



M2 ILSEN/IA – 2023/24

UE Business intelligence & Systèmes décisionnels

ECUE Application Business Intelligence

Vincent Labatut

AVIGNON
UNIVERSITÉ

Projet | Financement participatif

1 Présentation

François Pignon a envie de développer un jeu de construction permettant de réaliser des maquettes des sept merveilles du monde avec des allumettes. Il décide de passer par un financement participatif, via une plateforme dédiée. Comme il n'a jamais mené de projet de ce type, il se demande quelle est la meilleure stratégie à employer.

Justement, il dispose de données décrivant des projets soumis sur un site permettant de mettre en place ce type de financements. Il se propose de les analyser afin de voir s'il est possible d'en tirer des enseignements susceptibles d'augmenter ses chances de succès. Pour ce faire, il aborde la question sous la forme d'un problème de classification : peut-on, à partir des données disponibles, estimer si un projet a été financé ou pas ? Et si oui, quels sont les attributs les plus pertinents ?

Le but de ce projet est de répondre à ces questions, en appliquant la chaîne complète de fouille de données vue en cours et en TP : préparation des données, prédiction, calcul de performance, comparaison de performances, etc.

2 Description des données

Les données sont accessibles sur e-UAPV sous la forme d'un fichier CSV. Celui-ci comporte plus de 300 000 campagnes de financement participatif, décrites par les champs listés dans la Table 1.

Attribut	Signification
id	numéro unique assigné au projet par la plateforme
name	nom du projet
category	thème principal du projet (ex. <i>art</i>)
subcategory	sous-thème du projet (ex. <i>peinture, céramique, illustration...</i>)
country	pays de l'utilisateur qui porte le projet
sex	sexe du porteur
age	âge du porteur, exprimé en années
start_date	date de début de l'appel à financement du projet
end_date	date de fin de l'appel à financement du projet
currency	devise dans laquelle les montants sont exprimées (ex. <i>dollar, euro...</i>)
goal	montant demandé pour financer le projet (exprimé dans la devise currency)
pledged	montant promis par les utilisateurs soutenant le projet (lui aussi en currency)
backers	nombre d'utilisateurs soutenant le projet
state	état courant de l'appel à financement du projet (ex. <i>échec, succès, annulé...</i>)

Table 1. Attributs des données décrivant les campagnes Kickstarter.

Pour mémoire, le principe de fonctionnement d'une plateforme de financement participatif est le suivant. Un utilisateur propose d'abord un projet, qu'il décrit en détail. Il ouvre un appel à financement, qui s'étale sur une certaine période fixée à l'avance, et vise à réunir une certaine somme d'argent jugée nécessaire à la réalisation du projet. Pendant cette période, d'autres utilisateurs peuvent promettre des dons. Le projet est généralement réalisé

si, à la fin de la période, la somme totale des promesses atteint la limite fixée. Toutefois, il peut arriver aussi qu'un projet soit réalisé même si les fonds n'atteignent pas la limite fixée, et inversement.

3 Préparation des données

Avant de réaliser la classification proprement dite, il est nécessaire de préparer les données, en suivant un certain nombre d'étapes standard.

Exploration. Il est recommandé, dans un premier temps, de naviguer manuellement dans les données afin de mieux les appréhender, et d'extraire quelques statistiques descriptives. Pour chaque attribut, quelles sont les valeurs moyennes, modales, les quantiles, etc. ? Plus généralement, comment les attributs sont-ils distribués ? Produisez des graphiques pour visualiser ces résultats, et utilisez-les pour illustrer votre analyse dans le rapport rendu.

Nettoyage. Puis se pose la question de la préparation des données proprement dite. Ces données nécessitent-elles un nettoyage ? Faut-il écarter certaines instances qui ne sont pas liées au problème ? Y a-t-il des valeurs manquantes ? Des valeurs aberrantes ? Des attributs redondants ? Des attributs superflus ? Les valeurs numériques correspondent-elles vraiment à des attributs de nature numérique, ordinale, ou catégorielle ? Comment traiter ces différents problèmes ?

Recodage. En fonction des algorithmes de fouille que vous allez appliquer, il peut être nécessaire de recoder certains champs : discrétisation d'attributs réels, catégorisation d'attributs numériques, normalisation d'attributs numériques, numérisation d'attributs catégoriels... Certains outils ne peuvent pas du tout être appliqués sur des données dont le codage n'est pas approprié. D'autres fonctionneront mieux pour certains codages. Il est recommandé de tester l'effet du codage sur les différents outils considérés.

Prétraitement. Des méthodes de prétraitement peuvent être appliquées avant de réaliser le traitement proprement dit. Par exemple, effectuer une réduction de la dimension des données, peut permettre de rendre le problème traitable, computationnellement parlant (i.e. faire que l'outil de fouille s'exécute en un temps raisonnable), ou bien d'améliorer la qualité et/ou la lisibilité des résultats. Mais le prétraitement peut aussi les rendre difficiles à interpréter. Là encore, il est possible de tester différentes méthodes avec différents paramétrages.

Découpage. Une fois les données pré-traitées, il est nécessaire de les découper en trois, de manière à obtenir des sous-ensembles d'*apprentissage*, de *validation* et de *test*. Rappelons que le premier jeu de données sert uniquement à construire le modèle de classification, tandis que le deuxième sert à estimer les hyper-paramètres (aussi appelés méta-paramètres) et à comparer les différents modèles construits, afin de sélectionner le meilleur. Le troisième jeu sert à estimer comment les performances de ce meilleur modèle se généralisent.

4 Méthodes de classification

Après avoir préparé les données, vous devez produire l'outil prédictif.

Sélection. Vous devez appliquer 4 algorithmes de classification. Trois d'entre eux vous sont imposés : il s'agit du *séparateur à vaste marge* (SVM), de la méthode des *k plus proches voisins* (*k*NN), et du classificateur bayésien naïf (*Naive Bayes*). Le quatrième est libre. Vous devez donc passer en revue les autres outils de fouille disponibles (en privilégiant ceux utilisés en cours et en TP), puis sélectionner celui qui vous paraît approprié. Notez que certains d'entre eux peuvent nécessiter une modification supplémentaire des données, comme par exemple un recodage (et parfois, ce recodage peut se faire de différentes façons). Il est donc tout à fait possible d'avoir un prétraitement différent pour chaque outil sélectionné, et même plusieurs prétraitements possibles pour un même outil. Cependant, les trois jeux de données doivent toujours être les mêmes (dans le sens qu'ils doivent toujours contenir les

instances fixées à l'étape de découpage), sinon les performances obtenues ne seront pas comparables.

Apprentissage. Appliquez ensuite ces algorithmes à vos données d'*apprentissage*, afin d'estimer leurs paramètres. Notez que certains outils ne font pas à proprement parler de l'apprentissage, mais nécessitent quand même d'avoir accès aux données d'apprentissage lors du traitement de nouvelles instances. Par exemple, k -NN estime la classe d'une nouvelle instance en fonction de celles des instances voisines dans le jeu d'apprentissage. Si plusieurs prétraitements alternatifs ont été identifiés pour un algorithme donné, son apprentissage doit être effectué séparément pour chacun d'eux.

Comparaison. Une fois l'apprentissage réalisé, appliquez vos classificateurs aux données de *validation*. Il va de soi que celles-ci doivent subir le même prétraitement que les données d'apprentissage. Cette étape vous permet de comparer non seulement les différentes méthodes sélectionnées, mais aussi les différents prétraitements identifiés pour une même méthode, et/ou les différents paramétrages possibles pour une même méthode. Par exemple : pour k -NN on testera plusieurs valeurs de k , tandis que pour SVM on décidera de la meilleure valeur de seuil de décision à appliquer au score produit. Pour réaliser vos comparaisons, vous devez déterminer avec quelles *mesures* vous allez quantifier la performance des outils de fouille, et comment vous allez établir la *signification statistique* des résultats obtenus.

Raffinage. Il est probable que vous deviez utiliser une approche itérative par *raffinage*, car plusieurs facteurs peuvent influencer les performances d'un outil de fouille : paramètres spécifiés par l'utilisateur, nature de la normalisation appliquée aux données, etc. L'approche par raffinage consiste à obtenir des premiers résultats sur le jeu de validation de façon un peu grossière, puis à les améliorer en effectuant plusieurs passes basées sur des analyses successives des résultats, et en faisant varier les facteurs pertinents. Attention toutefois, l'étape de *test* ne fait pas partie de ce cycle : elle ne doit être réalisée qu'une seule fois, sur le modèle sélectionné à l'issue de la dernière validation.

Évaluation. Une fois le raffinage *terminé*, vous pouvez effectuer la comparaison définitive des différentes combinaisons de prétraitement/méthode/paramétrage que vous avez considérées, et ainsi identifier la plus performante sur les données de validation. Vous devez alors l'appliquer aux données de test, afin d'établir son niveau de généralisation, pour répondre à la question : est-ce que le niveau de performance du modèle se maintient quand on classe d'autres données ?

Interprétation. Après avoir obtenu un modèle aux performances jugées suffisantes, il faut comprendre comment celui-ci effectue ces prédictions. Concrètement, il s'agit d'identifier quels attributs (et quelles valeurs de ces attributs) sont importants dans le processus de prédiction. Le but est ici de répondre à la question posée dans le sujet à propos de la pertinence des attributs.

5 Implémentation

Vous devez fournir un script (ou un ensemble de scripts) en Python (le langage est imposé) qui, une fois lancé, effectuera l'intégralité du traitement à partir des fichiers originaux : préparation des données, application des algorithmes de fouille, calcul des performances, comparaison des algorithmes, etc. Aucune étape ne doit faire l'objet d'une intervention manuelle, de manière à pouvoir être facilement reproduite par la suite.

La manière dont ce script doit être exécuté devra être clairement expliquée à la fois dans le rapport (cf. la Section 2.4 du modèle de rapport mentionné en Section 6) et dans un fichier `readme.txt` à placer dans le dossier contenant le(s) script(s).

Tout ce qui peut être réalisé avec les bibliothèques utilisées en cours et TP (prétraitement des données, apprentissage des outils de fouille, calcul et comparaison des performances...) doit l'être en priorité. Si vous avez besoin de fonctionnalités supplémentaires, vous pouvez

utiliser d'autres bibliothèques que celles-ci, mais *cela doit être justifié dans le rapport* (et le mieux est d'en discuter oralement en séance avec l'encadrant). Tout le reste du traitement doit être implémenté dans le script lui-même.

6 Rapport

En plus de votre code source (script, fichiers de configuration...), vous devez rendre un rapport décrivant le traitement que vous avez mis en place pour résoudre le problème proposé.

Structure et forme du rapport. Le plan du rapport est disponible en ligne sur Overleaf, à l'adresse suivante :

<https://www.overleaf.com/read/hbqywmgkwjgp>

Ce plan de rapport n'est accessible qu'en lecture seule. Donc, si vous décidez d'utiliser \LaTeX pour écrire votre rapport, vous devez d'abord en créer une copie avant de pouvoir l'éditer. Le rapport rendu doit être conforme aux instructions contenues dans le tutoriel suivant :

<https://www.overleaf.com/latex/templates/modele-rapport-uapv/pdbgdpszgwrp>

Le plus simple est donc pour vous de cloner le tutoriel ci-dessus, qui est aussi un modèle Overleaf, et d'y copier-coller le code \LaTeX du plan de rapport.

Notez que vous n'êtes pas tenus d'utiliser \LaTeX : n'importe quel autre outil fait l'affaire, tant que le rapport rendu prend la forme d'un PDF et est correctement mis en forme. En revanche, la structure du rapport est imposée, vous devez la suivre *obligatoirement*, en respectant les titres et la numérotation indiquée. De plus, la gestion de la bibliographie doit respecter les standards \LaTeX (cf. le tutoriel indiqué ci-dessus).

Utilisation de ressources. Vous avez le droit (et c'est même recommandé) d'utiliser n'importe quelle ressource qui pourra vous aider dans votre travail : rapports, articles, code source, pages Web, etc. La seule restriction est que vous ne pouvez pas utiliser des ressources produites par d'autres groupes de ce projet.

De plus, toute ressource doit explicitement être indiquée dans le texte de votre rapport, là où elle est pertinente. Le détail de la source doit apparaître dans la dernière section du rapport (bibliographie), comme expliqué dans le tutoriel \LaTeX .

Avertissement : L'utilisation (citée ou non) d'une ressource issue d'un autre groupe, et l'utilisation non-citée ou incorrectement citée d'une ressource extérieure constituent des plagats. En cas de plagiat, les groupes concernés seront sanctionnés en conséquence. Vous trouverez plus de détail sur la notion de plagiat dans le tutoriel \LaTeX cité précédemment.

7 Organisation

Le projet est à réaliser en groupes de deux ou trois personnes. Les étapes *très fortement recommandées* pour ce travail sont les suivantes :

1. Explorez les données, détectez les erreurs et problèmes et corrigez-les, nettoyez les données et effectuez leur analyse descriptive. Rédigez la partie du rapport correspondante.
2. Identifiez et étudiez les outils de fouille disponibles et susceptibles de résoudre le problème consistant à classer les clients de façon supervisée. Considérez notamment les paramètres possibles et les types de données supportés. Rédigez la partie correspondante du rapport.
3. Identifiez la préparation des données à effectuer pour que celles-ci soient exploitables par les outils sélectionnés. Il est possible de prévoir plusieurs types de préparations pour le même outil, par exemple dans le but de déterminer laquelle de ces préparations est la meilleure. Écrivez les scripts implémentant la préparation, et rédigez la partie

correspondante du rapport.

4. Développez les scripts permettant d'invoquer les outils de fouille sélectionnés. Le passage à la pratique peut vous révéler de nouvelles informations concernant ces outils, à intégrer dans la partie de votre rapport qui décrit les outils sélectionnés. Vérifiez leur fonctionnement sur une partie des données d'apprentissage, si celles-ci sont trop grandes.
5. Une fois que vos scripts fonctionnent, appliquez ces scripts aux données d'apprentissage. **Les outils effectuant de l'apprentissage doivent exporter les modèles qu'ils ont estimés, afin de pouvoir les réutiliser par la suite sans avoir besoin de refaire l'apprentissage.**
6. Sélectionnez les mesures de performance et les tests statistiques destinés à évaluer et comparer les classificateurs. Écrivez les scripts permettant de les appliquer. Rédigez la partie correspondante du rapport.
7. Évaluez les performances issues de l'apprentissage (cela doit être scripté, bien sûr) et discutez-les. Rédigez la partie correspondante du rapport.
8. En réutilisant les modèles issus de l'apprentissage, évaluez les performances des classificateurs sur les données de validation. Là encore, cela doit être scripté. Commentez, comparez, et identifiez la meilleure combinaison de paramétrage/prétraitement/outil de fouille. Rédigez la partie correspondante du rapport.
9. Appliquez cette meilleure combinaison (toujours en réutilisant le même modèle) aux données de test (là encore : à scripter). Évaluez ses résultats, discutez, complétez la partie correspondante du rapport.
10. Effectuez l'interprétation du modèle sélectionné. Discutez les attributs pertinents (et leurs valeurs) identifiés relativement aux questions du sujet, complétez la partie correspondante du rapport.
11. Finalisez le rapport. S'il reste du temps, vous pouvez tester des prétraitements ou des outils supplémentaires.