**MLR Model**

% Load the dataset

data = readtable('slab.csv');

% Prepare the dataset

X = data{:, {'Area', 'd', 'fc', 'p', 'E'}};

y = data.Vn;

% Fit a linear regression model

mdl = fitlm(X, y);

% Display the fitted regression model

disp(mdl);

% You can use the model coefficients directly to form your regression equation

coefficients = mdl.Coefficients.Estimate;

fprintf('Vn = %.2f + (%.2f \* Area) + (%.2f \* d) + (%.2f \* fc) + (%.2f \* p) + (%.2f \* E)\n', ...

coefficients(1), coefficients(2), coefficients(3), coefficients(4), coefficients(5), coefficients(6));

ANN model

% Load and prepare your dataset

data = readtable('slab.csv');

inputs = normalize(data{:, {'Area', 'd', 'fc', 'p', 'E'}});

targets = data{:, 'Vn'};

% Split data into training and testing sets

cv = cvpartition(size(inputs, 1), 'HoldOut', 0.3);

idx = cv.test;

XTrain = inputs(~idx, :);

XTest = inputs(idx, :);

yTrain = targets(~idx, :);

yTest = targets(idx, :);

% Define the ANN architecture

layers = [

featureInputLayer(size(XTrain, 2))

fullyConnectedLayer(10)

reluLayer

fullyConnectedLayer(1)

regressionLayer];

% Specify training options

options = trainingOptions('adam', ...

'MaxEpochs', 100, ...

'MiniBatchSize', 20, ...

'InitialLearnRate', 0.01, ...

'Shuffle', 'every-epoch', ...

'ValidationData', {XTest, yTest}, ...

'Plots', 'training-progress', ...

'Verbose', false);

% Train the ANN model

net = trainNetwork(XTrain, yTrain, layers, options);

% Predict Vn using the trained model

yPredicted = predict(net, XTest);

% Assess the model's performance

performance = sqrt(mean((yPredicted - yTest).^2)); % RMSE

disp(['RMSE on test set: ', num2str(performance)]);

% Calculate R^2 value

SSres = sum((yTest - yPredicted).^2);

SStot = sum((yTest - mean(yTest)).^2);

R2 = 1 - SSres/SStot;

disp(['R^2 value: ', num2str(R2)]);

% Generate scatter plot of actual vs. predicted values

figure;

scatter(yTest, yPredicted, 'filled');

hold on;

maxVal = max(max(yTest), max(yPredicted));

plot([0, maxVal], [0, maxVal], 'r', 'LineWidth', 2); % Plotting the line y = x for reference

xlabel('Actual Punching Shear Strength (Vn)');

ylabel('Predicted Punching Shear Strength (Vn)');

title(['Actual vs. Predicted Punching Shear Strength with R^2 = ', num2str(R2)]);

legend('Predicted vs. Actual', 'Perfect Prediction', 'Location', 'Best');

grid on;