AY: 2024-2025	L3-S5: Dept. of Electrical Engineering
EXAM	**Machine Learning**	Teacher: A. Mhamdi
Dec. 2024	Time Limit: $1\frac{1}{2}$h

This document contains **6** pages numbered from **1/6** to **6/6**. As soon as it is handed over to you, make sure it is complete. The **2** tasks are independent and can be treated in the order that suits you.

The following rules apply:

❶ **A handwritten double-sided A4** sheet is permitted.

❷ **Any electronic material**, except basic calculator, is prohibited.

❸ **Mysterious or unsupported answers** will not receive full credit.

❹ **Round results** to the nearest <u>thousandth</u> (*i.e., third digit after the decimal point*).

<center>⋘∘⋙⋙∘⋙</center>

Task № 1   ⏳ 50mn | (9 points)

Examine the code below, where we developed a straightforward program that relies on logistic regression to estimate an individual's likelihood to buy a product based on his age and income.

```
[1]: import numpy as np
     import pandas as pd
```

```
[2]: df = pd.read_csv('../Datasets/Social_Network_Ads.csv')
```

```
[3]: df.head()
```

```
[3]:    Age  EstimatedSalary  Purchased
     0   19            19000          0
     1   35            20000          0
     2   19            76000          0
     3   27            58000          0
     4   27            84000          0
```

```
[4]: df.Purchased.value_counts()
```

```
[4]: 0    221
     1    138
     Name: Purchased, dtype: int64
```

```python
[5]: X = df.iloc[:, :-1].values; X[:, -1] = X[:, -1] / 1000
     y = df.iloc[:, -1].values
```

```python
[6]: from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.
       ↪75, random_state=147, stratify=y)
```

```python
[7]: np.sum(y == 1) / len(y)
```

```python
[8]: np.sum(y_train == 1) / len(y_train)
```

```python
[9]: np.sum(y_test == 1) / len(y_test)
```

```python
[10]: from sklearn.linear_model import LogisticRegression
```

```python
[11]: clf = LogisticRegression()
      clf.fit(X_train, y_train)
      clf.coef_, clf.intercept_
```

```
[11]: (array([[0.25667617, 0.03747588]]), array([-13.48631476]))
```

```python
[12]: sample = np.array([[47, 62]])
```

```python
[13]: clf.predict_proba(sample)
```

```python
[14]: clf.predict(sample)
```

```python
[15]: y_pred = clf.predict(X_test)
```

```python
[16]: pd.crosstab(y_pred, y_test, rownames=['Predicted'],
      colnames=['Expected'], margins=True)
```

```
[16]: Expected    0   1  All
      Predicted
      0          48   8   56
      1           7  27   34
      All        55  35   90
```

```python
[17]: from sklearn.metrics import classification_report
```

```python
[18]: print(classification_report(y_test, y_pred))
```

(a) (2 points) What would be the results of instructions in cells #7, #8 and #9.

> The instruction at cell #7 will evaluate to $\frac{138}{138 + 221} \approx 0.384$. The stratify parameter takes the target variable (labels) as input and ensures that the proportion of each class is preserved in both the training and testing sets. Following that, stratify=y will ensure that the random split has:
>
> - **38.4%** train (*equiv.* test) data is ones;
> - **61.6%** train (*equiv.* test) data is zeros.

```
[8]: 0.38440111420612816
```

```
[9]: 0.3828996282527881
```

```
[10]: 0.3888888888888889
```

(b) (3 points) What are the outputs of cells #13 and #14.

```
[14]: array([[0.28885115, 0.71114885]])
```

```
[15]: array([1])
```

The computation is done basically as follows:

```
[19]: z = np.sum(clf.coef_ * sample) + clf.intercept_
o = 1/(1+np.exp(-z))
o
```

```
[19]: array([0.71114885])
```

(c) (4 points) Rewrite the classification report on your paper and fill in the gaps.

	precision	recall	f1-score	support
0	0.86	0.87	0.86	55
1	0.79	0.77	0.78	35
accuracy			0.83	90
macro avg	0.83	0.82	0.82	90
weighted avg	0.83	0.83	0.83	90

AY: 2024-2025

L3-S5: Dept. of Electrical Engineering

EXAM | Machine Learning

Dec. 2024

Teacher: A. Mhamdi

Full Name: ....

ID: ....

Class: ....

Room: ....

Time Limit: $1\frac{1}{2}$h

ANSWER SHEET

Task Nº2

⏳ 40mn | (11 points)

(a) ($\frac{1}{2}$ point) What symbol(s) do you use to assess equality between two elements?

○ && ○ || √ == ○ !=

(b) ($\frac{1}{2}$ point) What value would be returned by this check for equality?

```
5 != 6
```

○ Yes ○ No ○ false ○ true ○ False √ True

(c) ($\frac{1}{2}$ point) What is the term used to describe items that may be passed into a function?

√ arguments ○ paradigms ○ attributes ○ decorators

(d) ($\frac{1}{2}$ point) What is the term to describe this code?

```
count, fruit, price = (2, 'apple', 3.5)
```

○ tuple assignment ○ tuple matching √ tuple unpacking ○ tuple duplication

(e) ($\frac{1}{2}$ point) What is the correct syntax for creating a variable that is bound to a set?

√ my_set = {0, 'apple', 3.5}

○ my_set = 0, 'apple', 3.5

○ my_set = to_set(0, 'apple', 3.5)

○ my_set = (0, 'apple', 3.5).to_set()

○ my_set = (0, 'apple', 3.5).set()

(f) ($\frac{1}{2}$ point) How is machine learning related to artificial intelligence?

○ Artificial intelligence focuses on classification, while machine learning is about clustering data.

○ Artificial intelligence is form of unsupervised machine learning.

√ Machine learning is a type of artificial intelligence that relies on learning through data.

○ Machine learning and artificial intelligence are the same thing.

✂ -

(g) (1 point) To predict a quantity value. use "_____".

√ regression   ○ classification   ○ clustering   ○ dimensionality reduction

(h) (1 point) What is one reason not to use the same data for both your training set and your testing set?

○ You will almost certainly underfit the model.

○ You will pick the wrong algorithm.

√ You will almost certainly overfit the model.

○ You might not have enough data for both.

(i) (1 point) You created machine learning system that interacts with its environment and responds to errors and rewards. What type of machine learning system is it?

√ reinforcement learning

○ supervised learning

○ semi-supervised learning

○ unsupervised learning

(j) (1 point) What is the purpose of an if/else statement?

○ It tells the computer which chunk of code to run if the instructions you coded are incorrect.

○ It runs one chunk of code if all the imports were successful, and another chunk of code if the imports were not successful.

○ It tells the computer which chunk of code to run if there is enough memory to handle it, and which chunk of code to run if there is not enough memory to handle it.

√ It executes one chunk of code if a condition is true, but a different chunk of code if the condition is false.

(k) (1 point) What is the correct syntax for instantiating a new object of the type Game?

○ my_game = class.Game()

○ my_game = class(Game)

√ my_game = Game()

○ my_game = Game.init()

(l) (1 point) Which choice is best for binary classification?

○ K-means   √ k-NN   ○ Linear regression   ○ PCA

✂ -
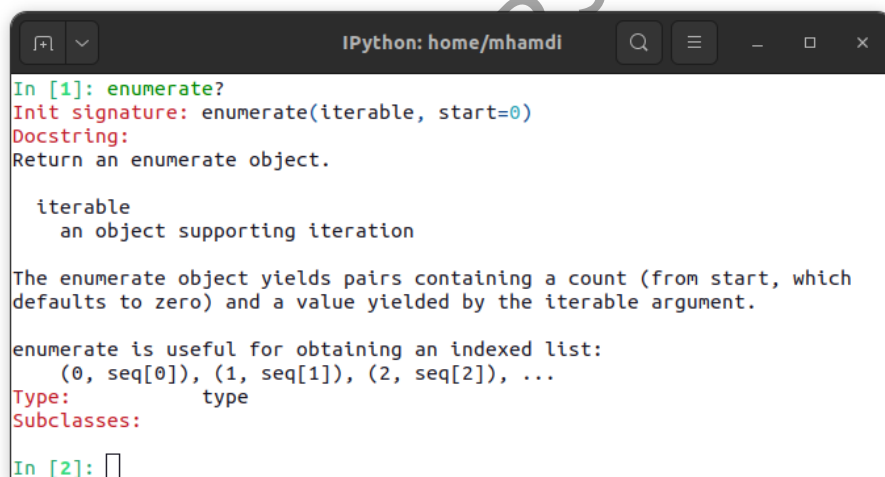
(m) (1 point) In k-nearest neighbors, the closer you are to neighbor, the more likely you are to "_____".

√ share common characteristics

◯ be part of the root node

◯ have a Euclidean connection

◯ be part of the same cluster

(n) (1 point) What would this expression return?

```
college_years = ['Freshman', 'Sophomore', 'Junior', 'Senior']
list(enumerate(college_years, 2019))
```

```
In [1]: enumerate?
Init signature: enumerate(iterable, start=0)
Docstring:
Return an enumerate object.

  iterable
    an object supporting iteration

The enumerate object yields pairs containing a count (from start, which
defaults to zero) and a value yielded by the iterable argument.

enumerate is useful for obtaining an indexed list:
    (0, seq[0]), (1, seq[1]), (2, seq[2]), ...
Type:           type
Subclasses:

In [2]:
```

◯ [('Freshman', 2019), ('Sophomore', 2020), ('Junior', 2021), ('Senior', 2022)]

◯ [(2019, 2020, 2021, 2022), ('Freshman', 'Sophomore', 'Junior', 'Senior')]

◯ [('Freshman', 'Sophomore', 'Junior', 'Senior'), (2019, 2020, 2021, 2022)]

√ [(2019, 'Freshman'), (2020, 'Sophomore'), (2021, 'Junior'), (2022, 'Senior')]