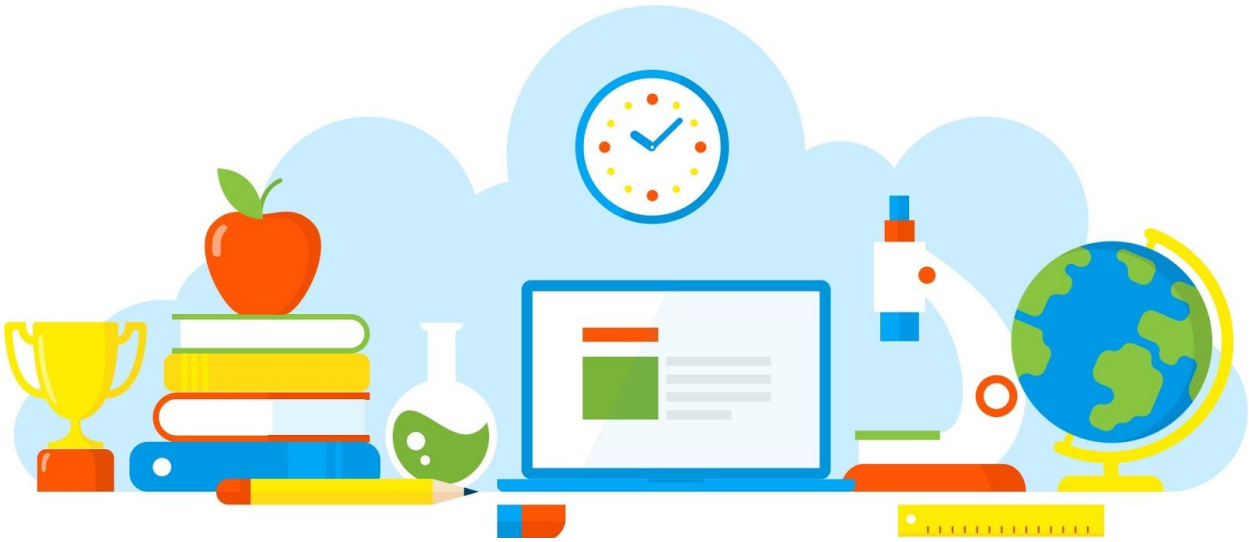


Rapport TP2

Data Analytics

BOURRELY Thomas
10/02/2020

Analyse exploratoire

Question 1.1.1

Les attributs influencés par la taille démographique du pays sonts :

- Le nombre total de chercheurs
- L'inégalité des revenus
- Les taux de chômage (homme et femme)

Question 1.1.2

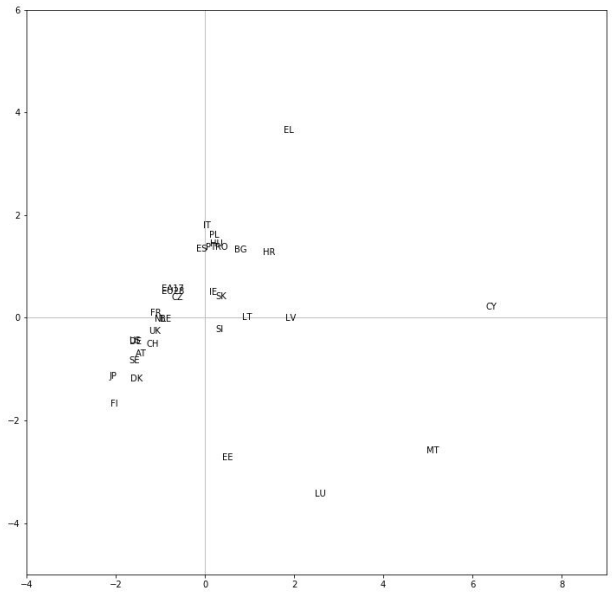
Le filtre StandardScaler modifie les valeurs afin que la distribution de la colonne soit représentée par une gaussienne centrée en 0 et avec un écart-type de 1.

Son utilisation est nécessaire, car cela permet la distribution uniforme des données, facilitant ainsi l'apprentissage du modèle.

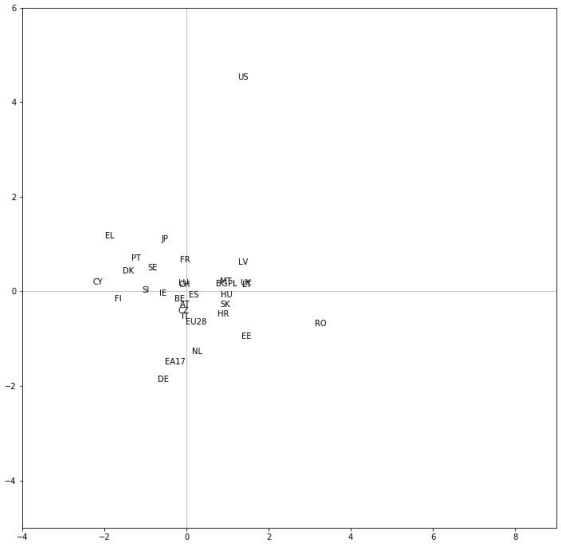
Le filtre est évidemment uniquement appliqué sur les valeurs numériques.

Question 1.1.3

CP1 (abscisse) et CP2 (ordonnée)



CP3 (abscisse) et CP4 (ordonnée)



Question 1.1.4

Les 4 premiers facteurs de l'ACP représentent les informations suivantes :

CP1 : teilmF, teilmM, teimf050, tsc00001, tec00115

CP2 : tet00002, tsc00004, tsdsc260

CP3 : tec00115, tsc00001

CP4 : tec00118

Sur le diagramme contenant les composantes principales 1 et 2, je définirais 8 groupes.

Un groupe pour chaque pays isolé : EE, LU, MT, CY, EL. Les pays restants seraient divisés en 3 groupes. On s'aperçoit que certains pays sont extrêmement proches, tels que la France, la Belgique et les Pays-Bas.

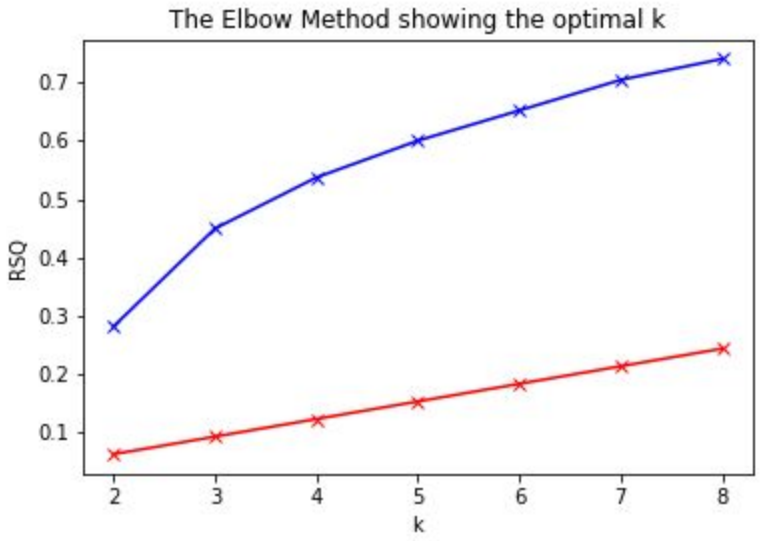
Sur le second diagramme (CP 3 & 4), je propose d'utiliser 5 groupes.

Tout d'abord, les pays plutôt excentrés, US, RO, EE auraient chacun un groupe.

Puis, NL, EA17 et DE seraient ensemble dans un groupe. Pour finir, le reste des pays seraient rassemblés dans un seul groupe.

Dans ce schéma, les pays ne sont pas à la même distances quand dans le premier. La france n'apparaît plus aussi proches de la Belgique et des Pays bas qu'auparavant.

Question 1.1.5



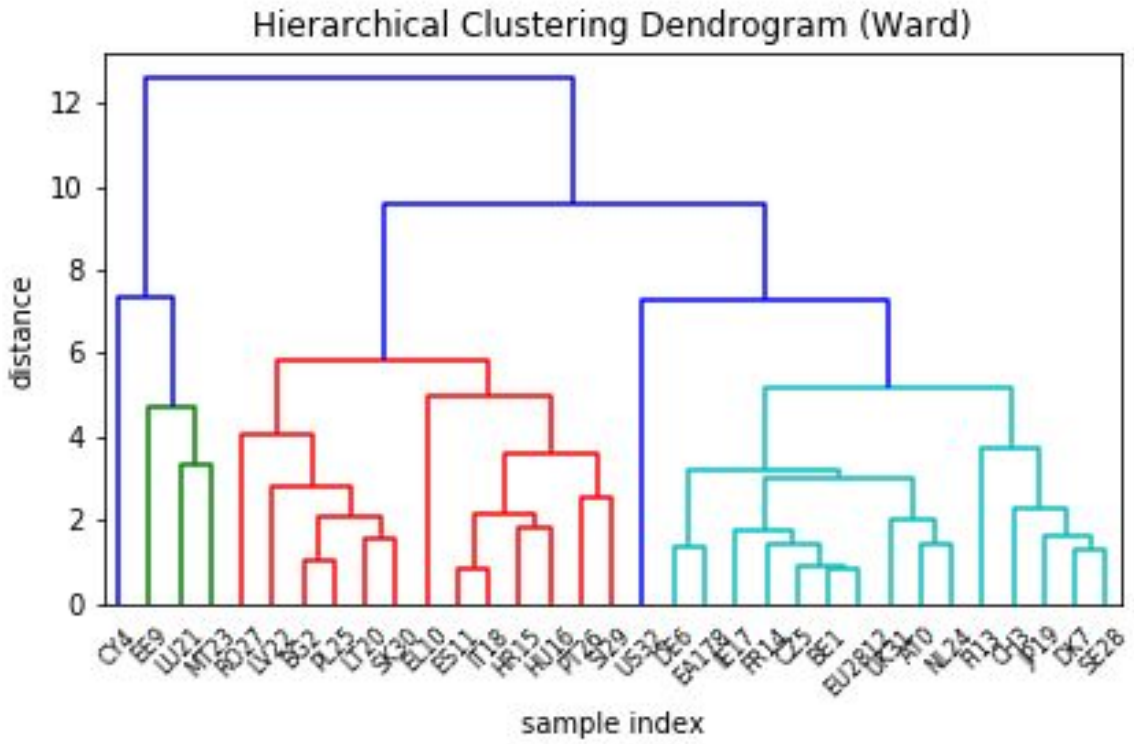
La courbe forme un coude lorsque k=3, je retiens donc cette valeur pour la suite des questions.

Question 1.1.6

```
Num cluster for FR: 1
     list of countries: Autriche, Belgique, Suisse, République
tchèque, Allemagne, Danemark, Zone euro, Estonie, Union européenne,
Finlande, France, Japon, Pays-Bas, Suède, Slovénie, Royaume-Uni,
États-Unis
     centroid: [ 0.03379404 -0.37950641 -0.38241934  0.23289022
-0.61412286 -0.33604868
 -0.06402644  0.84115809  0.58217842]
Num cluster for JP: 1
     list of countries: Autriche, Belgique, Suisse, République
tchèque, Allemagne, Danemark, Zone euro, Estonie, Union européenne,
Finlande, France, Japon, Pays-Bas, Suède, Slovénie, Royaume-Uni,
États-Unis
     centroid: [ 0.03379404 -0.37950641 -0.38241934  0.23289022
-0.61412286 -0.33604868
 -0.06402644  0.84115809  0.58217842]
```

D’après le résultat ci-dessus, la France et le Japon sont dans le même groupe, les deux appartiennent au même cluster. Les pays semblent plutôt êtres des pays développés.

Question 1.1.7 et 1.1.8



Il apparaît que certains pays proches dans le dendrogramme soient aussi proches géographiquement.

Si nous comparons l’Espagne et l’Italie, ces deux pays sont relativement similaires sur un bon nombre de critères, ce qui explique leur proximité dans le diagramme ci-dessus.

Afin d’obtenir un nombre de groupes cohérents, je choisirais de faire une coupure à Distance=7, ce qui impliquerait la création de 5 groupes.

En bleu foncé sont deux valeurs atypiques, Chypre et les États-Unis. Le premier est un petit pays, plutôt pauvre. Le deuxième est l’opposé. Il sont donc chacun dans un groupe.

En vert, nous trouvons des pays plutôt petits, qui n’investissent pas énormément dans la recherche.

Question 1.2.1

Les ? sont des valeurs inconnues.

Les attributs ont tous un type nominal. Cependant 2 groupes sont utilisés : {t}, {low,high}.

Les “?” représentent une valeur manquante. Pour les lignes (data) du fichier, la plupart ont un point d'interrogation pour chaque attribut, excepté le dernier (total).

La ligne 13 créé un vecteur one-hot à partir des données du fichier “.arff”.

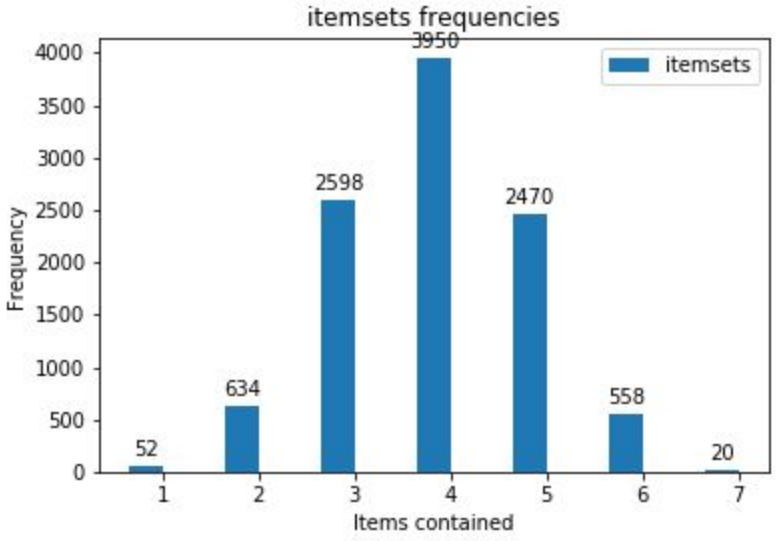
La ligne 14 supprime du DataFrame les colonnes dont le nom termine par ‘?’. Cela élimine donc les colonnes correspondant aux valeurs vides.

Question 1.2.2

La mesure “support” défini la popularité d’un ensemble d’éléments. En le fixant à 0,1, nous indiquons que nous considérons des éléments comme “fréquents”, dès lors qu’un groupe d’éléments apparaît dans minimum 10% des transactions.

En fixant min_support à 0,1, nous obtenons 10282 itemsets.

Ci-dessous, la fréquence d’itemsets en fonction du nombre d’items qu’ils contiennent.



Question 1.2.3

Le résultat de la fonction association_rules donne 24570 règles.

Première règle obtenue :

```
antecedents                (0)
consequents               (11)
antecedent support     0.226281
consequent support     0.719689
support                0.171601
confidence             0.758357
lift                    1.05373
leverage             0.00874991
conviction              1.16002
```

Cette règle associe la 1ère colonne à la 11ème. Il apparaît que l'indicateur "support", ne soit que de 17%, ce qui me semble plutôt faible. D'autant plus que la mesure "lift" est relativement égale à 1, ce qui veut dire que la confiance de la règle divisée par le support du conséquent, est égale à 1. On peut donc en déduire que acheter 0 multiplie par 1 l'achat de 11. La règle ne me semble donc pas pertinente.

Question 1.2.4

Parmi les règles obtenues, 8811 impliquent 4 antécédents et 1 conséquence.

En utilisant la méthode “describe” sur le dataframe contenant les règles avec 5 items, on obtient les informations suivantes :

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
count	8811.000000	8811.000000	8811.000000	8811.000000	8811.000000	8811.000000	8811.000000
mean	0.148910	0.603212	0.117313	0.791140	1.332255	0.027822	1.978909
std	0.022173	0.082035	0.015988	0.054682	0.177603	0.009443	0.413518
min	0.108494	0.362870	0.100065	0.700000	1.062178	0.006364	1.189966
25%	0.133348	0.563000	0.105252	0.747244	1.220572	0.020979	1.675085
50%	0.144586	0.604063	0.112816	0.783489	1.290764	0.026576	1.930478
75%	0.159282	0.640156	0.124919	0.829582	1.388577	0.032749	2.231455
max	0.283337	0.719689	0.202939	0.929795	2.199346	0.072145	4.517902

On se rend compte que l'écart-type des différentes colonnes est très peu élevé. Les valeurs sont donc concentrées autour de la moyenne.

Par ailleurs, on voit que la valeur moyenne du support est également faible. De plus, la valeur de lift est en moyenne de 1.33, les règles sélectionnées semblent donc pour la plupart peu pertinentes.

Question 1.2.5

confidence : **support**(A -> B) / **support**(A)

lift : **confidence**(A -> B) / **support**(B)

leverage : **support**(A -> B) - **support**(A) * **support**(B)

conviction : (1 - **support**(B)) / (1 - **confidence**(A -> B))

Les meilleurs règles pour chaque métrique sont :

Confidence :

```
antecedents           (37, 77, 15, 80, 122, 29)
consequents                                (11)
antecedent support                     0.110223
consequent support                     0.719689
support                                0.103307
confidence                             0.937255
lift                                    1.30231
leverage                              0.0239807
conviction                              4.46746
```

Lift :

```
antecedents           (38, 77, 15, 80, 29)
consequents                     (122, 11)
antecedent support                0.143289
consequent support                0.305381
support                           0.102658
confidence                         0.71644
lift                               2.34605
leverage                         0.0589004
conviction                         2.44964
```

Leverage :

```
antecedents           (24, 29, 38)
consequents                  (122)
antecedent support        0.201643
consequent support         0.36287
support                   0.146315
confidence                0.725616
lift                       1.99966
leverage                 0.0731451
conviction                 2.32204
```

Conviction :

```
antecedents           (66, 12, 77, 122)
consequents                      (80)
antecedent support             0.113897
consequent support             0.639939
support                         0.10482
confidence                     0.920304
lift                            1.43811
leverage                      0.0319325
conviction                      4.5179
```

Question 1.2.6

D'après les données fournies, les produits souvents achetés conjointement avec du café et du lait-crème sont :

- bread-and-cake
- baking needs
- biscuits
- frozen foods
- margarine
- fruit
- vegetables

Analyse descriptive

Question 2.1.1

Le fichier contient 16 attributs. Deux types sont utilisés par ces derniers : nominal et numeric.

Les valeurs manquantes sont représentées par des points d'interrogation. Le nombre de valeurs manquantes semble relativement conséquent, car chaque ligne en possède au moins une.

La variable à prédire est l'acceptation ou non de contrats, "bad" ou "good". Dans le fichier, il apparait deux fois plus d'éléments indiqués "good" que "bad". Le nombre de lignes "bad" est 20, contre 37 pour "good".

Question 2.1.2

DummyClassifier prédit la classe la plus fréquente dans le jeu de données.

Ce classifieur pourrait donner les meilleurs résultats parmi ceux testés dans le cas où la répartition des classes serait fortement inégale.

Son utilisation permet la constitution d'une baseline.

Question 2.1.3

Le résultat de la validation croisée est :

```
--------------------
Accuracy of Dummy classifier on cross-validation: 0.65 (+/- 0.03)
---
           pred:bad  pred:good
true:bad          0         20
true:good         0         37
---
--------------------
Accuracy of GaussianNB classifier on cross-validation: 0.86 (+/- 0.15)
---
           pred:bad  pred:good
true:bad         14          6
true:good         2         35
---
--------------------
Accuracy of Decision tree classifier on cross-validation: 0.77 (+/- 0.22)
---
           pred:bad  pred:good
true:bad         15          5
true:good         8         29
---
--------------------
Accuracy of Logistic Regression classifier on cross-validation: 0.89 (+/- 0.14)
---
           pred:bad  pred:good
true:bad         16          4
true:good         2         35
---
--------------------
Accuracy of SVC classifier on cross-validation: 0.89 (+/- 0.18)
---
           pred:bad  pred:good
true:bad         16          4
true:good         2         35
---
```

On peut constater que deux classifieurs obtiennent le même score : SVC et LogisticRegression. Ce sont les plus performants en terme de taux de classification.

L'intérêt d'utiliser une validation croisée plutôt qu'un découpage des données, est que l'intégralité des données est utilisée pour entraîner et valider le modèle. Cela est d'autant plus adapté ici car nous avons peu de données à notre disposition.

La prédiction fait plus d'erreur lorsque la classe attendue est "bad".

Question 2.1.4

Résultats obtenus :

```
--------------------
Accuracy of Dummy classifier on cross-validation: 0.65 (+/- 0.03)
---
           pred:bad  pred:good
true:bad          0         20
true:good         0         37
---
--------------------
Accuracy of GaussianNB classifier on cross-validation: 0.90 (+/- 0.13)
---
           pred:bad  pred:good
true:bad         16          4
true:good         2         35
---
--------------------
Accuracy of Decision tree classifier on cross-validation: 0.84 (+/- 0.25)
---
           pred:bad  pred:good
true:bad         15          5
true:good         6         31
---
--------------------
Accuracy of Logistic Regresion classifier on cross-validation: 0.91 (+/- 0.20)
---
           pred:bad  pred:good
true:bad         15          5
true:good         0         37
---
--------------------
Accuracy of SVC classifier on cross-validation: 0.91 (+/- 0.20)
---
           pred:bad  pred:good
true:bad         15          5
true:good         0         37
---
```

Les scores (excepté celui du DummyClassifier) ont légèrement augmentés, cependant l'ordre de classement des classifieurs reste inchangé.

Question 2.1.5

Résultat du traitement :

```
--------------------
Accuracy of Dummy classifier on cross-validation: 0.65 (+/- 0.03)
---
           pred:bad  pred:good
true:bad          0         20
true:good         0         37
---
--------------------
Accuracy of GaussianNB classifier on cross-validation: 0.90 (+/- 0.13)
---
           pred:bad  pred:good
true:bad         16          4
true:good         2         35
---
--------------------
Accuracy of Decision tree classifier on cross-validation: 0.79 (+/- 0.28)
---
           pred:bad  pred:good
true:bad         16          4
true:good         5         32
---
--------------------
Accuracy of Logistic Regresion classifier on cross-validation: 0.95 (+/- 0.15)
---
           pred:bad  pred:good
true:bad         18          2
true:good         1         36
---
--------------------
Accuracy of SVC classifier on cross-validation: 0.96 (+/- 0.09)
---
           pred:bad  pred:good
true:bad         18          2
true:good         0         37
---
```

En utilisant tous les attributs disponibles, le classement des classifieurs change. Le classifieur SVC passe devant Logisitc Regression Classifier, de 0.01 point.

Par ailleurs, seuls les scores des classifieurs SVC et Logistic Regression ont augmentés. Celui de GaussianNB n'a pas changé, tandis que celui de l'arbre de décision a baissé, en passant de 0.84 précédemment (attributs catégoriels uniquement) à 0.79.

Question 2.1.6

Dans les questions précédentes, nous avons appliqué des méthodes de normalisation et transformation des données, sur l'ensemble des données. De ce fait, lors de la validation croisée, les données utilisées pour la validation sont biaisées, car transformées.

L'outil pipeline proposé par sklearn permet de regrouper différentes étapes, qui seront exécutées à la suite. On y regroupe des éléments de transformation, puis le dernier élément doit permettre la prédiction. Cela créer une chaîne d'actions assurant la linéarité du traitement. En utilisant une pipeline, nous pouvons appliquer ces traitements aisément, en passant la pipeline à la fonction "cross_val_score" de scikit-learn. Le traitement ne sera alors appliqué que lors de la phase d'apprentissage.

Question 2.2.1

Le modèle de régression est le modèle Ridge, aussi appelé régularisation de Tikhonov. Ce modèle utilise la méthode des moindres carrés comme fonction de coût. La régularisation utilisée est L2.

Le modèle a déterminé que X5, “latitude”, était la variable la plus importante.

Le coefficient attribué à l’age de construction est : **-0.25277252725928306**. Le coefficient étant négatif, cela se traduit par le fait que plus l'âge de construction est grand, plus l’estimation du bien est revue à la baisse.

Question 2.2.2

La valeur prédite est **54.273411202775605**. La valeur prédite est inférieure à la valeur réelle indiquée dans le sujet. Cela peut s’expliquer par le manque de données suffisamment diversifiées.

De plus, le score de R^2 est 0.55. Cela signifie que la droite de régression n’est capable de déterminer que 55% de la distribution des points. Nous avons donc une marge d’erreur significative.