Nom :	Jacques-Brissette
Prénom :	Dominique

Code permanent : JACD16069105
Signature : *Dominique Jacques-Brissette*

ÉTÉ 2015 – EXAMEN FINAL
GABARIT POUR VOS RÉPONSES

COURS :	MTI830 Forage de données textuelles et audiovisuelles
GROUPE(S) :	01
ENSEIGNANT(S) :	Sylvie Ratté, professeure
DATE :	du 22 au 29 juillet 2015
HEURE :	Examen de type « apportez-moi à la maison »
DURÉE :	7 jours
PONDÉRATION :	30 %
DOCUMENTATION :	Utilisation de la calculatrice ☒ Autorisée ☐ Interdite. Toute documentation permise.
COMMENTAIRE(S) :	**Répondre dans ce gabarit**

IMPORTANT POUR LES ÉTUDIANTS(ES)

La mention « ÉCHEC » sera appliquée si le questionnaire n'est pas remis avec son rapport d'examen.

- L'étudiant doit s'assurer qu'il a rempli au complet la page de titre du gabarit (ci-dessus) et du questionnaire.
- L'étudiant DOIT MODIFIER l'entête de la section 2 afin de s'assurer que son nom et prénom apparaissent sur toutes les pages.
- L'étudiant PEUT INTÉGRER des pages numérisées dans son rapport final mais il/elle doit s'assurer que son nom et prénom apparaissent sur toutes les pages.
- Chaque question doit obligatoirement débuter sur une nouvelle page.

QUESTION 1 (20 points) cours 2

1.1 (5 points)

Résultats :

Age	p	1-p	H	Proportion
Between16and45	0.5	0.5	1	0.4
Between46and64	0.1428571429	0.8571428571	0.5916727786	0.35
Over65	0.2	0.8	0.7219280949	0.25
Entropie Moyenne	0.7875674962			

FederoffsBiomarkers	p	1-p	H	Proportion
Low	0.3333333333	0.6666666667	0.9182958341	0.45
Medium	0.6666666667	0.3333333333	0.9182958341	0.4
High	0.125	0.875	0.5435644432	0.15
Entropie Moyenne	0.8620861254			

FiveYearsExpectancy	p	1-p	H	Proportion
Promising	0.5555555556	0.4444444444	0.9910760598	0.45
Not_promising	0.09090909091	0.9090909091	0.4394969869	0.55
Entropie Moyenne	0.6877075697			

Détails des calculs :

Les calculs ont été effectués dans un fichier excel (les tableaux-ci-haut ont été copiés de là).

Pour les résultats de **`P`** et **`1-P`**, je prends la proportion du nombre de résultats positifs parmis tous les exemples ayant la même valeur pour cet attribut, je m'explique :
Par exemple pour l'âge "**`Over65`**", il y a 5 exemples de patients au total avec cette

valeur d'attribut, dont un seul qui est positif, le **`P`** est donc de **⅕** soit **`0.2`**, les patients négatifs sont donc de **⅘** soit **`0.8`** ou encore **`1-P`**.

Pour les résultats de l'entropie H, j'utilise la formule suivante :
$H = \ -p \times log_2(p) - (1-p) \times log_2(1-p)$

En utilisant le même exemple (Attribut "Age" ayant la valeur "Over65"), j'ai :
$H_{Age:Over65} = \ -0.2 \times log_2(0.2) - (0.8) \times log_2(0.8) = 0.721928095$

Ensuite la proportion représente la fraction que représente le nombre de patients groupés selon une valeur d'un attribut comparé au nombre total des exemples. Comme il y a 20 patients en tout, et que ceux ayant l'âge "**`Over65`**" représent 5 patients, leur proportion est donc de **`5/20`** soit **`0.25`**.

Finalement, pour l'entropie moyenne (`EM`), pour chaque valeur d'un attribut on multiplie l'entropie (`H`) par la proportion (`Prop`) de cette valeur, puis on fait la sommation de ces entropies pondérées pour toutes les valeurs différentes d'un attribut. Par exemple avec l'âge :

$$EM_{Age} = H_{16-45} \times Prop_{16-45} + H_{46-64} \times Prop_{46-64} + H_{Over65} \times Prop_{Over65}$$

$$EM_{Age} = 1 * 0.4 \ + \ 0.59167 * 0.35 \ + \ 0.721928 * 0.25 \ = \ 0.78756$$

Réponse :

L'attribut placé à la racine de l'arbre des décision serait "**`FiveYearsExpectancy`**" puisque c'est l'attribut ayant l'entropie moyenne la moins élevée des 3 attributs.

1.2 (5 points)

P(X | Y)	Between46and64	Medium	Promising	Probabilité appartenance
Positive	P(Age46-64 | Positive) 0.1666666667 1/6	P(Medium | Positive) 0.3333333333 2/6	P(Promising | Positive) 0.8333333333 5/6	1/6 * 2/6 * 5/6 = 10/216 **0.0462962963**
Negative	P(Age46-64 | Negative) 0.4285714286 6/14	P(Medium | Negative) 0.07142857143 1/14	P(Promising | Negative) 0.2857142857 4/14	6/14 * 1/14 * 4/14 = 24/2744 **0.008746355685**

La classification de la nouvelle instance sera “Positive” puisque c’est celle ayant la meilleure probabilité selon NaiveBayes.

1.3 (5 points)

Réponse :

Comme l'algorithme ID3 se base sur l'entropie des attributs (un peu comme on a fait en 1.1), et que les proportions P et 1-P ne seront pas modifiées (car, par exemple, 1/6 est la même chose que 6/36, ou encore pour n'importe quel [1*d]/[6*d]), les entropies ne changeront pas et n'affectera donc pas l'arbre de décision.

En effet, on passerait de $p = \frac{a}{b}$ à $p = \frac{a*d}{b*d} = \frac{a}{b}$ ce qui est exactement la même fraction.

L'arbre ne prendrait que plus de temps à être calculé, mais au final serait intact.

Donc au final, NON, la performance de l'arbre de décision ID3 ne serait pas améliorée.

1.4 (5 points)

La valeur K doit nécessairement être plus grande que dans l'ensemble non-dupliqué parce que si par exemple on a d=3 (donc 3 duplicats par exemple) et qu'on a K=2, les 2 plus proches voisins vont toujours être les 2 autres duplicats, tandis que dans l'ensemble non dupliqué K=2 aurait trouvé d'autres voisins.

QUESTION 2 (20 points) cours 3

2.1 (7 points)

Selon la mesure Kappa de Cohen, la paire d'expert qui a le meilleur score (taux d'accord le plus élevé) est la paire A-C avec un coefficient Kappa de 0.90066.

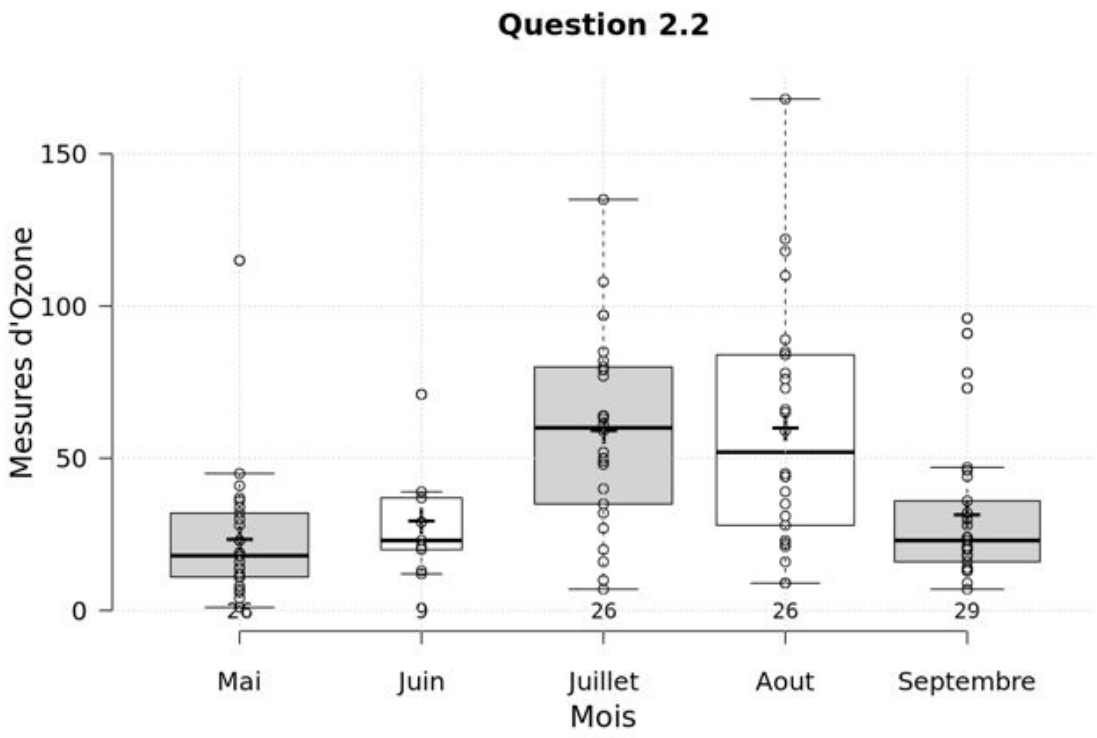
*Note: les calculs sont détaillés dans un fichier Excel, mais l'énoncé ne demande pas de détailler les calculs. J'ai pris le soin de calculer chaque valeur (nombre d'accords/de désaccords pour chaque paire d'experts, nombre de résultats pour chaque classe pour chaque expert, puis les multiples de ratios A_{sport}*B_{sport}, etc.) afin de pouvoir trouver les Pr(a) et Pr(e) pour chaque paire puis ensuite calculer le Kappa avec ceux-ci.*

2.2 (7 points)

J'ai fait un script Python pour modifier le format des données vers un format qui ressemble à celui-ci :

```
"Mai","Juin","Juillet","Aout","Septembre"
41,29,135,39,96
36,71,49,9,78
12,39,32,16,73
18,23,64,78,91
28,21,40,35,47
23,37,77,66,32
19,20,97,122,20
8,12,97,89,23
7,13,85,110,21
16,"",10,44,24
11,"",27,28,44
14,"",7,65,21
……suite
```

Ensuite j'ai utilisé ce fichier sur le site http://boxplot.tyerslab.com/ pour les visualiser à l'aide de boxplot (avec "Tukey whiskers"), cela me donne le résultat suivant :



2.3 (6 points)

Pour C1:

$$Précision = \frac{VraiPositifs}{VraiPositifs + FauxPositif} = \frac{D1}{D1 + D4} = 1/2$$

$$Rappel = \frac{VraiPositifs}{VraiPositifs + FauxNegatif} = \frac{D1}{D1 + D3 + D5} = 1/3$$

Pour C2:

$$Précision = \frac{VraiPositifs}{VraiPositifs + FauxPositif} = \frac{D1 + D2 + D4}{D1 + D2 + D4} = 3/3 = 1$$

$$Rappel = \frac{VraiPositifs}{VraiPositifs + FauxNegatif} = \frac{D1 + D2 + D4}{D1 + D2 + D4} = 3/3 = 1$$

Donc la réponse est (B), puisque ça y correspond parfaitement.

QUESTION 3 (20 points) cours 4

3.1 (3 points)

Itemset	Support absolu	Support relatif
{e}	8	8/10 = 80%
{b, d}	2	2/10 = 20%
{b, d, e}	2	2/10 = 20%

3.2 (3 points)

Indice de confiance pour règles d'associations

{b, d} ⇒ {e} : C = S{b,d,e} / S{b,d} = 2/2 = 100%

{e} ⇒ {b, d} : C = S{b,d,e} / S{e} = 2/8 = 25%

3.3 (4 points)

Seuls les items individuels {Beer}, {Nuts}, {Diapers} et {Eggs} ont un support d'au moins 50%, car {Coffee} et {Milk} ont seulement 40% de support.

Itemset	**Support absolu**	**Support relatif**
{Beer}	4	4/5 = 80%
{Nuts}	4	4/5 = 80%
{Diapers}	4	4/5 = 80%
{Eggs}	3	3/5 = 60%
{Beer, Nuts}	3	3/5 = 60%
{Beer, Diapers}	3	3/5 = 60%
~~*{Beer, Eggs}*~~	~~*2*~~	~~*2/5 = 40%*~~
{Nuts, Diapers}	3	3/5 = 60%
~~*{Nuts, Eggs}*~~	~~*2*~~	~~*2/5 = 40%*~~
~~*{Diapers, Eggs}*~~	~~*2*~~	~~*2/5 = 40%*~~
~~*{Beer, Nuts, Diapers}*~~	~~*2*~~	~~*2/5 = 40%*~~

{Beer} ⇒ {Nuts} : C = S{Beer, Nuts} / S{Beer} = **3/4** **= 75%**

{Beer} ⇒ {Diapers} : C = S{Beer, Diapers} / S{Beer} = **3/4** **= 75%**

~~{Beer} ⇒ {Eggs} : C = S{Beer, Eggs} / S{Beer} = 2/4 = 50%~~

{Nuts} ⇒ {Beer} : C = S{Beer, Nuts} / S{Nuts} = **3/4** **= 75%**

{Nuts} ⇒ {Diapers} : C = S{Nuts, Diapers} / S{Nuts} = **3/4** **= 75%**

~~{Nuts} ⇒ {Eggs} : C = S{Nuts, Eggs} / S{Nuts} = 2/4 = 50%~~

{Diapers} ⇒ {Beer} : C = S{Beer, Diapers} / S{Diapers} = **3/4** **= 75%**

{Diapers} ⇒ {Nuts} : C = S{Diapers, Nuts} / S{Diapers} = **3/4** **= 75%**

~~{Diapers} ⇒ {Eggs} : C = S{Eggs, Diapers} / S{Diapers} = 2/4 = 50%~~

On ne peut pas continuer, car on utiliserait toujours des itemsets ne respectant pas le minsup de 50%.

On trouve donc 6 règles d'association, la réponse est donc (E).

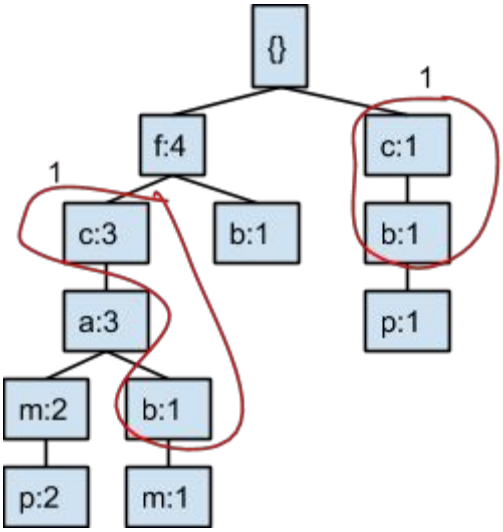
3.4 (5 points)

La base de données f-conditionnelle est la réponse :

A. {c, b, p}: 1

3.5 (5 points)

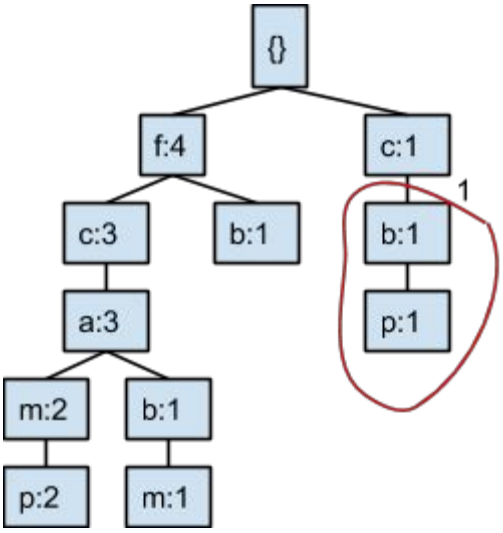
A. Itemset {b, c}



1+1 = 2 >= minsup(2) → OK!

Réponse A qualifiée

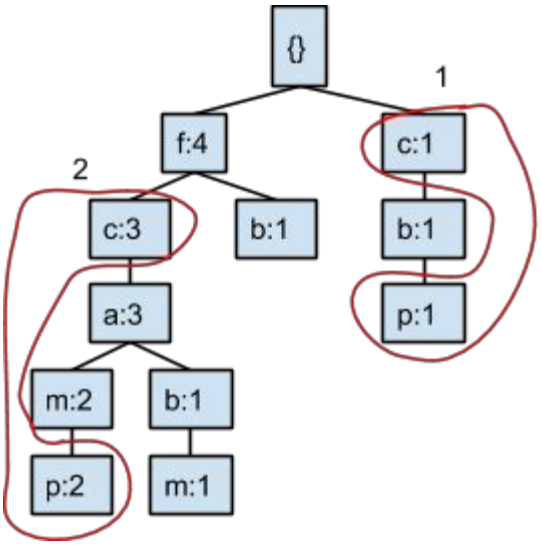
B. Itemset {b, p}



1 < minsup(2) → NOT OK!

Réponse B non valide

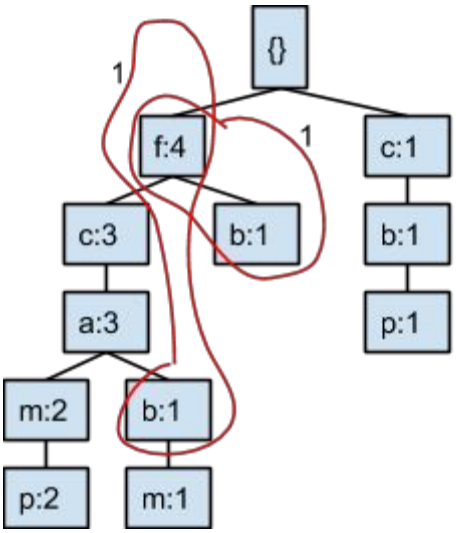
C. Itemset {c, p}



1+2 = 3 > minsup(2) → OK!

Réponse C qualifiée

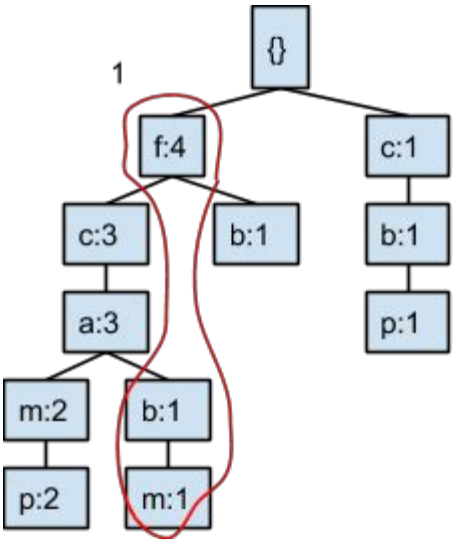
D. Itemset {f, b}



1+1 = 2 >= minsup(2) → OK!

Réponse D qualifiée

E. Itemset {f, b, m}



1 < minsup(2) → NOT OK!

Réponse E non valide

Les réponses possibles sont donc : A, C et D.

QUESTION 4 (20 points) cours 5

4.1 (8 points)

```
import re
import math
from collections import Counter

def cosine_similarity(vectorA, vectorB):
	cosineResult = 0
	intersection = set(vectorA.keys()) & set(vectorB.keys())
	num = sum([vectorA[x] * vectorB[x] for x in intersection])

	sumA = sum([vectorA[x]**2 for x in vectorA.keys()])
	sumB = sum([vectorB[x]**2 for x in vectorB.keys()])
	denom = math.sqrt(sumA) * math.sqrt(sumB)

	if denom != 0:
		cosineResult = float(num) / float(denom)

	return cosineResult


aWord = re.compile(r'\w+')

txtA = "all grown-ups were once children but only few of them remember it"
bowA = Counter(aWord.findall(txtA))
txtB = "all children should be very understanding of grown-ups"
bowB = Counter(aWord.findall(txtB))

print "\n\n\nCosine similarity of :"
print '    "' + txtA + '"'
print '  and'
print '    "' + txtB + '"'
print "is : " + str(cosine_similarity(bowA, bowB))
```

Avec le code ci-dessus, j'obtiens une similarité cosinus de 0.462250163521.

4.2 (4 points)

Si le terme apparaît dans un seul document, il aura un score élevé (surtout s'il est fréquent dans celui-ci). Ce terme est donc propice à ce document et est considéré plus important.

Si le terme apparaît dans tous les documents, il aura un score bas (par exemple les mots comme "le" et "un", etc. ont un score bas puisqu'ils apparaissent dans tous les documents, c'est bien puisqu'ils ne sont pas importants).

La transformation TF-IDF permet d'identifier les mots les plus importants, ou mots principaux d'un document. De cette manière, même les mots très fréquents qui apparaissent dans tous les documents ne seront pas identifiés comme importants, car ils ne permettent pas d'identifier de quoi le document parle par rapport aux autres.

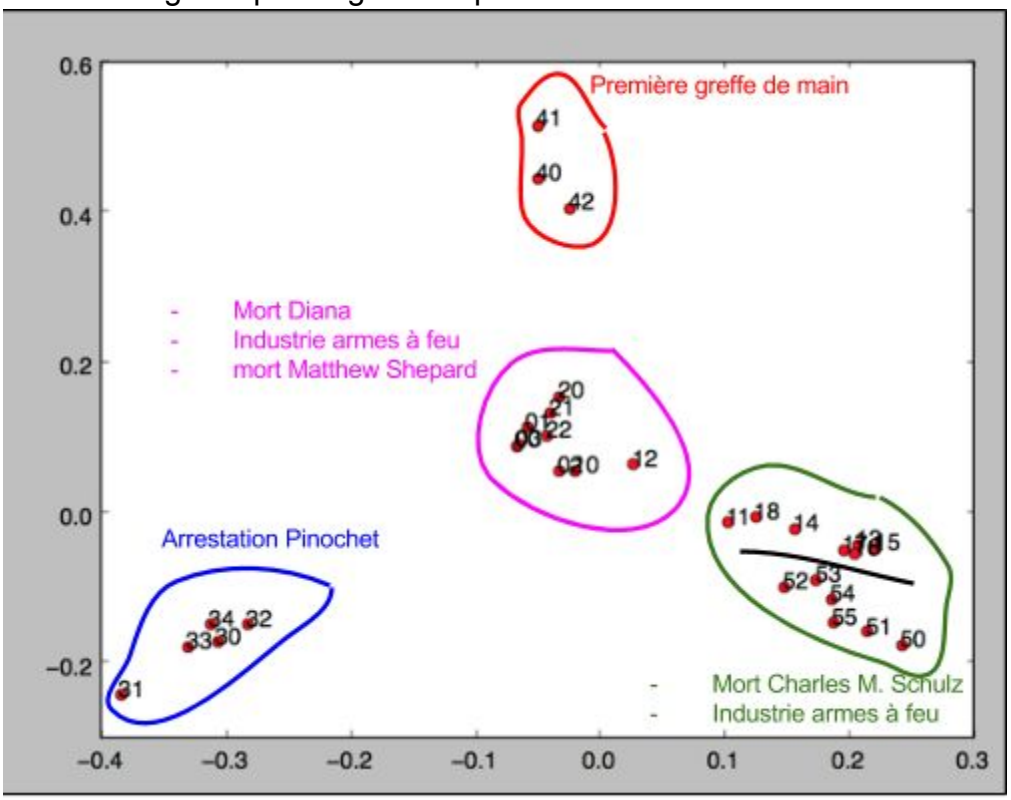
4.3 (8 points)

Dans ce cas, ça nous permettrait d'identifier quels pays ont des besoins ou préférences typiques/uniques (pas nécessairement uniques mais différentes) en termes de consommation de produits. Par exemple, si tous les pays achètent du papier de toilette, cet item aura un score bas, donc pas considéré comme important, tandis que si seulement un ou quelques pays achètent des pains baguettes, on peut alors découvrir cette tendance/préférence à l'aide de TF-IDF, puisque les baguettes auront un gros score pour ces pays.

QUESTION 5 (24 points) cours 6

5.1 (1 point)

Voici le nuage de points généré après transformation.



On voit clairement que le groupe 3 (Arrestation Pinochet) et le groupe 4 (Greffe) sont très éloignés de tout le reste, ils occupent leur propre espace et sont clairement distincts par rapport à tous les autres, ce qui a beaucoup de sens puisque ces 2 sujets n'ont rien en commun l'un et l'autre ainsi qu'avec les autres.

Tandis que le groupe au centre porte sur la mort de Diana, l'industrie des armes à feu, et la mort de Matthew Shepard. Cela a du sens puisqu'il s'agit de violence et de tuerie.

Dans le dernier groupe incluant la mort de Charles et l'industrie des armes à feu, on peut remarquer qu'ils sont relativement près l'un de l'autre, mais qu'ils occupent chacun leur espace propre. Effectivement, on pourrait presque créer 2 sous-groupes, puisque les 2 "topics" sont composés d'articles qui sont pratiquement alignés linéairement. Ils sont donc plus ou moins reliés, puisque les 2 font référence à la mort (armes à feu sont destinés à tuer, et la mort d'un individu) et se passent aux États-Unis, mais qu'ils parlent quand même de sujet différents.

Ça fait aussi beaucoup de sens de voir le groupe du centre (rose) placé près du groupe vert, puisqu'il s'agit de topics qui utilisent plusieurs mots en commun. De plus, le groupe du centre est plus éloigné de la mort de Charles, car en effet les sujets diffèrent plus.

5.2 (2 points)

En m'inspirant du code de la fonction `transformDocumentToString` fournie, j'ai créé ma propre fonction pour enlever la ponctuation et les stopwords, voici le code en question.

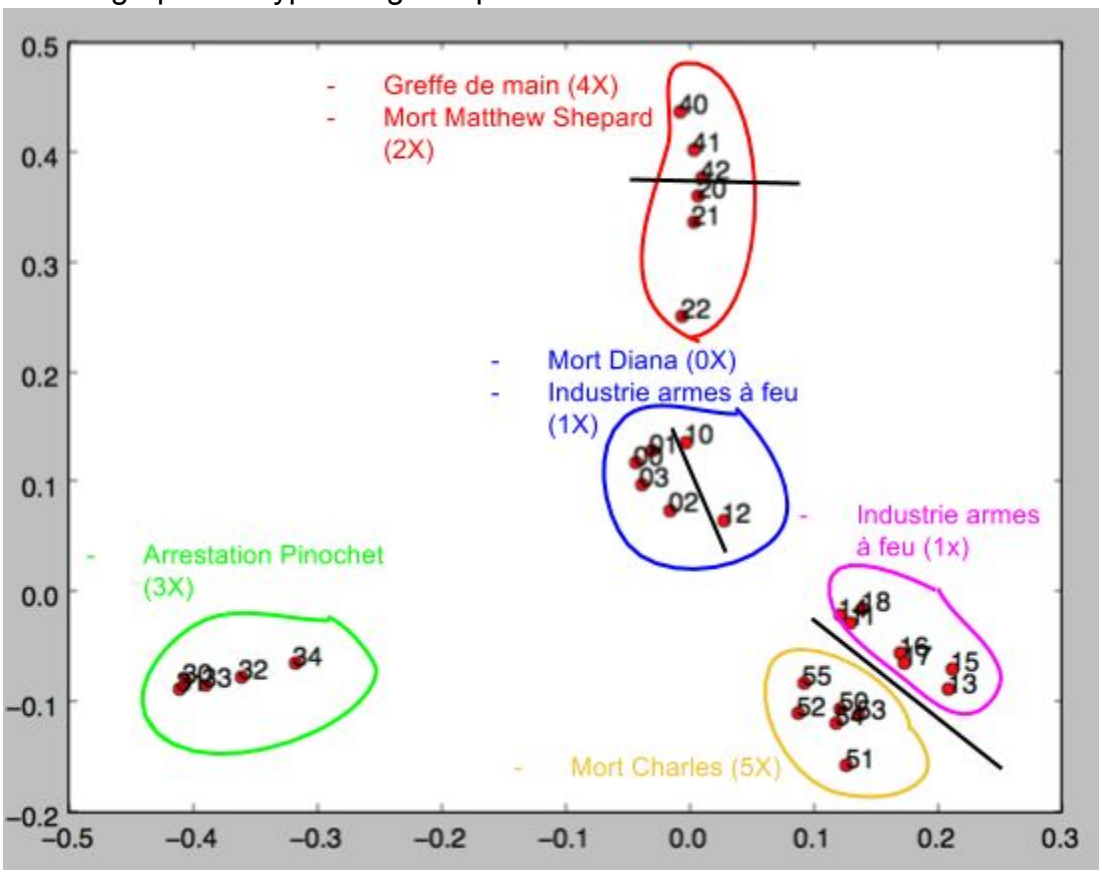
```
# ajout de ces imports supplémentaires au début du fichier
import string # to remove punctuation
from nltk.corpus import stopwords #stopwords
# …
def transformDocToStringNoStopwords(collection):
  cachedStopWords = stopwords.words("english")
  stringDocuments = []
  for d in documents:
    sen = ''
    for s in d:
      #remove punctuation
      exclude = set(string.punctuation)
      s = ''.join(ch for ch in s if ch not in exclude)
      #remove stopwords
      s = ' '.join([word for word in s.split() if word not in stopwords.words("english")])
      sen = sen+s
    stringDocuments.append(sen)
  return stringDocuments
```

Puis j'ai modifié le restant du code pour utiliser cette nouvelle collection.

```
#no punctuation and stopwords
stringDocsNoStopwords = transformDocToStringNoStopwords(documents)
tf_idf_noStopwords = createTFIDFMatrix(stringDocsNoStopwords)

#no stopwords/punctuation
dense_string_docsNoStopwords = tf_idf_noStopwords.todense()
U2, s2, V2 = np.linalg.svd(dense_string_docsNoStopwords, full_matrices=True)

#Coordonnées X,Y
coord_X2,coord_Y2 = getXAndYCoordinates(U2)

fig = plt.figure()
ax = fig.add_subplot(111)
plt.plot(coord_X2, coord_Y2, 'ro')
i = 0

while i < len(nameOfFiles):
    tag = nameOfFiles[i]
    xy = (coord_X2[i],coord_Y2[i])
    ax.annotate('%s'%tag, xy=xy, textcoords='offset points')
    i = i+1

plt.show()
```

5.3 (1 point)

Voici le graphe de type nuage de points suite à 5.2.



5.4 (5 points)

Je crois que cette manipulation a été bénéfique, car elle sépare les deux groupes en bas à droite (groupe “Mort de Charles” et groupe “Industrie des armes à feu”). En effet, on voit mieux la séparation entre ces deux topics, mais on peut voir une certaine similarité (les deux font référence à la mort, et les deux se passent aux États-Unis). Bref, ils sont encore relativement près, mais les documents dans chacun des groupes sont mieux regroupés ensemble.

Pour ce qui est du sujet de Pinochet, aucune différence significative est remarquée, sauf que les documents semblent mieux alignés en 1 ligne.

Pour le groupe central, on remarque l'absence du sujet de Matthew, quoi que celui-ci n’est pas très loin vers le haut. De plus, les sujets de la mort de Diana et de l’industrie de l’arme à feu sont similaires puisqu’ils parlent de violence et de meurtre, mais on voit qu’ils sont respectivement placés de part et d’autre d’un axe qui les sépare. De plus, les documents de ces topics semblent mieux alignés de façon linéaire.

Finalement, pour ce qui est du groupe au dessus, portant sur la greffe et la mort de Matthew Shepard, on voit qu’ils sont relativement près, mais qu’ils sont également placés de part et d’autre d’une ligne séparatrice. Ces 2 sujets se passent aux États-Unis et chacun des 6 documents contiennent le nom “Matthew” (dans un cas il s’agit du patient, et dans l’autre le nom photographe). C’est probablement ce qui explique leur proximité (les autres documents des autres topics ne contiennent probablement pas ce nom, alors la valeur TF-IDF de “Matthew” est grande dans les 2 topics.

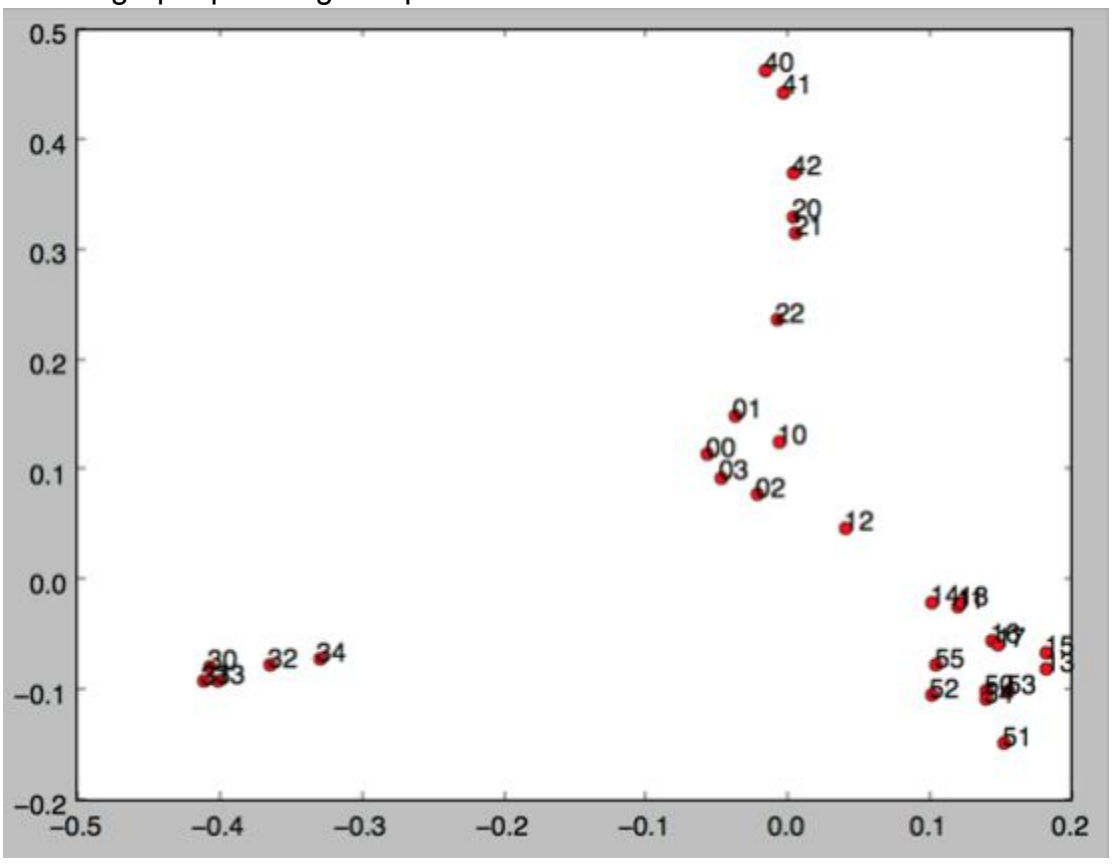
5.5 (4 points)

Je me suis basé sur le code modifié en 5.2 pour ajouter la lemmatisation.

```
def lemmatizeDocToStringNoStopwords(collection):
    cachedStopWords = stopwords.words("english")
    stringDocuments = []
    for d in documents:
        sen = ''
            #remove punctuation
            exclude = set(string.punctuation)
            s = ''.join(ch for ch in s if ch not in exclude)
            #remove stopwords & lemmatize
            s = ' '.join([lemmatizer.lemmatize(word) for word in s.split() if word not in
stopwords.words("english")])
            sen = sen+s
        stringDocuments.append(sen)
    return stringDocuments
```

5.6 (1 point)

Voici le graphique nuage de points fait à la suite des modifications en 5.5.



5.7 (4 points)

La différence par rapport au graphique en 5.3 est très minimale. Ce qui est un peu normal, puisque l'utilisation de `lemmatizer.lemmatize(word)` ne permet pas à WordNet de savoir s'il s'agit d'un nom, d'un adjectif ou d'une forme de verbe, alors la quantité de mots qu'il peut lemmatiser parfaitement est limitée. Il faudrait un algorithme plus poussé pour pouvoir déterminer en premier le type de mot, afin de donner un "indice" ou un "tag" à Wordnet pour qu'il lemmatise d'une façon plus efficace.

Dans mon cas, il n'y a pas vraiment eu d'améliorations en terme de séparation des clusters par topic, même que dans le cas du coin inférieur droit, la situation s'est empirée (c'est-à-dire que les topic 1 et 5 sont pratiquement indiscernables l'un de l'autre). Encore une fois, je crois que c'est dû à la lemmatisation non-efficace de `lemmatizer.lemmatize(word)`. Peut-être qu'une autre librairie de lemmatisation aurait mieux fonctionné.

5.8 (6 points)

Selon les profils de formes et de densités observés, je crois que DBSCAN ou encore OPTICS serait le plus approprié, car les clusters ont des densités relativement différentes et sont souvent formés selon une ligne. En effet, beaucoup de mesures de distance ou de similarité ne sont pas capables de suivre des formes ou lignes (connectivité), car ils ne mesurent souvent que le rapprochement.

QUESTION 6 (20 points) cours 7

6.1 (10 points)

$$P(w1|d2) = \sum_{z\varepsilon\{z1,z2,z3\}} P(w1|z) \times P(z|d2)$$

$$P(w1|d2) = P(w1|z1) \times P(z1|d2) \;+\; P(w1|z2) \times P(z2|d2) \;+\; P(w1|z3) \times P(z3|d2)$$

$$P(w1|d2) = (0.6 \times 0.4) \;+\; (0.1 \times 0.5) \;+\; (0.2 \times 0.1) \;=\; 0.31$$

La probabilité P(w1|d2) (qui pourrait se lire comme étant “la probabilité de retrouver le mot 1 étant donné le deuxième document) est donc de 31%.

6.2 (10 points)

Après “lemmatisation” manuelle et après avoir gardé seulement les noms et les adjectifs :

- D1 : chat chien gros (3 mots)
- D2 : chat gros chien heureux (4 mots)
- D3 : gros chat gros (3 mots)

Probabilité des mots

- chat 3/10
- chien 2/10
- gros 4/10
- heureux 1/10

En tout on a 10 mots au total.

Selon la “Bayes Rule”

$$P\,(hypothesis \mid data) = \frac{P(data \mid hypothesis)\cdot P(hypothesis)}{P(data)}$$

Donc pour P(heureux,chat|d2) on aurait :

$$P\,(heureux, chat \mid d2) = \frac{P(d2 \mid heureux,chat)\cdot P(heureux,chat)}{P(d2)} = \frac{1 \cdot 1/3}{1/3} = 1$$

Donc pour P(gros|d3) on aurait :

$$P\,(gros \mid d3) = \frac{P(d3 \mid gros)\cdot P(gros)}{P(d3)} = \frac{2/4 \cdot 4/10}{1/3} = 3/5 = 0.6$$

On assume que P(d2) et P(d3) sont de ⅓ , soit que chacun des 3 documents a une probabilité égale.

QUESTION 7 (28 points) cours 8

7.1 (8 points)

Premièrement, j'ai explosé les valeurs catégoricielles en autant de valeurs binaires. Par exemple, au lieu d'avoir un attribut "safety" à {low, med, high}, j'ai créé 3 valeurs binaires : safetyLOW, safetyMED et safetyHIGH, dont une seule des 3 peut être à 1 pour une voiture donnée. J'ai fait ça pour tous les attributs (donc "maint" en 4 variables binaires, "persons" en 3 variables binaires, etc.)

J'arrive à la matrice suivante :

		Voiture 1	
		1	**0**
Voiture 2	**1**	q=2	r=4
	0	s=4	t=11

Avec l'équation de distance symétrique :

$$d(obj1, obj2) = \frac{r+s}{q+r+s+t} = \frac{4+4}{2+4+4+11} = 0.380952381$$

7.2 (8 points)

Le coefficient de corrélation est donné par la formule suivante :

$$\hat{\rho}_{12} = \frac{\hat{\sigma}_{12}}{\hat{\sigma}_1 \cdot \hat{\sigma}_2} = \frac{\sum_{i=1}^{5}\left[(x_{i1} - \hat{\mu}_1)\cdot(x_{i2} - \hat{\mu}_2)\right]}{\sqrt{\sum_{i=1}^{5}\left[(x_{i1} - \hat{\mu}_1)^2\right]\cdot\sum_{i=1}^{5}\left[(x_{i2} - \hat{\mu}_2)^2\right]}}$$

où 1 et 2 font référence à “Sepal length” et “Sepal width” respectivement.

$$\hat{\mu}_1 = \frac{5.1 + 4.9 + 4.7 + 4.6 + 5.0}{5} = 4.86$$

$$\hat{\mu}_2 = \frac{3.5 + 3.0 + 3.2 + 3.1 + 5.4}{5} = 3.64$$

À l’aide d’un chiffrier Excel, je trouve :

$$\sum_{i=1}^{5}\left[(x_{i1} - \hat{\mu}_1)\cdot(x_{i2} - \hat{\mu}_2)\right] = 0.398$$

$$\sum_{i=1}^{5}\left[(x_{i1} - \hat{\mu}_1)^2\right] = 0.172 \quad \text{et} \quad \sum_{i=1}^{5}\left[(x_{i2} - \hat{\mu}_2)^2\right] = 4.012$$

On a donc :

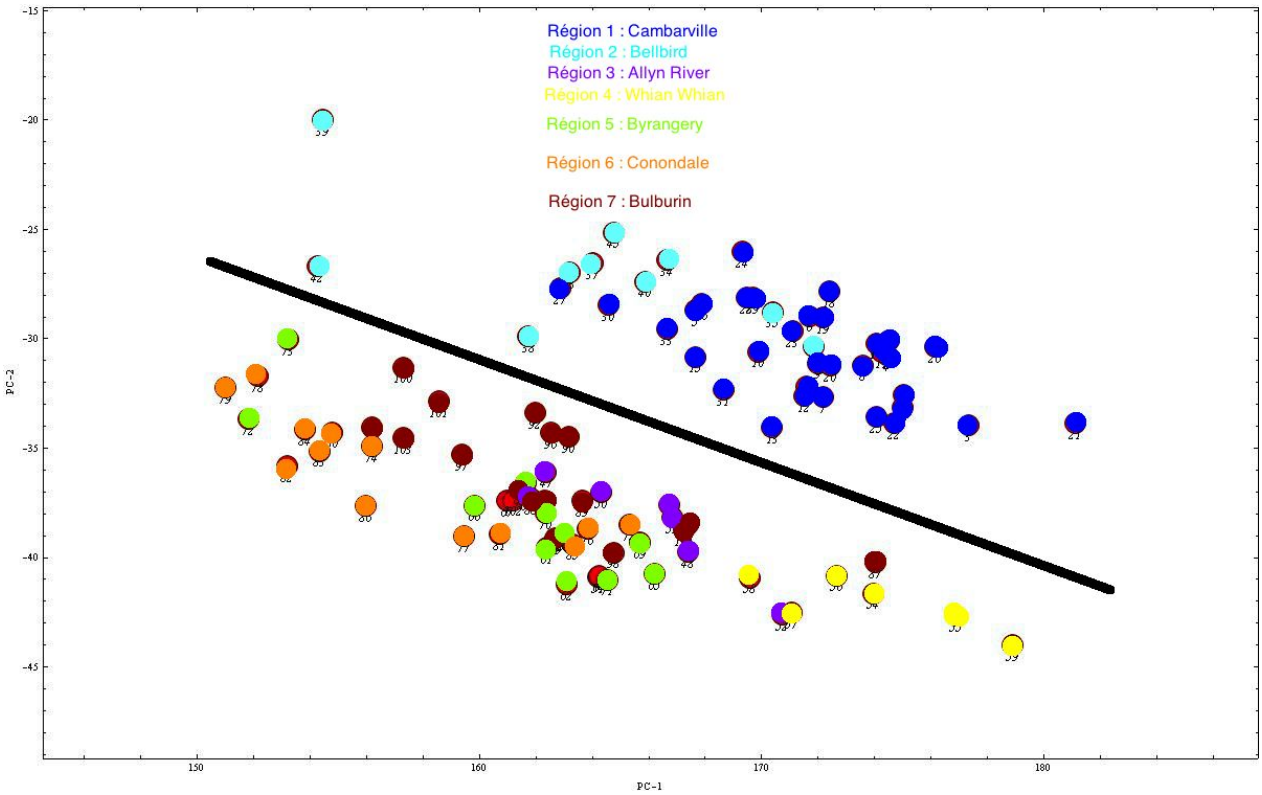
$$\hat{\rho}_{12} = \frac{\hat{\sigma}_{12}}{\hat{\sigma}_1 \cdot \hat{\sigma}_2} = \frac{0.398}{\sqrt{0.172 \cdot 4.012}} = 0.479113476$$

On trouve un coefficient de corrélation supérieur à zéro, ils sont donc positivement corrélés : la longueur de sépale tend à augmenter lorsque la largeur augmente et vis versa.

7.3 (12 points)

J'ai utilisé l'outil de PCA en ligne du site : http://www.morpho-tools.net/pca_online.html

J'ai formaté les données en enlevant les colonnes "Site", "Pop", "Sex" et "Age" mais j'ai laissé la colonne "Case" pour permettre de distinguer les spécimen par leur numéro, ce qui me permet ensuite de les colorer manuellement selon leur site. Après analyse, j'obtiens ceci :



PC axis	1	2	3	4	5	6	7	8
Eigenvalue	46.5833	24.0147	7.4924	4.6961	3.08907	2.55572	1.62028	1.21917
Variance (%)	50.5688	26.0693	8.13342	5.09787	3.35336	2.77438	1.7589	1.32347
Cum. Var. (%)	50.5688	76.638	84.7714	89.8693	93.2227	95.9971	97.756	99.0794

On remarque tout de suite que les opossums situés à l'extrême sud (Sites 1 et 2, soit Cambarville et Bellbird en Victoria) ont une morphologie assez distincte des autres possums plus au Nord. Ils tendent à avoir un score PC2 plus élevé et leur score PC1 se concentre majoritairement entre 160 et 175. Tel qu'en regardant les scores PC1 et PC2 d'un nouveau spécimen, il serait assez facile de dire s'il vient de Victoria ou non.

QUESTION 8 (28 points) cours 10

8.1 (2 points)

La réponse est **A** et **E**, soit **Pureté** et **F-mesure**, car elles comparent les clusters au "Ground Truth").

Pour ce qui est de la pureté, c'est le nombre de points d'un cluster qui appartiennent au Ground Truth, et pour la F-mesure il s'agit de calculer le rappel et la précision par rapport au Ground Truth (faux négatifs, vrais positifs, etc.).

8.2 (2 points)

Pour trouver la pureté des clusters, on observe la valeur maximale de chacun par rapport aux 3 Ground Truths. On fait la somme de ces valeurs maximales puis on divise par le nombre total.

C\T	**T1**	**T2**	**T3**	**Somme**
C1	20	**30**	10	60
C2	30	**40**	10	80
C3	0	0	**60**	60
mj	50	70	80	**200**

Pureté totale : $\frac{30+40+60}{200} = 0.65$

La pureté est donc de 65%.

8.3 (2 points)

8.4 (3 points)

8.5 (6 points)

(1)

Non.

Le point (2,3) a une distance minimale de 2 avec le cluster (2,1)

et le point (-0.5,0) a une distance minimale de ~1.11 avec le cluster (0,1).

(2)

Le Centroïde-moyen serait de ((0+2-2)/3 ; (3+1+2)/3) = (0,2).

(3)

Le centroïde-médian serait (2, 1) car il représente le point médian de ce cluster.

8.6 (9 points)

A. Fonction noyau polynomiale (h=2)

	y1	y2	y3	y4	y5	y6	y7	y8	y9	y10	y11	y12	y13
x1	1	1	1	1	1	1	1	1	1	1	1	1	1
x2	1	1	0	1	4	1	16	1	36	9	25	9	25
x3	1	1	1	1	1	1	1	1	1	1	1	1	1
x4	1	1	4	1	0	1	36	1	16	25	9	25	9
x5	1	1	1	1	1	1	1	1	1	1	1	1	1
x6	1	1	16	1	36	1	576	1	676	361	441	361	441
x7	1	1	1	1	1	1	1	1	1	1	1	1	1
x8	1	1	36	1	16	1	676	1	576	441	361	441	361
x9	1	1	1	1	1	1	1	1	1	1	1	1	1
x10	1	1	9	1	25	1	361	1	441	225	289	225	289
x11	1	1	25	1	9	1	441	1	361	289	225	289	225
x12	1	1	25	1	9	1	441	1	361	289	225	289	225
x13	1	1	9	1	25	1	361	1	441	225	289	225	289

B. Fonction base radiale gaussienne (faite avec excel)

(Écrit sous format scientifique pour sauver de l'espace)

1.00E+00	1.00E+00	9.99E-01	1.00E+00	9.99E-01	1.00E+00	9.80E-01	1.00E+00	9.80E-01	9.87E-01	9.87E-01	9.87E-01	9.87E-01
9.99E-01	9.99E-01	9.97E-01	9.99E-01	1.00E+00	9.99E-01	9.72E-01	9.99E-01	9.87E-01	9.80E-01	9.93E-01	9.80E-01	9.93E-01
1.00E+00	1.00E+00	9.99E-01	1.00E+00	9.99E-01	1.00E+00	9.80E-01	1.00E+00	9.80E-01	9.87E-01	9.87E-01	9.87E-01	9.87E-01
9.99E-01	9.99E-01	1.00E+00	9.99E-01	9.97E-01	9.99E-01	9.87E-01	9.99E-01	9.72E-01	9.93E-01	9.80E-01	9.93E-01	9.80E-01
1.00E+00	1.00E+00	9.99E-01	1.00E+00	9.99E-01	1.00E+00	9.80E-01	1.00E+00	9.80E-01	9.87E-01	9.87E-01	9.87E-01	9.87E-01
9.80E-01	9.80E-01	9.72E-01	9.80E-01	9.87E-01	9.80E-01	9.23E-01	9.80E-01	1.00E+00	9.37E-01	9.99E-01	9.37E-01	9.99E-01
1.00E+00	1.00E+00	9.99E-01	1.00E+00	9.99E-01	1.00E+00	9.80E-01	1.00E+00	9.80E-01	9.87E-01	9.87E-01	9.87E-01	9.87E-01
9.80E-01	9.80E-01	9.87E-01	9.80E-01	9.72E-01	9.80E-01	1.00E+00	9.80E-01	9.23E-01	9.99E-01	9.37E-01	9.99E-01	9.37E-01
1.00E+00	1.00E+00	9.99E-01	1.00E+00	9.99E-01	1.00E+00	9.80E-01	1.00E+00	9.80E-01	9.87E-01	9.87E-01	9.87E-01	9.87E-01
9.87E-01	9.87E-01	9.80E-01	9.87E-01	9.93E-01	9.87E-01	9.37E-01	9.87E-01	9.99E-01	9.50E-01	1.00E+00	9.50E-01	1.00E+00
9.87E-01	9.87E-01	9.93E-01	9.87E-01	9.80E-01	9.87E-01	9.99E-01	9.87E-01	9.37E-01	1.00E+00	9.50E-01	1.00E+00	9.50E-01
9.87E-01	9.87E-01	9.93E-01	9.87E-01	9.80E-01	9.87E-01	9.99E-01	9.87E-01	9.37E-01	1.00E+00	9.50E-01	1.00E+00	9.50E-01
9.87E-01	9.87E-01	9.80E-01	9.87E-01	9.93E-01	9.87E-01	9.37E-01	9.87E-01	9.99E-01	9.50E-01	1.00E+00	9.50E-01	1.00E+00

C.

La distance euclidienne représente la plus courte distance d'un point à l'autre (ligne droite), tandis que la distance de Manhattan représente la somme de la différence des deux coordonnées (|x1-x2|+|y1-y2|). La distance de Euclidienne sera toujours plus petite, cependant comme les données transformées sont très différentes, il est difficile de faire un parallèle entre les 2. Cependant, comme les différences numériques dans le A sont beaucoup plus grandes qu'en B, je crois qu'on pourrait voir des clusters très éloignés et plus distincts qu'en B. En effet, en B les clusters risquent d'être beaucoup plus rapprochés et difficiles à discerner, malgré le fait que la distance de Manhattan produise une valeur plus grande que l'euclidienne.

8.7 (2 points)

La valeur du seuil est telle qu'il y a deux grandes "vallées", il y aurait donc **2 clusters** identifiés.

8.8 (2 points)

Les clusters B et C seraient fusionnés en premier dû à leur distances relatives moins élevées que par rapport à l'autre cluster.

QUESTION 9 (20 points) cours 11

9.1 (2 points)

Les réponses sont B et D, car on peut les substituer l'un pour l'autre.
Par exemple, on peut dire "je me promène en voiture/véhicule" et dans les deux cas les phrases auront le même sens. On peut également dire "Je programme avec mon ordinateur/laptop" et les phrases auront encore le même sens.

9.2 (2 points)

La valeur IDF est "Inverse Document Frequency" ou encore "Fréquence de Document Inverse". Donc, un mot qui n'est pas très fréquent aura un score IDF élevé, tandis qu'un mot très fréquent aura un score IDF bas.

Dans ce cas, comme le mot "un" sera utilisé dans tous les documents (ou presque), il aura un score IDF bas. Le mot "apprentissage" apparaîtra bien moins souvent/fréquemment dans les documents que "un", alors il aura un score IDF plus élevé.

La réponse est donc A, soit "apprentissage".

9.3 (2 points)

Comme "un" apparaîtra toujours dans les documents (ou presque) son entropie sera de zéro ou très près de zéro.

Comme "apprentissage" n'est pas un mot qui apparaît souvent et plus aléatoirement, son entropie sera certainement plus élevée.

Donc la réponse est "X_{un}".

9.4 (2 points)

Soit : $Sim(dA,\ dB) = dA \cdot dB = dA_1 \times dB_1 + dA_2 \times dB_2 + ... + dA_n \times dB_n$

$= 0.1 * 0.2 + 0.5 * 0.4 + 0 * 0.3 + 0.4 * 0 + 0 * 0.1 + 0 * 0 = 0.22$

La similarité EOWC est donc : 0.22

9.5 (3 points)

$$0.5 * 0.1 + 0.5 * 0.4 = 0.05 + 0.2 = 0.25$$

La réponse est donc C:0.25

9.6 (4 points)

Selon la règle de Bayes

$$P(\theta_1|"ordinateur") = \frac{P("ordinateur"|\theta_1) \cdot P(\theta_1)}{P("ordinateur")} = \frac{0.1 \cdot 0.5}{0.25} = 0.2$$

$$P(\theta_2|"ordinateur") = \frac{P("ordinateur"|\theta_2) \cdot P(\theta_2)}{P("ordinateur")} = \frac{0.4 \cdot 0.5}{0.25} = 0.8$$

La réponse est donc A (0.2 et 0.8).

9.7 (5 points)