First name:	**Noma:**
Family name:	

LELEC2870 : Machine Learning – January 2022 exam

PLEASE READ THIS FIRST!

Organization of the exam:

- The exam duration is 3 hours, starting from 2pm. If you have finished you are allowed to leave the room between 2:30 p.m. and 4:45 p.m. Those who haven't left by 4:45 p.m. will be asked to stay until 5 p.m.
- At the end of the exam (5 p.m.) we will ask you
  1. to stop writing (all of you at the same time)
  2. to move row by row, keeping 1m50 distance, and to come to the front of the room to give your documents. You have to hand in **all sheets**:
     - Including this one, but not any other one (no supplementary sheet, draft sheet, etc.)
     - Arranged in the **correct order** (page 1 first on top, then page 2,…)
     - With your first name, family name and NOMA indicated on **every sheet.**
- For sanitary reasons we are not allowed to come close to you during the exam to answer to some questions. We are thus sorry that will not be able to answer any question you may have. An exception will be made during the first 15 minutes of the exam: you will have the possibility to ask a question from your seat, while everybody in the room can listen.

Instructions:

- Write **your answers only in the frames**. This gives a good indication of the expected length of your answer. You don't have to use small handwriting to write more! Please do not write anything outside the frames, it will not be considered.
- The exam is open book; this means that **only written notes** are allowed; any electronic device with or without connection to the internet is not allowed.
- Fill in your **first name, name** and **NOMA** on **every page.** Write clearly in capital letters, as this will be interpreted by a computer.

Rating system:

- The project counts for 10 points on 20, including part B of this exam (the part related to the project). Part B of this exam won't alter your project points by more than 2/10. Please prioritize Part A!
- Part A counts for 10 points on 20.

First name: **Noma:**

Family name:

Part A. Questions on the Course

Question 1 – principal component analysis (PCA)

A five-dimensional data **X** set is processed with principal component analysis (PCA) in Python (rows are observations, columns are variables).
The covariance matrix is computed and decomposed into eigenvalues and column eigenvectors. All the matrices are represented just below.

```
Data
[[ 0.33409696 -0.25328959  0.31648363 -0.02542596  0.57660034]
 [-0.42498507 -0.51588714 -0.57047672 -0.47003082 -0.59297642]
 [-0.27384601 -0.42644977 -0.19760956 -0.53275262 -0.28029364]
 [-0.38534929 -0.29596003 -0.45450851 -0.4836354  -0.44260005]
 [ 1.01009163  1.33640415  0.31103699  0.65577541  0.47930981]
 [-0.73839091 -0.59451366 -0.09885576 -0.54747531 -0.2135602 ]
 [-1.7256793  -1.14324985 -1.32856725 -1.22078536 -2.02664098]
 … ]
```

```
Covariance
[[1.12 0.92 0.88 0.92 1.26]
 [0.92 0.86 0.72 0.77 1.00]
 [0.88 0.72 0.76 0.76 1.08]
 [0.92 0.77 0.76 0.94 1.09]
 [1.26 1.00 1.08 1.09 1.55]]
```

```
Eigenvalues
[4.89e+00 1.57e-01 1.36e-01 4.13e-02
4.67e-03]
```

```
Eigenvectors
[[-0.47 -0.21 -0.27 -0.79  0.20]
 [-0.39 -0.79 -0.08  0.42 -0.19]
 [-0.39  0.25 -0.17  0.42  0.76]
 [-0.41  0.06  0.91 -0.08  0.02]
 [-0.55  0.52 -0.26  0.14 -0.58]]
```

What percentage of the variance is preserved in a two-dimensional PCA projection? (one number)

Write the corresponding linear projection W operator in the empty array (not all cells have necessarily to be filled in) that would allow you to compute **Y** = **X W** :

Do you need to center **X** prior to post multiplication with **W** to get the 2D (non-whitened) PCA projection **Y**? (tick the correct answer)

Yes	No

Which further operations do you have to achieve in order to whiten the 2D projection **Y** ? (strike the incorrect option twice in [correct | ~~incorrect~~])

[Divide | Multiply] the projection **Y** by the [square | square root] of the eigenvalues.

First name: **Noma:**

Family name:

Question 2 – deep learning (DL) and convolutional neural networks (CNNs)

We want to classify handwritten characters (capital letters from A to Z) stored in square images with 22^2 pixels. The characters are not necessarily centered in the images.

For the purpose of classification, the architecture of a deep convolutional neural network is designed. Each layer of neurons is here decomposed in operational layers (e.g., convolutional neurons consist of a convolution and an activation function).

Design the architecture yourself in the table below:

- Fill in all columns “L??” by ticking with a cross just one cell among “Fully” to “Smax” (see meaning below the table). Select the element with the best expected performance level.
- Fill in the missing sizes in the row “size”.
- Fill in the missing number of neurons (row “#neu”) for the output layer L5a-L5b.

	Image	L1a	L1b	L1c	L2a	L2b	L2c	L3a	L3b	L3c	L4a	L4b	L5a	L5b
Size	22	20		10			4			1	1	1	1	1
#neu		8			16			32			64			
Fully														
Conv														
MaxP														
AvgP														
Sigm														
Tanh														
ReLU														
Smax														

Fully:	Fully connected scalar product
Conv:	Convolution 3*3 with stride 1
AvgP:	Average Pooling 2*2 with stride 2
MaxP:	Max Pooling 2*2 with stride 2
Sigm:	Sigmoid
Tanh:	Hyperbolic tangent
ReLU:	Rectified linear unit
SMax:	Softmax

What is the best loss function for the intended application? (tick one)

Mean Square Error (MSE)	Mean Absolute Error (MAE)	Cross-Entropy (CE)

How many parameters are there in the complete network? (one number)

First name:	**Noma:**
Family name:	

Question 3 – model selection

Among the following methods, which are the ones whose aim is to measure overfitting (indicate M in the corresponding box), or to decrease the overfitting (indicate D in the corresponding box). Indicate N/A if the proposed method does not directly measure or decrease the overfitting.

	Cross-validation
	Leave-one-out
	Bootstrap
	Bayesian Information Criterion / Minimum Description Length (BIC / MDL)
	Confusion matrix
	Optimal brain damage/surgeon

In a standard cross-validation process, none of the validation sets are ever used to train the parameters of the model. Therefore, why is a third set of data, often called test set, needed to assess the model performances? Why couldn't you use the (average of the) errors estimated on the validation sets? And what is the drawback (or the "price to pay") of this need for a third set?

When you estimate the performances of a classifier, explain why only using the final accuracy (= the percentage of correct classifications) is a bad idea, and why using a confusion matrix is a much better one?

Give a simple example of a binary classification problem for which a confusion matrix is really needed by the user of the model.

First name: **Noma:**

Family name:

How can you explain that, in a standard classification problem, precision and recall are most often contradictory objectives (if you want to increase one you will probably decrease the other one, for a specific model)?

First name: **Noma:**

Family name:

Question 4 – kernel methods

A kernel method (for example a Support Vector Machine - SVM) transforms the input features in a nonlinear way; the new features are then used in a standard classification algorithm (for example a large margin classifier in the SVM). It is easy to understand that working with new features will change the classification performances, and that if you are lucky with the nonlinear transformation of inputs, the performances can increase. But besides the "chance" to increase the performances, how can you explain that generally speaking, indeed the performances increase? What are the conditions that you have to satisfy (on the nonlinear transformation) in order to guarantee (again, generally speaking) that the performances will increase?

For a simple, linearly separable, binary classification problem, explain why you would prefer (or not) to use a large-margin classifier rather than a perceptron.

First name:	**Noma:**
Family name:	

Part B. Questions on the Project

Question 5 – Categorical Variables

The 'how do you get aroung' categorical variable from the project dataset has values from the following set: walking, bike, motorbike, public transportation, car.

1. Give **one advantage** and **one disadvantage** of encoding this as a one-hot vector instead of a numerical value?

2. Let's say you decide to assign a numerical value between 0 and 1 to each answer from the set but you hesitate between the 2 following options:
   a. [0 (walk), 0.25 (bike), 0.5 (motorbike), 0.75 (pub. tr.), 1.0 (car)]
   b. [0.9 (walk), 1.0 (bike), 0.7 (motorbike), 0.5 (pub. tr.), 0 (car)]

   Option (a) assigns a random equidistant value to each possible answer, option (b) supposes that biking consumes more energy than walking or diving a car and assigns the values according to human perception.

   Explain briefly: Is one option better than the other for a **KNN** ? and for a **Tree-based algorithm***?

* Tree-based algorithms (e.g. Decision trees, Random forest etc.) use a cascade of nodes to group similar datapoints together. Each node separates the data based on a threshold applied to a particular feature