

## MATCH INTERCLASSES EDSC

### AS3 VS ISE3

#### Partie Théorique

Question 1 : Quelle méthode de sélection de caractéristiques permet de sélectionner des caractéristiques en fonction de leur importance pour le modèle ?

- A. Forward Selection
- B. Backward Selection
- C. Sélection basée sur l'arbre de décision
- D. Toutes les réponses ci-dessus

Réponse : D

Question 2 : Quel terme décrit le processus de division des données en plusieurs sous-ensembles de taille fixe, souvent utilisé dans les réseaux neuronaux convolutifs pour l'extraction de caractéristiques ?

Réponse : Pooling

Question 3 : Quelle est la méthode en Python utilisée pour optimiser des fonctions non linéaires en utilisant des dérivées partielles pour ajuster les poids du modèle ?

- A. Gradient Boosting
- B. Stochastic Gradient Descent
- C. Backpropagation
- D. Logistic Regression

Réponse : C

Question 4 : En deep learning, quel est le terme utilisé pour la technique qui permet de régulariser un modèle en désactivant de manière aléatoire un ensemble de neurones pendant l'entraînement ?

Réponse : Dropout

Question 5 : Quel est le nom de l'algorithme de clustering qui fonctionne en recherchant des densités de points élevés et les sépare en différents clusters ?

Réponse : DBSCAN

Question 6 : Dans Git, quelle commande permet de mettre à jour votre branche locale avec les modifications de la branche distante correspondante ?

Réponse : git pull

Question 7 : En apprentissage automatique, la régularisation L2 est aussi connue sous le nom de Lasso Regression. (Vrai ou Faux)

Réponse : Faux (L2 est Ridge Regression, L1 est Lasso Regression)

Question 8 : Quelle technique NLP permet de diviser un texte en unités plus petites comme des mots ou des phrases ?

Réponse : Tokenization

Question 9 : Quelle composante de Hadoop est utilisée pour le stockage distribué de grandes quantités de données ?

Réponse : HDFS (Hadoop Distributed File System)

Question 10 : En POO, comment appelle-t-on une fonction qui est liée à une classe plutôt qu'à une instance de cette classe ?

Réponse : Méthode statique

Partie Pratique

Résolution de problème

Problème

---

Ecrire une fonction qui permet de calculer l'accélération d'un corps entre deux points de l'espace à partir de la mesure de sa vitesse à chacune de ces différentes positions et du temps séparant ces mesures. Sa vitesse à chacune de ces positions est entrée dans une liste de taille deux. On conviendra que les vitesses sont exprimées en m/s et le temps en seconde.

Exemple :

fonction vitesses = 20,10 , temps = 100 fonction vitesses = 40,40 , temps = 100

---

---

Problème 2

Calculez l'IMC d'un individu à partir de son poids donné en kilogramme (kg) et de sa taille donnée en centimètre (cm). Vous devrez ensuite donner le résultat suivi de l'interprétation appropriée. Voici

comment s'interprète la valeur obtenue : moins de 16,5 – dénutrition, 16,5 à moins de 18,5 – maigreur, 18,5 à moins de 25 - poids normal, 25 à moins de 30 –surpoids, 30 à moins de 35 - obésité modérée, 35 à moins de 40 - obésité sévère, 40 et plus-obésité morbide ou massive.

Exemple :

- *fonction (poids = 75, taille = 175) = '24,49 poids normal'*

- *fonction (poids = 75, taille = 200) = '18,75 poids normal'*

### Problème 3

Vous voulez partager un gâteau de forme circulaire de rayon r en n

---

parts	égales.	Disposant d'une	machine	pouvant	réaliser	cette
-------	---------	-----------------	---------	---------	----------	-------

---

opération, il vous est demandé d'entrer la surface que doit avoir chaque tranche pour que tout le monde puisse en avoir et qu'il puisse rester 3 parts supplémentaires pour les retardataires et les personnes non prises en compte.

On prendra  $a = 3,1415$  et le rayon en cm.

Exemple :

- *fonction (part = 10, rayon = 100) = 48,33*

- *fonction (part = 7, rayon = 200) = 125,66*

### Détection d'erreur

On supposera que toutes les bibliothèques utilisées ont été installées au préalable, donc les erreurs liées aux installations de bibliothèques ne seront pas valables.

#### Equipe A Code 1

```
class Dog:
def __init__(self, name, age):
name = name
age = age

def bark(self):
print("Woof! My name is", self.name) my_dog = Dog("Buddy", 5)
my_dog.bark()
```

#### Equipe B Code 1

```
class Animal:
def __init__(self, name, ):
self.name = name
```

```

def speak(self):
    pas

class Dog(Animal):
    def __init__(self, name, breed):
        super().__init__(name)
        self.breed = breed
my_dog = Dog("Buddy", "Golden Retriever")
my_dog.speak()

```

Equipe A Code 2

```

class Base:
    def __init__(self):
        self.value = "Base2"

class Derived(Base):
    def __init__(self):
        self.derivedvalue = "Derived"

derived = Derived()
print(derived.value)

```

Equipe B Code 2

```

from sklearn.linear_model import LinearRegression import numpy
as np

X = np.array([[1, 2], [3, np.na], [7, 8]]) y = np.array([1, 2, 3])

model = LinearRegression()
model.fit(X, y)

```

Equipe A Code 3

```

from sklearn.linear_model import LogisticRegression import numpy
as np

X_train = np.array([[1, 2, 3], [3, 4, 5], [5, 6, 7]]) y_train = np.array([0,
1, 0])
X_test = np.array([[2, 3]])

model = LogisticRegression()
model.fit(X_train, y_train)

model.predict(X_test)

```

Equipe B Code 3

```

class Parent:
    def __init__(parent):
        parent.parent = None
        print("Parent initialized")

```

```

class Child(Parent):
def __init__(child):
super().__init__()

child_instance = Child()

```

Equipe A Code 4

```

from sklearn.ensemble import XGBClassifier import numpy as np
X = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]]) y = np.array([0, 0,
0, 0, 1])

model = XGBClassifier()
model.fit(X, y)

predictions = model.predict(X)
print("Predictions:", predictions)

```

Equipe B Code 4

```

from sklearn.model_selection import cross_val_score from
sklearn.linear_model import LinearRegression import numpy as np
X = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]]) y = np.array([1, 2,
3, 4, 5])

model = LinearRegression()
scores = cross_val_score(model, X, y, cv=10) print("Cross-validation
scores:", scores)

```

Equipe A Code 5

```

import numpy as np
x = np.random.rand(100)
y = np.random.rand(100)
correlation = np.corrcoef(x, y)
corr_xy = correlation[0, 2]
print("\nCorrélation entre x et y:", corr_xy)

```

Equipe B Code 5

```

from sklearn.linear_model import Perceptron import numpy as np
import matplotlib.pyplot as plt

np.random.seed(0)
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)

model = Perceptron()
model.fit(X, y.ravel())

y_pred = model.predict(X)

```

Décryptage de code

Equipe A Code 1

```
import cv2

def my_function(input):
    a = cv2.imread(input, cv2.IMREAD_GRAYSCALE)
    b = cv2.Canny(a, 100, 200)
    cv2.imshow('Edges', b)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

Equipe B Code 1

```
import re

def my_function(input):
    a = r"\d+(?:°[CF])?"
    result = re.findall(a, input)
    return result
```

Equipe A Code 2

```
def my_function(input):
    if len(input) <= 1:
        return input
    pivot = input[len(input) // 2]
    left = [x for x in input if x < pivot]
    middle = [x for x in input if x == pivot]
    right = [x for x in input if x > pivot]
    return my_function(left) + middle + my_function(right)
```

Equipe B Code 2

```
from textblob import TextBlob

def my_function(text):
    blob = TextBlob(text)
    return blob.sentiment.polarity, blob.sentiment.subjectivity
```

Equipe A Code 3

```
import re

def my_function(input):
    a = r"\b\d+(?:[.,]\d+)?\s*CFA\b"
    result = re.findall(a, input)
    return result
```

Equipe B Code 3

```
from itertools import groupby

liste = ['apple', 'banana', 'orange', 'avocado', 'blueberry']
liste_triee = sorted(liste, key=lambda x: x[0])
```

```

resultat = {key: list(group) for key, group in
groupby(liste_triee, key=lambda x: x[0])}
print(resultat)

```

Equipe A Code 4

```

import nltk
nltk.download('brown')
from textblob import TextBlob

def my_function(input):
blob = TextBlob(input)
return blob.noun_phrases

text = "Python is a popular programming language created by Guido
van Rossum."
print(my_function(text))

```

Equipe B Code 4

```

import difflib

def my_function(input1, input2):
a = difflib.Differ()
result = list(a.compare(input1.splitlines(), input2.splitlines()))
return '\n'.join(result)

```

Equipe A Code 5

```

from bs4 import BeautifulSoup
from requests import get

URL="https://www.jumia.ci/epicerie/" page = get(URL)
print(page.text)

```

Equipe B Code 5

```

def my_function(input):
result = {}
with open(input, "r", encoding="utf-8") as f: for ligne in f:
mots = ligne.strip().split()
for mot in mots:
result[mot] = result.get(mot, 0) + 1 return result

```