Final Report:

Remaining Useful Life (RUL) Predictor for NMC-LCO 18650 Batteries

Problem statement:

Accurately estimating the Remaining Useful Life (RUL) of a battery is crucial for various applications, including electric vehicles, renewable energy storage systems, and portable electronics. A precise RUL prediction not only enhances operational efficiency but also aids in identifying optimal charge-discharge patterns to extend battery lifespan. This project focuses on developing a reliable RUL prediction model specifically for NMC-LCO 18650 batteries.

Objectives:

Develop an RUL Predictor: Leverage the available dataset to construct a predictive model that accurately estimates the Remaining Useful Life (RUL) of the batteries, closely aligning with observed values.

Explore Various ML Algorithms: Evaluate multiple machine learning algorithms to determine the most effective approach for accurate RUL prediction.

Dataset Overview

The dataset is built by Ignacio Vinuales, and the process is described in https://github.com/ignavinuales/Battery_RUL_Prediction. The data is available for download on https://www.kaggle.com/datasets/ignaciovinuales/battery-remaining-useful-life-rul.

The Hawaii Natural Energy Institute conducted tests on 14 NMC-LCO 18650 batteries, each with a nominal capacity of 2.8 Ah. These batteries were cycled 1,000 times at a temperature of 25°C, using a constant current-constant voltage (CC-CV) charge rate of C/2 and a discharge rate of 1.5 C. The dataset contains detailed measurements for all 14 batteries over the course of these 1,000 cycles, with various features and corresponding descriptions, as outlined in the table below.

S. N.	Features	Description
i.	Cycle_Index	Represents a particular battery cycle. Starts with 1 for each battery and will reach its maximum until the battery state of health (SOH) goes below the threshold value.
ii.	Discharge Time (s)	Time in seconds that takes the voltage to reach its minimum value in one discharge cycle.
iii.	Decrement 3.6-3.4V (s)	Represents the time in seconds which the voltage takes to drop from 3.6 V to 3.4 V during a discharge cycle.
iv.	Max. Voltage Dischar. (V)	Initial and maximum voltage in the discharging phase.
v.	Min. Voltage Charg. (V)	Initial value of Voltage when charging.
vi.	Time at 4.15V (s)	Time to reach 4.15 V in charging phase.
vii.	Time constant current (s)	Time in which the current stays constant at its maximum value.
viii.	Charging time (s)	Total time in seconds for charging.
ix.	RUL	Remaining useful life in terms of number of remaining charge discharge cycles before the SOH goes below threshold.

Data Wrangling:

The dataset consists of 15,064 rows and 9 columns, as described in the previous table. An initial inspection revealed the following:

i. Since this dataset is derived from experimental data, no missing values were found, and all features are of numeric type.

ii. Negative values were observed in two columns, *Decrement 3.6-3.4V (s)* and *Time at 4.15V (s)*, which are theoretically impossible. These anomalies may have resulted from machine malfunctions or other sources of error. Consequently, these values were removed to ensure data integrity before conducting further outlier analysis.

iii. Outliers were identified using the Profile Report from *ydata_profiling* and subsequently confirmed by examining box plots for each feature.

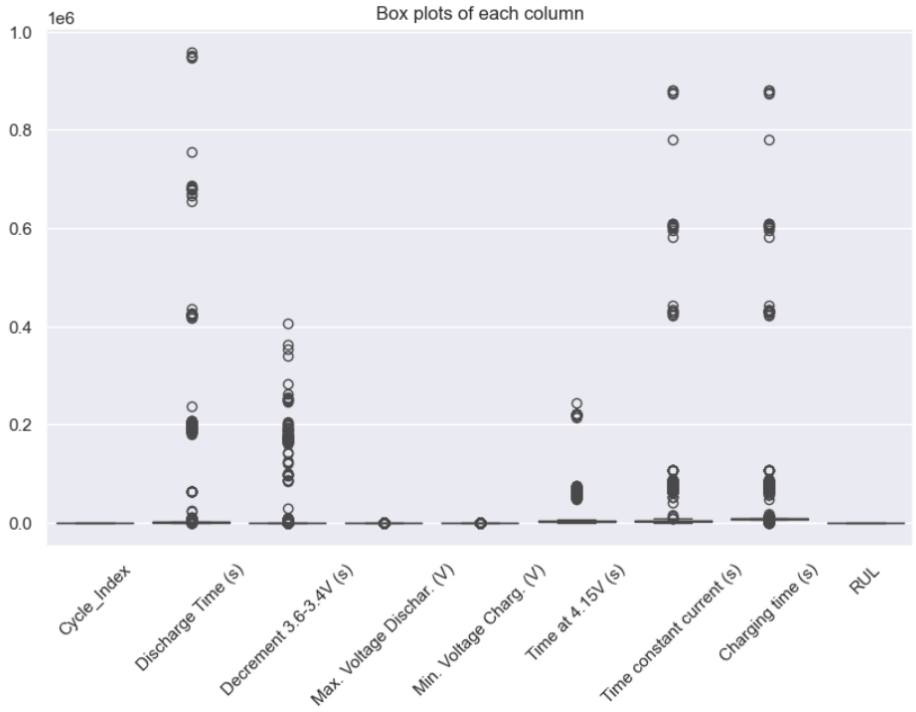

Figure 1. Box plot before removing outliers

Upon closer examination, I observed instances where time values (in seconds) abruptly changed by more than a factor of five between consecutive rows. Drawing on my prior experience with battery systems, I concluded that such anomalies likely arise from defects in the battery, equipment malfunctions, sudden movements during measurements, or other random factors. Researchers typically remove such data to maintain dataset quality. Before proceeding with deletion, I analyzed the frequency of these occurrences. It was found that this anomaly appeared only 191 times across more than 15,000 rows, falling into two distinct scenarios.

- **Condition 1:** When the *Discharge Time (s)* is significantly less than the median, and the next cycle shows an increase by more than fivefold, the previous cycle is considered defective and should be

removed. This can be addressed by applying a lower threshold, such as 5% of the median, to filter out the defective cycles.

- **Condition 2:** When the *Discharge Time (s)* is approximately equal to the median, but the next cycle increases by more than fivefold, the subsequent cycle is deemed defective and should be removed.

This can be accomplished by removing data points that exceed a specified upper threshold. A common best practice is to drop values that are more than three times the interquartile range (IQR) above the median. After applying this method, the cleaned dataset contained 14,696 rows and 9 columns.

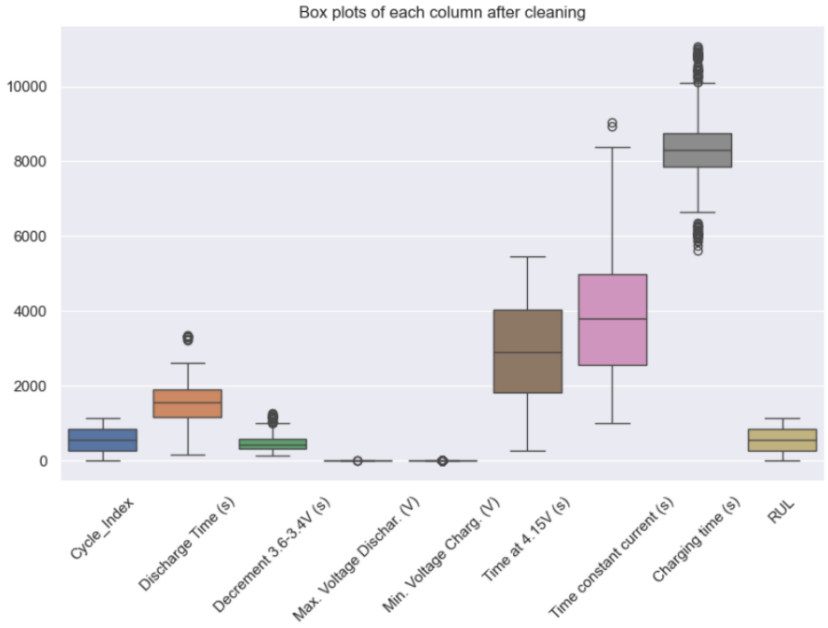

Figure 2. Box plot after removing outliers

iv. I analyzed the correlation among features both before and after removing outliers and erroneous data, as illustrated below. It became evident that these anomalies were contributing to weak correlations between certain features. After removing the outliers and errors, the dataset displayed much stronger correlations between the features.

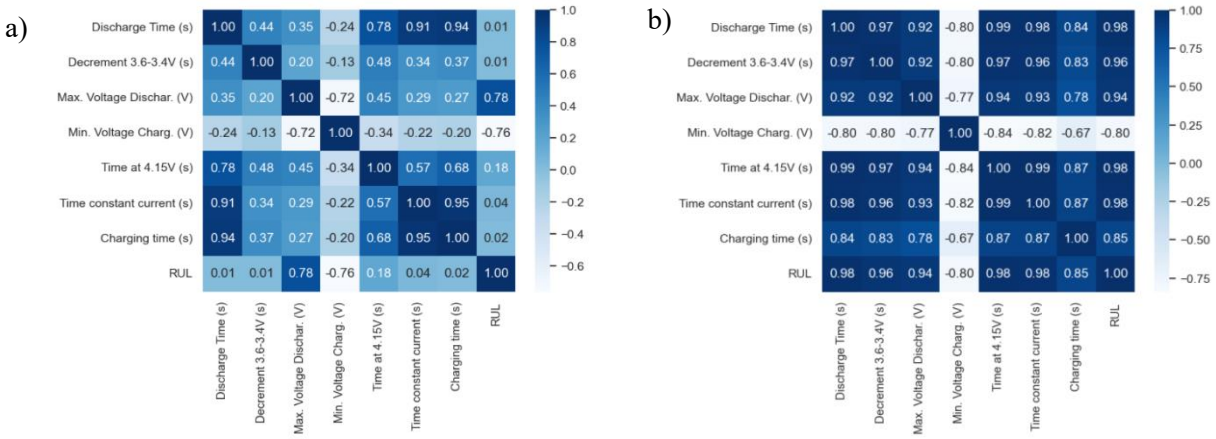

Figure 3. Correlation heatmap among features a) before, b) after removing errors and outliers

Exploratory data analysis (EDA):

A significant drawback of having highly correlated features is the potential for multicollinearity, which can introduce redundant information and reduce model efficiency. To address these issues, careful feature selection, dimensionality reduction techniques, or regularization methods are often required. I conducted a Principal Component Analysis (PCA) and discovered that the data is nearly one-dimensional, with the potential to be interpreted as two-dimensional. This second dimension accounts for nearly 95% of the variance, as clearly illustrated in the following figures.

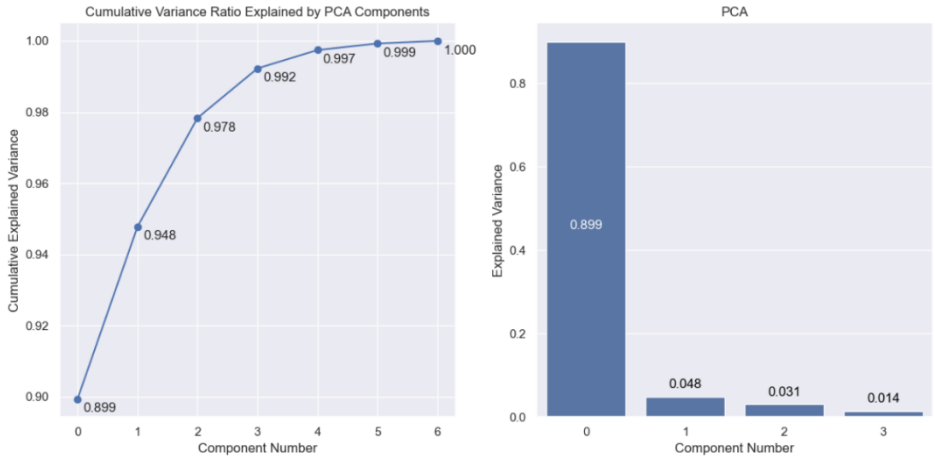

Figure 4. PCA Analysis and variance explained by each component

Among features, I looked at "**Time at 4.15V (s)**" to visualize its correlation with other features as it is one the features with very high correlation with all other features.

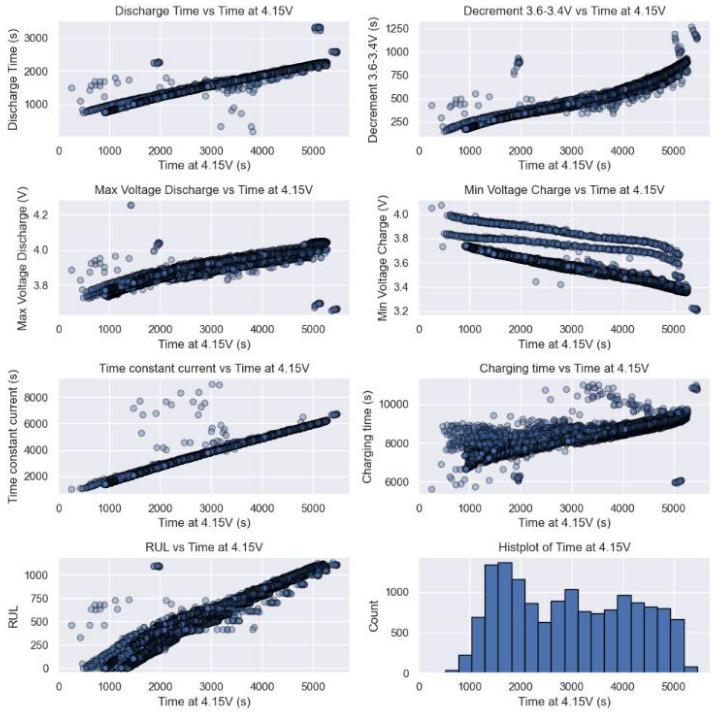

Figure 5. (first seven plots) Scatter plots between "Time at 4.15V (s)" and all other features. (last plot) Histogram of "Time at 4.15 V (s)".

Feature Engineering:

Given that most features exhibit high correlation with one another, it is crucial to identify which of them are most significant in predicting the target variable. L1 Regularization (Lasso) is particularly effective for enhancing interpretability and addressing multicollinearity in non-categorical data.

To begin, I standardized the features, which is essential for ensuring accurate and comparable coefficient values. This step leads to more reliable feature selection and interpretation.

U By employing L1 Regularization (Lasso) for feature selection on the dataset, I identified the most significant features and their corresponding coefficients. The analysis revealed that *Discharge Time (s)*, *Time at 4.15V (s)*, and *Max. Voltage Dischar. (V)* are the most influential positive features, whereas *Decrement 3.6-3.4V (s)* has a negative impact. The feature importance was visualized using a bar plot, which clearly illustrates each feature's contribution to the model, as shown below,

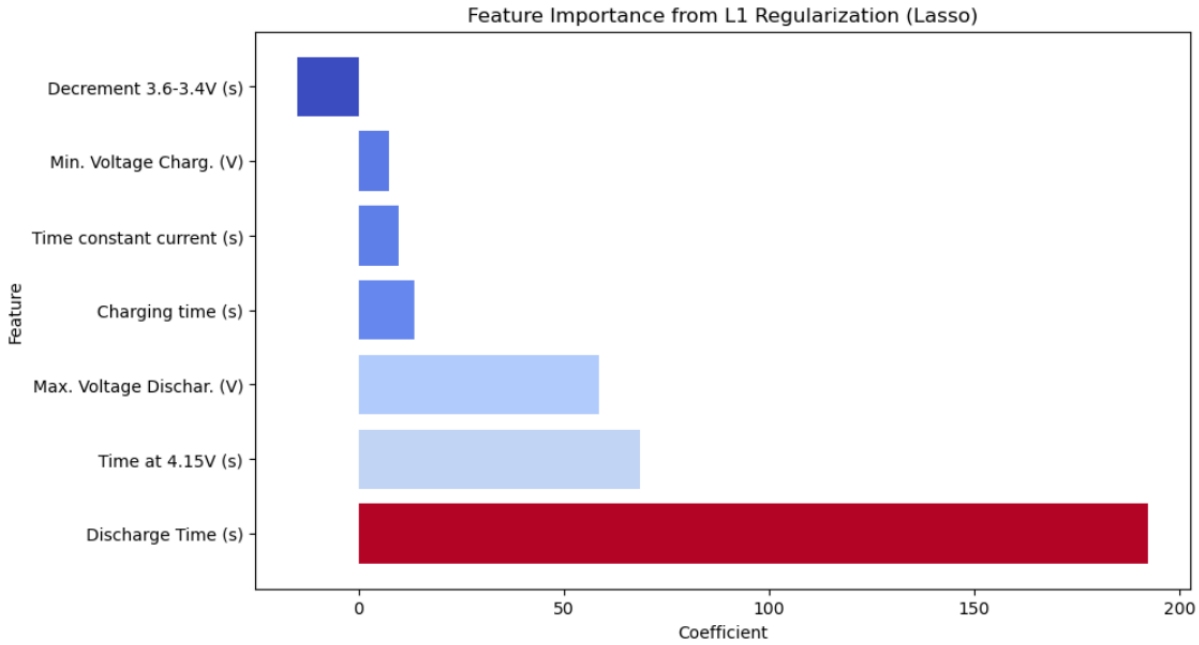

Figure 6. Feature Importance from L1 Regularization (Lasso)

While it may be tempting to remove all features with low importance scores, it is important to remember that these importances are specific to the Lasso model and may differ for other models. For this reason, I decided to retain these features for the time being..

Model Development:

After splitting the dataset into training and testing sets and standardizing the features, I developed and fine-tuned four different models. The performance of each model was evaluated using key metrics, including the *R^2 score*, *Root Mean Square Error (RMSE)*, and *Mean Absolute Error (MAE)*. This comparative analysis facilitated the identification of the final best-performing model.

Model 1 : Linear Regression

The simple linear regression model demonstrated significant promise, explaining approximately 97.6% of the variance in both the training and testing datasets, indicating a strong fit. The *MAE* and

RMSE values further suggest that the linear model generalizes well, as the performance metrics for both training and testing sets are closely aligned.

After incorporating all features into the linear model, I assessed the impact of using a reduced number of features on model performance by applying the *GridSearchCV* method.

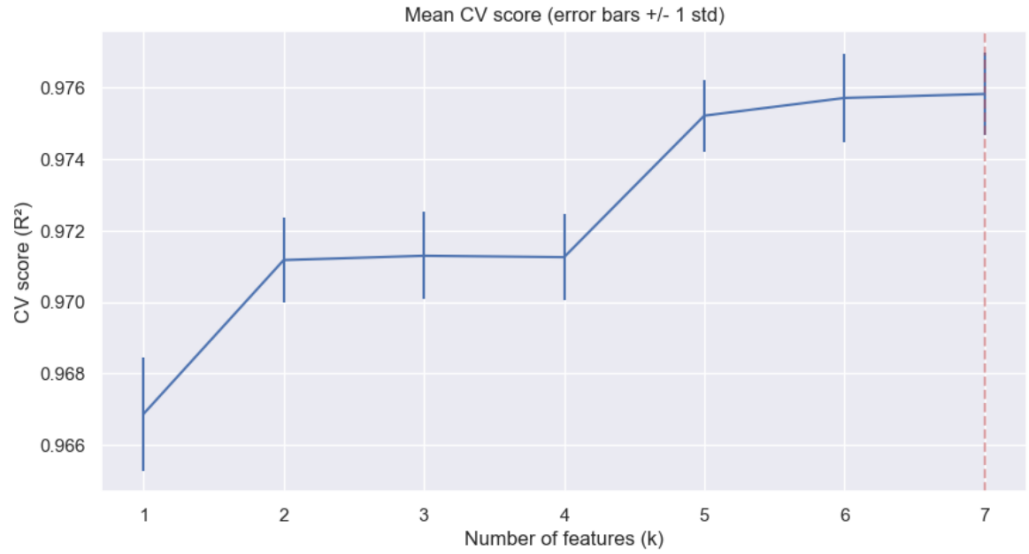

Figure 7. GridSearchCV to find best number of features in Linear regression model

The graph above indicates that even with a single feature, a high R^2 score of 0.9669 can be achieved, underscoring that certain features in the dataset are strongly correlated with the target variable. As additional features are incorporated, R^2 scores increase only marginally, peaking at 0.9758 when all seven features are included. This suggests that while adding more features yields slight improvements in model performance, the initial few features capture most of the variance in the data.

Furthermore, the feature importance calculations yielded results nearly identical to those presented earlier in Figure 6. However, given that the total number of features is only seven and that including all of them leads to improved predictions, there is no necessity to exclude any features for the sake of reducing training time.

Model 2 : Lasso (L1 regularization)

Given the strong intercorrelation among features, applying Lasso regression (L1 regularization) is a suitable next step. Lasso helps mitigate multicollinearity by shrinking the coefficients of less important features to zero, effectively selecting a subset of relevant features. This approach not only identifies the most significant predictors but also simplifies the model, potentially enhancing its generalization capabilities.

The results from applying Lasso regression with various regularization parameters (alpha) demonstrate that the model's R^2 score remains relatively high across different values. Overall, the findings indicate that the features are strong predictors of the target variable, and even with regularization, the model maintains robust predictive performance. The slight decline in R^2 at higher alpha values suggests that regularization begins to restrict the influence of less important features.

Ultimately, I selected an alpha value of 0.01 to fit the Lasso model and compared its performance with that of the linear regression model. The results revealed that the two models exhibit almost identical performance in terms of evaluation metrics such as R^2, MAE, and RMSE, indicating that Lasso

regularization has minimal impact. This outcome underscores the robustness of the features utilized in the prediction.

Model 3 : Gradient Boosting Regressor

Unlike the previous two models, both the Gradient Boosting Regressor and the Random Forest Regressor are computationally intensive. Therefore, hyperparameter tuning was conducted using GridSearchCV, evaluating one parameter at a time. For the Gradient Boosting Regressor, the optimal hyperparameter values were determined to be: n_estimators = 300, learning_rate = 0.10, and max_depth = 10, as illustrated in the following figures.

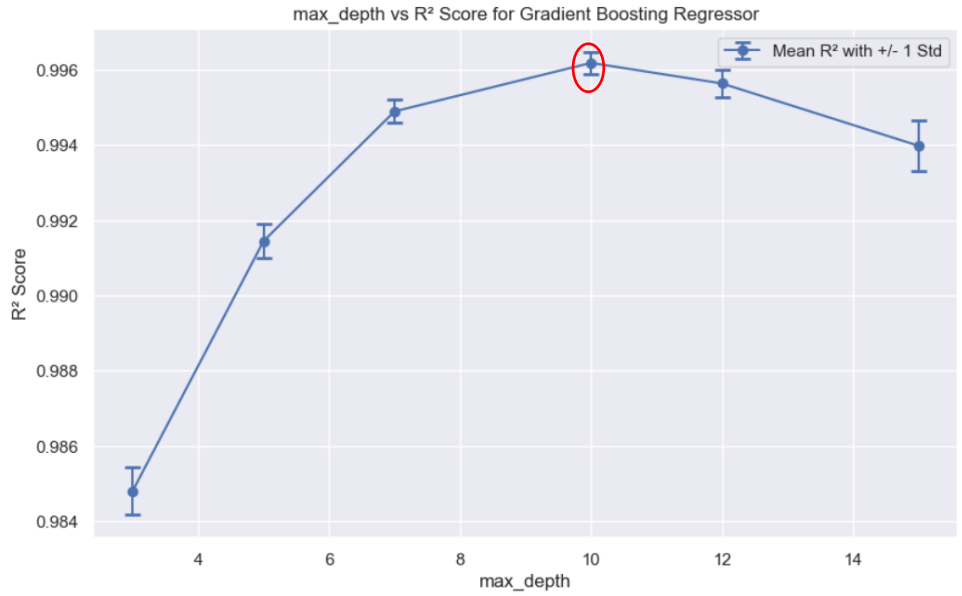

Figure 9. Hyperparameter tuning for max_depth

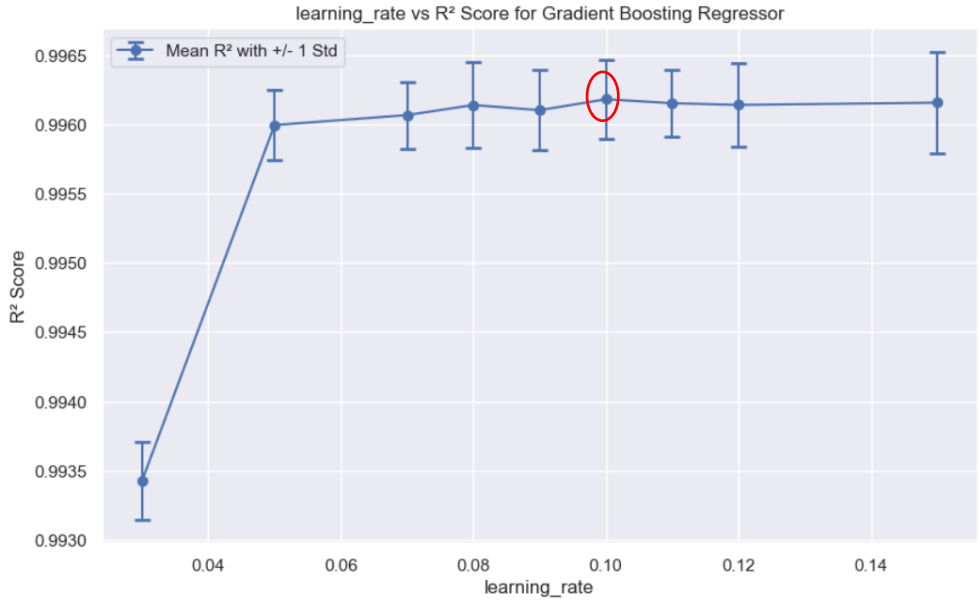

Figure 8. Hyperparameter tuning for learning_rate

Figure 10. Hyperparameter tuning for n_estimators

The model with n_estimators = 1000 achieves the highest R^2 score of 0.9964, indicating optimal performance with a very low standard deviation. As the number of estimators increases, the R^2 score improves gradually, suggesting that additional estimators enhance the model's capacity to fit the data, leading to better overall performance. While the R^2 score continues to improve with more estimators, the increases become marginal beyond 300 estimators, and the standard deviation remains stable across all values.

The model demonstrates an almost perfect R^2 score on the training set (0.999985) and a very high score on the test set (0.997332), reflecting excellent predictive performance. The training Mean Absolute Error (MAE) is very low (1.224533), while the test MAE is higher (16.561884), which may indicate some degree of overfitting. Similarly, the training Root Mean Square Error (RMSE) is low (0.898504), while the test RMSE is higher (8.434242), further suggesting potential overfitting.

Feature importance:

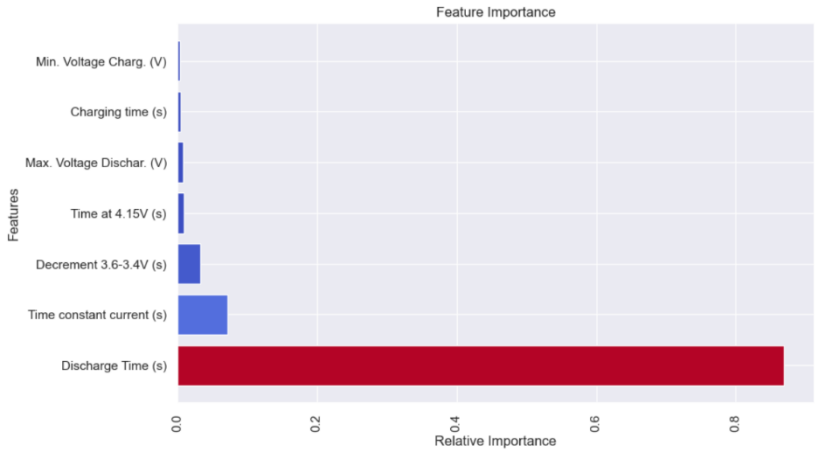

*Figure 11. Feature importance plot for **Gradient Boosting Regressor.***

Discharge Time (s) has the highest importance score (0.870830), consistent with the findings from both linear regression and Lasso, indicating that it is the most significant feature for predicting the target variable. Although two additional features may also warrant inclusion, the remaining features can be discarded based on the insights from the graph above.

Model 4 : Random Forest Regressor

Figure 12. Hyperparameter tuning for n_estimators

The optimal value for n_estimators is 400, achieving the highest R^2 score of 0.996326 with a very small standard deviation (0.000327), indicating excellent and stable performance. As the number of estimators increases from 10 to 400, the R^2 score shows consistent improvement, suggesting that additional estimators generally enhance model performance. However, the gains become marginal beyond 200 estimators.

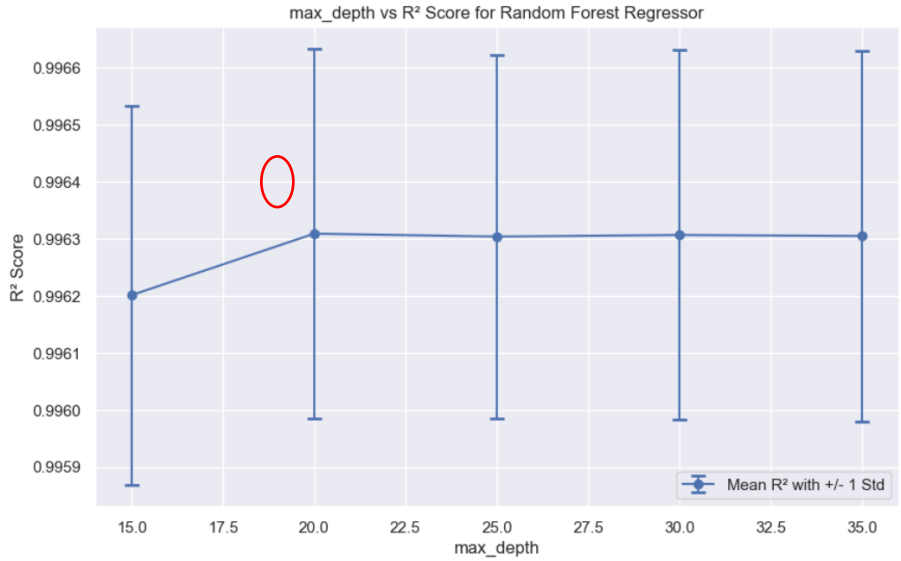

Figure 13. Hyperparameter tuning for max_depth

The optimal value for max_depth is 20, achieving the highest R^2 score of 0.996309 with a stable standard deviation of 0.000324. The R^2 score improves as max_depth increases from 15 to 20, but beyond 20, it remains relatively stable with only slight fluctuations. For the Random Forest Regressor with n_estimators set to 200, a max_depth of 20 yields the best performance, maximizing the R^2 score while maintaining stability in the model's performance.

The model demonstrates exceptional performance, with an R^2 score of 0.999542 on the training data and 0.997222 on the test data, indicating high accuracy in explaining variance. The training Mean Absolute Error (MAE) is 6.837620, while the test MAE is 16.900478, suggesting a degree of overfitting. The training Root Mean Square Error (RMSE) is 3.419183, and the test RMSE is 8.674764, further supporting the indication of potential overfitting.

These results are consistent with those of the Gradient Boosting Regressor, suggesting that both models perform exceptionally well but may be overfitting the training data to some extent.

Feature importance:

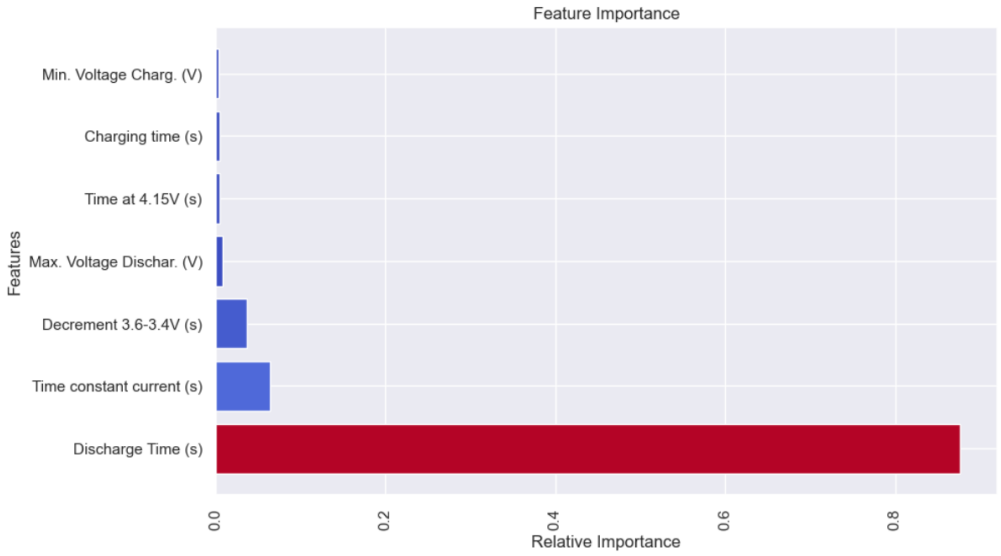

Figure 14. Feature importance plot for Random Forrest Regressor

These results indicate that, similar to the Gradient Boosting Regressor, the *Discharge Time (s)* feature plays a dominant role in the model's predictions, while the other features contribute less significantly to the model's overall performance.

Model Evaluation:

Metrics	Linear Regression		Lasso		**Gradient Boosting Regressor**		**Random Forest Regressor**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
R^2	0.976022	0.976887	0.976022	0.976884	0.999985	0.997332	0.999542	0.997222
MAE	49.483541	48.745986	49.483648	48.748702	1.224533	16.561884	6.837620	16.900478
RMSE	37.018648	36.684213	37.026780	36.692422	0.898504	8.434242	3.419183	8.674764

The Gradient Boosting Regressor (GBR) model, as highlighted in the table above, demonstrated the highest R^2 scores along with the lowest MAE and RMSE on the training set, although its performance on the test set was slightly better than that of the Random Forest Regressor (RFR).

Conclusions:

After evaluating various models, including Linear Regression, Lasso Regression, Gradient Boosting Regressor (GBR), and Random Forest Regressor (RFR), the following findings were observed:

Linear Model and Lasso Regression: Both models exhibited strong predictive capabilities, evidenced by high R^2 scores. The Lasso Regression, with an alpha of 0.01, demonstrated nearly identical performance metrics to the linear model, indicating that feature selection was not a significant factor in this context.

Gradient Boosting Regressor (GBR): This model achieved the highest R^2 scores along with the lowest MAE and RMSE on the training set, although its performance on the test set was slightly better than that of the RFR. It is important to note that GBR requires considerably more computational time for training.

Random Forest Regressor (RFR): This model provided performance metrics that were very close to those of GBR, featuring slightly higher MAE and RMSE but similar R^2 scores. RFR is more computationally efficient and trains faster than GBR.

Future Recommendations:

The current dataset comprises only one experimental setup, specifically a nominal capacity of 2.8 Ah, cycled over 1000 times at 25 °C with a CC-CV charge rate of C/2 and a discharge rate of 1.5 C. However, real-world environmental conditions are rarely controlled in this manner. To improve the model's applicability, it would be beneficial to acquire real-world data that encompasses a variety of charging and discharging conditions, as well as additional variables such as temperature, humidity, and pressure.

Furthermore, increasing the number of batteries in the dataset could enhance the model's predictability and robustness, allowing for a more comprehensive understanding of battery performance under diverse conditions.

Credits:

I would like to express my sincere gratitude to **Ignacio Vinuales** for making this dataset publicly available, providing invaluable opportunities for aspiring data scientists to practice and refine their machine learning skills. I would also like to extend my heartfelt thanks to my Springboard mentor, **Vinit Koshti**, for his insightful guidance and thoughtful advice throughout this journey.