Exploratory Data Analysis and Machine Learning for Battery Temperature Prediction

Student Name: Ved Pramod Patel

Roll Number: 24BEE209

Faculty Name: Dr. Vipin Shukla

Dataset File: BC_35.csv

Submission Date: 20th February 2026

Department of Electrical Engineering

School of Energy Technology

Pandit Deendayal Energy University

1 Table of Contents

2 List of Figures

3 List of Tables

4 Introduction

In battery performance, safety, and lifetime, Temperature plays a crucial role in modern battery management systems. This report presents a complete exploratory data analysis (EDA) and machine learning (ML) workflow applied to a real battery cycling dataset. The primary objective is to predict the average battery temperature (Tavg) using electrical and operational parameters recorded during charge–discharge cycles.

The study systematically progresses through statistical analysis, graphical visualization, correlation assessment, and feature selection. Two machine learning models are subsequently implemented in MATLAB: Multiple Linear Regression as a baseline model and a tree-based regression model to capture nonlinear behaviour. Through standard metrices (R^2, RMSE, and MAE), performance evaluation of these two models has been implemented. Overall, this project emphasizes the importance of systematic data exploration, informed feature selection, and appropriate model choice when applying machine learning to engineering datasets.

5 Dataset Description

The given dataset contains measurements of the following variables taken during charge/discharge cycles of the battery:

- Dchg/Chg Cycle- Represents the step index within a charge–discharge process. It indicates the specific phase or stage of operation during a battery cycling test.
- BC- Stands for Battery Code (or Battery Channel). It acts as an identifier for the specific battery being tested.
- Vact- Represents the instantaneous measured voltage of the battery during operation.
- Discharge Current- Indicates the current drawn from the battery during the discharge phase (in Amperes).
- Charge Current- Represents the current supplied to the battery during charging (in Amperes).
- Chg/Dchg Cycle- Denotes the overall cycle number of the battery test.
- Ah- Represents the accumulated charge transferred through the battery.
- Tavg- Indicates the average temperature of the battery (in °C).

Complete statistical analysis is performed on all these variables. In ML prediction and analysis of this dataset, Tavg (Average Temperature) is treated as the target variable, whereas all remaining variables are considered as input features or predictors.

6 Statistical Analysis

For the given variables in dataset, following statistical measures are calculated: Mean, Median, Standard Deviation, Variance, Minimum, Maximum, Skewness and Kurtosis. Measures for all respective variables are mentioned in Table- 1 given below.

Table 1: Descriptive Statistics of Battery Dataset

Variable	Mean	Median	Std. Deviation	Variance	Minimum	Maximum	Skewness	Kurtosis
Dchg_ChgCycle	1.91	2	1.05	1.10	1	4	0.79	2.29
BC	35	35	0	0	35	35	NaN	NaN
Vact	2.96	3.02	0.20	0.04	1.99	3.53	-1.23	5.4
DischargeCurrent	38.22	39.98	21.02	441.96	0.08	80.05	0.78	2.29
ChargeCurrent	0.01	0.01	0.01	3.88E-05	0	0.22	10.52	346.69
Chg_DchgCycle	10	10	0	0	10	10	NaN	NaN
Ah	9.68	9.69	5.59	31.24	4.4E-05	19.50	1.63E-05	1.80
Tavg	35.82	34.93	1.67	2.79	34.20	41.45	1.3	3.76

6.1 Interpretation of the table

- Discharge Current possess the huge curve spread as Standard Deviation is very high (≈ 21). At the same time, its kurtosis is 2.29 (less than 3). Hence the curve will be platykurtic. It means that occasionally very high current is drawn from battery.
- Charge Current is extremely leptokurtic and may contain high number of outliers. It shows that battery's charging behaviour is not stable or high number of impulsive charging is present.
- Voltage behaviour is mostly stable but strong sudden drops in voltage exists. (Leptokurtic behaviour)
- Ah has low kurtosis and moderate standard deviation representing It as a stable and meaningful predictor for thermal behaviour.

The table already suggest deviations from normality, presence of heavy tails, and redundant features. Histogram analysis is therefore essential to validate these statistical findings, detect outliers, and guide appropriate preprocessing and model selection prior to machine learning.

7 Graphical Analysis

The interpretation stated in previous section will be verified by the histogram plots of each variable.

- Histograms show whether the data is evenly spread or concentrated in certain ranges. If most values lie in a narrow region, the model may struggle to learn patterns properly.
- It helps to identify extreme values that can disturb linear regression and increase prediction error. If the shape is clearly not symmetric, it indicates that simple linear assumptions may not be suitable for modeling.
- So, by studying histograms it will provide clarity regarding selecting the features for machine learning models by observing the values which are constants and other characteristics.

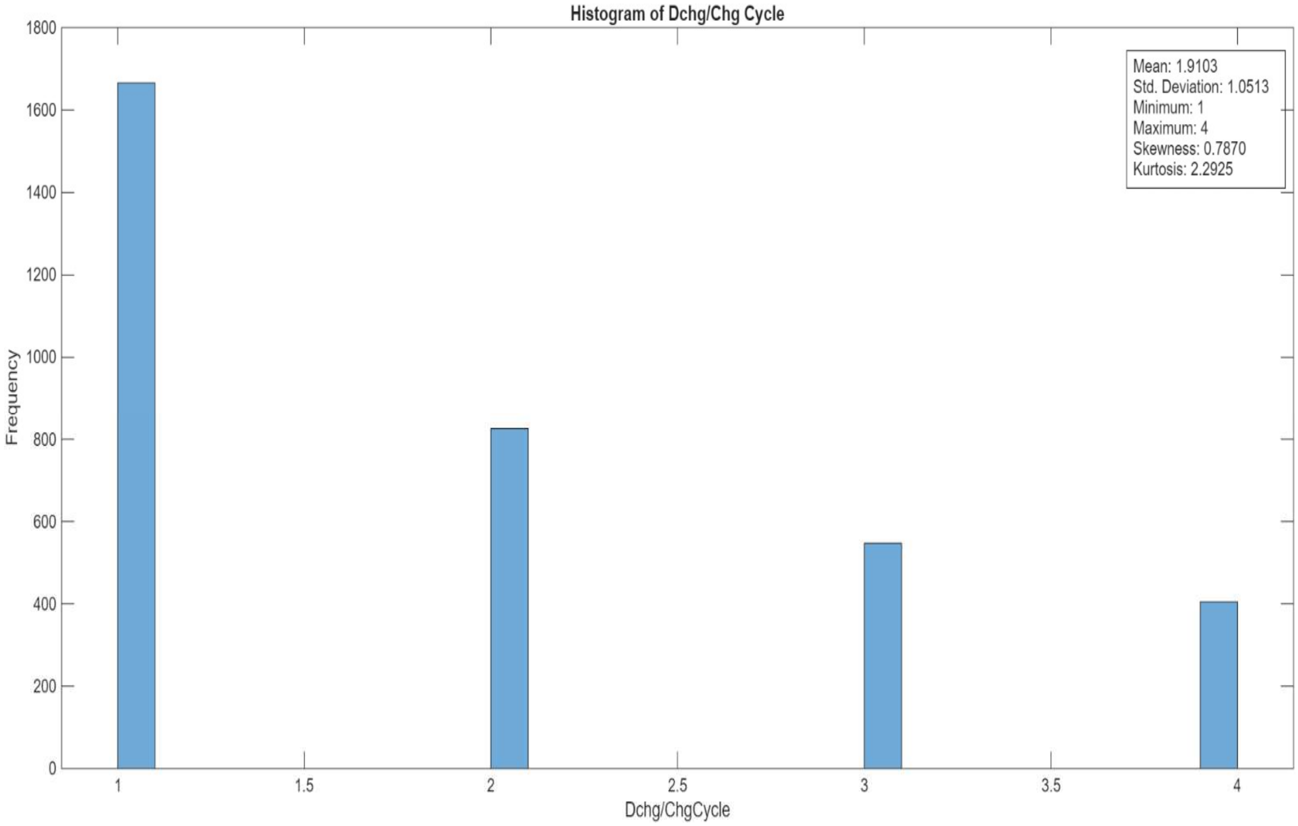


Figure 1: Histogram of Dchg/ChgCycle

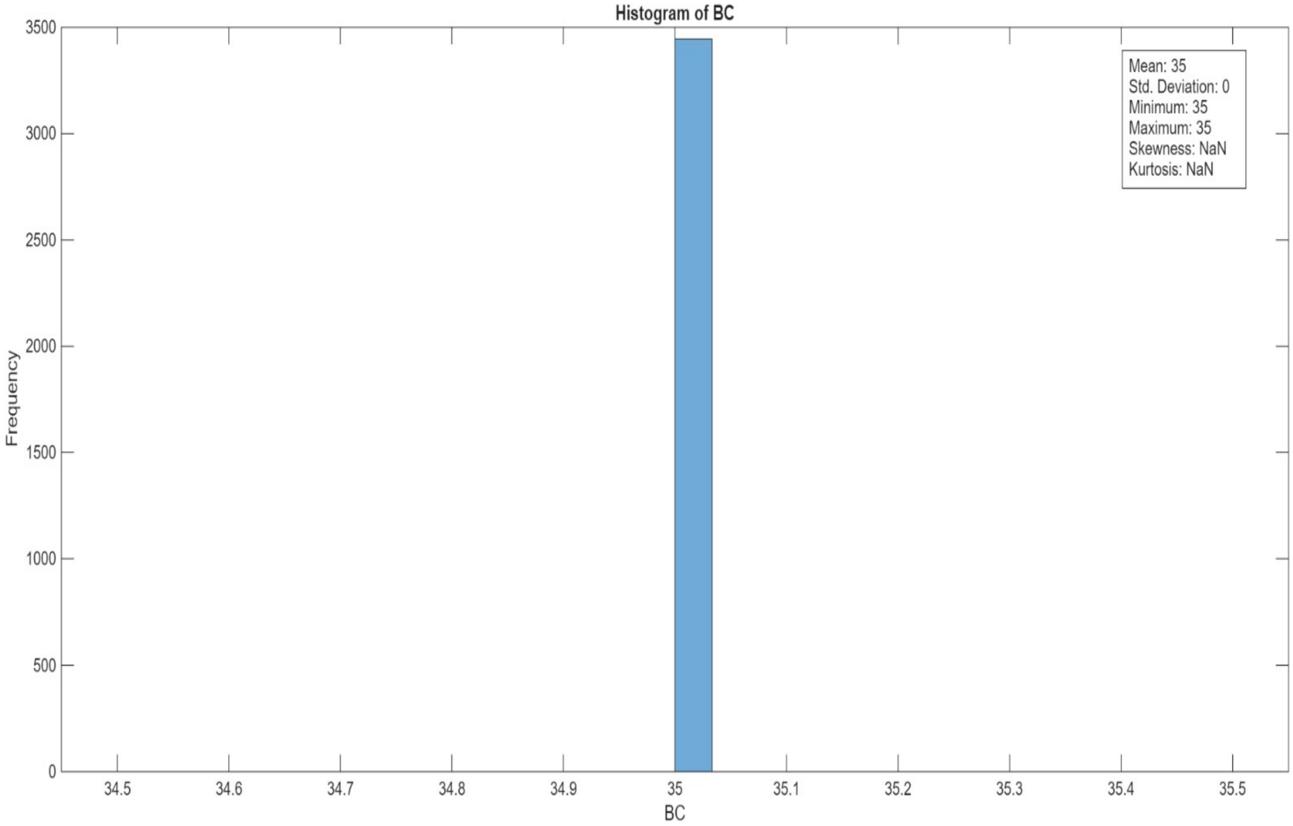


Figure 2: Histogram of BC

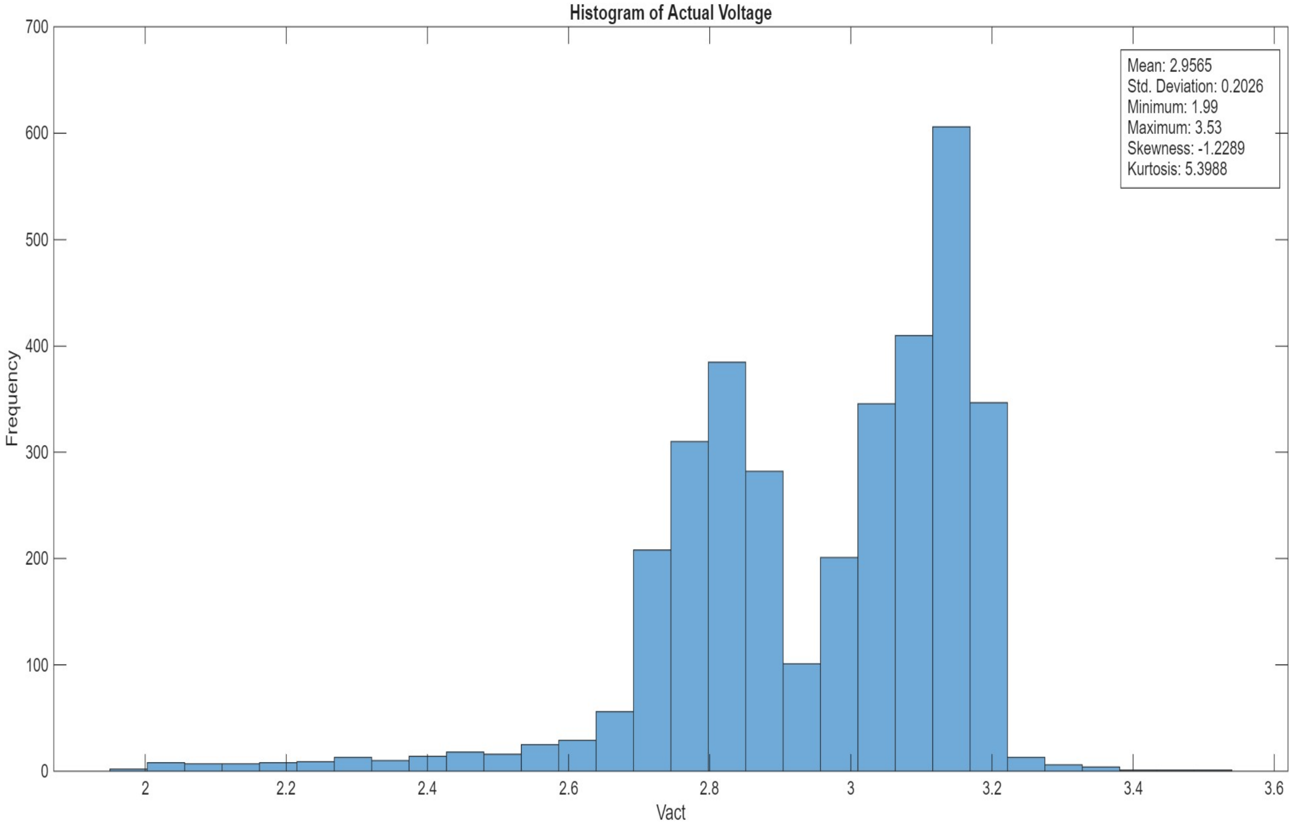


Figure 3: Histogram of Actual Voltage

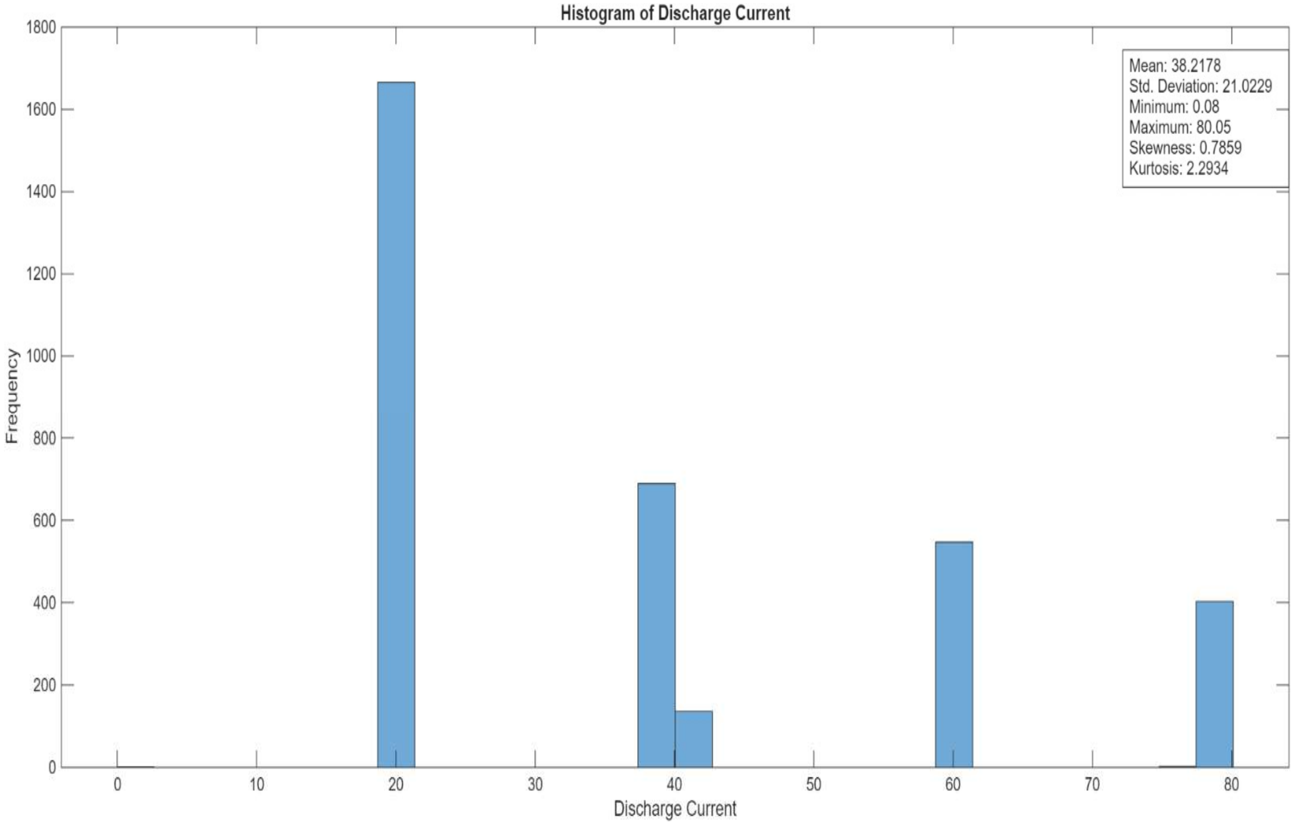


Figure 4: Histogram of Discharge Current

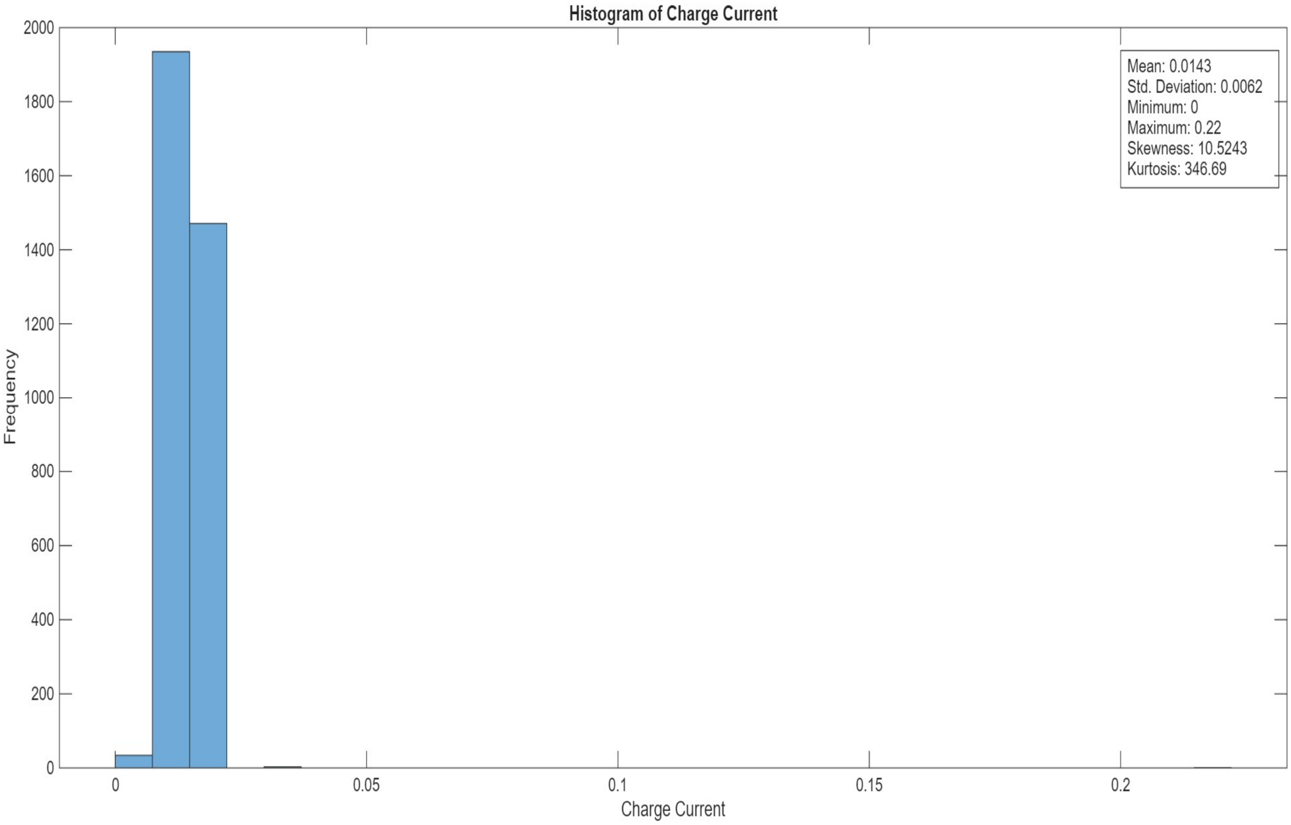


Figure 5: Histogram of Charge Current

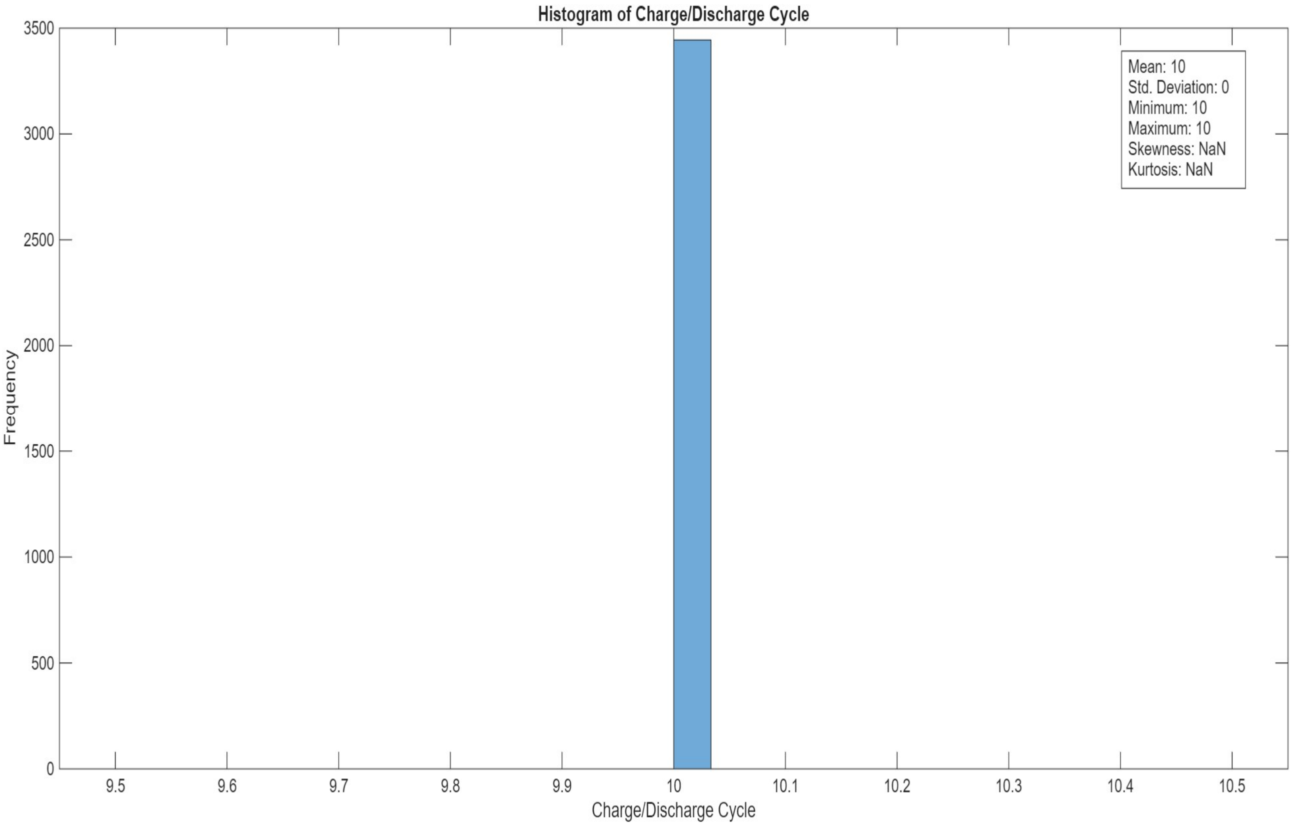


Figure 6: Histogram of Charge/Discharge Cycle

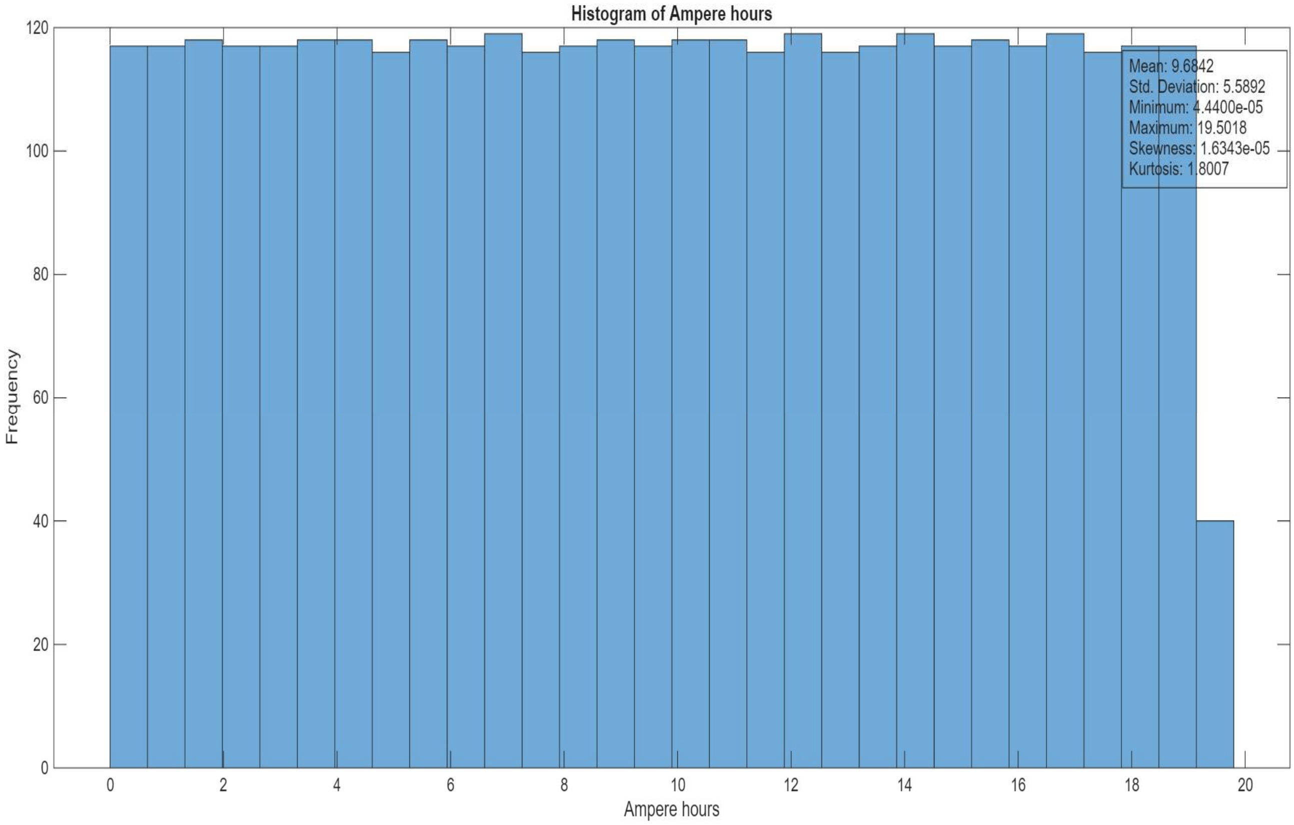


Figure 7: Histogram of Ampere hours

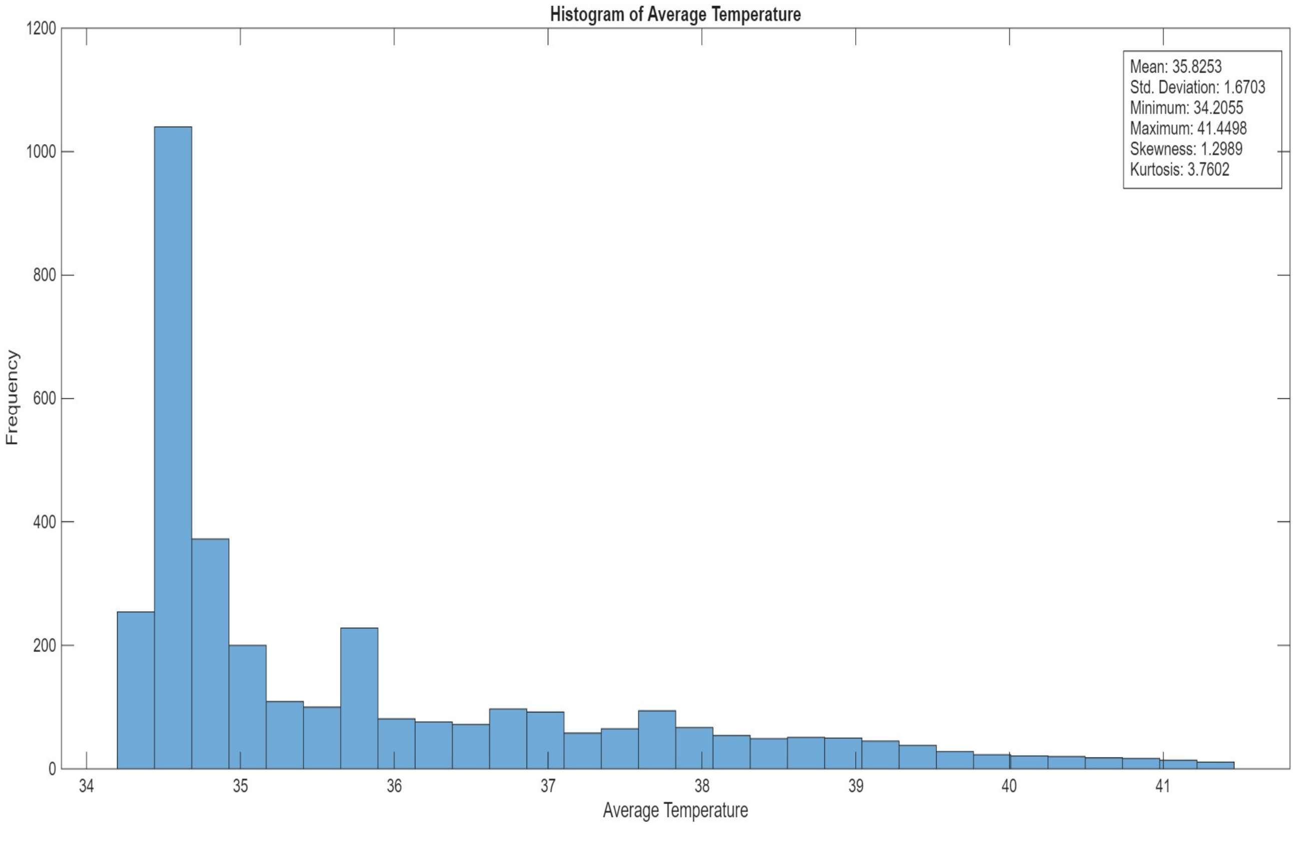


Figure 8: Histogram of Average Temperature

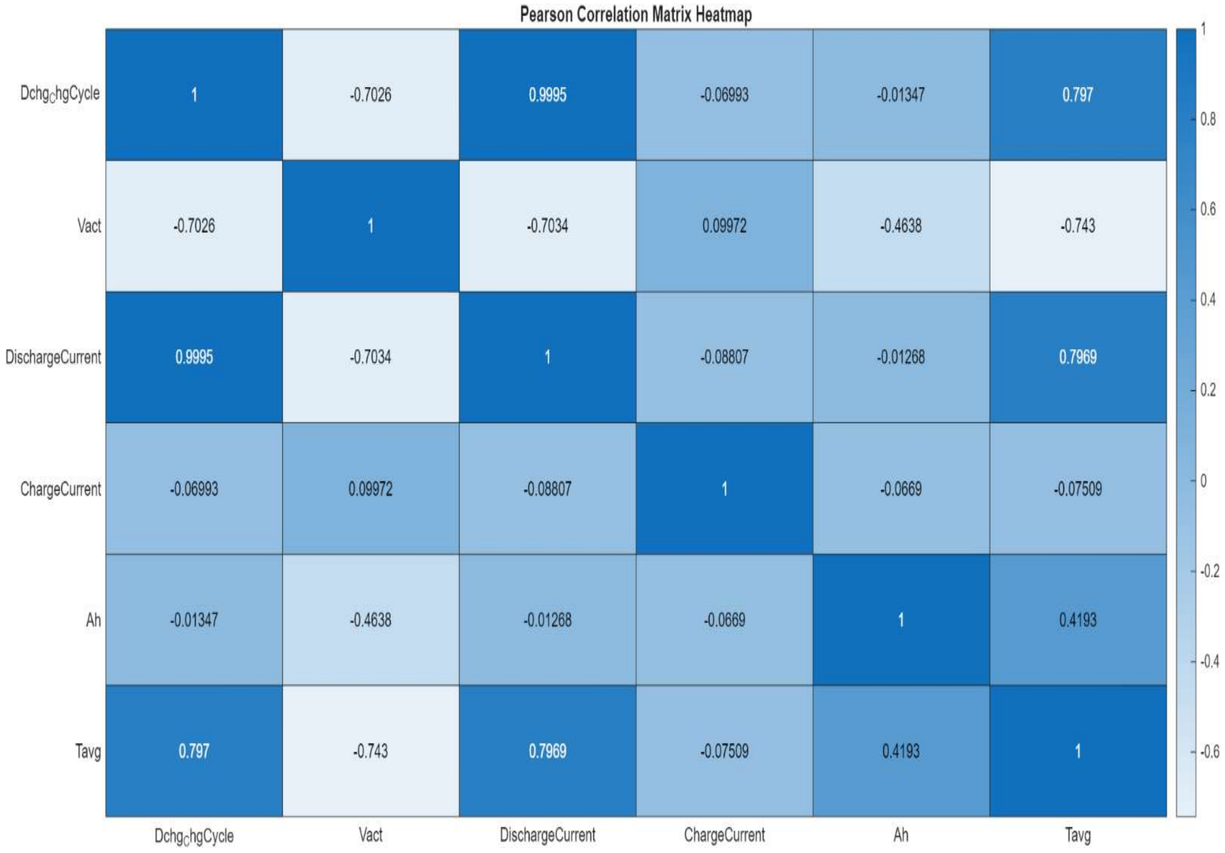
7.1 Interpretation of Histograms

- Figure 2 and Figure 6 represents the constant values of BC and Charge/Discharge Cycle respectively. It makes obvious fact that they carry no information and must be removed from modeling.
- In charge current, kurtosis numerically appears extremely high, the histogram visually explains why: charging occurs rarely but intensely. This impulsive behaviour strongly affects regression models and cannot be understood from the measured values alone.
- The Ah histogram appears uniform across its range, confirming stable accumulation of charge. Statistical skewness near zero suggests symmetry, but the histogram confirms there is no dominant region, meaning Ah contributes smoothly to temperature variation.
- The data of discharge current gives information about the curve by skewness and kurtosis. But it does not reveal the fact that the distribution is grouped or at discrete levels instead of being continuous which is evident from the Figure 4.

In general, the presence and frequency of outliers can be seen from the graph. For regression modeling, normalization and scaling of this dataset becomes important as variables have vastly different ranges and unscaled data biases model learning.

8 Correlation Analysis

8.1 Linear Correlation using Pearson Correlation Matrix



	Dchg$_C$hgCycle	Vact	DischargeCurrent	ChargeCurrent	Ah	Tavg
Dchg$_C$hgCycle	1	-0.7026	0.9995	-0.06993	-0.01347	0.797
Vact	-0.7026	1	-0.7034	0.09972	-0.4638	-0.743
DischargeCurrent	0.9995	-0.7034	1	-0.08807	-0.01268	0.7969
ChargeCurrent	-0.06993	0.09972	-0.08807	1	-0.0669	-0.07509
Ah	-0.01347	-0.4638	-0.01268	-0.0669	1	0.4193
Tavg	0.797	-0.743	0.7969	-0.07509	0.4193	1

Figure 9: Pearson Correlation Matrix Heatmap

8.2 Non-Linear Correlation using Spearman Correlation Matrix

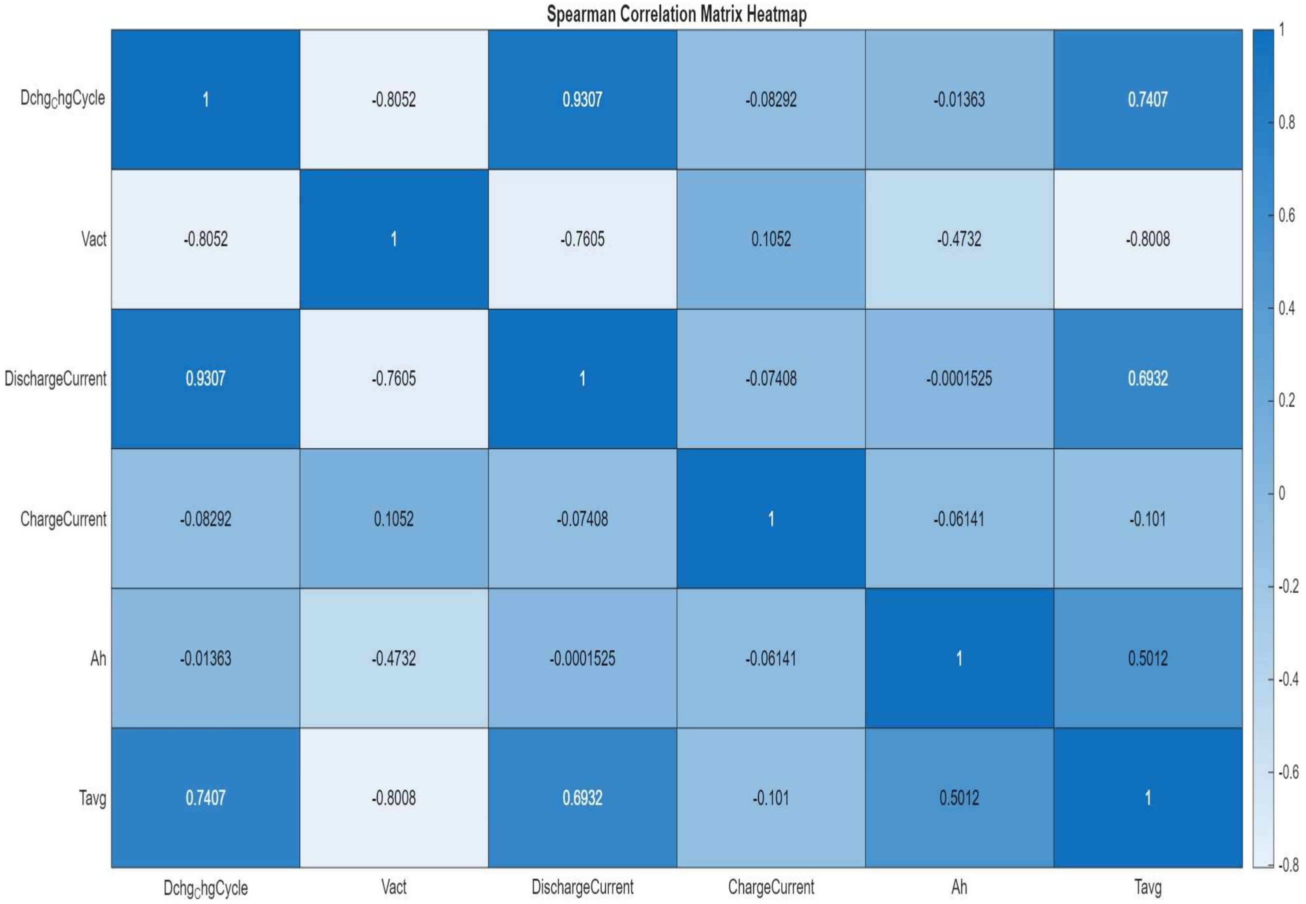


Figure 10: Spearman Rank Correlation Matrix Heatmap

To plot the heatmap of Linear and Non-Linear Correlation Matrix, variables ‘BC’ and ‘Charge/Discharge Cycle’ are removed as they do not show any variation in their value. Correlation Analysis shows the inter dependency between variables given in dataset.

8.3 Interpretation of Correlation Analysis

- Charge Current does not have significant dependencies on other variables as the correlation scores are nearly 0. Charging events are sparse and inconsistent (confirmed earlier by histograms). Hence, this variable becomes irrelevant for ML modeling.
- From both heatmaps, Discharge Current and Dchg/ChgCycle indicate a strong relationship. Practically, they contain identical information. Due to this, the model parameters in regression modeling would not be accurate. Hence, multicollinearity issues arise.
- Primary reason for the different values in both heatmaps is because Pearson Correlation measures the linear relationship between the variables and Spearman Correlation measures the monotonicity between the variables. Monotonicity indicates that variable increase or decrease with respect to other irrespective of relationship. (Linear, quadratic, etc.)

- Correlation analysis helped identify redundant variables, weak predictors (such as Charge Current), and nonlinear trends. Without this step, the model would include unnecessary features leading to poor accuracy.

9 Feature Selection for ML Modeling

Based on statistical and correlation analysis, following variables are selected as features for models to predict average temperature (Tavg).

1. Discharge Current
2. Actual Voltage
3. Ampere-hours

9.1 Justification

- Dchg/ChgCycle- Possess strong relationship with Discharge current as observed from heatmap. To prevent multicollinearity, it is discarded.
- Charge Current- removed as it does not contain much information to predict average temperature. (Discussed in previous section).
- BC and Charge/Discharge Cycle- removed because they are constant.
- Discharge Current, Actual Voltage, and Ampere-hours were selected as input features because they show the strongest and most meaningful correlation with battery temperature while representing the key electrical and energy factors that physically govern thermal behaviour.

10 Machine Learning Model Development

In this project, 2 Machine Learning Models are developed to predict the Average Temperature.

Model 1: Multiple Linear Regression

Model 2: Decision Tree

For both models, data set is randomly divided into 80-20 ratio, for training and testing respectively. Furthermore, the accuracy of both models is calculated using R^2, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) performance metrices.

11 Model Performance Evaluation

Table 2: Model Performance Comparison

Model	Dataset	R^2	RMSE	MAE
Multiple Linear Regression	Train	0.8204	0.7121	0.5263
Multiple Linear Regression	Test	0.8226	0.6854	0.5205
Decision Tree	Train	0.9998	0.0239	0.0146
Decision Tree	Test	0.9996	0.0344	0.0191

11.1 Conclusion from comparison

- The performance metrics clearly indicate that the Decision Tree model significantly outperforms Multiple Linear Regression in predicting average battery temperature.
- While Multiple Linear Regression achieves moderate accuracy ($R^2 \approx 0.82$) with relatively higher RMSE and MAE values, the Decision Tree model demonstrates near-perfect performance ($R^2 \approx 0.999$) with extremely low error values on both training and testing datasets.
- Importantly, the training and testing results for the Decision Tree are nearly identical, indicating excellent generalization and no significant overfitting. In contrast, the linear model shows stable but limited performance, suggesting that the underlying relationship between electrical parameters and temperature is not purely linear.
- These results confirm that battery thermal behaviour exhibits nonlinear characteristics, which are better captured by tree-based models. Therefore, the Decision Tree model is the more appropriate and reliable choice for temperature prediction in this dataset.

12 Plots for Multiple Regression Model

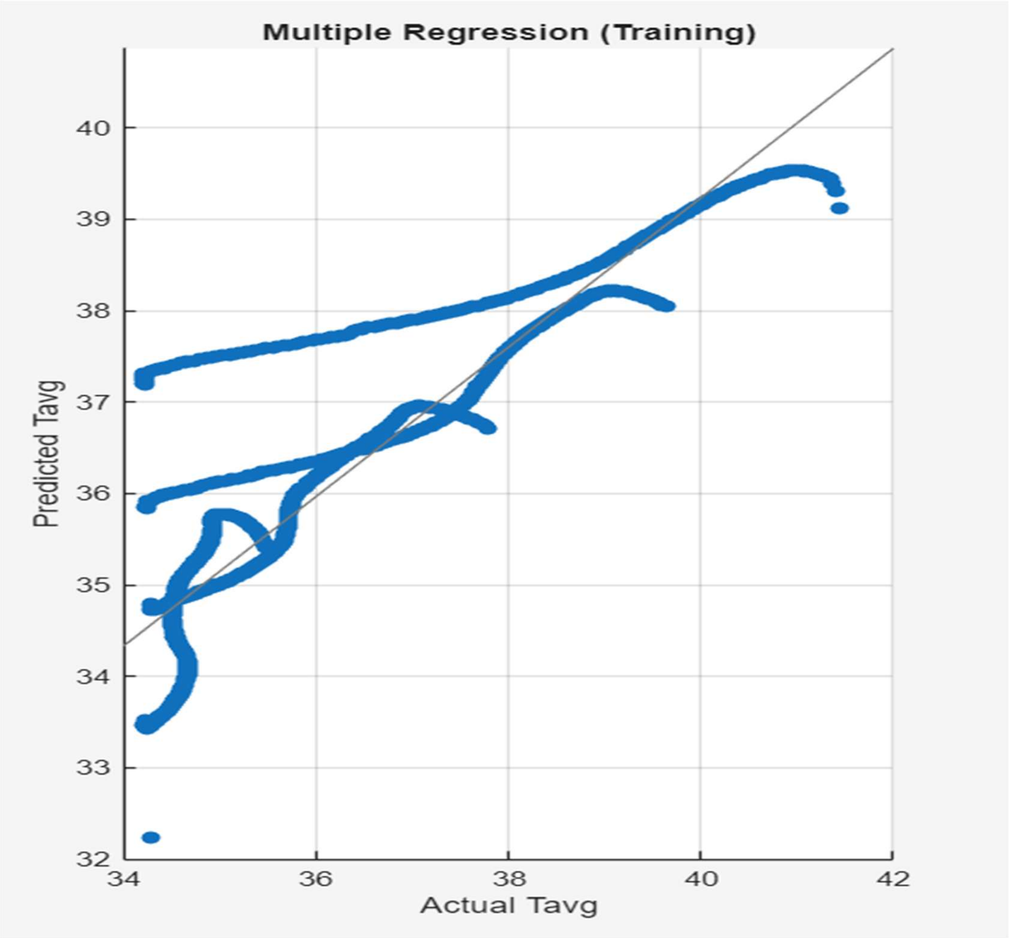


Figure 11: Predicted vs Actual (Training)

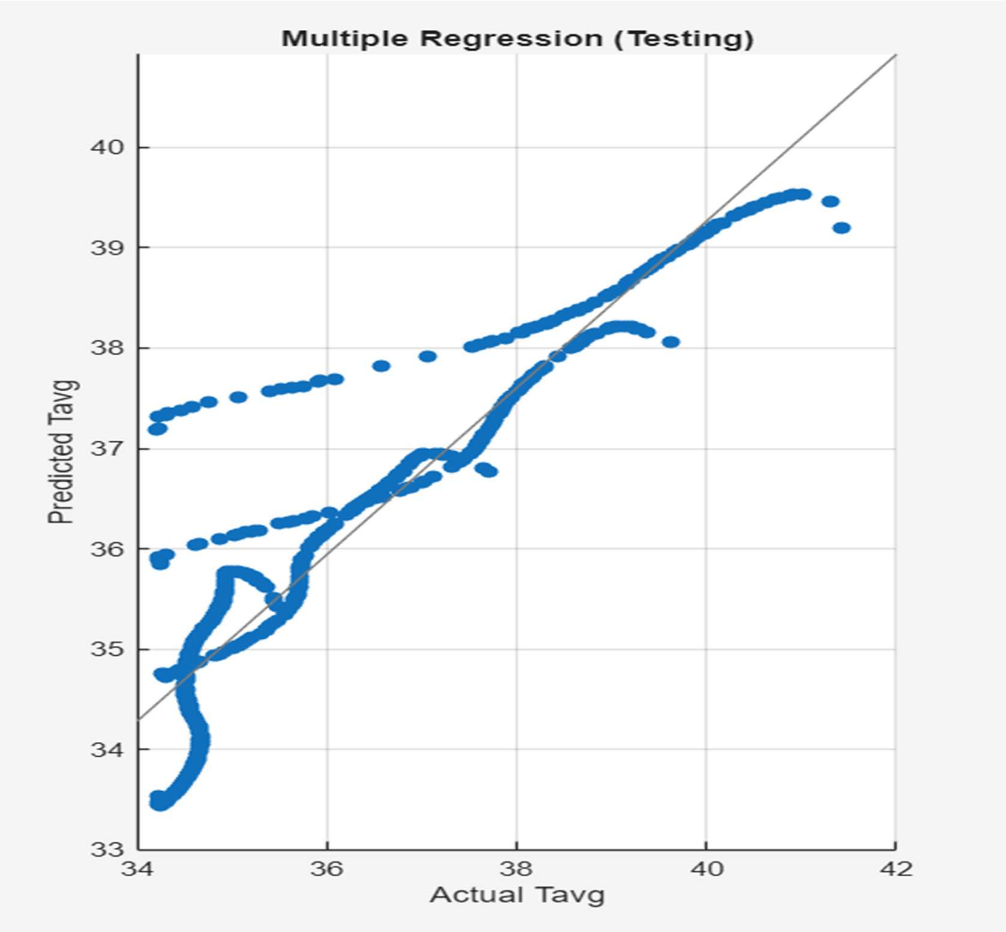


Figure 12: Predicted vs Actual (Testing)

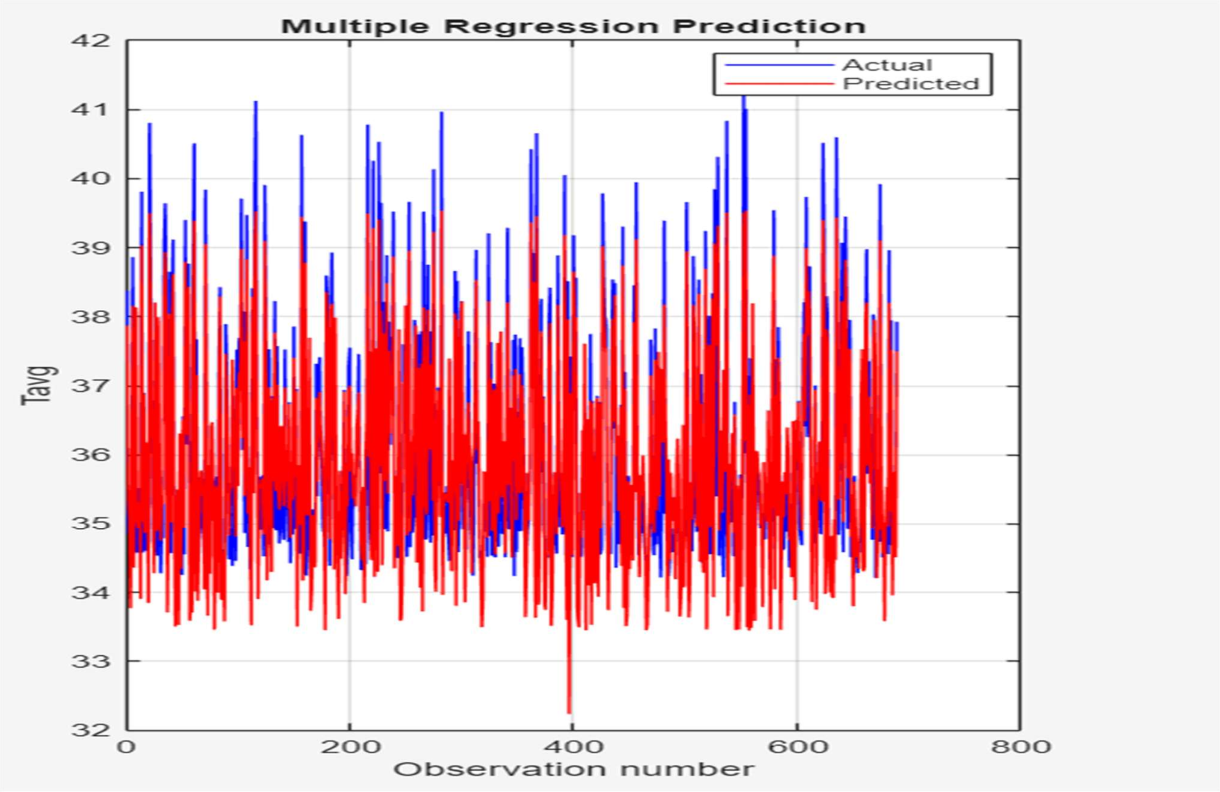


Figure 13: Prediction Plot

12.1 Interpretation of Plots (Multiple Regression)

- From Figure 11 and Figure 12, the scattering of points indicates that the model is not fully capturing the true relationship.
- This means that the relationship between chosen inputs (features) and output (average temperature) is nonlinear.
- In Figure 13, plot shows what is the trend differences between actual and predicted values. This graph proves the issue of underfitting. Hence, the value of R² which was obtained in performance evaluation for multiple linear regression (≈ 0.8), is proven from this plot.
- Basically, graph visually indicates low R² because the model fails to capture variance in the actual data, which is a sign of underfitting.

13 Plots for Decision Tree Model

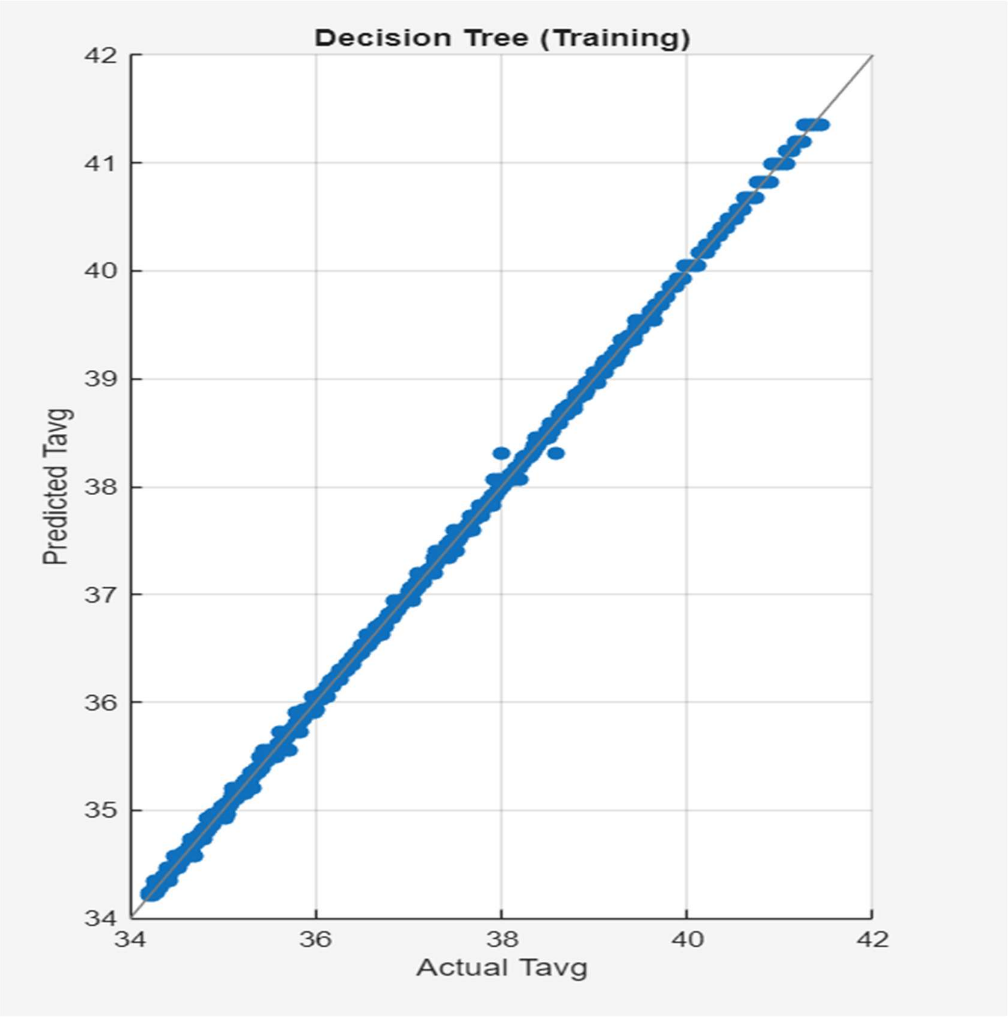


Figure 14: Predicted vs Actual (Training)

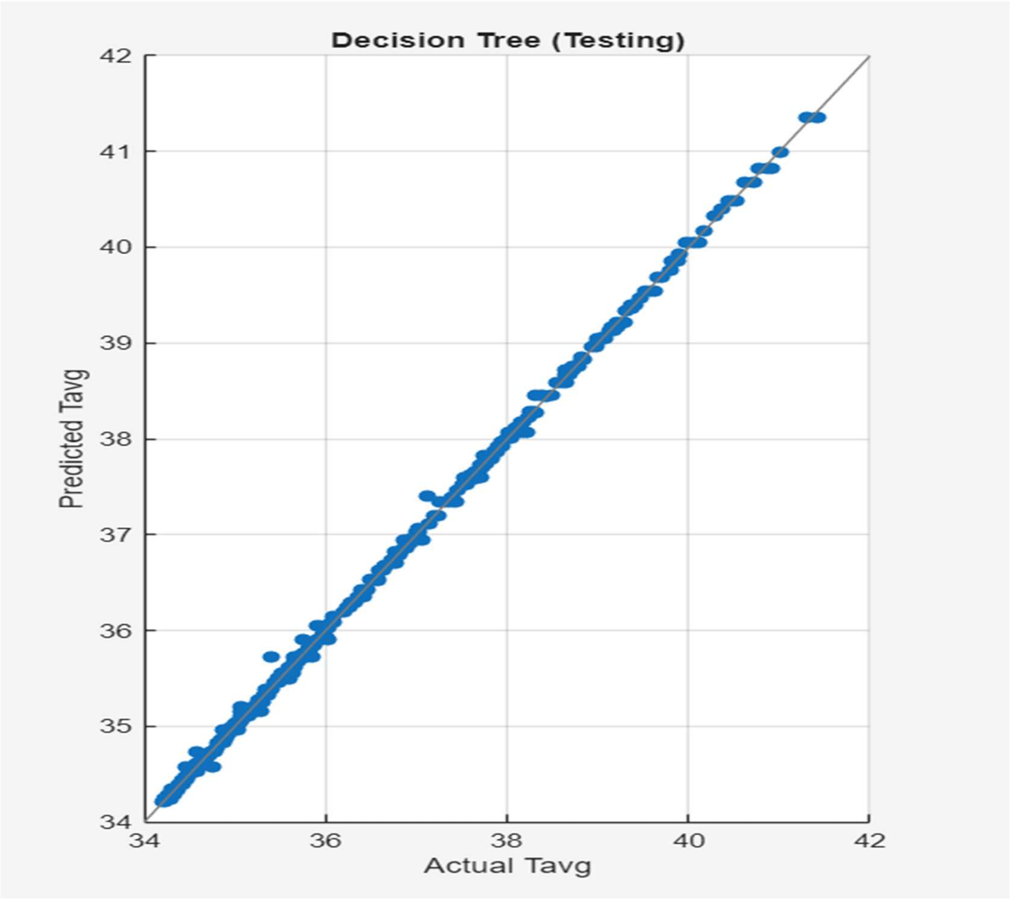


Figure 15: Predicted vs Actual (Testing)

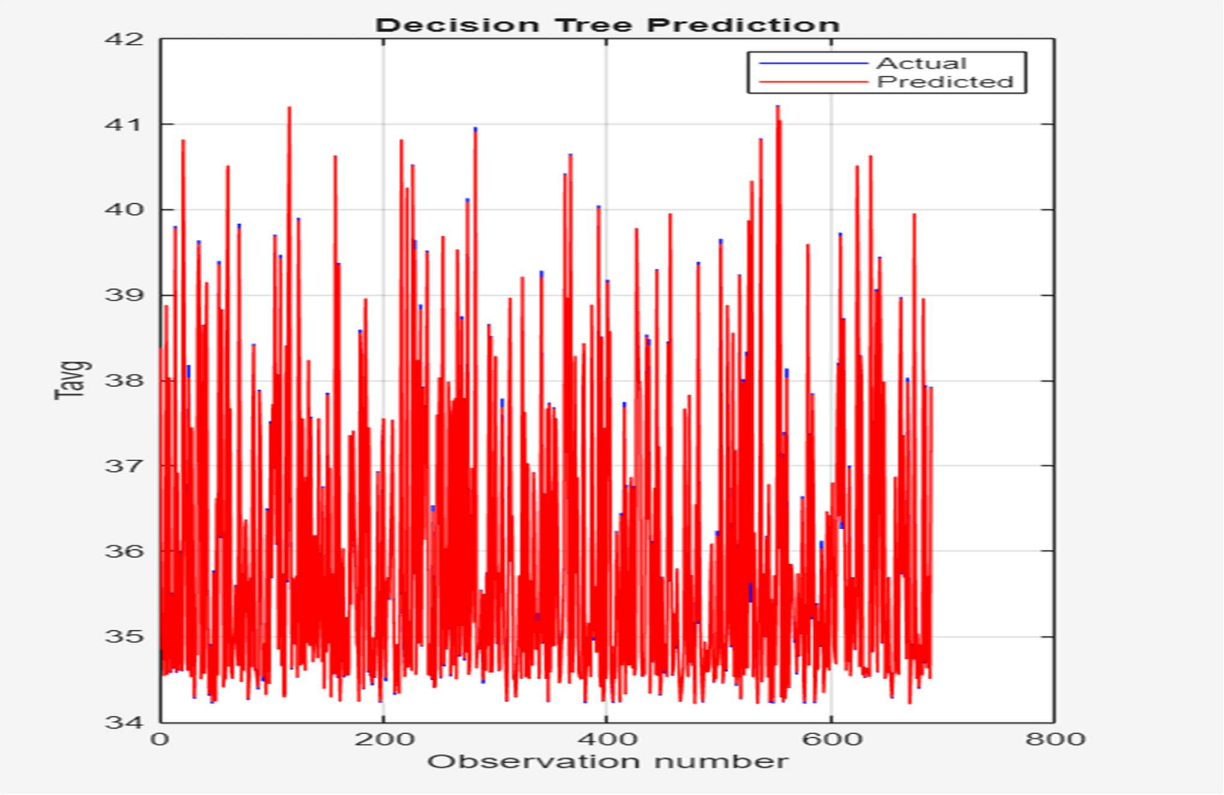


Figure 16: Prediction Plot

13.1 Interpretation of Plots (Decision Tree)

- The model captures both low and high temperature values accurately, unlike multiple regression, showing its ability to model nonlinear relationships. Almost all points lie tightly along the 45° reference line (negligible scattering of points).
- The similarity between training and testing plots confirms strong generalization and no noticeable overfitting, which indicates a well-tuned model.
- From Figure 16, predicted values almost cover the actual values. Hence the R^2 value obtained in performance testing i.e. ≈ 1, is proved from this graph.
- Therefore, the issue of underfitting is also not arising in this model explaining the capability of handling nonlinear relationships.

14 Conclusion

This project applied a structured machine learning workflow to predict battery average temperature using operational and electrical parameters. Exploratory analysis revealed redundant variables, nonlinear relationships, and discrete operating regimes, guiding effective feature selection.

While Multiple Linear Regression provided a baseline, its limited performance highlighted the inadequacy of linear assumptions for battery thermal behavior. The Decision Tree model demonstrated superior accuracy and generalization, successfully capturing nonlinear temperature variations driven by discharge current, voltage, and energy throughput.

The study emphasizes that careful data exploration and correlation analysis are essential before model development, as they directly influence feature selection and model choice. Quantitatively, the Decision Tree model achieved a high coefficient of determination (R^2) with reduced RMSE and MAE compared to Multiple Linear Regression. Overall, it confirms that tree-based approaches are more suitable for modeling complex electro-thermal dynamics in batteries.

15 Appendix

15.1 Appendix A1: Computing Statistical Measures and Plotting Histograms

```
%% Forming the table
clc; clear all; close all;
T = readtable("BC_35.csv");
V = T.Properties.VariableNames;
M = mean(T{:,:});
Md = median(T{:,:});
Sd = std(T{:,:});
Vr =var(T{:,:});
Mn = min(T{:,:});
Mx = max(T{:,:});
Sk = skewness(T{:,:});
Kr = kurtosis(T{:,:});
Full_table = table(V', M',Md', Sd', Vr', Mn', Mx',Sk', Kr','VariableNames',{'Variable','Mean','Median',[" ...
   'Std. Deviation'],'Variance','Minimum','Maximum','Skewness','Kurtosis'});
writetable(Full_table,'fulltable.xlsx');
%% Histogram Plots
dim = [0.8 0.6 0.3 0.3];
figure;
str = {'Mean: 1.9103','Std. Deviation: 1.0513','Minimum: 1','Maximum: 4', 'Skewness: 0.7870', 'Kurtosis: 2.2925'};
histogram(T{:,1},30);
xlabel("Dchg/ChgCycle");
ylabel("Frequency");
annotation('textbox',dim,'String',str,'FitBoxToText','on');
figure;
```

```
str1 = {'Mean: 35','Std. Deviation: 0','Minimum: 35','Maximum: 35', 'Skewness: NaN', 'Kurtosis: NaN'};

histogram(T{:,2},30);

xlabel("BC");

ylabel("Frequency");

annotation('textbox',dim,'String',str1,'FitBoxToText','on');

figure;

str2 = {'Mean: 2.9565','Std. Deviation: 0.2026','Minimum: 1.99','Maximum: 3.53', 'Skewness: -1.2289', 'Kurtosis: 5.3988'};

histogram(T{:,3},30);

xlabel("Vact");

ylabel("Frequency");

annotation('textbox',dim,'String',str2,'FitBoxToText','on');

figure;

str3 = {'Mean: 38.2178','Std. Deviation: 21.0229','Minimum: 0.08','Maximum: 80.05', 'Skewness: 0.7859', 'Kurtosis: 2.2934'};

histogram(T{:,4},30);

xlabel("Discharge Current");

ylabel("Frequency");

annotation('textbox',dim,'String',str3,'FitBoxToText','on');

figure;

str4 = {'Mean: 0.0143','Std. Deviation: 0.0062','Minimum: 0','Maximum: 0.22', 'Skewness: 10.5243', 'Kurtosis: 346.69'};

histogram(T{:,5},30);

xlabel("Charge Current");

ylabel("Frequency");

annotation('textbox',dim,'String',str4,'FitBoxToText','on');

figure;

str5 = {'Mean: 10','Std. Deviation: 0','Minimum: 10','Maximum: 10', 'Skewness: NaN', 'Kurtosis: NaN'};
```

```
histogram(T{:,6},30);

xlabel("Charge/Discharge Cycle");

ylabel("Frequency");

annotation('textbox',dim,'String',str5,'FitBoxToText','on');

figure;

str6 = {'Mean: 9.6842','Std. Deviation: 5.5892','Minimum: 4.4400e-05','Maximum: 19.5018', 'Skewness: 1.6343e-05', 'Kurtosis: 1.8007'};

histogram(T{:,7},30);

xlabel("Ampere hours");

ylabel("Frequency");

annotation('textbox',dim,'String',str6,'FitBoxToText','on');

figure;

str7 = {'Mean: 35.8253','Std. Deviation: 1.6703','Minimum: 34.2055','Maximum: 41.4498', 'Skewness: 1.2989', 'Kurtosis: 3.7602'};

histogram(T{:,8},30);

xlabel("Average Temperature");

ylabel("Frequency");

annotation('textbox',dim,'String',str7,'FitBoxToText','on');
```

15.2 Appendix A2: Plotting the heatmap for Linear and Non-Linear Correlation Analysis

```
clc; clear all; close all;

T = readtable("BC_35.csv");

% Dropping the variables with zero variance

T.BC=[];

T.Chg_DchgCycle = [];

V1 = T.Properties.VariableNames;

% Plotting the heatmap

R1 = corr(T{:,:}, 'Type', 'Pearson');
```

```
figure;

H1 = heatmap(V1,V1,R1);

title("Pearson Correlation Matrix Heatmap");

R2 = corr(T{:,:},'Type','Spearman');

figure;

H2 = heatmap(V1, V1, R2);

title("Spearman Correlation Matrix Heatmap");
```

15.3 Appendix A3: Development of ML models

```
clc; clear; close all;

T = readtable("BC_35.csv");

% Inputs

x1 = T.DischargeCurrent;

x2 = T.Vact;

x3 = T.Ah;

% Output

y = T.Tavg;

%% Remove NaN rows (important)

M = [x1 x2 x3 y];

M = M(all(~isnan(M),2),:);

x1 = M(:,1);

x2 = M(:,2);

x3 = M(:,3);

y  = M(:,4);

%% Multiple Linear Regression

X = [ones(size(x1)) x1 x2 x3];

b = regress(y,X);
```

```
%% 3D Scatter (only x1,x2,y can be visualized)

figure

scatter3(x1,x2,y,'filled')

hold on

%% Create grid

x1fit = linspace(min(x1),max(x1),40);

x2fit = linspace(min(x2),max(x2),40);

[X1FIT,X2FIT] = meshgrid(x1fit,x2fit);

% Fix Ah at rms value for visualization

x3rms= rms(x3);

%% Regression surface

YFIT = b(1)+ b(2).*X1FIT+ b(3).*X2FIT+ b(4).*x3rms.*ones(size(X1FIT));

%% Plot surface

mesh(X1FIT,X2FIT,YFIT)

xlabel('Discharge Current')

ylabel('Vact')

zlabel('Tavg')

title('Multiple Linear Regression: Tavg Prediction')

view(50,10)

grid on

hold off

%%  Decision Tree

I = [x1 x2 x3];

Mdl = fitrtree(I,y);

view(Mdl,'Mode','graph')
```

15.4 Appendix A4: Evaluating the performance metrices of ML models

```
%% Train Test Split (80 / 20)

N = length(y);

idx = randperm(N);

Ntrain = round(0.8*N);

trainIdx = idx(1:Ntrain);

testIdx  = idx(Ntrain+1:end);

x1_tr = x1(trainIdx);  x1_te = x1(testIdx);

x2_tr = x2(trainIdx);  x2_te = x2(testIdx);

x3_tr = x3(trainIdx);  x3_te = x3(testIdx);

y_tr = y(trainIdx);

y_te = y(testIdx);

%% MODEL 1: Multiple Linear Regression

Xtr = [ones(size(x1_tr)) x1_tr x2_tr x3_tr];

Xte = [ones(size(x1_te)) x1_te x2_te x3_te];

b = regress(y_tr,Xtr);

y_pred_lin = Xte*b;

% Training Metrics (Linear Regression)

y_pred_lin_tr = Xtr*b;

RMSE_lin_tr = sqrt(mean((y_tr - y_pred_lin_tr).^2));

MAE_lin_tr  = mean(abs(y_tr - y_pred_lin_tr));

R2_lin_tr   = 1 - sum((y_tr - y_pred_lin_tr).^2)/sum((y_tr-mean(y_tr)).^2);

%% Metrics (Linear Regression)

RMSE_lin = sqrt(mean((y_te - y_pred_lin).^2));

MAE_lin  = mean(abs(y_te - y_pred_lin));

R2_lin   = 1 - sum((y_te - y_pred_lin).^2)/sum((y_te-mean(y_te)).^2);

%% MODEL 2: Decision Tree
```

```
I_tr = [x1_tr x2_tr x3_tr];

I_te = [x1_te x2_te x3_te];

Mdl = fitrtree(I_tr,y_tr);

y_pred_tree = predict(Mdl,I_te);

% Training Metrics (Decision Tree)

y_pred_tree_tr = predict(Mdl,I_tr);

RMSE_tree_tr = sqrt(mean((y_tr - y_pred_tree_tr).^2));

MAE_tree_tr  = mean(abs(y_tr - y_pred_tree_tr));

R2_tree_tr   = 1 - sum((y_tr - y_pred_tree_tr).^2)/sum((y_tr-mean(y_tr)).^2);

%% Metrics (Decision Tree)

RMSE_tree = sqrt(mean((y_te - y_pred_tree).^2));

MAE_tree  = mean(abs(y_te - y_pred_tree));

R2_tree   = 1 - sum((y_te - y_pred_tree).^2)/sum((y_te-mean(y_te)).^2);

%% Display Results

fprintf('\n================================\n');

fprintf('MODEL PERFORMANCE\n');

fprintf('================================\n');

%% Linear Regression

fprintf('\nMultiple Linear Regression:\n');

fprintf('--- Training Set ---\n');

fprintf('R2   = %.4f\n',R2_lin_tr);

fprintf('RMSE = %.4f\n',RMSE_lin_tr);

fprintf('MAE  = %.4f\n',MAE_lin_tr);

fprintf('--- Testing Set ---\n');

fprintf('R2   = %.4f\n',R2_lin);

fprintf('RMSE = %.4f\n',RMSE_lin);

fprintf('MAE  = %.4f\n',MAE_lin);
```

```
%% Decision Tree
fprintf('\nDecision Tree:\n');
fprintf('--- Training Set ---\n');
fprintf('R2   = %.4f\n',R2_tree_tr);
fprintf('RMSE = %.4f\n',RMSE_tree_tr);
fprintf('MAE  = %.4f\n',MAE_tree_tr);
fprintf('--- Testing Set ---\n');
fprintf('R2   = %.4f\n',R2_tree);
fprintf('RMSE = %.4f\n',RMSE_tree);
fprintf('MAE  = %.4f\n',MAE_tree);
%% TASK 7 : Regression & Prediction Plots
% -------- Multiple Regression --------
% Predictions on training data
y_pred_lin_tr = Xtr*b;
figure
subplot(1,3,1)
scatter(y_tr,y_pred_lin_tr,'filled')
lsline
xlabel('Actual Tavg')
ylabel('Predicted Tavg')
title('Multiple Regression (Training)')
grid on
subplot(1,3,2)
scatter(y_te,y_pred_lin,'filled')
lsline
xlabel('Actual Tavg')
ylabel('Predicted Tavg')
```

```
title('Multiple Regression (Testing)')
grid on
subplot(1,3,3)
plot(y_te,'b'); hold on
plot(y_pred_lin,'r')
legend('Actual','Predicted')
xlabel('Observation number')
ylabel('Tavg')
title('Multiple Regression Prediction')
grid on
%% -------- Decision Tree --------
% Predictions on training data
y_pred_tree_tr = predict(Mdl,I_tr);
figure
subplot(1,3,1)
scatter(y_tr,y_pred_tree_tr,'filled')
lsline
xlabel('Actual Tavg')
ylabel('Predicted Tavg')
title('Decision Tree (Training)')
grid on
subplot(1,3,2)
scatter(y_te,y_pred_tree,'filled')
lsline
xlabel('Actual Tavg')
ylabel('Predicted Tavg')
title('Decision Tree (Testing)')
```

```
grid on

subplot(1,3,3)

plot(y_te,'b'); hold on

plot(y_pred_tree,'r')

legend('Actual','Predicted')

xlabel('Observation number')

ylabel('Tavg')

title('Decision Tree Prediction');

grid on;
```