Nous travaillons sous Python et Excel dans cet exercice.

Construction d'une grille de score

On souhaite scorer les personnes selon leur propension à souscrire un PEP (un produit financier quelconque) suite à une sollicitation commerciale. Le fichier "**bank-grille-score.xlsx**" comporte 600 observations et 7 variables, toutes qualitatives. **PEP** est la variable à prédire, "YES" est la modalité cible (positive).

Traitements sous Python

Ces tutoriels devraient vous aider **:**

- **Vidéo 1** : Régression logistique avec Python / Scikit-learn – Sélection de variables RFE (https://www.youtube.com/watch?v=ecre14FY5ZM)
- **Vidéo 2** : One Hot Encoding pour la régression logistique (https://www.youtube.com/watch?v=WfbJKLQPNEo)

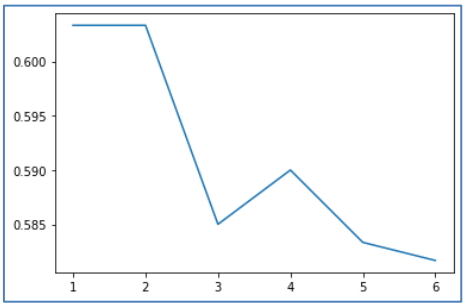
Régression logistique

1. Chargez le fichier de données dans un data.frame. Nous modélisons à partir de la totalité des données dans cet exercice. *In fine*, notre objectif est de produire sous Excel une grille de score équivalente au classifieur élaboré à l'aide de la régression logistique sous Scikit-Learn / Python.
2. Isolez dans des structures distinctes la variable cible (**PEP**) et les explicatives (les autres).
3. Affichez les fréquences absolues des classes (NO : 326, YES : 274).
4. Les explicatives sont toutes qualitatives, elles ne sont pas utilisables directement dans une régression logistique, il nous faut les coder en 0/1. Nous utilisons un codage disjonctif. Pour chaque variable, la modalité de référence, celle qui est omise, est la première (pandas.get_dummies, attention à l'option drop_first ; **Vidéo 2, 09:21**).
5. Combien de variables recodées 0/1 disposons-nous maintenant ? (**6** ; SEX_MALE … mortgage_YES). Affichez les 10 premières lignes de la matrice des explicatives.
6. Vérifiez que votre version de Scikit-learn est supérieure ou égale à **1.0**.
7. Construisez un modèle prédictif à l'aide de la régression logistique en utilisant les variables explicatives recodées (**Vidéo 2, 13:25**). Pour notre part, à la différence du tutoriel, nous conservons les paramètres part défaut, à l'exception de (random_state = 0) pour nous assurer d'obtenir les mêmes résultats.

8. Affichez les coefficients de la régression. Est-ce que nous pouvons les exploiter pour identifier les indicatrices (les variables 0/1) qui ont le plus d'impact dans le modèle ?

Sélection de variables

9. Instanciez un objet **StratifiedKFold** (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html) avec les paramètres suivants (voir la documentation pour comprendre leur signification) : n_splits = 10, shuffle = True, random_state = 0).
10. Instanciez un objet **RFECV** (https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html) (**Vidéo 1, 36:24**). Passez en paramètres la régression implémentée plus haut (estimator), l'objet validation croisée (cv). Nous optimisons le taux de reconnaissance (scoring = 'accuracy').
11. Affichez les performances pour chaque scénario (solution à 1 variable, à 2 variables, etc.). Attention, contrairement à ce que l'on voit dans la vidéo (**Vidéo 1, 40:34**), la propriété (.grid_scores_) est obsolète aujourd'hui (Scikit-learn > 1.0). Il nous faut plutôt regarder du côté de (.cv_results_) pour obtenir les performances estimées en validation croisée (moyennes des « accuracy » calculés pour chaque fold).
12. Tracez une courbe avec en abscisse le nombre de variables sélectionnées et les performances en validation croisée.



13. On souhaite travailler sur la solution à 2 variables (sinon la grille de score serait trop simple…). Affichez le contenu de la propriété (.ranking_) de l'objet RFECV. Lisez attentivement la documentation pour comprendre sa signification. Exploitez cette propriété pour extraire le nom des variables sélectionnées (married_YES, save_act_YES).
14. Réalisez une nouvelle régression avec ces 2 variables (régression = paramètres par défaut, sauf random_state = 0). Affichez les coefficients et l'intercept.
15. Effectuez une prédiction sur les données ayant servi à la modélisation. Calculez la matrice de confusion (en resubstitution donc). Elle nous servira de référence

lorsque nous construirons la grille de score sous Excel (on devrait retrouver exactement la même).

col_0	NO	YES
pep		
NO	242	84
YES	154	120

Traitements sous Excel

16. Nous revenons dans Excel Sur le fichier "bank-grille-score.xlsx". Nous souhaitons construire la grille de score à partir des coefficients et intercept du modèle simplifié (après sélection de variables) obtenu à l'étape précédente (sous Python). **Le score doit être calibré entre 0 et 1000**.
17. Calculez également le seuil d'affectation permettant d'attribuer une classe à un individu.
18. A l'aide de la grille de score, calculez les scores des individus de la base.
19. En déduire la prédiction pour chaque individu (en comparant le score au seuil)
20. A partir de PEP observé et la prédiction, construire la matrice de confusion.

Y a-t-il équivalence avec la matrice de confusion sous Python ? Si oui, tout va bien ; si non, il y a un gros problème.

Nombre de pep	PREDICT		
pep	NO	YES	Total général
NO	242	84	326
YES	154	120	274
Total général	396	204	600