SD701 Exploration de grands volumes de données

Sujet : Classifications et visualisations de données en graphe sur des pages Facebook

Table des matières

I- Présentation du projet

1- Le jeu de données utilisé

Le jeu de données est tiré de la base de données du centre du machine learning et des systèmes intelligents de l'université de Californie, Irvine. Ce jeu de données, dont une partie est un graphe, est composé de trois fichiers :

- Le fichier *« musae_facebook_target.csv »* qui contient les nodes du graphe (id des pages Facebook) ainsi que leur type. Les pages sont classées en quatre types : *tvshow, governement, politician et company*. Il contient aussi les liens vers ces pages mais ceux-ci ne seront pas utilisés dans ce projet. Au total le fichier comporte 22470 nœuds.

- Le fichier *« musae_facebook_edges.csv »* qui contient les edges du graphe, ceux-ci représentant un like mutuel des pages. Le stockage est assez simple, sous la forme d'un tableau à deux colonnes, la première contenant l'id du Node de départ et le second de celui d'arrivée. Les edges n'apparaissant qu'une seule fois, et les id étant classés par ordre croissant si un edge part du point 5 vers le point 12235 alors une ligne contiendra 5 et 12235, mais la ligne 12235 ne sera pas explicitement écrite. Au total le fichier contient 170 002 edges.

- Enfin le dernier fichier *« musae_facebook_features.json »* contient un dictionnaire associant à chaque id le nombre de mots dans les différentes descriptions présentes sur la page.

2- Les objectifs de ce projet

Ce jeu de données permet d'aborder divers axes de l'exploration de données. J'ai décidé de me concentrer sur deux axes. La première partie sera donc dédiée à la visualisation de la donnée, quand la seconde essayera de développer une méthode de classification efficace des pages

par type. Ces deux parties sont toutefois tout à fait en lien l'une avec l'autre, la première permettant de donner des intuitions pour réaliser la seconde.

II- Visualisation de la donnée

1- Package utilisé pour la création du graphe initial

Afin de créer le graphe à partir des fichiers initiaux j'ai fait appel au package networkX. Celui-ci a l'avantage de permettre de stocker les données du graphe (nodes et edges) et d'en ressortir une disposition en 3 dimensions. Il est assez simple à prendre en main et permet de ressortir des coordonnées de points dans la dimension souhaitée en appliquant une force d'attraction entre les nœuds reliés par des edges. Il est cependant limité au niveau de la visualisation graphique à mes yeux. Ce package m'a donc principalement servi à calculer les positions dans l'espace des points, et à utiliser des algorithmes de partitionnement en communauté.

2- Visualisation 2D des données

Dans un premier un temps j'ai commencé la visualisation de mes données directement avec le package networkX, mais celui-ci se limite à la 2D. Les données affichées se présentent comme ci-dessous :

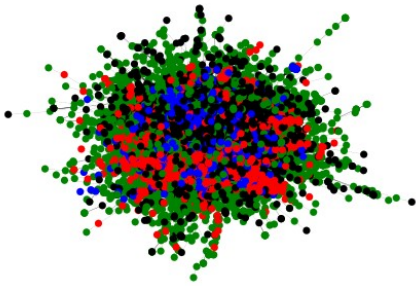


Figure 1: Graphe complet

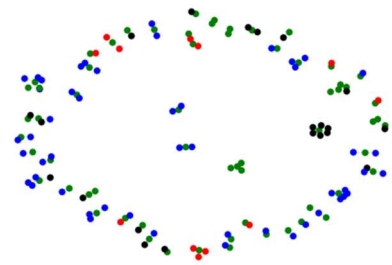
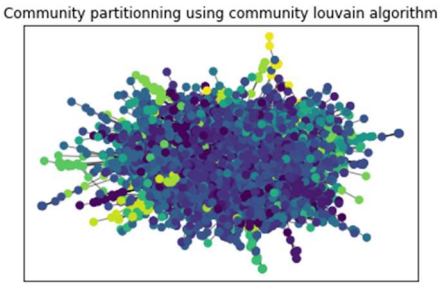
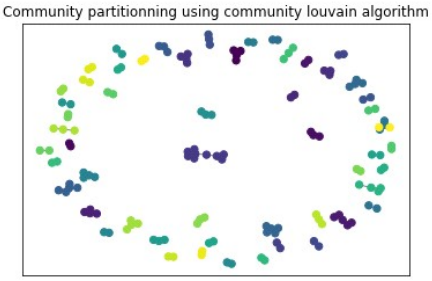


Figure 2: Graphe d'une partie des points sélectionnés aléatoirement

On peut voir sur ces premiers graphes qu'une partie des nœuds de même type sont regroupés en cluster grâce à l'amas de certaines couleurs. Pour autant il apparait que certains points se retrouvent tout de même assez isolés et l'on peut d'ores et déjà supposer qu'ils seront assez difficiles à classifier par la suite. Enfin le second permet de voir que les groupes présentent souvent une similarité de type.

3- Visualisation 2D d'un partitionnement en communauté





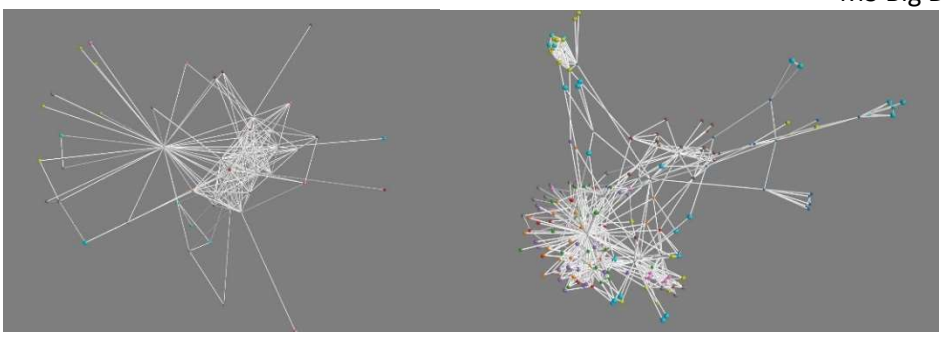
En utilisant l'algorithme de Louvain pour la détection de communauté nous obtenons le partitionnement représenté ci-dessus. Ce genre de méthode pourrait être utile à la détection de communautés spécifiques dans les pages (par exemple pour les politiciens le parti) mais ici on peut voir que les communautés ne permettent pas de distinguer facilement les types des pages. Cela reste toutefois logique, le partitionnement en communauté ne prenant en compte que les edges, il est évident, étant donné les types des pages étudiées, que celles-ci présentent des liens avec des pages d'un autre type.

4- Visualisation 3D

Les graphes précédents avaient l'avantage d'être assez simples à produire mais l'inconvénient de ne pas représenter de façon lisible les données du fait du nombre important de points. Une représentation en 3 dimensions s'avérait plus judicieuse pour ce type de données. Toutefois le package NetworkX ne contenait pas d'outils appropriés pour ce genre de tâches. J'ai donc essayé deux méthodes pour explorer la donnée en 3 dimensions.

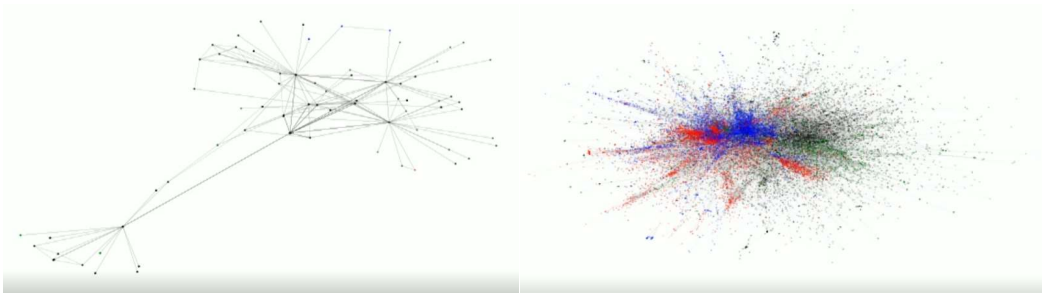
a- Utilisation de mayavi

Le premier package que j'ai utilisé est mayavi. Ce package a l'avantage de permettre de créer des graphes 3D dynamiques, et permet donc de faire tourner ou de zoomer dans le graphe facilement. Il permet aussi d'enregistrer ces graphes en format .x3d si souhaité. Cependant ce package possède deux principaux inconvénients. Le premier est qu'il ne permet pas beaucoup d'options sur les graphes, ici il était par exemple impossible de choisir les couleurs des points en fonction de leur type ce qui est un inconvénient majeur dans notre cas. Le second inconvénient est d'ordre plus hardware, bien que la création d'une partie raisonnable des nœuds et edges du graphe se réalise rapidement, le passage à l'échelle est totalement impossible pour mon pc ou ceux de l'école avec la totalité des données du graphe. En effet, l'affichage de 10% des points du graphe entraine une consommation de ram de près de 15Go. Voici quelques captures d'écran de graphes obtenus à l'aide de ce package.



b- Utilisation de matplotlib 3D

Pour pouvoir afficher la totalité du graphe et avoir la main sur de plus nombreux paramètres j'ai eu recours à l'utilisation de la librairie assez classique en création de graphe, matplotlib qui dispose de possibilité de graphes en 3D. Ici les possibilités de personnalisation des graphes sont bien plus élevées et la ram nécessaire à l'affichage de la totalité du graphe devient raisonnable. Cependant les graphes ne sont pas dynamiques, pour résoudre ce problème et mieux visualiser les nœuds et leurs positions dans l'espace j'ai enregistré le graphe sous différents angles puis ai associé les images pour créer une vidéo où le graphe tourne sur lui-même, celles-ci ne peuvent être insérées dans un pdf et sont disponibles dans les annexes. En voici des captures, la première en représentant les liens jusqu'au 3ème degré en partant d'un point au hasard, le second contenant les points en allant jusqu'au 8ème degré (95% des données sont alors présentes).



Comme sur le premier graphe on peut voir apparaitre des zones de couleurs assez distinctes représentant un réseau de edges important entre des pages du même type, mais aussi des nodes peu reliés au reste du réseau et assez éloignés dont la classification sera la plus complexe.

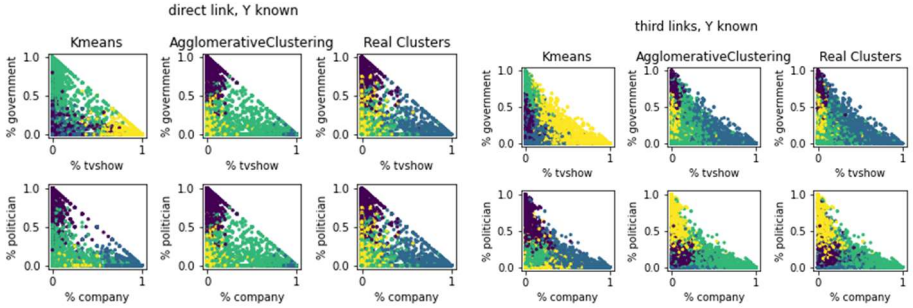
5- Première idée de données explicatives pour la classification

a- Les données explicatives

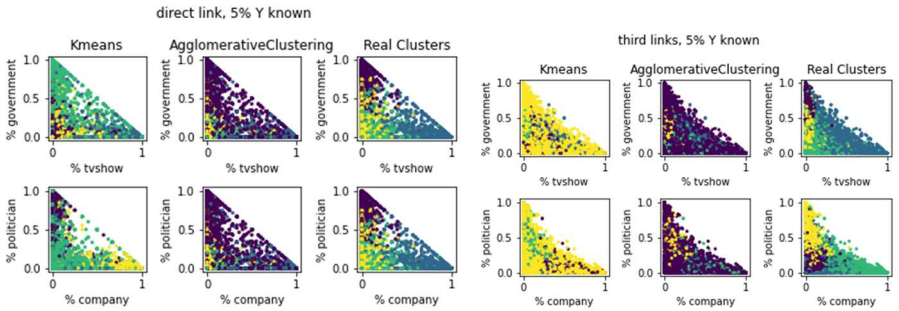
Comme nous avons pu le voir les points d'un même type ont tendance à se regrouper en groupes assez proches même si certains peuvent se retrouver assez éloignés ou liés à d'autres groupes. Pour la classification, n'ayant que peu d'autres informations qualitatives ou quantitatives sur les pages en dehors des graphes, j'ai décidé de prendre comme connu le type d'une partie des pages, ici 5%. L'objectif de la classification va alors être de chercher à classifier les autres pages, partant de ces 5% connus. Pour ce faire j'ai donc créé une matrice X contenant pour chaque id le nombre de liens au 1er degré ainsi que le pourcentage de chaque type de nœuds auxquels il est lié si leur type est connu. J'ai ensuite rajouté les même informations au second et troisième degré, afin d'obtenir plus d'informations pour les points excentrés du graphe. J'ai alors cherché à visualiser si l'information était suffisante pour créer des clusters dans les données via la méthode des Kmeans et de clustering agglomératif.

b- Clustering en 2D

J'ai d'abord représenté les clusters en 2 dimension, en fonction du % de chaque type de page lié au noeud, et en affichant les résultats obtenus par les différentes méthodes face aux valeurs réelles. J'ai effectué cela sur une matrice X construite en prenant toutes les valeurs de Y connues, et une seconde ne prenant que 5% des valeurs de Y connues. J'ai affiché ci-dessous les résultats par rapport au premier et second lien. Les couleurs ne sont pas en relation les unes avec les autres entre les graphes des différentes méthodes.

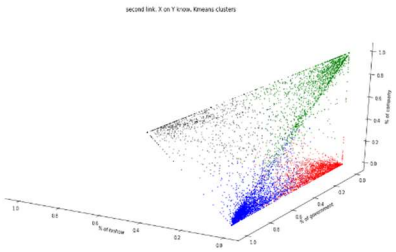


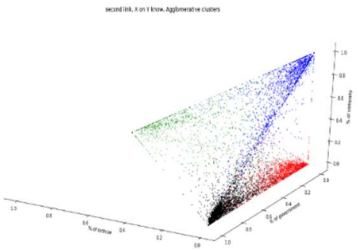
On peut voir que dans le cas où X est créée sur Y connu alors la répartition des clusters est assez précise et l'on peut donc penser que la matrice X apporte une information de qualité pour la classification de nos nœuds.

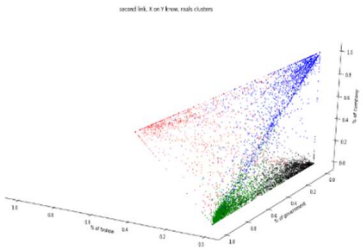


Dans le cas d'une matrice X basée sur 5% de Y connu uniquement toutefois les résultats sont plus mitigés, il apparait que les types sont moins identifiables mais ne sont pour autant pas placé de manière totalement aléatoire.

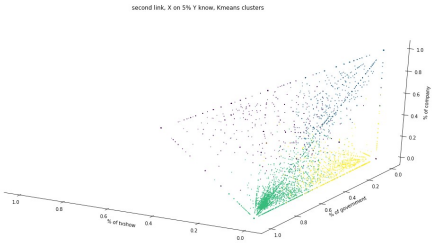
c- Clustering en 3D

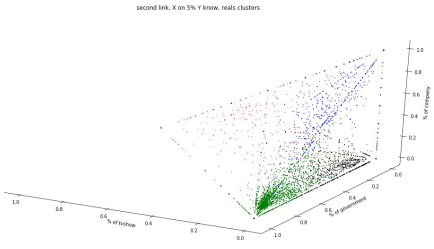


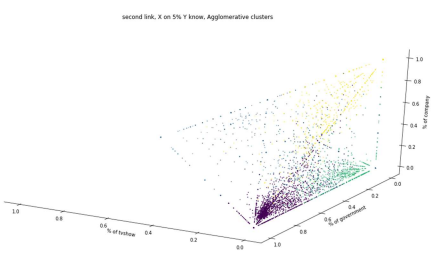




Afin de rendre les résultats plus visuels j'ai décidé d'appliquer la même méthode mais de créer les graphes en 3D. Ceux-ci permettent de représenter les 4 types de pages assez facilement sur un même graphe, le dernier type se déduisant par l'absence dans les autres zones. Pour Y connu la distinction entre les types se voit très aisément. On voit que les données se concentrent principalement dans les angles du graphe et qu'une partie assez contenue se situe à mi-chemin sur les arrêtes.







Ici voici les graphes pour 5% de Y connu uniquement (les arrêtes étant toujours les pourcentages de type de page au second degré). Il apparait qu'en trois dimensions la séparation entre les zones des points parait plus lisible et que la matrice X semble apporter des informations assez significatives pour essayer de classifier les nœuds sur la base de ces informations.

III- Classifications des données

1- Classification par arbre de décision

a- Les données explicatives

Comme détaillé ci-dessus après l'analyse graphique les données explicatives dans la matrice X sont les nombres de edges et le pourcentage de chaque type de page (pour les pages dont le type est connu évidemment). A cela s'ajoute la communauté à laquelle le nœud est associé via le partitionnement par l'algorithme de Louvain. Enfin quelques valeurs tirées du document de features telles que le nombre de mots moyen, la variance de ce nombre de mots, le nombre de descriptions, le nombre total de mots ainsi que la moyenne divisée par la variance ont été ajoutés.

b- Le modèle de base utilisé

Etant donné notre type de données, d'une part des données à expliquer qualitatives, et des données explicatives numériques, un arbre de décision m'a semblé judicieux. Le modèle a donc été entrainé par cross validation afin d'en définir les paramètres sur un set de train puis appliqué sur un set de tests. J'ai initialement utilisé un train set représentant 55% des valeurs, un test en représentant 45%, les 5 derniers pourcents étant les id dont le type est connu. Le résultat est alors assez encourageant, les résultats obtenus sur la partie train en cross validation présentent une précision de 83.6, qui est confirmée sur la partie de test (82 et 81 respectivement dans le cas où la communauté obtenue par l'algorithme de Louvain n'est pas prise en compte).

```
0.8355727408816053 {'criterion': 'gini', 'max depth': 8, 'min impurity decrease': 0.0, 'min samples leaf': 3, 'min samples split': 3, 'splitter': 'best'}

              precision    recall  f1-score   support

     company       0.83      0.83      0.83      2600
  government       0.85      0.88      0.86      2760
  politician       0.85      0.87      0.86      2281
      tvshow       0.76      0.68      0.72      1346

    accuracy                           0.83      8987
   macro avg       0.82      0.81      0.82      8987
weighted avg       0.83      0.83      0.83      8987
```

c- L'amélioration du modèle

En partant du modèle initial, j'ai cherché à améliorer celui-ci. Pour ce faire, je suis parti du fait que la matrice X de variables explicatives n'était créée qu'en utilisant les 5% des informations connues des valeurs réelles. Toutefois, le résultat du modèle nous renvoie des informations sur les autres points présents dans notre set de données. Ainsi j'ai donc changé le modèle pour itérer plusieurs fois tout en ajoutant aux données initiales connues les valeurs prédites à l'itération précédente de Y. A chaque itération sur la zone de test, l'arbre et ses paramètres sont obtenus par cross validation. On peut ainsi voir que sur la partie train cross validée la précision du modèle converge vers 91%, et vers 87% sur la partie test ce qui représente une amélioration non négligeable de la précision. Les résultats sont affichés ci-dessous :

```
              precision    recall  f1-score   support

     company       0.83      0.83      0.83      2600
  government       0.85      0.88      0.86      2760
  politician       0.85      0.87      0.86      2281
      tvshow       0.76      0.68      0.72      1346

    accuracy                           0.83      8987
   macro avg       0.82      0.81      0.82      8987
weighted avg       0.83      0.83      0.83      8987

current classification report:
              precision    recall  f1-score   support

     company       0.87      0.88      0.87      2600
  government       0.89      0.88      0.89      2760
  politician       0.85      0.90      0.87      2281
      tvshow       0.81      0.75      0.78      1346

    accuracy                           0.86      8987
   macro avg       0.86      0.85      0.85      8987
weighted avg       0.86      0.86      0.86      8987

current classification report:
              precision    recall  f1-score   support

     company       0.87      0.88      0.88      2600
  government       0.89      0.89      0.89      2760
  politician       0.87      0.90      0.89      2281
      tvshow       0.82      0.75      0.78      1346

    accuracy                           0.87      8987
   macro avg       0.86      0.86      0.86      8987
weighted avg       0.87      0.87      0.87      8987

current classification report:
              precision    recall  f1-score   support

     company       0.85      0.89      0.87      2600
  government       0.89      0.89      0.89      2760
  politician       0.88      0.89      0.89      2281
      tvshow       0.81      0.73      0.77      1346

    accuracy                           0.87      8987
   macro avg       0.86      0.85      0.85      8987
weighted avg       0.87      0.87      0.86      8987
```

```
starting iteration 0 on training part
0.8355727408816053 {'criterion': 'gini', 'max_depth': 8, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 3, 'min_samples_split': 3, 'splitter': 'best'}
starting iteration 1 on training part
0.8841236989859065 {'criterion': 'gini', 'max_depth': 7, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'best'}
starting iteration 2 on training part
0.8976370558674802 {'criterion': 'gini', 'max_depth': 7, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'}
starting iteration 3 on training part
0.9005503320721011 {'criterion': 'gini', 'max_depth': 9, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 4, 'min_samples_split': 4, 'splitter': 'random'}
starting on test set
```

En utilisant l'ensemble du set d'entrainement comme connu dans la partie de test, la précision de la classification atteint même 92%.

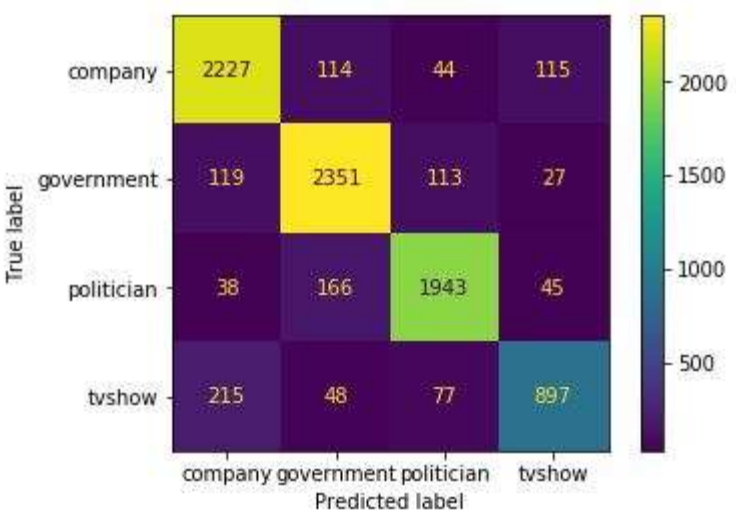
d- Analyse de l'importance des features

Les tableaux sont disponibles dans le dossier sous /images projet/Features importance. Afin de regarder l'importance des features j'ai utilisé deux méthodes. La première consiste à appliquer un modèle Lasso sur la première et seconde itération après avoir transformé les types en dummy variables afin d'identifier l'importance de chaque features pour chaque type de page. Cela permet de confirmer le lien logique entre le type d'une page et la répartition des types de pages auxquelles elle est liée. De plus ce lien se retrouve à chacun des degrés et non uniquement au premier. Par ailleurs on voit aussi que la plupart des features extraites du fichier de features ne présentent pas d'intérêt, et que seul le nombre de descriptions apportait une information significative. Enfin l'information apportée par le nombre de liens des points n'est pas très significative. Cela permet de voir que pour une page celle-ci sera majoritairement liée à des pages du même type, mais que le nombre de liens qu'elle aura, et que la taille de ses descriptions ne sont pas des éléments permettant de différencier le type des pages.

Dans un second temps j'ai récupéré l'importance des features sur l'arbre de décision pour chaque itération (cas où 5% connu). Cela permet de voir que pour la première itération que la majorité des variables sont utilisées comme les communautés mais aussi les features provenant du

fichier de features, ceci à l'exception des liens au premier degré. Cependant on observe que celles utilisées dans les itérations suivantes changent, les liens au premier degré prennent plus d'importance car plus nombreux, et la part des communautés et des informations du fichier de features devient négligeable.

e- Analyse des résultats

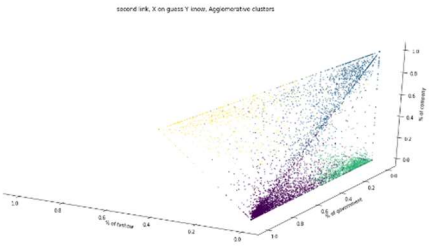


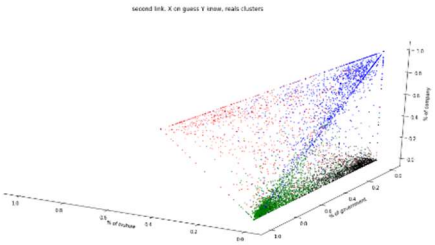
La précision du modèle final peut être considérée comme assez bonne. En effet le taux d'erreur est inférieur à 10% pour une classification dans 4 classes. En effet les classes sont assez équilibrées, la classe la plus représentée étant « gouvernement » qui représente 30% des observations. En comparaison d'une classification uniquement par la classe la plus représentée, le taux d'erreur est divisé par 7.

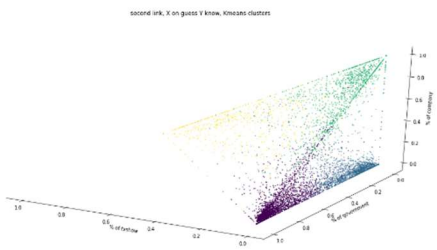
En observant la matrice de confusion toutefois il apparait quand même que la classe tv-show qui est la moins représentée dans le dataset (15% des observations) montre un taux d'erreur plus important que les autres catégories.

f- Visualisation 3D des résultats

g- Clusters obtenus sur la matrice X déduite après une itération







En observant la répartition des points en fonction de la matrice X obtenue après plusieurs renforcements du modèle on peut voir que les points affichent une répartition où les clusters sont plus espacés selon les variables, et qu'ainsi la matrice X déduite apporte une meilleure information pour notre classification. Cela permet de confirmer les valeurs numériques observées dans les résultats de la classification.

2- Classification par propagation de label

a- Présentation de la propagation de label

Afin de classifier les données il est ici possible d'utiliser plusieurs méthodes. L'une des plus intuitive pour des données en graphes peut-être la propagation de label. Pour ce faire j'ai décidé de ne pas m'aider d'un package existant mais plutôt d'utiliser les concepts uniquement afin de recréer un classifier moi-même.

La propagation de label se base sur un principe assez simple, et peut-être effectuée de façon un peu différente tout en gardant l'idée générale. Celle-ci consiste à se déplacer dans l'arbre et d'utiliser l'information des nœuds précédents ou des nœuds adjacents pour classifier chaque nœud au fur et à mesure.

En cela cette méthode se rapproche de la méthode des K plus proches voisins, mais diffère dans le sens où les voisins ne sont pas représentés par une distance dans l'espace mais par les liens du graphe.

b- La méthode utilisées et les variantes développées

Comme pour les K plus proche voisins plusieurs variantes sont possibles, la plus simple étant d'utiliser une pondération uniforme aux voisins. Toutefois l'utilisation d'une fonction de pondération selon la distance est parfois judicieuse (dans le cas où les classes sont à valeur réelle évidemment). Ici l'idée a été la même, mais au lieu de prendre une distance prendre la profondeur du lien (1er, 2nd etc.. degré) et le type du voisin.

Pour réaliser celle-ci j'ai commencé par une première version assez simple. Sur la partie d'entrainement du modèle une matrice de passage de taille 4x4 était créée, contenant la probabilité de passage d'un type à un autre. Ensuite en partant de points connus et choisis aléatoirement, l'algorithme parcourt le graphe et affecte aux points une probabilité d'appartenir à chaque type.

J'ai ensuite amélioré cette première version en y ajoutant une phase de renforcement. Celle-ci sélectionne d'abord les points dont la probabilité la plus grande d'appartenir à un type est les plus faible (ceux qui sont les moins bien classifiés). Pour chacun de ces points et toujours à l'aide de la matrice de passage la probabilité est recalculée en utilisant l'ensemble des voisins.

Enfin dans un troisième temps et sur la dernière version dans le code un paramètre depth est disponible. Celui-ci permet de réaliser la même tâche mais en prenant les liens au depth—ième degré. La matrice de passage est donc une matrice 4x4x4 pour depth=2, 4x4x4x4 pour depth=3 etc..

Sur cette dernière version plusieurs paramètres sont donc à choisir pour la classification : la profondeur utilisée, le pourcentage de valeur renforcée ainsi que le nombre de renforcement utilisé. Une fonction de cross validation sur la partie d'entrainement a donc été créée. Celle-ci fonctionne sous le même principe que la fonction GridSearchCV de sklearn. Cependant son utilisation dans un graphe possédant de nombreux edges pour chaque point, et une profondeur importante (>=3 dans notre cas) s'avère extrêmement couteuse en temps de calcul (peu monter à 1 semaine). En effet, le nombre de liens au 3eme degré pour chaque point est bien plus élevé qu'au 1er et 2nd degré, et ralentit fortement la phase de renforcement.

A la fin tous les points ne sont cependant pas classifiés, en effet certains points dans l'échantillon de test ne présentent tout simplement pas de lien. D'autres points présentent eux des probabilités à la fin de la classification qui ne permettent pas de voir la prédominance réel d'un type. Le nombre de points non classifiés est aussi affiché en output, en affichant d'abord le pourcentage total non classifié, puis le pourcentage de ceux dont aucune classification n'aurait été possible (aucun lien).

c- Les résultats obtenus

Ces résultats ont été obtenus sans faire appel à la fonction de crossvalidation par gridsearch, celle-ci étant très couteuse en temps de calcul. J'ai fait tourner pour un niveau de profondeur utilisé de 1 et (les résultats sont sur la page suivante : à gauche pour 1 profondeur, à droite pour 2)

On voit que les résultats des classifications sont assez bons avec des valeurs de précision s'établissant entre 76% et 92%. Les valeurs obtenues avec une profondeur de 1, obtenus dans des temps beaucoup plus courts, semblent être meilleures. Cependant ces résultats sont en trompe l'œil, en effet il apparait que pour une profondeur de 1 la proportion de points non classés est beaucoup plus importante. Pour 5% de valeurs connues 13.8% ne sont pas classées avec une profondeur de 1 contre seulement 3.6% avec une profondeur de 2. Le temps d'exécution toutefois est près de 15 à 60 fois supérieures avec une profondeur de 2.
L'utilisation de cette méthode à l'aide d'une profondeur élevée ne se montre donc que peu efficace en comparaison du modèle développé à l'aide d'arbres de décision précédemment, cependant dans le cas où une majorité des points est connues, ou dans le cas où l'on souhaite classifier facilement et rapidement une partie des points l'utilisation de ce modèle avec une profondeur de 1 est judicieux car bien plus rapide.

```
pourcentage de valeurs connus utilisé: 0.05
Creating the edges dictionnary
creating the passage matrix
launching prediction, it can take some time for big graph
starting spreading
doing reinforcment number: 0
doing reinforcment number: 1
doing reinforcment number: 2
13.818424566088117% of the points can't be classified at all due to edges and parameters selected
12.854481473814944% of points totaly unclassifiable
              precision    recall  f1-score   support

     company       0.91      0.73      0.81      1128
  government       0.79      0.95      0.87      2405
  politician       0.91      0.84      0.87      1682
      tvshow       0.90      0.70      0.79       606

    accuracy                           0.85      5821
   macro avg       0.88      0.81      0.83      5821
weighted avg       0.86      0.85      0.85      5821

Execution time for this pct_used: 75.23388910293579 seconds
pourcentage de valeurs connus utilisé: 0.1
Creating the edges dictionnary
creating the passage matrix
launching prediction, it can take some time for big graph
starting spreading
doing reinforcment number: 0
doing reinforcment number: 1
doing reinforcment number: 2
7.4321317311971752% of the points can't be classified at all due to edges and parameters selected
6.4628482972136222% of points totaly unclassifiable
              precision    recall  f1-score   support

     company       0.89      0.84      0.86      1600
  government       0.82      0.94      0.88      2523
  politician       0.92      0.84      0.87      1819
      tvshow       0.87      0.75      0.81       792

    accuracy                           0.87      6734
   macro avg       0.88      0.84      0.86      6734
weighted avg       0.87      0.87      0.86      6734

Execution time for this pct_used: 73.84131669998169 seconds
pourcentage de valeurs connus utilisé: 0.2
Creating the edges dictionnary
creating the passage matrix
launching prediction, it can take some time for big graph
starting spreading
doing reinforcment number: 0
doing reinforcment number: 1
doing reinforcment number: 2
4.583889630618603% of the points can't be classified at all due to edges and parameters selected
3.5261896610749743% of points totaly unclassifiable
              precision    recall  f1-score   support

     company       0.95      0.83      0.88      1734
  government       0.82      0.95      0.88      2579
  politician       0.92      0.87      0.89      1934
      tvshow       0.92      0.86      0.89       905

    accuracy                           0.89      7152
   macro avg       0.90      0.88      0.89      7152
weighted avg       0.89      0.89      0.89      7152

Execution time for this pct_used: 80.9761712551117 seconds
pourcentage de valeurs connus utilisé: 0.5
Creating the edges dictionnary
creating the passage matrix
launching prediction, it can take some time for big graph
starting spreading
doing reinforcment number: 0
doing reinforcment number: 1
doing reinforcment number: 2
2.6924788607031 6% of the points can't be classified at all due to edges and parameters selected
1.538559348973234% of points totaly unclassifiable
              precision    recall  f1-score   support

     company       0.96      0.87      0.91      2165
  government       0.86      0.96      0.91      2686
  politician       0.94      0.90      0.92      2100
      tvshow       0.92      0.88      0.90      1081

    accuracy                           0.91      8032
   macro avg       0.92      0.90      0.91      8032
weighted avg       0.91      0.91      0.91      8032

Execution time for this pct_used: 116.29387402534485 seconds
```

```
pourcentage de valeurs connus utilisé: 0.05
Creating the edges dictionnary
creating the passage matrix
launching prediction, it can take some time for big graph
starting spreading
doing reinforcment number: 0
doing reinforcment number: 1
doing reinforcment number: 2
3.6381842456608813% of the points can't be classified at all due to edges and parameters selected
3.3843924653280889% of points totaly unclassifiable
              precision    recall  f1-score   support

     company       0.91      0.62      0.74      1952
  government       0.64      0.95      0.77      2621
  politician       0.87      0.72      0.79      2142
      tvshow       0.82      0.60      0.69      1008

    accuracy                           0.76      7723
   macro avg       0.81      0.72      0.75      7723
weighted avg       0.80      0.76      0.76      7723

Execution time for this pct_used: 975.3733003139496 seconds
pourcentage de valeurs connus utilisé: 0.1
Creating the edges dictionnary
creating the passage matrix
launching prediction, it can take some time for big graph
starting spreading
doing reinforcment number: 0
doing reinforcment number: 1
doing reinforcment number: 2
1.9359145527369825% of the points can't be classified at all due to edges and parameters selected
1.6834365325077 4% of points totaly unclassifiable
              precision    recall  f1-score   support

     company       0.93      0.73      0.82      2186
  government       0.67      0.97      0.79      2632
  politician       0.91      0.71      0.80      2188
      tvshow       0.88      0.68      0.76      1057

    accuracy                           0.80      8063
   macro avg       0.85      0.77      0.79      8063
weighted avg       0.83      0.80      0.80      8063

Execution time for this pct_used: 1623.8396124839783 seconds
pourcentage de valeurs connus utilisé: 0.2
Creating the edges dictionnary
creating the passage matrix
launching prediction, it can take some time for big graph
starting spreading
doing reinforcment number: 0
doing reinforcment number: 1
doing reinforcment number: 2
1.1682242990654206% of the points can't be classified at all due to edges and parameters selected
0.8986648408079424% of points totaly unclassifiable
              precision    recall  f1-score   support

     company       0.95      0.76      0.84      2350
  government       0.71      0.97      0.82      2643
  politician       0.91      0.82      0.86      2252
      tvshow       0.91      0.71      0.80      1134

    accuracy                           0.83      8379
   macro avg       0.87      0.81      0.83      8379
weighted avg       0.86      0.83      0.84      8379

Execution time for this pct_used: 2656.334876537323 seconds
pourcentage de valeurs connus utilisé: 0.5
Creating the edges dictionnary
creating the passage matrix
launching prediction, it can take some time for big graph
starting spreading
doing reinforcment number: 0
doing reinforcment number: 1
doing reinforcment number: 2
0.44503782821539 83% of the points can't be classified at all due to edges and parameters selected
0.25430733040879905% of points totaly unclassifiable
              precision    recall  f1-score   support

     company       0.96      0.81      0.88      2470
  government       0.76      0.98      0.85      2668
  politician       0.94      0.84      0.88      2284
      tvshow       0.91      0.80      0.85      1202

    accuracy                           0.87      8624
   macro avg       0.89      0.86      0.87      8624
weighted avg       0.88      0.87      0.87      8624

Execution time for this pct_used: 5903.747905731201 seconds
```