rapport

February 7, 2018

1 Projet Data Mining

Ngoc-Bien NGUYEN et Xian YANG, février 2018
Répertoire Github : https://github.com/MarchesLearning/salaryDiscovery
Lien pour le vidéo démo: https://drive.google.com/file/d/1-AM_5Fj9A-Wgn3Cc9jUqjMzqfXtlyrTd/view?usp=sharing

1.1 0. Introduction du rapport

Le rapport résume une analyse sur un jeu de données issu d'un sondage aux Etats-Unis de 42.882 sujets. Le but de notre projet data mining est d'étudier et d'appliquer les méthodes non supervisées, tout en se familarisant avec la procédure entière du traitement de données. Le rapport est organisé comme suit.

Dans la section 1 on introduit le jeu de données et discute certaines techniques de feature selection. Dans la section 2 on fouille les données de façon à trouver les formes plus intéressantes en utilisant l'exceptional model mining. Dans la section 3 on cherche à réduire la dimension du jeu de données afin de le visualiser. Dans la section 4 on voit de différentes méthodes de clustering. dans la section 5 on étudie de façon détaillée la recherche de motifs fréquents et de règles d'association avec notre jeu de données.

Au contraire de notre cours rapport, le code dans les fichiers `Jupyter` est bien rangé et commenté et ces fichiers sont enrichis de plein de graphiques. Nous vous conseillons *fortement* de jeter un coup d'oeil dans ces 4 fichiers `Jupyter` qui ne sont pas longs du tout, au moins sur les graphes là-bas, qui ont été plottés avec coeur. Manque de temps pour la mise en page on n'a pas transmis les graphes dans ce rapport. Donc veuillez retrouver là-bas les éléments les plus parlants.

1.2 1. Introduction et pré-traitement du jeu de données

Le jeu de données qu'un analyste a dans ses mains, tout comme celui de notre projet, est souvent rempli de données brutes. Il peut y avoir des attributs quantitatifs, qualitatifs ou encore ordinaux. Il peut comtenir des valeurs manquantes. Il peut y avoir des informations redondantes ou inutiles. Ainsi, il convient, avant tout d'autres traitements et de mises en place des algorithmes, de bien observer le jeu de données et d'ajuster les variables pour qu'elles soient sous formes pertinantes pour les prochaines étapes. En effet, la bonne compréhension er la bonne représentation du jeu de données jouent un rôle beaucoup plus important que beaucoup le pensent.

1.2.1 1.1 Première constatation du jeu de données et ajustement des variables (feature selection)

Nous avons pour notre projet un jeu de données appelé "adult income dataset" issu d'un certain bureau de recensement gouvernemental américain. Ce jeu de données a relevé, pour 48.842 sujets de l'enquête apparamment tous résidant aux Etats-Unis, leur situation familiale, parcours académique, profession, âge, race, sex, pays d'origine, temps de travail hebdomadaire, fortune personnelle et, ce qui est le plus important plus tard pour la partie Machine Learning, *leur salaire*. Lorsque ce dernier est au coeur de la prédiction là-bas, ici on le considère comme toutes les autres variables. Mais toutefois, dans l'ensemble de notre analyse on mettra quand-même l'accent sur la relation entre le salaire et les autres variables puisqu'il est visiblement le but de recensement original.

On regarde maintenant les variables de près.

Parmi les 15 variables 6 sont quantitatives : âge, numérotation du parcours académique, fortune personnelle en gain, fortune personnelle en perte, temps de travail hebdomadaire et une certaine "fnlwgt". Les 9 autres sont qualitatives : secteur d'embauche, parcours académique, état civil, situation familiale, profession, race, sexe, pays d'origine et le salaire. Il est à remarquer que le salaire n'est pas numérique : en revanche, il est juste indiqué si le salaire est supérieur ou inférieur à 50k dollars par an.

Qu'est-ce qu'on peut encore faire à part cette observation ? On doit regarder si les variables sont présentées de façon pertinente qui faciliteront notre analyse. Le jeu de données nous permet d'illustrer cas perticuliers.

La variable parait merveilleuse mais on ne sait pas de quoi elle parle. Ce genre de variable doit être supprimée et on ne doit pas faire de l'analyse à l'aveugle avec elle. La variable "fnlwgt" correspond apparament à une certaine classification de dossier au bureau de recensement. De toute façon sans information précise on ne sait pas l'interpreter. On la supprime donc sans hésitation.

Les variables semblent de parler des différents aspects d'un même sujet et de pouvoir être fusionnées. Qu'on les fusionne ou pas, nécessite une analyse cas par cas. La fortune en gain et la fortune en perte sont des variables opossantes : un sujet possède au plus une valeur non-nulle dans ces deux variables. On pourrait réfléchir que celles-là représentent en effet une même information et qu'on pourrait les mettre ensemble. Mais pourtant, vu qu'une fortune négative puisse avoir un sens particulier : les gens qui n'ont pas eu de succès dans leurs investissements ne sont pas forcément des pauvres mais en revanche souvent des riches, on croit que le signe de la fortune n'est pas un signe pûrement mathématique. De ce fait, on préfère garder les deux variables telles quelles.

Les variables sont redondantes et on doit en garder une seule. La numérotation du parcours académique est juste un codage numérique de la variable qualitative (ordinale) parcours académique. Allant de 1 à 16, 16 représente le professorat, 15 la thèse, 14 le diplôme de master et ainsi de suite jusqu'à ce que 1 représente un à 4 ans d'école primaire. La variable qualitative doit être enlevée, puisque la garde des deux amène des informations redondantes qui vont biaser le jeu de données. En plus, la raison pour laquelle on préfère garder la variable quantitative est simplement que cette variable est de nature ordinale. Il convient de prendre la forme qui peut révéler ce sens d'ordre.

La variable est numérique mais elle n'est peut-être pas linéairement proportionnelle à son vrai sens. Une transformation peut-être considérée. Toujours à la variable numérotation du parcours académique, on peut y avoir deux réflexions. Premièrement, les informations sont trop détaillées de manière qu'un numéro plus grand ne représente pas forcément un diplôme avec plus de valeur, bien que la tendance générale soit bonne. Par exemple, *Assoc-voc*, *Assoc-acdm* et *Some-college* sont codées respectivement en 11, 12 et 13. On ne sait pas trop ce que c'est aux états-unis et si ces situations-là délivrent vraiment des regards différents chez les employeurs (on pense toujours à la relation entre la variable en question et le *salaire*). On veut donc les regrouper. Pareil, tout ceux qui n'ont même pas réussi leur collège n'ont pas de grande différence au marché d'emploi, indépendemment du nombre d'années qu'ils ont passé à l'école. On va donc les confondre aussi. Deuxièmement, cette numérotation n'est proportionnelle ni à la nombre d'années d'études qu'il faut pour obtenir le diplôme, ni à la valeur du diplôme que les gens pensent en général. Par exemple, le doctorat est codé par 15 et le master par 14, alors que le bachelor est codé par 13 et le Bac par 9. Souvent une thèse est beaucoup plus cherchée qu'un diplôme de master. Donc on va donner à cette variable plutôt une échelle exceptionnelle de telle façon que le doctorat ou le professorat vaut 1, le master que la moitié, le bachelor qu'un quart et ainsi de suite.

La variable est qualitative et elle est trop détaillée. On doit penser à regrouper certaines valeurs. C'est le cas de la variable pays d'origine. Elle a une trentaine d'étiquettes. Vu que l'enquête a probablement été faite aux Etats-Unis et que 9/10 des sujets sont de nationalité américaine, on peut penser à remplacer cette variable pas deux : 1. le sujet est américain ou non, 2. le sujet est ressortissant d'un pays développé ou non. Ces deux variables permettent de mettre en évidence cette information.

1.2.2 1.2 Traitement des données manquantes

Le traitement des données manquantes est souvent une étape de pré-traitement qu'il faut bien soigner. Heureusement, le jeu de données de notre projet ne possède que 7% de valeurs manquantes sur un nombre d'observations de 14.882. Donc même supprimer toutes ces observations peut être une option. En effet, c'est les trois variables qualitatives : secteur d'embauche, profession et pays d'origine qui contiennent des valeurs manquantes. Ainsi, les méthodes de remplacer une valeur manquante par la moyenne ou encore une valeur spécifique n'ont pas d'application ici. On croit que le fait que les gens n'ont pas renseigné d'information sur leur emploi ou leur pays d'origine peut avoir un sens particulier. Voire il se peut que le sujet sans ce renseignement est en chômage (cette catégorie n'existe par exemple pas dans la variable profession). On va donc créer pour chacune des trois variables une nouvelle catégorie composée des valeurs manquantes.

1.2.3 1.3 Statistique descriptive et résumé du jeu de données

Il est toujours utile de faire une statistique descriptive sur chaque variable (si pas trop nombreuses) pour avoir un aperçu du jeu de données avant tout algorithme compliqué. Par exemple, dans notre exemple on peut voir que 9/10 des sujets sont des américains et parmi les étrangers les méxicains sont le plus nombreux ; la plupart des sujets travaille dans le secteur privé ; il y a trois fois autant de sujets dont le salaire ne dépasse pas 50K qu'il y a de sujets dont le salaire dépasse ce seuil mais en revanche, ce taux est inversé parmi les gens qui ont le titre de doctorat ; il y a deux fois d'hommes qu'il y a de femmes etc.

On peut aussi étudier les relations entre variables. Dans ce cas-là, une matrice de corrélation de toutes les variables est conseillée dans un premier plan (voir `Exploration.ipynb`). On y voit

des informations un peu moins évidentes : bien qu'en nombre total il y a plus d'hommes que de femmes, dans le sous-groupes jamais-mariés il y a beaucoup plus de femmes que d'hommes.

1.2.4 1.4 Standardisation des variables quantitatives et binarisation des variables qualitatives (one-hot transformation)

Une fois le traitement de valeurs manquantes fixé et les variables ainsi que les modalités restantes considérées pertinentes, on peut transformer les variables qualitatives en colonnes par la représentation *one-hot*(binaire). Ici il y a deux cas de figures. Si dans une variable catégorique il y a des valeurs manquantes, on adopte l'approche qui transforme toutes les autres modalités chacune en une colonne et ne garde pas les valeurs manquantes. Mais en effet, l'information qu'une observation est en valeur manquante dans cette variable n'est pas perdue puisqu'elle a 0 partout dans les nouvelles colonnes produises par les modalités non-manquantes. Par contre si une variable catégorique ne contient aucune valeur manquante, on va ignorer la modalité la plus nombreuse et transformer les restes, pour la même raison qu'avant. En tout cas, il vaut mieux d'éviter les informations redondantes.

D'ailleurs, pour les variables numériques, puisqu'elles ont toujours leur propre échelle, on les standardise dans l'intervalle [0, 1].

Finalement lorsque les données sont prêtes, on peut passer à notre analyse.

1.3 2. Exceptional Model Mining

L'exceptional model mining est une famille d'approches qui sert à détecter des sous-groupes *qui n'ont pas forcément une valeur abberante selon un critère simple, mais plutôt se comportent de façon différente du reste du groupe*. Nous avons utilisé les algorithmes proposés par cet article, qui contient de riches discussions théoriques qu'on va laisser ici.

Pour illustrer cette merveilleuse méthode avec notre jeu de données, on se pose deux questions intéressantes.

```
1. Par de simples analyses statistiques on peut donner à chacun des pays d'origine étudiés une n

2. Lorsqu'on attend en général une augmentation salariale avec celle de son âge d'un salarié, ce
```

On essaie de répondre à ces deux questions avec l'EMM. Pour la première question, on construit pour chaque pays un modèle de régression logistique uni-varié qui prédit la classe de salaire de ses ressortissants avec leurs niveaux d'études

$$P(salaire >= 50K \mid formation) = \frac{1}{1 + e^{-(\beta * formation + \beta_0)}}.$$

Lorsque le taux de bonnes prédictions dépasse 70%, on considère que le modèle est fiable pour ce pays d'origine. Une fois qu'on obtient tous les modèles de RL ajustés, on peut faire un nouveau classement de la pente β de la fonction de lien. Plus β est grande, mieux le diplôme d'études est recompensé par le salaire. On peut argumenter que cette pente réflète d'une certaine manière l'égalité sociale. Le résultat n'est pas étonnant : le système américain est "le plus juste" pour les américains, c'est-à-dire que les américains sont en moyenne les plus motivés à faire des études parce qu'il est le plus rentable. Ce n'est pas le cas chez les pays asiatiques qui ont les meilleurs scores tout à l'heure en termes de diplômés. La raison derrière pourrait être les contraîntes sur le marché du travail pour les étrangers.

On pourrait aussi faire une étude pareil sur la rentabilité du niveau d'études au sein des américains mais avec les différentes éthniques.

Pour la deuxième question, on regarde pour chaque profession la corrélation entre l'âge des salariés et leurs nombres d'heures de travail hebdomadaires de ce sous-groupe. Lorsque pour la plupart des professions il n'y a pas de forte corrélation, cette valeur est de 0,4 chez les forces de l'ordre et de -0,1 chez les gardes. Cela pourrait être expliqué par le fait que l'expérience qui se cumule avec l'augmentation de l'âge est importante pour les forces de l'ordre, tant dis que les conditions physiques jouent un rôle essentiel pour les gardes.

1.4 3. Méthodes de réduction de dimensions : t-SNE et Auto-encodeur

Les méthodes de réduction de dimensions peuvent servir non seulement comme un prétraitement avant le clustering, mais elles sont aussi un outil essentiel pour visualiser les données en 2 dimensions.

La méthode de réduction de dimensions la plus courante en data mining est l'analyse en composantes principales. Néanmoins, après une analyse de variances expliquées on se rend vite compte que cette méthode n'est pas pertinente pour la visualisation de notre jeu de données, puisque les deux premières composantes principales n'expliquent ensemble même pas 30% de la variance totale. Il faut donc en chercher d'autres.

Dans notre projet on a essayé deux nouvelles approches : le *t-distributed stochastic neighbor embedding (t-SNE)* et l'*Auto-encodeur*, toutes deux étant des méthodes non-linéaires.

1.4.1 3.1 t-distributed stochastic neighbor embedding

Le t-SNE tente à garder les points qui sont originalement proches l'un de l'autre dans l'espace projeté. L'un de ses caractéristiques c'est que l'image d'arrivée fait toujours un *seul* nuage de points. On illustre cette méthode avec notre jeu de données en les colorant selon de différents critères. Les graphes sont à voir dans le fichier *tsne_autoencoder.ipynb* (les premiers six sub-plots).

On voit que les deux classes de salaires se dégagent bien dans la présentation graphique, ce qui indique que la variable salaire est une forme importante de ce jeu de données, même si elle n'occupe qu'une seule colonne. Les étrangers sont aussi à reconnaître dans le graphe en se regroupant dans le centre du disque. En revanche, le sexe, la profession, l'état civil et la formation ne montrent pas de forme évidente, au moins pas dans le graphe de t-SNE, ce qui nous amène à vouloir essayer une autre méthode de réduction de dimensions, l'auto-encodeur.

1.4.2 3.2 Auto-encodeur

L'auto-encodeur est une méthode de réseau de neuronnes composé d'un encodeur et d'un décodeur, où le dernier a exactement la même structure que le premier juste dans le sens opposé. L'essentiel de cette méthode consiste à encoder les entrées dans un espace de dimensions réduites (souvent 2 pour but de visualisation) en cherchant à perdre le minimum d'informations possible. Elle ne pose aucune métrique de distance au préalable. Ici on illustre également cette méthode avec notre jeu de données en les colorant selon les mêmes critères qu'avant. Les graphes sont à voir dans le fichier *tsne_autoencoder.ipynb* (les deuxièmes six sub-plots).

Cette fois-ci ce ne sont plus un seul nuage de points mais les individus se regroupent quasiment parfaitement en environ 7 sous-groupes. En plus, la coloration nous permet de bien identifier les salariés élevés, le sexe et les américains. Il est aussi à remarquer que sur le dernier graphe de formation, les points jaunes (donc les meilleurs diplômés) se trouvent pour la plupart dans la

région des salariés dépassant 50K, lorsque sur le graphe d'état civil en bas au milieu on voit que les jamais-mariés se trouvent principalement dans la région des salariés moins de 50K (ce qui est logique, pour la raison d'âge) et qu'en plus, ce sont notamment des femmes.

1.5 4. Clustering

Dans cette section, on utilise le t-SNE et l'encodeur de la méthode auto-encodeur pour visualiser les différentes approches de clustering.

Les méthodes de clustering servent à aider les analystes de détecter des formes dans les données. dans notre projet, on part plutôt du but de retrouver dans les clusters les sous-groupes originaux obtenus par différentes variables. On se concentre sur 6 méthodes de clustering de base : K-Means, Hiérarchique, BDSCAN, Mean Shift, Affinity Propagation et Spéctral. On voit que pas toutes ces méthodes de clustering sont compatibles avec notre jeu de données et que même au sein d'une même méthode de clustering, les différents paramètres de départ ou d'autres détails peuvent mener à des résultats très différents.

Pour voir les graphes de clustering en 2D, dirigez-vous vers `Clustering.ipynb`.

On commence par **K-Means**. On met par hasard le nombre de clusters à 5. On voit que le résultat présenté par l'auto-encodeur est assez bon : il a retrouvé les salariés > 50K ; les femmes dans le sous-groupe salaire <= 50K ainsi que d'autres points qui se regroupent de façon naturelle.

Ensuite on essaie le clustering **hiérarchique** (agglomératif). On peut faire plusieurs choses avec cette méthode. En premier lieu, on fait un double-clustering par rapport aux observatiosn et par rapport aux variables. Puis on compare la performance des trois linkages : "ward", "average" et "complete", en mettant par hasard le nombre de clusters à 6. On voit que "ward" et "complete" fonctionne aussez bien.

On s'intéresse aussi à **DBSCAN**. Cette méthode est basée sur la densité des nuages de points. Ainsi, il peut subir du fléau de la dimension dans notre exemple puisque dans la haute-dimension les points sont creux partout.

Finalement le clustering **spéctral**, basé sur la théorie des graphes, peut révéler quelques formes, alors que l'**affinity propagation**_ et le **mean shift** n'a pas révélé d'information très intéressante, pour une raison similaire que *DBSCAN*. Toutefois, on peut bien imginer que DBSCAN va fonctionner parfaitement si on diminue la dimension avec l'auto-encodeur avant de donner les données à cet algorithme de clustering.

Pour plus de détails, veuillez lire le fichier `Clustering.ipynb`.

1.6 5. Recherche de motifs fréquents et de règles d'association

Chercher les ensembles fréquents. Nous utilisons R pour cette partie, nous allons utiliser le package "arules" et "arulesViz". Les détails de ces packages peuvent être trouvées sur le site: https://cran.r-project.org/web/packages/arules/arules.pdf Et https://cran.r-project.org/web/packages/arulesViz/arulesViz.pdf

Nous commençons à chercher les ensembles fréquents par l'algorithme Apriori avec le support 0.1, nous trouvons alors 2733 ensembles fréquents! Nous regardons 50 premiers:

```
In [4]: from IPython.display import Image
        Image(filename='image1.png')
```

```
Out[4]:
```

```
> inspect(head(itemsets, n=50))
     items                                          support   count
[1]  {V8=Unmarried,,V11=0,}                         0.1001812 3262
[2]  {V8=Unmarried,,V12=0,}                         0.1030681 3356
[3]  {V7=Sales,,V14=United-States,}                 0.1033138 3364
[4]  {V7=Sales,,V11=0,}                             0.1022696 3330
[5]  {V7=Sales,,V12=0,}                             0.1062007 3458
[6]  {V7=Adm-clerical,,V15=<=50K}                   0.1002119 3263
[7]  {V7=Adm-clerical,,V14=United-States,}          0.1059243 3449
[8]  {V7=Adm-clerical,,V11=0,}                      0.1085655 3535
[9]  {V7=Adm-clerical,,V12=0,}                      0.1118823 3643
[10] {V7=Exec-managerial,,V9=White,}                0.1119744 3646
[11] {V7=Exec-managerial,,V14=United-States,}       0.1147078 3735
[12] {V7=Exec-managerial,,V11=0,}                   0.1078898 3513
[13] {V7=Exec-managerial,,V12=0,}                   0.1158441 3772
[14] {V7=Craft-repair,,V10=Male,}                   0.1190688 3877
[15] {V7=Craft-repair,,V9=White,}                   0.1134486 3694
[16] {V7=Craft-repair,,V14=United-States,}          0.1131722 3685
[17] {V7=Craft-repair,,V11=0,}                      0.1163969 3790
[18] {V7=Craft-repair,,V12=0,}                      0.1198366 3902
[19] {V7=Prof-specialty,,V9=White,}                 0.1121280 3651
[20] {V7=Prof-specialty,,V14=United-States,}        0.1134179 3693
[21] {V7=Prof-specialty,,V11=0,}                    0.1100703 3584
[22] {V7=Prof-specialty,,V12=0,}                    0.1181168 3846
[23] {V6=Divorced,,V15=<=50K}                       0.1222321 3980
[24] {V6=Divorced,,V9=White,}                       0.1166119 3797
[25] {V6=Divorced,,V14=United-States,}              0.1278216 4162
[26] {V6=Divorced,,V11=0,}                          0.1276066 4155
[27] {V6=Divorced,,V12=0,}                          0.1313842 4278
[28] {V6=Never-married,,V8=Own-child,}              0.1377415 4485
[29] {V2=Private,,V8=Own-child,}                    0.1187617 3867
[30] {V8=Own-child,,V15=<=50K}                      0.1535886 5001
```

Nous avons des items qui ne sont pas très intéressants, par exemple, “units-state” de V14 qui se présente sur ~90%, “<=50” de V15 qui compte 76%, “White” de V9 ~85,5% et “Private” de V2 ~70% des observations. Ces valeurs-là seront très probables de se présenter dans tous les ensembles fréquents fermés. Les attributs “females” sont très dominés par “male”(qui est en fait double). Dans la variable V6(situation familiale) l’attribut “Married-civ-spouse” est aussi dominant des autres. Nous allons donc réduire le nombre des ensembles fréquents en augmenter le support minimal.

```
In [6]: Image(filename='image2.png'); Image(filename='image3.png')
```

Out[6]:

```
> inspect(frequentItems)
     items                                                 support   count
[1]  {V9=White,V10=Male,V14=United-States}                 0.5415421 26450
[2]  {V10=Male,V14=United-States}                          0.5983170 29223
[3]  {V9=White,V10=Male}                                   0.5883256 28735
[4]  {V2=Private,V9=White,V14=United-States}                0.5433848 26540
[5]  {V2=Private,V14=United-States}                        0.6171942 30145
[6]  {V2=Private,V9=White}                                 0.5942427 29024
[7]  {V2=Private,V15=<=50K}                                0.5429548 26519
[8]  {V9=White,V14=United-States,V15=<=50K}                0.5835347 28501
[9]  {V14=United-States,V15=<=50K}                         0.6784734 33138
[10] {V9=White,V15=<=50K}                                  0.6378731 31155
[11] {V9=White,V14=United-States}                          0.7881127 38493
[12] {V14=United-States}                                   0.8974243 43832
[13] {V9=White}                                            0.8550428 41762
[14] {V15=<=50K}                                           0.7607182 37155
[15] {V2=Private}                                          0.6941976 33906
[16] {V10=Male}                                            0.6684820 32650
```

Nous essayons de lire ces résultats, pour savoir si cela vient de hasard ou il y a vraiment des relations entre ces attributs. Interprétation de [1]: Supposons que V9, V10, V14 sont indépendantes. Donc, la probabilité qu'on observe l'ensemble { V9=White, V10= Male, V14= Unit-State} est Pr(White)*Pr(Male)*Pr(Unit-State)=0.5129505. D'autre part, nous avons cet ensemble avec le support 0.5415421 qui est un peu plus grand que la proba. Donc, on peut dire que cette combinaison qui vient plus tôt du hasard! Par la même critère, on peut dire que les combinaisons en [2], [3],[5]-[11] sont venues de hasard! Les [12]-[16] sont simplement de pourcentage des attributs correspondants. La relation en [4] qui donne la probabilité 0,4164 si hasard et le support est assez loin de ce probabilité. Il y a fortement une relation entre ces attributs. On peut donc dire que les Blancs d'origine des Etats-Unis ont une tendance de travailler indépendamment.

1.6.1 Chercher des règles d'association

Pour filtrer des règles d'associations, nous jouons avec les paramètres support et conf. Nous nous rappelons que supp(X->Y)=supp(X) et conf(X->Y)=supp(XY)/supp(X). Mathématiquement, le premier est simplement la proba de X(d'observer X) et le deuxième est la proba de Y sachant X.

```
In [7]: Image(filename='image4.png')
```

Out[7]:

```
> rules <- apriori (data1, parameter = list(supp = 0.3, conf = 0.95))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
       0.95    0.1    1 none FALSE            TRUE       5     0.3      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 14652

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[104 item(s), 48842 transaction(s)] done [0.03s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.05s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [21 rule(s)] done [0.00s].
creating S4 object  ... done [0.01s].
```

In [10]: `Image(filename='image5.png')`

Out[10]:

```
> inspect(rules)
     lhs                                                          rhs                     support
[1]  {V6=Never-married}                                        => {V15=<=50K}             0.3149748
[2]  {V8=Husband}                                              => {V6=Married-civ-spouse} 0.4034233
[3]  {V8=Husband}                                              => {V10=Male}              0.4036485
[4]  {V6=Married-civ-spouse,V8=Husband}                        => {V10=Male}              0.4034028
[5]  {V8=Husband,V10=Male}                                     => {V6=Married-civ-spouse} 0.4034028
[6]  {V6=Married-civ-spouse,V10=Male}                          => {V8=Husband}            0.4034028
[7]  {V8=Husband,V9=White}                                     => {V6=Married-civ-spouse} 0.3654232
[8]  {V8=Husband,V14=United-States}                            => {V6=Married-civ-spouse} 0.3628230
[9]  {V8=Husband,V9=White}                                     => {V10=Male}              0.3656075
[10] {V8=Husband,V14=United-States}                            => {V10=Male}              0.3630482
[11] {V6=Married-civ-spouse,V8=Husband,V9=White}               => {V10=Male}              0.3654027
[12] {V8=Husband,V9=White,V10=Male}                            => {V6=Married-civ-spouse} 0.3654027
[13] {V6=Married-civ-spouse,V9=White,V10=Male}                 => {V8=Husband}            0.3654027
[14] {V6=Married-civ-spouse,V8=Husband,V14=United-States}      => {V10=Male}              0.3628025
[15] {V8=Husband,V10=Male,V14=United-States}                   => {V6=Married-civ-spouse} 0.3628025
[16] {V6=Married-civ-spouse,V10=Male,V14=United-States}        => {V8=Husband}            0.3628025
[17] {V8=Husband,V9=White,V14=United-States}                   => {V6=Married-civ-spouse} 0.3382540
[18] {V8=Husband,V9=White,V14=United-States}                   => {V10=Male}              0.3384382
[19] {V6=Married-civ-spouse,V8=Husband,V9=White,V14=United-States} => {V10=Male}          0.3382335
[20] {V8=Husband,V9=White,V10=Male,V14=United-States}          => {V6=Married-civ-spouse} 0.3382335
[21] {V6=Married-civ-spouse,V9=White,V10=Male,V14=United-States} => {V8=Husband}          0.3382335
```

Nous avons donc plusieurs règles intéressantes avec de très bonne confiance. Par exemple, Si on n'est jamais marié, on aura une ressource de moins de 50 000 dollars par an avec une proba de 95,45%. Tous les hommes mariés ont une mariée de type conjoint-conjoint. La règle [20] est encore plus fort: un américain blanc marié aura une mariée conjoint-conjoint avec la proba de 99,94%. Les autres règles sont triviales.

Nous changeons les paramètres supp et conf pour chercher des autres règles: Cette fois nous mettons supp=0,2 et conf=0,95, donc, nous obtenons 51 règles d'association.

In [11]: `Image(filename='image7.png')`

Out[11]:

```
> rules <- apriori (data1, parameter = list(supp = 0.2, conf = 0.95))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
       0.95    0.1    1 none FALSE            TRUE       5     0.2      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 9768

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[104 item(s), 48842 transaction(s)] done [0.03s].
sorting and recoding items ... [13 item(s)] done [0.01s].
creating transaction tree ... done [0.04s].
checking subsets of size 1 2 3 4 5 6 done [0.00s].
writing ... [51 rule(s)] done [0.00s].
creating S4 object  ... done [0.01s].
```

Ici, nous pourrons traduire quelques règles intéressantes: [4] Les célibataires qui travaillent indépendamment aurons une ressource annuelle moins de 50000 dollars avec une probabilité de 0.96. [6] Tous les blancs célibataires aurons une ressource de moins de 50000 dollars par an avec proba=0.96 [11] Les mariés avec une ressource de moins de 50000 dollars par an auront une relation de conjoint-conjoint [18] Tous les célibataires américains qui travaillent indépendamment aurons une ressource annuelle de moins de 50000 dollars avec une proba=0.96

[32] Les bancs mariés qui travaillent indépendamment aurons presque surement une relation marié conjoint-conjoint. Mais [50] nous donne une affirmation encore plus fort: Les américains blancs mariés qui travaillent indépendamment aurons presque sûrement une relation marié conjoint-conjoint. Pour filtrer encores les attributs, nous allons supprimer les variables avec un statut qui est trop importante
et tous les variables numériques, donc, nous supprimons V1, V2, V3, V5, V9, V10, V11, V12, V13, V14.

Nous changeons le paramètre pour chercher d'autres règles: Nous choisissons le support=0.05 et conf =0.9, nous obtenons donc 26 règles, et nous déduisons quelques unes intéressantes: Nous pouvons donc avoir des règles intéressantes suivantes: [2] Les gens qui ont un status Other-service aurons une ressource annuelle de moins de 50000 dollars. [3] Ce qui ne sont pas mariés qui auront une ressource annuelle de moins de 50000 dollars. [4] Ce qui a un enfant tout seul auront une ressource annuelle de moins de 50000 dollars. [5] Ce qui ne sont jamais mariés auront une ressource annuelle de moins de 50000 dollars. [6] Ce qui ont un statut mari auront une ressource annuelle de moins de 50000 dollars. [11] Ce qui avoir un enfant tout seul et qui ont le collège auront une ressource annuelle de moins de 50000 dollars. Brièvement, la plus parte de ces règles emmène le statut "ressource annuelle de moins de 50000 dollars" ou " être marié".

Finalement, nous utilisons la fonction is.signifiant() dans le package arules pour filtrer toutes les règles d'association significatives. Le niveau de test est 1%, donc un test assez exacte. Nous obtenons les règles significatives suivantes (avec support >=0.1 et conf>=0.7) [1] {V8=Husband} => {V6=Married-civ-spouse} [2] {V6=Married-civ-spouse} => {V8=Husband}
[3] {V6=Married-civ-spouse,
V10=Female} => {V8=Wife}
[4] {V8=Husband,
V10=Male} => {V6=Married-civ-spouse} [5] {V6=Married-civ-spouse,
V10=Male} => {V8=Husband}
[6] {V6=Married-civ-spouse,

V9=White,
V10=Female} => {V8=Wife}
[7] {V6=Married-civ-spouse,
V10=Female,
V14=United-States} => {V8=Wife}
[8] {V6=Married-civ-spouse,
V9=White,
V10=Male} => {V8=Husband}
[9] {V6=Married-civ-spouse,
V10=Male,
V14=United-States} => {V8=Husband}
[10] {V6=Married-civ-spouse,
V9=White,
V10=Male,
V14=United-States} => {V8=Husband}