CN7050- Intelligent Systems Module

Week-02 Tutorial

This tutorial does not carry any marks toward the assessment component.

Learning Objectives: By the end of this lab, students will be able to:

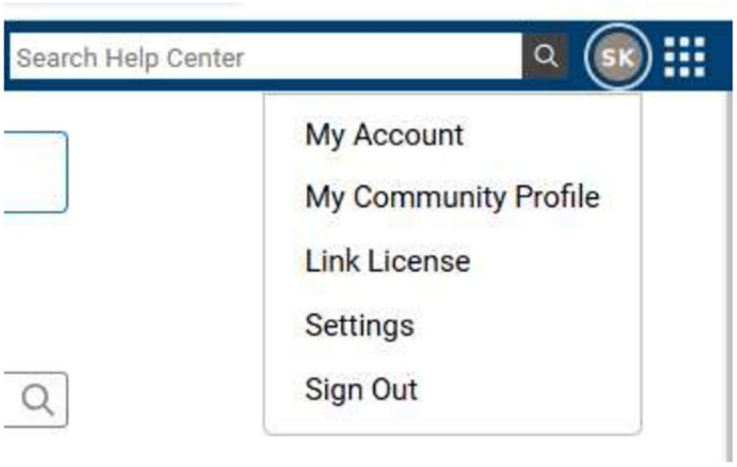
- Understand the structure of expert systems (knowledge base + inference engine).
- Design simple rule-based decision systems in MATLAB.
- Use if-else rules to simulate medical diagnosis or decision support logic.
- Visualize how kNN classifies new inputs based on their similarity to training examples, and trace predictions by examining nearest neighbors in the feature space.

Task 1: Build a Rule-Based Expert System in MATLAB

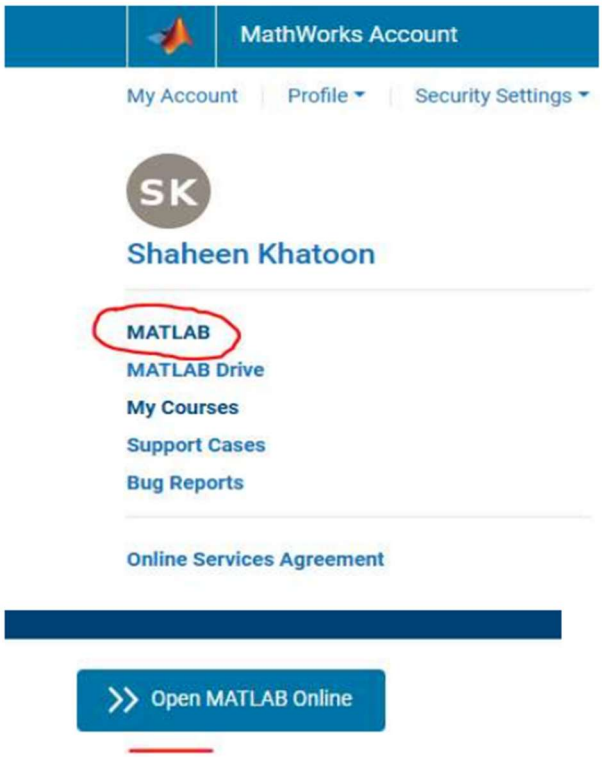
- Let's create a simple **medical diagnosis system** that gives a likely condition based on user inputs.

Open MATLAB Online

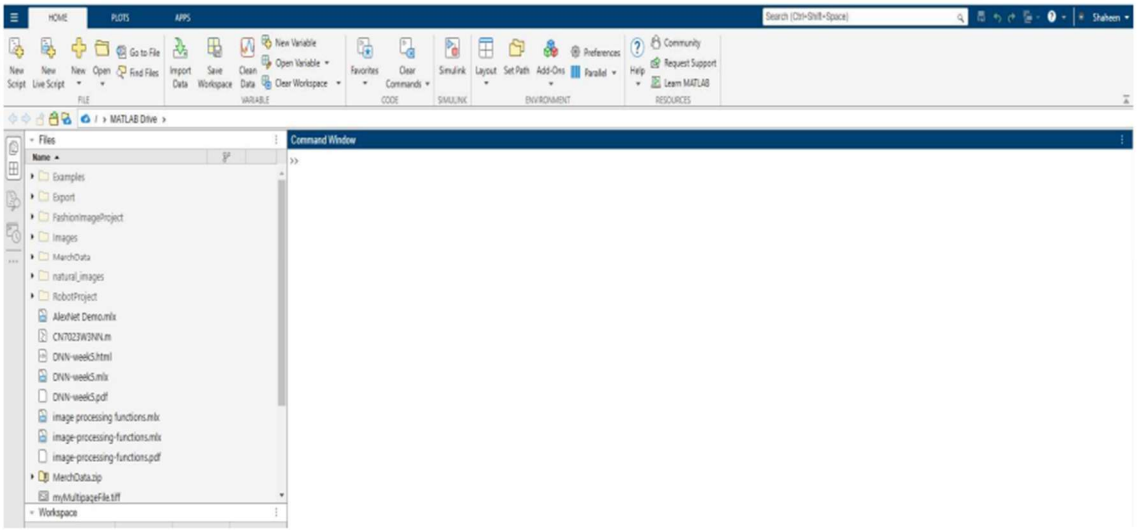
1. Click on: https://uk.mathworks.com/academia/tah-portal/university-of-east-london-31543424.html
2. Sign in, click on your account initials and then click **My Account**



3. Click on **MATLAB** and then click **Open MATLAB Online**

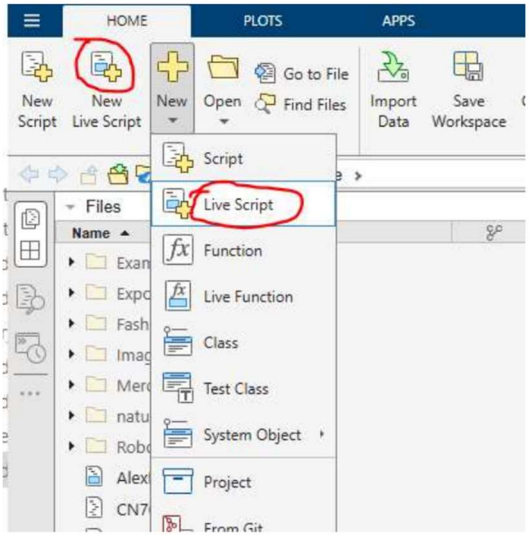


4. MATLAB Online editor will open.

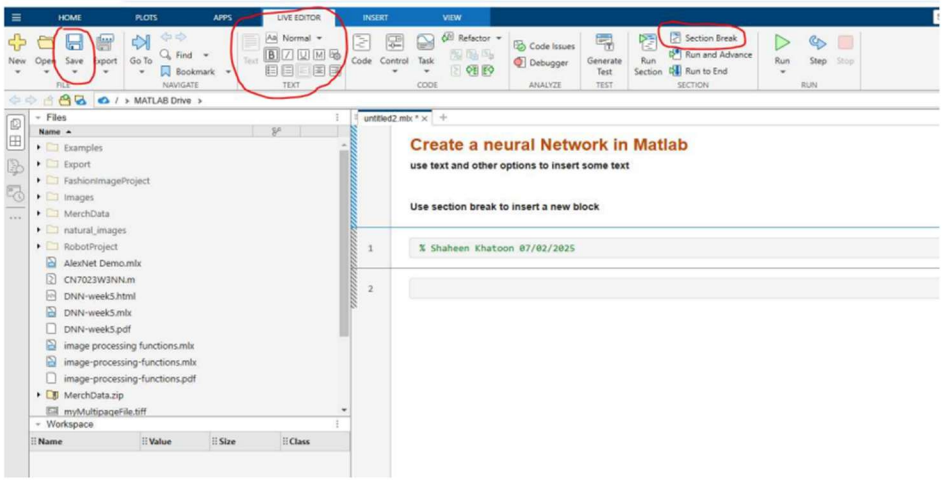


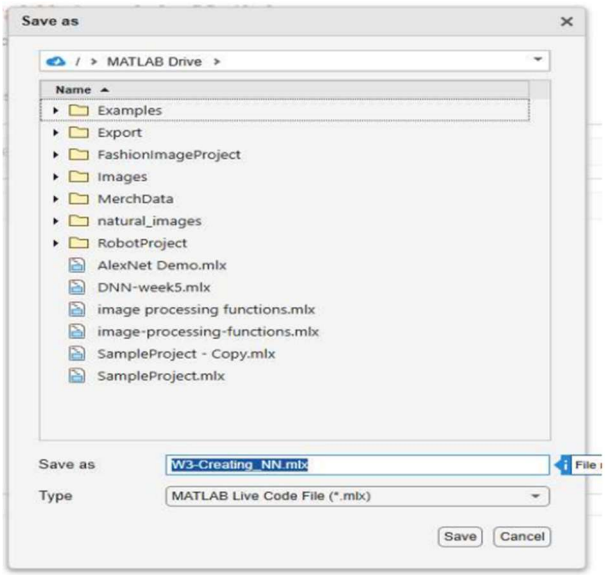
Implement a Reactive Agent in MATLAB

1. **Create a new MATLAB Live script**



2. Add your **name, date, and purpose** of the code at the top as a comment. Save the file.





3. Add the following section codes step-by-step and run each to observe outputs.

4. **Clear the environment:**

```
%Close all open figures
close all
%Clear the workspace
clear
%Clear the command window
clc
```

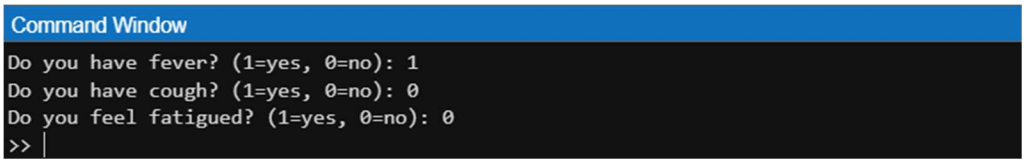
5. **Add the following code in the next section**

```
% Sample symptoms
has_fever = input('Do you have fever? (1=yes, 0=no): ');
has_cough = input('Do you have cough? (1=yes, 0=no): ');
has_fatigue = input('Do you feel fatigued? (1=yes, 0=no): ');

% Rule evaluation
if has_fever && has_cough && has_fatigue
    disp('Diagnosis: You may have FLU');
elseif has_fever && ~has_cough
    disp('Diagnosis: You may have a viral infection');
elseif has_cough && ~has_fever
    disp('Diagnosis: Likely cold or allergy');
else
    disp('Diagnosis: Symptoms not conclusive');
end
```

6. Run the code by pressing the **Run Section** button on the toolbar.

You will see the questions appearing at the command prompt; See the next image for a sample of provided answers. (Answer the questions by pressing a 0 or 1).



7. After answering all the three questions about the symptoms of the disease, a diagnosis will appear as shown in the next image. (You can experiment with different answers by running the section again).

Task 2: Use MATLAB KNN

Use built-in ML methods to **train a knn model** from data.

We can also achieve the similar results using a machine learning k nearest neighbor or similar algorithm to train a model and predict the diagnosis depending upon the provided symptom values. Copy the below given code in matlab to train a model and show predictions for different symptoms.

```
%% Symptom Diagnosis using kNN

% Features: [fever, cough, fatigue]
X = [
    0 0 0;   % Unknown
    1 1 1;   % Flu
    1 1 0;   % Flu
    1 0 1;   % Viral
    1 0 0;   % Viral
    0 1 1;   % Cold
    0 1 0;   % Cold
    0 0 1;   % Unknown
    0 0 0;   % Unknown
];

% Labels
Y = {'Unknown'; 'Flu'; 'Flu'; 'Viral'; 'Viral'; 'Cold'; 'Cold'; 'Unknown'; 'Unknown'};

%% Train kNN classifier
knn = fitcknn(X, Y, 'NumNeighbors', 3);

%% Test different patient inputs
tests = [
    1 1 1;   % High fever + cough + fatigue
    1 0 0;   % Fever only
    0 1 0;   % Cough only
    0 0 1;   % Fatigue only
    0 0 0;   % No symptoms
];

predictions = predict(knn, tests);

%% Display results
disp('Test Inputs and Predictions:')
disp(table(tests(:,1), tests(:,2), tests(:,3), predictions, ...
    'VariableNames', {'Fever','Cough','Fatigue','Predicted_Disease'}));
```

Once you will run this section, you will see the below given result as output. These are the predictions for the test symptoms that we have provided.

```
Test Inputs and Predictions:
    Fever    Cough    Fatigue    Predicted_Disease
    _____    _____    _______    _________________

      1        1         1          {'Flu'    }
      1        0         0          {'Flu'    }
      0        1         0          {'Cold'   }
      0        0         1          {'Unknown'}
      0        0         0          {'Unknown'}
```

Note that model sometimes can give us the wrong input, as it uses the data to learn and infer the rules automatically, whilst if we use the code and write the rules ourself that is more deterministic, however; not feasible if we have a larger dataset.

How to get help

- Please speak with your Practical Group tutor if you have any questions during the practical session.
- If you have any questions about your studies, not relating to this course, please contact your Year Tutor