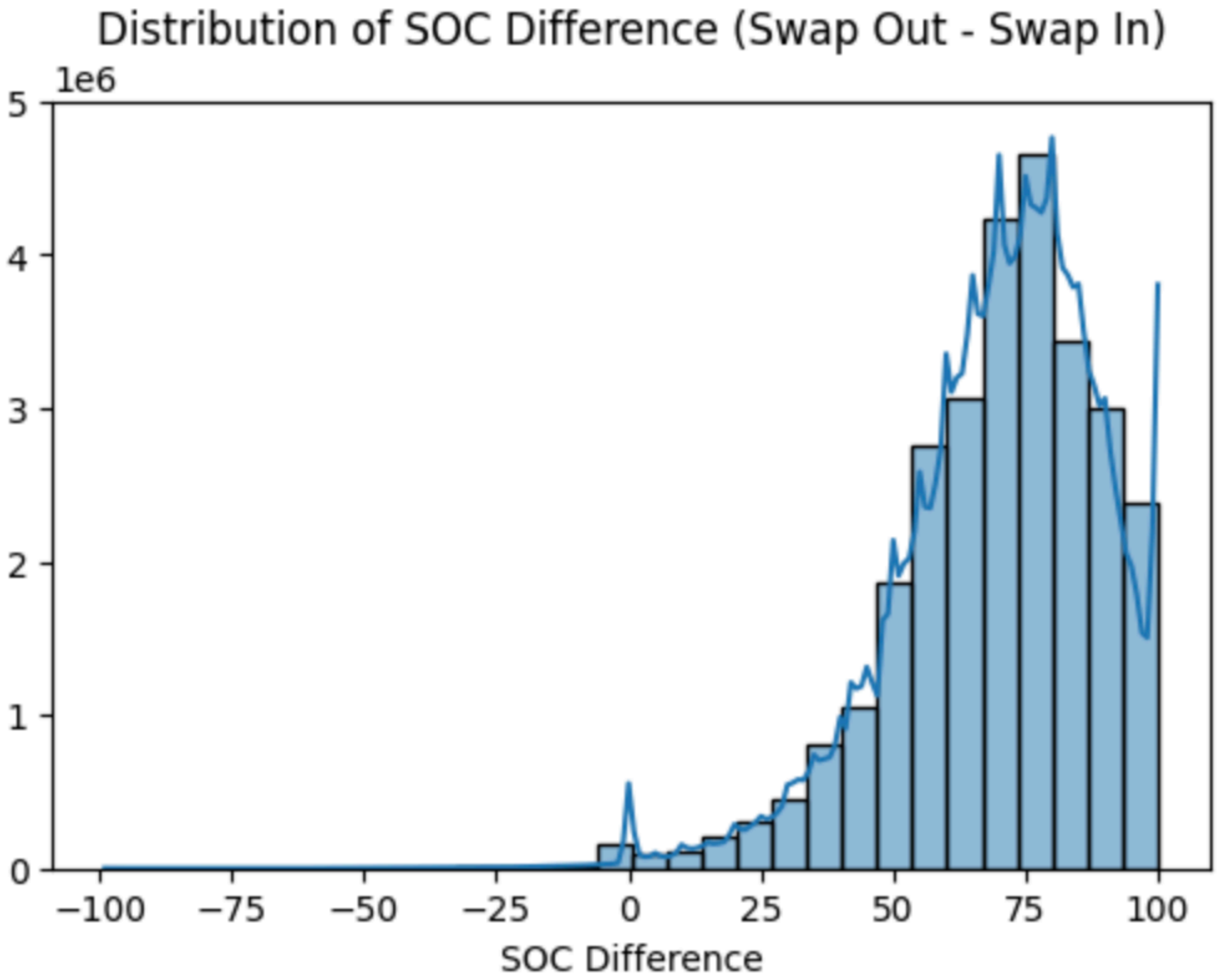
```
<Figure size 1000x600 with 0 Axes>
```



- Discrepancy in the cell voltage across cycles

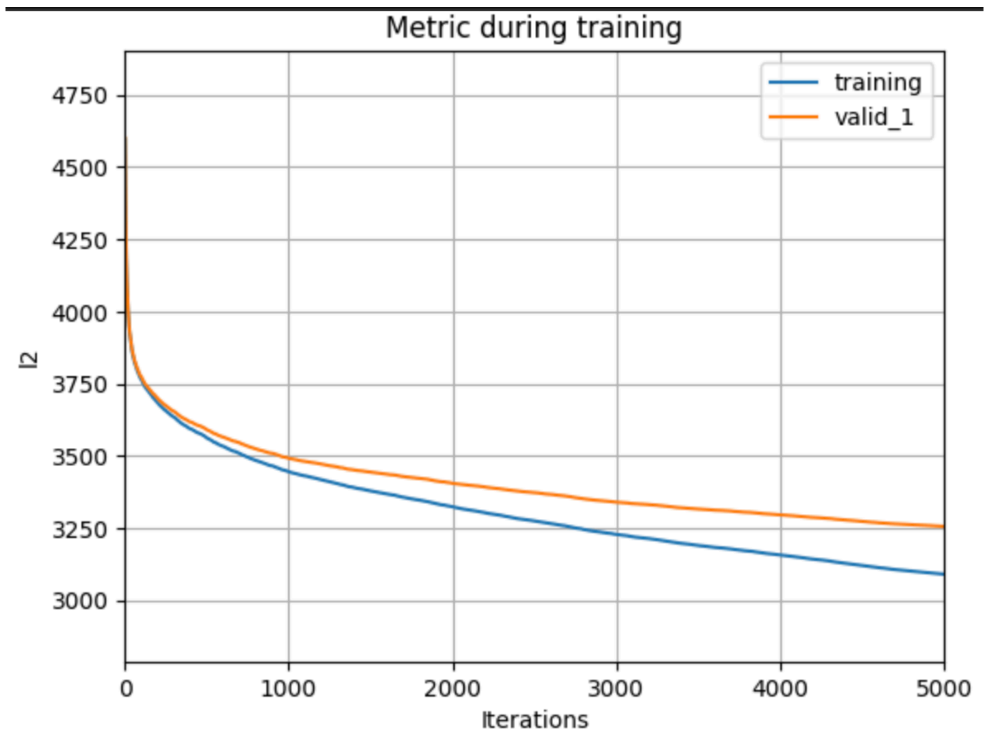
EDA

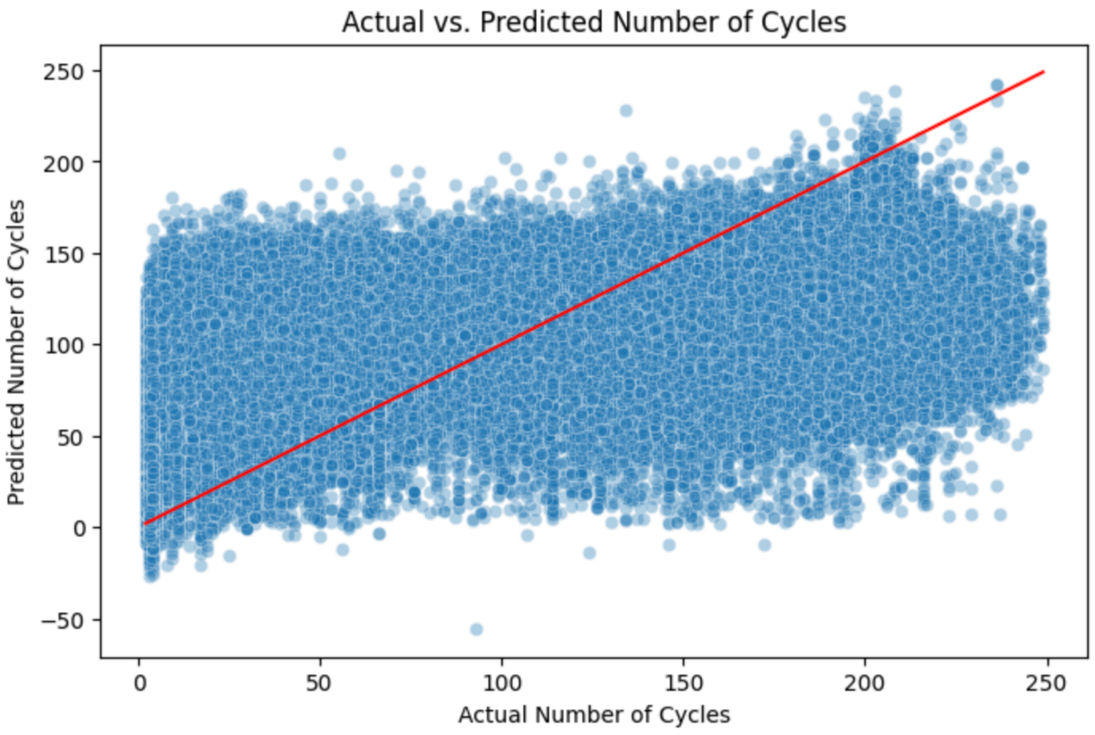


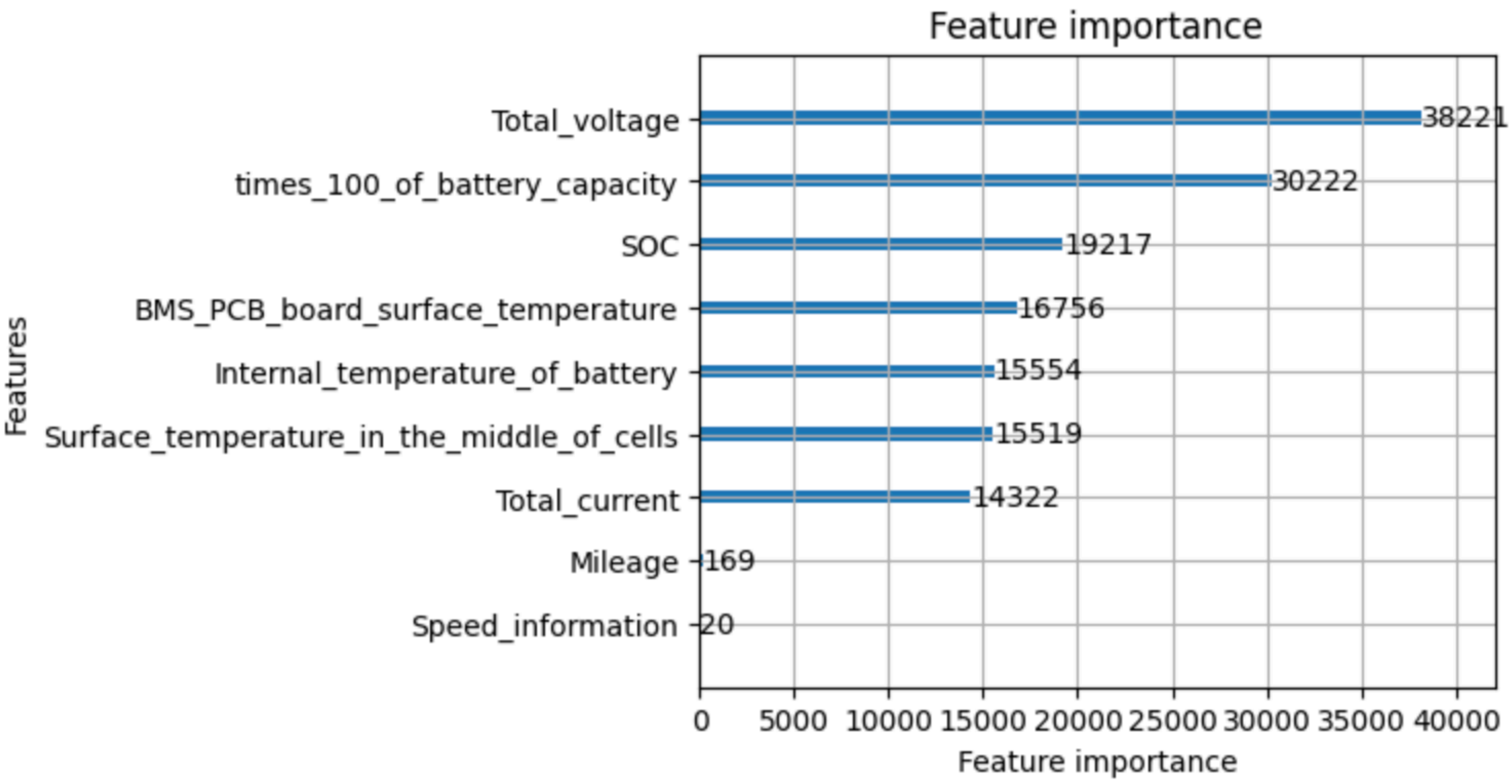


- A peak around 75 suggests that, on average, batteries tend to gain approximately 75% of charge during a swap cycle.

Predictive Modelling







To do

- Automate ingestion of new data to big query
    - Cleaning of data
- Investigate root causes of missing & invalid data
- Monitor alarm logs for frequent or critical alarms. Investigate root causes.
- Implement a predictive maintenance schedule based on cycle count and temperature data.