Retour d'expérience sur l'analyse des données d'un tunnelier

Marie Le Guilly, Jean-Marc Petit, Vasile-Marian Scuturici

Retour d'expérience sur l'analyse des données d'un tunnelier

Marie Le Guilly
Université de Lyon, Insa de Lyon, CNRS, LIRIS
Villeurbanne, France

Jean-Marc Petit
Université de Lyon, Insa de Lyon, CNRS, LIRIS
Villeurbanne, France

Marian Scuturici
Université de Lyon, Insa de Lyon, CNRS, LIRIS
Villeurbanne, France

ABSTRACT

Ce papier présente un retour d'expérience tiré d'une étude réalisée pour une entreprise de travaux publics d'un grand groupe industriel, portant sur la valorisation des données générées par un tunnelier. Les tunneliers sont d'énormes machines, ressemblant à des "petits trains" équipées de nombreux capteurs pour le guidage, le forage, l'excavation des matériaux, ... L'objectif était de déterminer si les données issues d'un tunnelier permettaient de prédire des événements sur le chantier, par exemple pour améliorer la maintenance prédictive du tunnelier et/ou réduire les coûts liés aux incidents. L'étude a duré six mois et a permis de toucher les étapes classiques de ce type de projet : identification des sources de données, intégration des données, analyse des données et validation avec les experts métiers. L'étude a montré l'intérêt des données pour l'entreprise, tout en pointant qu'une amélioration significative des résultats ne serait possible qu'en améliorant les processus menant à l'acquisition des données lors du fonctionnement d'un tunnelier. Nous tirons quelques leçons de cette expérience qui a servi au lancement d'une offre plus large de valorisation des données, nommée `DataValor`, au sein du LIRIS pour faciliter le transfert entre le monde académique et le monde industriel.

KEYWORDS

Tunneliers, Intégration de données, Apprentissage automatique, Valorisation des données, Retour d'expérience de transfert industriel

Utilisés sur des chantiers de grande envergure afin de creuser des tunnels, les tunneliers sont des machines pouvant atteindre neuf mètres de diamètre pour plus d'une centaine de mètres de long.

Ces machines génèrent, tout en creusant, un volume important de données, issues des très nombreux capteurs (plus de 1200) chargés du monitoring permettant de visualiser en temps réel l'avancée du tunnelier via des applications dédiées au monitoring. Ces données donnent ainsi des indications au conducteur du tunnelier, ainsi qu'aux personnes pouvant suivre à distance l'avancement du creusement du tunnel, et pouvant intervenir si certaines alertes se déclenchent. Il s'agit de données brutes, et seuls des experts métier, ayant l'habitude d'être confrontés à ces données, sont donc capables d'en tirer des conclusions et de prévenir certains événements. Il s'agit d'une connaissance acquise qui peut être difficile à formaliser et à transmettre, et qui est limitée à certains "motifs" identifiés par ces personnes. Ainsi, cette vaste masse de données n'est finalement que peu ou pas exploitée à posteriori.

L'étude était relativement large et couverte par des clauses de confidentialité. Dans ce papier, nous dévoilons, en accord avec l'entreprise, une partie de l'analyse des données issues de l'exploitation du tunnelier, dans le but d'aller au delà des applications de monitoring utilisées pour l'instant. Elle offrait un cadre intéressant pour la valorisation des données. De plus, pouvoir prédire certains événements de la vie du chantier à partir de ces données peut s'avérer crucial pour l'entreprise, car le moindre incident peut entraîner des retards considérables pour le chantier, et les conséquences financières, voire humaines, peuvent être importantes.

Ainsi, nous avons analysé ces données, pour voir si des modèles d'apprentissage automatique pouvaient permettre de prédire certains de ces événements importants pour la vie du chantier. De telles prédictions pourraient ensuite être utilisées pour créer de nouvelles alertes, ce qui permettrait d'anticiper les incidents. Ainsi, ces prédictions permettraient d'automatiser certaines intuitions que peuvent avoir les experts du métier vis-à-vis des données, mais aussi d'identifier de nouveaux motifs pertinents dans les données. Par exemple, nous pouvons relier la vitesse d'avancement du tunnelier avec les différents paramètres du sol pour prédire des événements pendant le passage du tunnelier.

Nous allons donc à présent présenter la manière dont des prédictions ont été faites à partir des données du tunnelier, de la préparation des données à la construction de modèles et l'analyse de leurs résultats. Nous tirerons ensuite les leçons de cette étude, riche en enseignements.

1 PRÉDICTIONS À PARTIR DE DONNÉES D'UN TUNNELIER

Une multitude de sources de données a été livrée au lancement de l'étude, avec des données de différents chantiers de tunnelier. Les différentes sources sont rappelées ci-dessous :

- des fichiers "Datx", un format binaire propriétaire à l'industriel
- un "dump" de base de données
- des planches de consigne (fichiers pdf) ; elles donnent des instructions de pilotage (vitesse, direction, risques) au pilote du tunnelier ;
- des descriptions textuelles de quelques sondages et des rapports de creusement (en général des captures d'écran sous forme d'images) ; un sondage dans un point (x, y) décrit en détail la composition du sol jusqu'à une profondeur donnée ; ces sondages permettent de construire le modèle géotechnique correspondant au chantier ;
- des rapports d'anneau (capture d'écran sous forme d'images) ; un anneau est l'unité élémentaire d'avancement du tunnelier ; il correspond à la longueur des traverses de béton à utiliser pour consolider le tunnel après le passage de la coupe du tunnelier (ex. 1 m) ; après la pose de chaque anneau un rapport est rédigé pour décrire les divers événements qui se sont produits pendant l'excavation ;

La première constatation a été que ces données n'étaient pas toutes exploitables et qu'il fallait trier celles qui pouvaient l'être ou pas !

Figure 1: Photo d'un tunnelier

Dans le cadre de l'étude, nous nous sommes focalisés sur deux types de support :

- Des fichiers *Datx* dans un format spécifique aux tunneliers utilisés par l'entreprise. Le système d'acquisition des données du tunnelier stocke chaque seconde une trame de données, qui correspond à l'ensemble de données de chaque capteur du tunnelier pour cette seconde. Ces trames sont sauvegardées dans ces fichiers particuliers, dont l'encodage permet de les compresser. Le tunnelier comporte des milliers de capteurs, ce nombre variant en fonction des chantiers, voire parfois au cours du temps de la réalisation du tunnel. Au total nous avons récupéré plus de 40 Go de données. Le format étant particulier, nous avons extrait les données dans des fichiers *csv* afin de pouvoir les exploiter et reconstruire une base de données.
- Une base de données correspondant à une agrégation des données contenues dans les fichiers décris précédemment (des agrégations toutes les dix secondes). L'accès à une base de données à l'avantage de permettre de manipuler facilement les données disponibles grâce notamment à des requêtes SQL. De plus cette base contenait des informations supplémentaires avec notamment des mesures effectuées sur le chantier, et quelques événements, ce qui ouvrait la voie à l'apprentissage pour essayer de corréler ces événements et mesures avec les données du tunnelier. La structure de la base était cependant particulière, puisque les données du tunnelier étaient stockées dans une seule table comportant 1250 colonnes, c'est à dire une colonne par capteur. De plus, le nommage des attributs de cette table n'était pas explicite, les noms étant *Data001*, *Data0002* et ainsi de suite jusqu'à *Data1250*. Enfin, les contraintes entre les différentes tables de la base (comme les différentes clés par exemple) n'étaient pas spécifiées dans le schéma, ce qui ajoutait une difficulté supplémentaire pour lier les données de la table du tunnelier aux tables liées à des mesures ou des événements.

Pour la suite nous avons choisi de porter l'étude sur une mesure effectué le long du tracé du tunnelier, que l'on nommera par la suite $var1$, pour des raisons de confidentialité. Cette mesure est cruciale pour la sécurité du chantier et des alentours, et pouvoir prédire sa valeur pourrait avoir des répercussions importantes sur le chantier, en permettant de prévenir certains accidents avant qu'ils ne surviennent.

1.1 Préparation d'un jeu de données

Clairement, les données n'étaient pas directement exploitables : il était nécessaire de construire un jeu de données afin de pouvoir appliquer des algorithmes d'apprentissage automatique. Afin d'obtenir une granularité temporelle de l'ordre de la seconde, nous avons utilisé une base de données reconstruite à partir des *Datx* et importé les autres données utiles à partir de leur base de données.

Afin de construire un jeu de données, nous avons d'abord discuté avec des experts du métier afin de mieux comprendre les paramètres du tunnelier. Cela était d'autant plus crucial que le nommage des attributs dans la base ne permettait pas de les interpréter. Après discussion, nous sommes parvenus à établir une liste de 93 paramètres, parmi les 1250 disponibles, pouvant être pertinents vis-à-vis de l'étude. Nous avons également obtenu le nom de ses attributs dans la base, ainsi que leur dénomination réelle (par exemple force de poussée, pression, volume de béton injecté...). Cela a ainsi permis de réduire le nombre d'attributs du jeu d'apprentissage qui ne contiendrait donc que ces 93 attributs. Les autres attributs non utilisés étaient pour la plupart dérivables à partir de ces 93 attributs, car ils correspondaient à des conversions dans d'autres unités, ou à des variables calculées.

Une fois ces attributs identifiés, vient une étape plus délicate, visant à lier ces attributs à un événements ou une mesure ($var1$). Le paramètre cible identifié, ainsi que la table de la base de données dans laquelle il est stocké, il a été nécessaire de réfléchir à la manière de lier les mesures de cette variable avec les enregistrements des paramètres du tunnelier, dans le but d'avoir un jeu de données "classique", où l'on souhaite prédire la valeur d'un attribut en fonction des autres données. Or, les contraintes de la base de données n'étant pas spécifiées, la jointure entre les deux tables n'était pas évidente de prime abord. Il fallait identifier comment faire le lien entre les paramètres du tunnelier et $var1$. Ce lien a finalement été fait grâce au *point métrique* [1], qui correspond à la position dans le tunnel à laquelle un paramètre du tunnelier a été enregistré (position de la roue de coupe avant), et où $var1$ a été relevée. Il faut cependant noter que $var1$ n'est mesurée qu'à certains endroits du tunnel de manière ponctuelle (environ 1m60 de distance entre deux mesures successives), tandis que, compte tenu de la très faible vitesse d'avance du tunnelier, à un seul point métrique correspond de très nombreux enregistrements dans la base de données. Ainsi, un tuple de la table contenant les valeurs de table de $var1$ correspond à plusieurs centaines de lignes de la table du tunnelier. De plus, le point métrique de $var1$ est moins précis que celui d'une ligne de paramètres du tunnelier. Enfin, au vu de la taille du tunnelier, il semble possible que les paramètres du tunnelier aux alentours d'un point métrique donné puissent également avoir une influence sur la valeur de $var1$. Ainsi, le jeu de données a été construit de la manière suivante :

- Pour chaque ligne de paramètre du tunnelier correspondant à un point métrique pour lequel une mesure de $var1$ existe, on associe la valeur de $var1$ correspondante.

- On procède ensuite à une propagation de la valeur de $var1$ considérée aux alentours du point métrique, c'est à dire pour les lignes d'enregistrement du tunnelier ayant un point métrique proche de celui considéré. À un point métrique pm fixé pour une valeur de propagation x donnée, on associe la valeur de $var1$ à toutes les lignes d'enregistrement du tunnelier pour lesquelles le point métrique est compris dans l'intervalle $[pm - x; pm + x]$. Dans le cadre de l'étude nous avons fixé $x = 10cm$.

Ce procédé est illustré par la figure 2, qui montre bien la méthode de propagation de la mesure. Une fois ce processus achevé, on obtient alors un jeu de données grâce auquel on peut construire un modèle afin de prédire la valeur de $var1$ en fonction des autres attributs du tunnelier.

Il demeure cependant une dernière subtilité. En effet, afin d'éviter le sur-apprentissage, pour chaque modèle testé, 80% du jeu initial a été dédié à la construction du modèle, les 20% restant étant alloués à sa validation. Or, une division aléatoire du jeu de données initial conduirait, pour un même point métrique (ou ses alentours immédiats), à avoir certains des enregistrements du tunnelier à la fois dans le jeu d'apprentissage et le jeu de validation. Or les valeurs aux alentours d'un même point métrique sont très similaires, et le jeu d'apprentissage et de validation seraient alors quasiment identiques. Il a donc été décidé que toutes les lignes du jeu de données concernant le même voisinage de point métrique seraient toutes utilisées dans le même jeu, soit dans celui d'apprentissage, soit dans celui de test.

1. Un point métrique représente la distance précise par rapport à un point précisé - en général le point de départ du tunnelier.

1.2 Algorithmes d'apprentissage

Dans le cadre de cette étude, tout le code a été développé en *Python 3*, les algorithmes pour l'apprentissage provenant de la librairie *scikit-learn* [5]. Comme SGBD, nous avons utilisé une base de données *MySQL*.

Afin de caractériser $var1$, nous avons testé plusieurs approches, qui vont à présent être décrites.

1.2.1 Régression. $var1$ étant une variable continue, nous avons commencé par effectuer de la régression afin de prédire la valeur de $var1$ en fonction des paramètres du tunnelier. Nous avons commencé par le plus simple, à savoir une régression linéaire. Afin de pouvoir comparer différents modèles, nous avons également appliqué la méthode des K Nearest neighbors (KNN) avec pondération des valeurs des points voisins [4]. Avec cette méthode, la valeur de $var1$ d'une nouvelle donnée est égale à une pondération des valeurs de $var1$ de ses K plus proches voisins. La pondération est basée sur la distance entre le point considéré et ses voisins, il est donc important de définir la mesure de distance (dans ce cas précis, la distance euclidienne).

1.2.2 Classification. Dans un second temps, il a été décidé d'appliquer des méthode de classification pour voir si cela impactait la qualité des prédictions. Il a donc fallu discrétiser $var1$, ce qui a pu se faire facilement en discutant avec des experts du tunnelier. En effet, ils utilisent déjà eux-mêmes une échelle discrète pour qualifier les valeurs prises par $var1$, et nous avons donc pu reprendre directement ces catégories, qui sont au nombre de trois :

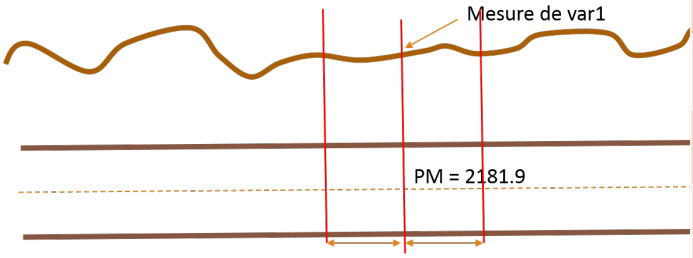
- Positif : lorsque la valeur de $var1$ est positive
- Négatif faible : si $var1$ est comprise entre 0 et -1000
- Négatif modéré : entre -1000 et -2000

D'autres catégories peuvent exister sur les chantiers, mais n'ont pas été rencontrées dans le cadre du chantier considéré pour l'étude. Cette catégorisation a été simple à mettre en place, mais a révélé un déséquilibre entre les classes. Enfin, s'il existe un nombre assez important de valeurs positives, la catégorie des négatives faibles reste largement majoritaire : cela signifie que seuls ces valeurs faibles risquent d'être réellement bien prédites, alors même que leur prédiction n'apporte que peu d'intérêt car cela indique un déroulement normal du creusement du tunnel.

Malgré tout, nous avons appliqué plusieurs techniques de classification aux données. Nous avons commencé par réutiliser l'algorithme de KNN, mais cette fois pour la classification : pour une donnée, on lui attribue la classe majoritaire parmi celles de ses K plus proches voisins.

Nous avons également effectué une régression logistique, qui permet de mesurer l'association entre des paramètres d'entrée (du tunnelier) et une variable de sortie discrète (la catégorie du classement) [2].

De plus, nous avons étudié des arbres de décision [6], qui se construisent comme des graphes permettant au fur et à mesure,



Point métrique	Attribut 1	Attribut 2	...	Attribut 93	Var1
2181.8	2	0.67	...	0.97	
2181.9	10	1.04	...	0.76	-500
2181.9	5	1.03	...	0.98	-500
2182.0	4	1.42	...	0.42	

Figure 2: Construction du jeu de données par propagation aux alentours du point métrique de la mesure

grâce à des décisions sur un attribut, de séparer les données en groupes jusqu'à leur attribuer une classe. Pour aller un peu plus loin, nous avons également utilisé des forêts d'arbres de décision (random forest), qui se basent sur plusieurs arbres de décision, ayant chacun été entraînés sur des sous-ensemble de données légèrement différents. A la fin, un vote entre les différents arbres permet de retenir la classe majoritairement prédite. De plus, nous avons également utilisé des machines à vecteurs de support [7]. Nous avons cependant rencontré un problème de convergence, peut-être dû au grand nombre de paramètres disponibles pour le modèle, ou la trop grande variété de données.

Enfin, dans toutes les méthodes étudiées, nous avons essayé de mettre en place une sélection automatique des attributs, en utilisant une méthode heuristique connue, appelée *forward selection* [1]. L'objectif étant de déceler de manière automatique des attributs qui seraient plus pertinents parmi les 93 initialement identifiés avec les experts du métier, et ainsi diminuer la dimensionalité des données.

1.3 Résultats

1.3.1 Mesures de performance. L'utilisation de différents algorithmes nous a conduit à obtenir plusieurs modèles, et pour déterminer le plus adapté, nous avons du définir des mesures de performance pour pouvoir les comparer entre eux.

Pour la régression, nous avons utilisé une mesure statistique appelée coefficient de détermination (et noté R^2) [3], défini par la formule suivante :

$$R^2 = 1 - \frac{\sum_i \hat{y}_i - \bar{y}}{\sum_i y_i - \bar{y}}$$

avec $\hat{y}_i$ la valeur prédite, y_i la valeur mesurée et $\bar{y}$ la moyenne des valeurs

Pour la classification, nous avons utilisé tout simplement le ratio de bonnes prédictions par rapport au nombre de prédictions total. Cependant, pour pouvoir étudier la manière dont chaque classe est prédite, nous avons également utilisé des matrices de confusion afin de regarder si certaines catégories étaient mieux prédites que d'autres (ce qui était fort probable au vu du déséquilibre entre les classes). Les résultats pour la régression et la classification sont résumés dans la table 1.

Méthode	Tous les attributs	Forward selection
Régression (coefficient de détermination)		
KNN	0.29	0.55
Régression linéaire	0.003	0.51
Classification (prédictions correctes/prédictions totales)		
KNN	0.48	0.76
Régression logistique	0.75	0.66
Arbre de décision	0.70	0.71
Forêt d'arbres de décisions	0.79	0.81

Table 1: Récapitulatif des résultats obtenus pour la prédiction de $var1$

1.3.2 Récapitulatif. D'une manière générale, on peut constater que la méthode de forward sélection semble donner de bien meilleures performances. En terme de classification, la forêt d'arbres et la régression logistique se détachent sensiblement des deux autres méthodes. On peut noter que KNN fonctionne beaucoup mieux avec la méthode de *forward sélection*, la réduction de la dimensionalité des données permettant d'obtenir des distances plus significatives.

Pour la classification, les matrices de confusion permettent de comparer pour chaque catégorie la répartition des prédictions ; dans l'idéal, les pourcentages sur la diagonale doivent être les plus élevés possible, car cela signifie que majoritairement, les données de la classe sont correctement prédites. Nous avons choisi de donner ici celles des deux méthodes donnant les résultats les plus probants. Il s'agit des matrices dans le cas où le score présenté dans la table 1 est le meilleur.

		Classe prédite		
		Positif	faible	moyen
Classe	positif	55%	40%	5%
	Faible	8%	91%	1%
	Moyen	60%	38%	2%

Table 2: Matrice de confusion pour la forêt d'arbres de décision

		Classe prédite		
		Positif	faible	moyen
Classe	positif	62%	36%	2%
	Faible	8%	88%	4%
	Moyen	43%	48%	9%

Table 3: Matrice de confusion pour la régression logistique

On voit donc nettement que les valeurs faibles sont mieux prédites, ce qui est logique car comme ils sont beaucoup plus nombreux, ils influencent beaucoup plus le modèle. Les tentatives de rééquilibrage n'ont pas donné de résultats probants, et ont même conduit à de moins bonnes performances. Nous avons notamment essayé de prendre la classe avec le moins de données, et pour chaque autre classe, prendre seulement autant de données que cette classe minimale. Malheureusement, le déséquilibre était vraiment trop important. Cela explique également que les valeurs négatives moyens, très faiblement présentes dans le jeu de données initial, ne puissent être mieux prédites.

Enfin, pour avoir une idée de la manière dont la classification s'effectue, les arbres de décision permettent de comprendre et de visualiser les décisions prises par l'algorithme. Cela permet, au delà des performances, d'avoir une idée des règles qui régissent les décisions, et de les confronter à l'expérience concrète des personnes qui arrivent, par l'expérience du terrain, à faire des prédictions. Ce type de modèle est ainsi plus parlant et permet une bonne base de discussion avec des experts métiers.

2 LEÇONS SUR LES DONNÉES

Dans cette section, nous tirons les leçons avec un regard académique sur cette étude, corroborées par d'autres études réalisées par les auteurs.

Sur la posture tout d'abord, il s'agit bien de s'intéresser à un problème réel pour *faire du sens à partir des données* en temps limité. Qu'il s'agisse de compétences en bases de données, en apprentissage automatique ou plus globalement en informatique, l'essentiel est donc de proposer des techniques si possible à la pointe de l'innovation et d'être force de proposition. Les compétences verticales et pointues requises pour un travail de recherche classique en informatique façonnent notre capacité à prendre en compte les problèmes posés et à y apporter des réponses, mais ne sont que rarement "mobilisables" en l'état. Cela nous oblige aussi à aller un peu au delà de notre zone de confort, générant un peu d'adrénaline utile.

D'un point de vue plus technique, la *récupération des données* s'est avérée beaucoup plus complexe que prévue car elle impliquait des aller/retour avec les prestataires plus ou moins enclins à répondre aux sollicitations, des déplacements sur site pour récupérer le disque dur de tous les fichiers (des images du terrain ou de croquis faits à la main, documents en pdf, des tableurs, bases de données). Par ailleurs, cela a permis de se poser des questions simples : à qui appartiennent les données ? Où sont-elles stockées ? Quelle législation s'applique ? Dans un contexte de collaboration forte (analyse préalable réalisée par des entreprises spécialisées, location du tunnelier auprès d'un prestataire ...), ces questions deviennent très rapidement épineuses et complexes, mais finissent par se résoudre.

Au cours de l'étude, il y a aussi pu avoir des craintes des spécialistes du métier de se voir déposséder d'une partie de leur connaissance. Les méthodes d'analyse de données permettent parfois d'expliciter la connaissance implicite des experts, rarement formalisés et engendrent donc des réticences.

Ensuite, nous retenons qu'il faut avoir une forme d'*empathie pour les données*. Les informaticien.ne.s ont tendance à les voir comme des "entités" purement abstraites, – ce qu'elles sont par ailleurs – et donc négligent leur signification et le contexte de leur acquisition. Comprendre pourquoi l'entreprise"fait comme cela" est souvent utile et permet d'expliquer les raisons de telle ou telle structuration ou choix technologique. Cette empathie est aussi un bon préalable pour avoir les éléments de langage entre l'informaticien et l'industriel. En un temps relativement court, il s'agit bien de reconstruire une partie de l'expertise de l'entreprise, ce qui est passionnant à bien des égards !

Enfin, le dernier enseignement important concerne le thème de *la qualité des données*, notion vraiment cruciale qui se trouve être extrêmement difficile. Clairement, il ne s'agit pas que de prendre des données de mauvaise qualité pour leur appliquer telle ou telle cure de nettoyage algorithmique pour le résoudre. C'est souvent beaucoup plus profond car mettant en cause les processus internes de l'entreprise, principalement lors de l'acquisition des données. Comme indiqué par Gartner [2], la qualité des données n'est pas un problème qui relève uniquement de l'informatique, mais un problème qui relève du business. Par exemple, imposer à un conducteur de tunnelier de notifier les erreurs dans un journal de bord ne relève pas de la compétence des informaticien.ne.s. Tout au plus, nous pouvons fournir les bonnes applications pour faciliter cette acquisition. Cela a des répercussions sur le métier de ces personnes et va donc bien au delà de l'informatique comme discipline scientifique et technique.

3 CONCLUSION

Cette étude de six mois, relativement courte dans le temps académique, était relativement conséquente dans le temps de l'entreprise. Nous avons obtenu une compréhension globale des données générées par un tunnelier en identifiant les points importants liés à leur acquisition. Sur les possibilités liées à l'apprentissage de modèles à partir des données pour différents incidents, les résultats ont validé l'intérêt de monter en compétence sur la valorisation des données générées par un tunnelier.

Cette étude a contribué à donner naissance à l'offre `DataValor` du laboratoire LIRIS (UMR 5208 CNRS) portée par `InsaValor`, la société de transfert et de valorisation de l'INSA de Lyon. L'objectif est

2. 'Dirty Data' is a Business Problem, Not an IT Problem www.gartner.com/newsroom/id/501733

de faciliter le transfert des compétences sur les données (entendues au sens large et incluant l'image, le son et la vidéo) du monde académique vers le monde industriel, aidé par un ingénieur transfert payé sur ressources propres. Cette offre permet d'élargir l'assiette des relations partenariales du LIRIS en permettant des études de courte durée (typiquement de 2 à 6 mois), répondant ainsi à une des attentes forte du monde industriel et fortement encouragées par le législateur.

RÉFÉRENCES

[1] Rich Caruana and Dayne Freitag. 1994. Greedy Attribute Selection. In *In Proceedings of the Eleventh International Conference on Machine Learning*. Morgan Kaufmann, 28–36.

[2] D. Freedman. 2005. *Statistical Models : Theory and Practice.* Cambridge University Press.

[3] Nico JD Nagelkerke. 1991. A note on a general definition of the coefficient of determination. *Biometrika* 78, 3 (1991), 691–692.

[4] Amir Navot, Lavi Shpigelman, Naftali Tishby, and Eilon Vaadia. 2005. Nearest Neighbor Based Feature Selection for Regression and its Application to Neural Activity. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*. 996–1002.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn : Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[6] J. R. Quinlan. 1986. Induction of decision trees. *Machine Learning* 1, 1 (1986), 81–106. https://doi.org/10.1007/BF00116251

[7] B. Schölkopf and AJ. Smola. 2002. *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA. 644 pages.