Traitement des classes déséquilibrées (PYTHON + SCIKIT-LEARN)

Tutoriel de référence

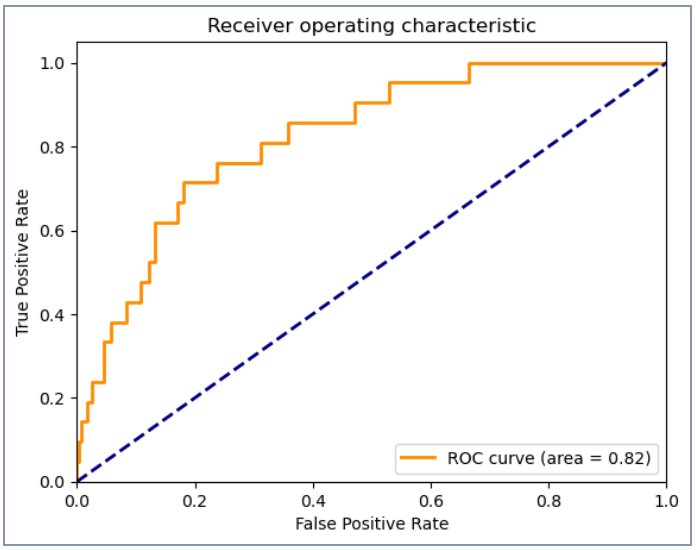
- Notre tutoriel de référence est : http://tutoriels-data-science.blogspot.com/p/tutoriels-en-francais.html#1766550852310168124 [**TUTO**].
- Ce site est également très instructif concernant les stratégies à mettre en place pour traiter les classes déséquilibrées : https://mlr.mlr-org.com/articles/tutorial/over_and_undersampling.html

Importation, caractérisation et préparation des données

1. Importez le fichier « **imbalanced_dataset.xlsx** ». Combien y a -t-il de variables et d'observations ? [info()] (3900 observations, 7 variables).
2. ''objective'' est la variable cible. Comptabilisez les effectifs par classe [value_counts()] (3831 negative, 69 positive ; pour être déséquilibré, c'est vraiment déséquilibré !).
3. Subdivisez les données en échantillons d'apprentissage (2700 obs.) et de test (1200 obs.) [**TUTO, page 8**]. Fixez (random_state = 1) pour que nous ayons la même partition ; demandez un tirage stratifié selon la variable cible.
4. Calculez les fréquences absolues des classes dans l'échantillon d'apprentissage (2652, 48) et de test (1179, 21).

Stratégie 1 – Données représentatives

5. Implémentez une régression logistique (modele1) [**TUTO, page 9**]. Fixez (**random_state = 1** et **solver = 'liblinear'**).
6. Lancez les calculs sur les données d'apprentissage [fit()] et affichez les coefficients du modèle [**TUTO, page 9**].
7. Calculez les scores (probabilité d'être positif) des individus de l'échantillon test (predict_proba) [**TUTO, page 17**]. A titre de vérification, quelle est la moyenne des scores ? (0.016). Qu'est-ce que cette valeur vous inspire ?
8. Affichez le boxplot des scores. Que constatez-vous ?
9. Calculez les valeurs de TPR (true positive rate) et FPR (false positive rate) (cf. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html) [roc_curve]. Affichez ensuite la courbe ROC et donnez la valeur de l'AUC [auc] (0.82).



10. Effectuez la prédiction sur ce même échantillon test [predict] (**TUTO, page 11**). Comptabilisez le nombre de prédictions négatives et positives (la prédiction est un vecteur numpy, cf. https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.unique.html) (1197, 3).

11. Calculez la matrice de confusion et le F1-Score (F1-Score = 0.08).

Moralité : *le modèle attribue préférentiellement un score plus élevé aux individus de la classe positive (AUC), mais la prédiction est mauvaise (F1-Score) parce que les scores d'appartenance sont sous-estimés (moyenne et distribution des scores).*

Stratégie 2 – Données équilibrées

12. On souhaite entraîner le modèle sur un échantillon d'apprentissage explicitement équilibré. Des données d'apprentissage ci-dessus, conservez tous les individus positifs (48), et effectuez un tirage aléatoire du même effectif parmi les négatifs (Remarque : pour ma part, j'ai isolé les négatifs dans un dataframe distinct, puis j'y ai effectué une extraction aléatoire en utilisant la fonction train_test_split de ''scikit-learn'', j'ai fusionné ensuite ces données avec le dataframe des positifs ; d'autres pistes sont possibles).

13. Dans le nouveau jeu d'entraînement, combien y a-t-il d'observations ? (96) De positifs et de négatifs ? (48 vs. 48)

14. Construisez le modèle (modele2) à partir de ces données et affichez les coefficients.

15. Calculer les probabilités d'appartenance aux classes sur l'échantillon test (predict_proba). Quelle est la probabilité moyenne d'être positif ? (0.29). Que vous inspire ce résultat ?

16. Affichez le boxplot des scores. Que constatez-vous ?

17. Calculez de nouveau l'AUC de la courbe ROC sur l'échantillon test (0.83). Le modèle élaboré à partir de l'échantillon d'entraînement équilibré est-il meilleur ? (pas tellement, au sens de la courbe ROC et de l'AUC en tous les cas).

18. Effectuez la prédiction sur l'échantillon test (predict), combien y a-t-il de prédictions positives maintenant ? (292). Etait-ce prévisible ? (oui, on donne plus d'importance aux positifs dans l'échantillon, mais est-ce vraiment à juste titre ?).

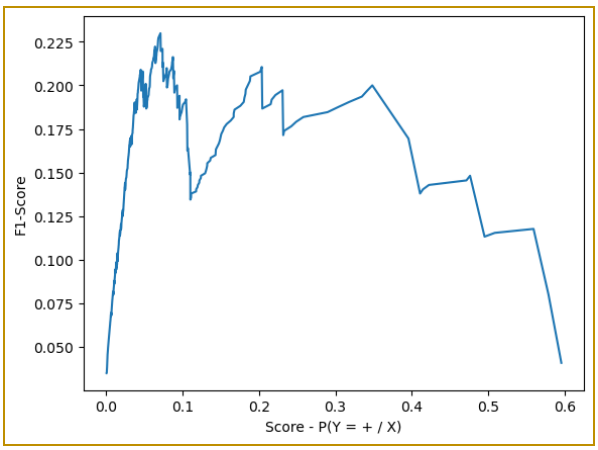
19. Calculez la matrice de confusion et le F1-Score. Que constatez-vous ? (F1-Score = 0.102). **Que faut-il en penser, quelle serait la cause de cette apparente contradiction : pas d'amélioration de l'AUC, mais amélioration en revanche du F1-Score ? (***la qualité intrinsèque du modèle n'est pas modifiée [c'est ce que révèle la courbe ROC], l'équilibrage des classes a simplement corrigé les scores vers des valeurs plus élevées, d'où une amélioration de la sensibilité qui a pesé sur le F1-Score***).**

Stratégie 3 – Données représentatives + ajustement du seuil d'affectation

20. Revenons sur le 1er modèle (modele1) élaboré sur la totalité de l'échantillon d'apprentissage. Nous allons essayer de mettre au point un procédé pour identifier le seuil d'affectation permettant d'optimiser le F1-Score en généralisation.

21. Calculez les scores [P(Y=+/X)] sur les données d'apprentissage (sur les 2700 observations). A titre de vérification, quelle en est la moyenne ? (0.0179).

22. Triez les données d'apprentissage selon ces scores décroissants et, pour chaque valeur du score, calculez le rappel et la précision. En déduire le vecteur des F1-Score (inspirez-vous de l'approche utilisée pour la construction de la courbe de gain, **TUTO, pages 17 et 18 ;** dans le tuto, on y voit comment construire le vecteur correspondant au rappel, il vous reste à produire les vecteurs pour la précision et pour le F1-Score). Représentez dans un graphique le couple : score (en abscisse) et F1-Score (en ordonnée).



Remarque : Une autre piste aurait été d'utiliser la courbe « précision - rappel » mais, dans ce cas, le seuil d'affectation n'apparaît pas explicitement (http://tutoriels-data-science.blogspot.com/p/tutoriels-en-francais.html#6907751194906372654)

23. A partir des données de ce graphique, identifier le seuil d'affectation qui permet d'optimiser le F1-Score sur l'échantillon d'apprentissage (**seuil = 0.07** ; F1-Score = 0.23).
24. Refaites la prédiction sur l'échantillon test (2000 obs.) en utilisant ce nouveau seuil (`si P(Y=+/ X) > seuil Alors y^ = ''+'' Sinon y^ = ''-''`). Comptabilisez le nombre de prédictions négatives et positives (1152 vs. 48).
25. Calculez la matrice de confusion et en déduire le F1-Score (F1-Score = 0.147).
26. **Conclusion : Que faut-il penser de tout ceci ?**

Remarque finale sur l'identification du seuil d'affectation : Nous avons utilisé l'échantillon d'apprentissage pour identifier le seuil optimal parce que le risque de surapprentissage était faible (ratio nombre d'observations sur nombre de prédicteurs élevé, classifieur – la régression logistique – relativement stable). Dans le cas contraire c.-à-d. s'il y a risque de surapprentissage, nous devons utiliser un échantillon à part (échantillon de validation) ou, à défaut, la validation croisée / leave-one-out pour le déterminer. En tous les cas, il était hors de question d'utiliser l'échantillon test pour cette tâche.