Test recrutement:
Document d'analyses du jeu de données US Census

Gen YANG

18 mai 2016

1 Plan d'analyse

L'objectif de ce document est de fournir une analyse primaire du jeu de données US Census, et de mettre en place un modèle d'apprentissage supervisé pour modéliser et prédire les populations qui ont une revenue supérieure à 50,000$. La variable d'analyse est une variable binaire qui séparent les populations en deux catégories : "-50000." pour celles qui ont une revenue inférieure au seuil étudié, et "50000+." pour celles supérieure.

Nous commençons par donner quelques statistiques descriptives sur les attributs, et de tester ensuite plusieurs algorithmes d'apprentissage supervisé de l'état de l'art afin de comparer leur performance prédictive à l'aide d'une validation croisée. Au final, nous testerons l'algorithme ayant la meilleure performance avec les données de test.

Les codes sont réalisés sous Python 2.7 avec notamment les packages *pandas* et *scikit-learn*.

2 Statistiques descriptives

Les codes de cette partie sont résumés dans le fichier *descStat.py*. Le jeu de données comporte 42 variables en totalité y compris la variable de sortie (que nous appellerons "classe"). Les 40 variables d'entrée (que nous appellerons également "attributs") sont composés de 8 variables numériques dont une variable "instance weight" de type flottant et 7 variables de type entier, et le reste sont catégorielles.

La variable "instance weight" représente la proportion qu'une instance de donnée correspond en réalité, elle est surtout utile pour la phase interprétation. Comme elle n'est pas à proprement parler une caractéristique des populations, nous allons l'écarter pour la phase de modélisation.

Dans cette partie, nous présenterons en deux temps, les distributions des variables numériques et ensuite des celles catégorielles, à l'aide des tableaux résumés et des histogrammes.

2.1 Distributions des variables numériques

Nous commençons par résumer la distribution de ces variables dans le Tableau **??**. Nous donnons dans le tableau le nombre d'instances sans valeur manquante (*count*), la valeur moyenne (*mean*), l'écart-type (*std*), la valeur minimale (*min*), les quartiles $(25\%, 50\%, 75\%)$, la valeur maximale et enfin le pourcentage de valeurs manquantes (*Miss(%)*).

Nous pouvons constater plusieurs caractéristiques intéressantes. Premièrement, aucune de ces attributs n'a de valeur manquante, ce qui facilitera le traitement ultérieur. Deuxième, nous remarquons que beaucoup d'instances sont de valeur nulle (troisième quartile à 0) pour les attributs *wage per hour, capital gains, capital losses, dividende from stocks*. De plus, il y a des

écarts de valeurs énormes entre les valeurs extrêmes. Ceci nous permettra potentiellement de discriminer les classes de manière efficace, nous vérifierons ceci plus tard.

	count	mean	std	min	25%	50%	75%	max	Miss(%)
age	199523	34.49	22.31	0	15	33	50	90	0.00
wage per hour	199523	55.43	274.90	0	0	0	0	9999	0.00
capital gains	199523	434.72	4697.53	0	0	0	0	99999	0.00
capital losses	199523	37.31	271.90	0	0	0	0	4608	0.00
dividends from stocks	199523	197.53	1984.16	0	0	0	0	99999	0.00
num p. worked for employer	199523	1.96	2.37	0	0	1	4	6	0.00
weeks worked in year	199523	23.17	24.41	0	0	8	52	52	0.00

TABLE 1 – Statistiques descriptives pour les variables numériques

Pour illustrer la distribution de ces attributs, nous traçons également des histogrammes, sur la Figure 1, pour avoir une vue plus précise sur les distributions. Nous pouvons constater que, effectivement, à part l'attribut *age*, tous les autres ont des distributions très hétérogènes. Presque la totalité des attributs *wage per hour, capital gains, capital losses, dividende from stocks* sont à valeur zéro.

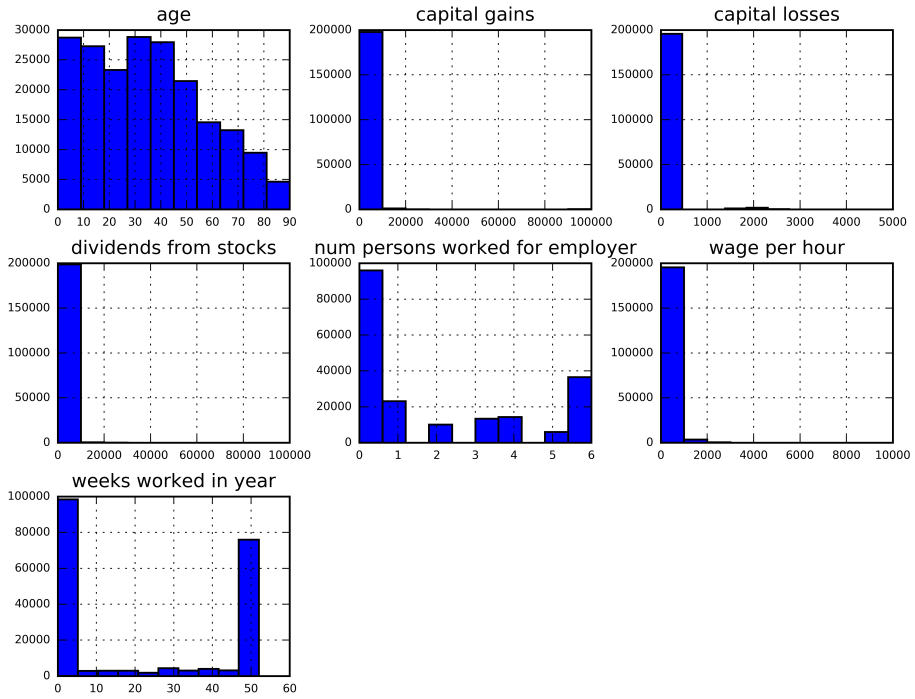

FIGURE 1 – Histogramme des variables numériques

2.2 Distributions des variables catégorielles

Nous faisons la même analyse pour les 33 variables catégorielles et la classe. Le Tableau 2 résument les informations sur le nombre d'instances (*count*), le nombre de modalités (*unique*), la

modalité la plus fréquente (*top*) et sa fréquence correspondance (*freq*), ainsi que le pourcentage de valeurs manquantes (*Miss(%)*). Pour une question de lisibilité, nous n'allons pas inclure les 33 graphiques d'histogrammes dans ce document, mais les fichiers restent accessibles sous le dossier "Histogrammes".

	count	unique	top	freq	Miss(%)
class of worker	199523	9	Not in universe	100245	0.00
detailed industry recode	199523	52	0	100684	0.00
detailed occupation recode	199523	47	0	100684	0.00
education	199523	17	High school graduate	48407	0.00
enroll in edu inst last wk	199523	3	Not in universe	186943	0.00
marital stat	199523	7	Never married	86485	0.00
major industry code	199523	24	Not in universe or children	100684	0.00
major occupation code	199523	15	Not in universe	100684	0.00
race	199523	5	White	167365	0.00
hispanic origin	198649	9	All other	171907	0.44
sex	199523	2	Female	103984	0.00
member of a labor union	199523	3	Not in universe	180459	0.00
reason for unemployment	199523	6	Not in universe	193453	0.00
full or part time employment stat	199523	8	Children or Armed Forces	123769	0.00
tax filer stat	199523	6	Nonfiler	75094	0.00
region of previous residence	199523	6	Not in universe	183750	0.00
state of previous residence	198815	50	Not in universe	183750	0.35
detailed household and family stat	199523	38	Householder	53248	0.00
detailed household summary in household	199523	8	Householder	75475	0.00
migration code-change in msa	99827	9	Nonmover	82538	49.97
migration code-change in reg	99827	8	Nonmover	82538	49.97
migration code-move within reg	99827	9	Nonmover	82538	49.97
live in this house 1 year ago	199523	3	Not in universe under 1 year old	101212	0.00
migration prev res in sunbelt	99827	3	Not in universe	84054	49.97
family members under 18	199523	5	Not in universe	144232	0.00
country of birth father	192810	42	United-States	159163	3.36
country of birth mother	193404	42	United-States	160479	3.07
country of birth self	196130	42	United-States	176989	1.70
citizenship	199523	5	Native- Born in the United States	176992	0.00
own business or self employed	199523	3	0	180672	0.00
fill inc questionnaire for veteran's admin	199523	3	Not in universe	197539	0.00
veterans benefits	199523	3	2	150130	0.00
year	199523	2	94	99827	0.00
class	199523	2	- 50000.	187141	0.00

TABLE 2 – Statistiques descriptives pour les variables numériques

Nous constatons que, contrairement aux variables numériques, maintenant nous avons plusieurs variables ayant des valeurs manquantes, notamment pour les 4 variables *migration* dont le pourcentage atteint presque 50%. En outre, les variables *hispanic origin, state of previous residence, country of birth father/mother/self* ont également certaines valeurs manquantes (inférieures à 4%).

(Remarque : par données manquantes, nous désignons les instances de données qui possèdent des valeurs spécifiées par " ?". Les valeurs "NA", "Not in universe" sont considérées comme des

modalités normales.)

2.3 Traitement des données manquantes

Il faut alors mettre en place des techniques pour le traitement des valeurs manquantes. Deux types de stratégies peuvent être mises en place pour ceci : soit nous pouvons tout simplement enlever ces instances de données, soit nous pouvons tenter de compléter ces données. La première est surtout intéressante quand le pourcentage de données manquantes est faible, car elle minimise l'impact de ces données que nous pourrions considérer comme étant de qualité inférieure. La deuxième est nécessaire si le pourcentage est trop important pour être ignoré (notamment le cas de *migration code* ici) ou si nous avons des informations à notre disposition pour inférer la complétion.

Une technique générique et assez courante de complétion est de compléter par la modalité la plus courante. La raison est que, par exemple pour la variable ***state of previous residence***, étant donné environ 92% des instances de données ont pour modalité "Not in universe", il est fort probable que ça soit également la modalité de ces instances dont l'information est manquante.

Cependant, cette technique n'est pas adapté à toutes les situations, notamment quand nous avons des informations *a priori* pour inférer la complétion. Par exemple, l'attribut "**hispanic origin**" a déjà pour modalité "Do not know" et "NA", qui peuvent correspondre toutes les deux au cas de la donnée manquante, même s'ils ne sont pas les modalités les plus fréquentes. En effet, une vérification rapide montre que aucune instance de données a pour modalité "NA" malgré qu'elle soit présente dans les méta-données. Nous décidons donc de compléter les données manquantes en utilisant "NA". De même, pour les variables "**migration**", comme la moitié des données sont manquantes, il n'est pas raisonnable d'utiliser la complétion par la modalité la plus fréquente. Il est beaucoup plus logique de compléter par la modalité "Not in universe".

De même, il est discutable de compléter les variables ***country of birth father/mother/self*** en utilisant la modalité la plus fréquente. Même si environ 80% des données partagent la modalité "United-States", nous pouvons nous interroger sur la raison de ce manquement. En effet, si un individu n'a pas de réticence ou de raison valable, il devrait répondre "United-States" sans souci particulier. Le manquement peut donc traduire un cas particulier qui empêche l'enquêté de répondre. En effet, nous pouvons voir dans les méta-données que ces attributs ne disposent pas d'un choix "NA" ou "Other" contrairement aux autres attributs. Ici nous pouvons alors choisir de créer une modalité "Other" pour gérer ces données manquantes et accroître la cohérence par rapport au reste des attributs.

Bien sûr, il est possible d'utiliser des techniques plus élaborées pour la complétion de ces valeurs manquantes. Par exemple, pour une variable disposant de valeurs manquantes, calculer la probabilité de chacune des modalités possibles conditionnellement aux autres variables, et de choisir la modalité ayant la probabilité maximale. Ou encore l'utilisation des connaissances d'experts. Une autre manière de traiter les variables ayant de fort pourcentage de données manquantes est de supprimer les variables en questions. Comme la variable en question peut être considérée comme étant de faible quantité due aux données manquantes, il peut avoir du sens de préférer cette solution.

3 Choix d'un algorithme d'apprentissage

Pour l'apprentissage de ce jeu de données, nous allons tester deux algorithmes : les arbres de décisions et la régression logistique (codes provenant du package *scikit-learn*). Nous allons les comparés sur plusieurs points : avantages théoriques, performance prédictive et temps de calcul. Le code correspondant à cette section se trouve dans le fichier *chooseModel.py*.

3.1 Discussion sur le traitement des variables catégorielles

L'avantage usuel des arbres de décision est que cette méthode permet d'utiliser directement les variables catégorielles sans recourir à des techniques de binarisation, contrairement à la régression logistique. En effet, comme ce jeu de données est composé en grande partie des variables catégorielles à forte cardinalité, les techniques de binarisation couramment utilisées pour transformer les variables catégorielles peuvent être moins efficaces.

Cependant, une contrainte technique fait que, la méthode d'arbres de décision implémentée par *scikit-learn* ne permet pas de traiter nativement les variables catégorielles. Préférant ne pas recourir à des codes moins éprouvés que l'on peut trouver sur *github*, je me suis restrains à la méthode d'arbres de décision implémentée par *scikit-learn* qui utilise une technique de binarisation pour gérer les variables catégorielles. Ce qui permet d'ailleurs de la comparer avec la régression logistique sur un pieds d'égalité, puisque les deux méthodes utilisent maintenant le même jeu de données dont les attributs sont binarisés.

Pour une analyse en profondeur du jeu de données, il faudrait soit utiliser une version des arbres de décision sans binarisation, soit une transformation des variables catégorielles en numériques de manière plus intelligente, en utilisant par exemple des probabilités conditionnelles. De même, nous avons gardé les paramètres par défaut fixés par *scikit-learn* pour les algorithmes utilisées, pour optimiser la performance prédictive, il faudrait également optimiser les paramètres (par exemple, le type de régularisation pour la régression logistique, le critère de split pour les arbres de décision...).

3.2 Performance prédictive et temps de calcul

Pour comparer les deux modèles de manière fiable en limitant les risques de sur-apprentissage, nous allons utiliser une validation croisée à 10 répétitions (10-fold cross validation). Nous intégrons également la variable "instance weight" en tant que poids des échantillons, puisque notre tâche finale est de donner des interprétations réelles sur la variable objective.

Dans les conditions expérimentales énoncées, nous constatons que **les arbres de décisions réalisent une performance de 93% de bonnes prédictions avec un temps de calcul de 370 secondes**, tandis que la **régression logistique réalise une performance de 94% avec un temps de calcul de 111 secondes**.

Il est alors évident que, en utilisant des données binarisées, la régression logistique soit préférable à l'arbre de décision dans notre cas.

4 Evaluation avec les données de test

Dans cette partie, nous allons évaluer la pertinence de notre classifieur choisi (la régression logistique) en examinant sa performance prédictive sur le jeu de données de test, et essayer de discerner les facteurs (attributs) clés qui caractérisent les populations qui ont une revenue supérieure à 50,000$ (*i.e.* de classe "50000+."). Le code correspondant à cette section est dans le fichier *factorsExplained.py*.

4.1 Traitement des données manquantes

Une première chose que nous constatons est que, malgré la procédure développée pour le traitement des données manquantes pour le jeu de données d'apprentissage. Des nouveaux cas de données manquantes sont apparues : des attributs jadis sans données manquantes ont maintenant des données manquantes.

Pour traiter ces valeurs manquantes, nous reprenons les idées que nous avons énoncées lors de l'apprentissage. Pour résumé, nous appliquerons deux règles génériques pour le traitement de ces valeurs manquantes :

- si la variable en question possède une modalité de type “NA" ou “Not in universe", nous convertissons alors les valeurs manquantes en cette modalité
- sinon, nous remplaçons les valeurs manquantes par la modalité la plus courante

Il est d'ailleurs important de noter deux spécificités liées au fait que nous sommes dans un environnement de test. Premièrement, il n'est pas possible de rajouter des modalités de type “NA" aux variables qui n'en possèdent pas, puisque le modèle est déjà construit. Deuxième, l'amputation des instances de données est impossible (nous ne pouvons pas dire à l'utilisateur “joker, je passe celle là"), et l'amputation par colonnes changera le modèle.

(cf. la méthode *clean_na_generic* dans *manipData.py*)

4.2 Evaluations

Après le traitement des données manquantes, nous constatons que notre classifieur de régression logistique donne une performance prédictive de 92.37% de bonnes prédictions, ce qui est en accord avec nos résultats de validations croisées. Nous résumons dans le Tableau 3 les facteurs (soit des attributs, soit des modalités d'attributs) les plus discriminant par rapport à la tâche de prédiction de revenue.

nom de l'attribut ou de la modalité	poids
detailed household and family stat=Householder	2.91929988338
detailed household summary in household=Householder	2.80456978848
detailed occupation recode=33	2.71505965373
country of birth mother=Vietnam	2.669794465
country of birth self=Vietnam	2.669794465
country of birth father=Vietnam	2.669794465
marital stat=Married-civilian spouse present	2.41515662639
race=Asian or Pacific Islander	2.01658319981
major industry code=Utilities and sanitary services	1.83460410167
education=12th grade no diploma	1.83460410167
detailed industry recode=31	1.83460410167
full or part time employment stat=Not in labor force	1.7052583495
year=95	1.69605517347
citizenship=Foreign born- Not a citizen of U S	1.6764850119
education=Prof school degree (MD DDS DVM LLB JD)	1.64382889934
detailed industry recode=35	1.62804917531
live in this house 1 year ago=Not in universe under 1 year old	1.60488074418
migration code-change in msa=Not in universe	1.60488074418
migration code-change in reg=Not in universe	1.60488074418
migration code-move within reg=Not in universe	1.60488074418

TABLE 3 – Les 10 attributs (ou modalités) les plus impactant sur la prédiction de revenue (dans l'ordre décroissant)

Nous allons tenter de regrouper plusieurs facteurs similaires ensembles, pour dégager des profils influençant les plus le revenue. Attention, normalement des études et des tests statistiques spécifiques sur la corrélation des variables sont normalement nécessaires pour établir ces profils

de manière fiable. Mais par contrainte de temps, nous allons nous contenter de dégager quelques profils de sens commun.

Voici plusieurs groupes de facteurs majeurs :

1. **Le fait d'être propriétaire** d'une maison est le facteur le plus impactant ("detailed household and family stat=Householder" et "detailed household summary in household=Householder" occupent les deux premières positions).
2. Le facteur lié à **l'emploi**, plus particulièrement des métiers recodés "33", "31" et "35". D'après une recherche sur le Web, ils correspondraient respectivement aux métiers de "opérateur et manager des fermes", "**agent de nettoyage ou de service aux bâtiments**" et "métiers reliés à l'agriculture". Le première et le troisième semblent pointer **les agriculteurs**. Et le deuxième rejoint par ailleurs le facteur "major industry code=Utilities and sanitary services" et a sûrement un impact fort mais négatif sur le seuil de revenue.
3. **L'origine ethnique** semble également avoir un rôle majeur sur le revenue. Plus particulièrement, le fait d'être **asiatique** ("country of birth mother/self/father=Vietnam" et "race=Asian or Pacific Islander") constitue un groupe de facteurs discriminants. Ces facteurs sont probablement aussi à mettre en corrélation avec le facteur "citizenship=Foreign born- Not a citizen of US".
4. **La stabilité familiale**, représentée par les attributs "marital stat", "live in this house 1 year ago" et "migration code" semble également jouer un rôle. Cependant, comme la modalité en question est "Not in universe", il s'agit plus d'un facteur indirect, qui semble suggérer un phénomène plus subtile qu'il est intéressant d'étudier de manière plus approfondie.
5. **Le niveau d'études** est aussi en partie relié au revenue. Le fait d'avoir fini le lycée sans diplôme ("education=12th grade no diploma") ou d'avoir un diplôme de profession ("education=Prof school degree (MD DDS DVM LLB JD)") de type médecin etc, semblent définir deux extrêmes : le premier a un impact négatif sur le revenue et le second en a un positif.
6. Au final, **les âges extrêmes** semblent également jouer un rôle, car le fait de **ne pas être actif professionnellement** ("full or part time employment stat=Not in labor force") est un facteur impactant. Et cet attribut peut sans doute être lié à l'âge.

5 Perspectives et réponses aux questions

Nous discutons maintenant de quelques points liés à la réalisation de ce test.

5.1 Perspectives

Par contrainte de temps, il s'agit d'une analyse assez primaire du jeu de données en question. Il y a notamment deux aspects qui méritent des études plus approfondies. Premièrement, le traitement des valeurs manquantes peut être pousser plus loin. Deuxièmement, et c'est le plus important, seul deux modèles très simples ont été mises en place, il convient d'étudier la performance des algorithmes plus performants. Par exemple, nous n'avons pas pu explorer les arbres de décisions intégrant nativement le traitement des variables catégorielles, ce qui a très probablement affecté la performance de la méthode. A cause de la quantité des variables catégorielles, il est également possible d'envisager des modèles bayésiens (réseau bayésien) qui reste des modèles facilement interprétable.

Au final, je tiens à dire qu'il s'agit d'un sujet d'analyse très intéressant, mais la contrainte en temps a fait que je n'ai pu qu'évoquer certaines pistes explorables.

5.2 Réponses aux questions posées dans l'email

Les parties les plus difficiles ont été :
- Le nettoyage des données a été une étape très chronophage pour moi.
- Je ne savais pas que le package *scikit-learn* de Python ne permet pas de traiter nativement les variables catégorielles, alors que je savais qu'il existe des méthodes qui en sont capables (par exemple la maximisation d'entropie en utilisant des probabilités). J'ai passé trop de temps à essayer de trouver une implémentation “reconnue".