TP 8

Républicains vs démocrates

Un jeu de données peut se mettre en python dans un "data frame" à la R : c'est une matrice dans lequel chaque ligne contient une donnée (un individu) et chaque colonne un attribut (une observation); nous utiliserons la librairie pandas, disponible en faisant:

```
>>> import pandas
```

On va chercher à déterminer automatiquement le “camp” d’un sénateur américain en fonction des lois pour lequel il a voté. Pour cela on va se baser sur une base de données qui décrit les votes des sénateurs et leur camp.
La base de donnée est *vote.csv*. Pour la charger, il suffit de faire (une seule ligne, et attention aux guillemets):

```
vote=pandas.read_csv(‘http://www.irit.fr/~Philippe.Muller/Cours/vote.csv’,index
_col=’id’)
```

vote est alors une matrice où chaque colonne porte un nom. Chaque ligne représente un sénateur. Les 16 premières colonnes représentent des votes à des lois (“y” pour “yes”, “n” pour “no” ou NA quand il n’a pas voté) et la dernière le camp (republican ou democrat).
Une colonne s’appelle une *série* de valeurs.
Vous avez un résumé des données en tapant juste vote dans l’interpréteur ou en l’affichant avec print vote.

Voici quelques méthodes et attributs utiles :

- les attributs (= les noms des colonnes) sont obtenus via `vote.columns`
- le nom de la colonne contenant le camp des sénateur est “Class”
- Pour récupérer le vote du 4eme sénateur sur la loi “crime”, il suffit d’appeler: `vote.get_value(4,’crime’) ou vote[‘crime’][4]`
- Pour récupérer tous les votes sur la loi crime, il suffit de faire `vote[‘crime’]`
- Pour récupérer tous les votes du sénateur numéro 4, on utilise `vote.ix[4]`
- On peut résumer certaines données, v.groupby(‘crime’).size() donne par exemple un résumé des valeurs de la colonne ‘crime’ sur l’ensemble des lignes (sénateurs). On peut ensuite faire un histogramme avec le module de visualisation pylab:

```
import pylab
a=vote.groupby(['crime']).size()
x=a.plot(kind='bar')
pylab.show()
```

- Une condition sur une série de valeurs est aussi une série de valeurs, essayez :

```
vote['Class']=='republican'
```

- On peut aussi sélectionner les valeurs d'une colonne ou du data frame selon une condition sur une autre colonne:
  ```
  vote['crime'][vote['Class']=='republican']
  vote[(vote['Class']=='republican') & (vote['crime']=='y')]
  ```
- Enfin, on peut itérer sur les colonnes de data frames avec un 'for', et sur les lignes associées à une colonne aussi avec un 'for'

Analyse des données

Pour chaque question, faire attention aux nom attribué

- combien y a t-il de sénateurs ?
- Quelle est le pourcentage de républicains et de démocrates ?
- Quel pourcentage de démocrates et de républicains ont voté yes au vote *duty-free-exports* ?
- Quel est le vote qui à reçu le plus de *yes* ?
- Quel est celui qui a reçu le plus de *no* ?

Classement à partir d'un attribut

On cherche maintenant à déterminer la loi qui classe le mieux les camps. Cette loi est celle pour laquelle une majorité des démocrates a répondu *yes* et une majorité des républicains a répondu *no*.

- Faire une fonction qui, à partir du nom d'une loi, renvoie le pourcentage de démocrates (parmi l'ensemble des démocrates) qui ont voté *yes* à cette loi
- Faire une fonction qui, à partir du nom d'une loi, renvoie le pourcentage de républicain(parmi l'ensemble des républicain) qui ont voté no à cette loi
- La loi qui permet le plus facilement de déterminer les camps est celle pour laquelle la somme des deux précédentes pondérée par le pourcentage de chaque camp est maximal. Trouver cette loi
- Refaire la même chose en cherchant la loi pour laquelle une majorité des démocrate a répondu *no* et une majorité des républicain à répondu *yes*.

Classement à partir des 10 plus proches voisins

On va chercher à prédire le camp d'un sénateur à partir des ses votes à l'aide de la méthode des 10 plus proches voisins. On définit une distance entre les sénateurs, et on cherche les dix sénateurs les plus proches d'un sénateur; La valeur prédite pour ce sénateur est celle la plus représentée parmi ses 10 voisins.

Questions

- Faire une fonction qui calcule la "distance" entre 2 sénateurs (le pourcentage de vote différent).
- Faire la fonction qui calcule les 10 plus proches voisins d'un sénateur (il ne peut pas

être lui même dans ses plus proches voisin).

- Étant donnés les 10 plus proches voisins d'un sénateur, donner le camp (démocrate ou républicain) prédit pour ce prédit pour ce sénateur.
- Prédire le camp pour chaque sénateur et calculer le taux de reconnaissance. (pourcentage d'exemple pour lesquelles le camp est bien prédit)
- On peut changer le nombre de voisins. Refaire la méthode avec un nombre de voisins différent. Trouver le nombre de voisins optimal.