MAIS 202 - PROJECT DELIVERABLE 2

1. Problem statement:

As stated in the first deliverable, our project's goal is to create a model capable of assisting healthcare professionals in diagnosing patients. With this project, we will thus try to answer the problem: How to use AI as a tool to help gain time and disencumber hospitals through a more efficient diagnosis? To do so, while trying to answer potential ethical dilemmas, the model will mostly assist by sorting symptoms based on its predicted disease, and potentially propose several options of diseases.

2. Data Preprocessing:

We continued our project with the same initial dataset we said that we would use in the first deliverable, which we had found on Kaggle. We wrote a script called "dataset" that can take the csv file containing our dataset and format it such that it can be fed to our model. The dataset has 4921 samples and 41 unique diseases. There are 17 input features, being the name of the disease and the associated symptoms (not all diseases have 16 symptoms, some of the inputs are empty). We did not modify or delete any data.

3. Machine Learning Model:

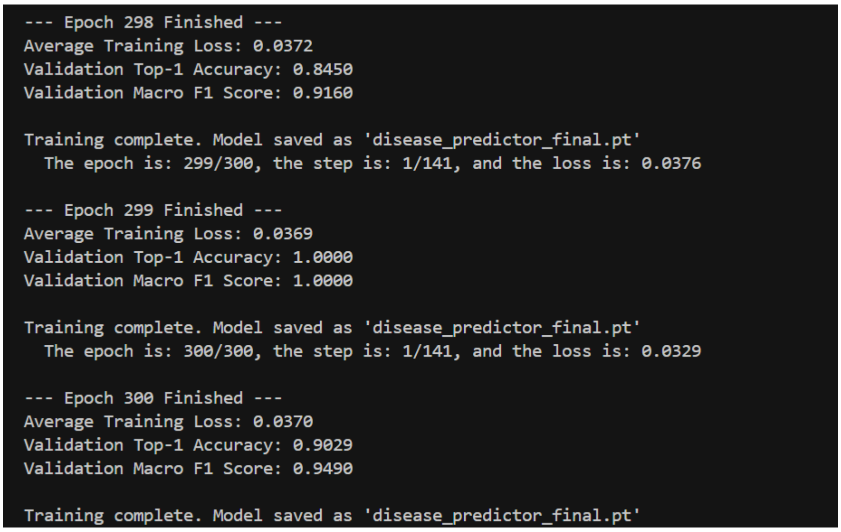
a. Specify the framework and tools that you used to implement your model.

Initially, we wanted to use a transformer, but we still use a neural network. We use PyTorch extensively, including its neural network features. We flow through 10 features, expand to 128 to learn complex patterns for better guesses and we shrink back to 41 to get disease predictions. Then we return the raw prediction scores (logits).

b. Justify any decision about training/validation/test splits, regularization techniques, optimization tricks, setting hyper-parameters, etc.

We wrote a script to train our model. We used 90% of the data to train and 10% for validation (data is picked randomly each time). To improve efficiency (optimization trick?) we use a batch size of 16. We calculate and minimize the loss function and update the weights for each batch.

Example of the training:



```
--- Epoch 298 Finished ---
Average Training Loss: 0.0372
Validation Top-1 Accuracy: 0.8450
Validation Macro F1 Score: 0.9160

Training complete. Model saved as 'disease_predictor_final.pt'
  The epoch is: 299/300, the step is: 1/141, and the loss is: 0.0376

--- Epoch 299 Finished ---
Average Training Loss: 0.0369
Validation Top-1 Accuracy: 1.0000
Validation Macro F1 Score: 1.0000

Training complete. Model saved as 'disease_predictor_final.pt'
  The epoch is: 300/300, the step is: 1/141, and the loss is: 0.0329

--- Epoch 300 Finished ---
Average Training Loss: 0.0370
Validation Top-1 Accuracy: 0.9029
Validation Macro F1 Score: 0.9490

Training complete. Model saved as 'disease_predictor_final.pt'
```

c. Description of validation methods

We tested accuracy and the f1 score. Our model is overfitting. We used a dropout rate of 0.01 to reduce overfitting.

d. Did you face any challenges implementing the model? If so, how did you solve it?

The biggest hurdle was ensuring that the model could be trained in order to actually recognize complex patterns, rather than making direct 1-to-1 correlations (limitation related to linear regression). We solved this by adding a hidden 128 neurons layer combined with the ReLu function (to ensure non-linearity), enabling the neural network to better find the complicated decision boundaries that classify the 41 diseases.

4. Preliminary Results:

We tested for accuracy and precision. We have high accuracy scores. We have lower precision, but it is acceptable given that we have a multilabel process. The guess with the highest probability is often correct, leading to a high accuracy, but since we have many other guesses, with lower probabilities than the correct guess, that reduces the precision score.

5. Next steps:

The next steps are to implement the text generation, write the website interface and to fine tune the model. Given our metrics, we think that our model works. The cons are that the architecture that we are using is complex and computationally heavy. However, in the health sector, high precision is required, which justifies the use of our algorithm.