# Project: Analyzing lithium-ion Battery Performance Data for Predictive Maintenance

# ****Why?****

* Battery health is a growing concern in electronics, EVs, and industrial equipment.
* Predictive maintenance helps in reducing downtime and improving efficiency.
* **Not many people** have explored this in depth, making it a fresh and valuable project.

✔ Easy to get data from open sources  
✔ ML + Power BI integration possible  
✔ Industry-relevant project for your portfolio

## Data Availability

## ✅ ****Kaggle datasets on lithium-ion batteries****

This dataset contains data related to the charging characteristics of electric vehicle (EV) batteries. The dataset includes various factors affecting battery performance, degradation, and charging duration, along with the target variable representing the optimal charging duration classification. The data is organized with both numerical and categorical features and is designed for analysis or prediction tasks.

Features:  
SOC (%): State-of-Charge of the EV battery, expressed as a percentage. This indicates the current charge level of the battery.  
Voltage (V): Voltage of the battery during charging, measured in volts (V).  
Current (A): The current (in amperes) flowing into the battery during the charging process.  
Battery Temp (°C): The temperature of the battery during charging, measured in degrees Celsius (°C).  
Ambient Temp (°C): The surrounding temperature of the environment in which the EV is being charged.  
Charging Duration (min): The duration (in minutes) of the battery charging process.  
Degradation Rate (%): A computed value representing the degradation rate of the battery, which affects its overall efficiency and charging capacity. This is calculated based on the charging duration, SOC, and battery temperature.  
Charging Mode: A categorical feature representing the charging mode used during the process. This can be one of three values: 'Fast', 'Normal', or 'Slow'.  
Efficiency (%): Battery efficiency calculated based on the degradation rate. The higher the degradation, the lower the efficiency.  
Battery Type: The type of battery used in the EV. This feature has two possible values: 'Li-ion' (Lithium-ion) and 'LiFePO4' (Lithium Iron Phosphate).  
Charging Cycles: The number of charging cycles the battery has undergone. A charging cycle is counted when the battery goes from full charge to empty and back to full charge.  
EV Model: The model of the electric vehicle, which can be one of the following: 'Model A', 'Model B', or 'Model C'.  
Optimal Charging Duration Class: The target variable representing the classification of the optimal charging duration:  
Short (0): Charging duration is short (≤ 40 minutes).  
Medium (1): Charging duration is moderate (≤ 80 minutes).  
Long (2): Charging duration is long (> 80 minutes).

✅ ****University of Maryland Battery Data Set**** ([Nasa.gov](https://data.nasa.gov/" \t "_new))  
✅ ****Stanford University Battery Aging Data Set****

These datasets contain:

* **Voltage, Current, and Temperature** readings over time
* **Battery cycles** (from charging/discharging)
* **Degradation metrics**

## Feature Engineering

### ****1️⃣ Battery Power (W)****

**Formula:**

Battery Power (W)=Voltage (V)×Current (A)

EV\_data['Battery Power (W)'] = EV\_data['Voltage (V)'] \* EV\_data['Current (A)']

✅ **Why?**  
Battery power gives a more **meaningful measure of energy flow** than voltage or current alone.

### ****2️⃣ Charging Speed (C-rate)****

**Formula:**

Charging Speed = Charging Cycles / Current (A)​

EV\_data['Charging Speed'] = EV\_data['Current (A)'] / (EV\_data['Charging Cycles'] + 1)

✅ **Why?**

* C-rate helps understand how aggressively a battery is charged.
* Adding +1 to Charging Cycles prevents division by zero.

### ****3️⃣ Temperature Difference****

**Formula:**

Temp Diff = Battery Temp (°C) − Ambient Temp (°C)

EV\_data['Temp Diff'] = EV\_data['Battery Temp (°C)'] - EV\_data['Ambient Temp (°C)']

✅ **Why?**

* High Temp Diff may indicate **battery overheating**, which affects degradation.

### ****4️⃣ Energy Efficiency Ratio****

**Formula:**

Energy Efficiency = Efficiency (%) / Charging Duration (min)

EV\_data['Energy Efficiency Ratio'] = EV\_data['Efficiency (%)'] / (EV\_data['Charging Duration (min)'] + 1)

✅ **Why?**

* Helps **normalize efficiency across different charging durations**.

### ****Depth of Discharge (DoD)****

**Why?**

* DoD measures how much of the battery capacity is used before recharging.
* Lower DoD values indicate **less stress on the battery** and **slower degradation**.

**Formula:**

DoD=100−SOC(%)

EV\_data['Depth of Discharge (%)'] = 100 - EV\_data['OC (%)']

✅ **More accurate than just looking at OC (%).**

### ****6. State of Charge (SoC) Change Per Cycle****

**Why?**

* Measures how much charge is used per cycle, which helps **predict degradation trends**.
* Batteries that fluctuate widely in SoC degrade faster.

**Formula:**

SoC Change Per Cycle = {SOC (%) / {Charging Cycles} + 1}

EV\_data['SoC Change Per Cycle'] = EV\_data['OC (%)'] / (EV\_data['Charging Cycles'] + 1)

✅ **Gives a better understanding of how battery charge level fluctuates over time.**

### ****7. Charging Stress Factor****

**Why?**

* Measures how much stress is applied during charging.
* High temperatures and high currents cause more stress on battery cells.

**Formula:**

Charging Stress = Current (A)×Battery Temp (°C)

EV\_data['Charging Stress'] = EV\_data['Current (A)'] \* EV\_data['Battery Temp (°C)']

✅ **High charging stress leads to faster degradation.**

### ****8. Charging Rate (C-Rate)****

**Why?**

* Batteries charged at higher rates degrade faster.
* Helps determine whether a battery is charged aggressively or slowly.

**Formula:**

C-Rate=Battery Capacity (Ah) / Current (A)​

🔹 **Do you have battery capacity (**Ah**) data?** If not, we can approximate it using power and voltage trends.

### ****9. Temperature Stress Factor****

**Why?**

* If the battery temperature is significantly higher than ambient temperature, **thermal stress is increasing degradation**.

**Formula:**

Temperature Stress = Charging Duration (min) / Temp Diff​

EV\_data['Temperature Stress'] = EV\_data['Temp Diff'] / (EV\_data['Charging Duration (min)'] + 1)

✅ **Batteries exposed to frequent high temperatures degrade faster.**

**How to Define Battery Health?**

Battery health is typically measured as:  
1️⃣ **Remaining Capacity (%)** - How much of the original capacity is still available?  
2️⃣ **Degradation Rate (%)** - How quickly the battery is losing its capacity?  
3️⃣ **Efficiency (%)** - How well the battery charges and discharges?

Since you **already have the "Degradation Rate"** feature, it can be used as a **proxy for battery health**.

🔹 **Final Target Variable for Prediction:**  
👉 **Battery Health (%) = 100 - Degradation Rate (%)**

· **Depth of Discharge (DoD)** = 100 - SOC (%)

· **SoC Change Per Cycle** = SOC (%) / (Charging Cycles + 1)

· **Charging Stress** = Current (A) \* Battery Temp (°C)

· **Temperature Stress** = Temp Diff / Charging Duration (min)

## Scaling & Normalization

Since different features have different units, we normalize them to ensure they contribute equally to the model.

**Why?**

* Normalization ensures that all features are on the **same scale (0 to 1)**.
* Helps **avoid bias toward larger numerical values** (e.g., voltage vs. efficiency).

## Feature Selection

**Correlation Heatmap**

To remove highly correlated features:

**Feature Importance (Using Random Forest)**

To check which features contribute the most:

## Machine Learning Potential

* **Regression Models** to predict battery health (%)
* **Classification Models** for failure detection (Healthy vs. Degraded Battery)
* **Time Series Forecasting** for Remaining Useful Life (RUL) prediction
* **Anomaly Detection** using unsupervised learning

## Power BI Dashboard for Battery Health Monitoring

Create an **interactive dashboard** with:

**Battery Performance Metrics** (Capacity over time, Efficiency trends)  
**Predictive Charts** (Health score, Failure risk)

Once the model is trained, you can use Power BI to visualize:

📊 **Battery Health Over Time** – Track degradation trends.  
📊 **Effect of Charging Mode on Health** – Compare slow vs. fast charging.  
📊 **Battery Efficiency vs. Health** – See how efficiency changes with degradation.

# **Geographical Mapping** (If data includes locations of battery usage)

-- STEP 1: Load Data into Power BI

-- Import 'ev\_battery\_charging\_data.csv' and name the table 'BatteryData'

-- STEP 2: Create Calculated Columns

-- Remaining Life Calculation

Remaining\_Life = 100 - BatteryData[Degradation Rate (%)]

-- Battery Efficiency Per Cycle

Battery\_Efficiency\_Per\_Cycle = DIVIDE( SUM(BatteryData[Remaining\_Life]), SUM(BatteryData[Charging Cycles]) )

-- STEP 3: Create Measures for KPIs

-- Average Remaining Life

Avg\_Remaining\_Life = AVERAGE(BatteryData[Remaining\_Life])

-- Average Degradation Rate

Avg\_Degradation\_Rate = AVERAGE(BatteryData[Degradation Rate (%)])

-- Total Charging Cycles

Total\_Charging\_Cycles = SUM(BatteryData[Charging Cycles])

-- STEP 4: Create Visualization Setup

-- 1. KPI Cards:

-- - Card for [Avg\_Remaining\_Life]

-- - Card for [Avg\_Degradation\_Rate]

-- - Card for [Total\_Charging\_Cycles]

-- 2. Line Chart: Degradation Rate vs. Charging Cycles

-- - X-axis: BatteryData[Charging Cycles]

-- - Y-axis: BatteryData[Degradation Rate (%)], Aggregation: AVERAGE

-- 3. Scatter Plot: Battery Temp vs. Efficiency

-- - X-axis: BatteryData[Battery Temp (°C)]

-- - Y-axis: BatteryData[Efficiency (%)]

-- - Size: BatteryData[Charging Cycles]

-- 4. Bar Chart: Battery Type vs. Avg Degradation

Avg\_Degradation\_By\_BatteryType = CALCULATE(AVERAGE(BatteryData[Degradation Rate (%)], GROUPBY(BatteryData, BatteryData[Battery Type])))

-- 5. Filters:

-- - Create slicers for BatteryData[Battery Type], BatteryData[EV Model], and BatteryData[Charging Mode]

-- FINAL STEP: Format visuals, apply colors, and customize dashboard layout

### **Potential Use Cases**

✅ **EV Manufacturers** to optimize battery life  
✅ **Industrial Equipment** maintenance prediction  
✅ **Consumer Electronics** companies to improve battery warranties

For **Lithium-Ion (Li-Ion) batteries**, the health is typically evaluated based on the **State of Health (SoH)**, which represents the remaining capacity relative to the original capacity when new.

### ****Battery Health (SoH) Thresholds:****

| **Battery Health (%)** | **Condition** | **Remarks** |
| --- | --- | --- |
| **100% - 90%** | Excellent | New or nearly new battery. |
| **89% - 80%** | Good | Still performs well, minor degradation. |
| **79% - 70%** | Fair | Noticeable capacity loss, reduced runtime. |
| **69% - 50%** | Poor | Significant degradation, shorter battery life. |
| **Below 50%** | Bad / Needs Replacement | Battery struggles to hold charge, should be replaced. |

### ****When is a Li-Ion Battery Considered "Bad"?****

* **Below 70%**: Performance degradation is significant.
* **Below 50%**: Battery may shut down frequently or not hold a charge for long. **Replacement is recommended**.

Most **smartphones, laptops, and EV manufacturers** consider battery replacement **below 80% - 70% health**, depending on the application.

****Presentation rough data :****

**<https://chatgpt.com/canvas/shared/67c6da7e9b3c81918926186dbaaaf554>**

## ****2. Machine Learning-Based Regression Models****

These models are useful when time dependency is not as strong, or when additional features (e.g., temperature, charging cycles) influence predictions.

### ****🔹 Random Forest Regression****

* **Best for**: Data with non-linear relationships.
* **Why use it?**: Handles outliers well, does not require assumptions about stationarity.
* **Limitation**: Does not capture temporal dependencies well.

### ****🔹 XGBoost Regression****

* **Best for**: Tabular time-series data with additional explanatory variables.
* **Why use it?**: Handles large datasets, missing values, and feature importance is interpretable.
* **Limitation**: Needs careful tuning to avoid overfitting.

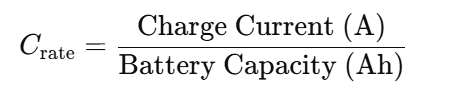
****Calculations:****

### ****df["Charge\_Rate"] = df["Voltage"] / df["Current"] --> 1Ah --> 1 hr --> 1 C (charging rate)****

### ****df["Capacity\_Loss"] = df["Initial\_Capacity"] - df["Remaining\_Capacity"]****

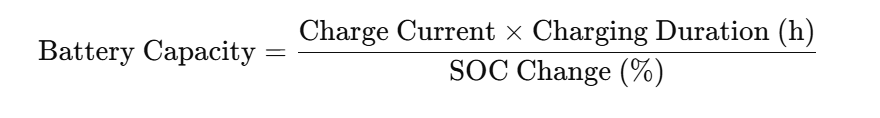
### ****1. Charge Rate Calculation (C-Rate)****

The **Charge Rate (C-Rate)** determines how fast the battery is charged relative to its capacity.

​

* **Charge Current (A)** → Column **"Current (A)"**
* **Battery Capacity (Ah)** → Not provided directly. If you have it, use that; otherwise, estimate based on EV model or battery type.

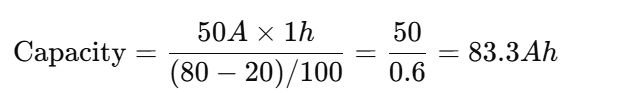
If **Charging Duration (min)** is given, you can estimate capacity as:



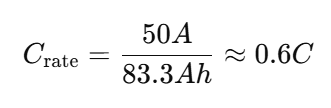
Example Calculation:

**Current = 50A**, Charging for **60 min (1 hr)**, SOC from **20% to 80%**

Estimated Battery Capacity:

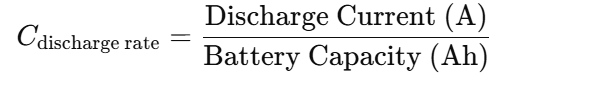


Charge Rate:



### ****2. Discharge Rate Calculation****

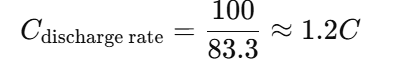
The **Discharge Rate (C-Rate)** follows a similar formula:



* **Discharge Current** → Use negative values from **"Current (A)"** (during vehicle operation).
* **Battery Capacity** → Estimated from charging phase.

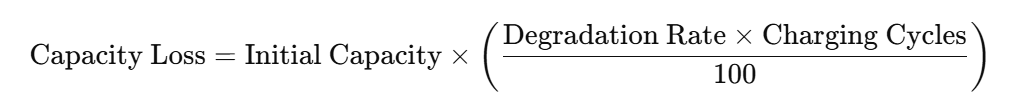
Example:

* **Current = -100A**, **Battery Capacity = 83.3Ah**



### ****3. Capacity Loss (Degradation)****

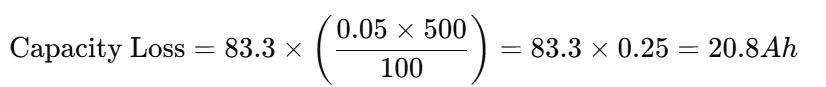
Capacity loss is related to the **Degradation Rate (%)** and **Charging Cycles**.



* **Degradation Rate (%)** → Column **"Degradation Rate (%)"**
* **Charging Cycles** → Column **"Charging Cycles"**
* **Initial Capacity** → Estimated from charging formula.

Example:

* Initial Capacity = **83.3Ah**, Degradation Rate = **0.05% per cycle**, **500 cycles**



**Remaining Capacity after 500 cycles**:

83.3−20.8=62.5Ah

### ****Implementation in Python (if needed)****

If you want to compute these for all data points, you can use **Pandas**:

python

KopierenBearbeiten

import pandas as pd

# Load dataset

df = pd.read\_csv("your\_data.csv")

# Assume an estimated battery capacity (Ah) based on the EV model or manufacturer data

estimated\_capacity = 83.3 # Example value

# Calculate Charge Rate (C-Rate)

df['Charge Rate (C)'] = df['Current (A)'].abs() / estimated\_capacity

# Calculate Discharge Rate (C-Rate) for negative currents

df['Discharge Rate (C)'] = df.apply(lambda x: abs(x['Current (A)']) / estimated\_capacity if x['Current (A)'] < 0 else 0, axis=1)

# Calculate Capacity Loss

df['Capacity Loss (Ah)'] = estimated\_capacity \* (df['Degradation Rate (%)'] \* df['Charging Cycles'] / 100)

# Save the results

df.to\_csv("processed\_data.csv", index=False)

****Lithium-ion Battery Recycling & Sustainability Analysis****

* **Goal:** Analyze battery waste data & predict best recycling strategies.
* **Data Needed:** Industry reports, recycling center data, material compositions.
* **Tech Stack:** NLP, Data Visualization (Power BI/Tableau), Clustering.
* **Outcome:** Data-driven policies for sustainable battery disposal.

As demand for lithium-ion (Li-ion) batteries grows, concerns around recycling, material recovery, and sustainability increase. This project aims to analyze battery waste data, identify trends, and suggest optimal recycling strategies using data science techniques.