**Covid Case Prediction using Artificial Neural Networks**

# Objective:

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people infected with COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

During the entire course of the pandemic, one of the main problems that healthcare providers have faced is the shortage of medical resources and a proper plan to efficiently distribute them. In these tough times, being able to predict what kind of resource an individual might require after being tested positive or even before that, will be of immense help to the authorities as they would be able to procure and arrange for the resources necessary for that patient.

The main goal of this project is to build a machine learning model that, given a Covid-19 patient's current symptom, status, and medical history, will predict whether the patient is at high risk or not. The dataset was provided by the Mexican government. This dataset contains an enormous number of anonymized patient-related information including pre-conditions. In this assignment, you will implement some simple multi-layer perceptrons and neural networks. In particular, you shall be doing the following tasks.

# Tasks to be done:

1. Starter code provided does this using python, can be used: Randomly divide the dataset into 80% training set and the rest as test set. Choose the important features from the dataset by modifying relevant parts of the starter code. Choose a mini-batch size to divide the dataset into batches.
2. Build the ANN model. These operations have been demonstrated in the starter code using Pytorch.
   1. Build the MLP classifiers by identifying the number of input and output nodes required for the problem, and specifying the number of hidden layers as:
      1. 0 hidden layers
      2. 1 hidden layer with 32 nodes
      3. 1 hidden layer with 64 nodes
   2. Use Sigmoid or ReLU activation function for the input and hidden layers. Use ReLU activation for the output layer.
   3. Define the forward and backward operations for your network. They are required for inference and weight updation of your model.
   4. Define the training function to train the model using a forward and a backward pass. Define the prediction function for obtaining the outputs from the network.
   5. Compare the implementation of your model compared to that using the Pytorch library, on the same dataset (code snippet provided).
3. Hyper-parameter tuning.
   1. For each of the architectures, vary the learning rates in the order of 0.1, 0.01, 0.001, 0.0001, 0.00001. Plot graph for the results with respect to accuracy and loss. (Learning rate vs accuracy/loss for each model).
   2. Report test set accuracy for all the learning rates in a tabular form and identify the best model.
4. Classification Report
   1. Create a classification report for comparing the performance of your algorithm, for your best performing algorithm in terms of accuracy, with that of the Pytorch algorithm.
   2. You need to calculate precision, recall, f1-score and accuracy of the model. You can use the sklearn code provided in the code snippet for this.
5. You can use any number of training epochs. Any additional analysis or findings from the dataset, as a result of your task, is well appreciated.

***Note:*** *The program can be written in Python programming language from scratch. No machine learning /data science /statistics package / library should be used for model creation.*

# Relevant information:

*Source: datos.gob.mx/busca/dataset/informacion-referente-a-casos-covid-19-en-mexico*

Dataset Filename: covid\_small.csv

Data Description:

Number of Instances: approx. 1,00,000 (extracted randomly from a raw dataset) Number of Attributes: 20 + class feature = 21

Values 1-3 mean that the patient was diagnosed with covid in different degrees. 4 or higher means that the patient is not a carrier of covid or that the test is inconclusive. In the Boolean features, 1 means "yes" and 2 means "no". values as 97 and 99 are missing data. The preprocessing from the sample code to handle these cases can be reused.

**Submission Details:**

ZIPPED folder containing code (with comments) and the dataset files

1. Report (in pdf format)

# Submission Guidelines:

1. You may use one of the following languages: Python.
2. Your Programs should run on a Linux Environment.
3. You are **not** allowed to use any library apart from these (Also explore all these libraries if doing in Python, or equivalent of these):

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv) from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score from sklearn.metrics import classification\_report import operator

from math import log

from collections import Counter from statistics import mean

Your program should be standalone and should **not** use any *special purpose* library for Machine Learning. Numpy and Pandas may be used. And, you can use libraries for other purposes, such as generation and formatting of data.

1. You should submit the program file and README file and not the output/input file.
2. You should name your file as <GroupNo\_ProjectCode.extension> (e.g., Group1\_WONN.pdf or Group1\_WONN.zip).
3. The submitted program file *should* have the following header comments: # Group Number

# Roll Numbers : Names of members (listed line wise) # Project Number

# Project Title

1. Submit through CSE-MOODLE only.

***You should not use any code available on the Web. Submissions found to be plagiarized or having used ML libraries (except for parts where specifically allowed) will be awarded zero marks.***