Test Quantmetry de François Biller

Conventions

[numéro] *comme [1]* indique une référence (source) en fin de ce document

#[numéro] *comme #[1]* indique une référence à un commentaire dans le code python fourni

Ce document est accompagné de 3 fichiers contenant du code python :

- STATDES.py
- LOGREG.py
- SVCLASS.py

Pour exécuter le code il convient d'installer :

```
pip install researchpy
pip install seaborn
pip install -U scikit-learn
pip install numpy
pip install scipy
python -m pip install -U matplotlib
pip install pandas-profiling
```

Mes commentaires au sujet de la réalisation de ce test

J'ai pris beaucoup de plaisir à résoudre ce test. J'ai d'ailleurs à la fois appris mais aussi mis en application des savoirs acquis lors de mon mba en IA.

Ceci constitue mon 1er cas d'implémentation en python d'algo de machine learning. Je me demande pourquoi je n'ai pas fait cela plus tôt. Il y a 5 semaines je n'avais jamais codé en python ! J'ai étudié tous les jours pour passer et enchainer 1 à 2 certifications certifications coursera pour apprendre à programmer en python. Coursera prévoyait 6 semaine pour 1 certification je les faisais en moins de 5 jours.

J'aurai pu faire appel à des experts pour m'aider à résoudre cet exercice, cela n'a pas été le cas. Je n'ai utilisé que les ressources accessible sur internet.

J'espère qu'au travers de ce travail vous serez rassuré sur ma capacité à apprendre et appliquer vite, à comprendre les problématiques, trouver des solutions pour le projet innerve, ma motivation, pour vous rejoindre, et que cela aboutira à la poursuite du processus de recrutement.

1 - Décrivez le jeu de données.

Présentez seulement les analyses et éventuels retraitements qui vous paraissent les plus pertinents et faites une première conclusion sur les variables à sélectionner en vue de la prédiction du succès ou de l'échec d'une candidature.

Le code python dont il question dans cette partie est STATDES.py (pour "statistitques descriptives"). Il s'exécute de la façon suivante avec le nom du fichier contenant les données :

python3 STATDES.py -d data.csv

L'utilisation de la librairie pandas_profiling #[1] permet de générer un rapport détaillé sur les données sous forme d'un fichier HTML (rapport_Data_Set.html)

Le jeu de données peut être qualifié de conséquent (20000 lignes) et de bonne qualité :

- peu de données sont manquantes sur des cellules de certaines lignes. Par exemple l'âge, la couleurs des cheveux, la note, le salaire pour un total de 999 cellules. On fera le choix d'ignorer les lignes pour lesquelles il manque une valeur. Cette opération s'exécute avec une seule ligne en utilisant la librairie Pandas (cf #[2.1])

- Il y a bien 2 valeurs uniques pour les champs embauche et sexe, des occurrences pour chaque valeur unique de diplôme, spécialité.

Le rapport (rapport_Data_Set.html) révèle des incohérences pour les variables exp et age (valeurs <0). On fera le choix d'ignorer les lignes dont l'expérience est inférieure à 0 et les lignes dont l'âge est inférieur à 0. Nous retirons ces lignes incohérentes (cf #[2.2] e).

Toutefois on pourrait même éliminer lignes pour les âges < 10 si l'on sait qu'il y a une incompatibilité avec un âge <10ans

Attention l'utilisation de la variable couleur des cheveux à des fin de recrutement est rédhibitoire, cela entre en conflit avec un usage éthique de la donnée dans le cadre du recrutement. Bien que l'on ne soit pas ici dans une définition RGPD, puisqu'il n'est pas possible d'identifier une personne à partir de cette variable. La couleur des cheveux pourrait discriminer des candidats sur la base d'une apparence physique et cela est interdit par la loi.

1 - Y a-t-il une dépendance statistiquement significative

(a) entre la spécialité et le sexe ?

Ce sont 2 variables catégorielles (qui possède un nombre fini de catégories). Pour savoir si ces 2 variables sont liées on utilise le test du Khi-2 [1]. En testant l'hypothèse nulle : "les 2 variables sexe et spécialité sont indépendantes".

Nous nous servons de la librairie scipy et de la fonction chi2_contingency nous déterminons la table de contingence (cf #[3]). La p_valeur étant inférieure à 5% on peut donc rejeter cette hypothèse. Ces deux variables sont donc statistiquement dépendante.

Cependant la relation entre ces 2 variable est modeste, le calcul du V de cramer corrigé [2][5] dans #[4] donne une valeur = 0,37 (plus cette valeur est supérieure à 0,9 plus on peut qualifier la relation de très forte, plus cette valeur est inférieure à 1 plus on peut qualifier cette relation de faible.

(b) La couleur de cheveux et le salaire demandé ?

Ce sont 2 variables une est catégorielle (couleur des cheveux) l'autre est continue (salaire). *Pour étudier ce type de corrélation on a recours à l'analyse de la variance (ANOVA) [1] à un facteur qui permet de comparer les moyennes d'échantillon.*

Il s'agit de conclure sur l'influence d'une variable catégorielle sur la loi d'une variable continue à expliquer. Est-ce que la variable couleur des cheveux a une influence sur le salaire demandé ? Il y a 4 catégories pour cette variable.

Notons m1, m2, m3, m4 les moyennes des salaires demandés pour chacune des 4 catégories.

Le raisonnement simple à appliquer est de dire que si la variable "couleur des cheveux" n'a pas d'influence sur les salaires demandés alors les moyennes devrait être identiques m1=m2= m3=m4. C'est l'hypothèse que nous testons quand on a recours à l'analyse de la variance.

C'est ce que nous vérifions dans le tableau ANOVA [#6.2] par couleur de cheveux, celui-ci donne des valeurs quasi identique pour la moyenne ANOVA [#6.1] quelque soit la couleur de cheveux, qui est de l'ordre d'un salaire moyen de 35000€, et un écart type de 5000.

#[6.1] ANOVA ===salaire seul===

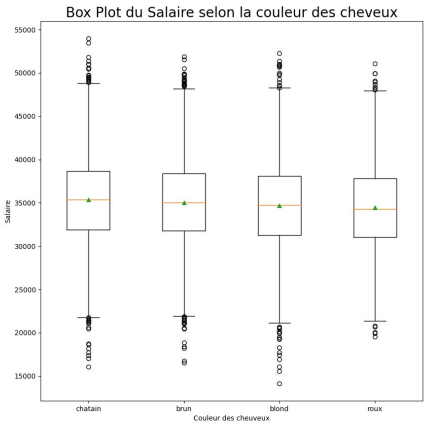
	Variable	N	Mean	SD	SE	95% Conf.	Interval
0	salaire	18991.0	34967.7434	5003.8361	36.3102	34896.572	35038.9147

#[6.2] ANOVA ===salaire groupé par type de cheveux===

cheveux	N	Mean	SD	SE	95% Conf.	Interval
blond	5644	34665.7271	5017.1326	66.7824	34534.8079	34796.6463
brun	5718	35027.1163	4936.2363	65.2791	34899.1446	35155.0880
chatain	5746	35381.5593	5006.8686	66.0516	35252.0733	35511.0454
roux	1883	34429.9304	5052.8025	116.4413	34201.5627	34658.2981

Cela se vérifie également quand on regarde le graphique construit avec les instructions de code [#6.5] et reproduit ci-dessous on voit bien que les 4 graphiques sont quasiment identiques.

Puisque les 4 moyennes sont identiques, donc la variable “couleur de cheveux n’a pas d’influence sur les salaires demandés. Donc il n’y a pas de corrélation entre les variables salaire et cheveux.



Nous avons codé le test statistique de F_oneway [3] en #[6.3] cependant nous n’arrivons pas à conclure. Le test indique avec une P_value 8.55e-18 étant inférieur à α < 0,05 rejette l’hypothèse nulle (autrement dit, il y a corrélation entre les 2 variables). Nous obtenons les mêmes valeurs avec le code #[6.6]

```
#[6.3]====foneway====
 F_onewayResult(statistic=27.58758818627994, pvalue=8.559223795642455e-18)
#[6.6]============table_anova==============
             df       sum_sq      mean_sq         F       PR(>F)
cheveux     3.0  2.063578e+09  6.878592e+08  27.587588  8.559224e-18
Residual  18987.0  4.734152e+11  2.493365e+07       NaN         NaN
```

Par contre le test de levene [4] [3] que nous avons codé en #[6.4] donne une pvalue = 0,26 > α (0,05) α étant le niveau de signification (typiquement 0,05). Ainsi, l'hypothèse nulle d'égale variance est validée et il est conclu qu'il n’existe pas une différence entre les variances dans la population. Donc la variable “couleur des cheveux” n’a pas d’influence sur les salaires demandés

```
#[6.4]====statlevene=====
 LeveneResult(statistic=1.3117225645651664, pvalue=0.2685841330956835)
```

(c) Entre le nombre d’années d’expérience et la note à l’exercice ?

Ce sont deux variable continues. Nous utiliserons alors le test de corrélation de Pearson [1]. L’hypothèse nulle à tester est : “les deux variables testées sont indépendantes”. Comme pour le test du chi-deux et de pearson, le test s’accompagne d’une p_valeur qui détermine le rejet ou non de l’hypothèse nulle.

Nous avons implémenté ce test dans notre code python #[7] et nous avons pour p-value = 0,15 > 0,05. On en déduit que les variables sont indépendantes. D’autre part le coefficient de pearson = -0,01, étant proche de 0 les variables sont bien décorrélées.

```
#[7] ======== Test de Pearson pour les variable ''exp'' et ''note''
             resultats_test
pearson_coeff      -0.010365
p-value             0.153182
```

2. Machine Learning

1. Concevez un modèle permettant de prédire la variable embauche et expliquez votre choix d’algorithme. Si votre modèle comporte des spécificités de paramétrage, justifiez également vos choix de paramètres.
2. Quelles sont les variables les plus importantes de votre modèle? Commenter.

3. Décrivez et justifiez le critère de performance utilisé.
4. Proposez deux à trois pistes d'amélioration de votre modèle.

1) Choix d'algorithmes et codes python

Nous pouvons utiliser plusieurs types d'algorithmes qui s'appliquent bien au problème proposé :

1) Nous pouvons utiliser une régression logistique [11] :

- la variable dépendante est bien binaire (embauche/pas embauché) et ce que l'on recherche à estimer (embauche)
- le jeu de données est d'un volume conséquent
- on enlèvera les variables qui ne sont pas utiles (indépendantes)
- les variables catégorielles devront être transformée en variables booléennes cf. #[1.4]

2) Les machines à vecteur de support (SVM) sont utilisé pour les tâches de classification et de régression. Ce qui est le cas de notre modèle. C'est une méthode d'apprentissage supervisée, adaptée à la prédiction de 2 résultats possible (ici embauche/pas embauche) en fonction de variables continue ou catégorielles. Ce qui est le cas de notre jeu de données [9].

Ces 2 méthodes nous sont confirmée par l'aide mémoire ML de Microsoft Azure ci-dessous [10]

Nous décidons d'implémenter ces 2 algorithmes et nous codons deux algos :

1) LOGREG.PY (python3 LOGREG.py -d [CSV] -t [value]) ou [CSV]=contient le nom du fichier de donnée CSV et [value] un ratio ou un pourcentage du fichier de donnée pour l'entraintement
2) SVCLASS.PY (python3 SVCLASS.py -d [CSV] -t [value]) ou [CSV]=contient le nom du fichier de donnée CSV et [value] un ratio ou un pourcentage du fichier de donnée pour l'entraintement

Pour ces 2 algorithmes nous rejetons les données suivantes :

- les colonnes inutiles #[1.1]. C'est la colonne couleur des cheveux, pour un recrutement, il est préférable de ne pas y avoir recours, nous supprimons également la colonne date, et l'index.
- données comportant des cellules vides #[1.2]
- incohérentes #[1.3]

Nous procédons à la transformation des variables catégorielles #[1.4] en variable booléennes. Car l'algorithme ne fonctionne qu'avec des valeurs numériques/

Nous constituons #[2] le jeux de données à l'aide de train_test_split(). La valeur [value] entrée en ligne de commande, donne la répartition du jeu d'entraînement et du jeux de test.

Nous procédons à l'élaboration du modèle de machine learning #[3], l'entraînement #[3.1], la prédiction #[3.2], son évaluation #[3.3] qui nous donne la précision, le rappel, le f1-score.

Résultat pour python3 LOGREG.py -d data_01.csv -t 80

	precision	recall	f1-score	support
0	0.88	1.00	0.94	3380
1	0.00	0.00	0.00	456
accuracy			0.88	3836
macro avg	0.44	0.50	0.47	3836
weighted avg	0.78	0.88	0.83	3836

Résultat pour python3 SVCLASS.py -d data.csv -t 80

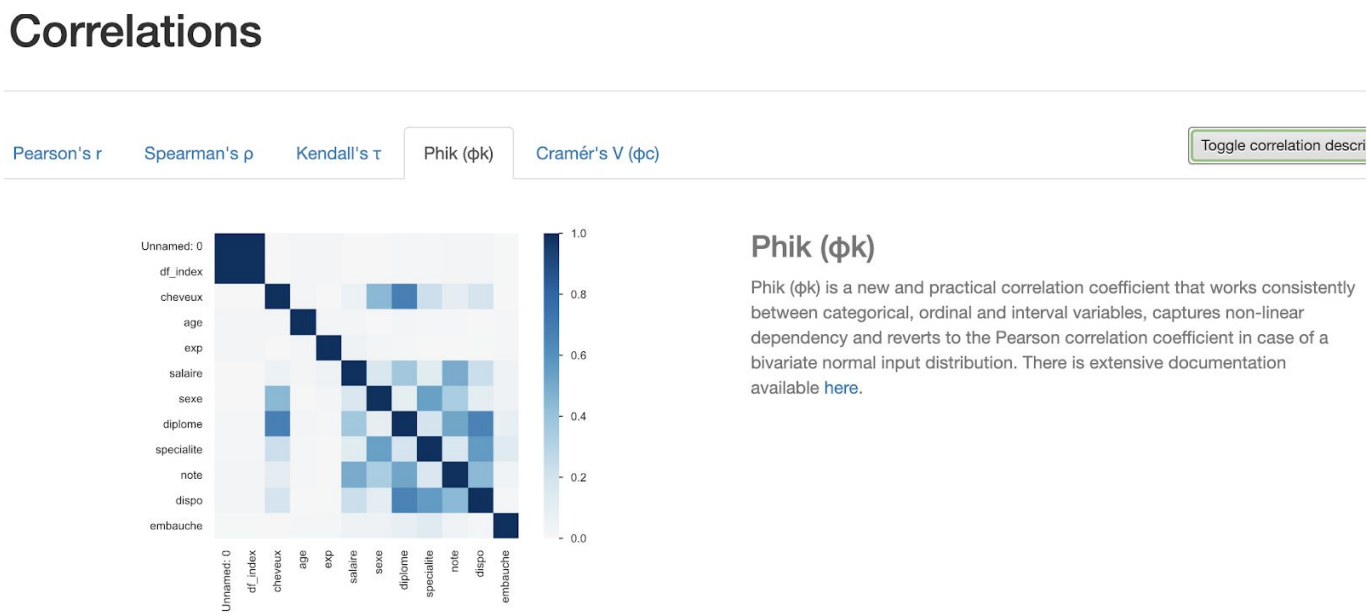
	precision	recall	f1-score	support
0	0.88	1.00	0.94	3380
1	0.00	0.00	0.00	456
accuracy			0.88	3836
macro avg	0.44	0.50	0.47	3836
weighted avg	0.78	0.88	0.83	3836

Pour les 2 algorithmes nous obtenons les mêmes résultats.

Cela dit ces résultats sont insatisfaisants puisque la matrice de confusion comporte le maximum d'erreurs de types 2 (le modèle a prédit une non embauche pour des personnes devant être embauchée). Nous avons essayé de comprendre pourquoi cela ne fonctionne pas pour la catégorie embauche et faire des recherche sur internet. Il ne nous a pas été possible de trouver une solution à cette erreur.

2 Quelles sont les variables les plus importantes de votre modèle ?

Le pandas profiling report obtenu en #[1.1] dans STATDES.py nous fourni un fichier rapport_Data_Set.html ce fichier nous présente un graphique de corrélations entre les différentes variables. Il y a une colonne embauche, on voit que les couleurs sont plus foncées (corrélation de Phik proche de 1) sont pour les variables spécialité, diplôme, salaire.



Pour notre modèle la table logit calculée grace à logit_model en #[3.4] dans LOGREG.py donne une indication des valeurs inutiles. En ce servant de la colonne P>Z, si la valeur excède 0,05 la valeur n'est pas pertinente [13] : donc pour notre modèle il faudrait supprimer :

- diplome_doctorat (p_value=0,12)
- exp (p_value=0,91)
- disponibilité (p_value=0,11)

3) Décrivez et justifiez le critère de performance utilisé.

Le critère de précision est donnée par logreg.predict (X_test) en #[3.5] pour LOGREG.py il est de 0,88 ce qui est un bon test de précision.

Si l'on compare les 2 algorithmes LOGREG.py et STATDES.py, LOGREG est plus performant il consomme moins de ressources machine et est plus rapide à calculer.

4) Proposez deux à trois pistes d'amélioration de votre modèle.

Si l'on suivant les recommandations de [13] on pourrait :

- Faire de l'élimination de variable (RFE)
- Over-sampling en utilisant SMOTE : Une fois nos données d'entraînement créées, procéder à un sur-échantillonnage de la non embauche en utilisant l'algorithme SMOTE (Synthetic Minority Oversampling Technique). A un niveau élevé, SMOTE : fonctionne en créant des échantillons synthétiques à partir de la classe mineure (non embauche) au lieu de créer des copies. En choisissant au hasard un des k-voisins les plus proches et en l'utilisant pour créer de nouvelles

Sources

[1] https://datascientest.com/correlation-entre-variables-comment-mesurer-la-dependance

[2] https://en.wikipedia.org/wiki/Cram%C3%A9r%27s_V

[3] https://www.pythonfordatascience.org/anova-python/

[4] https://fr.wikipedia.org/wiki/Test_de_Levene#

[5] https://stackoverflow.com/questions/20892799/using-pandas-calculate-cram%C3%A9rs-coefficient-matrix

[6] http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_Scikit_Learn_Decision_Tree.pdf

[7] https://lovelyanalytics.com/2016/08/20/random-forest-comment-ca-marche/

[8] https://makina-corpus.com/blog/metier/2017/initiation-au-machine-learning-avec-python-theorie

[9]https://docs.microsoft.com/fr-fr/azure/machine-learning/

[10] Aide-mémoire de l'algorithme Machine Learning Microsoft Azure

[11] https://towardsdatascience.com/logistic-regression-a-simplified-approach-using-python-c4bc81a87c31

[12] https://towardsdatascience.com/what-is-one-hot-encoding-and-how-to-use-pandas-get-dummies-function-922eb9bd4970

[13] https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8