Machine Learning
Projet

M2 Informatique BI&A

Guillaume Metzler
Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr

Ce projet sera effectué par groupe de deux étudiants maximum et porte sur l'apprentissage dans un contexte de classification binaire.

Vous aurez le choix entre deux projets :

- un projet avec lequel vous travaillerez avec des données classiques de la communauté en Machine Learning.
- un sujet qui vous proposera de travailler avec des données réelles de détection de fraudes bancaires. La tâche de classification associée à ce jeu de données est beaucoup plus difficile et vous verrez rapidement que les performances que vous pourrez atteindre seront beaucoup plus faibles.

Le travail sera à rendre par mail à **guillaume.metzler@univ-lyon2.fr** **avant le 28 Janvier 2025**. Ce travail se composera d'un dossier retraçant vos démarches et résultats entrepris pour traiter le sujet, de plus amples explications vous sont données ci-dessous. Vous êtes libres d'utiliser le langage de votre choix pour effectuer ce travail, R ou Python. Choisissez celui avec lequel vous êtes le plus à l'aise.

Première partie

Premier Sujet : Apprentissage avec des Benchmarks

1 A propos des données et préparation

Vous pourrez trouver plusieurs jeux de données à l'adresse suivante

Téléchargement des jeux de données.

Vous pourrez également trouver un code Python permettant de mettre en forme les différents jeux de données afin que ces derniers soient prêts à être employés.

Préparation des données

On pourra effectuer les différentes étapes :

- regarder le code précédent afin d'identifier les différentes étapes
- détecter d'éventuelles valeurs manquantes et faire de l'imputation
- identifier les caractéristiques des jeux de données (taille, dimension, type de problème, ratio des classes)

Vous trouverez à la fin de ce document quelques indications pour la rédaction de votre rapport.

2 Travail à effectuer

Votre travail se décomposera en trois parties. Pour les différentes phases de votre travail, vous devrez

- présenter les algorithmes/méthodes employé(e)s
- le protocole expérimental afin que la présentation permette la reproduction des résultats obtenus
- concevoir des variantes personnelles des algorithmes standards (*i.e.* combinaison avec des méthodes d'échantillonnage, cost-sensitive, ...)
- évaluer le temps d'apprentissage des algorithmes

On prendra par exemple soin de préciser quelles sont les mesures de performances employées ou encore les loss optimisées dans le cas où cette dernière est personnalisée.

2.1 Approches non paramétrique

Dans cette première partie, vous concentrerez sur des approches basées sur l'algorithme des k- plus proches voisins uniquement.
Vous devrez rappeler le fonctionnement de l'algorithme et la comparer au moins avec trois variantes que vous proposerez en fonction des jeux de données considérées et des difficultés qu'ils présentent.

2.2 Approches paramétrique linéaires

Dans cette deuxième partie, on va chercher à comparer les algorithmes qui apprennent des classifieurs linéaires

- SVM linéaire (ou noyau linéaire, éventuellement gaussien, mais ce n'est pas très *fair* en pratique)
- Régression logistique

A nouveau, on proposera au moins deux variantes pour ces deux algorithmes qui permettent d'améliorer les performances de base de ces algorithmes.

2.3 Approches non linéaires

Dans cette dernière partie, on va chercher à tester les approches non linéaires ou par boosting

- Arbres de décisions/forêts aléatoires
- Adaboost
- Gradient Boosting

3 Quelques suggestions

Je vous donne ci-dessous une liste non exhaustive des méthodes que vous pourriez utiliser pour votre travail.

- **Pre-Process sur les données :** utilisation d'algorithmes d'over-sampling (random - SMOTE - Adasyn - ...) ou encore des approches d'under-sampling (random - Tomek Link - Edited Nearest Neighbout - NearMiss - ...) vous pouvez aussi utiliser des méthodes dites cost-sensitive qui vont accorder plus de poids aux exemples d'une classe donnée, voire même des poids spécifiques à chaque exemple.
- **Post-traitement :** combinez les résultats issus de différents modèles (bagging) afin de créer un modèle potentiellement plus puissant.

N'hésitez pas à regarder sur internet quelques exemples d'utilisations des algorithmes sus-mentionnés et votre objectif sera de les adapter au contexte des données (consulter des sites comme Kaggle - MachineLearningMastery ou encore Medium qui seront pour vous une bonne source d'inspiration).

Deuxième partie

Deuxième Sujet : Détection de Fraudes Bancaires

Vous pouvez télécharger les données à l'adresse ci-dessous

Lien de téléchargement

4 A propos des données

Les données sur lesquelles vous allez travailler sont des données réelles. Elles sont issues d'une enseigne de la grande distribution ainsi que de certains organismes bancaires (*FNCI* et *Banque de France*). Chaque ligne représente une transaction effectuée par chèque dans un magasin de l'enseigne quelque part en France, elles ne sont pas brutes et plusieurs variables sont déjà des variables créées, *i.e.* sont issues du *feature engineering*, nous avons un ensemble de 23 variables qui ont la signification suivante :

- **ZIBZIN** : identifiant relatif à la personne, *i.e.* il s'agit de son identifiant bancaire (relatif au chéquier en cours d'utilisation)
- **IDAvisAutorisAtionCheque** : identifiant de la transaction en cours
- **Montant** : montant de la transaction
- **DateTransaction** : date de la transaction
- **CodeDecision** : il s'agit d'une variable qui peut prendre ici 4 valeurs
  0 : la transaction a été acceptée par le magasin
  1 : la transaction et donc le client fait partie d'une **liste blanche** (bons payeurs). Vous n'en rencontrerez pas dans cette base de données
  2 : le client fait d'une partie d'une **liste noire**, son historique indique cet un mauvais payer (des impayés en cours ou des incidents bancaires en cours), sa transaction est alors automatiquement refusée
  3 : client ayant était arrêté par le système par le passé pour une raison plus ou moins fondée
- **VérifianceCPT1** : nombre de transactions effectuées par le même identifiant bancaire au cours du même jour
- **VérifianceCPT2** : nombre de transactions effectuées par le même identifiant bancaire au cours des trois derniers jours
- **VérifianceCPT3** : nombre de transactions effectuées par le même identifiant bancaire au cours des sept derniers jours
- **D2CB** : durée de connaissance du client (par son identifiant bancaire), en jours. Pour des contraintes légales, cette durée de connaissance ne peut excéder deux ans

- **ScoringFP1** : score d'anormalité du panier relatif à une première famille de produits (ex : denrées alimentaires)
- **ScoringFP2** : score d'anormalité du panier relatif à une deuxième famille de produits (ex : électroniques)
- **ScoringFP3** : score d'anormalité du panier relatif à une troisième famille de produits (ex : autres)
- **TauxImpNb_RB** : taux impayés enregistrés selon la région où a lieu la transaction
- **TauxImpNB_CPM** : taux d'impayés relatif au magasin où a lieu la transaction
- **EcartNumCheq** : différence entre les numéros de chèques
- **NbrMagasin3J** : nombre de magasins différents fréquentés les 3 derniers jours
- **DiffDateTr1** : écart (en jours) à la précédente transaction
- **DiffDateTr2** : écart (en jours) à l'avant dernière transaction
- **DiffDateTr3** : écart (en jours) à l'antépénultième transaction
- **CA3TRetMtt** : montant des dernières transactions + montant de la transaction en cours
- **CA3TR** : montant des trois dernières transactions
- **Heure** : heure de la transaction
- **FlagImpaye** : acception (0) ou refus de la transaction (1)

Remarque La variable *CodeDecision* n'est pas une variable à utiliser pour faire de la prédiction car cette information est acquise post-transaction. On peut en revanche s'en servir lors de la phase d'apprentissage pour analyser les données par exemple.

Vous disposez donc d'un jeu de données comprenant 10 mois de transactions du "2017-02-01" au "2017-11-30". A vous de voir si toutes ces informations sont nécessaires ou non pour établir le modèle.

On définira les ensembles de la façon suivante :

- **Apprentissage :** transactions ayant eu lieu entre le "2017-02-01" et le "2017-08-31".
- **Test** : transactions ayant eu lieu entre le "2017-09-01" et le "2017-11-30"

5 Travail à effectuer : première partie

La variable à prédire est la variable *FlagImpaye*, il s'agit d'une variable qui ne peut prendre que deux valeurs possibles : 0 la transaction est acceptée et considérée comme "normale", 1 la transaction est refusée car considérée comme "frauduleuse".

Nous avons vu que plusieurs critères peuvent être utilisées pour évaluer la performance d'un modèle comme l'Accuracy, la précision, le rappel, la F-mesure ou encore

l'aire sous la courbe ROC (AUC ROC). Dans le cas présent, vous allez chercher à établir le modèle vous permettant d'obtenir les meilleurs résultats en classification en terme de F-mesure F dont la définition est rappelée ci-dessous :

$$F = \frac{2TP}{2TP + FN + FP},$$

où un TP est une fraude prédite fraude par votre modèle, un FN est une fraude non identifiée comme tel par votre modèle, un FP est une transaction non frauduleuse mais identifiée comme frauduleuse par votre modèle et enfin un TN est une transaction non frauduleuse.

Pour présenter votre travail, vous pourrez vous appuyer sur le modèle présenté en Annexe du présent document et suivre les indications sur le contenu des différentes sections.

6 Quelques suggestions

Idéalement, le travail effectué devrait comprendre au moins 5 procédures ou méthodes différentes vues en cours : une méthode peut être un algorithme de classification seul ou encore couplé ou non à une méthode d'échantillonnage par exemple. Je vous donne ci-dessous une liste non exhaustive des méthodes que vous pourriez utiliser pour votre travail.

- **Pre-Process sur les données** : utilisation d'algorithmes d'over-sampling (random - SMOTE - Adasyn - ...) ou encore des approches d'under-sampling (random - Tomek Link - Edited Nearest Neighbout - NearMiss - ...) vous pouvez aussi utiliser des méthodes dites *cost-sensitive* qui vont accorder plus de poids aux exemples d'une classe donnée, voire même des poids spécifiques à chaque exemple.
- **Algorithmes** : vous pourrez utiliser des algorithmes non supervisés comme des méthodes de clustering (k-means, clustering hiérarchique ou encore les auto-encodeurs) pour détecter les fraudes. Vous disposez également d'un large éventail d'algorithmes de classification supervisés que vous pouvez utiliser : decision trees, random forests, gradient boosting, nearest-neighbors, réseaux de neurones profonds, SVM (linéaire ou non ...), analyse discriminante, boosting, ...
- **Post-traitement :** combinez les résultats issus de différents modèles (bagging) afin de créer un modèle potentiellement plus puissant.

N'hésitez pas à regarder sur internet quelques exemples d'utilisations des algorithmes sus-mentionnés et votre objectif sera de les adapter au contexte des données

(consulter des sites comme Kaggle - MachineLearningMastery ou encore Medium qui seront pour vous une bonne source d'inspiration). Vous verrez que toutes les méthodes ne sont pas forcément applicables à ce type de données : si tel est le cas, n'hésitez pas à préciser dans votre rapport pourquoi une méthode n'a pas fonctionné selon vous.

Ce qui suit est une suggestion de rédaction pour le rapport de votre projet.

Introduction

Vous présenterez rapidement le contexte du projet les objectifs et vous pourrez également présenter les caractéristiques des jeux de données considérées.

Méthodologie

Pour chaque partie précédemment citée, vous procéderez de la façon suivante

Vous commencerez par présenter les notations que vous allez employer tout au long de la rédaction de votre rapport.

Vous présenterez ensuite les outils que vous allez utiliser dans la partie expérimentale. On commencera par parler de ce que l'on souhaite maximiser, *i.e.* la mesure de performance en la définissant avant de s'attaquer à la présentation des algorithmes (de façon très courte)

Par exemple, si vous faites une contribution basée sur du boosting et que vous combinez avec des méthodes à noyaux, il faudra rappeler ce qu'est le boosting, ce que sont les méthodes à noyaux (ce sont les approches de bases) et ensuite vous expliquerez comment vous combinez les deux pour résoudre le problème confié.
Il ne faut pas hésiter à présenter le processus de façon *abstraite*, c'est-à-dire avec des notations mathématiques et ne pas être uniquement verbeux.
Un pseudo-code est également appréciable pour synthétiser l'approche proposée.

Dans le cas ou vous proposez plusieurs approches à des fins de comparaisons, il faudra prendre soin de présenter les différentes approches et de justifier pourquoi vous intéressez à ces approches là.

Remarque : il n'est pas nécessaire de présenter tous les algorithmes employés, mais uniquement ceux qui servent à l'élaboration d'une version "exotique".

Expériences

Vous dresserez ensuite votre protocole expérimentale qui présentera la ou les méthodes sélectionnées pour répondre à la tache demandée. Celui-ci comprend en général

trois parties

Protocole expérimental

Pour chaque partie, vous procéderez de la façon suivante
On commencera par présenter les données que l'on utilise, en général sous la forme d'un tableau qui comprend : *le nom du jeu de données*, *le taille de l'échantillon*, *la dimension du jeu de données* et toute autre information que vous pourriez juger pertinente.

Vous présentez rapidement les expériences que vous allez faire, *i.e.* les différents algorithmes testés, le range des hyper-paramètres employés ainsi que la façon dont sont optimisées ces hyper-paramètres (cross-validation en k-folds, simple validation ou est-ce que vous faites le choix de les fixer). Quels sont vos ensembles d'entraînement/validation/test ?

Les informations que vous fournissez dans cette section doivent permettre au lecteur de pouvoir reproduire les résultats que vous allez présenter dans la sous-section suivante.

Résultats

Ici vous allez présenter et analyser les résultats obtenus à l'aide de graphiques et/ou tableaux. Outre les performances, on pourra aussi s'intéresser au critère de rapidité d'un algorithme.

L'analyse doit aussi permettre de mettre en exergue les avantages inconvénients des méthodes proposées. Cela peut passer par l'utilisation d'autres mesures de performances/critères pour évaluer/comparer vos algorithmes.

Pour ce qui est des résultats, on pourra présenter les résultats sous la forme d'un tableau similaire à celui présenté en Table 1.

Conclusion

Il s'agit de conclure quant à votre étude. Reprendre le travail proposé et les principales conclusions. Il est également important de proposer des perspectives à votre travail en fonction des résultats obtenus et de l'approche proposée. Quelle(s) méthode(s) non explorées ici auraient pu être utilisées pour améliorer les résultats et pourquoi cela vous semble pertinent étant données les expériences effectuées.

TABLE 1 – Mean test F1 over 10 iterations

Dataset	knn	svm_linear	svm_poly	svm_gauss
00.62% abalone20	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	**0.0** ± 0.0
01.39% abalone17	**13.7** ± 6.3	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
02.36% yeast6	**44.0** ± 14.6	0.0 ± 0.0	38.9 ± 18.0	0.0 ± 0.0
03.31% wine4	**5.0** ± 5.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
06.67% libras	81.1 ± 12.7	71.0 ± 10.0	**88.3** ± 7.9	75.6 ± 11.0
10.23% pageblocks	**84.9** ± 2.2	75.7 ± 2.9	82.3 ± 2.5	78.6 ± 2.5
10.98% yeast3	66.8 ± 5.9	65.6 ± 10.8	**76.4** ± 5.2	71.3 ± 8.2
13.60% abalone8	**22.7** ± 1.3	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
14.29% segmentation	**85.1** ± 3.3	45.8 ± 3.7	64.1 ± 4.9	50.0 ± 4.6
22.73% hayes	14.8 ± 10.2	0.0 ± 0.0	**39.4** ± 19.9	0.0 ± 0.0
23.52% vehicle	88.1 ± 3.4	60.3 ± 5.3	**92.6** ± 1.7	56.7 ± 6.5
30.00% german	41.3 ± 6.4	5.3 ± 7.6	**56.9** ± 6.8	16.8 ± 6.6
32.71% glass	**70.2** ± 4.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
33.15% wine	86.5 ± 4.3	89.6 ± 5.9	**90.1** ± 5.8	86.2 ± 6.1
34.90% pima	41.3 ± 6.6	5.0 ± 5.4	**44.7** ± 6.1	17.5 ± 4.8
35.90% iono	82.5 ± 4.1	88.9 ± 4.0	92.0 ± 3.4	**93.3** ± 3.4
37.50% autompg	82.4 ± 5.5	0.0 ± 0.0	**83.0** ± 3.4	0.0 ± 0.0
46.08% balance	95.3 ± 1.9	94.5 ± 1.2	**99.9** ± 0.3	97.3 ± 0.8
Mean	**55.9** ± 5.4	33.4 ± 3.2	52.7 ± 4.8	35.7 ± 3.0
Average Rank	2.00	3.72	1.72	2.56