close all

clear all

% Neural network for sky darkness level prediction in rural areas

% Load data from Excel file

data = xlsread('DATA Neural network for sky darkness level prediction in rural areas', 'TOTAL', 'A2:C21070');

% Separate columns into independent variables (inputs) and the dependent variable (target)

dia = data(:, 1);

hora = data(:, 2);

oscuridad = data(:, 3);

% Create the input matrix (inputs) and the output matrix (target)

inputs = [dia, hora]';

targets = oscuridad';

% Create a 1-layer hidden neural network with 10 neurons

hiddenLayerSize = 10;

net = feedforwardnet(hiddenLayerSize);

% Divide the data into training and validation sets

net.divideParam.trainRatio = 0.75; % 75% for training set

net.divideParam.valRatio = 0.25; % 25% for validation set

% Train the neural network

[net, tr] = train(net, inputs, targets);

% Evaluate the neural network

outputs = net(inputs);

errors = gsubtract(targets, outputs);

performance = perform(net, targets, outputs);

% Calculate MAPE

mape = mean(abs((targets - outputs) ./ targets)) \* 100;

% Calculate Mean Squared Error (MSE)

mse = mean((targets - outputs).^2);

% Calculate RMSE

rmse = sqrt(mse);

% Calculate the Coefficient of Determination (R^2)

ssres = sum((targets - outputs).^2);

sstot = sum((targets - mean(targets)).^2);

r\_squared = 1 - ssres / sstot;

% Calculate the standard deviation of the errors

std\_errors = std(errors);

% Calculate the Mean Absolute Error (MAE)

mae = mean(abs(errors));

% Display results

disp('Neural network performance:');

disp(performance);

disp(['Standard deviation of errors: ', num2str(std\_errors)]);

disp(['R^2: ', num2str(r\_squared)]);

disp(['MSE: ', num2str(mse)]);

disp(['RMSE: ', num2str(rmse)]);

disp(['MAE: ', num2str(mae)]);

disp(['MAPE: ', num2str(mape), '%']);

% Plot the actual and predicted values for all data

figure;

plot(targets, 'k', 'DisplayName', 'Real values');

hold on;

plot(outputs, 'r', 'DisplayName','Predicted values');

xlim([0, 21069]);

ylim([10, 24]);

xlabel('Sample index','FontSize', 22);

ylabel('NSB','FontSize', 22);

legend;

hold off;

% Plot last 20 actual and predicted values

figure;

plot(targets(end-19:end), 'k', 'DisplayName', 'Real Values');

hold on;

plot(outputs(end-19:end), 'r', 'DisplayName', 'Predicted Values');

xlabel('Sample index','FontSize', 22);

ylabel('NSB','FontSize', 22);

legend;

hold off;

% Generate predictions for each day of the year for hours 0, 1, 2, 3, 4, 5, 6

horas = 0:6;

predicciones = zeros(length(horas), 365);

for h = 1:length(horas)

for d = 1:365

nuevos\_datos = [d; horas(h)];

predicciones(h, d) = round(net(nuevos\_datos), 2);

end

end

%Plot the predictions for each day of the year at different times of the day

figure;

hold on;

colors = {'b', 'g', 'r', 'c', 'm', '#A2142F', 'k'};

for h = 1:length(horas)

plot(1:365, predicciones(h, :), 'DisplayName', ['Hour ', num2str(horas(h))], 'Color', colors{h});

end

xlabel('Day of the year', 'FontSize', 22);

ylabel('Predicted NSB', 'FontSize', 22);

legend;

xlim([1 365]);

hold off;

% Generate the measurements for each day of the year for the hours 0, 1, 2, 3, 4, 5, 6

horas = 0:6;

mediciones = zeros(length(horas), 365);

for h = 1:length(horas)

for d = 1:365

% Find the value of darkness corresponding to the day and time

indices = (dia == d & hora == horas(h));

mediciones(h, d) = mean(oscuridad(indices)); % Average if more than one measurement

end

end

% Prediction with new data:

% Request new\_day and new\_time values via a dialog box

prompt = {'Enter the day (1-365):', 'Enter the hour in fractions of 0.25 (0-6.75):'};

dlgtitle = 'Prediction Data';

dims = [1 35];

definput = {'1', '0'};

answer = inputdlg(prompt, dlgtitle, dims, definput);

% Convert the answers of the dialogue window to numbers

nuevo\_dia = str2double(answer{1});

nueva\_hora = str2double(answer{2});

% Check if the entered values are valid

if isnan(nuevo\_dia) || isnan(nueva\_hora) || nuevo\_dia < 1 || nuevo\_dia > 365 || nueva\_hora < 0 || nueva\_hora > 6.75

error('Invalid values entered. Please enter a day between 1 and 365, and an hour between 0 and 6.75.');

end

% Search for the darkness value corresponding to the day and time given

indice = find(dia == nuevo\_dia & hora == nueva\_hora);

% If the exact value is not found, search for the nearest value

if isempty(indice)

all\_points = [dia, hora];

nuevo\_punto = [nuevo\_dia, nueva\_hora];

[idx, ~] = knnsearch(all\_points, nuevo\_punto);

nuevo\_dia = dia(idx);

nueva\_hora = hora(idx);

nuevo\_valor\_oscuridad = oscuridad(idx);

else

nuevo\_valor\_oscuridad = oscuridad(indice);

end

% Prepare data for prediction

nuevos\_datos = [nuevo\_dia; nueva\_hora];

% Make the prediction

prediccion = round(net(nuevos\_datos), 2);

disp('Prediction for new data:');

disp(prediccion);

% Display the prediction in a dialogue window %

msgbox(['Forecast for the day ', num2str(nuevo\_dia), ' and hour ', num2str(nueva\_hora), ': ', num2str(prediccion)], 'NSB Prediction');