University No:_____

THE UNIVERSITY OF HONG KONG
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE
COMP7409 Machine Learning in Trading and Finance

Date: May 16, 2023 Time: 6:30pm-8:30pm

INSTRUCTIONS:

a. This paper has 4 questions. Answer ALL questions.
b. Total mark is 100.
c. Write your university number clearly at the beginning of your answer script. DO NOT write down your name.
d. Only approved calculators as announced by the Examinations Secretary can be used in the examination. It is the candidates' responsibility to ensure that their calculator operates satisfactorily, and candidates must record the name and type of the calculator used on the front page of your answer script.

University No:_____

1. (25%) We have the file "tranrecords.cvs" storing payment records for various departments in our company. There are some entries missing in the files, and we put a character ":" for each of these missing entries. We show the first few lines in this file below:

```
1599.5483533432368,C560169,Paid_UnReconciled,Finance,2015,Aug,1
1.1614787867721998,C817105,Paid_UnReconciled,Purchasing,2018,Aug,0
11.885565040510526,C476755,Paid_Reconciled,Finance,2017,Apr,0
13.086409040014,E335726,Paid_Reconciled,R&D,2011,Aug,0
21.369041770353846,A818773,Paid_UnReconciled,R&D,2015,Jun,0
0.31063476110127564,C141146,Paid_UnReconciled,R&D,2018,Jan,0
3.3556584009813974,E380954,Paid_Reconciled,Production,2012,Apr,0
2.140470211954084,E453358,Paid_Reconciled,Finance,2017,Aug,0
22.156267234847004,C641107,Paid_Reconciled,Finance,2013,Apr,0
0.791029696416332,:,Paid_UnReconciled,Purchasing,2016,Sep,0
151.3348287194061,C805852,Paid_UnReconciled,Marketing,2014,Aug,1
1.0224137650926193,E977677,Paid_Reconciled,R&D,2015,Nov,0
0.040415567013301,C320147,Paid_Reconciled,Purchasing,2014,Sep,0
```

and the name of the columns are

```
 #   Column
---  ------
 0   Amount
 1   TranNo
 2   Status
 3   Department
 4   Fiscal Year
 5   Month
 6   RedFlag
```

For example, the second line in tranrecords.cvs is for the transcaction with Amount=1.161... thousand, Transno=C817105, Status=Paid_unReconciled, Departmnet=Purchasing, Fiscal year=2018, Month=August, and RedFlag=0. In particular, for any transaction, if it is a fraud transaction, its RedFlag is set to 1; otherwise, its RedFlag is set to 0.

You are asked to design a system that applies machine learning techniques to predict fraud transactions based on the file tranrecords.cvs. You should describe all the steps needed to implement such a system. For example, you may explain how to clean up the data, how to perform Exploratory Data Analysis to pick important columns for the training, what machine learning model you will use, how to prepare the training and testing data, how to do the training and testing, and how to evaluate the performance of the system. This is an open question, and you may put down any step that you think appropriate. You don't need to give a complete program for the system, but you may include some Python code fragments if they help your explanation.

University No:_____

2. (a) (13%) A *contiguous subsequence* of a list S of positive numbers is a subsequence made up of consecutive elements of S. For instance, if S = 0.5, 1.5, 30, 10, 5, 0.4, 10, then 1.5, 30, 10 is a contiguous subsequence but 0.5, 10, 0.4 is not. Consider a list of positive numbers $a_1, a_2, \ldots, a_n$. For any $1 \le i \le j \le n$, define $prod(a_i, a_{i+1}, \ldots, a_j) = a_i \times a_{i+1} \times \ldots \times a_j$ to be the product of the numbers in the contiguous subsequence $a_i, a_{i+1}, \ldots, a_j$. Define $f(j) = \max_{1 \le i \le j} (prod(a_1, a_{i+1}, \ldots, a_j))$ to be the largest $prod(a_i, a_{i+1}, \ldots, a_j)$ among all the contiguous subsequences that end at a_j. This question asks you to describe how to find the values of f(1), f(2),..., f(n) efficiently by applying Bellman's principle of optimality, and then give a Python program that applies your idea to find the n values.
Hint: Recall that Bellman's principle of optimality is stated as follows:

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

To apply the principle, you may focus on determining how to express f(j) by other appropriate f(i)'s.

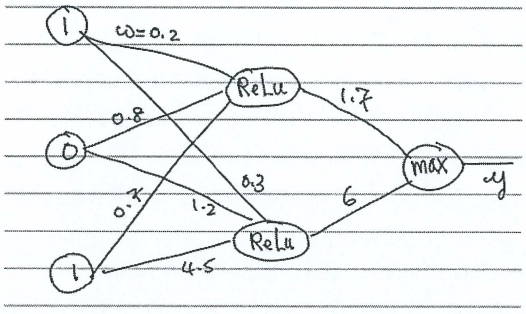
(b) (12%) We have studied the following program for playing the cart-pole game in our lecture:

```
!pip install gym


import numpy as np
import pandas as pd
import random
from pylab import plt, mpl
import time


import gym
env = gym.make('CartPole-v0')


env.seed(100)
env.action_space.seed(100)
random.seed(100)
np.random.seed(100)

 def run_one_episode(env):
   state = env.reset()
   for step in range(200):
     a = random.randint(0,1)
     state, reward, done, info = env.step(a)
     if done:
       break
     #env.render()
     #time.sleep(0.05)
     print(f'step={step:2d} | state={state} | action={a} | reward={reward}')
   return step
```

The performance of the program is poor because it takes a random move for every step. Try to improve the performance by using Q-learning.

University No:_____

3. (a)(7%) Determine the output of the following simple neural network (i.e., determine the value of y).



(b)(18%) Recall that the variance of a sequence of numbers is used to measure the variability of the numbers. We can extend the definition to measure the variability of two sequences numbers $x_1, x_2, \cdots, x_n$ and $y_1, y_2, \cdots, y_n$ by defining the *covariance* of these two sequences as follows:

$$((x_1 - \bar{x})(y_1 - \bar{y}) + (x_2 - \bar{x})(y_2 - \bar{y}) + \cdots + (x_n - \bar{x})(y_n - \bar{y}))/n$$

where $\bar{x} = (x_1 + x_2 + \cdots + x_n)/n$ and $\bar{y} = (y_1 + y_2 + \cdots + y_n)/n$ are the average of the x_i's. and the y_i's, respectively. Intuitively, if the covariance of the two sequences is (significantly) greater than zero, they are positively related, i.e., the greater values of one variable mainly corresponding with the greater values of the other values, and if the covariance is (significantly) smaller than zero, they are negatively related. If their covariance is zero, the two sequences do not have relation.
This question asks you to show that the above intuition is not always true by completing the following two sequences x = [-3, -2, -1, 0, ...] and y = [3, 2, 1, 0, ...] such that (i) the two sequences are closely related, i.e., given a value of x, we can determine the corresponding value of y correctly, and (ii) the covariance of the two sequences is zero. Justify your answer.

4. (25%) Compare the popularity of "Lenovo" and "Dell" by writing a python program that computes, for everyday between May 7, 2023 to May 16, 2023, the average sentiment scores of related tweets posted on twitters on that day using the sentiment analysis tool "flair". You may assume the Bearer Token for accessing the tweets is stored at the first line of the file "BearerToken.txt". Your program should include all necessary libraries.

END OF PAPER